



Anais do **WTDCGPI 2008**

**Workshop de Teses e Dissertações em Computação
Gráfica, Processamento de Imagens e Visão
Computacional**

**Evento Integrante do SIBGRAPI 2008
XXI Simpósio Brasileiro de Computação Gráfica e Processamento
de Imagens**

**Campo Grande, MS, Brasil
12 a 15 de outubro de 2008**



Índice

Prefácio.....	iii
Comitês.....	iv
Revisores	v
Computação Gráfica	6
Discrete Models for Animating Gas-Liquid and Fluid-Surface Interactions	7
Exploration of Volumetric Datasets through Interaction in Transfer Function Space	17
Fluid Warping	27
Image-Based Techniques for Surface Reconstruction of Adaptively Sampled Models	33
Least Squares and Point-based Surfaces: New Perspectives and Applications	43
Renderizações Não Fotorealísticas para Estilização de Imagens e Vídeos usando Areia Colorida	53
Sistema Composto para Amostragem e Geração de Luzes a partir de Mapas de Iluminação	63
Um Sistema Simplificado para Animação Física usando Malhas Tetraedrais	69
Processamento de Imagens e Visão Computacional.....	79
Automatização do Ajuste de Segmentadores Neurais de Imagens Coloridas	80
Contrast Enhancement in Digital Imaging using Histogram Equalization	86
Fusão de Métodos Baseados em Minúcias e em Cristas para Reconhecimento de Impressões Digitais	96
High-resolution image reconstruction using the Discontinuity Adaptive ICM algorithm	106
Mapeamento e Monitoramento Ambiental Usando Imagens Aéreas de Pequeno Formato	116
Reconhecimento Semi-Automático de Sinus Frontais para Identificação Humana Forense Baseado na Transformada Imagem-Floresta e no Contexto da Forma	126
Shape Descriptors based on Tensor Scale	136

Prefácio

O Workshop de Teses e Dissertações em Computação Gráfica, Processamento de Imagens e Visão Computacional (WTDCGPI) visa criar um espaço para debate e divulgação dos trabalhos de doutorado e mestrado desenvolvidos no país em Computação Gráfica, Processamento de Imagens, Visão Computacional e Visualização. O evento tem sido parte integrante do SIBGRAPI – Simpósio Brasileiro de Computação Gráfica e Processamento de Imagens, que em 2008 ocorre na cidade de Campo Grande, MS, no período de 12 a 15 de outubro de 2008.

Foram aceitas submissões de trabalhos de alunos regularmente matriculados em cursos de doutorado ou mestrado vinculados a cursos de Pós-Graduação stricto sensu no país, ou ainda ex-alunos que tenham defendido tese ou dissertação depois de 01/01/2007 e antes do prazo final para submissão em 09/07/2008.

Como forma de permitir uma maior disseminação dos benefícios do Workshop aos autores, decidiu-se por aceitar o maior número possível de trabalhos, respeitando os pareceres dos revisores (pelo menos 2 aceitações). Ao todo, foram aceitos 8 trabalhos na área de Computação Gráfica e 7 na área de Processamento de Imagens e Visão Computacional de um total de 18 submetidos.

Os trabalhos aceitos na área de Computação Gráfica versam sobre temas variados tais como: visualização e reconstrução a partir de modelos baseados em pontos, síntese de fontes de luz, visualização volumétrica, deformação de imagens e simulação e animação baseada em física.

Os trabalhos aceitos na área de Processamento de Imagens e Visão Computacional também contemplaram temas variados, incluindo a segmentação de imagens coloridas, realce de imagens, biometria a partir de impressões digitais, processamento de imagens aéreas, médicas e biológicas, bem como métodos para representação e análise de formas.

Claudio Esperança
Herman Martins Gomes

WTDCGPI 2008 Co-Chairs

Comitês

Workshop Co-Chairs

Claudio Esperança, UFRJ

Herman Martins Gomes, UFCG

Comitê de Programa

Carlos Morimoto, USP

Claudio Esperanca, UFRJ

Esteban Clua, UFF

Herman Martins Gomes, UFCG

Joao Marques de Carvalho, UFCG

João Comba, UFRGS

Luiz Henrique de Figueiredo, IMPA

Nelson Mascarenhas, UFSCar

Olga Bellon, UFPR

Waldemar Celes, PUC-Rio

Wu Shin-Ting, UNICAMP

Revisores

Arnaldo de Albuquerque, UFMG

Carlos Morimoto, USP

Chauã Queirolo, UFPR

Claudio Esperanca, UFRJ

Esteban Clua, UFF

Harlen Batagelo, UNICAMP

Herman Martins Gomes, UFCG

Joao Carvalho, UFCG

João Comba, UFRGS

Julio d'Alge, INPE

Junior Barrera, USP

Luciano Silva, UFPR

Luiz Henrique de Figueiredo, IMPA

Murillo Homem, UFSCar

Nelson Mascarenhas, UFSCar

Olga Bellon, UFPR

Waldemar Celes, PUC-Rio

Wu Shin-Ting, UNICAMP

Computação Gráfica

Discrete Models for Animating Gas-Liquid and Fluid-Surface Interactions

Sicilia F. Judice

Gilson A. Giraldi

National Laboratory of Scientific Computing

Ave Getúlio Vargas, 333, 25651-075, Petrópolis, RJ, Brasil

{sicilia,gilson}@lncc.br

Abstract

The past two decades showed a rapid growing of physically-based modeling of fluids and solid for computer graphics applications. In particular, techniques in the field of Computational Mechanics have been applied for realistic animation of systems that involve gas-fluid and fluid-surface interaction for computer graphics and virtual reality applications. The main goal of our work is the development of a particle based framework to create realistic animations of such systems. Specifically, we model and simulate the gas through a Lattice Gas Cellular Automata (LGCA), the liquid by Smoothed Particle Hydrodynamics (SPH) method and the surface through Mass-Spring systems. LGCA is a discrete model based on point particles that move on a lattice, according to suitable rules in order to mimic a fully molecular dynamics. SPH is a Lagrangian, meshfree method for numerical simulation which is based on particle systems and interpolation theory. Mass-Spring systems may be geometrically represented by regular meshes which nodes are treated like mass points and each edge acts like a spring. When combining these methods (LGCA, SPH and Mass-Spring), we get the advantage of the low computational cost of cellular automata and mass-spring systems and the realistic fluid dynamics inherent in the SPH to develop a new animating framework for computer graphics applications. In this work, we discuss the theoretical elements of our proposal and present some preliminary experimental results.¹

1. Introduction

Physically-based techniques for the animation of natural elements like fluids (gas or liquids), flood, elastic, plastic and melting objects, among others, have taken the attention of the computer graphics community [25, 36, 11].

In particular, techniques in the field of Computational Fluid Dynamics (CFD) have been applied for fluid animation in applications that involve fluid-fluid and fluid-surface interaction [35, 28, 19, 8, 31].

A common approach in this area relies on top down viewpoints that use 2D/3D mesh based techniques in conjunction with fluid/solid equations [8]. Other possibility is to apply Mass-Spring systems to model the elastic object [19], SPH to simulate fluids [28, 34] and Lattice Gas Cellular Automata (LGCA) techniques to simulate gas systems [21, 3].

Mass-Spring models are well suited to animation due to their flexibility to handle non-rigid solid properties, its easy manipulation and implementation. Besides, Mass-Spring models can be faster than their counterpart in continuous mechanics, and so, more suitable for real time applications specially when GPU capabilities are explored [33]. The LGCA is a *bottom up* discrete model, based on point particles that move on a lattice, according to suitable and local rules, that mimics a fully particle dynamics [17]. These methods are cheaper and more stable than the traditional ones for fluid simulation, because there is no need to solve Partial Differential Equations (PDEs) to obtain a high level of description [3]. SPH is a Lagrangian, meshfree method for numerical simulation which is based on particle systems and interpolation theory.

In this paper we propose a framework for animation of fluid-fluid and fluid-surface interaction based on Mass-Spring systems, LGCA techniques and SPH. We apply a traditional Mass-Spring model described in [33] that can be used for both surface and deformable solid modeling. Among the LGCA models [10] we apply the FHP one which was introduced by Frisch, Hasslacher and Pomeau [18] in 1986. The traditional FHP is a model for two-dimensional fluids that describes the motion of particles traveling in a discrete space and colliding with each other. The space is discretized in a hexagonal lattice. We proposed a three dimensional fluid simulation model based on the FHP and interpolation techniques. The SPH implementation follows the method presented in [34]. Finally, render-

¹ Full paper of M.Sc thesis.

ing techniques must be applied to ensure the desired level of realism or visual effect. In this step we apply the photon map [27] method because it is able to easily generate area light sources, color bleeding, soft shadows, indirect illumination and caustics, once it collects illumination information of the scene by a pre-trace from light sources.

The main contributions of this work are the 3D fluid simulation model based on the FHP and the development of a particle based framework, combining Mass Spring systems, LGCA techniques and SPH, to create realistic animations of systems that involve gas-fluid and fluid-surface interaction for computer graphics and virtual reality applications.

The paper is organized as follows. Section 2 gives a survey of related works. Section 3 offers some fundamental concepts in Navier-Stokes equations. Section 4 describes the SPH method. Section 5 describes the FHP model and our extension for 3D. In section 6 we present the Mass-Spring model used. The rendering model is discussed on section 7. Section 8 describes the proposed framework and some preliminary results. In section 9 we present the conclusions.

2. Related Works

The main focus of this work is the animation of gas-fluid and fluid-surface interaction. The former can be modeled by a FHP method in which there are two kinds of particles - gas particles and liquid particles [2]. Then, Boltzmann approximation is applied to calculate the surface tension as a function of population density.

In [43] an interface model is derived for two-phase flows with surface tension, density and viscosity differences between the phases. The derivation starts from the balance equations for a sharp interface and uses an ensemble averaging procedure on an atomic scale to obtain a diffuse interface version of the equations [12, 43]. Other approach models the interfacial tension is to detect the interfacial surface (boundary between the gas and liquid phases) and then to compute the interaction forces following some heuristic.

Fluid-surface interaction includes: (1) Representation of the object geometry; (2) The modeling of mechanical behavior of elastic surfaces; (3) A suitable model for fluid simulation; (4) A model for interaction of the flow with the object; (5) Rendering issues.

Bidimensional manifolds can be represented by using implicit surfaces [47], triangulated meshes or subdivision surfaces with local parameterization for representation [22, 42]. The mechanical behavior of deformable surfaces (item (2)) can be described by continuum elasticity models that describes how the objects deform under applied forces. Other possibility is to apply discrete models, based on Mass-Spring systems [19, 33]. In this case, the object geometry is represented by a mesh and its nodes are treated

like mass points while each edge acts like a spring connecting two adjacent mass points. It is known that methods that are based on the continuum mechanics are more realistic than their discrete counterparts [44]. However, Mass-Spring models can be faster, and so, more suitable for real time applications [44, 14].

The item (3) involves numerous works that can be coarsely classified in non-physically and physically based models [25, 11]. Our work belongs to the later class, which can be subdivided in PDEs and Lattice based techniques [17, 25].

PDEs methods involve continuous fluid equation, like the Navier-Stokes ones, and numerical techniques based on discretization approaches that can be Lagrangian Smoothed Particle Hydrodynamics (SPH) [29], method of characteristics [41], Moving-Particle Semi-Implicit [37] or Eulerian (Finite Element) ones [16].

Lattice based techniques, like HPP, FHP and Lattice Boltzmann methods, work following a different viewpoint [17, 5]. For instance, in the case of HPP and FHP, instead of applying continuous mechanics (and, consequently, PDEs) principles, they model the system as a set of point particles, that move on a lattice, interacting according to suitable and simple rules in order to mimics a fully dynamics [17]. These are *bottom up* approaches in which the macroscopic behavior of the fluid can be recovered by multiscale techniques [17].

Lattice models have a number of advantages over more traditional numerical methods, particularly when fluids mixing and phase transitions occur [38]. The simulation is always performed on a regular grid and can be efficiently implemented on a massively parallel computer. Solid boundaries and multiple fluids can be introduced in a straightforward manner and the simulation is performed equally efficiently, regardless of the complexity of the boundary or interface [7]. In addition there are not numerical stability issues because the evolution follows integer arithmetic. However, system parameterization (viscosity, for example) is a difficult task in such lattice models and they are less realistic than PDE based models.

The item (4), interaction between deformable manifolds and fluids, can be addressed by hybrid methods in which the fluid is a continuum medium, simulated by Navier-Stokes plus SPH or grid based techniques, and the surface is represented as a discrete one [19, 40, 4]. These approaches deal with the specific problem of preventing the leaking of fluid across the polygonal surface [22, 6]. In addition, fluid flows can be simulated on 2D manifolds represented by (continuous) subdivision surfaces that have a natural quad patch parametrization [42]. Interaction between Navier-Stokes fluids and digital terrain models is another subclass of fluid-surface interaction [46]. Besides, a hybrid

particle and implicit surface approach to simulating water was proposed in [15], which led to the particle level set method of [13].

Finally, visualization and rendering techniques must be applied to ensure the desired level of realism or visual effect. Photo-realistic rendering can properly account through several algorithms including path tracing, bidirectional path tracing [23], Metropolis light transport [45], and photon map [27]. The interested reader is also encouraged to browse interesting reviews in this area [1, 25].

In this paper, the proposed method for gas-liquid interaction is a hybrid one in the sense that the gas is simulated by a discrete system and the fluid is animated by Navier-Stokes plus SPH. On the other hand, the fluid-surface interactions model is a discrete one because the fluid is FHP-based and the surfaces are represented by polygonal meshes, or Mass-Spring systems in the case of deformable manifolds.

3. Navier-Stokes for Fluid Animation

The majority fluid models in computer graphics follow the Eulerian formulation of fluid mechanics that is based on a top down viewpoint of the nature: the fluid is considered as a continuous system subjected to Newton's and conservation Laws as well as state equations connecting the macroscopic variables of pressure P , density ρ and temperature T . So, the mass conservation, also called continuity equation, is given by [24]:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{u}) = 0 \quad (1)$$

The linear momentum conservation equation, also called Navier-Stokes, can be obtained by applying the third Newton's Law to a volume element dV of fluid. For incompressible flows it can be written as [24]:

$$\rho \left(\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} \right) = -\nabla P + \mathbf{F} + \mu \nabla^2 \vec{u}, \quad (2)$$

$$\nabla \cdot \vec{u} = 0. \quad (3)$$

where $\mathbf{F}$ is an external force field and μ is the viscosity of the fluid. Also, we may need an additional equation for the pressure field. This is a state equation which ties together all of the conservation equations for continuum fluid dynamics and must be chosen to model the appropriate fluid (*i.e.* compressible or incompressible). In the case of liquids, the pressure P is temperature insensitive and can be approximated by $P = P(\rho)$. Morris in [32] proposed an expression that have been used for fluid animation also [34]:

$$P = c^2 \rho, \quad (4)$$

where c is the speed of sound in this fluid [39]. Equations (2)-(4) need initial conditions

$(\rho(t=0, x, y, z), \vec{u}(t=0, x, y, z))$ as well as boundary conditions, like the usual no-slip one: $\vec{u}|_S = 0$.

4. Smooth Particle Hydrodynamics

In this section we follow the references [39, 30]. The two fundamental elements in the Smoothed Particle Hydrodynamics method are an interpolation kernel W and a particle system that represents a discrete version (sample) of the fluid. The kernel estimate of a scalar quantity $A(\mathbf{r})$ is defined by:

$$\langle A(\mathbf{r}) \rangle = \int_{Space} A(\mathbf{r}') W(\mathbf{r} - \mathbf{r}', h) d\mathbf{r}', \quad (5)$$

where the function $W(\mathbf{r} - \mathbf{r}', h)$ is an (interpolation) kernel which must satisfies the following properties [30]:

1) Volume conservation:

$$\int_{Space} W(\mathbf{r} - \mathbf{r}', h) d\mathbf{r}' = 1, \quad (6)$$

2) The kernel W should satisfy the Dirac delta function in the limit:

$$\lim_{h \rightarrow 0} W(\mathbf{r} - \mathbf{r}', h) = \delta(\mathbf{r} - \mathbf{r}'). \quad (7)$$

If we take a sampling of A then the $A(\mathbf{r}')$ in equation (5) will be known only at a discrete set of N points $\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_N$. Hence, thorough properties (6)-(7) it is possible to show that [39]:

$$\langle A(\mathbf{r}) \rangle = \sum_{j=1}^N \frac{m_j}{\rho(\mathbf{r}_j)} A(\mathbf{r}_j) W(\mathbf{r} - \mathbf{r}_j, h). \quad (8)$$

$$\langle \nabla_{\mathbf{r}} A(\mathbf{r}) \rangle = \sum_{j=1}^N \frac{m_j}{\rho(\mathbf{r}_j)} A(\mathbf{r}_j) \nabla_{\mathbf{r}} W(\mathbf{r} - \mathbf{r}_j, h). \quad (9)$$

An analogous expression can be obtained by the Laplacian. From equation (9) we can observe that there is no need for a mesh to compute spatial derivatives. With equations (8) and (9), we are ready to write the discrete version of the fluid equations of section 3. The smoothing length h is the width of the kernel and defines the distance at which a particle interacts with other particles. It is equivalent to the size of a grid-cell in finite difference methods.

We rewrite the terms of the Navier-Stokes equation (2), using this approach, as:

$$\begin{aligned}\vec{f}_i^{press} &= - \sum_j m_j \frac{p_i + p_j}{2\rho_j} \vec{\nabla} W(\mathbf{r}_i - \mathbf{r}_j, h) \\ \vec{f}_i^{visc} &= \mu \sum_j m_j \frac{\vec{v}_j - \vec{v}_i}{\rho_j} \vec{\nabla}^2 W(\mathbf{r}_i - \mathbf{r}_j, h) \\ \vec{f}_j^{grav} &= \rho_j \vec{g}_j\end{aligned}$$

where the $\vec{f}_i^{press}$ and $\vec{f}_i^{visc}$ are the pressure and viscosity forces. Only the gravity force $\vec{f}_i^{grav}$ is considered as external force. The density at each particle can be found from the following equation:

$$\rho_i = \sum_{j=1}^N \rho_j m_j W(\mathbf{r}_i - \mathbf{r}_j, h).$$

In order to have stability we adopt the following kernels [34]:

$$\begin{aligned}W_{grav}(r, h) &= \frac{315}{64\pi h^9} \begin{cases} (h^2 - r^2)^3, & 0 \leq r \leq h, \\ 0, & \text{otherwise} \end{cases} \\ W_{press}(r, h) &= \frac{15}{\pi h^6} \begin{cases} (h - r)^3, & 0 \leq r \leq h, \\ 0, & \text{otherwise} \end{cases} \\ W_{visc}(r, h) &= \frac{15}{2\pi h^3} \begin{cases} -\frac{r^3}{2h^3} + \frac{r^2}{h^2} + \frac{h}{2r} - 1, & 0 \leq r \leq h, \\ 0, & \text{otherwise} \end{cases}\end{aligned}$$

5. FHP

The FHP was introduced by Frisch, Hasslacher and Pomeau [18] in 1986 and is a model of a two-dimensional fluid. It can be seen as an abstraction, at a microscopic scale, of a fluid. The FHP model describes the motion of particles traveling in a discrete space and colliding with each other. The space is discretized in a hexagonal lattice.

The FHP particles move in discrete time steps, with a velocity of constant modulus, pointing along one of the six directions of the lattice. The dynamics is such that no more than one particle enters the same node at the same time with the same velocity. This restriction is the *exclusion principle*; it ensures that six Boolean variables at each lattice node are always enough to represent the microdynamics.

The velocity modulus is such that, in a time step, each particle travels one lattice spacing and reaches a nearest-neighbor node. When exactly two particles enter the same node with opposite velocities, both of them are deflected by 60 degrees so that the output of the collision is still a zero momentum configuration with two particles. The deflection can occur to the right or to the left, indifferently, as shown in Figure 1. For symmetry reasons, the two possibilities are chosen randomly, with equal probability.

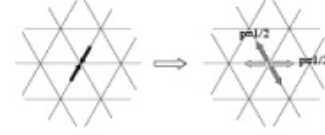


Figure 1. The two-body collision in the FHP.
Source [21]

When exactly three particles collide with an angle of 120 degrees between each other, they bounce back to where they come from (so that the momentum after the collision is zero, as it was before the collision). Both two- and three-body collisions are necessary to avoid extra conservation laws. For all other configurations no collision occurs and the particles go through as if they were transparent to each other.

The full microdynamics of the FHP model can be expressed by evolution equations for the occupation numbers defined as the number, $n_i(\mathbf{r}, t)$, of particle entering node $\mathbf{r}$ at time t with a velocity pointing along direction $\vec{c}_i$, where $i = 1, 2, \dots, 6$ labels the six lattice directions. The numbers n_i can be 0 or 1. We also define the time step as Δ_t and the lattice spacing as Δ_r . Thus, the six possible velocities $\vec{v}_i$ of the particles are related to their directions of motion by

$$\vec{v}_i = \frac{\Delta_r}{\Delta_t} \vec{c}_i. \quad (10)$$

The microdynamics of a LGCA is written as

$$n_i(\mathbf{r} + \Delta_r \vec{c}_i, t + \Delta_t) = n_i(\mathbf{r}, t) + \Omega_i(n(\mathbf{r}, t)) \quad (11)$$

where Ω_i is called the collision term [9].

Now we propose an extension to 3D, using the 2D FHP model explained above. In practice, the system of a given cellular automata rule cannot deal with an infinite lattice, it must be finite and have boundaries [9]. So, first we must define a domain in the three dimensional space, where the gas can evolve. Then, our proposal consists of regularly distribute planes along the x and z axis, like in Figure 2. Each of these planes is a system whose cellular automata rule is the 2D FHP.

Once simulated the two dimensional FHP in each plane independently, we perform a simple interpolation to generate a 2D macroscopic flow in each plane. In this step, we add new nodes to the FHP grid in order to complete a rectangular grid in each plane, as pictured in Figure 3.

Following the usual definition of statistical mechanics, we compute the macroscopic density in each node (x_i, y_i, z_i) of the plane $x = x_i$ through the expression:

$$\rho_x(x_i, y_i, z_i, t) = \sum_{j \in V} \sum_{k=1}^6 n_k(x_j, y_j, z_j, t), \quad (12)$$

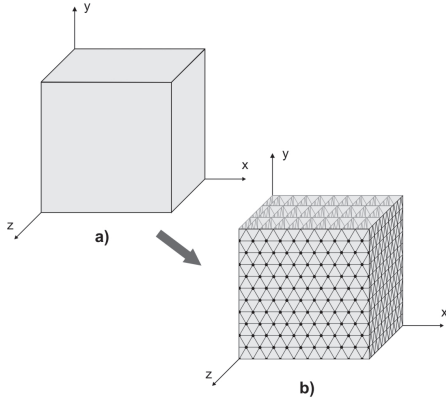


Figure 2. (a) Domain in 3D Space and (b) the distribution of 2D FHP planes.

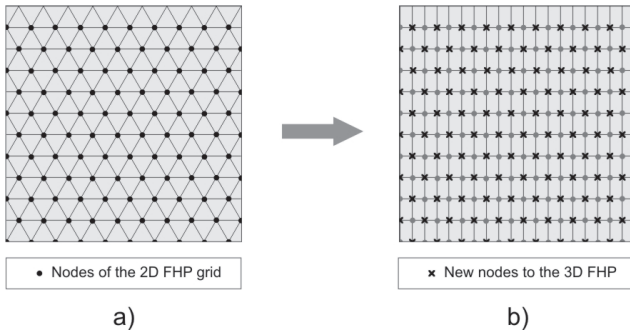


Figure 3. Extend the (a) FHP grid to generate a (b) rectangular mesh.

where V is a neighborhood of point (x_j, y_j, z_j) .

An analogous expression can be used for the plane $z = z_i$. Now, we must render a 3D macroscopic flow. We shall observe that each node (x_i, y_i, z_i) in Figure 2 belongs to the planes $x = x_i$ and $z = z_i$. So, the 3D density (or velocity) can be finally obtained through a simple mean of the corresponding values in the FHP planes, that means:

$$\rho(x_i, y_i, z_i, t) = \frac{\rho_x(x_i, y_i, z_i, t) + \rho_z(x_i, y_i, z_i, t)}{2}. \quad (13)$$

6. Mass-Spring Model

In this section we follow the reference [33]. The Mass-Spring is a discrete model in which surfaces are represented by polygonal meshes. The surface nodes works as masses and the edges defines the linear springs with damping. So,

given a particle i with mass m_i and position vector $\mathbf{r}_i$, the force system is composed by the elastic ($\vec{f}_i^{\text{elastic}}$), gravitational ($\vec{f}_i^{\text{grav}}$) and damping ($\vec{f}_i^{\text{damp}}$) forces, defined respectively, by:

$$\vec{f}_i^{\text{elastic}} = \sum_{j=1}^4 k_{ij} (l_{ij} - \|\mathbf{r}_i - \mathbf{r}_j\|) \frac{(\mathbf{r}_i - \mathbf{r}_j)}{\|\mathbf{r}_i - \mathbf{r}_j\|}, \quad (14)$$

where k_{ij} is the stiffness of the spring linking the nodes $\mathbf{r}_i$ and $\mathbf{r}_j$ and l_{ij} the spring rest length;

$$\vec{f}_i^{\text{grav}} = m_i \vec{g}, \quad (15)$$

$$\vec{f}_i^{\text{damp}} = \gamma_i \dot{\mathbf{r}}_i, \quad (16)$$

where $\vec{g}$ is the gravity field, γ_i is the damping factor and $\text{counter}(i)$ holds the number of particles accumulated in the corresponding position. Following Newton's Laws, we get the following evolution equation:

$$m_i \ddot{\mathbf{r}}_i = \vec{f}_i^{\text{elastic}} + \vec{f}_i^{\text{damp}} + \vec{f}_i^{\text{grav}}, \quad (17)$$

This system of ordinary differential equations can be efficiently solved by the Verlet integration technique [33]:

$$\mathbf{r}_i(t+h) = 2\mathbf{r}_i(t) - \mathbf{r}_i(t-h) + \ddot{\mathbf{r}}_i(t) h^2. \quad (18)$$

7. Rendering through Photon Map

Originally developed for global illumination simulation in scenes without participating media [26], Photon Map is a two-pass method where the first pass is the construction of structures to store the light information (photon maps) and the second is rendering using these information. The construction of the photon maps consists of photons emitted from the light sources and traced through the scene using photon tracing. Along the time evolution, if a photon hits a nonspecular surface, it is stored in the photon map.

Jensen [26] proposed the use of two photon maps: a caustics photon map and a global photon map. The caustics photon map stores all photons that have been traced from the light source through a number of specular reflections or transmissions before intersecting a diffuse surface, and the global photon map contains all photons representing indirect illumination on a nonspecular surface. In the rendering pass, the caustics photon map is used to render caustics directly and the global photon map is used to limit the number of reflections traced by the distribution ray tracer and to sample indirect illumination more efficiently.

It is possible to estimate radiance at any given surface position x using the photon map. By locating the n photons with the shortest distance to x it is possible to estimate the photon density around x [27]:

$$L_r(x, \vec{\omega}) \approx \sum_{p=1}^n f_r(x, \vec{\omega}_p, \vec{\omega}) \frac{\Delta\Phi_p(x, \vec{\omega}_p)}{\pi r^2}, \quad (19)$$

where f_r is the bidirectional reflectance distribution function, r is the distance to the n th nearest photon and $\Delta\Phi_p$ the flux carried by each photon p in direction $\vec{\omega}_p$. This approach can be seen as expanding a sphere centered at x until it contains n photons.

In [27], Jensen proposed an extension of the photon map method to be able to use in scenes with participating media, where photons can be scattered and absorbed by the media. To efficiently render the medium, it is necessary to store information about these scattering events. This storage of the photons occurs explicitly in the volume, given several advantages: the photons can be concentrated where necessary to represent intense illumination, the media do not have to be discretized and anisotropic scattering can be handled by storing the incoming direction of each photon.

The relationship between the density of the photons and the illumination is different on surfaces and in volumes. Then, the photons must be separated when the photon map is queried for information about the incoming flux. A separate volume photon map for the photons that are scattered in participating media was introduced in [27], and it is used to compute the illumination inside a participating medium while the global photon map is used, as before, to compute the illumination on surfaces.

Just like the original method, the first pass consists of building the photon maps using photon tracing. When a photon is traced within a participating medium, it can either pass unaffected through the medium, or an interaction can occur (be scattered or absorbed). If the photon interacts with the medium, and does not come directly from a light source, it is stored in the photon map. The cumulative probability density function, $F(x)$, expressing the probability of a photon interacting with a participating medium at position x is:

$$F(x) = 1 - \tau(x_s, x) = 1 - e^{-\int_{x_s}^x \kappa(\xi) d\xi}, \quad (20)$$

where x_s is the point at which the photon enters the medium. The transmittance $\tau(x_s, x)$ is computed using ray marching.

There is difference when calculating the density in the participating media. The density on a surface is computed using the projected area and the density in a medium is computed using the full volume, as shown in Figure 4. Once done the storage pass, we can use the photons stored in the volume photon map to compute an estimate of the in-scattered radiance:

$$L_i(x, \vec{\omega}) \approx \frac{1}{\sigma(x)} \sum_{p=1}^n f(x, \vec{\omega}_p, \vec{\omega}) \frac{\Delta\Phi_p(x, \vec{\omega}_p)}{\frac{4}{3}\pi r^3}, \quad (21)$$

where Φ is the in-scattered flux. Using (21) we can compute a radiance estimate at any given point inside a participating medium.

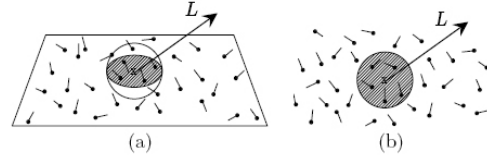


Figure 4. Radiance estimate for: (a) surfaces and (b) volumes. Source [27]

8. Proposal and Experimental Results

In this section we firstly present some results for the three-dimensional FHP rendered with a volume rendering technique, the single scattering method, available in the PBRT library (<http://www.pbrt.org/>). Then, we show two-dimensional results obtained by our team and discuss their extensions to 3D in the context of this work. The corresponding videos can be found in: http://www.lncc.br/~sicilia/assuntosAfins_pesquisa.htm.

8.1. Three-Dimensional FHP

Firstly, we highlight the simplicity and low computational cost of the FHP. Figure 5 shows an initial configuration with a volume of gas in the domain (semi-sphere in the top of the box).

The initialization is a very simple process: Firstly, the grid nodes inside the sphere are detected. Then, for each node detected, the algorithm randomly chose the quantity of particles and its directions.

Figure 6 shows a similar system, with the same initialization, but now using a solid sphere out of the fluid domain. It is used only to highlight transparency effects.

8.2. Gas-Fluid and Fluid-Surface Interactions

In [20] we combine the FHP and SPH to animate 2D two-phase systems composed by a gas simulated through FHP and a liquid modeled by SPH. The first point is how to model the interactions between the two phases in the interfacial area. For simplicity, in [20] we proceed as follows: given a point $\vec{r}$ in the interfacial area at a time t , we take a neighborhood and compute the particles mean velocity $\vec{u}_m$ through an expression similar to equation (12). Then, we set

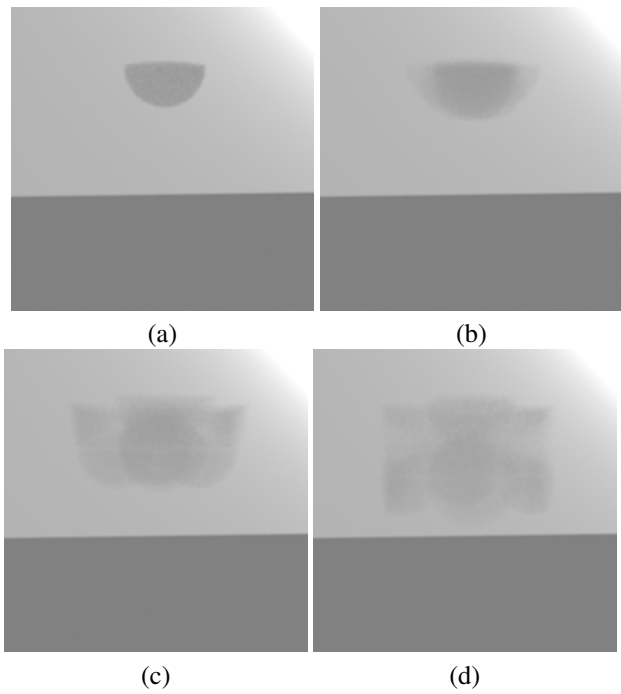


Figure 5. Configurations of the volume gas: (a) at initial step; (b) after 10 steps; (c) after 20 steps; and (d) after 50 steps of simulation.

the interaction force as:

$$\vec{F}_{int} = \tau \vec{u}_m(\mathbf{r}, t), \quad (22)$$

where τ is force scale parameter. This approach is more intuitive than the other ones [17] and allows the generation of interesting visual effects. Figure 7 shows examples of gas-liquid systems with the forces in the interfacial area given by expression (22). This figure shows a stream of 1000 particles with vertical macroscopic velocity and the liquid (blue) just before interaction.

The extension of this methodology to 3D depends on the following steps: (1) Three-dimensional FHP and SPH; (b) A model for the surface area and surface tension; (b) A suitable rendering technique.

Our team have implemented the SPH 3D presented in [34] and the three-dimensional FHP was described in section 5. The interfacial surface/tension model will also follows that reference. The interfacial surface is defined through the smoothed color field given by:

$$c_s(\mathbf{r}) = \sum_j m_j \frac{1}{\rho_j} W(\mathbf{r} - \mathbf{r}_j, h),$$

where m_j is the mass and ρ_j is the density of the particle j and W is an interpolation kernel of size is h [30]. A point belongs to the interface between the fluid and the gas if:

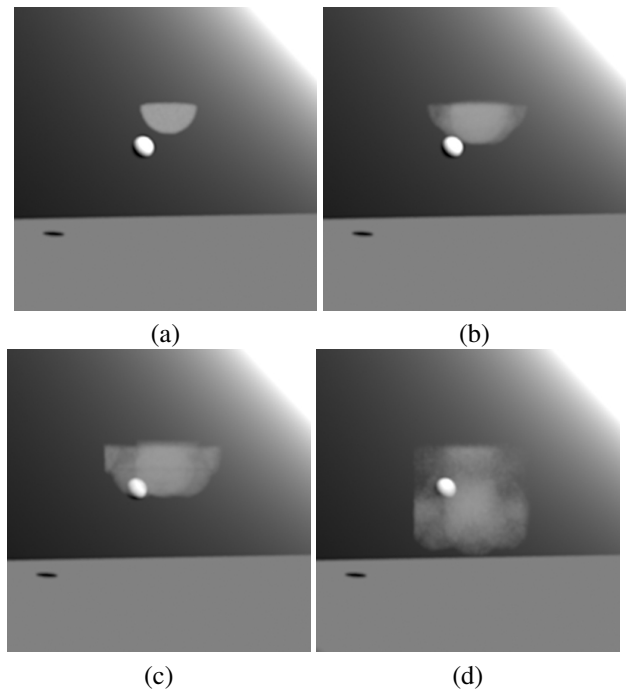


Figure 6. Configurations of the volume gas, with a solid sphere out of the system: (a) at initial step; (b) after 10 steps; (c) after 20 steps; and (d) after 70 steps of simulation.

$$\|\nabla c_s\| > T, \quad (23)$$

where T is a pre-defined threshold. Following such approach the force distribution in the interfacial can be computed through the expression [43]:

$$M = \sigma \nabla^2 c_s \nabla c_s. \quad (24)$$

Finally, we must consider a suitable rendering technique. Our choice is the photon map method because it provides effects like color bleeding, soft shadows, indirect illumination and caustics, which is very important in scenes with participating media. Firstly, we will use the implementation available in the PBRT library. Next we will implement a customized version of the photon map to get performance for real time applications. We could try to simulate both the phases through a Lattice Gas Model. However, a known problem of such approach is that no mathematical understanding is gained of which parameters lead to desired behaviors. Thus, the use of Navier-Stokes for the liquid modeling aims to allow standard ways to control the system.

The interaction between fluids and surfaces will be developed in two aspects: (1) Interaction between gas and a deformable surface. (2) Precipitation in terrain models.

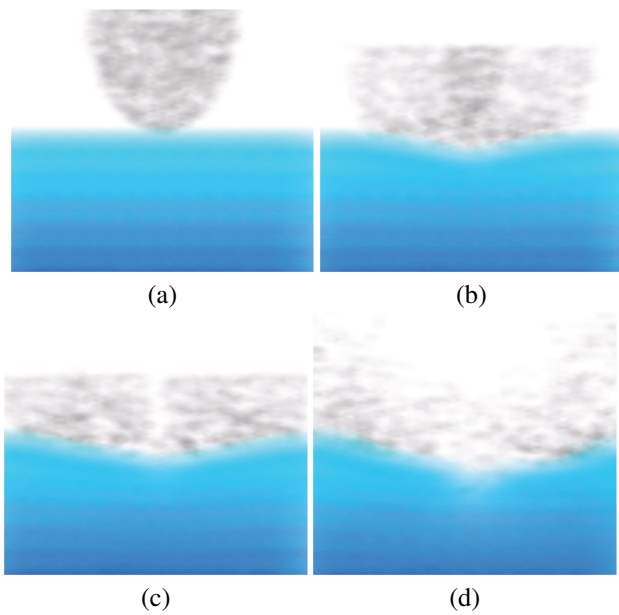


Figure 7. Configurations of liquid-gas: (a) at initial step; (b) after 15 steps; (c) after 25 steps; and (d) after 50 steps of simulation. Source [20]

In both cases, the fluid will be simulated by the three-dimensional FHP and the surface by a polygonal mesh. The first point is the collision detection. The system must be able to detect the collision between the particles of the FHP model and the surface. Once the surface is immersed in the FHP framework we just perform the rasterization of the surface in the FHP lattice and mark the obtained nodes (set a flag to 1). So, when a FHP particle reaches such a node of a rigid surface, it is just a matter of reflecting the particle following the FHP rules stated in section 5. The Figure 8 shows an example of this method for a sphere immersed in the three dimensional fluid simulation framework proposed in section 5.

If we have a deformable surface modeled by the Mass-Spring system described in section 6 then, for each node of the surface, we apply expression (22) to compute the interaction force and add this force in expression (17). After each time step of the Mass-Spring algorithm, we must rebuild the rasterization of the surface in the FHP lattice.

Finally, we aim to explore the three dimensional FHP model in the framework developed by our team for surface flow animation in digital terrain models [3]. The surface flow simulation follows a particle model, inspired in the LGCA technique, described in [3]. The basic data structures of the model are a polygonal representation of the surface and a regular lattice with nodes $(i, j) \in L \times L$, where $L \subset \mathbb{N}$. Particles move according to the terrain sur-

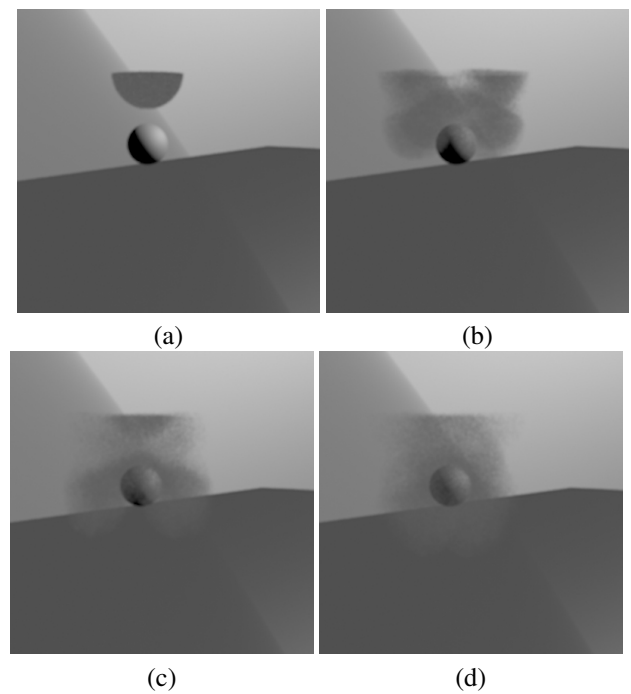


Figure 8. Configurations of a rigid sphere immersed in 3D FHP: (a) at initial step; (b) after 40 steps; (c) after 80 steps; and (d) after 90 steps of simulation.

face topography and the fluid configuration nearby. There is a counter in each lattice node used to keep the number of particles in the corresponding (i, j) position. A node may contain more than one particle in this model. The Figure 9 shows some snapshots of a simulation using this technique.

In this work we will embed the framework described in [3] in the 3D FHP framework (section 5) in order to simulate precipitation over terrains. The terrain surface is considered as a rigid one, so, the procedure for detection of collision between the particles of the FHP model and the terrain surface follows the idea also explained above. Given a lattice node (i, j) , we take a neighborhood of the corresponding surface point (i, j, z) and check the fluid density computed in the 3D FHP model. This quantity is used to quantify the precipitation value in the (i, j, z) point of the terrain.

9. Conclusions and Future Works

The main goal of our work is the development of a computational framework to create realistic animations of three dimensional systems involving gas-liquid and fluid-surface interactions. In this paper we propose to combine FHP, SPH and Mass-Spring systems for simulation tasks and photon

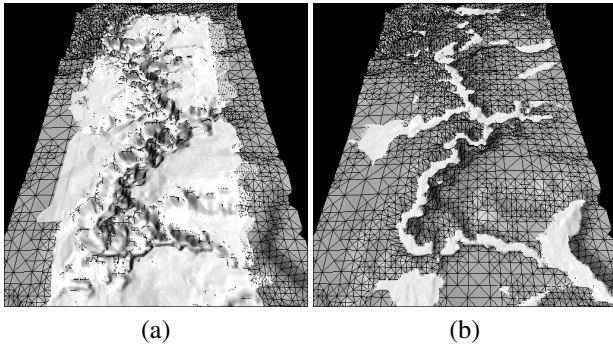


Figure 9. Configurations of simulation of flow in a terrain model: (a) at initial step; and (b) after 2500 steps of simulation. Source [3]

map realistic rendering. We discuss the theoretical elements of our proposal and present some experimental results. We do believe that the computational efficiency of the applied methods will generate a useful framework for real time applications, like games and virtual reality.

References

- [1] N. Adabala and S. Manohar. Techniques for realistic visualization of fluids: A survey. *Comput. Graph. Forum*, 2002.
- [2] C. Adler, D. d'Humières, and D. H. Rothman. Surface tension and interface fluctuations in immiscible lattic gases. *J. Phys. I*, 1994.
- [3] B. Barcellos, G. A. Giraldo, R. L. Silva, A. L. Apolinário, and P. S. S. Rodrigues. Surface flow animation in digital terrain models. In *SVR 2007 - IX Symposium on Virtual and Augmented Reality*, 2007.
- [4] C. Batty, F. Bertails, and R. Bridson. A fast variational framework for accurate solid-fluid coupling. *ACM Trans. Graph.*, 2007.
- [5] R. Benzi, S. Succi, and M. Vergassola. The lattice boltzmann equation: Theory and applications. *Physics Reports*, 1992.
- [6] R. Bridson, R. Fedkiw, and J. Anderson. Robust treatment of collisions, contact and friction for cloth animation. In *SIGGRAPH '02*, 2002.
- [7] J. M. Buick, W. J. Easson, and C. A. Greated. Numerical simulation of internal gravity waves using a lattice gas model. *Int. J. Numer. Meth. fluids*, 1998.
- [8] N. Chentanez, T. G. Goktekin, B. E. Feldman, and J. F. O'Brien. Simultaneous coupling of fluids and deformable bodies. In *SCA '06: Proc. of the 2006 ACM SIGGRAPH/Eurographics*, 2006.
- [9] B. Chopard and M. Droz. *Cellular Automata Modeling of Physical Systems*. Cambridge University Press, 1998.
- [10] B. Chopard, A. Dupuis, A. Masselot, and P. Luthi. Cellular automata and lattice boltzmann techniques: An approach to model and simulate complex systems. *Advances in Complex Systems*, 2002.
- [11] O. Deussen, D. S. Ebert, R. Fedkiw, F. K. Musgrave, P. Prusinkiewicz, D. Roble, J. Stam, and J. Tessendorf. The elements of nature: Interactive and realistic techniques. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Course Notes*, 2004.
- [12] D. A. Drew. Mathematical modeling of two-phase. *Annu. Rev. Fluid Mech.*, 1983.
- [13] D. Enright, S. Marschner, and R. Fedkiw. Animation and rendering of complex water surfaces. *ACM Trans. Graph.*, 2002.
- [14] O. Etzmuss, J. Gross, and W. Strasser. Deriving a particle system from continuum mechanics for the animation of deformable objects. *IEEE Trans. on Vis. and Comp. Graph.*, 2003.
- [15] N. Foster and R. Fedkiw. Practical animation of liquids. In *SIGGRAPH '01*, 2001.
- [16] N. Foster and D. Metaxas. Modeling the motion of a hot, turbulent gas. In *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, 1997.
- [17] U. Frisch, D. D'Humières, B. Hasslacher, P. Lallemand, Y. Pomeau, and J.-P. Rivet. Lattice gas hydrodynamics in two and three dimension. *Complex Systems*, 1987.
- [18] U. Frisch, B. Hasslacher, and Y. Pomeau. Lattice-gas automata for the navier-stokes equation. *Phys. Rev.*, 1986.
- [19] O. GENEVAUX, A. HABIBI, and J.-M. DISCHLER. Simulating fluid-solid interaction. In *Graphics Interface*, 2003.
- [20] G. Giraldo, A. Xavier, A. A. Jr, A. Neto, and P. Rodrigues. Animation of gas-liquid systems through lattice gas cellular automata and smoothed particle hydrodynamics. In *Proceedings of the CNMAC 2005*, 2005.
- [21] G. A. Giraldo, A. V. Xavier, A. L. Apolinario, and P. S. Rodrigues. Lattice gas cellular automata for computational fluid animation. Technical report, National Laboratory of Scientific Computing, 2005.
- [22] E. Guendelman, A. Selle, F. Losasso, and R. Fedkiw. Coupling water and smoke to thin deformable and rigid shells. In *SIGGRAPH '05*, 2005.
- [23] P. S. Heckbert. Adaptive radiosity textures for bidirectional ray tracing. In *SIGGRAPH '90*, 1990.
- [24] C. Hirsch. *Numerical Computation of Internal and External Flows: Fundamentals of Numerical Discretization*. John Wiley & Sons, 1988.
- [25] A. Iglesias. Computer graphics for water modeling and rendering: a survey. *Future Gener. Comput. Syst.*, 2004.
- [26] H. W. Jensen. Global illumination using photon maps. In *Proceedings of the eurographics workshop on Rendering techniques '96*, 1996.
- [27] H. W. Jensen and P. H. Christensen. Efficient simulation of light transport in scences with participating media using photon maps. In *SIGGRAPH '98*, 1998.
- [28] R. Keiser, B. Adams, D. Gasser, P. Bazzi, P. Dutre, and M. Gross. A unified lagrangian approach to solid-fluid animation. In *Point-Based Graphics, 2005. Eurographics/IEEE VGTC Symposium Proceedings*, 2005.
- [29] G. R. Liu and M. B. Liu. *Smoothed Particle Hydrodynamics : a Meshfree Particle Method*. World Scientific, 2003.

- [30] M. Liu, G. Liu, and K. Lamb. Constructing smoothing functions in smoothed particle hydrodynamics with applications. *Journal of Computational and Applied Mathematics*, 2003.
- [31] M. Müller, R. Keiser, A. Nealen, M. Pauly, M. Gross, and M. Alexa. Point based animation of elastic, plastic and melting objects. In *SCA '04: Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*, 2004.
- [32] J. Morris, P. Fox, and Y. Zhu. Modeling low reynolds number incompressible flows using sph. *JOURNAL OF COMPUTATIONAL PHYSICS*, 1997.
- [33] J. Mosegaard and T. S. Sorensen. Gpu accelerated surgical simulators for complex morphology. In *VR '05: Proceedings of the 2005 IEEE Conference 2005 on Virtual Reality*, 2005.
- [34] M. Müller, D. Charypar, and M. Gross. Particle-based fluid simulation for interactive applications. In *Proceedings of ACM SIGGRAPH symposium on Computer animation*, 2003.
- [35] M. Müller, S. Schirm, M. Teschner, B. Heidelberger, and M. Gross. Interaction of fluids with deformable solids: Research articles. *Comput. Animat. Virtual Worlds*, 2004.
- [36] A. Nealen, M. Muller, R. Keiser, E. Boxerman, and M. Carlson. Physically based deformable models in computer graphics. In *Eurographics 2005, State of The Art Report (STAR)*, 2005.
- [37] S. Premoze, T. Tasdizen, J. Bigler, A. Lefohn, and R. Whitaker. Particle-based simulation of fluids. In *EUROGRAPHICS*, 2003.
- [38] D. H. Rothman and S. Zaleski. Lattice-gas models of phase separation: Interface, phase transition and multiphase flows. *Rev. Mod. Phys*, 1994.
- [39] B. Schlatter. A pedagogical tool using smoothed particle hydrodynamics to model fluid flow past a system of cylinders. Master's thesis, Oregon State University, 1989.
- [40] B. Solenthaler, J. Schläfli, and R. Pajarola. A unified particle model for fluid-solid interactions. *Comput. Animat. Virtual Worlds*, 2007.
- [41] J. Stam. Stable fluids. In *Siggraph 1999*, 1999.
- [42] J. Stam. Flows on surfaces of arbitrary topology. In *SIGGRAPH '03*, 2003.
- [43] Y. Sun and C. Beckermann. Diffuse interface modeling of two-phase flows based on averaging: Mass and momentum equations. *Physica D*, 2004.
- [44] A. VanGelder and J. Wilhelms. Simulation of elastic membranes and soft tissue with triangulated spring meshes. Technical report, Univ. of Calif. at Santa Cruz, 1997.
- [45] E. Veach and L. J. Guibas. Metropolis light transport. In *SIGGRAPH '97*, 1997.
- [46] Z. Ye, D. Maidment, and D. McKinney. Map-based surface and subsurface flow simulation models: An object-oriented and gis approach. Technical report, Center for Research in Water Resources, University of Texas at Austin, 1996.
- [47] Y. Zhu and R. Bridson. Animating sand as a fluid. *ACM Trans. Graph.*, 2005.

Exploration of Volumetric Datasets through Interaction in Transfer Function Space

Francisco de M. Pinto and Carla M. D. S. Freitas
Instituto de Informática – Universidade Federal do Rio Grande do Sul
fmpinto@inf.ufrgs.br, carla@inf.ufrgs.br

Abstract

Direct volume rendering techniques allow visualization of volume data without extracting intermediate geometry. The mapping from voxel attributes to optical properties is performed by transfer functions which, consequently, play a crucial role in building informative images from the data. One-dimensional transfer functions, which are based only on a scalar value per voxel, often do not provide proper visualizations. On the other hand, multi-dimensional transfer functions can perform more sophisticated data classification, based on vectorial voxel signatures. The transfer function design is a non-trivial and unintuitive task, especially in the multi-dimensional case, and its controlled modification allows the user to selectively enhance different structures in the volume. In this paper we discuss the interactive approach of a transfer function design technique that allows the user to explore volumetric datasets by interacting with a derived space as well as with voxels in the volume space.

1. Introduction

In direct volume rendering (DVR), transfer functions (TFs) are used for emphasizing regions of interest inside the volumes. The most common type of transfer function is the one-dimensional TF, which assigns optical properties (usually color and opacity) to voxels based only on their scalar value. Notwithstanding, one-dimensional TFs have a very limited classification power because they can not make distinction between volume regions defined by scalar values within the same range. On the other hand, multi-dimensional transfer functions can perform better classification because they can take into account not only the scalar value of a voxel [8], but also other attributes like gradient magnitude, directional second derivative, curvature [7] and statistical measures [23].

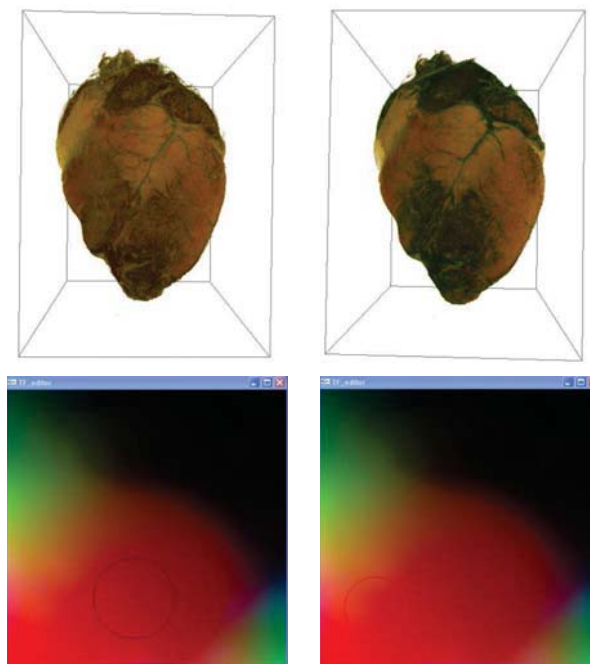


Figure 1. The sheep heart dataset viewed with two TFs, represented as circle in the color map. The circle represents a Gaussian opacity function with peak – maximum opacity – at the center. The color maps are fixed, with red, green and blue representing scalar value, gradient magnitude and directional second derivative, respectively.

Designing an appropriate transfer function, even a one-dimensional TF, is a difficult task and much attention has been given to this issue in the literature after Pfister et al.[15]. As the domain dimension increases, the visualization of the transfer function becomes more difficult, and so the interaction with it. Controlled modifications of color and opacity transfer functions allow enhancing parts of the datasets, thus hiding features and revealing others that the user is trying to isolate. This way, the representation of the

This paper is based on the M.Sc. dissertation named "Especificação de Funções de Transferência Unidimensionais e Multidimensionais para Visualização Volumétrica Direta" by the first author.

transfer functions and the ways the user can interact with them play an important role in the exploration of inner structures in volumetric datasets.

In this paper we present an interactive technique for the design of multi-dimensional transfer functions. With our approach the user can interact with a simplified representation of the TFs obtaining different informative visualizations in a very easy and fast way (Figure 1). We discuss the resulting method compared to a previous interface for specifying one-dimensional transfer functions.

The paper is organized as follows. Next sections present the closest related works, including a brief review of our previous work. Section 3 describes our approach in detail, while main implementation aspects are discussed in Section 4. Section 5 discusses the results obtained with our method, and finally, in Section 6 we draw some conclusions and point out future work.

2. Related work

2.1. One-Dimensional transfer function specification

Traditional approaches for TF specification rely on the user's effort in adjusting control points of a graphic plot mapping voxel values to opacity level and/or color tone. The control points are then interpolated in order to build the TF. But, with no clues or prior knowledge about the data, this is a "blind process". Some data-driven approaches provide to the user higher-level information [2][14] that helps in obtaining insight about the data distribution as well as supports the manual TF design. Other methods build abstractions of the TF specification process - the transfer functions can be hidden from the user [24] or a simplified space can be presented [6].

Kindlmann and Durkin [6] proposed a derived space for specification of opacity transfer functions in which the user specifies opacities for voxels as a function of the distance between the voxel and the nearest boundary. Informative histograms are built relating voxel values with the first and second derivative values associated to each voxel in the volume. From these histograms, the mean first and second derivative values associated to each voxel value are used to estimate the distance to the nearest border. Since the boundaries must be emphasized, voxel values with small estimated distances should receive larger opacity values. Prauchner et al. [18] used Kindlmann's method to classify the voxel values by the estimated distance to the nearest border. A set of voxel values with the

smallest distances is elected and random subsets are then built. The values of each subset are used as control points for the TF specification. Each of these points receives a random color and a random opacity value different from zero. The transfer functions are obtained by interpolating the control points. Consequently, each subset of the "best" voxel values derives a transfer function to be presented in a gallery of thumbnails, similar to the Design Galleries method [13]. This is the first level of the two-level interaction interface proposed by Prauchner et al. In the second level, the user can visualize a selected thumbnail in better resolution and refine its TF by adjusting the control points manually. The thumbnails can be randomly re-generated any time at interactive rates.

Following this two-level interaction interface approach, we [16] also adopted the gallery to present several thumbnails generated initially through a boundary emphasis technique, following Kindlmann and Durkin [6]. At this level, the user can generate new thumbnails (TFs) by either reapplying the boundary emphasis or selecting thumbnails as parents of a next generation of TFs, which are generated using an stochastic approach by He et al.[4]. The user can also select a specific thumbnail, and go to the second-level of interaction. At this level, looking at the rendered volume in high resolution, the user can modify the TF manually either by interacting with the TF graphic plot (Figure 2) or by picking voxels – to be emphasized by increasing the opacity for its scalar value – from a cutting plane (Figure 3). This work was published in SIBGRAPI 2006 and its extended version is to appear in a Computer and Graphics special issue.

2.2. Multi-dimensional transfer function specification

The design of multi-dimensional transfer functions brings challenges regarding both the visualization of the TF as well as the exploration of the TF domain.

It is possible to explicitly define a multi-dimensional transfer function by interacting in its domain with proper tools. Kniss et al. [8] proposed a volume rendering environment containing a set of direct manipulation widgets for volume inspection, visualization of data distribution and design of three-dimensional transfer functions, using dual domain interaction.

However, the difficulty of exploring the transfer function domain increases with its dimensionality; therefore some approaches for transfer function design provide interfaces based on interaction in a simplified space. Region growing techniques were used by Huang

and Ma [5] to segment volume data from seed points specified by the user; voxel signatures of the segmented region were used to automatically design a transfer function.

Tzeng and Ma [25] clustered voxel's signatures by similarity allowing the user to specify the desired classification by successively splitting and merging the clusters. The user sees the results by associating visual properties to each material class. The same authors [26] implemented multi-dimensional transfer functions using neural networks and support vector machines. They evaluate a classification function learned from training sets selected through a slice painting interface. The user paints the voxels of interest with a specific color, and the undesired ones with a different color. This way they implement a binary classification scheme.

Sêreda et al. [20] used hierarchical clustering to group voxels according to their LH signatures [21]. The user navigates through the hierarchy searching for the branches corresponding to regions of interest. Takanashi et al. [22] used independent component analysis (ICA) of multi-dimensional voxel signatures in order to represent them in a space where the classification is performed by moving axis aligned separation planes. Rezk-Salama et al. [19] created models of transfer functions that are carefully adjusted by specialists for several data sets of the same type in order to reveal the desired structures. Then, they applied PCA to represent the parameter set of each model by a single variable with an associated semantic. The models can be reused for new data sets by setting only that variable.

In order to have a generic design technique, we can consider designing a one-dimensional transfer function as a case of designing a multi-dimensional TF where the user selects only one variable to represent the voxel signature. In [17], we reported a method for designing nD-TFs, where voxel signatures are extracted from the volumetric dataset to be visualized, 2D or spherical self-organizing maps are built from the voxels signatures, and a dimensional reduction step results in voxels signatures being replaced by their coordinates in map space. These processes are performed off-line and perform non-linear dimensional reduction of voxel signatures. The result can be thought as a non-discrete, non-linear voxel classification scheme that group voxels by similarity and map them to a two-dimensional space: a square or a spherical surface. During rendering the user can specify color and opacity transfer functions by navigating on the map with a cursor that is the peak of a Gaussian opacity function. Next section presents this method in detail.

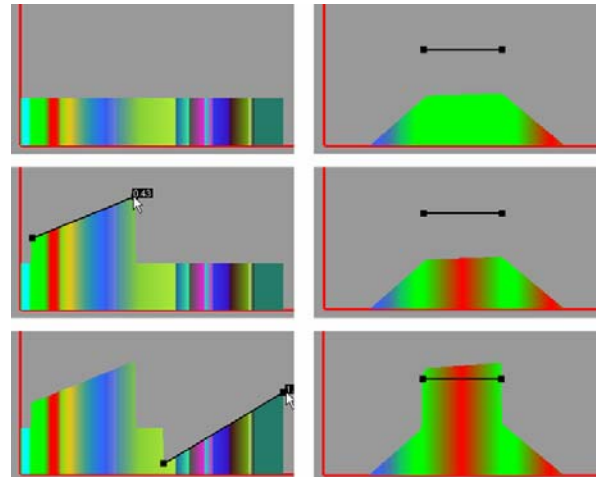


Figure 2: Manual design of one-dimensional opacity and color transfer functions.

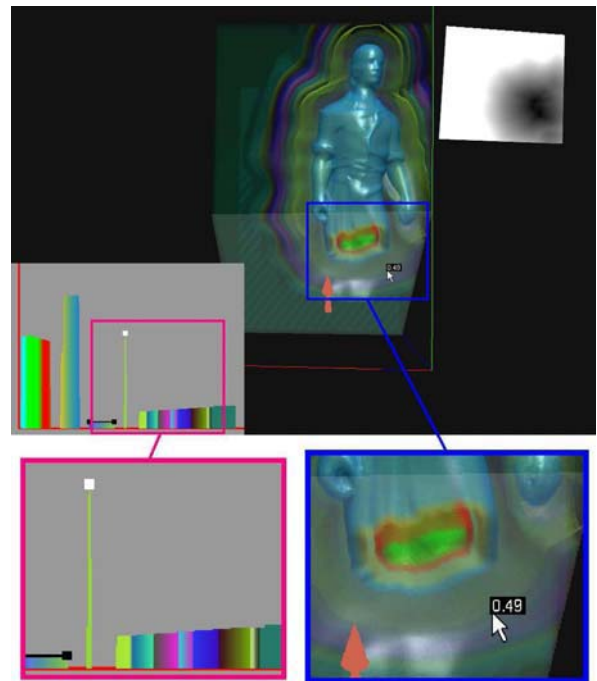


Figure 3. Top image: dataset¹ rendered using the TF shown at the left side. Bottom images: the voxel pointed by the cursor has the value marked as a white square in the TF plot. Opacity associated with this value can be interactively increased by the user.

3. Design of multi-dimensional transfer functions

Figure 4 shows the process of obtaining meaningful transfer functions for nD signatures. Our work on this subject [17] was published in EuroVis 2007.

¹ Dataset “Laçador” kindly provided by LACEM-UFRGS.

3.1. Map building process

The map building process starts with a preprocessing phase, when complex voxel's signatures (like derivative values, statistical measures, etc.) are extracted from the volume data and normalized. This way, each voxel has an nD signature (a set of scalar values represented as a vector) that can be used as a training case for the self-organizing (Kohonen) map building algorithm. It is important to mention that, depending on the source of volume data, there are many background voxels which do not carry useful information (air around scanned objects in CT/MRI volume data, for example), and would influence the map due to their high occurrence. Upon user decision, they can be partially removed from the input set of the training process by a very simple region growing technique using as seeds the voxels identified as background in the most exterior regions of the volume.

The signatures of all non-background voxels are employed as training cases presented in random order to the self-organizing map, and two types of neighborhood functions are applied. In a first stage we define the overall aspect of the map by training it using a Gaussian neighborhood function, and then, we continue the map training with a modified neighborhood function that depends not only on the topological distance, but also on the distance between the training case and the weight vector of the winner cell (refer to [17], for further details). This modified neighborhood function is designed in order to allow a voxel with a signature far from the weight vector of the corresponding winner cell (according to the distance metric) to have more influence on the map. Without this strategy, large homogeneous regions of the volume would tend to dominate the map, while important regions with fewer voxels would be badly represented (signatures far from their respective winner cells).

We use as topological distance the Euclidean distance between the integer 2D coordinates of two cells in the map grid. For spherical maps, the topological distance is the number of edges in the shortest path connecting two cells.

At the end of this process, we have a Kohonen (or spherical) map where each cell has an associated weight vector that represents a class of voxels, being the most similar weight vector for all elements in that class.

3.2. Dimensional reduction

Dimensional reduction was motivated by the need of providing a simplified space for the user to interact

with the multi-dimensional transfer functions. When using Kohonen maps, two-dimensional map space coordinates in the interval from zero to one can be associated to cells according to their position in the 2D grid. Dimensional reduction can be performed by replacing each voxel signature by the coordinates of its respective winner cell. However, this would cause unnecessary discretization. To avoid this, we create two multiquadric radial basis functions (multiquadric RBFs), for x and y map space coordinates, based on the weight vectors of all cells. For spherical maps we adopt x, y and z position coordinates ranging from -1 to 1, and use three RBFs to obtain the coordinates of voxel's signatures. Thus, the RBFs supports the final step in dimensional reduction of voxel signatures by producing, through interpolation, the proper x and y (and z) map coordinates for each nD voxel signature.

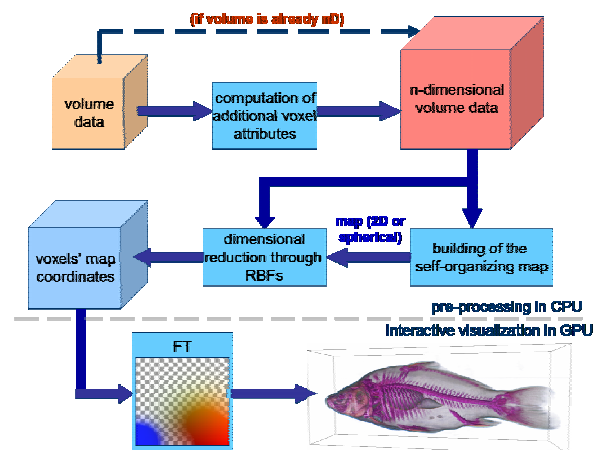


Figure 4. Distribution of processes between CPU and GPU for obtaining opacity and color TFs from nD voxel signatures.

Dimensional reduction normally implies loss and distortion of information, but volumetric data usually have properties that reduce this problem. The voxels signatures are usually not uniformly distributed in their domain (they form clusters, which are well represented in the map), and elements of the voxel signatures are often not completely independent [22]. Moreover, voxel signatures that are not present in the training set do not require space in the map.

3.3. Transfer functions specification

After the dimensional reduction step, the continuous map space defined by the RBFs becomes the TF domain. The user can interactively define the mapping from map coordinates (which represent voxel's signatures) to optical properties. We propose an

interface for specification of color and opacity transfer functions that provides dual domain interaction [8] as well as visualizations of the transfer function and of the voxel signatures.

3.3.1. Interaction in TF-domain. The visualization of voxel's signatures in our interface is obtained by directly mapping up to three elements of the weight vectors of the map cells (which actually are elements of the voxel signature) to the three color channels. The user decides which element of the nD signatures should be mapped to each basic color. One element can be associated to more than one color channel and a color channel may have no elements mapped to it. This interface feature illustrates the distribution of voxel's signatures on the map and can be used to build color TFs as described below. Figure 5 (a and b) shows the distribution of voxel's signatures of the well known engine data set. The same regions (clusters of signatures) can be found in both maps.

The transfer function is represented as an RGBA image and is displayed by blending it with a checkerboard pattern. The blending function allows TFs with small opacities to be clearly visualized. Figure 5 displays TFs on a Kohonen map (c) and on a spherical map (d) generated for visualizing the engine data set. The volumetric rendering of the data set using the TF in Figure 5c is shown in Figure 7a.

Transfer functions are composed by blending several 2D Gaussian opacity TFs, each one having an associated 2D color TF. We provide three types of color transfer functions that can be associated to a Gaussian opacity function: a constant color chosen from a colorpicker; map coordinates directly mapped to color channels; and elements of weight vectors of map cells mapped to color channels (for each map coordinate the weight vectors of the near cells are interpolated and mapped to colors). At each step a new Gaussian TF is specified and then blended with the current TF, for opacity and color. The result becomes the current transfer function and the composition continues until the desired TF is reached. At start, the current TF has zero opacity and RGB colors for all the map space.

In our interface, by clicking or dragging the mouse on the map representation, the user moves a circle whose center is the peak of a Gaussian function and whose radius is its standard deviation σ . The Gaussian TF is scaled by a constant k between zero and one which is linearly mapped to the circle color, with blue being zero and red, one. The parameters σ and k can be increased or decreased using the keyboard. In order to fully explore the spherical maps, they can be rotated by

dragging the mouse using the right button. The Gaussian opacity transfer function is defined in terms of the distance to the center of the circle.

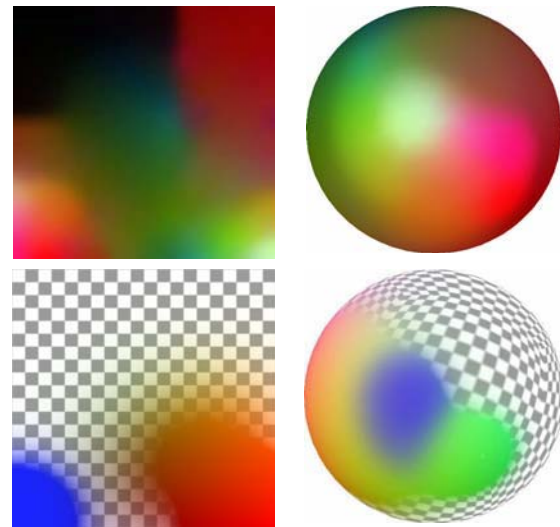


Figure 5. Maps of 3D voxel's signatures (a and b). Scalar value is mapped to red, gradient magnitude to green and second derivative in the gradient direction to blue. Transfer functions displayed on a Kohonen map (c) and on a spherical map (d).

The transfer function used for rendering is the composition of the current TF and the Gaussian function represented by the circle. This scheme provides interactive previewing of the effect of the composition while the user explores the map by moving the circle on it. When the desired effect is reached, the user can set the composition as the current TF using the space bar, and other Gaussian function can be further experimented.

Our interface (Figure 8) keeps track of all transfer functions defined during a session, and provides a tree representation of this evolution using static thumbnails of the volume rendered with the corresponding TF. This allows simple recovering of previous TFs by clicking on the thumbnails.

3.3.2. Interaction in TF-domain. At any time the user can rotate and translate the volume and place a clipping plane to better explore inner structures. The volume slice defined by the clipping plane is textured by mapping to color channels the map coordinates of the voxels sampled by the slice. This causes an interesting coloring effect that helps in inspecting the volume. The slice is blended with the rendered volume using an opacity value controlled by the user, as shown in Figure 6. When Kohonen maps are employed, the x and y map space coordinates of the voxels are mapped

to red and green. When using a spherical map the x, y and z map space coordinates are mapped to RGB colors.

The user can also click on the clipping plane to set the position of the Gaussian opacity function peak to the map coordinates of the voxel pointed by the mouse cursor, emphasizing that region. By moving the mouse on the clipping plane, the user can see the position of the pointed voxel depicted as a white cross in the map graphical representation (Figure 1). This spatial domain interaction mapped to TF domain helps in understanding the relationship between both domains.

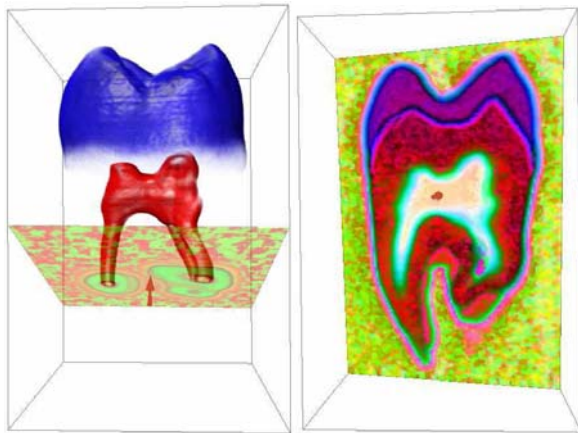


Figure 6. Visualizations of the tooth data set: a semi-transparent slice blended with the tooth image rendered using a transfer function specified in a 2D space built from a Kohonen map (a); and a fully opaque slice of the tooth colored according to voxel coordinates in a spherical map (b). The noisy regions can be clearly seen. The red arrow is the plane normal.

4. Implementation aspects

We implemented map training and dimensional reduction as offline processes, but rendering and transfer function specification demand interactive rates, which are achieved through an intensive use of the GPU (see Figure 4).

The map coordinates of the voxels are stored in a 3D RGB texture, which is sampled using view-aligned slices as proxy geometry. When using a Kohonen map, the transfer function is stored in a 2D RGBA texture which is accessed by using the R and G components (the x and y map coordinates) of the sampled 3D RGB texture. The blue component is used to identify background (zero) or non-background (one) voxels. Background voxels must receive zero opacity during rendering since they are not well represented in the map. Nevertheless, due to hardware interpolation, the blue component can assume values between zero and

one. With this in mind, the opacity is actually modulated by a smoothed step function of the blue component. When using a spherical map, the TF is stored in the GPU memory as an RGBA cube map and is accessed using the RGB values of the 3D texture, taken as vectors (the value of each color channel is first converted to the interval $[-1, 1]$). Background voxels have null vectors and the opacity is modulated by a smoothed step function of the L2-norm of the vectors. The blending of TFs and the evaluation of Gaussian opacity functions also run in GPU.

When sampling the three-dimensional texture for rendering, interpolation must be performed. The hardware can automatically interpolate the map coordinates stored in the 3D texture and generally this produces good results. However, in our approach, it is more correct to interpolate color and opacity associated to voxels (see [3] for better understanding). In our implementation, we use the GPU to create another 3D texture, with the same size, containing the RGBA values that result from the evaluation of the transfer function for each voxel, and this texture is sampled for rendering. When the transfer function changes, this texture must be recomputed, but this strategy is fast enough for our purposes. We also calculate another 3D RGB texture to store the gradient field of the opacity. This is done in GPU by applying central differences on each voxel. The opposite vector of the gradient of the opacity is used as surface normal for shading. Since we are using complex signatures for each voxel, this scheme for normal evaluation is more accurate than to sample a 3D texture containing the precomputed normals of the scalar field. Additionally, the normals of the opacity field do not have ambiguity in their orientation (see [11]). In our implementation, we set hardware interpolation of map coordinates and pre-computed normals as default options, but the user can select color and opacity interpolation and normals computed on the fly as high-quality rendering options.

As for the RBF design, we solve the systems of equations with the Lapack library [1]. GLUT and the GLUI libraries are used for the interface, while the rendering is based on OpenGL and Cg, with the framebuffer objects extension of OpenGL used in hardware accelerated computing (Figure 8 depicts the whole interface).

5. Results and discussion

5.1. Visualization

We tested our method using well-known data sets (see Figure 7), comprising scalar and multivariate

volume data. Similar results were obtained using Kohonen and spherical maps. Most of the data sets were successfully visualized using voxels signatures based on the scalar value, gradient magnitude and directional second derivative. For noisy scalar data, however, we achieved better results using statistical signatures like mean scalar value, standard deviation, and cubic root of the third-order statistical moment, taken from a small subvolume centered at the voxel under focus.

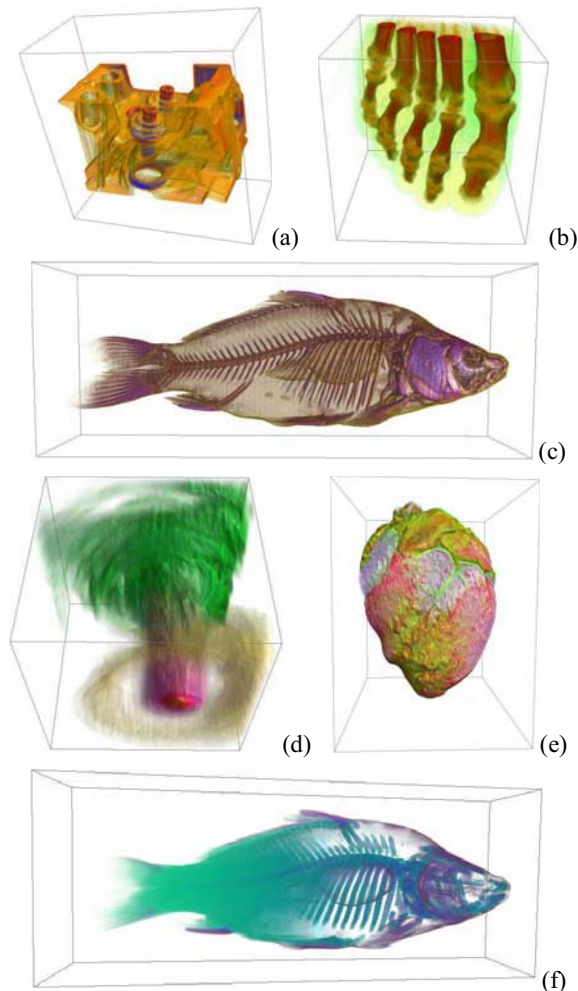


Figure 7. Visualizations obtained using Kohonen maps: (a) engine data set and (c) carp data set, both using scalar and derivative values (gradient magnitude and directional second derivative) as voxel signature; (b) foot data set, using statistical signatures (mean scalar value, standard deviation, cubic root of the third-order statistical moment); and (f) carp data set, using the normalized z coordinate of the voxels and same three statistical signatures, (z axis is horizontally represented); and Spherical maps: (d) hurricane data set at the 24th time step, using wind speed, pressure and temperature as voxel signature; and (e) sheep heart data set, using the same statistical signatures as (b).

Due to the loss and distortion of information usually caused by dimensional reduction, our method can not provide accurate quantitative information about the volume data during the transfer function specification. However, our approach is well suitable for revealing qualitative aspects like shape of structures and dissimilarity between regions.

Figure 7 shows visualizations of test data sets obtained using different sets of voxel attributes as signatures. In all these renderings, we used the automatic generated color transfer functions (see Subsection 3.3.1). For the hurricane data set we used only the colormap which assigns voxel attributes to color channels, since the attributes carry a clear physical meaning: temperature was mapped to red, pressure to green and wind speed to blue. However, the tooth data set (see Figure 6) was visualized using manually chosen colormaps and statistical signatures, achieving a very good separation of the pulp.

Since self-organizing maps group similar voxel signatures, the automatic generated color transfer functions produce very good results because they assign different colors to different regions of the map, which correspond to voxels with considerably different attribute's values.

The importance of each voxel attribute is defined by weights. We suggest associating smaller weights for higher-order voxel attributes. The visualizations presented in this paper were produced using weights of 1.3, 1.0 and 0.7 for the statistical variables formerly mentioned, respectively, and the same weights for scalar and first and second derivative values, respectively. For the hurricane data set the weights were 1.0 for wind speed and pressure, and 0.5 for temperature.

Regarding the shading using normals computed on the fly, results can be seen in Figure 7 (c, d and e). It is worth to mention that for the multivariate data sets, like the hurricane, we can not use meaningful pre-computed normals.

5.2. Interaction

Figure 1 (a and b) shows 2 views of the well-known sheep heart dataset obtained by simply moving the peak of the Gaussian TF on the map space (see Subsection 3.3.1) from the position in Figure 1(a) to that on Figure 1(b). The difference in the color of the circle also indicates a difference in the value of a specific parameter (k) from one opacity function to the other. Automatically generated color transfer functions are usually a good choice, which turns the design

process less difficult. This is the case in the examples shown here.

By moving the Gaussian opacity function on the map space, the user quickly obtains an overview of the main structures in the data volume. Users can tune opacity levels and combine TFs with simple (keyboard) input. Careful tuning of parameters of the Gaussian functions and their combination allow building meaningful visualizations.

In comparison to the previous design technique, targeted at one-dimensional transfer functions, the method described herein can be thought as including boundary emphasis since the derived attributes are often intended to capture the boundaries between regions. Thus, the interactive tasks of enhancing boundaries that a user had to accomplish with the previous interface are automatically included in the exploration of the map and setting of Gaussian functions at specific positions on it.

The history tree, briefly described in Subsection 3.3.1, provides a powerful mechanism for exploring the transfer function domain, allowing not only “undo” and “redo” operations, but navigation in the whole history of TF modifications. Both design interfaces have this feature, which proved to be necessary due to the interactivity of the TF design process, which can yield to situations where the user “looses” a good transfer function.

6. Conclusions

In this paper we presented a new approach for the design of multi-dimensional transfer function that uses self-organizing maps to perform dimensional reduction of the voxel attributes. The strongest points of our technique are simplicity and flexibility. Our approach allows building multi-dimensional transfer functions through the exploration of a simplified (reduced) space where traditional interaction techniques can be employed. A simple and effective interface for transfer function design is provided, and the user can interact with the system in both spatial and TF domains.

Self-organizing maps have the ability of representing clusters of voxel's signatures in a compact way, and this helps to understand the data distribution. All relevant voxel's signatures are represented in the map and every region of the map has voxels mapped to it. Moreover, exploring two-dimensional maps is easier and faster than navigating through a class hierarchy. The proposed dimensional reduction scheme requires a preprocessing step, but it has clear advantages in relation to volume segmentation techniques because it performs a non-discrete classification which can

represent uncertainty. In addition, with simple interaction, the user can change the transfer function defined in the map space, interactively obtaining new visualizations in real-time.

As future work, regarding interaction with the TF, we want to transport transfer functions designed in map space to the actual multi-dimensional space using the Gaussian multi-dimensional TFs proposed by Kniss et al. [9]. Another promising future work is the semi-automatic search for important structures in the map. This search could be aided by an interface that would provide additional information about the spatial distribution of voxels.

Wherever Times is specified, Times Roman or Times New Roman may be used. If neither is available on your word processor, please use the font closest in appearance to Times. Avoid using bit-mapped fonts if possible. True-Type 1 fonts are preferred.

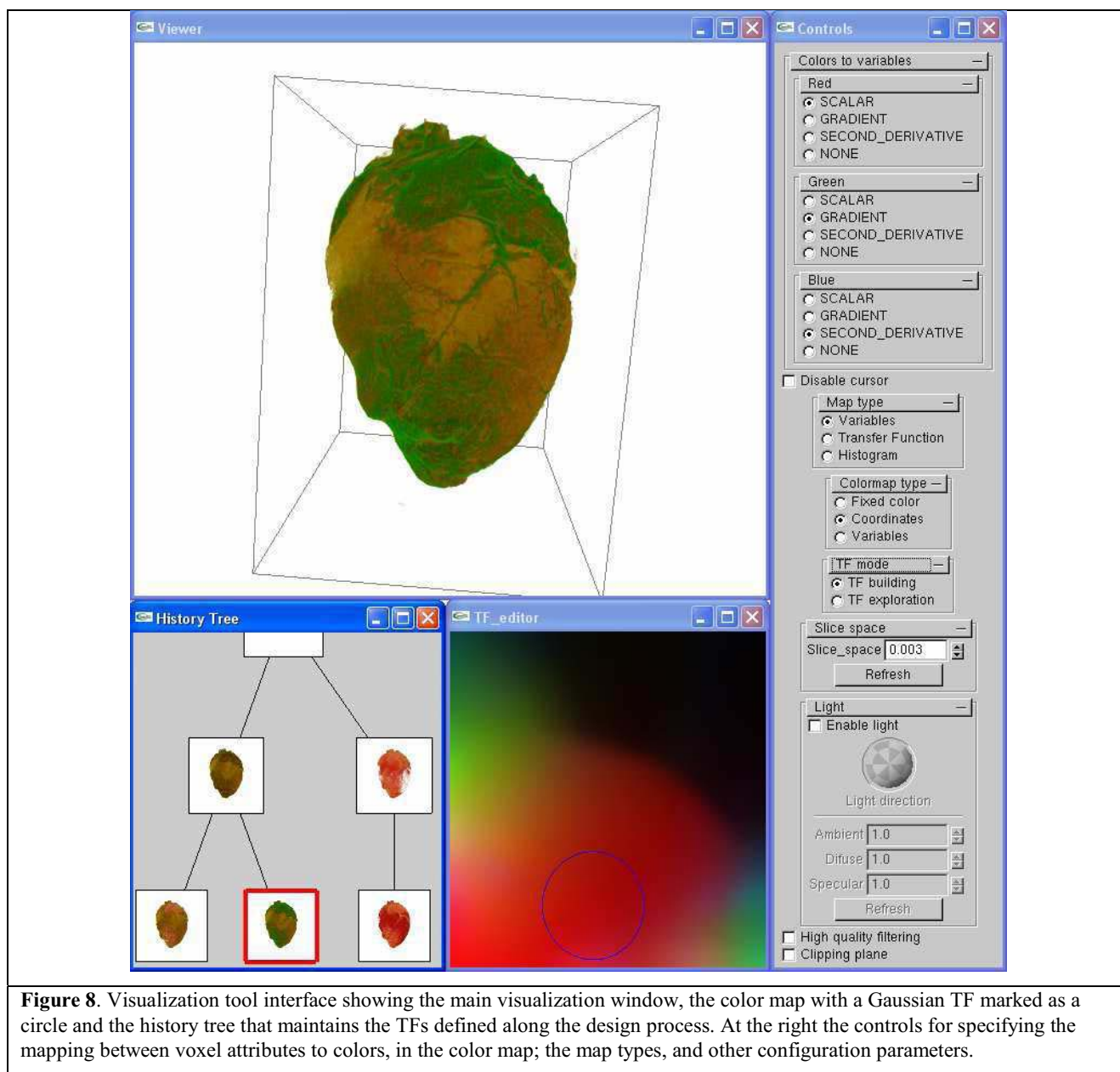
7. Acknowledgements

We acknowledge the financial support from CNPq, the Brazilian Scientific Research Funding Agency.

8. References

- [1] Anderson, E., Bai, Z., Bischof, C., Blackford, S., Demmel, J., Dongarra, J., Du Croz, J., Greenbaum, A., Hammarling, S., Mckenney, A., Sorensen, D. *LAPACK Users' Guide, third ed.* 1999.
- [2] Bajaj, C. L., Pascucci, V., Schikore, D. R., “The contour spectrum”, *Proceedings of the 8th conference on Visualization '97*, Los Alamitos, CA, USA, 1997, pp. 167–174.
- [3] Hadwiger, M., Berger, C., Hauser, H., “High quality two-level volume rendering of segmented data sets on consumer graphics hardware”, *Proceedings of IEEE Visualization*, 2003, pp. 40–47.
- [4] He, T., Hong, L., Kaufman, A., Pfister, H., “Generation of transfer functions with stochastic search techniques”, *Proceedings of the 7th conference on Visualization '96*, Los Alamitos, CA, USA, 1996, pp. 227–235.
- [5] Huang, R., Ma, K.-L., “RGVis: Region growing based techniques for volume visualization”, *Proceedings of the 11th Pacific Conference on Computer Graphics and Applications*, 2003, pp. 355–363.
- [6] Kindlmann, G., Durkin, J.W., “Semi-automatic generation of transfer functions for direct volume rendering”, *Proceedings of the 1998 IEEE symposium on Volume visualization*, New York, NY, USA, 1998, pp. 79–86.
- [7] Kindlmann, G., Whitaker, R., Tasdizen, T., Möller, T., “Curvature-based transfer functions for direct volume rendering: Methods and applications”, *Proceedings of IEEE Visualization*, 2003, pp. 513–520.

- [8] Kniss, J., Kindlmann, G., Hansen, C., "Multidimensional transfer functions for interactive volume rendering", *IEEE Transactions on Visualization and Computer Graphics* 8, 3, 2002, pp. 270–285.
- [9] Kniss J., Premoze, S., Ikits, M., Lefohn, A., Hansen, C., Praun, E., "Gaussian transfer functions for multi-field volume visualization", *Proceedings IEEE Visualization*, 2003, pp. 497–504.
- [10] Kniss, J. M., Uiter, R. V., Stephens, A., Li, G.- S., Tasdizen, T., Hansen, C., "Statistically quantitative volume visualization", *Proceedings of IEEE Visualization*, 2005, pp. 37–44.
- [11] Lum, E. B., Ma, K.-L., "Lighting transfer functions using gradient aligned sampling", *Proceedings of IEEE Visualization*, 2004, pp. 289–296.
- [12] Ma, F., Wang, W., Tsang, W. W., Tang, Z., Xia, S., Tong, X., "Probabilistic segmentation of volume data for visualization using SOM-PNN classifier", *Proceedings of IEEE Symposium on Volume visualization*, 1998, pp. 71–78.
- [13] Marks, J., Andalman, B., Beardsley, P. A., Freeman, W., Gibson, S., Hodgins, J., Kang, T., Mirtich, B., Pfister, H., Ruml, W., Ryall, K., Seims, J., Shieber, S., "Design galleries: a general approach to setting parameters for computer graphics and animation", *Proceedings of SIGGRAPH '97*, New York, NY, USA, 1997, pp. 389–40.
- [14] Pekar, V., Wiemker, R., Hempel, D., "Fast detection of meaningful isosurfaces for volume data visualization", *Proceedings of the conference on Visualization '01*, Washington, DC, USA, 2001, pp. 223–230.
- [15] Pfister, H., Lorensen, B., Bajaj, C., Kindlmann, G., Schroeder, W., Avila, L. S., Martin, K., Machiraju, R., Lee, J., "The transfer function bake-off", *IEEE Computer Graphics and Applications*, 21, 3, 2001, pp. 16–22.
- [16] Pinto, F.M., Freitas, C.M.D.S., "Two-level interaction transfer function design combining boundary emphasis, manual specification and evolutive generation", *Proceedings of XIX SIBGRAPI - Brazilian Symposium on Computer Graphics and Image Processing*, IEEE Press, 2006.
- [17] Pinto, F.M., Freitas, C.M.D.S., "Design of multidimensional transfer functions using dimensional reduction", *Proceedings of Eurographics/IEEE-TVCG Symposium on Visualization*, 2007.
- [18] Prauchner, J.L., Freitas, C.M.D.S., Comba, J. L. D., "Two-level interaction approach for transfer function specification", *In Proceedings of XVIII SIBGRAPI - Brazilian Symposium on Computer Graphics and Image Processing*, IEEE Press, 2005.
- [19] Rezk-Salama, C., Keller, M., Kohlmann, P., "High-level user interfaces for transfer function design with semantics", *IEEE Transactions on Visualization and Computer Graphics*, 12, 5, 2006, pp. 1021–1028.
- [20] Šereda, P., Bartoli, A. V., Gerritsen, F. A., "Automating transfer function design for volume rendering using hierarchical clustering of material boundaries", *Proceedings of IEEE/Eurographics Symposium on Visualization*, 2006, pp. 243–250.
- [21] Šereda, P., Bartoli, A. V., Serlie, I. W. O., Gerritsen, F. A., "Visualization of boundaries in volumetric data sets using LH histograms", *IEEE Transactions on Visualization and Computer Graphics*, 12, 2, 2006, pp. 208–218.
- [22] Takanashi, I., Lum, E. B., Ma, K.-L., Muraki, S., "ISpace: Interactive volume data classification techniques using independent component analysis", *Proceedings of the 10th Pacific Conference on Computer Graphics and Applications*, 2002, pp. 366–374.
- [23] Tenginkai, S., Lee, J., Machiraju, R., "Salient isosurface detection with model-independent statistical signatures", *Proceedings of IEEE Visualization*, 2001, pp. 231–238.
- [24] Tzeng, F.-Y., Lum, E. B., Ma, K.-L., "A novel interface for higher-dimensional classification of volume data", *Proceedings of the 14th IEEE Visualization*, Washington, DC, USA, 2003, pp. 66–73.
- [25] Tzeng, F.-Y., Ma, K.-L., "A cluster-space visual interface for arbitrary dimensional classification of volume data", *In Proceedings of the Symposium on Data Visualization*, 2004, pp. 17–24.
- [26] Tzeng, F.-Y., Lum, E. B., Ma, K.-L., "An intelligent system approach to higher-dimensional classification of volume data", *IEEE Transactions on Visualization and Computer Graphics*, 11, 3, 2005, pp. 273–284.



Fluid Warping

Dalia Bonilla Luiz Velho André Nachbin
Instituto de Matemática Pura e Aplicada

Luis Gustavo Nonato
Universidade de São Paulo - São Carlos

Abstract

Warping techniques can be complicated and difficult to use, but through the use of fluid dynamics the warping becomes simple and it is intuitively controlled by physical properties such as viscosity and forces. These properties are naturally associated with the image itself or with spatial control handles. The key idea is to think of the image domain as a two-dimensional incompressible and homogeneous fluid, and to use the Navier Stokes equations to change it by applying forces to the image function. In this way, the process does not move the image values as in fluid simulations, but transforms the coordinates of a parametrization of the image through a vector field generated by the simulation equations — effectively acting as a texture mapping.¹

1. Introduction

The process that changes the shape of objects in an image is called *warping*. The use of warping plays an important role in many applications, from the correction image distortions in medical data to the creation of special effects in the entertainment industry. Warping can also be used together with blending for creating transitions between different objects, a technique known as *morphing* [7].

More formally, given an image, $f : U \subset \mathbb{R}^2 \rightarrow C$, the mapping between source space (u,v) and destination space (x,y) is called *warping filter*. Such a map, $W(f) = g$, acts on the input image $f(u,v)$ giving rise to an output image $g(x,y)$ that can be regarded as a deformation of the image domain [8]. Furthermore, in the case of morphing, a composition operator combines the result of two synchronized warping filters applied to different images.

Spatial transformations based on Navier-Stokes equations, and the use of fluid dynamics in general, present a great potential in the above context because they are very powerful to derive warping transformations. In the present paper we develop a framework where such a technique is exploited.

¹ Extended abstract of doctorate thesis.

2. Related Work

There are many methods for image warping. They can be classified into: parameter-based; feature-based; free-form; and hybrid [9].

Parameter-based methods are warping techniques controlled by a family of transformations, such as scaling, twisting, and bending. This type of technique was introduced in Computer Graphics by Alan Barr[2].

Feature-based methods encompass a whole class of warping techniques, which differ regarding types of geometric features and reconstruction functions. In these methods a correspondence of features in the source and destination objects must be provided by the user. Typical reconstruction functions include scattered data interpolation, inverse distance weighted kernels, and radial basis.

Free-form based methods use specification by coordinate systems. For this purpose, they employ free-form curves (B-splines, Bézier etc.) to define the coordinate curves [4].

The pioneer work using fluid dynamics in image processing was introduced by Bertalmio et al. [10], with a method for digital inpainting. They think of the image intensity as a *stream function* and the Laplacian of the image intensity plays the role of the vorticity of the fluid, which is transported into the region to be inpainted by a vector field defined by the stream function. Even though their idea is based on Navier-Stokes equations, our work is different. In our technique, the fluid warping uses a vector field directly created by the velocity of fluid by Navier-Stokes equations: we neither use the stream function nor vorticity.

The proposed fluid warping technique carries the coordinates of a parametrization of the image through of a vector field generated by the Navier-Stokes equations. Warping is controlled by physical parameters associated with the characteristics of the image itself or by other auxiliary images and applied to the dynamic simulation.

Depending on the application (for example in those already mentioned above) an important aspect of fluid warping is user control. Deformations are easy to specify using fluid dynamics through physical properties, such as viscosity and forces.

3. Fluid Simulation

In this section we present the fluid simulation setting that will be adopted in our framework for image warping. We are interested in modeling homogeneous, incompressible fluids with variable viscosity. Such formulation gives a good compromise between simplicity and expressive power. Moreover, we require an efficient implementation, that allows large time-steps, for this purpose we will employ the Stable Fluids algorithm, extending it to handle variable viscosity.

3.1. Mathematical Formulation

First we review some mathematical concepts for the definition of the fluid equations. The fluid is defined over a region $D \in \mathbb{R}^2$. Let $\mathbf{x} = (x, y)$ be any point of D . Let $v(\mathbf{x}, t)$ denote the velocity of the particle of fluid moving through $\mathbf{x}$ at time t . The *velocity field of fluid* is denoted by v and it is a vector field tangent to the trajectory of the particle. For each time t assume that the fluid has mass density $\rho(\mathbf{x}, t)$. The fluid is called *incompressible* when

$$\nabla \cdot v = 0.$$

Assuming that the fluid has mass density constant in space (i.e., the fluid is *homogeneous*) and that ρ is constant in time. Then we have

$$\rho(\mathbf{x}, t) = \text{constant} \quad \text{in particular } \rho(\mathbf{x}, t) = 1.$$

Let μ be the *viscosity* of fluid and p the *pressure*. Then for constant viscosity, the basic Navier-Stokes equations for incompressible fluids are: (see [5])

$$\begin{aligned} \partial_t v &= -\nabla p - v \cdot \nabla v + \mu \Delta v + f \\ \nabla \cdot v &= 0. \end{aligned} \quad (1)$$

where the “ $\cdot$ ” denotes a dot product between vectors, while $\nabla = (\partial/\partial x, \partial/\partial y)$ is the vector of spatial partial derivatives, ∂_t is the partial derivative $\frac{\partial}{\partial t}$, and $\nabla \cdot$ is the divergent. We also adopt the notation $\Delta = \nabla \cdot \nabla$.

For more expressivity, we want to consider changes of viscosity in space and thus we have to formulate the Navier-Stokes equations for variable viscosity. The equations are:

$$\begin{aligned} \partial_t v &= -\nabla p - v \cdot \nabla v \\ &+ \nabla \mu(\mathbf{x})(\nabla v + \nabla v^\top) + \mu(\mathbf{x}) \Delta v \\ &+ f \end{aligned} \quad (2)$$

Where $\nabla v = \begin{pmatrix} \partial_x v^1 & \partial_y v^1 \\ \partial_x v^2 & \partial_y v^2 \end{pmatrix}$ and $\nabla v^\top = \begin{pmatrix} \partial_x v^1 & \partial_x v^2 \\ \partial_y v^1 & \partial_y v^2 \end{pmatrix}$ then

$$\nabla v + \nabla v^\top = \begin{pmatrix} 2\partial_x v^1 & \partial_x v^2 + \partial_y v^1 \\ \partial_y v^1 + \partial_x v^2 & 2\partial_y v^2 \end{pmatrix}.$$

We will use the Helmholtz-Hodge Decomposition theorem, where a vector field w on D can be uniquely decomposed in the form

$$w = u + \nabla q$$

such that u has zero divergence and q is a scalar field. If we have $w = u + \nabla p$, then

$$\nabla \cdot w = \nabla \cdot \nabla q = \Delta q, \text{ and } w \cdot n = \nabla q \cdot n = \frac{\partial q}{\partial n} = 0.$$

This is a Poisson equation for a scalar field with the Neumann boundary conditions. A solution to this equation can be used to compute u :

$$u = w - \nabla q.$$

Now we define the operator P which projects any vector field w onto its divergence free part u : $\nabla \cdot u = 0$ and $Pw = u$.

P is a linear operator and thus $w = Pw + \nabla p$, $u = Pu$, $P(\nabla p) = 0$. We apply the operator P to both sides of the basic Navier-Stokes equations (1) and obtain

$$\partial_t v = P(\partial_t v + \nabla p) = P(-(v \cdot \nabla)v + \mu \Delta v + f). \quad (3)$$

This form of equation eliminates the pressure and expresses $\partial_t v$ in terms of v alone. The pressure can then be recovered as the gradient part of $-(v \cdot \nabla)v + \mu \Delta v + f$. In the same way, we obtain for the variable viscosity Navier-Stokes equation (2).

$$\begin{aligned} \partial_t v &= \\ P(-(v \cdot \nabla)v + \nabla \mu(\mathbf{x})(\nabla v + \nabla v^\top) + \mu(\mathbf{x}) \Delta v + f) \end{aligned} \quad (4)$$

3.2. Computational Method

Now we briefly review the Stable Fluids algorithm that we adopt for the implementation of the fluid simulation. For further details we refer to the original papers [16] and [17]. Since the algorithm was designed for fluids with constant viscosity, we adapt it for our setting by developing an extension for variable viscosity fluids.

The algorithm solves the Navier-Stokes equations (4) and it is unconditionally stable. This method is based on an operator splitting strategy.

For each time step Δt the algorithm solves the equations in four stages, starting from a velocity field $w_0 = v(\mathbf{x}, t)$ of a previous time step and then sequentially resolving each term of the equations. The stages are

$$w_0 \xrightarrow{\text{add force}} w_1 \xrightarrow{\text{advect}} w_2 \xrightarrow{\text{diffuse}} w_3 \xrightarrow{\text{project}} w_4$$

The first stage is the addition of external force f . It adds the force field multiplied by the time step to velocity $w_1 = w_0 + \Delta t f(x, t)$.

The second stage accounts for the effect of advection of the fluid on itself. It is given by an advection equation

$$\partial_t w_2 = -(w_1 \cdot \nabla) w_2$$

and is solved by using a semi-Lagrangian technique [6]

$$w_2(x) = w_1(x - \Delta t w_1(x))$$

The basic idea behind the advection step is, instead of moving the particle forward in time through the velocity field, to move it backwards in time through the field and calculate a new velocity by interpolation (guaranteeing stability).

The third stage solves for the effect of viscosity and is given by equation

$$\partial_t w_3 = \mu \Delta w_3$$

it uses a simple implicit solver for the diffusion equation

$$(I - \Delta t \mu \Delta) w_3 = w_2$$

where I is the identity operator. One way to solve this equation is to get the solution for system

$$Aw_3 = w_2$$

by using the Jacobi method.

The fourth stage projects the velocity field onto the incompressible (divergence free) field. This step also involves the solution of a Poisson equation

$$\Delta q = \nabla \cdot w_3 \text{ and } w_4 = w_3 - \nabla q.$$

The methods used for solving this stage are finite difference schemes and Jacobi.

We will extend the formulation above to solve our equation (2) using the same stages, except that because we assume variable viscosity the diffusion stage will be different. More precisely, the original Stable Fluids solve the equation

$$\partial_t w_3 = \mu \Delta w_3$$

and we have to solve the equation

$$\partial_t w_3 = \nabla \mu(x) (\nabla w_3 + \nabla w_3^\top) + \mu(x) \Delta w_3.$$

To solve this step, let $w_3 = (w^1, w^2)$ and we write again in this form

$$\begin{cases} w_t^1 = 2\mu_x w_x^1 + \mu_y (w_y^1 + w_x^2) + \mu \Delta w^1 \\ w_t^2 = 2\mu_y w_y^2 + \mu_x (w_y^1 + w_x^2) + \mu \Delta w^2. \end{cases}$$

where the sub-indices t, x and y denote the partial derivatives ∂_t, ∂_x and ∂_y . And we discretize again using implicit difference schemes backward in time and central space to

w_3 and central space scheme to μ . The results are stable, just as we wanted. Once again time we have used Jacobi to solve this modified stage.

The Stable Fluids also includes a method to compute the motion of a density φ immersed in the fluid [17]. The equation for the evolution of this density is

$$\partial_t \varphi = -(v \cdot \nabla) \varphi + \kappa \Delta \varphi + S \quad (5)$$

where κ is a diffusion rate and S is a source of density. The density is advected by the fluid using a semi-Lagrangian technique, as in the second stage of the algorithm.

4. Warping with Fluids

The first idea to use fluids for image warping would be to consider the image as a density field immersed in the fluid, and as such could be transported by fluid motion. However, as can be seen in Figure 1 this approach does not work well.

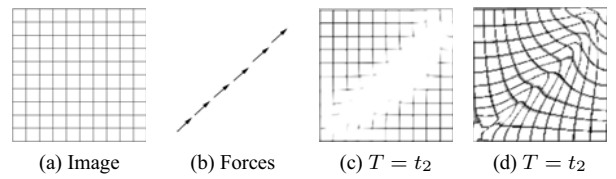


Figure 1: Image warping with fluids.

For this experiment we used the source image in Figure 1a and the force field in Figure 1b. We performed a fluid simulation using these external forces, and then applied the resulting velocity field to the image in two ways.

First, we treated the image intensity as density values immersed in the fluid and computed the motion of the image function using equation (5), which moves the grey level values of the image directly. The result is a blurred image and a fast disappearance of the image after applying additional forces, as shown in Figure 1(c). This phenomenon happens because of the intrinsic dissipation of the fluid equation.

The best strategy to avoid the above problem is to move through the fluid velocity field, instead of image values, the coordinates of a parametrization of the image. In this way, the image features are preserved by a texture mapping mechanism, as shown in Figure 1(d).

4.1. Texture Mapping and Warping

Based on the conclusions of the previous experiment, the image warping using fluid simulation can be formalized as a deformation induced on a texture.

Let $[0, M] \times [0, N]$ be the image domain. Given a force applied to the image domain, we regard a fluid in $[0, M] \times$

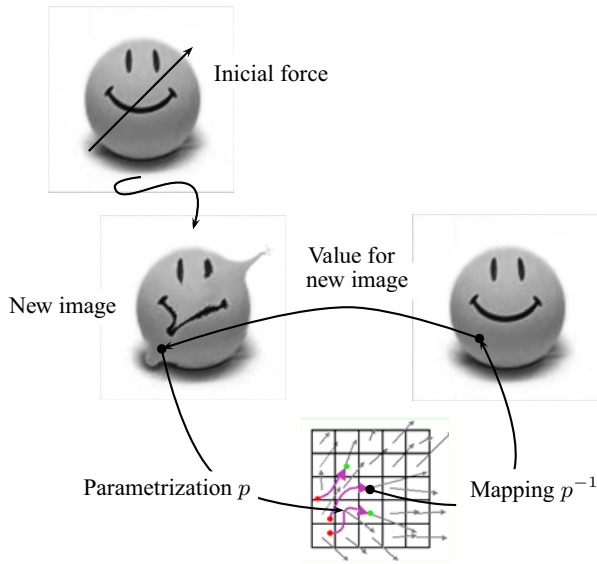


Figure 2: **Texture Warping:** image coordinates carried by vector field generated by the Navier Stokes equations.

$[0, N]$ and move this fluid taking external forces as initial data. For point (x, y) in the image domain, the coordinates are transported to point (p_1, p_2) in $[0, 1] \times [0, 1]$ by parametrization p .

Now, each coordinate p_1 and p_2 is interpreted as a density immersed in the fluid on $[0, M] \times [0, N]$. We move each coordinate separately through the fluid velocity field, by equation (5) getting for a time step Δt the coordinates q_1 and q_2 .

Finally, for a point (x, y) on domain of a new image the value is computed as follows. If (q_1, q_2) belongs to $[0, 1] \times [0, 1]$, then it is transported by the inverse parametrization at the point (w, z) of $[0, M] \times [0, N]$. The value of the new image at point (x, y) is the value of the original image at (w, z) . If (q_1, q_2) do not belong to $[0, 1] \times [0, 1]$ then the value of (x, y) is zero.

This computational scheme for fluid warping using texture mapping is illustrated in the diagram of Figure 2.

The warping function induced by the velocity field of a fluid simulation has many desirable properties, such as smoothness and continuity, that can be exploited in applications.

Additionally, the mapping is naturally time dependent, such that given the initial conditions (i.e., forces and parameters) at time $t = 0$, we have a one-parameter family of warpings W_t , $t \in \mathbb{R}_+$, which directly applies for animation, and may also be interpreted in terms of evolution.

4.2. Control Mechanisms

Fluid simulation is capable of producing potentially very complex deformations with good properties for image warping, as we have discussed so far.

However, in order to be useful, we need to be able to control the simulation such that the desired transformation is obtained.

The simplest way to control the fluid warping is through the direct specification of the simulation parameters.

By inspection of the fluid equations (2) is easy to verify that the available parameters are:

- external forces $f(x, t)$; and
- fluid viscosity $\mu(x)$.

An extra parameter is the total duration T of the simulation.

Surprisingly, just this small set of parameters already provide powerful and intuitive mechanisms for controlling the image warping.

First, note that both the forces and viscosity are spatially variable. Thus, they are identified with functions on the image domain which could be associated with image features. Therefore, one natural way to specify forces and viscosity is by auxiliary images.

We define the viscosity from the intensity of an auxiliary image function on $[0, M] \times [0, N]$. The viscosity values are computed from a normalization of the image values to $[0, 1]$ and global scaling factor.

While the viscosity is defined by a scalar field, forces are defined by a vector field, which can be encoded as an RGB image. However, in many situations it is also convenient to specify forces from point or curve sources. For this we employ procedural definitions.

Furthermore, for specification purposes it is convenient to take the total external force $f = \sum f_i$, as the additive combination of separate individual forces f_i .

Since forces vary in time too, we must take this fact into consideration for the definition of forces. More specifically, useful options are: 1) instantaneous forces (i.e., acting at specific time instants t_i , $i = 1, \dots, N$) In this case, it is common to use initial forces at $t = 0$; 2) constant forces (i.e., acting during a certain time interval $[t_0, t_1]$; 3) arbitrary forces (i.e., fully variable in time).

Also, the application of forces may be defined *a priori* or may depend on a sensing function on the simulation. This second option is related to the so-called force-feedback mechanism of control theory.

Up to now, we have adopted only basic specification of forces, such as instantaneous and constant force fields. We have also made use of a simple sensing mechanism, mainly for stopping the simulation.

5. Results and Examples

In this section we give some examples of the results that can be obtained with fluid warping.

The first set of examples demonstrate the use of forces and viscosity to control image deformations.

Figure 3 shows a warping of a train leaving the station, defined only by an instantaneous force at $t = 0$ along a curve on the top part of the image. The goal was to get a time-space distortion effect suggesting speed. Note that we are able to create extreme distortions just by running the simulation for a longer period.

Figure 4 exhibits a warping of a Van Gogh's painting, defined by forces and variable viscosity. Here, we generated forces from the gradient of a segmentation of the hat and the viscosity from a quantization of the image values. Note that the forces act to expand the hat and, because the surrounding background has variable viscosity, the hat deforms in a non-uniform way.

The second set of examples is an attempt to evaluate fluid warping as a regular warping technique. For this purpose, we make a comparison using examples from the paper of Arad et al [1]. In this seminal work, the authors describe a technique based on radial basis functions.

Figure 5 is an example in [1] for lifting the corner of a girl's mouth. To achieve this effect we constructed a viscosity function, shown in Figure 5(b), that imposes a restriction on the warping area. In this function, the region outside the desired warping is white and more viscous (i.e., opposing great resistance to fluid motion). The warping area is dark and less viscous (i.e., offering small resistance to fluid motion). The forces used in the process are given by the gradient field of the image shown in Figure 5(c). They exert an upward force at the mouth location, producing the desired effect. The results in Figures 5(d) and (e) demonstrate that we are able to closely match their technique.

6. Conclusions and Future Work

In this paper we introduced *fluid warping*, a framework for image deformation using fluid dynamics. Our technique provides good results and is simple to use.

Future work includes two avenues of investigation. One direction is towards a finer and more precise control of the warping transformation. This can be achieved by exploiting sophisticated force-feedback mechanisms, possibly combined with an optimization strategy. Another direction is to extend the framework for image morphing. In this context, the coupling of two parallel fluid simulations could be considered together with a composition operator for image blending. Ideally, such an operator should be physically inspired.

References

- [1] N. Arad and D. Reisfeld. Image warping using few anchor points and radial functions. *Computer Graphics Forum*, 14(1):35–46, 1995.
- [2] A. Barr. Global and local deformations of solid primitives. *In Proceedings of SIGGRAPH*, 1984.
- [3] T. Beier and S. Neely. Feature-based image metamorphosis. *SIGGRAPH Comput.*, 2(26):35–42, July 1992.
- [4] H. Birkholz and D. Jackel. Image warping with feature curves. *In Proceedings of SIGGRAPH*, pages 199–202, 2003.
- [5] A. J. Chorin and J. E. Marsden. *A mathematical Introduction to Fluid Mechanics*. Springer-Verlag, 1993.
- [6] R. Courant, E. Isaacson, and M. Rees. On the solution of nonlinear hyperbolic differential equations by finite differences. *Communications on Pure and Applied Mathematics*, 5:243–255, 1953.
- [7] J. Gomes and L. Velho. *Image Processing for Computer Graphics*. Springer Verlag, 1997.
- [8] Heckbert and P. *Fundamentals of Texture Mapping and Image Warping*. Master's Thesis, University of California, Berkeley, 1989.
- [9] G. Jonas, D. L. B. Costa, and L. Velho. *Warping and Morphing of Graphical Objects*. Morgan Kaufmann Publ., 1999.
- [10] B. M., B. A., and S. G. Navier-stokes fluid dynamics and image and video inpainting. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '01)*, pages 9–14, December 2001.
- [11] A. Nachbin. *Notas do Curso: Dinâmica dos Fluidos*. 2006.
- [12] L. S. W. G. C. K-Y, and Y. S. S. Image metamorphosis with scattered feature constraints. *IEEE Transactions on Visualization and Computer Graphics*, 2(4):337–354, December 1996.
- [13] S. S. McPhail, T. and W. J. Image deformation using moving least squares. *ACM Transactions on Graphics (TOG)*, 25(3), July 2006.
- [14] A. R. Smith. Planar 2-pass texture mapping and warping. *In Proceedings of SIGGRAPH*, 1987.
- [15] D. B. Smithe. A two-pass mesh warping algorithm for object transformation and image interpolation. *Technical memo, Industrial Light and Magic*, 1990.
- [16] J. Stam. Stable fluids. *SIGGRAPH 99 Conference Proceedings, Annual Conference Series*, pages 121–128, August 1999.
- [17] J. Stam. Flows on surfaces of arbitrary topology. *ACM Transactions On Graphics (TOG), Proceedings of SIGGRAPH*, pages 724–731, July 2003.
- [18] J. C. Strikwerda. *Finite Difference Schemes and Partial Differential Equations*. CRC Press, 1999.
- [19] G. Wolberg. Skeleton based image warping. *Visual Computer*, 5(1/2):95–108, March 1989.
- [20] G. Wolberg. Digital image warping. *IEEE Computer Society Press, Los Alamitos, CA.*, 1990.

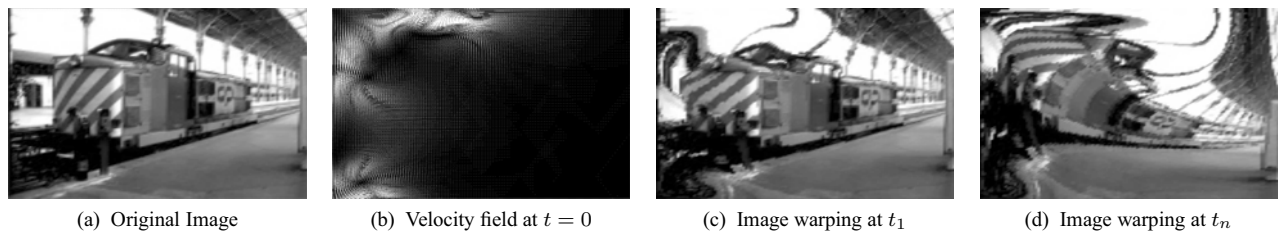


Figure 3: Train warping.

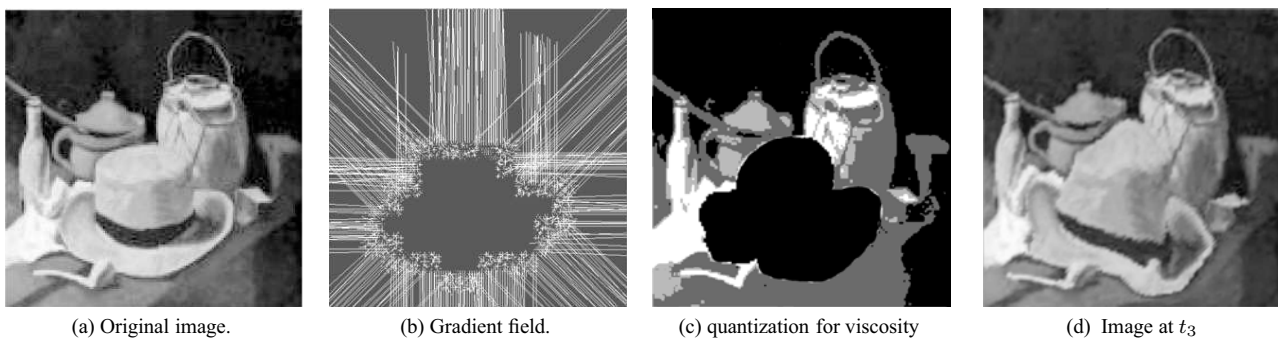


Figure 4: Melting the Van Gogh hat.

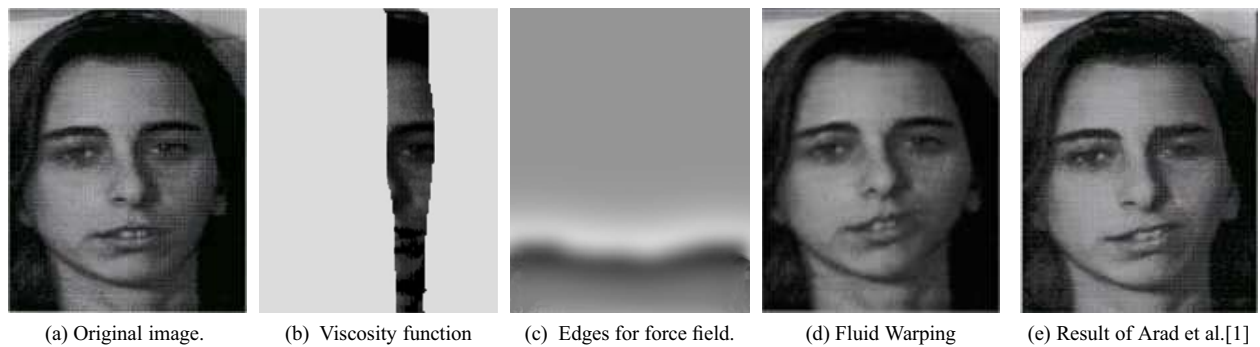


Figure 5: Comparison with example in Arad et al., (1)

Image-Based Techniques for Surface Reconstruction of Adaptively Sampled Models *

Ricardo Marroquim
ricardo@lcg.ufrj.br
Universidade Federal do Rio de Janeiro

Paulo Roma Cavalcanti
roma@lcg.ufrj.br
Universidade Federal do Rio de Janeiro

Abstract

Image based methods have proved to efficiently render scenes with a higher efficiency than geometry based approaches, mainly because one of their most important advantages: the bounded complexity by the image resolution, instead of by the number of primitives. Furthermore, due to their parallel and discrete nature, they are highly suitable for GPU implementations. On the other hand, during the last few years point-based graphics has emerged as a promising complement to other representations. However, with the continuous increase of scene complexity, solutions for directly processing and rendering point clouds are in demand. In this paper, algorithms for efficiently rendering large point models using image reconstruction techniques are proposed. Except for the projection of samples onto screen space, the reconstruction time is bounded only by the screen resolution. The method is also extended to interpolate other primitives, such as lines and triangles. In addition, no extra data-structure is required, making the strategy memory efficient.

1. Introduction

The area of point based graphics has grown during the last decade, reaching a high level of acceptance among the computer graphics community. With the advent of high-resolution registration equipments, such as 3D scanners, its importance became apparent as methods that could directly deal with the provided data. Furthermore, since scene complexity increases in a much higher rate than the output devices' resolution, the usefulness of some traditional representations were put on doubt for a set of applications.

Nevertheless, point-based graphics is still an emerging area. By far its goal is not to replace other representations, such as polygonal meshes, but to complement them where

it is advantageous. With a dedicated symposium during the last five years, devoted sessions in the most important computer graphics conferences, and a recently released point-based graphics book [6], the area has broadened its scope and embraced techniques on various fields, e.g., 3D acquisition, surface reconstruction, processing & modelling, rendering, and animation.

The main advantage of the point-based representation is its lack of explicit connectivity information. Each sample contains all information necessary for processing and rendering, implying in a few benefits: first, memory space is decreased, since there is no need to store connectivity; second, during processing and modelling connectivity and topological restrictions do not have to be preserved; third, there is no need for surface reconstruction from the point cloud, for example, to generate a polygonal mesh; and finally, as the geometric primitive approaches the image primitive, i.e., a point and a pixel, valuable gains can be obtained by exploiting the rendering pipeline.

This work is mainly focused on rendering techniques for point-based models. The first researches evolved from the image-based rendering field, but quickly became an independent area itself. Surface Splatting is today the most popular method for the direct rendering of unstructured point clouds. However, it has long distanced its foundation from the image-space techniques premises. It is at this point that we advocate such methods to reclaim one of its most important contributions: the decoupling of the algorithm cost from the scene complexity.

We direct our attention to the visualization of large datasets, varying from a few hundred thousand to a few million samples, where we believe point representation is more valuable. This is specially true, since for very dense models triangles are projected to less than one pixel, causing an unnecessary overhead due to the primitive assembly process. The main contributions of this work are:

- An efficient algorithm for surface reconstruction of point-based models using image-based techniques. This work was published at the Symposium on Point-Based Graphics 2007 [12].

* This paper contains extracts of the D.Sc. thesis "Interactive Point-Based Rendering" by the first author.

- The extension of the previous algorithm to deal with other primitives using the same pipeline. The possibility to render models with triangles, lines, or points further demonstrates the usefulness of the method. As a proof of concept, a scheme for rendering tree models using lines is described, where not only fewer primitives are needed, but performance is improved. This work was published in the Computer & Graphics Journal [13].
- Further investigation of the screen-based surface reconstruction has led to new insights on how to improve the rendering quality. By creating a more exact reconstruction strategy, silhouettes and details are better preserved, while the performance is not significantly degraded. This work is to be published at the Sibgraph 2008.
- Finally, we investigate the use of Level-of-Detail structures to improve the performance of the point projection phase (the only that depends on the number of samples). A first effort using recent GPU techniques is to be presented at the Siggraph 2008 posters session.

2. Related Work

The pioneer work on point-based rendering by Levoy and Whitted [11], proposed the conversion of any representation to points. In this manner, a standardized rendering algorithm could be employed using points as the *Lingua Franca* of computer graphics. Even if it did not cause much impact at the time, in a sense, they anticipated one of the reasons point-based representation has become popular recently: the continuous increase of visual complexity of graphics scenes. More than ten years later research on point based rendering was resumed, but this time coming from a different source, image-based techniques. For this reason here we present relevant related methods on both areas, allowing for a more thorough understanding of this work's motivations and purposes.

2.1. Image-Based Rendering

The area of image-based rendering is overlapped by several others, such as image-space and image-processing techniques. Here, we classify image-based as an approach that uses the image as graphics primitives, instead of the traditional geometric primitives. This classification is important, since methods such as ray tracing are accomplished in image-space, but rely heavily on the model's geometry, that is, are physically based.

The main motivation comes from photorealism: if the goal is to render images with photo quality, then why not

use it as the input primitive? However, even though a photography offers an enormous quantity of information about structure and appearance, it is still a static image. Image-based rendering is mainly about solving the following problem: given a set of static images describing the scene, how can new viewpoints be efficiently extracted to provide a dynamic high quality setting?

Two algorithms, which were developed almost simultaneously, describe appropriately the use of images as primitives: the "Light Field" by Levoy and Hanrahan [10], and the "Lumigraph" by Gortler et al. [5]. They are both based on the use of the plenoptic function to describe the rays of light traversing a scene. The full 7D plenoptic function is defined as the intensity of light rays passing through the camera center at every 3D location (V_x, V_y, V_z) , at every possible angle (θ, ϕ) , for every wavelength λ , at every time t , i.e., $P_7(V_x, V_y, V_z, \theta, \phi, \lambda, t)$ [9]. The authors realized that this function could be reduced to 4D by parameterizing the rays as a line intersecting two planes, one being the camera plane and the other the object plane. By using the 4D function $P_4(u, v, s, t)$, it is possible to treat the scene as a volume, where each specific viewpoint is characterized by the extraction of a 2D image from the 4D volume.

As the area evolved, it strayed away from a pure appearance based approach; new algorithms emerged adding geometric information to the image primitives (in fact, Lumigraph already used some geometric hint to accomplish better depth tests). Techniques such as *Billboards* and *Depth Sprites* use some geometric information to efficiently render dynamic scenes with few images. These strategies were specially employed in the game industry to add details to the environment, without having to resort to complex geometric models.

However, to provide a fully interactive environment, or a system able to render the scene from any viewpoint, it is necessary to turn to more complex structures. *3D Warping* uses multiple reference images from a scene and stores depth information per pixel. To render an arbitrary scene, the pixels are reprojected to 3D space and then projected to the new viewpoint.

In the same sense, Shade et al. [18] proposed the *Layered Depth Images*, or LDIs. Basically, it is the use of a single reference image with multiple color and depth information per pixel. To render from a new viewpoint, the reference image's pixels are projected to the new reference. However, this method is limited by its fixed resolution, implying that the sample density may not be convenient for every viewpoint. To overcome this problem, Chang et al. [3] proposed the LDI Trees, using an octree with an LDI stored per node. In this way, each pixel is stored in different resolutions and can be dynamically selected to compose the final image.

2.2. Point-Based Rendering

Following Levoy and Whitted's [11] initial proposal to use points as graphics primitives, Grossman and Dally [7] derived a point-based rendering system from previous image-based techniques. In fact, the idea was not yet to render point clouds, but to project the point data onto uniform structures, i.e., reference images. However, differently from 3D Warping, each sample also stores a normal, adding surface variational information. This is specially important because, with view-independent samples, it is not only possible to eliminate much of the redundancy from previous approaches, but also to compute dynamic illumination. The "pull-push" image-processing technique is employed to fill holes left by the reconstruction algorithm, that are usually caused by insufficient sampling under magnification.

The *Surfels* approach, by Pfister et al. [15], uses a similar methodology as the previous algorithm. However, instead of reference images, an LDI Tree is created. In addition, the samples are splatted onto the image buffer to perform a depth test. Nevertheless, they also make use of the "pull-push" algorithm to fill holes between the splats.

Surface Splatting [23, 24, 22] is today the most popular point-based renderer. It proposes to directly render unstructured point clouds by projecting to image space a surface patch, representing the sample's local approximation. These projections are called reconstruction kernels, and together with an anti-aliasing filter, compose the frame buffer by computing the contribution to every covered pixel. In a last phase, all contributions per pixel are weighted averaged to render the final surface reconstruction of the object.

Motivated by the high quality images provided by Surface Splatting, efficient implementations were proposed using modern GPU techniques [17, 2, 1, 8]. The main limitation of these algorithms is the impossibility of using the graphics hardware pipeline to perform the necessary ternary depth test. This test includes not only the visible/occluded traditional options, but also an extra merge configuration to allow splat composition of pixels on the same surface. In this way, GPU implementations relied on a two-pass approach: one for visibility and another for rendering. More recently, Zhang and Pajarola [20, 21] proposed a one-pass approach, but it depends on complex structures to separate the splats in non-overlapping groups.

3. Point Rendering Using Image Reconstruction

We propose to extend the use of image processing techniques used in early point-based works, to directly render unstructured point clouds. As will be shown, image reconstruction offers a complexity advantage over splatting, since

it is independent of the number of samples and their sizes. Furthermore, differently from other image reconstruction approaches for point based models, no extra data structure is needed, thus, much less restrictions on sampling density or uniformity are imposed.

An overview of the proposed *Pyramid Point Renderer (PPR)* algorithm is depicted in Figure 1. The input data is an unordered set of three-dimensional points with attributes, which are projected to the viewport. Each projection is rasterized as a single pixel. Then, by means of a pull-push interpolation the continuous surface is reconstructed. In a last pass, deferred shading is applied. These steps are described in the following subsections.

3.1. Point Projection

A point is projected to the viewport, and rasterized as a single pixel. Each point contains, apart from its three dimensional coordinates, a surface normal and a radius indicating the local sampling density. These attributes are enough to reconstruct the surface in image space. However, extra attributes such as color or texture coordinates can also be interpolated. During projection, backface culling is applied, as well as a depth test to deal with multiple projections onto the same pixel.

3.2. Pull-Push interpolation: Pull Phase

The pull-push algorithm was employed by Ogden et al. [14] to interpolate scattered data in two-dimensions. Its main purpose was to reconstructed missing pixels or patches from images. It was later employed in the Lumigraph pipeline [5], and following, in early point-based methods [7, 15]. The algorithm was recently adapted to the GPU by Strengert et al. [19], achieving interpolation times of a few milliseconds for high resolution settings, i.e., 1024^2 and 2048^2 . The pull phase consists of building a hierarchical pyramid of the image by reducing its dimensions by a factor of two. In the subsequent push phase, the pyramid is traversed from top to bottom synthesizing missing information, i.e., empty pixels.

During the pull phase, the pyramid is constructed by creating coarser resolutions of the framebuffer containing the projected samples. In each step, a coarser pixel is computed by averaging the corresponding four pixels of the finer level. However, only valid pixels are included in the average, that is, pixels that contain sample projections. If all four pixels are invalid, the coarser pixel is also marked as so and left to be reconstructed during the next phase.

A sample's projected attributes represent an ellipse in two-dimensions. In other words, it is the projection of the circular extent of the sample in object space. Since the radii are usually very small, the circle is orthogonally projected,

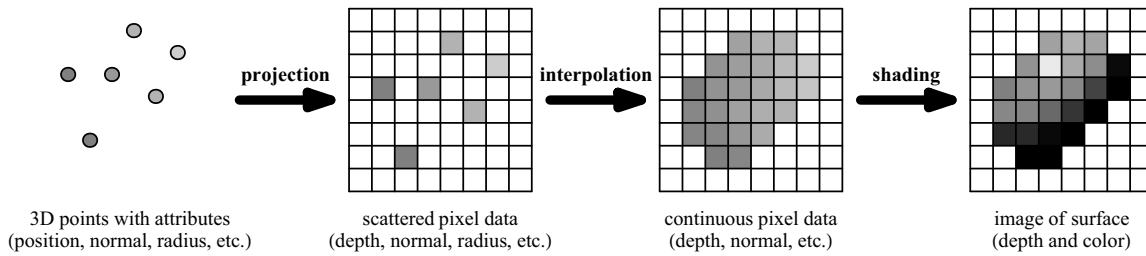


Figure 1. Data flow in the proposed point-based surface rendering technique.

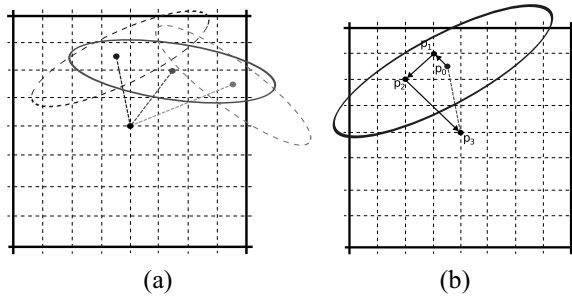


Figure 2. (a) A merge of two ellipses. (b) The displacement vector summed after three iterations of the pull algorithm, where p_0 is the original projected sample at the center of the ellipse, p_1 is the pixel at level 1, and so forth. The dotted line represents the displacement vector for p_3 at level 3.

instead of perspective, without introducing significant errors. Furthermore, an early depth test is employed to discard occluded samples. This is done by comparing the depth component for each sample to be averaged with the front most sample. Occluded pixels are not included in the average, and are left to be recomputed during the push phase.

A displacement vector is the last information stored per pixel. It indicates the 2D distance from the pixel's center to the center of the stored ellipse. An example of the evolution of the displacement vector during the pull phase is illustrated in Figure 2.

3.3. Pull-Push interpolation: Push Phase

Once the pyramid has been built from bottom to top, the push phase reconstructs the missing pixels in the opposite direction. Each invalid pixel computes its value by averaging up to four corresponding pixels in the immediately coarser level. An inside/outside test is carried out to determine if a coarser pixel will be used in the averaging scheme, i.e., if the pixel is inside its elliptical extent. The new at-

tributes are interpolated by averaging the coarser valid pixels, weighted by the elliptical distance from the pixel to the ellipse's center. This is similar to the elliptical kernels used in Surface Splatting.

As in the pull phase, a depth test is also employed here. It is applied to the valid pixels in order to determine if they are occluded and must also be recomputed. The pixel's depth is tested against the depth interval of the pixel directly above it in the pyramid.

3.4. Deferred Shading

After the push phase has been performed for all levels, the highest resolution level contains interpolated attributes for all pixels considered to be inside the reconstructed surface. Since each pixel has a normal vector, Phong shading can be computed in a $O(n)$ pass, where n is the number of pixels. If other attributes were also interpolated, such as color or texture coordinates, they can be included in the shading computation. Figure 3 illustrates the buffer containing the interpolated normals and the final image with deferred shading; while Figure 4 depicts the head model with and without per vertex color interpolation.

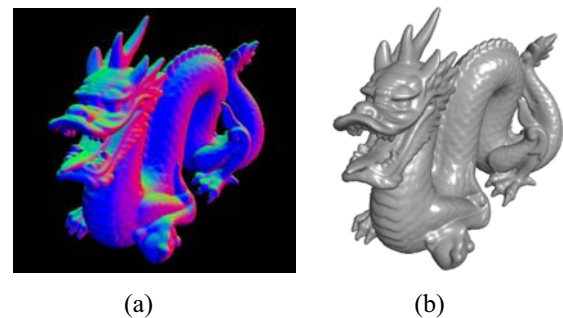


Figure 3. The normal map (a) generated by the pull-push algorithm, and the resulting image (b) after per-pixel shading.

model	# points	Without Color Buffer		With Color Buffer	
		fps	time per frame*	fps	time per frame*
Armadillo	173 K	89	11 ms (1.2 ms, 8.5 ms)	46	22 ms (1.5 ms, 18 ms)
Dragon	437 K	78	13 ms (2.2 ms, 8.9 ms)	44	23 ms (2.9 ms, 18 ms)
Happy Buddha	544 K	76	13 ms (2.6 ms, 8.8 ms)	42	24 ms (3.4 ms, 18 ms)
Asian Dragon	3610 K	36	28 ms (18 ms, 8.3 ms)	23	45 ms (25 ms, 17 ms)
Thai Statue	5000 K	29	35 ms (25 ms, 8.2 ms)	18	55 ms (34 ms, 18 ms)

Table 1. Models and rendering performance. *Each total rendering time per frame is followed in parentheses by the time for the point projection and the time for the pull-push interpolation (all rendering times are in milliseconds).

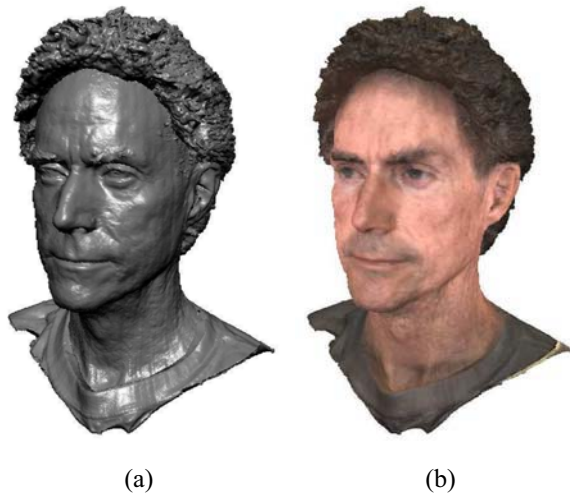


Figure 4. Deferred shading with (a) constant material color and (b) per-vertex diffuse color.

3.5. Results

We tested our algorithm on a GeForce 8800 GTS with 640 MB memory with an Intel Core 2 Duo 6600 CPU (2.4 GHz) with 2 GB RAM. The models were preprocessed to compute a normal vector and a radius of influence of each point.

Rendering times for several point-based models are summarized in Table 1, while exemplary renderings of these models are depicted in Figure 5. The fourth and sixth columns of Table 1 present rendering times in milliseconds without and with interpolation of a surface color, respectively. Apart from the total time per frame, two times in parentheses are also included: the time required for the pro-

jection of points, as described in Section 3.1, and for the pull-push interpolation, discussed in Sections 3.2 and 3.3. As expected, these two operations require most of the rendering time while other operations, e.g., the deferred shading, are almost negligible. Note that the interpolation time is nearly constant for all models. All renderings were performed with a 1024×1024 viewport.

For large models without interpolation of surface color, our implementation renders the equivalent to about 130 M splats per second, including surface reconstruction and deferred shading. If the surface colors are included, the rendering performance is reduced to about 90 M splats per second.

Our previous implementation using a GeForce 7800 GTX achieved the equivalent to between 50 M and 60 M splats per second. For comparison, Zhang and Pajarola [20] reported a performance of up to 24.9 M splats per second, and Guennebaud et al. [8] reported 37.5 M splats per second, both for the same viewport size on the same GPU.

4. Line Rendering Using Image Reconstruction

The work presented in Section 3 was extended to include line strips as rendering primitive. More specifically, two new types are included: flat line strips (ribbons) and cylindrical line strips (tubes). These two primitives are not only illuminated using different methods, but are also reconstructed differently using the pyramid algorithm. An identifier is included with the point sample to allow different types of primitives to be rendered in a single pass. A plant rendering system is described in Section 4.3 to exemplify the use of line primitives.

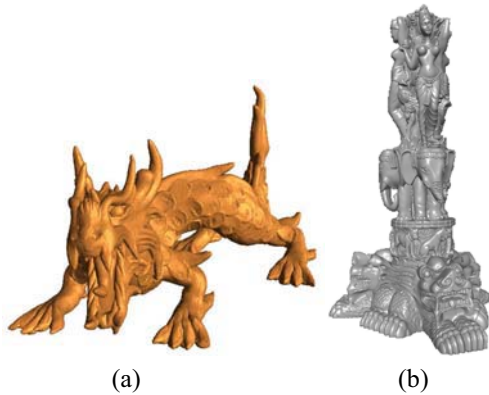


Figure 5. Renderings with our method of the (a) Asian Dragon and (b) Thai Statue.

4.1. Ribbons

In order to use the PPR algorithm to render flat line strips, or ribbons, they must be represented as projected samples in image space. This is achieved by rendering polylines, where each point is attributed with the local normal vector of the ribbon's surface, and half its width as the radius parameter. The rasterized polyline generates a sequence of pixels representing point samples. The scheme is depicted in Figure 6. However, in contrast to the point samples described in the previous section, backface culling is not employed to ribbons. Furthermore, lightning must be computed for both sides.

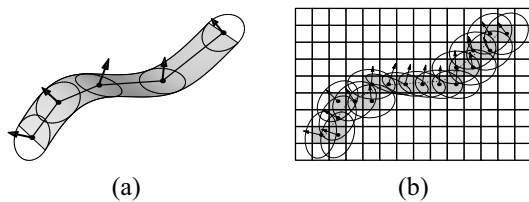


Figure 6. (a) Representation of a ribbon by a polyline consisting of five vertices with normals and radii. (b) Illustration of the rasterization of a one-pixel-wide polyline corresponding to the centerline of a ribbon. Normals and radii are interpolated for each pixel covered by the polyline.

4.2. Tubes

Tubes represent bended cylinders of varying width and curvature, as illustrated in Figure 7a. The tube's centerline is approximated by a polyline, where each vertex has radius equals half the diameter, and normal with the same direction as the tangent vector. Differently from ribbons and point-based surfaces, the interpolation algorithm always considers the normalized vector to the camera center as the surface's normal vector. As shown in Figure 7, this procedure renders disks parallel to the view plane approximating the width of the projected tube.

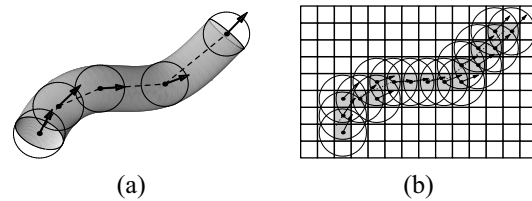


Figure 7. (a) Representation of a tube where each vertex stores a tangent vector and radius. (b) Illustration of the rasterization of a tube's polyline.

The deferred shading of tubes uses a basic approach for diffuse illumination of a line by a single light source. A surface normal n is determined by projecting the light vector l onto the plane orthogonal to the tangent vector t .

4.3. Example: Leaves and Branches

The use of lines is exemplified using a tree model, where branches are approximated by tubes and leaves by ribbons. The apple tree model from the pbrt book [16] is initially converted from polygons to lines. Leaves are converted from four triangles and six vertices to a polyline with four vertices; on the other hand, rings of the branches are heuristically determined to create the representing polylines. Figure 8 illustrates the conversion strategy.

The leaves were converted from 200K triangles and 451K vertices to 150K ribbons with 200K vertices. Likewise, 351K triangles and 371K vertices representing the branches, resulted in 63K tubes with 69K vertices. Only the 227 trunk triangles were not converted, since tubes are not appropriate for representing thick structures. The direct rendering of triangles using the pyramid algorithm is further discussed in Section 5.

The full original and line-based models are presented in Figure 9. Our implementation achieves 31fps against 28fps of the triangle model. However, as the number of primitives

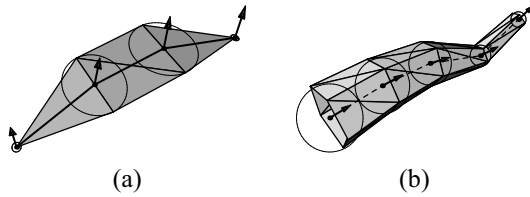


Figure 8. (a) Illustration of the conversion of a leaf (a) and a branch (b) to polylines.

increases so does the difference, e.g., the pyramid algorithm renders 10 trees in 15fps, while the traditional triangle renderer achieves 9fps. In addition, only about one third of the vertices are required for the tubes and ribbons representation.

5. Combining Points, Lines, and Polygons

As stated in Section 1, points complement other representations; it is unlikely one primitive is optimal for all elements of a scene, and LOD strategies may benefit from switching primitives dynamically. Fortunately, our algorithm can easily combine different primitives using the same rendering pipeline.

Apart from points and lines, triangles can also be rendered within the pyramid approach. The vertices are assigned normals as usual, and a radius size close to zero. When a triangle is rasterized, fragments are generated as point samples with radii limited to one pixel, that is, they will not expand during the interpolation. Since the performance of the pull-push is independent of the number of samples, this imposes no overhead. An example of models represented by different primitives is shown in Figure 10.

Even more, hybrid models containing different types of primitives can be similarly rendered. This allows, for example, for a smooth transition between primitive types when working with multiresolution hierarchies. Figure 11 illustrates an example of an hybrid model.

6. Pyramid with Templates

In this section, a new solution based on the pyramid structure is describe to render smoother and more precise silhouettes. This improves on one of the weakest points of the pyramid point renderer: the discontinuity caused by merging ellipses during reconstruction. The problem is more evident at the silhouettes, because some ellipses are only partially rasterized when they are not fully propagated to the highest resolution level.

The only difference in the pull phase, is that each sample is only available in one specific level. This is determined as



(a)



(b)

Figure 9. (a) The original triangle model and (b) the line-based model.

the first level that can cover the elliptical extent with a template of size $k \times k$ pixels. The correct level can be easily computed using the following equation:

$$l = \lceil \log_2 \frac{D}{t_s} \rceil, \quad (1)$$

where D and t_s are, respectively, the projected radius and the template size in pixels. In the same manner as the PPR algorithm, ellipses are merged when needed. Note, however, that less merging occurs since the ellipses are not propagated to the coarsest level. Even more, the larger the kernel the less they propagate, and consequently, the less they merge.

During the reconstruction phase, analogous to the push phase, each pixel in the highest resolution level searches for projections in all coarsest levels. The search is carried out only in a template of size $k \times k$, and, for each projection found, the inside/outside test is performed. If the ellipse



Figure 10. Combined rendering of a point-based model (Dragon) and a triangle-based model (Buddha).

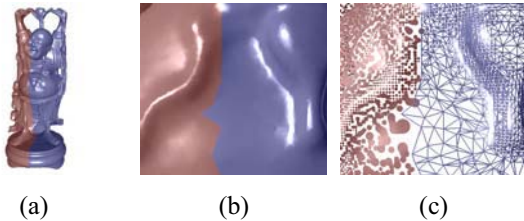


Figure 11. (a) A hybrid surface model with triangles (blue) and points (red). (b) Rendering of a detail and (c) illustration of the underlying primitives.

covers the pixel its contribution is added. The scheme is depicted in Figure 12. Note that this strategy does not create new ellipses during reconstruction, thus avoids discontinuities, i.e., even if ellipses are merged during the pull phase, the new ellipses are faithfully rasterized. In a last pass, deferred shading is also employed.

The template strategy depends on the kernel size k , which is a trade off between speed and quality. A 3^2 kernel achieves performance similar to the pyramid point renderer algorithm. However, since more ellipses merges more artifacts are noticeable. A 5^2 renders high quality results with performance dropping approximately 40%. Kernel sizes larger than 5^2 have not proven to increase the quality significantly. A detail of the Armadillo model rendered with different kernel sizes is shown in Figure 13, while a magnified view of the Neptune model can be seen in Figure 14.

Even with a higher complexity than the original algorithm, $O(n \log n)$ against $O(n)$, this new strategy proves to be a good alternative for high quality rendering. By raster-

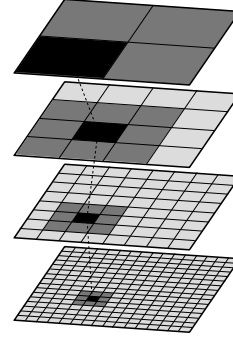


Figure 12. The search coverage (dark gray) for the projected pixel with a 3^2 search template. Painted black is the same pixel represented in different resolutions.



Figure 14. Magnified view of the Neptune model using a 5×5 kernel.

izing the ellipses in a more faithful manner, the results approach the Surface Splatting quality. In fact, if not for the merges during the pull phase, they can, in practice, be identical.

7. Level-Of-Detail for Point Based Models

Image reconstruction of surfaces using point representation proved to be a valuable alternative to traditional methods, mainly because its cost is independent of the number

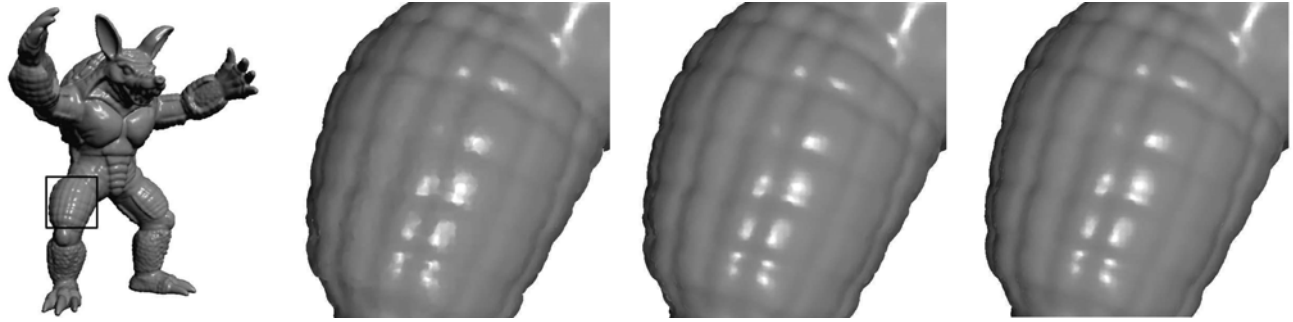


Figure 13. Same scene rendered with three different kernel sizes. From left to right, the kernel sizes are 3^2 , 5^2 , and 7^2 . Note how there is little qualitative gain between the 5^2 and 7^2 kernels, even though the performance drops around 38%.

of projected samples. However, the projection phase is still $O(s)$, where s is the number of samples. Since the reconstruction is carried out in a few milliseconds, the projection time must be decreased in order to improve performance. One obvious way is to reduce the number of projected points. In this work, a Level-Of-Detail structure is employed to determine which points are projected given a viewpoint.

We introduce Object Texture to create our LOD structure. An Object Texture stores the object's primitives grouped in patches, where each patch is associated with a single primitive that is actually sent to the GPU. To retrieve the entire object during rendering, each patch contains information of where its first primitive is stored and how many it contains.

The different resolution levels are created in a simple manner by merging adjacent samples. Each sample in a coarsest resolution level is created by combining up to four adjacent samples from the immediate higher resolution level. The number of levels is limited to four in this application, however, this restriction can be easily extended. The three highest resolution levels are stored in an Object Texture and ordered from coarsest to finest inside each patch. The fourth level, with the coarsest resolution, contains the vertices sent to GPU and used to reference the patches, thus called *patch vertices*. These vertices are coarse resolution samples containing some extra information: the texture coordinates of the first vertex of the patch and the number of vertices in each level.

Each merged sample stores a perpendicular error pre-computed as in *Sequential Trees*[4]. This error estimates the local surface smoothness prioritizing details along the silhouette, while, at the same time, taking into account the distance to the camera position. In the geometry shader, the error of the patch vertex (coarsest resolution) is projected and compared with a threshold to decide which level should be

used. The chosen primitives are projected for rendering using the PPR algorithm, described in Section 3.

Since the geometry shader is still not fully optimized in its initial releases, no significant improvement in performance was noted. Yet, we expect that next generation GPUs shall improve them enhancing the contributions of our LOD structure. This deficiency is mainly attributed to a parameter that sets the maximum number of primitives to be outputted by the geometry shader. As this maximum limit increases, even if it is not reached in practice, performance drops significantly. The number of primitives sent to GPU was approximately one order of magnitude less than the model's samples.

8. Conclusions

With the growing complexity of scenes, new bearings are needed for efficiently processing, modelling and visualizing models. Point based representation is gaining an increasing attention due to the evolution of registration devices. Even though there exists methods for converting point clouds to polygonal meshes, there are several advantages on working directly with points for some applications.

This work presents an image reconstruction method for direct rendering of unstructured point clouds. In general, this approach is a good compromise between image quality and rendering performance for large models. One of its most significant advantages is that the reconstruction cost is independent of the number of projected samples. While the projection phase does not share this characteristic, the computation per sample is very small, thus interactivity is possible for millions of points. By extending the framework to include other primitives, such as lines and triangles, we also improve on the method's applicability.

Even though it is not possible to fully profit from the triangle specialized pipeline of graphics card, the simplic-

ity of points together with image based methods map extremely well to the GPU. This allows for efficient implementations and competitive performance.

The algorithm still lacks appropriate anti-aliasing techniques. However, with the quality improvement gained from the template kernels, much of the artifacts and flickering from the original implementation have been eliminated.

In all, image-space techniques still have some drawbacks, specially due to resampling problems. Nevertheless, the field continues to prove its value by achieving high performance rates while generating quality images.

9. Acknowledgments

We would like to acknowledge the grant of the first author provided by Brazilian agency CNPq (National Counsel of Technological and Scientific Development), and all published works co-authors: Martin Kraus, André Maximo, Antonio Oliveira and Claudio Esperança.

References

- [1] M. Botsch, A. Hornung, M. Zwicker, and L. Kobbelt. High-quality surface splatting on today's gpus. In *PBG '05: Proceedings of the Eurographics Symposium on Point-Based Graphics*, 2005.
- [2] M. Botsch and L. Kobbelt. High-quality point-based rendering on modern gpus. In *PG '03: Proceedings of the 11th Pacific Conference on Computer Graphics and Applications*, page 335, Washington, DC, USA, 2003. IEEE Computer Society.
- [3] C.-F. Chang, G. Bishop, and A. Lastra. Ldi tree: a hierarchical representation for image-based rendering. In *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 291–298, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.
- [4] C. Dachsbacher, C. Vogelsgang, and M. Stamminger. Sequential point trees. *ACM Trans. Graph.*, 22(3):657–662, 2003.
- [5] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen. The lumigraph. In *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 43–54, New York, NY, USA, 1996. ACM Press.
- [6] M. Gross and H. Pfister, editors. *Point-Based Graphics*. Morgan Kaufmann Publishers, 2007.
- [7] J. P. Grossman and W. J. Dally. Point sample rendering. In *Proceedings Eurographics Workshop on Rendering Techniques '98*, pages 181–192, June 1998.
- [8] G. Guennebaud, L. Barthe, and M. Paulin. Splat/Mesh Blending, Perspective Rasterization and Transparency for Point-Based Rendering. In *IEEE/Eurographics/ACM Symposium on Point-Based Graphics, Boston, USA, 29/07/06-30/07/06*, pages 49–58, <http://www.eg.org/>, 2006. Eurographics.
- [9] S. B. Kang, Y. Li, X. Tong, and H.-Y. Shum. Image-based rendering. *Foundations and Trends in Computer Graphics and Vision*, 2(3):173–258, 2006.
- [10] M. Levoy and P. Hanrahan. Light field rendering. In *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 31–42, New York, NY, USA, 1996. ACM.
- [11] M. Levoy and T. Whitted. The use of points as a display primitive. Technical report, University of North Carolina at Chapel Hill, January 1985.
- [12] R. Marroquim, M. Kraus, and P. R. Cavalcanti. Efficient point-based rendering using image reconstruction. In *PBG '07: Proceedings of the Eurographics Symposium on Point-Based Graphics*, pages 101–108, September 2007.
- [13] R. Marroquim, M. Kraus, and P. R. Cavalcanti. Efficient Image Reconstruction for Point-Based and Line-Based Rendering. *Computer Graphics*, 32:189–203, 2008.
- [14] J. Ogden, E. Adelson, J. Bergen, and P. Burt. Pyramid based computer graphics. *RCA Engineer*, 30:4–15, 1985.
- [15] H. Pfister, M. Zwicker, J. van Baar, and M. Gross. Surfels: Surface elements as rendering primitives. In K. Akeley, editor, *Siggraph 2000, Computer Graphics Proceedings*, pages 335–342. ACM Press / ACM SIGGRAPH / Addison Wesley Longman, 2000.
- [16] M. Pharr and G. Humphreys. *Physically Based Rendering: From Theory to Implementation*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2004.
- [17] L. Ren, H. Pfister, and M. Zwicker. Object space ewa surface splatting : A hardware accelerated approach to high quality point rendering. *Computer Graphics Forum*, 21(3):461–470, 2002.
- [18] J. Shade, S. Gortler, L. wei He, and R. Szeliski. Layered depth images. In *SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 231–242, New York, NY, USA, 1998. ACM.
- [19] M. Strengert, M. Kraus, and T. Ertl. Pyramid Methods in GPU-Based Image Processing. In *Workshop on Vision, Modelling, and Visualization VMV '06*, pages 169–176, 2006.
- [20] Y. Zhang and R. Pajarola. Single-pass point rendering and transparent shading. In *Proceedings of the Eurographics/IEEE VGTC Symposium on Point-Based Graphics '06*, pages 37–48, 2006.
- [21] Y. Zhang and R. Pajarola. Deferred blending: Image composition for single-pass point rendering. *Computer & Graphics*, 31(2):175–189, 2007.
- [22] M. Zwicker. *Continuous Reconstruction, Rendering, and Editing of Point-Sampled Surfaces*. PhD thesis, Swiss Federal Institute of Technology, ETH, Zurich, 2003.
- [23] M. Zwicker, H. Pfister, J. van Baar, and M. Gross. Surface splatting. In *Siggraph 2001, Computer graphics Proceedings*, pages 371–378, New York, NY, USA, 2001. ACM Press / ACM SIGGRAPH / Addison Wesley Longman.
- [24] M. Zwicker, H. Pfister, J. van Baar, and M. Gross. Ewa splatting. *IEEE Transactions on Visualization and Computer Graphics*, 8(3):223–238, 2002.

Least Squares and Point-based Surfaces: New Perspectives and Applications

João Paulo Gois

Antonio Castelo Filho

Instituto de Ciências Matemáticas e de Computação
Universidade de São Paulo
{jpgois,castelo}@icmc.usp.br

Abstract

Surface approximation from unorganized points belongs to the state-of-art of computer graphics. In this work, we present approaches for surface reconstruction that are based on efficient numerical schemes for function approximation from scattered data and on sophisticated data structures. In addition, we develop a relevant surface reconstruction method to model moving interfaces, specifically, interfaces of numerically simulated multiphase fluid flow. Finally, from our accumulated experiences on numerical schemes and on the development of surface reconstruction methods, we propose a matrix-free approach for rendering arbitrary volumetric scattered data, which presents interesting properties to be implemented on GPU.

1. Introduction

It is not difficult to find reasons for surface reconstruction methods being a strong tendency in computer graphics. First, despite significant advances, the manipulation of complicate point clouds originated from arbitrary geometries is an important issue that still requires improvements. Second, several research areas have made use of surface reconstruction methods, for instance, reverse engineering, optimization and medicine.

In this work, we present our contributions to surface reconstruction from unorganized points, whose their developments were focused on novel moving-least-squares and partition of unity implicit methods. It will be noticed that our methods have common features of having low computational cost (mainly regarding matricial computations) and being robust. The use of efficient data structures to aid the surface reconstruction is also in the scope of our work [7, 9, 8, 10, 18, 6, 17]. In addition,

we address a surface reconstruction scheme for the front-tracking problem. An important contribution of such scheme is the absence of Lagrangian meshes. In fact, it is designed on the algebraic moving-least-squares method and on a approach for Lagrangian advection of particles. Finally, from our experiences on surface reconstruction and function approximation, we also developed a method for rendering scattered volumetric data based on a recently proposed matrix-free approach.

In the following sections, we describe relevant details of our approaches. Further information and comparisons can be found in [7, 9, 8, 10, 18, 6].

2. Twofold Adaptive Partition of Unity Implicit

We propose an approach based on partition of unity implicit [8, 9] which combines an adaptive and algebraic triangulation, named J_1^a [3], and a recursively defined local approximation method based on multivariate orthogonal polynomials [2].

Similar to other implicit methods for surface reconstruction, the partition of unity implicit defines the reconstructed surface as the zero set of a function F , in a finite domain Ω , from a linear combination of local approximations and weighting functions. For that purpose, we define a set of nonnegative weighting functions with compact support $\Phi = \{\omega_1, \dots, \omega_n\}$, satisfying $\sum_{i=0}^n \omega_i(\mathbf{x}) \equiv 1$, $\mathbf{x} \in \Omega$ and we also consider a set of signed local approximations $\mathbb{F} = \{f_1, \dots, f_n\}$, which completely define the function $F: \mathbb{R}^3 \rightarrow \mathbb{R}$:

$$F(\mathbf{x}) \equiv \sum_{i=0}^n f_i(\mathbf{x}) \omega_i(\mathbf{x}), \quad \mathbf{x} \in \Omega. \quad (1)$$

The *partition of unity* is then defined from a set of nonnegative functions with compact support θ :

$$\omega_i(\mathbf{x}) = \frac{\theta(\|\mathbf{x} - \mathbf{c}_i\|/R_i)}{\sum_{k=1}^n \theta(\|\mathbf{x} - \mathbf{c}_k\|/R_k)}, \quad (2)$$

0 Ph.D. thesis

where $\mathbf{c}_i$ and R_i are the center and the support radius of ω_i , respectively, and $\theta(s) = (1 - s^2)^4$ if $s < 1$; $\theta(s) = 0$ otherwise.

Thus, the implicit surface is given by $\mathcal{S} = \{\mathbf{x} \in \mathbb{R}^3 : F(\mathbf{x}) = 0\}$. Figure 1 illustrates a CSG operation between Neptune model reconstructed from our method and a cylinder.



Figure 1. A CSG operation from Neptune dataset reconstructed using our partition of unity implicits method and a cylinder [9].

Briefly, the main contributions of this approach are:

Adaptive isosurface extraction with topological guarantee: in contrast to previous works, our method is able to extract the isosurface directly from the data structure which subdivides the space (the J_1^a triangulation). In addition, since J_1^a triangulation is defined by tetrahedra, the isosurface algorithm is not affected by ambiguous cases.

Numerical stability: despite their simplicity, canonical polynomial basis leads to ill-conditioned normal equations. In order to avoid such an issue, we make use of the multivariate orthogonal polynomial basis proposed by Bartels and Jezioranski [2], which improves the approximation quality without increasing the computational burden.

Adaptiveness of the local fittings: the use of orthogonal polynomials allows to increase the polynomial degree recursively. For that reason, our method is not only adaptive

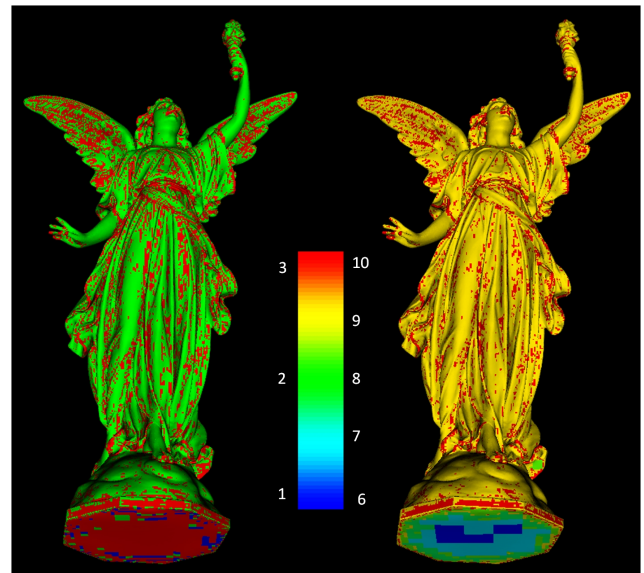


Figure 2. Twofold adaptiveness: Stanford Lucy with 16 M points. Reconstructed model with 7 M triangles. On the left, the color scale represents the polynomial degree of the local approximation in the surface. In the entire domain, the numbers of polynomials with degrees one to four are 946601, 144956, 38236 and 26862, respectively. On the right, the scale color represents the J_1^a triangulation depth level, ranging from 6 to 10 in the surface [9].

with respect to the spatial decomposition, but also with respect to the local fittings, defining a *twofold adaptiveness*. Figure 2 illustrates such a twofold adaptiveness.

Robustness: we propose not only computationally inexpensive, but also effective robustness criteria which allow our method to produce better results than previous works based on partition of unity implicits. In fact, our method is less susceptible to generate spurious surfaces.

Mesh enhancement: the aspect ratios of the triangles generated by the isosurface extractor based on the J_1^a triangulation, in general, are poor. For that reason, we define a simple, but efficient procedure to improve the surface mesh quality. This procedure is based on the displacement of the J_1^a vertices [4]. Figure 6 presents a result attesting that the mesh quality significantly increases. For quantitative analyses, see [9, 6].

Interactive implicit function edition: in order to satisfies the robustness criteria, details in some regions can be poorly approximate, since low degree polynomials are used. On the other hand, neglecting the robustness criteria is not a convenient choice, since spurious surface and artifacts can occur. In addition, increasing the restrictions on the robustness conditions may cause undesirable smoothing of

surface details without ensuring that all imperfections will disappear. Therefore, we developed an interactive implicit function edition which allows to locally change the local fittings without the need of re-computing completely the implicit function (Figures 3-(e)–(f) and 4).

Sharp features modeling: finally, we also properly adapt the scheme from Ohtake et al. [14] for detecting and modeling sharp features to our twofold partition of unity implicits method (Figure 5).

3. Front-tracking with moving-least-squares surfaces

We propose a meshfree front-tracking method for numerical simulation of fluid flows, where the interface is represented by an algebraic-moving-least-squares (AMLS) surface. Our method is interesting for being capable of preserving mass and geometry as well as mesh-based front-tracking methods [19] and being as flexible as level-set methods [15].

Let us again consider a finite set of points $\mathcal{P} = \{\mathbf{p}_1, \dots, \mathbf{p}_m\}$ almost regularly spaced at a distance h and $\{q_k\}$ an algebraic sphere basis [16], i.e., $q_1(\mathbf{x}) = 1$, $q_2(\mathbf{x}) = x_1$, $q_3(\mathbf{x}) = x_2$, $q_4(\mathbf{x}) = x_3$, $q_5(\mathbf{x}) = x_1^2 + x_2^2 + x_3^2$, where $\mathbf{x} = (x_1, x_2, x_3)$. Thus, we define a function $f_\alpha(\mathbf{x})$ whose zero level approximates $\mathcal{P}$:

$$f_\alpha(\mathbf{x}) = \sum_{k=1}^5 \alpha_k q_k(\mathbf{x}) \quad (3)$$

where $\alpha = (\alpha_1, \dots, \alpha_5)$ are the coefficients which completely defines f_α and α satisfies:

$$\alpha = \arg \min_{\beta \in Q} \sum_{j=1}^m w_j |f_\beta(\mathbf{p}_j)|^2, \quad (4)$$

where $w_j = w(\mathbf{x}) = \theta\left(\frac{\|\mathbf{x} - \mathbf{p}_j\|}{3h}\right)$, defining non-negative weights $\theta(s) = (1 - s^2)^4$ if $s < 1$; $\theta(s) = 0$ otherwise) and $Q = \{\beta \in \mathbb{R}^5 : \beta_2^2 + \beta_3^2 + \beta_4^2 - 4\beta_1\beta_5 = 1\}$, being the restriction used to ensure that $f_\alpha(\mathbf{x})$ approximates the signed distance of $\mathbf{x}$ to the surface $f_\alpha \equiv 0$.

The minimization (4) can be computed by a generalized eigenvalue problem in $\mathbb{R}^5$ [11]. In addition, for each point $\mathbf{p}_i$, the approximate normal is given by:

$$\mathbf{N}_i = \nabla f_{\alpha_i}(\mathbf{p}_i) / \|\nabla f_{\alpha_i}(\mathbf{p}_i)\|. \quad (5)$$

The sphere-fitting problem, previously described, can be substantially improved by introducing the normals [11]. Furthermore, the introduction of normal in the minimization problem allows the splitting of the problem in two main steps. First, we compute the subset:

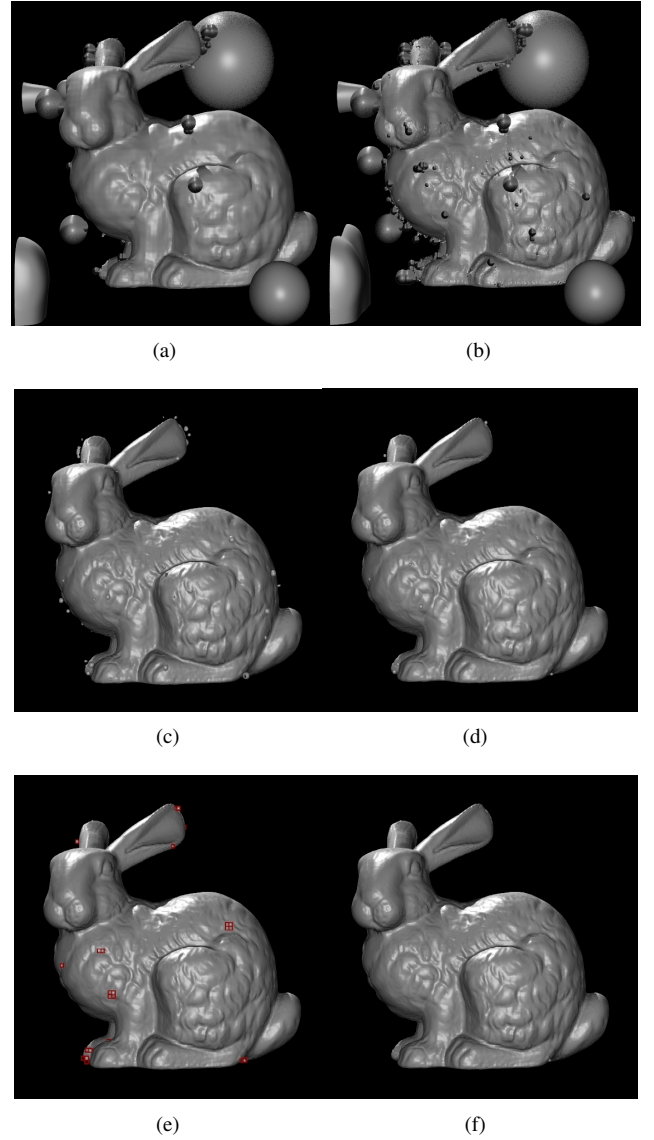


Figure 3. Interactive function edition: (a)-(b) present results from Ohtake et al. [14] with their default parameters and those one that we suggested, respectively; (c) our method but not using the coverage domain criterion [9]; (d) our method making use of the coverage domain criterion; (e) selecting surface artifacts; (f) the edited function eliminates imperfections [9].

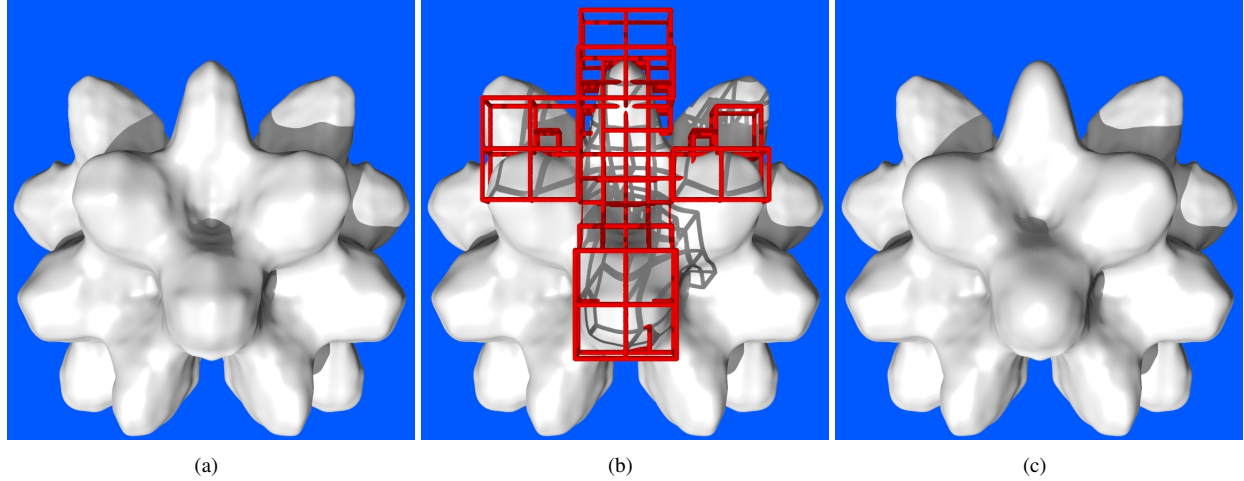


Figure 4. Reconstruction enhancement by the interactive function edition: (a) original result with low degree approximations, (b) support selection for function edition (c) final result [9].

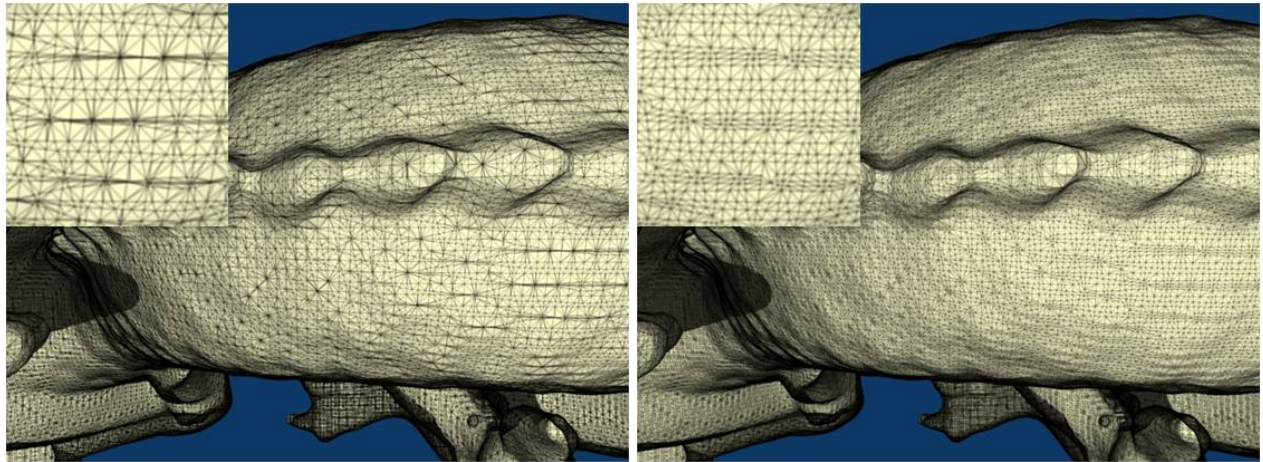


Figure 6. Mesh enhancement: Comparison between the mesh produced by the J_1^a triangulation without the displacement vertices procedure (left) and the mesh produced by the displacement vertices procedure (right) [9].

$\hat{\gamma}(\mathbf{x}) = (\gamma_2(\mathbf{x}), \dots, \gamma_5(\mathbf{x}))$ of $\gamma = (\gamma_1, \dots, \gamma_5)$ finding:

$$\hat{\gamma}(\mathbf{x}) = \arg \min_{\hat{\beta} \in \mathbb{R}^4} \sum_{i=1}^m w_i \|\nabla f_{\hat{\beta}}(\mathbf{p}_i) - \mathbf{N}_i\|^2. \quad (6)$$

where $\mathbf{N}_i$ is computed as previously described. Notice that the solution $\hat{\gamma}(\mathbf{x})$ of (6) can be determined simply by solving a 4×4 linear system.

Finally, the first term γ_1 of γ is computed by:

$$\sum_{i=1}^m w_i |F_{\hat{\beta}}(\mathbf{p}_i)|^2 \quad (7)$$

where $\hat{\beta} = \hat{\gamma}(\mathbf{x})$. Thus:

$$\gamma_1(\mathbf{x}) = - \frac{\sum_{i=1}^m w_i (\sum_{k=2}^5 \gamma_k(\mathbf{x}) q_k(\mathbf{p}_i))}{\sum_{i=1}^m w_i}. \quad (8)$$

We name such an approximation *RAMLS-Robust Algebraic Moving Least-Squares*. It can be seen as an improvement of the Guennebaud and Gross method [11], since we eliminate the parameter that weighs the normal restriction into the minimization problem (in fact, we assume that the parameter β , in [11], as $\beta \rightarrow \infty$).

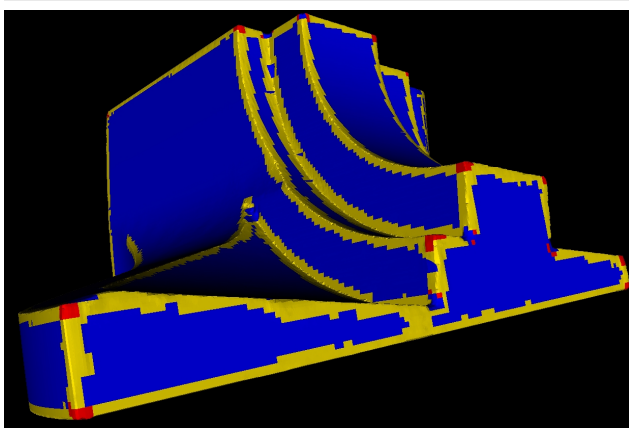


Figure 5. Sharp features detection and modeling (Fan Disk dataset): in yellow are the faces labeled as *sharp edges*, whereas in red are the faces labeled as *sharp corners* [9].

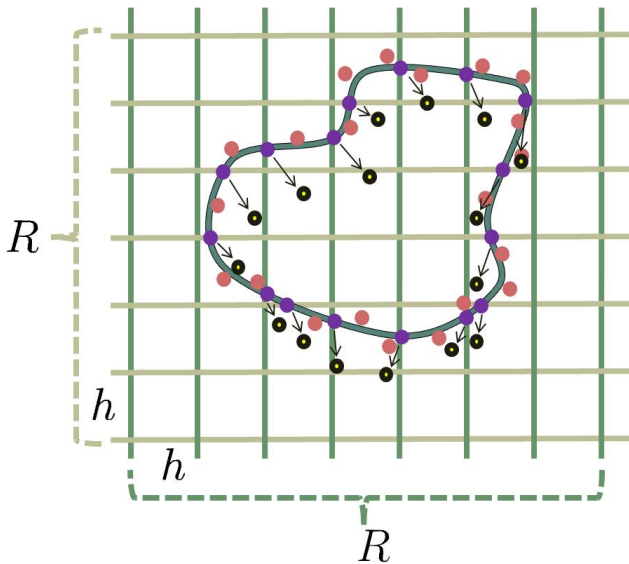


Figure 7. The set of rays R and the points $\mathcal{P}(t)$ (red) define the RAMLS surface $\mathcal{S}(t)$ (green curve). The new set of points $\mathcal{P}(t)$ (purple) is created at the intersections between R and $\mathcal{S}(t)$. The set of points is, thus, transported, defining the set $\mathcal{P}(t+\Delta t)$ (black points), which will define the new surface RAMLS $\mathcal{S}(t+\Delta t)$.

3.1. Point Advection

In order to update the RAMLS surface during the simulation, we move the points $\mathcal{P}$ according to the velocity field $\mathbf{v}$. Mathematically, we solve the following equation:

$$\frac{d\mathbf{p}_i}{dt} = \mathbf{v}(\mathbf{p}_i(t), t), \quad (9)$$

where $\mathbf{p}_i(0)$ are the initial points.

We denote by $\mathcal{P}(t) = \{\mathbf{p}_i(t)\}_{i=1,\dots,m}$ the set of points to be advected from t to $t+\Delta t$ (Δt is the time step). During the transport, the points in $\mathcal{P}(t)$ can become poorly distributed, which could compromise the surface $\mathcal{S}(t)$. In order to avoid such an issue, we propose to re-generate the point set periodically in a regularly spaced grid.

Re-generation points process: Let us consider a set of regularly spaced rays R with spacing h (Figure 7). The set of points $\mathcal{P}(t)$ is transported along the trajectories given by Equation 9. The point transportation is used to define the new RAMLS surface. The intersections between R and the surface define the new set of points $\mathcal{P}(t+\Delta t)$. We depict such a re-generation process in Figure 7.

It is important to mention that Equation 9 is solved by the fourth-order Runge-Kutta method. A ray-tracing-like algorithm is used to detect the intersections between R and the RAMLS surface. Since points can be re-generated very close, we remove points that are closer than $\frac{h}{2}$. Finally, the normals remains unchanged during the transportation in order to be used to orient the new normals in $\mathcal{P}(t+\Delta t)$.

Considering the convergence rates, for the two-dimensional case, we are able to show that RAMLS accomplishes order $O(h^3)$, $O(h^2)$ and $O(h)$ for geometrical, normal and curvature, respectively. Considering the three-dimensional case, RAMLS achieves rates of $O(h^2)$ and $O(h)$ for geometrical and normal curvature, respectively. During the transportation, the error is $O(h^r/\Delta t)$, where $r = 2, 3$ according to the geometrical error, previously argued.

For the Zalesak Sphere Test, we consider a sphere with radius 0.15, slope with length 0.10 and width 0.20, centered at (0.5, 0.75, 0.75) in a unity cube. The velocity field is given by a rigid rotation:

$$\mathbf{v}(x_1, x_2, x_3) = \frac{\pi}{314} (0.5 - x_2, x_1 - 0.5, 0). \quad (10)$$

We assume $\Delta t = 1$ and a complete turn takes 628 time steps. We show the geometry of the Zalesak spheres at the times $t = 0, 79, 157, 236, 314, 393, 471, 550$ and 628 in Figure 8. We adopt the mesh size $h = 1/256$, and the number of particles as about 19000 for each time step. It can be noticed that the sharp features become smooth along the time, but the volume and shape remain similar [7, 6].

We also test our method to a three-dimensional deforming problem. We consider a sphere as $\mathcal{S}(0)$ with

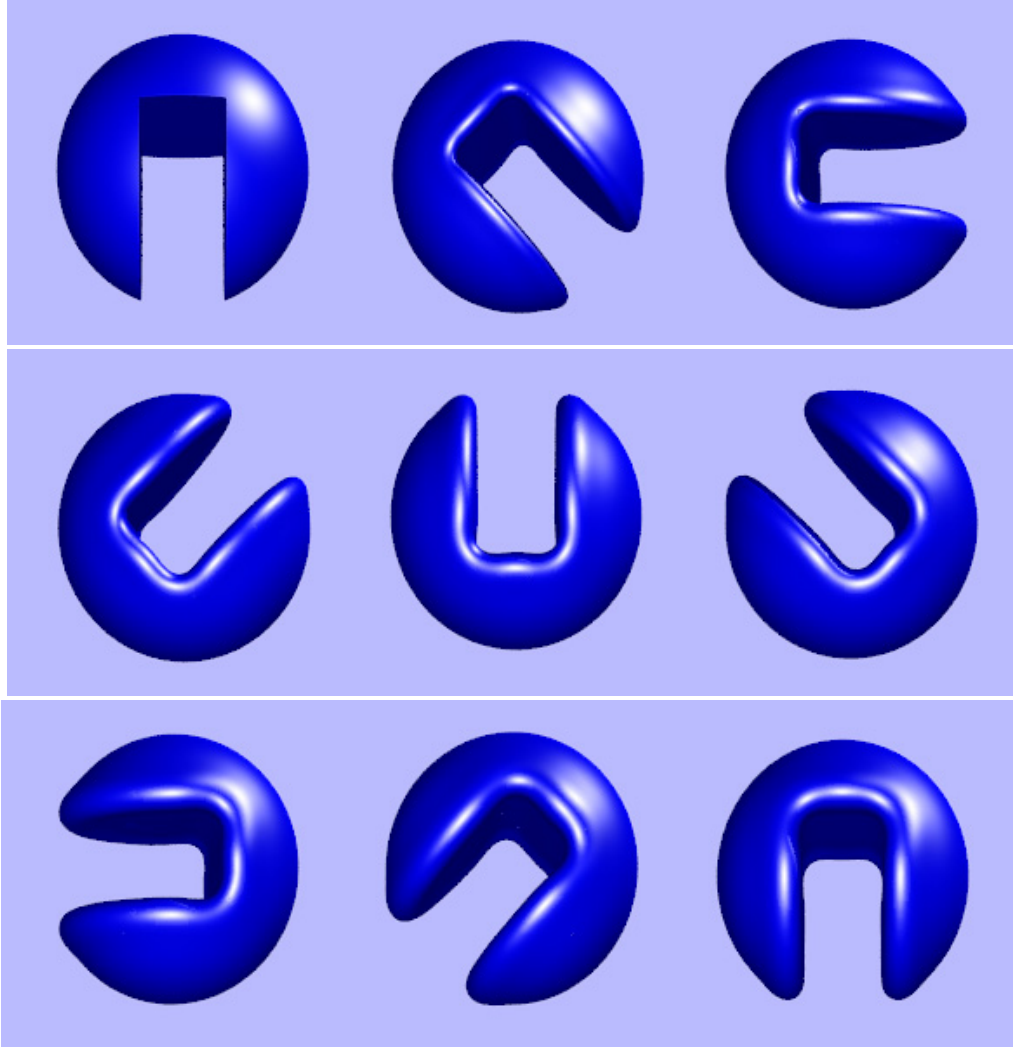


Figure 8. Zalesak sphere rotation ($h = 1/256$): (about 19000 points) and time steps $t = 0, 79, 157, 236, 314, 393, 471, 550$ and 628 time units (from left to right and from top to bottom) [7].

radius 0.15, centered at (0.35, 0.35, 0.35). The computational domain is the unity cube. The velocity field is given by:

$$\mathbf{v}(x_1, x_2, x_3) = \begin{pmatrix} 2 \sin^2(\pi x_1) \sin(2\pi x_2) \sin(2\pi x_3) \\ -\sin(2\pi x_1) \sin^2(\pi x_2) \sin(2\pi x_3) \\ -\sin(2\pi x_1) \sin(2\pi x_2) \sin^2(\pi x_3) \end{pmatrix}.$$

At $t = T/2 = 1$ time units, we reverse the velocity field in order to check the capability of the method in preserving topology and geometry. Figure 9 presents results of such a simulation, using $h = 1/512$, $\Delta t = 0.0064$ and 628 time steps.

4. Iterate Approximate Moving Least Squares Surfaces

One of the challenges to develop efficient surface reconstruction methods in modern GPU is the conformance of numerical computations to textures [18], represented by 4×4 matrices.

When implementing a moving-least-squares surface method, in order to achieve high order approximations, the local approximations must be constructed by higher order polynomial basis, and consequently, a single 4×4 matrix does not allow to achieve efficiently such a requirement. Studies to select a reduced polynomial basis [18, 10] have been done. However, the local approximation quality and high order approximations are not ensured.

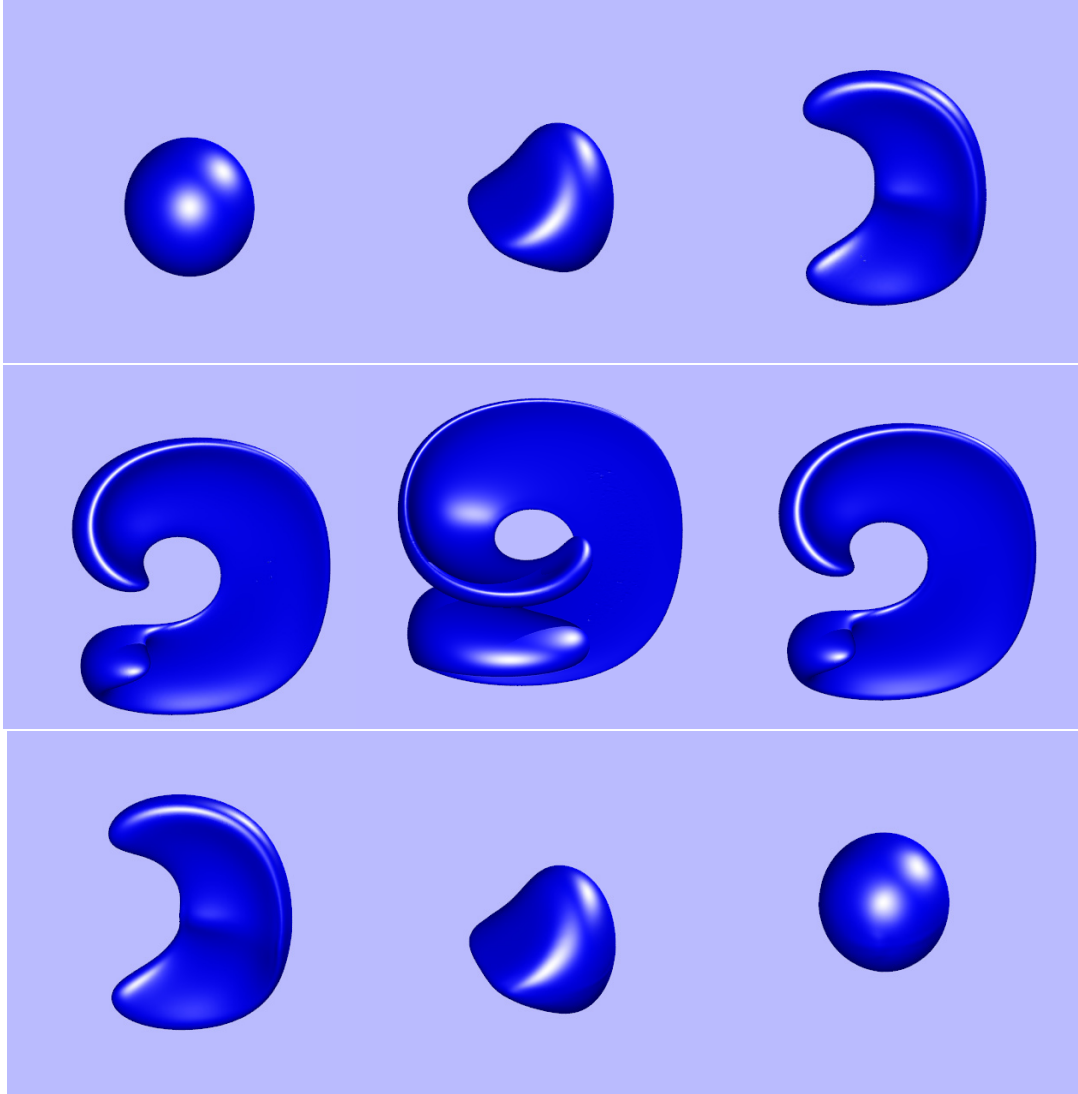


Figure 9. Three-dimensional torsion ($h = 1/512$): the number of points for each set is (from left to right and from top to bottom): 65000, 69314, 109728, 187571, 277021, 193238, 114898, 72774 and 68031 [7].

To overcome the matrix reliance, we propose an iterative method based on the “iterate approximate moving least squares – iaMLS” [5]. This method is interesting for being able to enhance features simply by controlling the number of iterations and the single smooth parameter ϵ . Formally, let us consider a function $F : \mathbb{R}^3 \rightarrow \mathbb{R}$, which its zero set approximates the surface. Let us also consider a set of points $\mathbf{x}_i$, equipped with their normal vector $\mathbf{n}_i$. Thus, we define:

$$F(\mathbf{x}) = \sum_{i=1}^m \mathcal{G}_i \Psi_i(\mathbf{x}), \quad (11)$$

where $\mathcal{G}_i = \langle \mathbf{x}_i - \mathbf{x}, \mathbf{n}_i \rangle$, $\Psi_i(\mathbf{x}) = \frac{\epsilon^3}{\pi^{3/2}} (r_i(\mathbf{x})) \exp(-r_i(\mathbf{x}))$, $r_i = \frac{\|\mathbf{x} - \mathbf{x}_i\|}{h}$ and h is the average spacing between the points

[6]. The functions Ψ_i are named second order generating functions [5].

From these definitions, and making use of the iterative process of the iaMLS [5], we define the *iaMLS surface* from the following iterative process:

$$F^{(0)}(\mathbf{x}) = \sum_{i=1}^m \mathcal{G}_i \Psi_i(\mathbf{x}) \quad (12)$$

$$F^{(n+1)}(\mathbf{x}) = F^{(n)}(\mathbf{x}) + \sum_{i=1}^m [\mathcal{G}_i - F^{(n)}(\mathbf{x}_i)] \Psi_i(\mathbf{x}), \quad (13)$$

where $F \equiv 0$ gives the implicit surface reconstruction.

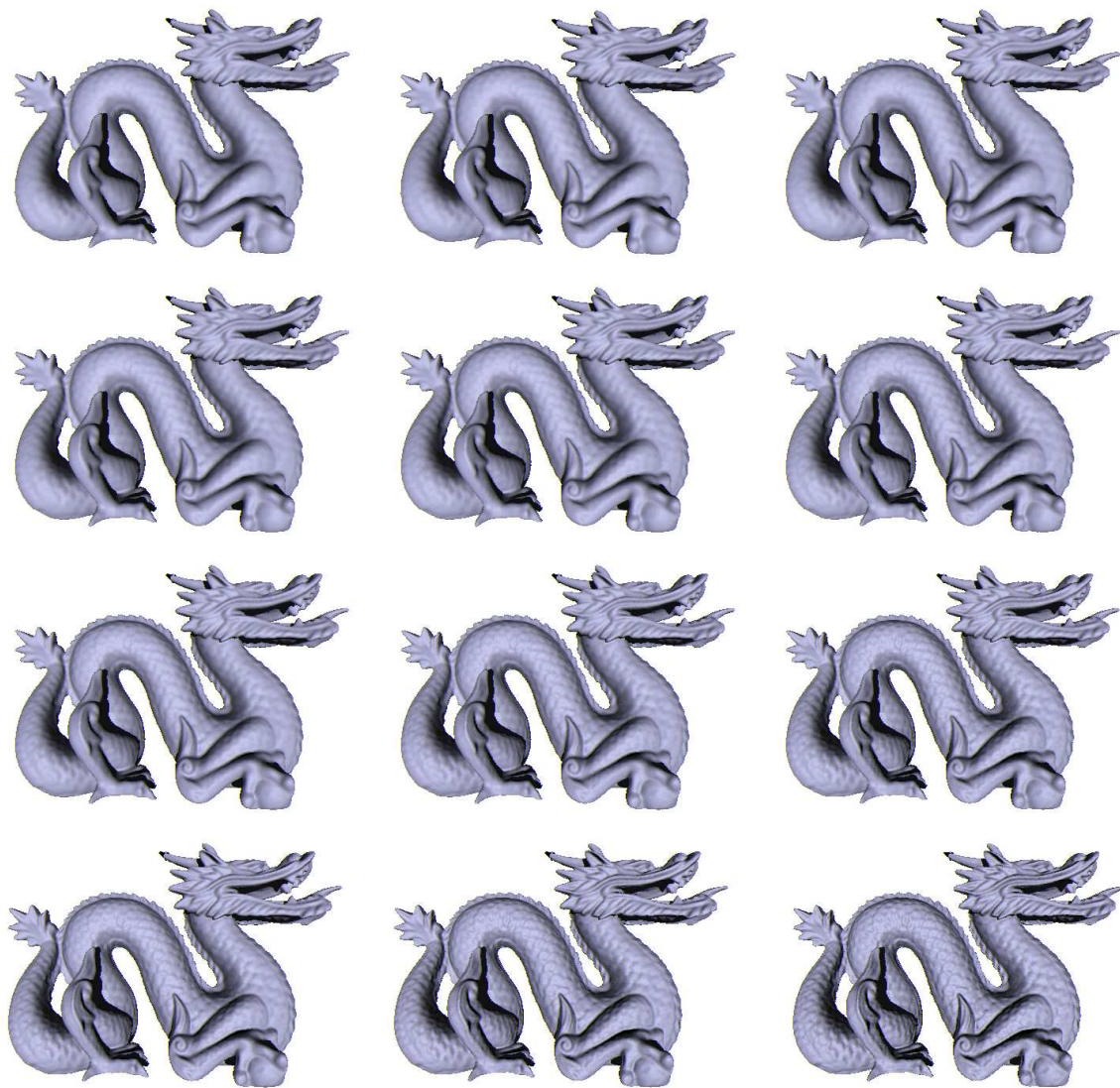


Figure 10. Effects caused by the number of iterations and the ϵ parameter. From left to right: 3,4 and 5 iterations. From top to bottom: $\epsilon = 0.4$, $\epsilon = 0.8$, $\epsilon = 1.22$, $\epsilon = 2.0$. The greater the number of iterations or ϵ are, the more detailed the surface becomes.

Figure 10 depicts how the number of iterations and the parameter ϵ affect the solution. Figure 11 presents comparisons of our approach with other implicit MLS surface methods. It can be noticed that, even using a small number of iterations, we are able to enhance object characteristics. We present comparisons assessing the computational cost of this scheme [6, 17].

5. Volumetric approximation from unorganized points

The last contribution of this work is related to unorganized volumetric data rendering. In fact, the goal is the definition of a method for rendering arbitrary meshes, i.e., a unified approach that explores the iaMLS. Firstly, in order to make the method more efficient for all mesh types, we make use of anisotropic spaces. Notice that the original iaMLS, to the best of our knowledge, does not make use of any anisotropy scheme [6, 17].

Our method is also able to render isosurfaces from

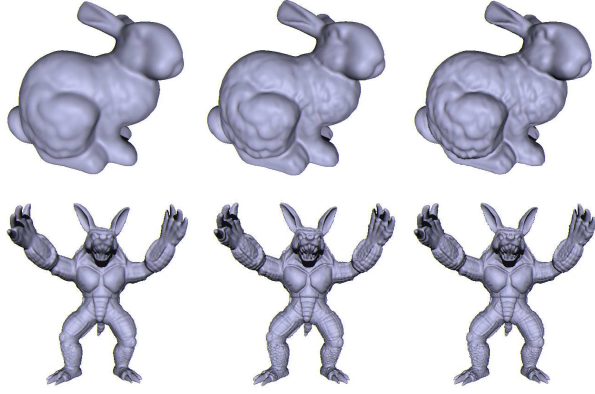


Figure 11. Comparison among Adamson and Alexa [1] (left), Kolluri [12] (middle) and iaMLS [6, 17] (right) methods. We use the same h parameter for each model. We performed 3 iterations and $\epsilon = 0.8$.

unorganized volumetric data. For that purpose, we also apply the iaMLS to estimate the gradient function [6, 17]. In fact, it is required to define, for each mesh vertex $\mathbf{x}_i$, a gradient vector estimate, using a least-squares approach. It is achieved by extending the approach by Mavriplis to the three-dimensional space [13]. It is important to notice that the use of such an approach remains our method independent of meshes.

Let us consider a mesh vertex $\mathbf{x}_i$, in which we evaluate the gradient, and its neighbors $\mathbf{x}_k = (x_k, y_k, z_k)$ in its star $\mathbb{V}_i$. Thus, we minimize the following equation with respect to $\nabla f(\mathbf{x}_i) = ((f_x)_i, (f_y)_i, (f_z)_i)$:

$$\sum_{\mathbf{x}_k \in \mathbb{V}_i} \exp(-\langle \mathbf{x}_k - \mathbf{x}_i, \mathbf{x}_k - \mathbf{x}_i \rangle) (E_{ik})^2 \quad (14)$$

where:

$$E_{ik} = ((f_x)_i \cdot dx_{ik} + (f_y)_i \cdot dy_{ik} + (f_z)_i \cdot dz_{ik} - df_{ik}) \quad (15)$$

and $df_{ik} = f(\mathbf{x}_k) - f(\mathbf{x}_i)$. By the same token, $dx_{ik} = x_k - x_i$, $dy_{ik} = y_k - y_i$, $dz_{ik} = z_k - z_i$.

The gradient vector, in an arbitrary point $\mathbf{x}$, indeed, is given by the iterative process:

$$\mathcal{Q}_{\nabla f}(\mathbf{x})^{(0)} = \sum_{\mathbf{x}_i \in \mathbb{X}} \nabla f(\mathbf{x}_i) \Psi_i(\mathbf{x}), \quad (16)$$

$$\mathcal{Q}_{\nabla f}^{(n+1)}(\mathbf{x}) = \mathcal{Q}_{\nabla f}^{(n)}(\mathbf{x}) + \sum_{\mathbf{x}_i \in \mathbb{X}} \left[\nabla f(\mathbf{x}_i) - \mathcal{Q}_{\nabla f}^{(n)}(\mathbf{x}_i) \right] \Psi_i(\mathbf{x}). \quad (17)$$

Another interesting feature of our approach is its capability to be efficient for GPU implementation. We are

able to provide pre-computations, which are stored in GPU textures. We perform two pre-computations based on the iterative process of iaMLS. They are related to the function evaluation and the gradient estimate.

Thus, we can rewrite the iterative process by Fasshauer [5] as:

$$\mathcal{Q}_f^{(n+1)}(\mathbf{x}) = \sum_{\mathbf{x}_i \in \mathbb{X}} \left[f(\mathbf{x}_i) + \sum_{j=0}^n \left(f(\mathbf{x}_i) - \mathcal{Q}_f^{(j)}(\mathbf{x}_i) \right) \right] \Psi_i(\mathbf{x}), \quad (18)$$

where, we can accumulate the results for each mesh vertex as:

$$g(\mathbf{x}_i) = f(\mathbf{x}_i) + \sum_{j=0}^n \left(f(\mathbf{x}_i) - \mathcal{Q}_f^{(j)}(\mathbf{x}_i) \right), \quad (19)$$

and store them in textures which accommodate the unorganized volumetric data. Then, during the rendering, the reconstructed function value is simply given by:

$$\mathcal{Q}_f^{(n+1)}(\mathbf{x}) = \sum_{\mathbf{x}_i \in \mathbb{X}} g(\mathbf{x}_i) \Psi_i(\mathbf{x}). \quad (20)$$

Rendering results are presented in Figure 12 for different meshes. We use $h = 0.25$ and $\epsilon = 0.9$. Both parameters are chosen empirically after numerical tests.

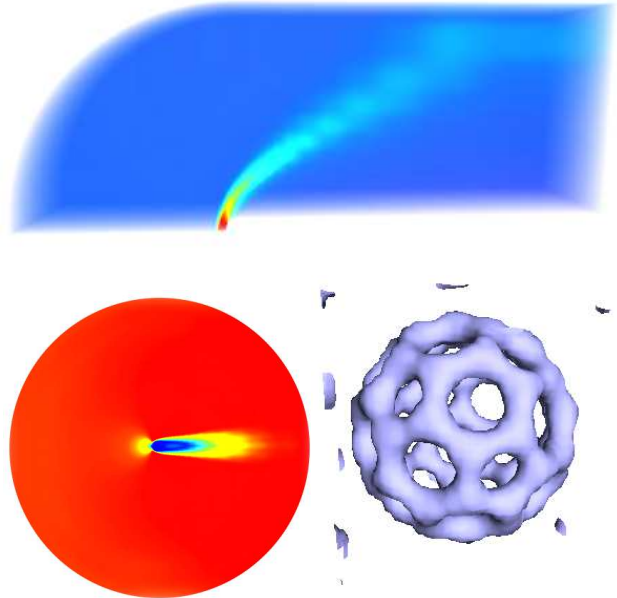


Figure 12. Volumetric rendering of Blunt Fin and Oxygen Post datasets. Isosurface extracted from the Bucky Ball dataset [6, 17].

6. Conclusion

In this work, we presented novel and important contributions towards the field of dynamic and static surface reconstruction methods from unorganized points.

There are several ways to extend the techniques here presented. First, we intend to investigate theoretically our surface reconstruction method based on iaMLS.

With respect to the twofold adaptive partition of unity implicits, we intend further to improve the function edition process by incorporating new local approximating functions. In addition, we want to improve the mesh enhancement technique, making use of the whole scheme by Dietrich et al. [4], but, in our case, considering the adaptiveness of J_1^a triangulation.

Considering the front-tracking with AMLS surfaces approach, the definition of a good curvature from RAMLS is a mandatory future work. In addition, we aim at applying this surface representation for numerical simulation of fluid flows. We are also investigating schemes to perform topological changes efficiently.

Finally, we intend to investigate the possibility to estimate automatically the parameters h and ϵ of the iaMLS defined in anisotropic domains for volumetric rendering.

Acknowledgments

This work was supported by FAPESP (Brazil) and DAAD (Germany). The methods in Sections 2 and 5 were developed also with Eduardo Tejada and Tomas Ertl (University of Stuttgart–Germany). The work in Section 4 was developed also with Gustavo C. Buscaglia and Anderson Nakano. The work on Section 3 was also developed together with Tiago Etienne, Valdecir Polizelli, Eduardo Tejada and Thomas Ertl. We gratefully thank Luis G. Nonato, an important collaborator for all of these works.

References

- [1] A. Adamson and M. Alexa. Ray-tracing point set surfaces. In *Proceedings of the Shape Modeling International*, pages 272–279, 2003.
- [2] R. H. Bartels and J. J. Jezioranski. Least-squares fitting using orthogonal multinomials. *ACM Transactions on Mathematical Software*, 11(3):201–217, 1985.
- [3] A. Castelo, L. G. Nonato, M. Siqueira, R. Minghim, and G. Tavares. The j1a triangulation: An adaptive triangulation in any dimension. *Computer & Graphics*, 30(5):737–753, 2006.
- [4] C. Dietrich, C. Scheidegger, J. Schreiner, J. Comba, L. Nedel, and C. Silva. Edge transformations for improving mesh quality of marching cubes. *IEEE Transactions on Visualization and Computer Graphics*, accepted.
- [5] G. Fasshauer and J. Zhang. Iterated approximate moving least squares approximation. In C. A. V. M. A. Leitao and C. A. Duarte, editors, *Advances in Meshfree Techniques*, to appear. Springer, 2007.
- [6] J. P. Gois. *Mínimos-Quadrados e Aproximação de Superfície de Pontos: Novas Perspectivas e Aplicações*. PhD thesis, Instituto de Ciências Matemáticas e de Computação – Universidade de São Paulo, 2008.
- [7] J. P. Gois, A. Nakano, L. G. Nonato, and L. C. Buscaglia. Front tracking with moving-least-squares surfaces. *Journal of Computational Physics*. DOI: 10.1016/j.jcp.2008.07.013
- [8] J. P. Gois, V. Polizelli-Junior, T. Etienne, E. Tejada, A. Castelo, T. Ertl, and L. G. Nonato. Robust and adaptive surface reconstruction using partition of unity implicits. In *Brazilian Symposium on Computer Graphics and Image Processing*, pages 95–102, IEEE CS, 2007.
- [9] J. P. Gois, V. Polizelli-Junior, T. Etienne, E. Tejada, A. Castelo, L. G. Nonato, and T. Ertl. Twofold adaptive partition of unity implicits. *The Visual Computer*. DOI: 10.1007/s00371-008-0297-x
- [10] J. P. Gois, E. Tejada, T. Etienne, L. G. Nonato, A. Castelo, and T. Ertl. Curvature-driven modeling and rendering of point-based surfaces. In *Brazilian Symposium on Computer Graphics and Image Processing*, pages 27–36. IEEE CS, 2006.
- [11] G. Guennebaud and M. Gross. Algebraic point set surfaces. In *ACM SIGGRAPH 2007*, Article 23
- [12] R. Kolluri. Provably good moving least squares. In *Proc. of the 16th annual ACM-SIAM symposium on Discrete algorithms*, pages 1008–1017, 2005.
- [13] D. J. Mavriplis. Revisiting the least-squares procedure for gradient reconstruction on unstructured meshes. Technical Report CR-2003-212683, NASA, 2006.
- [14] Y. Ohtake, A. Belyaev, M. Alexa, G. Turk, and H.-P. Seidel. Multi-level partition of unity implicits. *ACM Trans. Graph.*, 22(3):463–470, 2003.
- [15] S. Osher and R. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*, volume 153 of *Applied Mathematical Sciences*. Springer, 2003.
- [16] V. Pratt. Direct least-squares fitting of algebraic surfaces. *SIGGRAPH*, 21(4):145–152, 1987.
- [17] E. Tejada. *Towards Meshless Volume Visualization*. PhD thesis, University of Stuttgart, Germany, 2008.
- [18] E. Tejada, J. P. Gois, A. Castelo, L. G. Nonato, and T. Ertl. Hardware-accelerated extraction and rendering of point set surfaces. In *EUROGRAPHICS - IEEE VGTC Symposium on Visualization*, 2006.
- [19] G. Tryggvason, B. Bunner, A. Esmaeeli, D. Juric, N. Al-Rawahi, W. Tauber, J. Han, S. Nas, and Y. J. Jan. A Front-Tracking Method for the Computations of Multiphase Flow. *J. Comput. Phys.*, 169:708–759, 2001.

Renderizações Não Fotorealísticas para Estilização de Imagens e Vídeos usando Areia Colorida

Laurindo S. Britto Neto*

Departamento de Ciências da Natureza/Picos - UFPI
Rua Cícero Eduardo, S/N, Junco, Picos, PI, 64.600-000 - Brasil
laurindobneto@gmail.com

Bruno M. Carvalho (Orientador)

Departamento de Informática e Matemática Aplicada - UFRN
Campus Universitário, S/N, Lagoa Nova, Natal, RN, 59.072-970 - Brasil
bruno_m_carvalho@yahoo.com

Abstract

Non-Photorealistic Rendering (NPR) can be defined as the processing of scenes, images or videos into artwork. This work presents a new method of NPR for stylization of images and videos, based on a typical artistic expression of the Northeast region of Brazil, that uses colored sand to compose landscape images on the inner surface of glass bottles. This method is comprised by one technique for generating 2D procedural textures of sand, and two techniques that mimic effects created by the artists using their tools. We also present a method for generating $2\frac{1}{2}D$ animations of stylized videos as if they were placed in a sandbox. The temporal coherence within these stylized videos can be enforced on individual objects with the aid of a video segmentation algorithm.

Resumo

Renderização Não Fotorealística (NPR) pode ser definida como o processamento de cenas, imagens ou vídeos para geração de trabalhos artísticos. Este trabalho apresenta um novo método de NPR para estilização de imagens e vídeos baseado em uma expressão artística típica da região Nordeste do Brasil, que usa areia colorida para compor imagens de paisagens na superfície interna de garrafas de vidro. Este método possui uma técnica para geração de texturas procedurais de

areia 2D, e duas técnicas que imitam os efeitos criados pelos artesãos usando sua ferramenta. Além disso, apresentamos um método para geração de animações $2\frac{1}{2}D$ em caixas de areia a partir do vídeo estilizado. A coerência temporal nos vídeos estilizados pode ser obtida nos objetos individuais do vídeo com auxílio de um algoritmo de segmentação de vídeo.

1. Introdução

A *Renderização Não Fotorealística (Non-Photorealistic Rendering* ou NPR) é uma classe de técnicas definida de forma negativa a partir da renderização fotorealística, ou seja, seu objetivo é a criação de técnicas alternativas à renderização de imagens que buscam reproduzir cenas fotorealísticas, e é também conhecida como renderização estilizada. Outra forma de definir a NPR é como o processamento de cenas, imagens ou vídeos em trabalhos de arte, gerando cenas, imagens ou vídeos que podem ter o atrativo visual de peças de arte, expressando características visuais e emocionais de estilos artísticos.

Neste trabalho desenvolvemos um método de renderização não fotorealística para a estilização de imagens e vídeos com areia colorida denominado *Csand*, do inglês “*colored sand*”. Esse método é baseado na confecção artesanal de garrafas com areia colorida, um estilo artístico típico da região Nordeste do Brasil, desenvolvido principalmente nos estados do Rio Grande do Norte e Ceará. Desenvolvemos também um método para a criação de animações $2\frac{1}{2}D$ em caixas de areia a partir do vídeo estilizado. Desta forma, as principais contribuições do trabalho são:

* Este artigo completo contém trechos da dissertação de mestrado [3] do primeiro autor, defendida no PPgSC do DIMAp/UFRN. Esta dissertação é apoiada pelo CNPq sobre o processo PDPG-TI 506555/04-6. A versão completa e vídeos relacionados estão disponíveis em: <http://www.ufpi.br/laurindoneto>.

- Criação do método de renderização não fotorealística *Csand*;
- Criação de uma técnica para geração de texturas procedurais de areia 2D;
- Criação de duas técnicas que simulam efeitos produzidos pelos artesãos, com sua ferramenta, na confecção das garrafas com areia colorida;
- Criação de um método para a geração de animações $2\frac{1}{2}$ D em caixas de areia.

2. Trabalhos Relacionados

Os trabalhos relacionados com a renderização não fotorealística para estilização de imagens e vídeos usando areia, podem ser agrupados em dois grandes grupos: trabalhos de renderizações não fotorealísticas para estilização de vídeos e trabalhos que usam areia como material, mas com o foco voltado para o fotorealismo.

No primeiro grupo, renderizações não fotorealísticas para estilização de vídeos, Meier (1996) [13] apresenta o primeiro trabalho relacionado a renderização não fotorealísticas a partir de animações em cenas tridimensionais (3D). Logo em seguida, Litwinowicz (1997) [12] apresenta o primeiro trabalho propondo o uso de renderizações estilizadas para o processamento de vídeos.

Um dos fatores mais importantes na produção de vídeos estilizados, seja a partir da descrição de uma cena artificial 3D ou de um vídeo real, é a exibição de coerência temporal dos elementos de desenho (e.g. pinceladas, curvas, círculos, etc). No caso de cenas 3D, isso é feito movendo-se os elementos de desenho de acordo com a superfície dos objetos que estão sendo desenhadas. Caso isto não seja feito, a animação parecerá como sendo vista através de um vidro texturizado, também chamado de efeito *shower door* [13].

A falta de coerência temporal na renderização estilizada de vídeos reais pode causar oscilações e/ou tremulações chamados de *flickering*. O efeito de *flickering*, também conhecido como *swimming* [7], surge tanto em objetos que estão em movimento e são renderizados com elementos que não os seguem corretamente quanto em áreas estáticas que são renderizadas diferentemente nos quadros adjacentes, devido a ruídos na captura dos vídeos ou algumas diferenças de sombreamento e iluminação.

Várias abordagens tentam impor a coerência temporal na estilização de vídeos. No trabalho de Litwinowicz [12], foi introduzido um método para manter a coerência temporal em pinturas impressionistas. Para isto, foram utilizadas informações de um método de fluxo óptico [8] para rastrear movimentos no vídeo, e desta forma poder movimentar, adicionar ou remover pinceladas de um quadro para outro. Um método para a renderização coerente de áreas estáticas

em quadros sucessivos foi proposta por Hertzmann e Perlin em [11], onde são mantidas as pinceladas das áreas que não apresentam movimentos. A coerência temporal é obtida através do deslocamento de pontos de controles das pinceladas usando informações de um método de fluxo óptico.

No trabalho de Wang et al. [15], os autores propõem um método para criar *cartoons* a partir de vídeo utilizando um algoritmo de segmentação para segmentar o vídeo por completo. Após a segmentação, o usuário especifica pontos em quadros chaves do vídeo através de uma interface gráfica. Esses pontos são então usados para interpolar as bordas das regiões entre os quadros.

No método proposto por Collomosse et al. [7], o vídeo é tratado como um volume 3D $I(x, y, z)$, onde as faixas de valores para as coordenadas x , y e z são a largura, a altura dos quadros do vídeo e o número de quadros do vídeo, respectivamente. Neste método é aplicado um algoritmo de segmentação 2D em cada quadro do vídeo seguido de um algoritmo que, baseando-se em regras heurísticas, associa as regiões segmentadas entre os quadros. O resultado deste método são objetos temporalmente convexos segmentados forçando a coerência temporal na renderização dos mesmos.

Winnemöller [16] fornece um certo nível de coerência temporal a vídeos estilizados como *cartoons*, através de uma quantização suave das cores dos quadros abstraídos. A quantização suave das cores faz com que mudanças de iluminação causadas na captura do vídeo sejam espalhadas pelo vídeo diminuindo a notoriedade dessas mudanças.

Em Gomes et al. [9], é apresentado um método para forçar a coerência temporal intra-objeto em NPR de vídeos utilizando a combinação de um algoritmo de segmentação [6] e um algoritmo de fluxo óptico [1]. O resultado do algoritmo de segmentação é utilizado para restringir a área de busca do algoritmo de fluxo óptico para as bordas do objeto segmentado. A restrição da área na qual o fluxo óptico é calculado possibilita a obtenção de mapas de fluxos mais precisos. Desse modo, a informação de fluxo óptico pode ser utilizada para forçar a coerência temporal nestes vídeos.

No segundo grupo, trabalhos que usam o material areia, mas que objetivam o fotorealismo, Hanrahan e Krueger (1993) [10] descrevem um modelo de reflexão de superfícies que possuem camadas, tais como tecidos biológicos (e.g. pele, folha, etc.) ou materiais inorgânicos (e.g. areia, neve, tinta, etc.), causada pela dispersão da luz através das camadas da superfície. Sumner et al. (1999) [14] introduz um modelo para deformar materiais de terrenos como areia, lama e neve, onde objetos rígidos em 3D se movimentam sobre estes terrenos deixando seus rastros ou pegadas. Zhu e Bridson (2005) [17] descrevem um método baseado em física para animar areia, onde é abstraído os grãos individuais da areia em um objeto contínuo, e através de pequenas mudanças em um simulador de água existente é gerado um simulador de areia. Bell et al. (2005) [2] apresentam

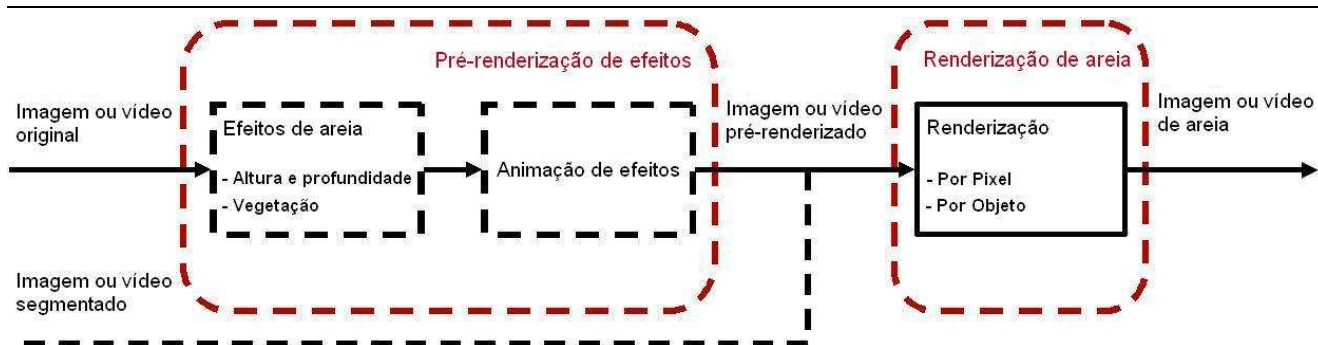


Figura 1. Pipeline de renderização Csand.

um método que simula materiais granular, tais como areia e grãos, onde partículas representam elementos discreto do material simulado e fenômenos altamente dinâmicos como avalanches e esparrames são gerados.

3. A Garrafa com Areia Colorida

Um exemplo típico de garrafa preenchida com areia colorida produzida por um artesão pode ser vista na Figura 3.

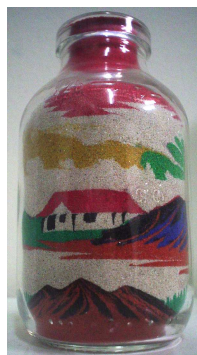


Figura 2. Garrafa com areia colorida.

A técnica para fazer tais peças de arte consiste no preenchimento de uma garrafa de vidro vazia com areia de diferentes cores, movendo-as para formar uma imagem usando uma ferramenta de ferro onde uma das pontas é fina e aguda (Figura 3(a)) e a outra é chata parecida com uma chave de fenda (Figura 3(b)).

4. Renderização Estilizada de Areia

O processo para geração das imagens renderizadas com areia é realizado através de uma interface gráfica, na ordem do *pipeline* visto na Figura 1, que recebe como entrada uma imagem ou um vídeo. Os retângulos tracejados do *pipeline* representam passos opcionais no processo (o

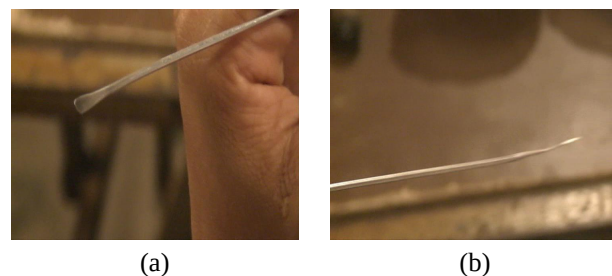


Figura 3. A pontas chata (a) e a ponta aguda (b) da ferramenta usada pelos artesãos.

passo *animação de efeitos* só aparece se algum efeito de areia estiver presente) enquanto que a linha de entrada tracejada representa uma entrada opcional ao *pipeline*.

4.1. Texturas Procedurais de Areia

Neste trabalho, utilizamos uma função de ruído para gerar texturas 2D. Assim a imagem texturizada será colocada dentro de um modelo de garrafa 3D e todas as interações utilizadas para simular os efeitos dos movimentos manuais da ferramenta dos artesãos, chamados aqui de *pinceladas de areia*, serão feitas na interface da areia com a superfície interna da garrafa.

Em Carvalho et al. (2006) [5], introduzimos a técnica desenvolvida para a produção de texturas procedurais de areia. Para gerar uma textura adequada que represente a areia artificial usada na geração dessas garrafas, foram fotografados vários exemplos de areia de várias cores usadas pelos artesãos. Notamos, pela análise dessas imagens, que a variação do matiz é muito menor do que as variações observadas na saturação e no brilho e/ou intensidade. Desta forma, utilizamos o sistema de cores HSV, decompondo as fotografias nos três canais de cores para analisar e determinar qual a melhor distribuição que as represente. Os histogramas dos exemplos de areia coletados, para todos os três

canais, mostrou uma representação que se aproxima de uma distribuição normal, também chamada de distribuição gaussiana.

Deste modo, a função de ruído gaussiano que foi utilizada para gerar perturbações nos valores dos pixels dos três canais do espaço de cores HSV, é dada por

$$P(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \quad (1)$$

onde μ é a média e σ o desvio padrão para as cores primárias de texturas de areia, que são determinados pelos valores das amostras de areia. A partir daí, geramos as texturas primárias de areia colorida para cada 30° no espaço do matiz ($[0^\circ, 360^\circ]$), armazenando-as na memória com o mesmo tamanho das imagens ou vídeos de entrada. A Figura 4 mostra um exemplo de fotografia real de areia em (a) e a textura procedural de areia que simula a fotografia real em (b). Neste exemplo, pode-se observar como a textura sintética consegue simular de forma realística a fotografia real de areia.

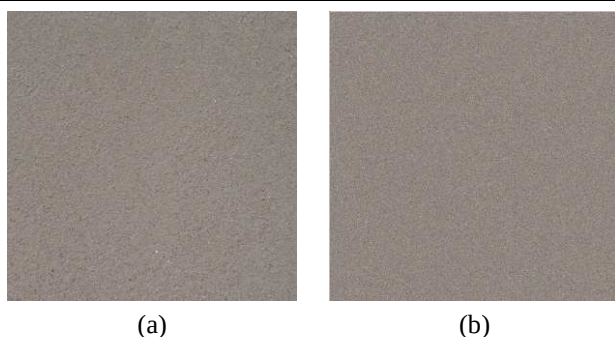


Figura 4. Fotografia real de areia em (a) e textura procedural de areia em (b).

4.2. Renderização da Areia

Quando uma imagem é renderizada, computamos, para cada pixel, o pixel correspondente em uma das texturas primárias de areia previamente armazenadas na memória, e desenhamos o valor do pixel da textura na imagem original na posição correspondente do pixel. Entretanto, há alguns casos especiais que devem ser tratados diferentemente, e para isso foram criados alguns limiares. Por exemplo, se o canal valor (V do modelo de cores HSV) do pixel é igual a 0, a cor é preta. Então criamos um limiar para os pixels que tem o V próximo a 0, aplicando a textura de areia preta nos pixels com valor abaixo do limiar. Criamos também outro limiar para configurar o canal de saturação, pois quando o S é igual a 0 o valor de H é indefinido, logo a cor do pixel está

na escala de tons de cinza, e deve ser representada usando-se uma ou mais das texturas de areia preta, cinza e branca. No caso geral, aplicamos a textura cujo valor de V é mais próximo, ou criamos uma mistura entre duas texturas que definam um intervalo contendo o valor de V.

Se a cor do pixel está localizada entre duas cores primárias no espaço do matiz (H), calcula-se uma mistura entre duas texturas primárias correspondentes as duas cores primárias. Para isso, sorteamos um número aleatório que determinará de qual textura de areia será o valor do pixel retornado. Por exemplo, se a cor do pixel sob análise pode ser decomposta em 30% de laranja e 70% de amarelo, e um número aleatório menor do que 0.7 (no intervalo de $[0,1]$) é sorteado para renderizar o pixel, o pixel terá o seu valor retornado da textura amarela, enquanto que se um valor maior que 0.7 for sorteado o pixel terá o seu valor retornado da textura laranja.

Para simular o processo manual de forma mais realística na criação das misturas, é necessário controlar a proporção entre as areias misturadas, evitando a criação de misturas com quantidades muito desproporcionais. Dessa forma, foram criados dois limiares para o controle de misturas sendo um inferior e outro superior. Esses limiares controlam, dentro de cada espaço entre duas cores primárias do matiz, o tamanho do intervalo onde ocorrerá misturas entre as texturas que representam essas duas cores.

Foram criados dois métodos de renderização, que são: *renderização por pixel* e *renderização por objeto*. No método de renderização por pixel, para cada pixel da imagem de entrada, é analisado o valor dos canais HSV a fim de escolher a textura mais apropriada para representá-lo. Após escolhida a textura, acessamos a posição do pixel correspondente na textura de areia previamente armazenada na memória, e escrevemos o valor do pixel da textura na posição correspondente do pixel na imagem. No método de renderização por objeto, é necessária a segmentação dos objetos do vídeo. Para cada objeto do vídeo são usadas uma ou mais texturas de areia de acordo com a média calculada dos canais HSV dos pixels pertencentes ao objeto.

5. Simulando Efeitos de Pinceladas de Areia

Foram simulados dois efeitos típicos usados por artistas nas garrafas com areia colorida, que introduzimos em Britto Neto e Carvalho (2007) [4], e denominamos como: *efeito de altura e profundidade* e *efeito de vegetação*.

5.1. Efeito de Altura e Profundidade

Na criação do efeito de altura e profundidade, representado na montanha da Figura 6, o artesão traz areia de diferentes cores do centro da garrafa para a superfície interna da

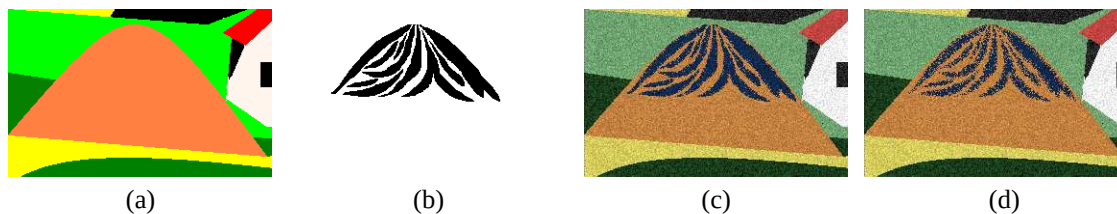


Figura 5. Simulação do efeito de altura e profundidade, onde são exibidos: a imagem original (a), máscara (b), efeito simulado com $\alpha = 20^\circ$ (c) e $\alpha = 40^\circ$ (d).

garrafa, usando a ponta chata da ferramenta, trazendo mais ou menos areia de acordo com o ângulo de aproximação que a ferramenta faz com o eixo vertical. Definimos como 0° quando a ferramenta traz mais areia e 90° quando a ferramenta traz menos areia.



Figura 6. Efeito de altura e profundidade.

O usuário então desenha uma máscara indicando onde os efeitos serão aplicados, seleciona a textura de areia que será arrastada para frente da areia original e o ângulo de aproximação da ferramenta através da interface gráfica desenvolvida. Um mapa de distância da máscara desenhada é calculado e seus valores são armazenados. Este mapa de distância é então usado para determinar se a areia selecionada substituirá a areia atual naquela posição ou não, de acordo com uma distribuição gaussiana com média 0 e desvio padrão dado pela equação

$$\sigma = \begin{cases} \sigma_{min}, & \text{se } 1 - \frac{\alpha}{\alpha_{max}} < t, \\ \sigma_{max} \left(1 - \frac{\alpha}{\alpha_{max}}\right), & \text{caso contrário,} \end{cases} \quad (2)$$

onde α e α_{max} são o ângulo que a ferramenta faz com a superfície interna da garrafa e o ângulo máximo permitido pela ferramenta (90°), respectivamente, t é um limiar configurado para limitar o valor do menor desvio padrão, e σ_{max} e σ_{min} são o máximo e o mínimo valores de desvio padrão permitidos, respectivamente. O valor do σ_{max} para a máscara é configurado como o maior valor do mapa de distância da máscara. Então, baseando-se no valor do mapa de distância de um pixel, a probabilidade da distribuição gaussiana é calculada e um número aleatório é sorteado para determinar se o pixel será ou não substituído por um pixel da

textura de areia selecionada ou se permanecerá com o valor do pixel da textura de areia associada antes da aplicação do efeito. Um exemplo de simulação deste efeito pode ser visto na Figura 5.

5.2. Efeito de Vegetação

Para a simulação do efeito de vegetação, representado pelo objeto verde na Figura 7, o usuário desenha duas linhas horizontais para delinear a variação da altura da vegetação simulada, e especifica a largura mínima e máxima da vegetação que será renderizada. O algoritmo então percorre a imagem verticalmente procurando pela primeira coluna onde ambas as linhas aparecem. A partir daí, ele vai desenhando quadriláteros entre as duas linhas, sendo que a largura do topo e da base são definidos por números aleatórios no intervalo de $[1, min]$ para o topo e no intervalo de $[min, max]$ para a base, onde min é a largura mínima em pixels e max a largura máxima em pixels que a base pode assumir, de acordo com uma distribuição uniforme, até que a última coluna forme um quadrilátero completo. O topo de cada quadrilátero é substituído pelo recorte da linha superior, fazendo com que o topo fique curvo ou retilíneo dependendo da linha desenhada. Com isso, foi gerada uma máscara (Figura 8 (a)) que é invertida para gerar sua máscara complementar (Figura 8 (b)) formada apenas por triângulos.



Figura 7. Efeito de vegetação

Depois das máscaras geradas, é computado um mapa

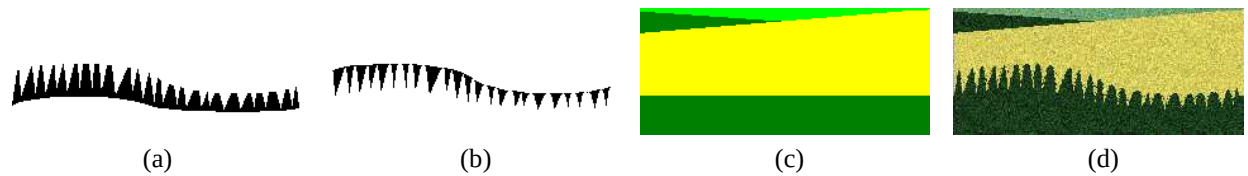


Figura 8. Máscaras original (a) e complementar (b) aplicados na imagem original (c) para gerar o efeito de vegetação visto em (d).

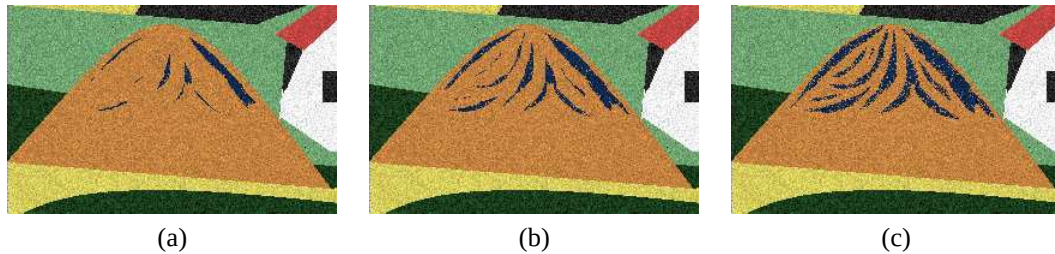


Figura 9. Três quadros da animação de um efeito.

de distância modificado para a máscara complementar, de maneira similar ao usado no efeito de altura e profundidade, mas com uma pequena diferença. Para cada triângulo, depois que o mapa de distância padrão é calculado, calcula-se o centro de cada um, conectando cada centro com os vértices das respectivas bases dos triângulos, criando uma máscara auxiliar. Os valores do mapa de distância para essa máscara auxiliar são substituídos pelo valor máximo do mapa de distância. Isto é feito para limitar o efeito da mistura nas bordas verticais dos triângulos, deixando a área da base dos triângulos intacta. A renderização deste efeito é realizada para ambas as máscaras, sendo que a inferior é renderizada por completo e a superior é renderizada como no efeito de altura e profundidade, permitindo um pouco de mistura somente nas bordas verticais.

5.3. Animando Efeitos de Pinceladas de Areia

Na criação de animações com esses efeitos, foi implementado um mecanismo simples que permite ao usuário incluir animações nos mesmos. O usuário define os quadros inicial e final do vídeo onde a sequência irá aparecer, seleciona uma função (linear ou sigmóide) que irá controlar a aplicação do efeito e um fator de inclinação (que dirá quão rápido o efeito irá aparecer ou desaparecer). Baseado nessa função e no fator de inclinação escolhido, os valores armazenados no mapa de distância (mapa de distância modificado no caso da vegetação, que poderá ser aplicado também na máscara inferior), serão multiplicados por um fator (entre 0 e 1) antes da renderização do efeito ser realizada. A

Figura 9 mostra 3 quadros de uma animação criada usando este método.

6. Animações $2\frac{1}{2}$ D em Caixas de Areia

Para a criação das animações $2\frac{1}{2}$ D, adiciona-se ao vídeo estilizado informações 3D através de um campo de alturas, onde o objeto que está em movimento causa deslocamentos nos grãos de areia de áreas estáticas, deslocando as colunas do campo de alturas verticalmente para que o objeto deixe seu rastro e crie amontoados de areia por onde passa. Na criação dessas animações utilizou-se uma abordagem semelhante a proposta por Sumner (1999) [14].



Figura 10. Um quadro do vídeo de exemplo.

No método desenvolvido neste trabalho são necessárias seis etapas para criação das animações, a saber: segmentação do vídeo, estilização do vídeo com areia colorida, modelagem 3D do objeto de interesse, compressão, deslo-

cimento e erosão. A Figura 10 mostra um quadro de um vídeo real, que será usado para exemplificar nosso método. O vídeo exibe uma caneca movimentando-se da esquerda para a direita, através de um fio de náilon.

A primeira etapa para gerar a animação é a extração do objeto de interesse no vídeo real que vai causar os deslocamentos na areia. Para isto utilizou-se o algoritmo rápido de segmentação *fuzzy* [6] para segmentar o vídeo. A segunda etapa é a fase de estilização do vídeo com areia colorida. Na Figura 11 temos o quadro do vídeo da caneca segmentado em (a) e o quadro do vídeo estilizado com areia colorida em (b), correspondendo a primeira e segunda etapa respectivamente.

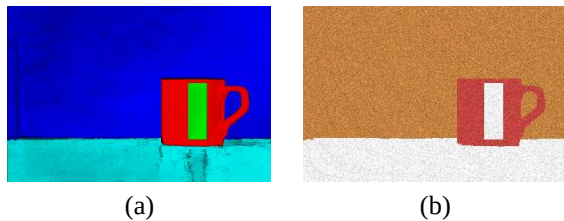


Figura 11. Quadro do vídeo segmentado (a) e quadro do vídeo estilizado com areia colorida (b).

Na terceira etapa, definimos uma forma 3D abstrata (veja Figura 12) para o objeto de interesse, para que o objeto possa deformar a malha da superfície da caixa. Para isto, calculamos, a partir do mapa de segmentação, o mapa de distância do objeto que será usado para definir uma profundidade δ ao objeto. O objeto 3D abstrato é formado por dois campos de alturas, um superior e outro inferior. Os valores das alturas, dos campos de alturas do objeto, são os valores do mapa de distância até a profundidade δ definida. A Figura 12 mostra o objeto 3D extraído do vídeo da caneca, utilizando o método do mapa de distância para dar uma forma 3D abstrata, em algumas visões diferentes.

A próxima etapa é a fase de compressão, onde as colunas que estão sob o objeto de interesse são comprimidas de acordo com a profundidade definida. Nesta fase, o mapa de segmentação nos fornece as informações sobre quais colunas serão comprimidas e o mapa de distância nos fornece as informações sobre o quanto essas colunas serão comprimidas. Em cada quadro do vídeo, as alturas das colunas que estão na região do objeto de interesse e que ainda não sofreram compressão até a profundidade δ , são decrementadas até a superfície da malha ficar abaixo do objeto.

No deslocamento de materiais, o material é distribuído para as colunas ao redor do objeto. Uma taxa de compressão α , que é um valor controlado pelo usuário, é usada para cal-

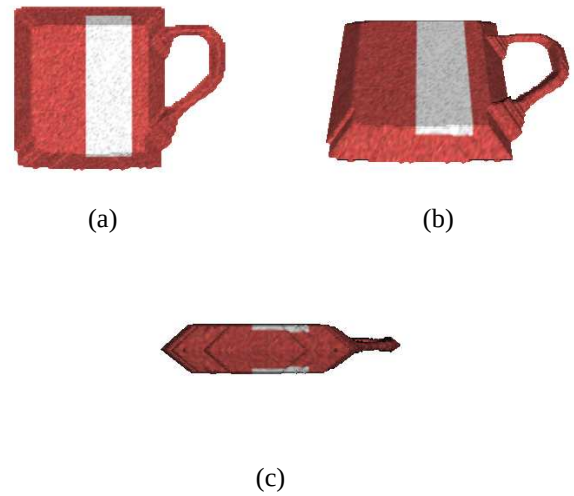


Figura 12. Forma 3D abstrata do objeto. Visão superior (a), visão em 45° (b) e visão de fundo (c).

cular a quantidade de material que será distribuído, Δh , que é calculado por $\Delta h = \alpha m$, onde m é a quantidade total de material que foi deslocado. O material não comprimido é distribuído igualmente entre seus vizinhos com menor valor de distância (de acordo com o mapa de distâncias) até que todo o material fique depositado nas colunas ao redor do objeto, formando um anel de colunas.

Na etapa de erosão, as colunas do anel são erodidas. O algoritmo verifica a inclinação entre cada par de colunas adjacentes na malha. Para cada coluna ij e um vizinho kl , a inclinação s , é dada por

$$s = \tan^{-1}(h_{ij} - h_{kl})/d \quad (3)$$

onde h_{ij} é a altura da coluna ij e d é a distância entre as duas colunas. Se a altura entre as duas colunas vizinhas é maior que um limiar θ_{out} , então o material é movido da maior coluna para a coluna menor. Em um caso especial onde uma das colunas está em contato com o objeto, um limiar diferente, θ_{in} , é usado para fornecer um controle independente da inclinação interna. O material é movido usando o cálculo da diferença média das alturas, Δh_a , para as n colunas vizinhas com grande inclinação, que é dada por:

$$\Delta h_a = \frac{\sum(h_{ij} - h_{kl})}{n} \quad (4)$$

A diferença média das alturas é multiplicada por uma constante fracionária, σ , e a quantidade resultante é igualmente distribuída entre os vizinhos. O algoritmo repete este passo até todas as inclinações ficarem abaixo de um limiar, θ_{stop} . A Figura 13 mostra as quatro etapas finais do processo para geração da animação 2 $\frac{1}{2}$ D.

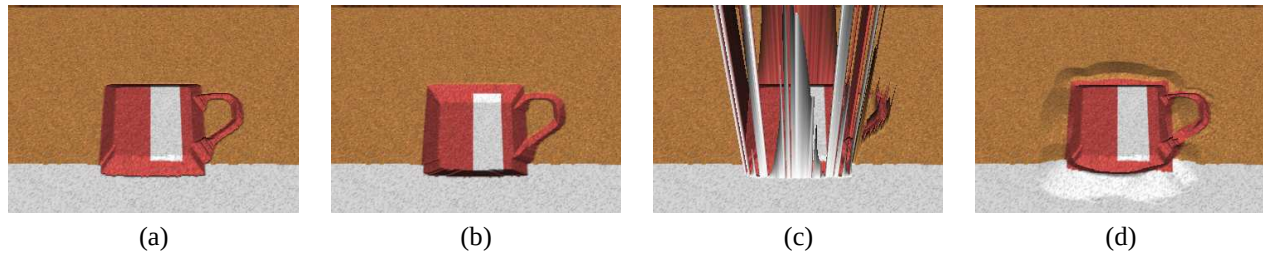


Figura 13. Modelagem 3D (a), compressão (b), deslocamento (c) e erosão (d).

7. Experimentos

Agora descrevemos alguns experimentos realizados para mostrar o funcionamento do método proposto. A Figura 14 exibe exemplos dos métodos de renderização por pixel e por objeto utilizados na estilização do vídeo de um sapo de pelúcia que se movimenta da esquerda para direita da tela. A renderização por pixel é mais proveitosa quando deseja-se manter alguns detalhes da imagem. Neste exemplo, pode-se observar como a renderização por pixel mantém alguns pequenos detalhes dentro do corpo do sapo, tais como olhos, nariz, e parte da boca. Por outro lado, quando a intenção é limitar as cores usadas para renderizar um objeto, utiliza-se o método de renderização por objeto.

A Figura 14(b) mostra o mapa de segmentação do quadro original da imagem exibida na Figura 14(a), onde o matiz indica a qual objeto pertence o conjunto de pixels, e a intensidade (mapeada em cada objeto, no intervalo de $[0,1]$) reflete o grau de pertinência que o pixel tem em relação ao objeto. A Figura 14(c) mostra uma renderização por pixel do quadro, enquanto que a 14(d) foi gerada usando o mapa de segmentação mostrado em 14(b) para restringir as texturas usadas para renderizar os pixels pertencentes ao objeto azul (barriga e joelhos) para a textura verde. Para inserir detalhes na renderização por objeto, é necessário que objetos extras sejam adicionados no passo de segmentação. Por outro lado, a renderização por objetos pode ser usada para manter a aparência de todas as estruturas, forçando um certo nível de coerência temporal.

A Figura 15 mostra oito quadros de quatro animações diferentes, sendo que duas delas dentro da garrafa, onde as imagens que compõem essas animações lembram o tipo de cena geralmente produzida pelos artesãos, e as outras duas mostram o resultado do método para geração de animações $2\frac{1}{2}$ D em caixa de areia. Em (c) e (d) temos a estilização de um vídeo real, onde uma caneca é puxada através de um fio de náilon, e em (h) e (g) temos a estilização de um vídeo sintético, onde uma joaninha anda pela areia deixando seu rastro e escrevendo a palavra *aloha*.

8. Conclusão

Neste trabalho propomos um método de renderização não fotorealística, que simula uma expressão artística típica da região Nordeste do Brasil, usando areia colorida para compor imagens de paisagens na superfície interna de garrafas de vidro. Uma técnica para geração de texturas procedurais de areia 2D foi descrita, e duas técnicas que imitam efeitos criados pelos artesãos na produção de garrafas foram apresentadas. Propomos também um método para geração de animações $2\frac{1}{2}$ D em caixas de areia.

As técnicas descritas aqui são usadas para estilização de vídeos, algo quase impossível de ser produzido por um artesão na vida real, tornando possível a criação de animações usando garrafas virtuais com areia colorida. Empregamos o algoritmo rápido de segmentação *fuzzy* [6] para segmentar os vídeos como volumes 3D, e assim, permitir que o algoritmo de segmentação trate oclusão de objetos e objetos não temporalmente convexos. A coerência temporal nos vídeos estilizados são forçadas em objetos individuais através dos resultados da segmentação, utilizados na restrição das texturas de areia usadas em alguns objetos.

Referências

- [1] S. S. Beauchemin and J. L. Barron. The computation of optical flow. *ACM Comput. Surv.*, 27(3):433–466, 1995.
- [2] N. Bell, Y. Yu, and P. J. Mucha. Particle-based simulation of granular materials. In *SCA '05: Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 77–86, New York, NY, USA, 2005. ACM Press.
- [3] L. S. Britto Neto. Renderizações não fotorealísticas para estilização de imagens e vídeos usando areia colorida. Dissertação de Mestrado, Departamento de Informática e Matemática Aplicada, Universidade Federal do Rio Grande do Norte, Setembro 2007.
- [4] L. S. Britto Neto and B. M. Carvalho. Message in a bottle: Stylized rendering of sand movies. In *SIBGRAP '07: Proceedings of the 20th Brazilian Symposium on Computer Graphics and Image Processing*, pages 11–18, Los Alamitos, CA, USA, 2007. IEEE CS Press.

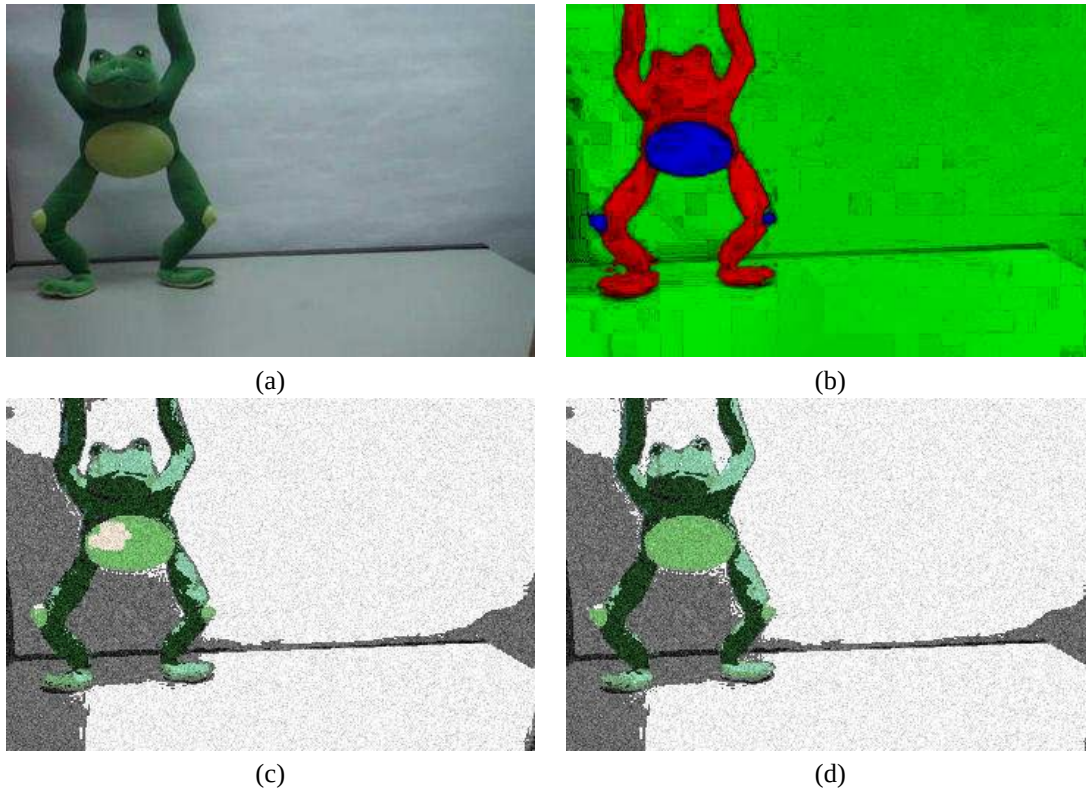


Figura 14. Um quadro do vídeo do sapo (a), o mapa de segmentação associado (b), a renderização por pixel do quadro (c), e o quadro onde o objeto azul de (b) foi renderizado usando o método de renderização por objeto (d).

- [5] B. M. Carvalho, L. S. Britto Neto, and L. M. Oliveira. Bottled sand movies. In *CGIV '06: Proceedings of the International Conference on Computer Graphics, Imaging and Visualization*, pages 402–407, Sidney, Australia, July 2006. IEEE CS Press.
- [6] B. M. Carvalho, L. M. Oliveira, and G. S. Silva. Fuzzy segmentation of color video shots. In *Proceedings of the Digital Geometry for Computer Imagery*, volume 4245, pages 402–407, London, 2006. Springer-Verlag.
- [7] J. P. Collomosse, D. Rowntree, and P. M. Hall. Stroke surfaces: Temporally coherent artistic animations from video. *IEEE Transactions on Visualization and Computer Graphics*, 11:540–549, 2005.
- [8] E. R. Davies. *Machine Vision: Theory, Algorithms, Practicalities*, 2nd Ed. Academic Press, London, 1996.
- [9] R. Gomes, T. S. Santos, e B. M. Carvalho. Coerência temporal intra-objeto para NPR utilizando fluxo óptico restrito. Em *Revista Eletrônica de Iniciação Científica*, número II, junho 2007.
- [10] P. Hanrahan and W. Krueger. Reflection from layered surfaces due to subsurface scattering. In *SIGGRAPH '93: Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques*, pages 165–174, New York, NY, USA, 1993. ACM Press.
- [11] A. Hertzmann and K. Perlin. Painterly rendering for video and interaction. In *Proceedings of the International Symposium on Non-Photorealistic Animation and Rendering*, pages 7–12, 2000.
- [12] P. Litwinowicz. Processing images and video for an impressionist effect. In *SIGGRAPH '97: Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, pages 407–414, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co.
- [13] B. Meier. Painterly rendering for animation. In *Proceedings of the ACM SIGGRAPH*, pages 477–484, 1996.
- [14] R. Sumner, J. F. O'Brien, and J. K. Hodgins. Animating sand, mud, and snow. *Computer Graphics Forum*, 18(1):17–26, March 1999.
- [15] J. Wang, Y. Xu, H. Y. Shum, and M. F. Cohen. Video tooning. *ACM Transactions on Graphics*, 23:574–583, 2004.
- [16] H. Winnemöller, S. C. Olsen, and B. Gooch. Real-time video abstraction. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Papers*, pages 1221–1226, New York, NY, USA, 2006. ACM Press.
- [17] Y. Zhu and R. Bridson. Animating sand as a fluid. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Papers*, pages 965–972, New York, NY, USA, 2005. ACM Press.



Figura 15. Resultados da aplicação das técnicas de renderização de areia na garrafa e do método para geração de animações $2\frac{1}{2}D$ em caixas de areia.

Sistema Composto para Amostragem e Geração de Luzes a partir de Mapas de Iluminação

Aldo R. Zang
Laboratório VISGRAF - IMPA
zang@impa.br

Luiz Velho
Laboratório VISGRAF - IMPA
lvelho@impa.br

Abstract

In this paper¹ we introduce a new approach to the problem of direct illumination in physically-based rendering of 3D scenes using *illumination maps* captured from real environments. We developed a system that takes advantage of the best features of the current solutions to the problem: namely, the approximation of illumination maps through directional lights; and stochastic sampling of the light maps. Our framework is flexible and can be used with most rendering programs.

Keywords: Three-dimensional Graphics and Realism, Sampling, Rendering, Environment Mapping, Importance Sampling, Deterministic Sampling.

Resumo

Neste artigo¹ apresentamos uma nova abordagem para o problema de iluminação direta na renderização fotorealista com mapas de iluminação provenientes de cenas reais. Desenvolvemos um sistema que extrai o melhor das duas principais abordagens existentes para o problema: aproximação dos mapas de iluminação com luzes direcionais, e amostragem estocástica dos mapas.

1. Introdução

A renderização de cenas complexas, com iluminação indireta, iluminação ambiente, e outros tipos de fontes de luz representa um problema desafiador e de grande interesse da comunidade de computação gráfica. Atualmente a renderização fotorealista de cenas sintéticas com iluminação proveniente de mapas de iluminação extraídos de ambientes reais é um ponto muito estudado no

processo de síntese de imagem, e também com grande demanda de soluções eficientes, uma vez que técnicas de incorporação de objetos sintéticos em filmes e comerciais são muito procurados. É verdade que um resultado fotorealista depende de vários fatores além da iluminação, como a modelagem dos objetos e propriedades dos materiais, mas tudo isto não pode ser apreciado se não tivermos um método de visualização efetivo e uma iluminação realista.

Para renderizar objetos sintéticos em cenas reais é preciso dispor da iluminação do ambiente real onde será feita a inserção, que pode ser capturada como uma imagem esférica de alta variação dinâmica (HDR - High Dynamic Range). Esta técnica de captura de iluminação foi introduzida por Paul Debevec [1], está presente em muitos programas comerciais e foi utilizada em vários filmes.

1.1. O problema de iluminação direta

O problema central do rendering é calcular a equação

$$L_o(p, \omega_o) = \int_{S^2} f(p, \omega_o, \omega_i) L_i(p, \omega_i) |\cos \theta_i| d\omega_i$$

da radiância L_o que chega ao ponto p na direção ω_o ; onde $L_i(p, \omega_i)$ é a radiância incidente em p na direção ω_i , $f(p, \omega_o, \omega_i)$ é a BSDF (função de reflectância bidirecional), e θ_i é o ângulo de incidência [4].

No contexto de iluminação direta estamos interessados na radiância proveniente diretamente das fontes luminosas (denotadas $L_d(p, \omega)$), portanto a integral anterior pode ser escrita como

$$\int_{S^2} f(p, \omega_o, \omega_i) L_d(p, \omega_i) |\cos \theta_i| d\omega_i$$

A mesma pode ser quebrada num somatório sobre as fontes de luz da cena

$$\sum_{j=1}^{luzes} \int_{S^2} f(p, \omega_o, \omega_i) L_{d(j)}(p, \omega_i) |\cos \theta_i| d\omega_i,$$

¹ Resumo estendido de dissertação de mestrado.

onde $L_{d(j)}$ denota a radiância incidente em p proveniente da j -ésima fonte de luz e

$$L_d(p, \omega_i) = \sum_j L_{d(j)}(p, \omega_i).$$

Como num ambiente real temos iluminação proveniente de todas as direções, considerando um mapa de ambiente tradicional de 1024×512 pixels, teríamos 1024×512 somas, o que torna a solução computacionalmente inviável. Para reduzir este somatório, entram em cena os métodos de integração probabilísticos com estimadores de Monte Carlo. Então escolhem-se N direções ω_i com algum método probabilístico, obtendo a estimativa

$$\frac{1}{N} \sum_{j=1}^N \frac{f(p, \omega_o, \omega_i) L_d(p, \omega_j) |\cos \theta_j|}{p(\omega_j)}.$$

Neste artigo, propomos uma abordagem que pretende extrair as vantagens dos métodos de integração de Monte Carlo, introduzindo a utilização de pré-amostragem e geração de luzes direcionais. A maioria dos softwares de renderização não permite muita escolha de modelagem da iluminação real. Em alguns softwares pode-se trabalhar com o próprio mapa, o qual é utilizado pelas rotinas de amostragem de Monte Carlo, como no caso do PBRT [4], ou em outros casos é preciso realizar algum pré-processamento e transformar o mapa numa coleção de luzes pontuais ou direcionais, como é feito no Maya.

Nosso objetivo é apresentar um sistema de pré-processamento multifuncional que permita trabalhar com Mapa de Iluminação de variadas formas: obtendo uma coleção de luzes direcionais mediante métodos de amostragem como Median Cut de Debevec [2]; ou amostragem por importância baseado em Penrose Tiling de Ostromoukhov et al. [3]; ou selecionando um número grande de amostras para ser passadas às rotinas de Monte Carlo de renderizadores como PBRT e POV-Ray entre outros. Também trataremos os Mapas de Iluminação de maneira intermediária, estratificando-os, selecionando regiões das quais extrairemos luzes direcionais e outras que manteremos para passá-las às rotinas de Monte Carlo. Um sistema desta índole é muito útil, pois permite ter um controle fino dos resultados nas renderizações. Permite adaptar um Mapa de Iluminação para a maioria dos softwares de renderização, ou pela utilização do mapa como um todo, ou por partes (luzes e um conjunto de amostras).

2. Mapa de Iluminação

Existem diversos formatos para o Mapa de Iluminação, entretanto adotaremos o formato retangular latitude-

longitude. Dado um mapa de dimensões $W \times H$ neste formato, um ponto (x, y) pode ser identificado com uma direção $\vec{\omega}_i = (\phi, \theta)$ (coordenadas esféricas) tomando

$$(\phi, \theta) = \left(\frac{2x\pi}{W}, \frac{y\pi}{H} \right).$$

A direção $\vec{\omega}_i$ também pode ser representada por coordenadas retangulares (x, y, z) em função das coordenadas esféricas (ϕ, θ) com

$$x = \sin \theta \cdot \cos \phi, \quad y = \sin \theta \cdot \sin \phi, \quad z = \cos \theta. \quad (1)$$

2.1. Representação do Mapa de Iluminação

Em lugar de entrar diretamente com a imagem HDR do mapa de iluminação, calculamos e incluímos a deformação de área induzida pela transformação F que mapeia o mapa latitude-longitude na esfera de direções. Os pixels das primeiras e últimas linhas da imagem do mapa correspondem a uma minúscula região circular com centro no pólo norte e sul da esfera respectivamente. Para fazer a correção, aplicamos ao mapa a função

$$F(x, y) = L(x, y) \sin \left(\frac{y}{H} \pi \right) = L(x, y) \sin(\theta)$$

onde, (x, y) é a posição no mapa latitude-longitude; $L(x, y)$ é o valor da luminância no ponto (x, y) ; e $\theta = \frac{y}{H} \pi$ é o ângulo subtendido entre o pólo norte e a direção associada ao ponto (x, y) na esfera unitária (i.e $\theta \in [0, \pi]$). Desta forma temos a seguinte relação entre o mapa original e o corrigido

$$M_c(x, y) = \sin \left(\frac{y}{H} \pi \right) M_o(x, y).$$

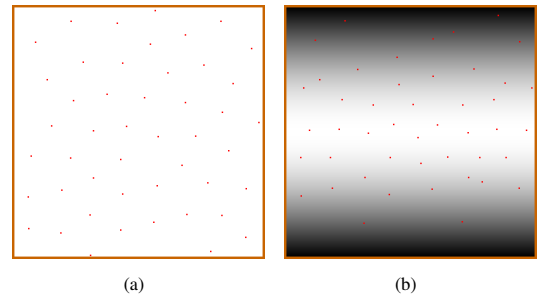


Figura 1. Amostragem com Penrose Tiling [3]: 1(a) mapa homogêneo, 1(b) mapa homogêneo com correção de área.

Na Figura 1 vemos o efeito de amostrar um mapa isoluminante e sua transformação. Como resultado, as posições

das amostras mudam quando considerarmos a deformação induzida pelo mapeamento esférico.

Com esta estratégia podemos amostrar várias superfícies curvas desde que se tenha uma parametrização da superfície no plano que permita proceder com a amostragem.

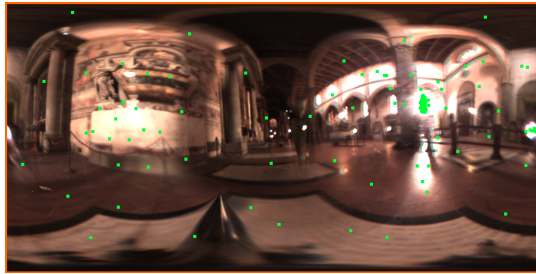
Feita esta correção do mapa, aplicamos a técnica de amostragem por Penrose Tiling de [3]. Uma vez que dispomos das amostras adaptadas, podemos trabalhar com elas.

2.2. Geração de luzes direcionais

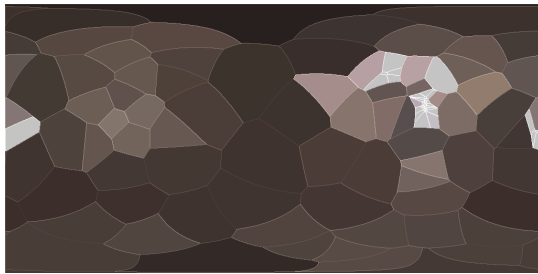
Com a localização das amostras no mapa de iluminação, calculamos sua posição (x, y, z) na esfera unitária com a equação (1). Em seguida calculamos as células de Voronoi sobre a esfera, e finalmente somamos a radiância em cada célula e dividimos pela sua área para obter a radiância média que representará cada luz direcional. Utilizamos o conjunto de luzes para renderizar a cena.

2.3. Geração de mapa de amostras

Uma alternativa à geração de luzes direcionais e à amostragem direta do mapa por métodos de Monte Carlo é realizar uma pré-amostragem do mapa e passar estas amostras ao renderizador para utilizá-las na renderização.



(a)



(b)

Figura 2. Mapa HDR Galileo [1], 124 amostras com Penrose Tiling [3]: 2(a) amostragem do mapa com prévia correção da função de importância (luminância), 2(b) células de Voronoi da amostragem 2(a).

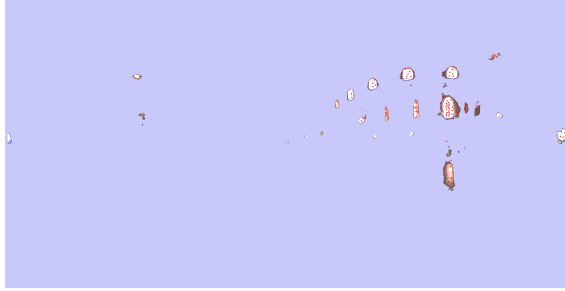
Desenvolvemos um plugin (*FHIS*) para o PBRT [4] que utiliza um mapa de pré-amostras para realizar a renderização. Na criação do mapa de pré-amostras guardamos: a posição das amostras no mapa (Figura 2(a)); uma imagem latitude-longitude da divisão do mapa em células de Voronoi sobre as amostras, calculada sobre a esfera (Figura 2(b)); e uma lista da radiância total de cada célula de Voronoi. Esta aproximação do mapa de iluminação original por uma segmentação de Voronoi diminui entre 20 e 25 por cento o tempo de renderização da cena, além de contribuir para uma pequena melhora na qualidade visual da imagem final.

3. Método composto

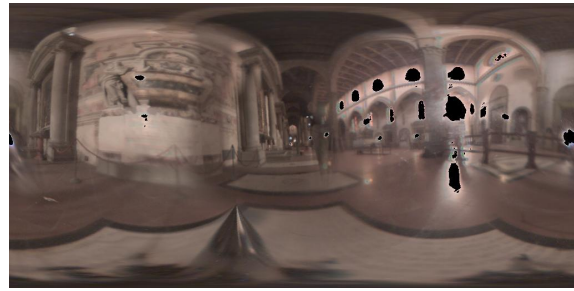
Gostaríamos de aproveitar os benefícios de ambas as abordagens vistas anteriormente, i.e. obter diminuição da variância na cena utilizando luzes direcionais importantes e suavizar as bordas das sombras amostrando com métodos de Monte Carlo nas regiões do mapa onde a iluminação é mais fraca. Na prática existe uma infinidade de combinações a serem testadas. Até o presente momento realizamos os primeiros testes nesta nova abordagem e obtivemos resultados relevantes. Descreveremos a continuação sobre o pré-processamento realizado:

Primeiramente, estratificamos o mapa de iluminação utilizando a função luminância para definir os estratos, criando duas regiões A e B . Por exemplo, podemos tomar n estratos $E_1, E_2, \dots, E_n$, então a região $B = \{E_1, \dots, E_{n/2}\}$, formada pelos $n/2$ primeiros estratos, delimitará a região do mapa que acumula 50% da luminância, somando a luminância em ordem crescente; logo a região $A = \{E_{n/2+1}, \dots, E_n\}$, formada pelos $n/2$ estratos restantes, acumula 50% da luminância proveniente das zonas mais brilhantes do mapa. Em geral, quando dispomos de um mapa de iluminação proveniente de um ambiente real, a área da região A representa menos que o 5% da área total do mapa (por exemplo, no caso do Mapa Galileo a área da região A é aproximadamente 1% da área do mapa, Figura 3(a)). Esta acumulação de luminância numa pequena área pode ser aproveitada tirando amostras da região (A) e convertendo-as num conjunto Ω de luzes direcionais, Figura 3(a). Como a área é pequena, podemos representar bem a região com poucas amostras, i.e poucas luzes direcionais. A região complementar (B), cuja área representa a maior parte do mapa de iluminação, pode ser tratada de duas formas:

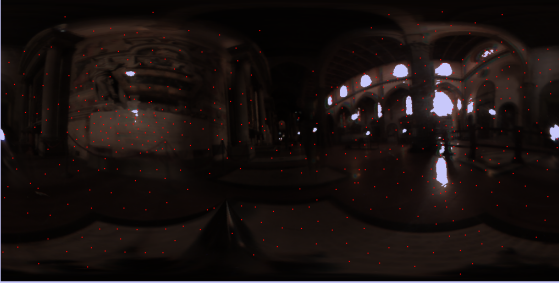
- Guardar a região B como um mapa HDR M_B , onde os pixels da região A têm radiância nula (Figura 3.b). Neste caso a renderização da cena será feita utilizando o plugin *InfiniteSample (IS)* do PBRT [4] com o mapa M_B , e a renderização do conjunto Ω de luzes direcionais obtidas a partir da região A .



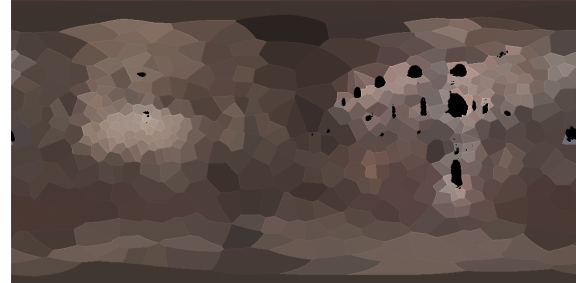
(a) amostragem sobre estrato A com [3], acumula 50% da radiância.



(b) estrato $B = Mapa - A$, 50% da radiância do mapa.



(c) amostragem sobre o estrato B com [3].



(d) células de Voronoi da amostragem do estrato B .

Figura 3. Estratificação do Mapa HDR Galileo [1] e amostragem pelo método de Penrose Tiling [3]: 3(a) 64 amostras transformadas em luzes direcionais, 3(c) 493 amostras, 3(d) células de Voronoi esféricas das amostras de 3(c).

- Amostrar com Penrose Tiling [3], calcular as células de Voronoi sobre a esfera e guardar um mapa HDR de Voronoi M_{BV} . Aqui a renderização será feita com o plugin *FHIS*, criado para o PBRT [4], para renderizar a iluminação dada pelo mapa M_{BV} , juntamente com renderização do conjunto Ω de luzes direcionais.

Nos testes apresentados neste artigo trabalhamos com limiares de 50% de luminância para determinar cada uma das regiões A e B . Estes limiares podem variar, se $X\%$ é o limiar utilizado para determinar A o limiar que determina B é $(100 - X)\%$. É evidente que a medida que aumentamos o limiar X , temos um conseqüente aumento da área de A . Em mapas provenientes de cenas reais a relação entre o limiar e a respectiva área é de tipo exponencial. O objetivo é tomar um limiar X alto, mas com a área da região A associada relativamente pequena, pois uma vez que a região A delimite uma área substancialmente grande do mapa, não é conveniente extrair luzes direcionais, uma vez que vamos precisar de uma quantidade grande de luzes para obter suavidade entre as sombras. Visamos continuar trabalhando neste tópico de escolha do limiar, para desenvolver um método de automação do cálculo do limiar ótimo para cada mapa.

Juntamente com a estratificação, também tivemos cuidado de introduzir controles adicionais, dado que o método Penrose Tiling de Ostromoukhov [3] apresenta alguns inconvenientes. Como o método de Ostromoukhov [3] apli-

cado a um mapa de iluminação faz amostragem hierárquica por importância de uma região retangular contínua do plano com as dimensões do mapa, e função de importância constante por partes igual à luminância do mapa, tem-se frequentemente uma alta condensação de amostras em regiões de área muito pequena. Por exemplo, muitas vezes temos duas ou mais amostras numa área que corresponde a um pixel do mapa. Este fenômeno acontece porque nos mapas extraídos de ambientes reais a variação de luminância é muito grande. Como o objetivo é obter luzes direcionais que iluminem a cena, precisamos de amostras devidamente espalhadas para cobrir bem a região A (por este motivo não temos interesse em amostras cujas proximidades é inferior a 1 pixel). Por fim, agrupamos estas amostras, aplicando um método de clusterização sugerido por Velho et al. [5], eliminando desta forma as amostras redundantes. Implementamos em nosso sistema a clusterização com três opções para a frequência F definidas abaixo.

Dado um cluster com dois elementos $K = \{c_i, c_j\}$, o nível de quantização ótima utilizando a métrica do quadrado da distância geodésica sobre a esfera é:

$$c = \frac{F_i}{F_i + F_j} c_i + \frac{F_j}{F_i + F_j} c_j,$$

$$E(c_i c_j) = \frac{F_i F_j^2 + F_j F_i^2}{(F_i + F_j)^2} ||\cos(c_i \cdot c_j)||^2,$$

onde as opções para F_i são

- $F_i = 1$,
- $F_i = \int_{Vor(c_i)} L(x)dx$,
- $F_i = \frac{\int_{Vor(c_i)} L(x)dx}{Área\ Vor(c_i)}$,

4. Resultados

Desenvolvemos o plugin FHIS (Fast Hierarchical Importance Sampling) para o PBRT [4] que realiza amostragem por importância sobre uma pré-amostragem do mapa de iluminação pelo método de Ostromoukhov [3]. Realizamos comparações numéricas com vários métodos: aproximação do mapa de iluminação por luzes direcionais (D.L.) obtidas com algoritmos de Median Cut (MC) [2] e Ostromoukhov (OS) [3], amostragem direta por importância (Plugin *IS*, PBRT [4]).

Imagem	Método	Amostras por pixel	Erro RMS	Tempo
Fig.6(a)	IS	64 (4x16)	0.112324	90.0s
	IS	128 (4x32)	0.070817	173.1s
	IS	256 (4x64)	0.049604	339.5s
Fig.6(e)	FHIS	64 (4x16)	0.104842	73.2s
	FHIS	128 (4x32)	0.062812	141.3s
	FHIS	256 (4x64)	0.043516	275.3s
Fig.6(i)	FHIS + D.L.(OS)	128 (4x16 + 64)	0.041018	157.6s
Fig.6(m)	FHIS + D.L.(OS)	192 (4x32 + 64)	0.032888	224.0s
Fig.5(b)	FHIS + D.L.(OS)	320 (4x64 + 64)	0.028287	357.8s
	FHIS + D.L.(OS)	128 (4x16 + 64)	0.040586	156.6s
	FHIS + D.L.(OS)	192 (4x32 + 64)	0.032419	225.0s
	D.L.(OS) [3]	128 (4x + 128)	0.088624	173.6s
	D.L.(OS) [3]	256 (4x + 256)	0.036659	337.6s
	D.L.(MC) [2]	256 (4x + 256)	0.049800	334.2s

Tabela 1. Comparação entre renderizações. A imagem de referência foi renderizada com *IS* no PBRT [4], com 4096 (8x512) amostras (Figura 5(a)). (AMD64 3000, 2Gb RAM).

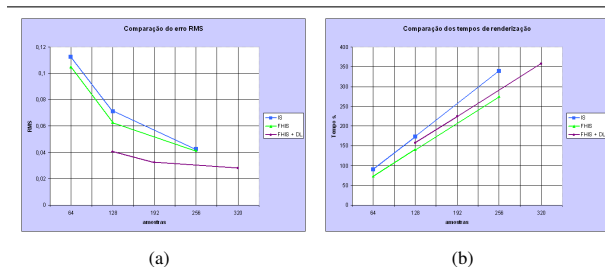
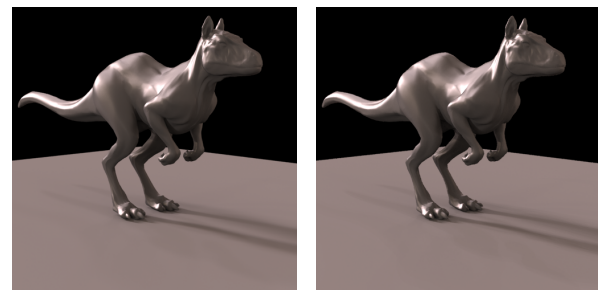


Figura 4. Comparações: 4(a) Erro RMS, 4(b) Tempo de renderização.

Na Tabela 1 incluímos os resultados dos testes mais relevantes. Na Figura 4 pode-se apreciar a comparação dos erros RMS (Figura 4(a)) e dos tempos de renderização (Figura 4(b)). Na Figura 5 mostramos a imagem de referência e o resultado obtido utilizando o método misto, e na Figura 6 mostramos as renderizações de uma cena 3D com os parâmetros da tabela 1 e detalhes ampliados de uma região das mesmas, onde pode-se ver a diferença de qualidade na reconstrução para valores similares de amostragem.

5. Conclusões

Estamos propondo uma abordagem alternativa para o tratamento de problemas de renderização fotorealista baseados em iluminação real que oferece ao usuário liberdade de escolha dos parâmetros para ter um controle mais fino do resultado da renderização. O sistema apresentado neste trabalho utiliza o método de amostragem hierárquica por importância, desenvolvido por Ostromoukhov et al. [3]. Porém, poderíamos utilizar outros métodos de amostragem, como por exemplo Median Cut [2]; aspecto que pretendemos continuar pesquisando.



(a) Imagem de referência. 4096 amostras, 256 amostras por raio, 4 raios por pixel para anti-aliasing. (b) *FHIS* 64 amostras por raio + 64 D.L.(OS) (luzes direcionais). 4 raios por pixel para anti-aliasing.

Figura 5. Renderização do modelo com o Mapa Galileo.

Referências

- [1] P. DEBEVEC. Rendering synthetic objects into real scenes: bridging traditional and image-based graphics with global illumination and high dynamic range photography. *SIGGRAPH*, 1998.
- [2] P. DEBEVEC. A median cut algorithm for light probe sampling. *SIGGRAPH Poster*, 2005.
- [3] J. P. OSTROMOUKHOV V., DONOHUE C. Fast hierarchical importance sampling with blue noise properties. *ACM Trans. Graph.*, 23(3):488–495, 2004.
- [4] H. G. PHARR M. *Physically Based Rendering: From theory to Implementation*. Morgan Kaufmann, 2004.
- [5] S. M. V. R. VELHO L., GOMES J. Color image quantization by pairwise clustering. *SIBGRAPI*, 1998.

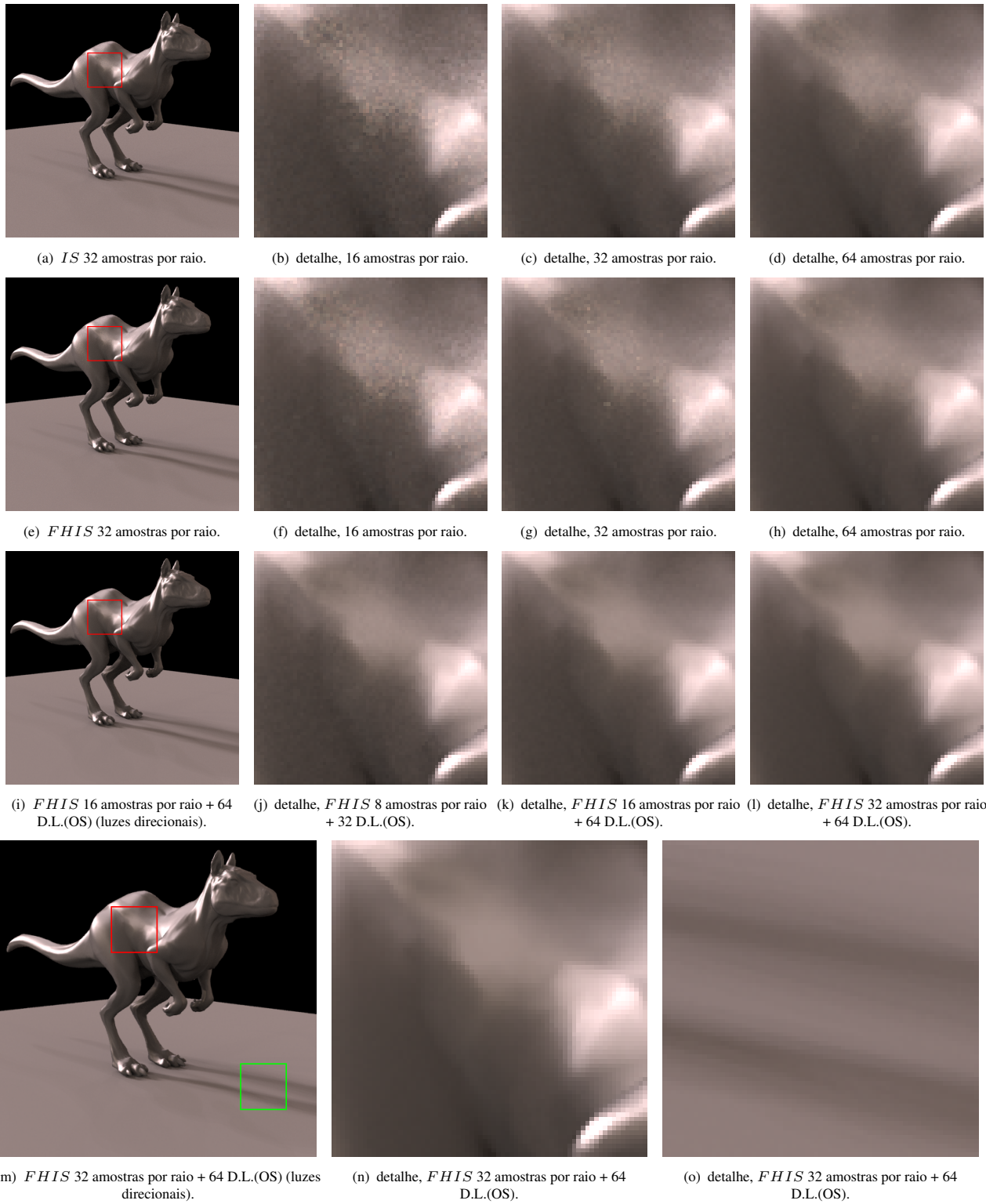


Figura 6. Renderização do modelo killeroo com o Mapa Galileo. Foram amostrados 4 raios por pixel para o anti-aliasing.

Um sistema simplificado para animação física usando malhas tetraedrais

Guina Sotomayor Alzamora Claudio Esperança
University Federal of Rio of Janeiro
Computer Graphics Laboratory
Rio de Janeiro - Brasil
gsa,esperanc@lcg.ufrj.br

Abstract

We present a simplified approach for animation of geometrically complex deformable objects represented as tetrahedral meshes. Our prototype system detects and responds to collisions of objects subject to elastic deformations of variable stiffness. The proposed approach combines several techniques, namely, collision detection using a Spatial Hashing, collision response through a contact surface that use a consistent penetration depth using propagation, an estimate for displacement vector of the deformation region and binary search to separate objects. The dynamics is based on shape matching and a modal analysis scheme, using an Euler explicit-implicit integrator. Preliminary results show that collisions between objects containing several hundreds tetrahedra can be animated in real-time.

Resumo

Apresenta-se uma abordagem simplificada para animação de objetos deformáveis geometricamente complexos, representados como malhas tetraedrais. O sistema detecta e responde a colisões de objetos sujeitos a deformações elásticas de rigidez variável. A abordagem combina várias técnicas, como a detecção de colisões usando Hashing espacial, resposta às colisões através do cômputo da superfície de contato, que usa o cálculo da profundidade de penetração por propagação, a estimativa dos vetores de deslocamento dos vértices da região de deformação e busca binária para separar os objetos. A dinâmica está baseada no casamento de formas e na análise modal, na integração do sistema é usado um esquema de Euler explícito-implícito. Resultados preliminares mostram a interação entre objetos constituídos de várias centenas de tetraedros em tempo real.

1. Introdução

O que se entende pelo termo “animação baseada em física” (*physically based animation*, em inglês) é um processo computacional que visa obter animação de objetos com plausibilidade física. Isto contrasta com o termo “animação física”, empregado usualmente para designar animações que visam replicar processos físicos com alto grau de acurácia, embora por vezes os dois termos sejam usados indistintamente. Este trabalho insere-se mais no contexto do primeiro termo, uma vez que trata de abordagens onde a preocupação com desempenho leva a um tratamento simplificado de determinadas interações entre objetos.

Embora os modelos baseados em física não pretendam reproduzir a realidade, eles tentam produzir movimentos com base nos mesmos princípios físicos. A parte da física que estuda o movimento dos corpos é a dinâmica, que é baseada nas *Leis de Movimento de Newton*.

A animação de objetos rígidos e deformáveis frequentemente se baseia em sistemas de partículas, sendo que a animação de objetos deformáveis deve adotar algum modelo de deformação físico, que permita deformações elásticas ou plásticas. Tais deformações podem ser obtidas usando métodos baseados em malhas como os sistemas massa-mola e elementos finitos ou métodos sem malhas [14, 13, 16]. Mas, objetos deformáveis têm normalmente representação complexa quando comparados com corpos rígidos, já que podem mudar de forma no tempo pela interação com ele mesmo ou com outros agentes no cenário, pesquisas recentes nessa área vêm adaptando diversas técnicas para animação de corpos rígidos.

Este trabalho enfoca o uso de uma metodologia simples para animação de objetos deformáveis, geometricamente complexos, que são representados por malhas tetraedrais. O objetivo é obter uma simulação dinâmica estável, com comportamento físico plausível. Ressalve-se, entretanto, que as propriedades do material não foram modeladas neste

protótipo. O método apresentado combina várias técnicas empregadas para a modelagem de objetos deformáveis, a saber:

- A detecção de colisões usa uma abordagem de volumes limitantes para diminuir o número de colisões potenciais e uma abordagem baseada em um *Hashing* espacial (usando subdivisão espacial) para encontrar as colisões reais [23]).
- A resposta às colisões envolve o cômputo da profundidade de penetração por propagação [7], o cálculo de uma região de deformação e seus vetores de deslocamento [7], e a resolução das colisões assimétricas [8] dos vértices envolvidos na colisão. Finalmente, a separação dos objetos é efetuada usando Busca Binária [19] encontrando a superfície de contato.
- A dinâmica está baseada numa técnica de casamento de formas na qual não é necessário o uso de malhas, embora estas sejam usadas no processamento de colisões e na integração do sistema, é usado um esquema de Euler explícito-implícito [13].

Resultados preliminares mostram a interação entre objetos constituídos de várias centenas de tetraedros em tempo real.

O resto do texto está organizado na seguinte forma: a seguinte seção apresenta os trabalhos relacionados, a seção 3 descreve o pré-processamento necessário, a seção 4 apresenta como a detecção de colisão é realizada em duas etapas, a seção 5 trata o processo de resposta às colisões, a seção 6 descreve o modelo dinâmico usado, a seção 7 apresenta o método de integração usado, a seção 8 mostra os resultados obtidos, e finalmente a seção 9 apresenta as conclusões e trabalhos futuros.

2. Trabalhos relacionados

O processo de detecção de colisão usualmente é dividido em duas etapas, uma etapa que determina grupos de objetos em colisão potencial (*broad phase*, em inglês), e a segunda etapa que executa verificações de colisão exatas entre os objetos em colisão potencial (*narrow phase*, em inglês).

2.1. Detecção de colisão grosseira

Esta etapa tipicamente é realizada usando um esquema de varredura e poda (*sweep and prune*, em inglês) [5]. Esta técnica mantém uma lista ordenada para cada eixo principal, onde os elementos na lista são as projeções dos objetos nos eixos. Estas listas são atualizadas frequentemente usando as formas atuais dos objetos. Assim, um par de objetos pode intersectar apenas se os intervalos de suas projeções intersectam nos três eixos.

2.2. Detecção de colisão exata

As técnicas de detecção de colisão são classificadas em três grandes grupos: baseadas em volumes limitantes, baseadas em subdivisão espacial e as baseadas no espaço-imagem. As técnicas que usam hierarquias de volumes limitantes, tem mostrado maior eficiência para detecção de colisão entre objetos rígidos [6, 11] do que para objetos deformáveis [25, 9], já que precisam ser atualizados frequentemente. Os volumes limitantes permitem aproximar um objeto complexo por outro de geometria mais simples, ajustando o objeto original da melhor forma possível. Assim, testes de colisões são realizados apenas entre pares de objetos cujos volumes limitantes se intersectam. Já as técnicas baseadas em subdivisão espacial são mais adequadas para objetos deformáveis, já que o espaço do mundo é subdividido em células, onde cada célula contém primitivas do objeto que a intersectam. Uma dificuldade é a escolha da estrutura de dados que represente o espaço 3D. Teschner apresentou uma estrutura de dados baseada numa tabela de dispersão (*hash*), que é eficiente em memória e tempo computacional [23, 24]. Em contraste com as técnicas anteriores, as técnicas baseadas no espaço-imagem fazem uso de funcionalidades das placas gráficas (GPUs). Estas técnicas são apropriadas para aplicações interativas já que não requerem de pré-processamentos nem de atualizações de estruturas de dados complexas [24], mas em geral são restritas a objetos convexos [2, 17].

2.3. Resposta a colisões

Existem dois esquemas para obter a resposta a colisões: os métodos baseados em restrições e os métodos baseados em penalidades.

Os métodos baseados em restrições evitam a interpenetração entre objetos e são particularmente interessantes na dinâmica de corpos rígidos ou semi-rígidos, já que estes fornecem menos graus de liberdade [10, 16]. Já os métodos baseados em penalidades computam uma força de resposta para cada ponto colidido, cujo valor está relacionado com uma medida de interpenetração. Consequentemente, o esforço numérico cresce com a intensidade da penetração [12], o que torna necessária uma computação robusta da profundidade de penetração dos pontos colididos.

Pode-se dizer que os métodos baseados em penalidades são mais rápidos, enquanto que os métodos baseados em restrições são mais robustos, permitindo intervalos de tempo maiores. Entretanto, os sistemas de restrições são mais custosos e inadequados para resolver colisões em tempo real.

2.4. Animação

Na dinâmica, pode-se empregar métodos como [15]: os métodos de elementos finitos (FEM) são usados para discretizar o objeto em um conjunto de células disjuntas (malha), e resolve comportamentos elásticos para cada partícula. Nos sistemas massa-mola [4, 3, 22] os modelos consistem de partículas conectadas entre si por uma rede de molas, onde o alongamento das molas gera forças elásticas em cada massa. São intuitivos e simples de se implementar. Entretanto, eles não são necessariamente exatos porque não são construídos sobre elasticidade contínua. Os métodos sem malhas se originam no FEM, mas não requerem nenhuma informação de conectividade, uma vez que os objetos são tratados como sistemas de partículas [13].

3. Pré-processamento

As estruturas de dados são iniciadas e os parâmetros iniciais da tabela de dispersão (*hash*) são calculados [23] (veja Figura 1):

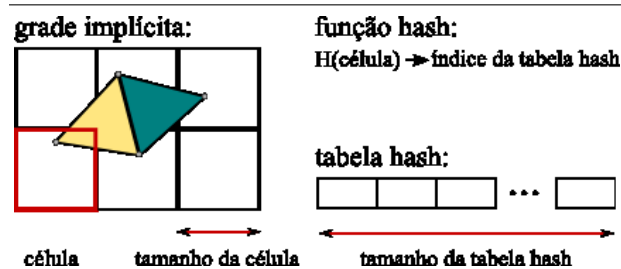


Figura 1. parâmetros da tabela de dispersão (*hash*).

tamanho da tabela: o tamanho ótimo tem relação direta com o número de primitivas a serem avaliadas; tipicamente é um número primo grande e influi no desempenho do algoritmo;

tamanho da célula: influi diretamente no número de primitivas que intersectam uma célula da grade. O sistema usa a média do comprimento das arestas dos tetraedros;

função “*hash*”: serve para encontrar um índice que distribua as células arbitrariamente na tabela de dispersão:

$$h = \text{hash}(i, j, k) = (i \cdot \alpha \oplus j \cdot \beta \oplus k \cdot \gamma) \% n, \quad (1)$$

onde i, j, k são coordenadas do vértice nas coordenadas da grade, α, β e γ são números primos grandes e n é o tamanho da tabela de dispersão.

4. Detecção de colisões

A detecção de colisões é um componente fundamental na simulação física e um problema amplamente pesquisado nas últimas décadas. Seu objetivo não é apenas saber quais objetos colidem, mas também os pontos exatos de colisão e assim permitir computar uma resposta realista à colisão. Em se tratando de objetos deformáveis, problemas como auto-colisões, busca de informação de colisão (profundidade de penetração, pontos de contato, etc.) e eficiência no desempenho têm que ser avaliados. Entre as abordagens para tratar os problemas mencionados temos os métodos de partição do objeto, que empregam hierarquias de volume limitante para estruturar o processo geométrico dos objetos envolvidos em colisões e os métodos de subdivisão espacial, que repartem o espaço de forma explícita ou implícita, a fim de detectar regiões ocupadas por mais de um objeto.

Enquanto que abordagens de subdivisão espacial mapeiam todo o espaço do universo da simulação, o método proposto mapeia somente as regiões onde existem colisões potenciais, evitando assim atualizar desnecessariamente a tabela, para tanto primeiro se usa uma fase de filtragem grosseira (em inglês, *broad phase*), que trata colisões entre os volumes limitantes dos objetos diminuindo o número de colisões, e subsequentemente a fase de filtragem exata, para encontrar as colisões reais nessas regiões, usando o *Hashing* espacial, baseado em subdivisão espacial.

4.1. Filtragem grosseira

O objetivo dessa fase é descartar regiões onde não há colisões. Para tanto, é efetuado um teste simples de colisão entre cada par de objetos, considerando apenas seus volumes limitantes. Para um par de objetos A e B , existe colisão se a distância entre os centros de suas esferas é menor que a soma de seus raios (ver equação). Assim, para cada par de objetos, são identificados os vértices envolvidos na colisão potencial ($v \subset A \cap B$), usando apenas consultas para verificar se um vértice v do objeto A está dentro da esfera do objeto B e vice-versa.

$$(c_1 - c_2)(c_1 - c_2) < (r_1 + r_2)^2 \quad (2)$$

4.2. Uso da subdivisão espacial

Após ter encontrado as regiões de colisão potencial, encontramos as colisões reais, isto é, encontrarmos os vértices que realmente colidem. Para encontrar tais vértices usamos um *Hashing* espacial [23], que subdivide implicitamente o espaço do mundo numa grade uniforme de células regulares, onde cada célula é uma caixa alinhada com os eixos coordenados e contém uma lista das primitivas de objetos que a intersectam, armazenando-as num endereço da

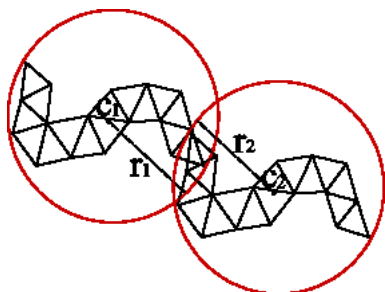


Figura 2. intersecção entre duas esferas.

tabela de dispersão obtido usando a função “hash” = $h = hash(i, j, k)$. Cada célula tem três coordenadas inteiras i, j e k que a associam a uma posição na tabela de dispersão. Este esquema atenua o desperdício de memória mantendo acesso rápido às células.

A tabela é atualizada somente nas células onde existe colisão potencial. Também são mapeados os tetraedros e as faces incidentes nesses vértices e inseridos na tabela. Para evitar inserções duplicadas, é armazenado em cada célula um inteiro que identifica em qual quadro da animação esta foi alterada. Este identificador é conhecido como *timestamp*. Assim, cada célula tem uma lista de tetraedros e faces que a intersectam e vice-versa. As células são acessadas por um índice inteiro como mostra a Figura 3.

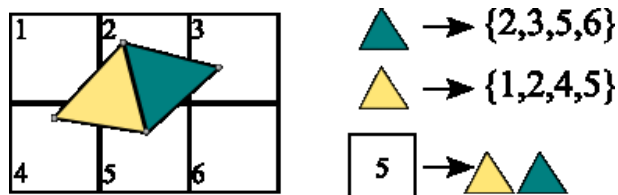


Figura 3. a célula 5 contém as faces verde e amarela, do mesmo modo, estas faces contêm as células 2, 3, 5, 6 e 1, 2, 4, 5, respectivamente.

A seguir, a inserção das primitivas:

vértices: cada vértice é armazenado na célula correspondente da forma:

$$(i, j, k) := \lfloor x/l \rfloor, \lfloor y/l \rfloor, \lfloor z/l \rfloor.$$

Logo, uma função *hash* insere a célula (i, j, k) num índice $h = hash(i, j, k)$ da tabela de dispersão como mostra a Figura 4;

tetraedros: para simplificar o armazenamento do tetraedro nas suas células correspondentes é utilizado o *AABB* do tetraedro. Assim, os vértices mínimo e máximo

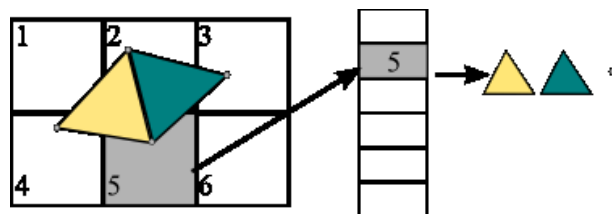


Figura 4. exemplo de mapeamento de primitivas de objetos. As faces amarela e verde são armazenadas na célula 5. Por outra parte, a célula 5 é mapeada num índice arbitrário da tabela de dispersão.

que descrevem o *AABB* representam o intervalo das células que intersectam o *AABB*. Finalmente, todos os índices h encontrados são inseridos na tabela de dispersão, como mostra a Figura 4;

faces: para mapear as faces, segue-se o mesmo esquema usado para inserir tetraedros, usando as *AABB* das faces;

arestas: usa-se uma técnica [1] que permite armazenar somente as células cruzadas por arestas de intersecção, avaliando a vizinhança da aresta na partição do espaço 3D.

A cada quadro da animação, os tetraedros de cada objeto são inseridos nas células que os intersectam, indistintamente do objeto. Numa segunda etapa, um teste de colisão é efetuado para encontrar os *vértices colididos*, que são os vértices que penetram tetraedros de outros objetos ou do mesmo objeto (auto-colisão). Logo, para cada entrada da tabela é executado um teste de intersecção entre os vértices que pertencem à região em colisão potencial e os tetraedros contidos nessa célula, encontrando assim os vértices colididos.

Para cada vértice v é realizado um teste de intersecção com cada tetraedro t da célula. Para acelerar o processo, primeiro se verifica se o vértice está dentro do *AABB* do tetraedro ($v \subset t.AABB()$). Em caso afirmativo, se verifica se está dentro do tetraedro. Se a verificação resulta verdadeira, o vértice é marcado como vértice colidido. Este processo também detecta auto-intersecção, como mostra a Figura 5. A verificação ponto-tetraedro pode ser feita usando coordenadas baricêntricas.

5. Resposta às colisões

O processo de detecção de colisão, descrito na seção anterior, fornece uma lista de vértices colididos. Estes são usados na resposta às colisões. O processo começa com o cálculo da profundidade de penetração dos vértices colididos, seguido do cômputo da região de deformação, a

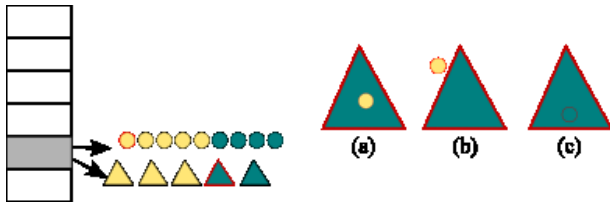


Figura 5. para cada entrada não vazia da tabela de dispersão, testa-se interseção entre vértices e tetraedros, para verificar se existe: (a) colisão, (b) não colisão ou (c) auto-colisão.

qual contém todos os vértices envolvidos na colisão. Finalmente, os objetos são separados usando uma técnica de Busca Binária.

5.1. Profundidade de penetração

A profundidade de penetração entre dois objetos é a translação mínima necessária para separá-los.

O *Algoritmo baseado em malhas tetraedrais* [7] computa a profundidade e a direção de penetração para cada vértice colidido fazendo uso das malhas tetraedrais. Tal profundidade pode ser usada para computar forças de penalidade que forneçam respostas realistas às colisões.

A idéia é classificar os vértices colididos em relação a sua profundidade de penetração. Assim, primeiro são avaliados os vértices mais próximos à superfície. Essa informação é então propagada aos vértices que possuem maior penetração, até que todos os vértices colididos sejam processados. No final, cada vértice colidido possui uma profundidade de penetração d e um vetor de direção de penetração $\vec{r}$.

Finalmente, um triângulo de contato é calculado para cada vértice colidido. Tal triângulo de contato é uma face na superfície do objeto penetrado que intersecta com o vetor de direção de profundidade de penetração.

Classificação: se um vértice colidido tem um ou mais vértices incidentes não colididos este é um *vértice da borda*, caso contrário é um *vértice interno* (Figura 6).

Profundidade de penetração dos vértices da borda: primeiro são identificadas as *arestas de intersecção*, ou seja, arestas que possuem um vértice da borda e um vértice não colidido. A seguir, procura-se a face da superfície mais próxima que intersecta a aresta. O teste pode ser feito usando coordenadas baricêntricas, já que permitem computar a normal da face intersectada n_{srf} e obter o ponto exato de intersecção p_{int} . Estes dados são armazenados na aresta e servem para encontrar a profundidade de penetração dos vértices da borda (Figura 7). Para cada vértice da borda v ,

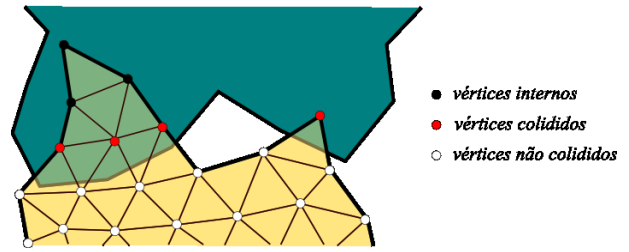


Figura 6. vértices colididos: da borda ou internos.

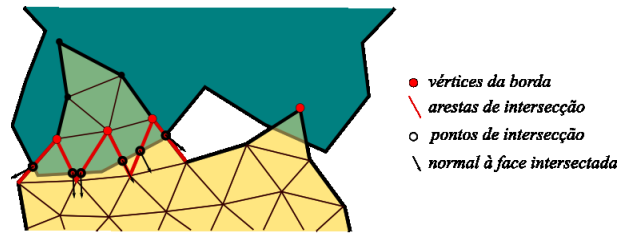


Figura 7. vértices da borda, arestas de intersecção, pontos exatos de intersecção e normais às faces intersectadas.

é computada sua profundidade de penetração $d(v)$ e seu vetor de direção de penetração $\vec{r}(v)$ da seguinte forma: para cada aresta de intersecção incidente em v , computa-se um peso ponderado w entre o ponto de intersecção da aresta p_{int} e v :

$$w(p_{int}, v) = \frac{1}{\|p_{int} - v\|^2}, \quad (3)$$

logo, são computados $d(v)$ e $\vec{r}(v)$ através das fórmulas:

$$d(v) = \frac{\sum_{i=1}^k (w(p_i, v) \cdot (p_i - v) \cdot n_i)}{\sum_{i=1}^k (p_i, v)}, \quad (4)$$

$$\vec{r}(v) = \frac{\sum_{i=1}^k (w(p_i, v) \cdot n_i)}{\sum_{i=1}^k (p_i, v)}, \quad (5)$$

onde k é o número de arestas de intersecção incidentes em v , p_i e n_i representam o i -ésimo p_{int} e n_{srf} a serem avaliados. A Figura 8 ilustra a profundidade de penetração para os vértices da borda v . No final, todos os vértices da borda são inseridos numa lista de vértices processados.

Profundidade de penetração dos vértices internos: após o processamento de todos os vértices da borda, a informação da profundidade de penetração obtida é propagada para os vértices internos v_i (Figura 9).

Encontra-se uma lista de vértices a processar, isto é, dada uma lista de vértices processados, é obtida uma lista de vértices a processar. Desta forma, vê-se que o cômputo da

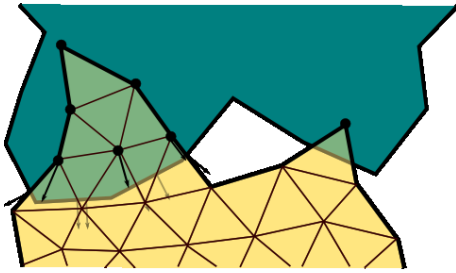


Figura 8. profundidade de penetração de vértices da borda.

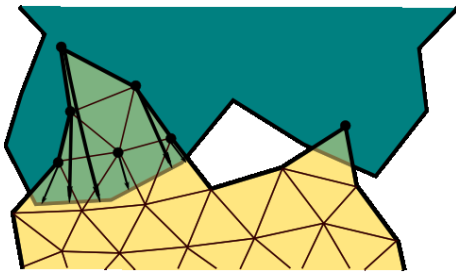


Figura 9. profundidade de penetração de vértices internos usando propagação.

profundidade de penetração é feito por níveis, até que todos os vértices internos sejam computados.

Computa-se a profundidade de penetração $d(v)$ e a direção de penetração $\vec{r}(v)$ dos vértices internos v , enquanto existam vértices internos a serem processados. Primeiro, para cada vértice processado v_{inc} incidente no vértice v , computa-se um peso ponderado μ :

$$\mu(v_{inc}, v) = \frac{1}{\|v_{inc} - v\|^2},$$

depois, computa-se $d(v)$ e $\vec{r}(v)$ segundo as fórmulas:

$$d(v) = \frac{\sum_{j=1}^k (\mu(v_j, v) \cdot ((v_j - v) \cdot \vec{r}(v_j) + d(v_j)))}{\sum_{j=1}^k \mu(v_j, v)}, \quad (6)$$

$$\vec{r}(v) = \frac{\sum_{j=1}^k \mu_j \vec{r}(v_j)}{\sum_{j=1}^k \mu_j}, \quad (7)$$

onde k é o número de vértices incidentes no vértice v , e v_j e o j -ésimo v_{inc} avaliado.

5.2. Região de deformação

5.2.1. Resolução de colisões assimétricas Após o cálculo da profundidade de penetração, os objetos devem ser separados, a fim de fornecer um estado fisicamente correto. Para

tal encontra-se uma *região de deformação* dos vértices envolvidos na colisão. Esta região de deformação considera os vértices colididos e os vértices dos triângulos de contato destes, que não necessariamente colidem (x_i , x_j e x_k na Figura 10, colisões com essa configuração são chamadas de *colisões assimétricas*). Para todos os vértices da região de deformação é computado um vetor de deslocamento, que será usado no processo de separação. Nesta etapa também são resolvidas as colisões assimétricas, usando o método de projeção de Jakobsen [8]. Este método projeta os vértices colididos para fora do obstáculo, movendo-os até que fiquem livres da intersecção.

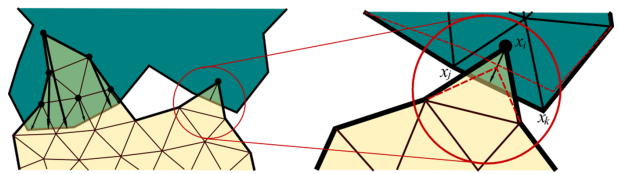


Figura 10. solução de colisões assimétricas.

Por exemplo, na Figura 10, o vetor de deslocamento de x_i é sua profundidade de penetração, e os vetores de deslocamento dos vértices não colididos x_j e x_k são calculados da forma:

$$\vec{s}_j = \frac{\alpha_1}{\alpha_1^2 + \alpha_2^2} (x'_i - x_i),$$

$$\vec{s}_k = \frac{\alpha_2}{\alpha_1^2 + \alpha_2^2} (x'_i - x_i),$$

onde x_i é o vértice colidido, x'_i sua projeção na aresta de contato (x_j e x_k), s_j e s_k são os vetores de deslocamento de x_j e x_k , respectivamente, os valores α representam a proporção de deslocamento dos vértices da aresta de contato, sendo que $\alpha_1 + \alpha_2 = 1$.

5.2.2. Busca binária para computar a superfície de contato Após ter computado os vetores de deslocamento, é estimada uma superfície de contato implícita que permite a separação dos objetos. Este deslocamento deve ser algo em torno da metade do comprimento dos vetores de deslocamento, para tal pode ser usado um esquema de busca binária [19]:

$$x^{i+1} \leftarrow x^i \pm \frac{1}{2^{(i+2)}} s, \quad (8)$$

onde s é o vetor de deslocamento, x é a posição do vértice de deslocamento original. Em outras palavras, para cada iteração, o intervalo de deslocamento é dividido por dois.

6. Dinâmica do sistema

Na animação dos objetos é aplicado o método de casamento de formas usando mínimos quadrados [13], onde

os nós da malha são tratados como partículas e animados como um sistema de partículas simples, sem conectividade. Esta técnica permite encontrar uma correspondência entre os vértices no estado inicial $\mathbf{x}_i^0$ e estado atual $\mathbf{x}_i$. Ou seja, a correspondência com o menor erro quadrático.

O objetivo é encontrar a melhor matriz de transformação afim, que permita efetuar deformações plausíveis e que permita que o objeto volte à sua forma original (Figura 11).

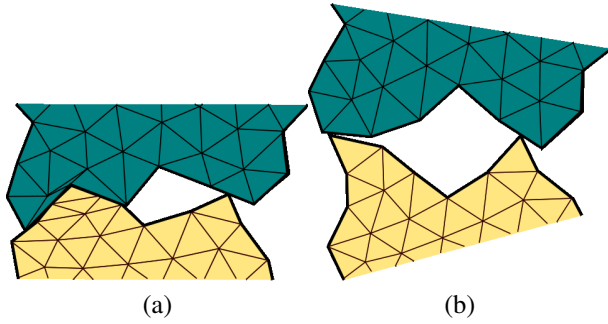


Figura 11. (a)superfície de contato, objeto deformado e (b) a possível solução da deformação.

O algoritmo tem dois componentes principais: (1) encontrar uma transformação rígida ótima que aproxime uma nova posição e a orientação do objeto (problema de correspondência) e (2) mover as partículas para as posições alvo aplicando um modelo de deformação linear.

Considerando os pesos das partículas, uma transformação linear composta de uma translação $\mathbf{t}$ e uma rotação $\mathbf{R}$ sobre o ponto $\mathbf{t}_0$ pode ser encontrada minimizando:

$$\sum_i w_i (\mathbf{R}(\mathbf{x}_i^0 - \mathbf{t}_0) + \mathbf{t} - \mathbf{x}_i)^2.$$

Podemos estabelecer para o problema $w_i = m_i$, isto é, que a ponderação das partículas é representada por suas massas, e que os vetores de translação ótimos são o *centro de massa* da forma inicial e o centro de massa da forma atual. Assim, temos

$$\mathbf{t}_0 = \mathbf{x}_{cm}^0 = \frac{\sum_i m_i \mathbf{x}_i^0}{\sum_i m_i}, \quad e \quad \mathbf{t} = \mathbf{x}_{cm} = \frac{\sum_i m_i \mathbf{x}_i}{\sum_i m_i}.$$

Para encontrar um $\mathbf{R}$ ótimo é necessário encontrar uma matriz de transformação linear $\mathbf{A}$. Observe que $\mathbf{A}$ não necessariamente é orto-normal. Para encontrar $\mathbf{A}$ define-se as posições relativas

$$\mathbf{q}_i = \mathbf{x}_i^0 - \mathbf{x}_{cm}^0 \quad e \quad \mathbf{p}_i = \mathbf{x}_i - \mathbf{x}_{cm}.$$

Estas posições definem o campo de deformação das partículas, tratando deformações elásticas de forma implícita.

O problema de correspondência é resolvido com a técnica de mínimos quadrados, que é um método clássico usado para estabelecer a melhor correspondência com o menor erro quadrático. Assim, a matriz de transformação linear ótima $\mathbf{A}$ é encontrada minimizando o termo:

$$\sum_i m_i (\mathbf{A} \mathbf{q}_i - \mathbf{p}_i)^2.$$

Onde:

$$\mathbf{A} = (\sum_i m_i \mathbf{p}_i \mathbf{q}_i^T) (\sum_i m_i \mathbf{q}_i \mathbf{q}_i^T)^{-1} = \mathbf{A}_{pq} \mathbf{A}_{qq}, \quad (9)$$

na equação $\mathbf{A}_{pq}$ é uma matriz de correlação e $\mathbf{A}_{qq}$ é uma matriz simétrica que pode conter escala mas não rotações. Conclui-se portanto que a estimativa de $\mathbf{A}$ requer que se estime o componente de rotação da matriz $\mathbf{A}_{pq}$, o que pode ser realizado através de algum método de decomposição. Em particular, o trabalho de Müller et al. [13] emprega decomposição polar de matrizes [18] e o algoritmo Jacobi [20], que é um algoritmo simples e estável, usado para a diagonalização de matrizes e o cômputo de auto-vetores. O funcionamento do algoritmo se caracteriza pela aplicação de sucessivas operações elementares chamadas de *Rotações de Jacobi*. Geralmente são realizadas de 5 a 10 *Rotações de Jacobi* para diagonalizar uma matriz 4×4 .

Assim, a matriz de rotação ótima é:

$$\mathbf{R} = \mathbf{A}_{pq} (\sqrt{\mathbf{A}_{pq}^T \mathbf{A}_{pq}})^{-1}.$$

Finalmente, havendo encontrado a matriz de rotação é realizada a seguinte transformação linear para rotação e translação do objeto, a fim de encontrar a melhor posição alvo de cada partícula:

$$\mathbf{g}_i = \mathbf{R}(\mathbf{x}_i^0 - \mathbf{x}_{cm}^0) + \mathbf{x}_{cm}.$$

Os pontos são movidos para as posições $\mathbf{g}_i$, exatamente a cada intervalo de tempo.

7. Integração física

Nosso esquema deriva de um método de integração de Euler, que inclui uma parte explícita (a atualização da velocidade) e uma parte implícita (a atualização da posição).

Para computar as posições dos objetos, as acelerações e velocidades são integradas numericamente para cada intervalo de tempo utilizando as seguintes equações:

$$\mathbf{v}_i(t + \Delta t) = \mathbf{v}_i(t) + \alpha \frac{\mathbf{g}_i(t) - \mathbf{x}_i(t)}{\Delta t} + \frac{\Delta t}{m_i} f_{ext}(t), \quad (10)$$

$$\mathbf{x}_i(t + \Delta t) = \mathbf{x}_i(t) + \Delta t \mathbf{v}_i(t + \Delta t), \quad (11)$$

onde $\mathbf{v}_i(t + \Delta t)$ é a velocidade no próximo intervalo de tempo, $\mathbf{v}_i(t)$ é a velocidade no intervalo de tempo atual, α

é o parâmetro que simula rigidez, $\mathbf{g}_i(t)$ é a melhor posição do vértice $\mathbf{x}_i(t)$ (para resolver a deformação), Δt é a magnitude do intervalo de tempo, m_i é a massa da partícula e $\mathbf{f}_{ext}(t)$ são as forças externas.

8. Resultados

O protótipo destinado a servir como prova de conceito para o sistema foi desenvolvido utilizando a linguagem C++, a API OpenGL e a biblioteca Glut. Uma observação importante é o uso do Vertex Buffer Object (VBO), extensão do OpenGL, para a obtenção de uma boa taxa de renderização dos objetos. Para avaliar o protótipo desenvolvido foram conduzidos experimentos envolvendo objetos deformáveis de formas variadas. Todos os experimentos foram executados numa estação de trabalho com sistema operacional Linux (Fedora 8) com processador Intel Core 2 Duo a 2.4Ghz e 1 GB de memória.

Os experimentos visam avaliar o desempenho do sistema em função do número de objetos envolvidos na simulação e da complexidade desses objetos. A Tabela 1 mostra as resoluções em vértices, faces e tetraedros dos objetos usados.

Objeto	Vértices na superfície	vértices	faces	tetraedros
coelho	436	510	868	1750
pato	424	519	846	1819
tubo	220	340	436	1270
esfera	386	729	768	2560

Tabela 1. objetos com resoluções diferentes.

Todas as simulações usaram coeficiente de rigidez $\alpha = 0.8$ nos objetos (quase rígidos).

Foram coletados alguns indicadores em termos de quadros por segundo, quantidade de primitivas em colisão (vértices em colisão potencial, faces, tetraedros e vértices em colisão real) processadas a cada intervalo de tempo e a porcentagem gasta por cada sub-processo (detecção de colisões potenciais, detecção exata de colisões, profundidade de penetração, casamento de formas) em milisegundos.

Um primeiro experimento consiste na simulação física de uma cena com diferentes tipos de objetos (veja a Figura 12): 3 patos, 2 coelhos e 3 esferas (Tabela 1). Este resultado mostra que o protótipo lida com objetos de geometria arbitrária desde que estejam triangulados adequadamente.

Do experimento podemos obter algumas informações, por exemplo, a Figura 13 mostra o tempo gasto em milisegundos nos sub-processos mais importantes: detecção de colisões potenciais, detecção exata de colisões, computação

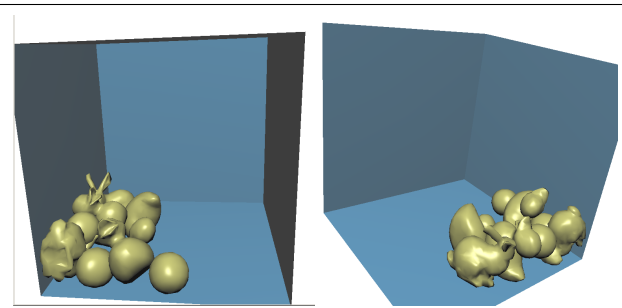


Figura 12. oito objetos em contato: 3 patos, 2 coelhos e 3 esferas. A cena contém 2952 vértices e 9917 tetraedros animados a 32 fps.

da profundidade de penetração e casamento de formas a cada intervalo de tempo. Pode-se ver que a detecção de colisões, nas duas fases, toma a maior parte do tempo, sendo que o cálculo da profundidade de penetração se mantém quase constante e o casamento de formas, usando todos os vértices da superfície dos objetos, toma um tempo insignificante em relação aos outros sub-processos.

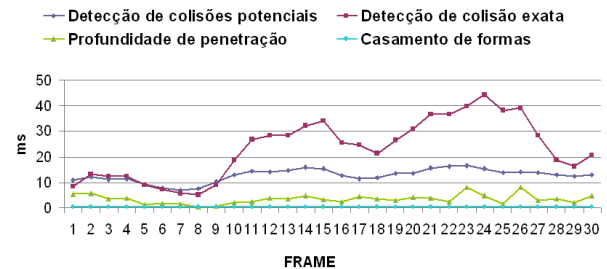


Figura 13. tempo gasto em milisegundos para cada sub-processo a cada intervalo de tempo.

A Figura 14 mostra, para o mesmo experimento, a quantidade de primitivas em colisão (vértices em colisão potencial, faces, tetraedros e vértices em colisão real) por intervalo de tempo. Note que a filtragem grosseira detecta uma quantidade de vértices em colisão potencial significativamente inferior ao total de vértices na cena. Em média, a simulação registra 487 vértices em colisão potencial, sendo que as colisões reais envolvem 37 faces, 82 tetraedros e 28 vértices em média.

Foram também conduzidos outros experimentos usando quantidades variáveis de esferas de forma a avaliar a escalabilidade do método. A Figura 15, mostra a simulação de

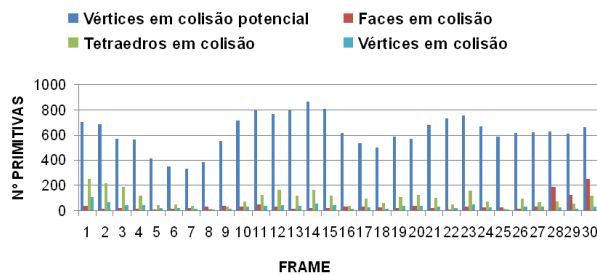


Figura 14. número de primitivas em colisão, a cada intervalo de tempo.

cenas com 8, 18 e 27 esferas (Tabela 1). As cenas contêm 5832 vértices e 20480 tetraedros (8 esferas), 13122 vértices e 46080 tetraedros (18 esferas) e 19683 vértices e 69120 tetraedros (27 esferas). O desempenho do método para esses experimentos foi de 62, 41 e 22 quadros por segundo em média, respectivamente. Um gráfico com a taxa de quadros por segundo a cada intervalo de tempo é mostrado na Figura 16.

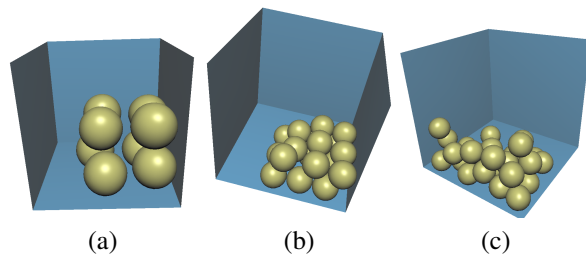


Figura 15. experimentos com 8 (a), 18 (b) e 27 (c) esferas.

9. Conclusões e Trabalhos Futuros

Foi apresentado um protótipo de sistema de animação baseada em física, que usa várias técnicas, como a detecção de colisões usando um *Hashing* espacial [21] e volumes limitantes, a resposta às colisões através do cômputo da superfície de contato, que utiliza o cálculo da profundidade de penetração por propagação [7], a estimativa dos vetores de deslocamento [7] e a resolução das colisões assimétricas [8] dos vértices da região de deformação, e Busca Binária para separar os objetos [19], a animação é feita usando uma técnica de casamento de formas e um integrador de Euler explícito-implícito [13].

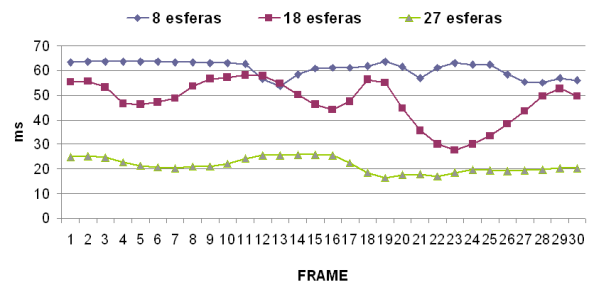


Figura 16. tempo gasto em milissegundos a cada intervalo de tempo de esferas com 8, 18 e 27 objetos.

A estrutura original de detecção de colisões foi estendida usando-se um mecanismo de detecção de regiões de colisão potencial, para minimizar as regiões atualizadas pela tabela de dispersão, onde se efetuam as colisões reais. Também, no esquema de resposta a colisões, é usado o método de projeção de Jakobsen [8] para resolver colisões assimétricas.

A detecção e resposta às colisões permitem identificar e resolver colisões dos objetos baseados em malhas tetraedrais, em objetos rígidos e deformáveis.

O sistema oferece simulações plausíveis, produzindo respostas às colisões de forma convincente e próxima às metodologias tradicionais, conseguindo manipular objetos complexos e objetos empilhados.

Para trabalhos futuros, planeja-se estender o protótipo do sistema apresentado, na animação, da deformação (agora linear) à quadrática, e plástica; na detecção de colisões o uso de uma hierarquia de esferas limitantes, e finalmente, para aperfeiçoar o código e alcançar um melhor desempenho, desenvolver o sistema usando *GPU*.

Referências

- [1] J. Amanatides and A. Woo. A fast voxel traversal algorithm for ray tracing. In *Eurographics '87*, pages 3–10. Elsevier Science Publishers, Amsterdam, North-Holland, 1987.
- [2] G. Baciú, W. Wong, and H. Sun. Recode: An image-based collision detection algorithm. *The Journal of visualization and Computer Animation*, 10(4):181–192, 1998.
- [3] D. Baraff and A. Witkin. Large steps in cloth simulation. In *SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 43–54, New York, NY, USA, 1998. ACM.
- [4] D. Breen, D. House, and M. Wozny. Predicting the drape of woven cloth using interacting particles. In *SIGGRAPH '94: Proceedings of the 21st annual conference on Compu-*

- ter graphics and interactive techniques, pages 365–372, New York, NY, USA, 1994. ACM.
- [5] J. D. Cohen, M. C. Lin, D. Manocha, and M. Ponamgi. I-collide: an interactive and exact collision detection system for large-scale environments. In *SI3D '95: Proceedings of the 1995 symposium on Interactive 3D graphics*, pages 189–ff., New York, NY, USA, 1995. ACM.
 - [6] S. Gottschalk, M. C. Lin, and D. Manocha. Obbtrees: a hierarchical structure for rapid interference detection. In *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 171–180, New York, NY, USA, 1996. ACM.
 - [7] B. Heidelberger, M. Teschner, R. Keiser, and M. Müller. Consistent penetration depth estimation for deformable collision response. In *Proceedings of Vision, Modeling, Visualization VMV'04*, pages 157–164, Stanford, USA, 2004.
 - [8] T. Jakobsen. Advanced character physics. In *Proceedings, Game Developer's Conference 2001*, SJ, USA, 2001. GDC Press.
 - [9] D. L. James and D. K. Pai. BD-Tree: Output-sensitive collision detection for reduced deformable models. *ACM Transactions on Graphics (SIGGRAPH 2004)*, 23(3), Aug. 2004.
 - [10] D. Kaufman, T. Edmunds, and D. Pai. Fast frictional dynamics for rigid bodies. *ACM Trans. Graph.*, 24(3):946–956, 2005.
 - [11] J. T. Klosowski, M. Held, J. S. Mitchell, H. Sowizral, and K. Zikan. Efficient collision detection using bounding volume hierarchies of k-dops. *IEEE Transactions on Visualization and Computer Graphics*, 04(1):21–36, 1998.
 - [12] M. Moore and J. Wilhelms. Collision detection and response for computer animation. *SIGGRAPH Comput. Graph.*, 22(4):289–298, 1988.
 - [13] M. Müller, B. Heidelberger, M. Teschner, and M. Gross. Meshless deformations based on shape matching. In *Proceedings of SIGGRAPH'05*, pages 471–478, New York, NY, USA, 2005.
 - [14] M. Müller, R. Keiser, A. Nealen, M. Pauly, M. Gross, and M. Alexa. Point based animation of elastic, plastic and melting objects. In *SCA '04: Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 141–151, Aire-la-Ville, Switzerland, Switzerland, 2004. Eurographics Association.
 - [15] A. Nealen, M. Müller, R. Keiser, E. Boxerman, and M. Carlson. Physically based deformable models in computer graphics. In *Computer Graphics Forum*, pages 809–836. Blackwell Publishing Ltd., 2006.
 - [16] M. Pauly, D. Pai, and L. Guibas. Quasi-rigid objects in contact. In *SCA '04: Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 109–119, Aire-la-Ville, Switzerland, Switzerland, 2004. Eurographics Association.
 - [17] M. Shinya and M.-C. Forgue. Interference detection through rasterization. In *The Journal of Visualization and Computer Animation*, volume 2, pages 132–134, 1991.
 - [18] K. Shoemake and T. Duff. Matrix animation and polar decomposition. In *Proceedings of the conference on Graphics interface '92*, pages 258–264, San Francisco, CA, USA, 1992. Morgan Kaufmann Publishers Inc.
 - [19] J. Spillmann and M. Teschner. Contact surface computation for coarsely sampled deformable objects. In *Proceedings of Vision, Modeling, Visualization VMV'05*, pages 16–18, Stanford, USA, 2005.
 - [20] H. Sumali. *A New Adaptive Array of Vibration Sensors*. PhD thesis, Mechanical Engineering Virginia Polytechnic Institute and State University, Virginia, USA, 1992.
 - [21] M. Teschner, B. Heidelberger, D. Manocha, N. Govindaraju, G. Zachmann, S. Kimmerle, J. Mezger, and A. Fuhrmann. Collision handling in dynamic simulation environments. In *Eurographics Tutorial # 2*, pages 1–4, Dublin, Ireland, 29 August 2005. Eurographics Association.
 - [22] M. Teschner, B. Heidelberger, M. Müller, and M. Gross. A versatile and robust model for geometrically complex deformable solids. In *CGI '04: Proceedings of the Computer Graphics International (CGI'04)*, pages 312–319, Washington, DC, USA, 2004. IEEE Computer Society.
 - [23] M. Teschner, B. Heidelberger, M. Müller, and D. Pomeranets. Optimized spatial hashing for collision detection of deformable objects. In *Proceedings of Vision, Modeling, Visualization VMV'03 Proceedings of SPM 2005*, pages 47–54, 2003.
 - [24] M. Teschner, S. Kimmerle, B. Heidelberger, G. Zachmann, L. Raghupathi, A. Fuhrmann, M.-P. Cani, F. Faure, N. Magnetat-Thalmann, W. Strasser, and P. Volino. Collision detection for deformable objects. In *Computer Graphics Forum*, pages 61–81. Eurographics Association, Eurographics Association and Blackwell Publishing, 2005.
 - [25] G. van den Bergen. Efficient collision detection of complex deformable models using aabb trees. *J. Graph. Tools*, 2(4):1–13, 1997.

Processamento de Imagens e Visão Computacional

Automatização do Ajuste de Segmentadores Neurais de Imagens Coloridas*

Fernando H. B. Cardoso[†] e Herman M. Gomes

Universidade Federal de Campina Grande, Departamento de Sistemas e Computação

Laboratório de Visão Computacional

Av. Aprígio Veloso, S/N, Bloco CN, Sala 202

{fernando, hmg}@dsc.ufcg.edu.br

Resumo

O ajuste de segmentadores de imagens que detectam materiais é executado manualmente, com pouca automatização. Uma vez que esta segmentação é geralmente um passo intermediário de uma aplicação maior da Visão Computacional, a falta de automatização faz com que esforço seja investido em uma tarefa secundária. Neste trabalho é proposta uma técnica de automatização do ajuste de Redes Neurais Artificiais que segmentam imagens ao detectar materiais baseando-se em informações de cor. Esta técnica não necessita de supervisão humana, permitindo uma redução no esforço de obtenção dos segmentadores. As redes neurais obtidas detectam 2,56% a mais da área da superfície que se deseja segmentar – em relação a outros segmentadores também testados – com taxa de falsas aceitações de 6,89%.

Abstract

Tuning material-detecting-capable image segmenters is performed manually, with little automation. Once the segmentation is usually an intermediary step to a myriad of applications within Computer Vision, the lack of automation leads to effort wasting in secondary tasks. In this work, we propose a technique to automatically tune neural networks that segment images based on color and texture information. This technique does not require human supervision, reducing the effort in obtaining segmenters. The automatically tuned neural networks detect 2.56% more of the material's surface – related to other segmenters also tested – with false acceptance rate of 6.89%.

* Resumo estendido de dissertação de mestrado.

[†] Este trabalho é parcialmente financiado pelo CNPq (831590/1999-8).

1. Introdução

A detecção de superfície de materiais em imagens é um passo importante para diversas aplicações em Visão Computacional. A detecção de pele humana, por exemplo, pode restringir o espaço de busca de algoritmos de localização de faces [4, 7, 11] ou fornecer informação semântica das imagens que indica a presença de pornografia [8]. Outro exemplo é a detecção de grama em quadros de vídeos esportivos, para facilitar a sua sumarização [3, 15].

A obtenção de segmentadores de imagens capazes de detectar materiais envolve (i) construir um conjunto de imagens segmentadas manualmente – ou, eventualmente, utilizar uma base pública de imagens –; (ii) analisar os melhores parâmetros – espaço de cores, quantidade de níveis de quantização, tamanho da janela de *pixels* a ser considerada, dentre outras – para detectar o material em questão; (iii) ajuste de um classificador que decide se cada *pixel* faz parte ou não da superfície do material e (iv) testar o segmentador para verificar a sua qualidade.

Estas tarefas são feitas com pouca ou nenhuma automatização, uma vez que, após extensiva revisão da literatura, não foi encontrado um método automático para obtenção e ajuste de segmentadores deste tipo. Assim, é necessário investir tempo e esforço para uma atividade secundária durante um projeto em Visão Computacional, que poderiam ser investidos em outras atividades que contribuem mais diretamente com o seu objetivo geral.

Desta maneira, a automatização dos processos de obtenção e ajuste de segmentadores introduz uma maior agilidade para alcançar resultados nos projetos de pesquisa em Visão Computacional que necessitam da detecção de materiais como passo intermediário, não sendo mais necessário desviar esforços para resolver um problema periférico.

A principal contribuição deste trabalho é a proposição de um método automatizado de obtenção e ajuste de segmentadores neurais capazes de detectar materiais (Seção 2), cujo protótipo preliminar foi testado para detectar dois tipos de material: grama e pele (Seção 3).

São apresentados também uma visão comparativa da técnica proposta com trabalhos relacionados (Seção 4) e as conclusões finais, juntamente com sugestões de trabalhos futuros (Seção 5).

2. Método Proposto

O método proposto neste trabalho tem por objetivo automatizar a obtenção e o ajuste de segmentadores neurais que são capazes de detectar materiais em imagens digitais: cada *pixel* da imagem é submetido a um classificador neural baseado em redes *perceptron* de múltiplas camadas – MLP – [5] para que seja determinada a sua pertinência ou não à superfície do material em questão.

2.1. Avaliação dos Classificadores

As métricas utilizadas para verificar a qualidade de um segmentador ζ capaz de detectar materiais são a Taxa de Detecção – $D_r(\zeta)$ – e a Taxa de Falsas Aceitações – $P_r^f(\zeta)$. A Taxa de Detecção é a razão entre a área da superfície que foi corretamente detectada – $P^t(\zeta)$ – e a área total da superfície do material em questão – S^P –, conforme a Equação 1. Já a Taxa de Falsas Aceitações é a razão entre a área da superfície que foi erroneamente considerada como pertencente ao material em questão – $P^f(\zeta)$ – e a área total da superfície que não faz parte do mesmo – S^N –, conforme a Equação 2.

$$D_r(\zeta) = \frac{P^t(\zeta)}{S^P} \quad (1)$$

$$P_r^f(\zeta) = \frac{P^f(\zeta)}{S^N} \quad (2)$$

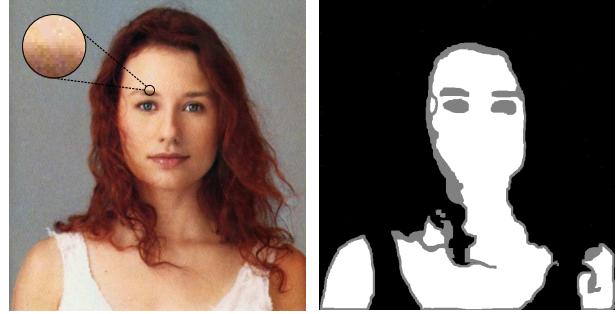
Para unificar estes dois valores em apenas um, utiliza-se como critério de decisão entre dois segmentadores a diferença entre as taxas de detecção e falsas aceitações $\delta(\zeta)$, conforme a Equação 3.

$$\delta(\zeta) = D_r(\zeta) - P_r^f(\zeta) \quad (3)$$

Com base nesta Equação, pode-se considerar o segmentador ζ_a melhor que um concorrente ζ_b quando $\delta(\zeta_a) > \delta(\zeta_b)$.

2.2. Composição dos Conjuntos de Imagens de Treinamento

O único insumo para obtenção de um segmentador utilizando a técnica deste trabalho é um conjunto de imagens acompanhadas do resultado de sua segmentação manual, da qual será extraído o conjunto de treinamento para a rede neural artificial. O rotulamento manual deve estar em conformidade com as seguintes diretrizes (Figura 1):



(a) Imagem original.

(b) Imagem segmentada.

Figura 1. Exemplo de imagem e sua respectiva segmentação manual para composição da base de treinamento – material: pele. A área em destaque exibe um exemplo de região de difícil segmentação.

- Os *pixels* correspondentes ao material que se quer detectar – superfície P – devem ser marcados de branco;
- Os que não fizerem parte de nenhuma superfície do material em questão – superfície N – são marcados de preto;
- Há ainda os *pixels* que são de difícil segmentação por estarem em regiões de transição entre superfícies ou que não se pode determinar com precisão em que superfície se encontram. Estes *pixels* são marcados de cinza (50% de brilho).

A extração do conjunto de treinamento é feita respeitando a igualdade entre as amostras de *pixels* pertencentes à superfície do material que se deseja detectar: em cada imagem da base segmentada manualmente são extraídos números iguais de amostras das duas classes. Isto garante também a igual distribuição no conjunto de treinamento como um todo e desta forma é possível treinar a rede neural sem que haja efeitos colaterais devido à maior presença de uma das classes.

2.3. Parâmetros dos Segmentadores

A técnica de automatização da obtenção e do ajuste dos segmentadores se baseia na determinação dos valores que seus parâmetros devem assumir para maximizar a taxa de detecção e minimizar a taxa de falsas aceitações. Os parâmetros dos segmentadores no estado atual do trabalho são: Espaço de Cor, Quantidade de Níveis de Quantização, Tamanho da Janela de *Pixels* e Quantidade de Neurônios na Camada Oculta.

2.3.1. Espaços de Cor O espaço de cor “padrão” no qual as imagens digitais são codificadas é o RGB. No entanto,

nem sempre este é o espaço que provê maior separabilidade entre as superfícies P e N . Desta maneira, são explorados também os espaços de cor HSV, YCbCr, XYZ, HLS, CIE $L^*a^*b^*$ e $L^*u^*v^*$.

2.3.2. Níveis de Quantização A quantidade de níveis de quantização depende do número de *bits* utilizados para representar as intensidades de cada canal. Apesar de, normalmente, as imagens digitais utilizarem 256 níveis de quantização em cada canal, a separabilidade entre P e N pode ser aumentada caso a quantidade de níveis de quantização seja reduzida. Para tanto, a quantidade de *bits* em cada canal é reduzida e são explorados também 128, 64, 32, 16, 8, 4 e 2 níveis de quantização em cada canal.

2.3.3. Tamanho da Janela de Pixels Considerar uma vizinhança em redor do *pixel* que está sendo classificado dá a rede maior informação acerca da superfície em que este se encontra, uma vez que considera a sua localização em uma região maior em vez de considerar sua cor isoladamente. O tamanho desta vizinhança é um dos parâmetros a serem determinados pela técnica proposta, que além de janelas de tamanho 1 – o *pixel* isolado – considera tamanhos ímpares de janela variando de 3 a 9.

2.3.4. Neurônios na Camada Oculta A qualidade dos resultados de uma rede neural artificial do tipo MLP depende da quantidade de neurônios em sua camada oculta [5]. Desta forma, é necessário explorar a quantidade ideal para cada tipo de material que se deseja detectar. No estágio atual do trabalho os valores explorados para a quantidade de neurônios na camada escondida vai de 3 a 12.

2.4. Obtenção e Ajuste dos Segmentadores

A obtenção e o ajuste do segmentador neural capaz de detectar o material desejado ocorre variando-se automaticamente parâmetros especificados na Seção 2.3 um a um para que se encontre a combinação mais adequada dos mesmos (Figura 2). Primeiramente fixa-se a quantidade de níveis de quantização em 256, o tamanho da janela de *pixels* em 1 e a quantidade de neurônios na camada escondida em 6 para que se possa testar todos os espaços de cor a fim de se determinar o espaço de cor C_{best} cujo segmentador apresenta menor δ .

Em seguida, para que seja obtida a quantidade de níveis de quantização adequada Q_{best} , testam-se todos os valores que este parâmetro pode assumir, utilizando C_{best} como espaço de cor e fixando-se o tamanho da janela em 1 e a quantidade de neurônios na camada escondida ainda em 6 e adota-se a que apresentar δ mais alto.

O terceiro parâmetro a ser ajustado é o tamanho da janela de *pixels*: para obter o tamanho W_{best} ideal são testados segmentadores obtidos com cada um dos tamanhos possíveis, no espaço de cores C_{best} com Q_{best} níveis de quanti-

zação em cada canal e é escolhido o que apresentar melhor δ .

Por fim, a quantidade de neurônios na camada escondida é ajustada da mesma maneira: são obtidos segmentadores para cada uma das quantidades de neurônios na camada escondida e a quantidade H_{best} ideal é a que apresentar maior δ .

Vale ressaltar que, para cada configuração de rede neural MLP testada (combinação de espaço de cor, quantidade de níveis de quantização, tamanho de janela e quantidade de neurônios na camada oculta) são treinadas 10 redes diferentes, a fim de se reduzir os efeitos da aleatoriedade inerente à inicialização das MLPs [5]. Dentre estas 10, é adotada a que apresentar maior D_r .

3. Resultados Obtidos

Com o objetivo de verificar a aplicabilidade da técnica proposta, ela foi utilizada para obter automaticamente segmentadores para dois tipos de materiais: pele e grama. Os resultados fornecidos por estes segmentadores foram então confrontados com os resultados de outros trabalhos, e de classificadores genéricos, a fim de se estabelecer uma análise comparativa da técnica abordada.

3.1. Detecção de Grama em Quadros de Vídeos Desportivos

A detecção de grama em quadros de vídeos desportivos é útil para prover informações que subsidiam a sua posterior sumarização (geração automática de resumos das partidas, contendo apenas os melhores momentos), através da distinção entre tomadas de longa distância - lances que provavelmente serão interessantes para o espectador - e de curta distância (*closes* e tomadas da torcida, por exemplo) - que provavelmente não são interessantes para o entendimento da partida -, da determinação da posição da bola com relação ao campo de jogo, entre outros [10].

O conjunto de imagens que compõe o conjunto de treinamento para obtenção e ajuste do segmentador foi composta de 25 imagens extraídas da base de dados utilizada por Firmino Jr. e Gomes [3] para parametrizar seu detector de grama baseado em casamento de padrões. Devido ao fato de a base ter sido preparada por terceiros, não há regiões de indecisão (áreas marcadas de cinza) neste conjunto de treinamento. A Figura 3 exibe um exemplo de imagem contida neste conjunto.

Os parâmetros ideais encontrados para o segmentador de grama foram:

- Espaço de Cor: RGB;
- Níveis de Quantização: 128;
- Tamanho da Janela: 1 *pixel* e

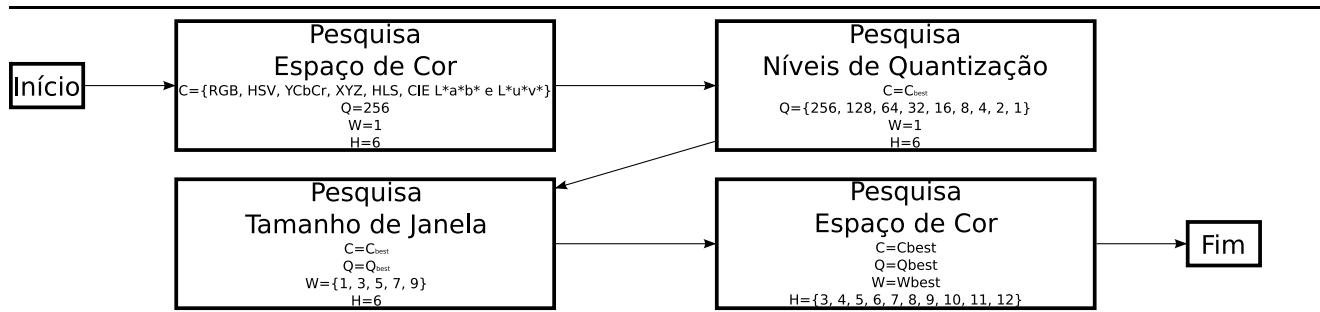


Figura 2. Esquema do ajuste dos segmentadores utilizando a técnica proposta.



(a) Imagem original.

(b) Imagem segmentada.

Figura 3. Exemplo de imagem e sua respectiva segmentação manual para composição da base de treinamento para o segmentador de grama.



(a) Imagem Original.

(b) Imagem Segmentada.

Figura 4. Exemplo de imagem segmentada com a técnica proposta.

Segmentador (ζ)	$D_r(\zeta)$	$P_r^f(\zeta)$	$\delta(\zeta)$
MLP Auto-treinada	88,42%	5,93%	82,50%
Naïve Bayes	86,25%	6,42%	79,83%
Firmino Jr. e Gomes	98,12%	20,79%	77,33%

Tabela 1. Resultados Obtidos para Segmentação de Grama

- Neurônios na Camada Oculta: 3.

O segmentador obtido com estes parâmetros foi testado, juntamente com um classificador *bayesiano* – gerado através da plataforma de mineração de dados Weka [16] – e com o segmentador de grama proposto por Firmino Jr. e Gomes [3], em um conjunto de imagens de teste composto pelas demais imagens da base de dados da qual foi extraído o conjunto de treinamento. A Tabela 3.1 exibe os resultados obtidos e a Figura 4 exibe dois exemplos de imagens segmentadas utilizando a rede neural MLP automaticamente obtida.

A Tabela de mostra que os resultados obtidos pela rede neural MLP treinada automaticamente utilizando o método proposto neste trabalho foram os melhores para detecção de grama em quadros de vídeos desportivos, uma vez que ob-

teve o maior δ (ver Equação 3).

3.2. Detecção de Pele em Fotografias

A detecção de pele em fotografias é utilizada para reduzir o espaço de busca em outras aplicações da Visão Computacional, como a detecção de faces humanas, ou a detecção de nudez, uma vez que em ambas aplicações só faz sentido executar buscas em regiões compostas de pele humana.

Para que se pudesse utilizar a técnica proposta e obter um segmentador de pele, foi utilizado um subconjunto da base de imagens fornecida por Jones e Regh [8]. Este subconjunto foi re-segmentado manualmente devido a alguns erros encontrados na segmentação semi-manual executada pelos autores. O subconjunto utilizado contava com 120 imagens. A Figura 1 (página 2) exibe um exemplo de imagem contida neste conjunto.

Os parâmetros ideais encontrados para o segmentador de pele foram:

- Espaço de Cor: YCbCr;
- Níveis de Quantização: 128;
- Tamanho da Janela: 9 *pixel* e
- Neurônios na Camada Oculta: 8.

O segmentador obtido com estes parâmetros foi testado, juntamente com um classificador *bayesiano* – também gerado através da plataforma de mineração de dados Weka –,

Segmentador (ζ)	$D_r(\zeta)$	$P_r^f(\zeta)$	$\delta(\zeta)$
HSV Cluster	88,03%	12,01%	76,02%
MLP Auto-treinada	81,40%	6,89%	74,51%
RGB-H-CbCr	61,17%	7,52%	53,63%
Naïve Bayes	79,60%	31,90%	47,70%
Cardoso e Gomes	72,77%	34,87%	37,90%

Tabela 2. Resultados Obtidos para Segmentação de Pele



(a) Imagem Original.

(b) Imagem Segmentada.

Figura 5. Exemplo de imagem segmentada com a técnica proposta.

com um segmentador de pele probabilístico de nossa autoria [2], outro que utiliza o modelo de pele RGB-H-CbCr proposta por Nusirwan *et al.* [12] e por fim, com um segmentador baseado na técnica de *cluster* no espaço de cores HSV, proposto por Herodotou *et al.* [6].

O conjunto de imagens de teste composto por 118 imagens também retiradas da base. A Tabela 3.2 exibe os resultados obtidos e a Figura 5 exibe dois exemplos de imagens segmentadas utilizando a rede neural MLP automaticamente obtida.

A Tabela de mostra que os resultados obtidos pela rede neural MLP treinada automaticamente utilizando o método proposto neste trabalho não foram os melhores para detecção de pele humana. No entanto, se forem consideradas a automatização da técnica proposta (não é uma técnica específica para detecção de pele) e a proximidade dos valores de δ da MLP Auto-treinada e do segmentador por *cluster* no espaço HSV, os resultados da técnica proposta são muito satisfatórios.

4. Trabalhos Relacionados

Para melhor organizar esta Seção, ela foi dividida em duas subseções: uma dedicada à detecção de materiais e outra à utilização das redes neurais artificiais na segmentação de imagens.

4.1. Detecção de Materiais

Após exaustiva revisão da bibliografia, foi encontrado apenas um grupo de pesquisa na área de detecção de materiais em imagens: o projeto PERMM [9], apesar de seu objetivo principal ser a disponibilização de imagens para recuperação baseada em conteúdo (RIBC).

Para alcançar este objetivo, as imagens são primeiramente segmentadas – utilizando uma técnica de segmentação baseada em diagramas de Voronoi [14] – e posteriormente fornecidas a extratores de características que analisam informação de cor, textura, forma e conteúdo. Por fim, as características extraídas são utilizadas como entrada de redes neurais artificiais MLP que classificam semanticamente o conteúdo da imagem, ou seja, determinam que tipo de material está presente nas mesmas.

De acordo com a descrição do projeto, as suas redes neurais são capazes de detectar os seguintes materiais: pele, céu, grama, árvores, água, areia, tijolos, tecidos, paredes interiores, nuvens, neve, asfalto e madeira. Cada classificador utiliza diferentes combinações das características extraídas para classificar as imagens: os detectores de paredes interiores e tecidos, por exemplo, não se utilizam de informação de matiz quando analisando a informação de cor. Todos os classificadores têm acurácia entre 85% e 95%, mas não foram localizados valores exatos.

A principal diferença entre a técnica proposta neste trabalho e o projeto PERMM é que este utiliza vários segmentadores dedicados e previamente ajustados manualmente para cada classe de material. Na técnica deste trabalho, cada classificador é obtido automaticamente após a aplicação de um método sistemático.

4.2. Segmentação com Redes Neurais Artificiais

Redes Neurais Artificiais do tipo MLP são modelos matemáticos do cérebro humano, dedicadas à resolução de problemas de classificação [5]. Elas podem, portanto, ser aplicadas a problemas de detecção de materiais: cada *pixel* da imagem é classificado como sendo pertencente ou não à superfície de um dado material. Elas têm também a vantagem de serem classificadores que necessitam de pouca memória, o que as fazem boas candidatas para execução em dispositivos de recursos restritos, como PDAs, Máquinas Fotográficas, dentre outros [13].

Especialmente no domínio da detecção de pele, as redes neurais MLP já foram utilizadas com relativo sucesso em alguns trabalhos. Phung *et al.* [13] demonstraram, em seu *survey* de técnicas para detecção de pele, que a rede MLP foi o método mais preciso entre os demais – foram investigados também a Mistura Gaussiana (2D e 3D) e classificadores Bayesianos, dentre outros. Neste trabalho, a rede MLP foi treinada a partir de informação de cor extraída de

30.000 *pixels* de cada classe (pele e não-pele), no espaço de cores RGB. A acurácia máxima obtida foi de 89,49% (em uma base de teste diferente da usada neste trabalho).

Akbari e Nakajima [1] também adotaram redes neurais para detectar pele em fotografias a partir de informação de cor. Em seu estudo, eles propuseram uma técnica de homogeneização para ser utilizada após a classificação das redes neurais. O modelo neural adotado foi o *Learning Vector Quantization* (LVQ), e também considerava como entrada os valores de cada *pixel* no espaço de cores RGB. A camada de saída era composta de dois neurônios – um para determinar os *pixels* de pele e outro para os de não-pele. Seu classificador foi capaz de detectar corretamente 92,75% da área de pele (em uma base de testes diferente da utilizada neste trabalho).

5. Conclusão e Trabalhos Futuros

Este trabalho propõe uma técnica automática de obtenção e ajuste de segmentadores neurais de imagens, baseados em informação de cor. O objetivo desta segmentação é detectar determinados materiais – grama, pele, dentre outros – em imagens.

De acordo com os resultados obtidos, e comparando-os com outras técnicas mais tradicionais, a segmentação produzida com a utilização das redes neurais automaticamente treinadas tem alta qualidade. Desta forma a técnica apresentada contempla o seu objetivo de fornecer ao usuário segmentadores de alta qualidade, sem a necessidade do emprego de esforço na supervisão e ajuste dos mesmos.

Futuramente, técnicas de otimização global serão incorporadas no protótipo apresentado neste trabalho, com o objetivo de facilitar a descoberta da arquitetura ideal da rede neural artificial a ser utilizada para cada um dos materiais para os quais se deseja obter segmentadores. Assim, a exploração da quantidade de neurônios da camada escondida será restrita a menos valores, além de serem ampliadas as possibilidades no que diz respeito aos outros aspectos arquiteturais das redes neurais: número de camadas escondidas, conectividade entre neurônios, dentre outros.

Além disto, extratores de características de textura serão adicionados ao protótipo, fornecendo mais pistas sobre a pertinência de um *pixel* a determinado material. Com isto, surgirão oportunidades de automatização tanto para descoberta dos parâmetros do extrator de textura, quanto para descobrir a melhor combinação entre os classificadores de cor e textura para compor o resultado final da segmentação.

Referências

- [1] M. A. Akbari and M. Nakajima. A novel color region homogenization and its application in improving skin detection accuracy. In *GRAPHITE '05: Proceedings of the 3rd international conference on Computer graphics and interactive techniques in Australasia and South East Asia*, pages 269–272, New York, NY, USA, 2005. ACM.
- [2] F. H. Cardoso and H. M. Gomes. A probabilistic approach to skin detection. In *Anais do Simpósio Brasileiro de Computação Gráfica e Processamento de Imagens 2007*, 2007.
- [3] Firmino and H. M. Gomes. Detecção de lances no campo de ataque em uma partida de futebol. In *Anais do Simpósio Brasileiro de Computação Gráfica e Processamento de Imagens 2007*, 2007.
- [4] C. Garcia and G. Tziritas. Face detection using quantized skin color regions merging and wavelet packet analysis. *Multimedia, IEEE Transactions on*, 1(3):264–277, 1999.
- [5] S. Haykin. *Redes Neurais, Princípios e Práticas*. Editora Bookman, 2a edition, 2001.
- [6] N. Herodotou, K. N. Plataniotis, and A. N. Venetsanopoulos. *Image Processing Techniques for Multimedia Processing*. 2000.
- [7] R.-L. Hsu, M. Abdel-Mottaleb, and A. K. Jain. Face detection in color images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(5):696–706, 2002.
- [8] M. J. Jones and J. M. Rehg. Statistical color models with application to skin detection. *International Journal of Computer Vision*, 46(1):81–96, 2002.
- [9] A. Laboratories. Permm: Image analysis. Project Home Page, 2007.
- [10] E. O. Lima. Sumarização de vídeos de futebol com baixo custo computacional utilizando ritmo visual: descritores de cor e movimento de câmera. Master's thesis, UNIFOR, 2006.
- [11] R. Montufar-Chaveznaza, F. H. Gallardo, and S. P. Hernandez. Face detection by polling. In *Intelligent Signal Processing, 2005 IEEE International Workshop on*, pages 292–297, 2005.
- [12] Nusirwan, K. C. Wei, and J. See. Rgb-h-cbcr skin colour model for human face detection. In *Proceedings of The MMU International Symposium on Information & Communications Technologies (M2USIC 2006)*, 2006.
- [13] S. L. Phung, A. Bouzerdoum, and D. Chai. Skin segmentation using color pixel classification: Analysis and comparison. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(1):148–154, January 2005.
- [14] D. Sinclair. Voronoi seeded colour image segmentation. Technical report, AT&T Laboratories, 1999.
- [15] J. Wang, C. Xu, E. Chng, X. Yu, and Q. Tian. Event detection based on non-broadcast sports video. In *Image Processing, 2004. ICIP '04. 2004 International Conference on*, volume 3, pages 1637–1640 Vol. 3, 2004.
- [16] I. H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques, Second Edition (Morgan Kaufmann Series in Data Management Systems)*. Morgan Kaufmann, June 2005.

Contrast Enhancement in Digital Imaging using Histogram Equalization

David Menotti
Universidade Federal de Ouro Preto
Departamento de Computação
35.400-000, Ouro Preto-MG, Brazil
menottid@gmail.com

Laurent Najman
Université Paris-Est
UMR CNRS 8049 - A2SI-ESIEE
93162, Noisy-le-Grand, France
l.najman@esiee.fr

Arnaldo de A. Araújo and Gisele L. Pappa
Universidade Federal de Minas Gerais
Departamento de Ciência da Computação
31.270-010, Belo Horizonte-MG, Brazil
{arnaldo, glpappa}@dcc.ufmg.br

Jacques Facon
Pontifícia Universidade Católica do Paraná
Departamento de Informática
80.215-901, Curitiba-PR, Brazil
facon@ppgia.pucpr.br

Abstract

This work proposes two methodologies for fast image contrast enhancement based on histogram equalization (HE), one for gray-level images, and other for color images. For gray-level images, we propose a technique called Multi-HE, which decomposes the input image into several sub-images, and then applies the classical HE process to each one of them. In order to decompose the input image, we propose two different discrepancy functions, conceiving two new methods. Experimental results show that both methods are better in preserving the brightness and producing more natural looking images than other HE methods. For color images, we introduce a generic fast hue-preserving histogram equalization method based on the RGB color space, and two instantiations of the proposed generic method, using 1D and 2D histograms. HE is performed using shift hue-preserving transformations, avoiding the appearance of unrealistic colors. Experimental results show that the value of the image contrast produced by our methods is in average 50% greater than the value of contrast in the original image, still keeping the quality of the output images close to the original.

1. Introduction

Nowadays digital cameras are certainly the most used devices to capture images. They are everywhere, including mobile phones, personal digital assistants (PDAs - a.k.a. pocket computers or palmtop computers), robots, and surveillance and home security systems. There is no doubt

that the quality of the images obtained by digital cameras, regardless of the context in which they are used, has improved a lot since digital cameras early days. Part of these improvements are due to the higher processing capability of the systems they are built-in and memory availability. However, there are still a variety of problems which need to be tackled regarding the quality of the images obtained, including: 1) contrast defects; 2) chromatic aberrations; 3) various sources of noises; 4) vignetting; 5) geometrical distortions; 6) color demosaicing; and 7) focus defects.

Among the seven problems related above, some are more dependent on the quality of the capture devices used (like 2-7), whereas others are related to the conditions in which the image was captured (such as 1). When working on the latter, the time required to correct the problem on contrast is a big issue. This is because the methods developed to correct these problems can be applied to an image on a mobile phone with very low processing capability, or on a powerful computer.

Moreover, in real-time applications, the efficiency of such methods is usually favored over the quality of the images obtained. A fast method generating images with medium enhancement on image contrast is worth more than a slow method with outstanding enhancement.

With this in mind, this work¹ proposes two methodologies for contrast enhancement in digital imaging using histogram equalization (HE)². Although there has been a lot of

¹ This paper comes from the doctoral thesis of David Menotti [10], submitted to the Department of Computer Science, Universidade Federal de Minas Gerais, in April 2008.

² In this work, the contrast is defined as the standard variation of the image gray-levels or luminance.

research in the image enhancement area for 40 years [5, 10], there is still a lot of room for improvement concerning the quality of the enhanced image obtained and the time necessary to obtain it.

HE is a histogram specification process [3] which consists of generating an output image with a uniform histogram (*i.e.*, uniform distribution). In image processing, the idea of equalizing a histogram is to stretch and/or redistribute the original histogram using the entire range of discrete levels of the image, in a way that an enhancement of image contrast is achieved. HE is a technique commonly used for image contrast enhancement, since it is computationally fast and simple to implement. Our main motivation is to preserve the best features the HE methods have, and introduce some modifications which will overcome the drawbacks associated with them.

In the case of **gray-level image contrast enhancement**, methods based on HE have been the most used. Despite its success for image contrast enhancement, this technique has a well-known drawback: it does not preserve the brightness³ of the input image on the output one. This problem makes the use of classical HE techniques [5] not suitable for image contrast enhancement on consumer electronic products, such as video surveillance, where preserving the input brightness is essential to avoid the generation of non-existing artifacts in the output image [14, 10].

In order to overcome this problem, variations of the classic HE technique, such as [6, 22, 2, 1], have proposed to first decompose the input image into two sub-images, and then perform HE independently in each sub-image (Bi-HE). These works mathematically show that dividing the image into two rises the expectance of preserving the brightness. Although Bi-HE successfully performs image contrast enhancement and also preserves the input brightness to some extent, it might generate images which do not look as natural as the input ones. Unnatural images are unacceptable for use in consumer electronics products [14, 10].

Hence, in order to enhance contrast, preserve brightness and produce natural looking images, we propose a generic Multi-HE (MHE) method that first decomposes the input image into several sub-images, and then applies the classical HE process to each of them. We present two discrepancy functions to decompose the image, conceiving two variants of that generic MHE method for image contrast enhancement, *i.e.*, Minimum Within-Class Variance MHE (MWCVMHE) and Minimum Middle Level Squared Error MHE (MMLSEMHE). Moreover, a cost function, which takes into account both the discrepancy between the input and enhanced images and the number of decomposed sub-images, is used to automatically determine in how many

sub-images the input image will be decomposed on.

Regarding **color image contrast enhancement**, the classical methods are also based on HE. The extension of HE methods to color images is not straightforward, because there are some particular properties of color images that need to be properly taken into account during image contrast enhancement. These properties include the luminance (L) (or intensity (I)), saturation (S), and hue (H) attributes of the color.

The luminance represents the achromatic part of the color (*e.g.*, it can be defined as a weighted function of the R (red), G (green), and B (blue) color channels), whereas the saturation and hue refer to the chromatic part of the image. The saturation can be seen as measure of how much white is present in the color, and the hue is the attribute of the color which decides its “real color”, *e.g.*, red or green. For the purpose of enhancing a color image, the hue should not be changed for any pixel, avoiding output images with unnatural aspect.

Color spaces such as *HSV*, *HSI*, *CIELUV*, and *CIELAB* were conceived based on these three attributes. However, color images in digital devices, such as mobile phones, cameras, and PDAs, are commonly transmitted, displayed, and stored in the *RGB* color space (*i.e.*, R -red, G -green, and B -blue). This color space is not the most appropriated one for image processing tasks, since the meaning of the attribute colors is not explicitly separated as it would be in other color spaces. The conversion from the *RGB* color space to a Luminance-Hue-Saturation (*LHS*)-based color space is trivial, but can be both not suitable for real-time applications and the digital devices referred above. Moreover, working on a *LHS*-based color space requires tackling the well-known gamut problem [16].

The literature of HE methods for color image contrast enhancement presents works based on the *RGB*, *LHS*, *CIELUV*, and other color spaces. Neither methods based on the *RGB* color space nor methods based on other color spaces present all the characteristics required for use in portable devices: to be fast, improve the images contrast and still preserve the hue. Methods based on the *RGB* space do not preserve the hue, while methods based on other color spaces are slower due to conversions required among color spaces and may also be not hue-preserving. In order to achieve all these three requirements, this work presents a generic fast hue-preserving HE method based on the *RGB* color space for image contrast enhancement.

From the generic method we create two variants, which are characterized by the histograms dimension they use, *i.e.*, $1D$ or $2D$. The equalization is performed by hue-preserving transformations directly in the *RGB* color space, avoiding the gamut problem, keeping the hue unchanged, and the requirement of conversion between color spaces. Moreover,

3 In this work, the brightness is defined as the mean of the image gray-levels

our methods improve the image contrast (*i.e.*, improve the variance on the luminance attribute) and, simultaneously, the saturation is modified according to the equalization of the RGB histogram. The methods estimate the *RGB* 3D histogram to be equalized through *R*, *G*, and *B* 1D histograms and *RG*, *RB*, and *GB* 2D histograms, respectively, yielding algorithms with time and space complexities linear with respect to the image dimension. These characteristics make these methods suitable for real-time applications.

The remainder of this work is organized as follows. Section 2 present the Multi-HE methods for gray-level images, whilst Section 3 introduces our fast hue-preserving HE for color images. Section 4 shows some experimental results, and finally conclusions are pointed out in Section 5.

2. Multi-Histogram Equalization Methods for Contrast Enhancement and Brightness Preserving

As mentioned before, the classic HE method enhances the contrast of an image but cannot preserve its brightness (which is shifted to the middle gray-level value). As a result, it can generate unnatural and non-existing objects in the processed image. In contrast, Bi-HE methods [6, 22, 2, 1] can produce a significant image contrast enhancement and, to some extent, preserve the brightness of the image. However, the generated images might not have a natural appearance [14, Figure 1]. To surmount such drawbacks, the main idea of our proposed methods is to decompose the image into several sub-images, such that the image contrast enhancement provided by the HE in each sub-image is less intense, leading the output image to have a more natural look. The conception of this method arises two questions.

The first question is how to decompose the input image. As HE is the focus of the work, the image decomposition process is based on the histogram of the image. The histogram is divided into classes determined by threshold levels, where each histogram class represents a sub-image. The decomposition process can be seen as an image segmentation process executed through multi-thresholding selection [7]. The second question is in how many sub-images an image should be decomposed on. This number is directly related to how the input image is decomposed.

In order to answer these questions, Section 2.1 presents two functions to decompose an image based on threshold levels, whereas the algorithm used to find the optimal threshold levels is presented in Section 2.2. Finally, a criterion for automatically select the number of decomposed sub-images is exposed in Section 2.3.

Note that the methods described in this section are published in [14].

2.1. Multi-Histogram Decomposition

Many HE-based methods have been proposed in the literature to decompose an image into sub-images by using the value of some statistical measure based on the image gray-level [6, 22, 2, 1]. These methods aim to optimize the entropy or preserve the brightness of the image. Here, we will focus our attention on decomposing an image such that the enhanced images still have a natural appearance. For such aim, we propose to cluster the histogram of the image into classes, where each class corresponds to a sub-image. By doing that, we want to minimize the brightness shift yielded by HE process into each sub-image. By minimizing this shift, we expect to preserve both the brightness and the natural appearance of the processed image.

From the multi-threshold selection literature point of view, the problem stated above can be seen as the minimization of the within-histogram class variance (the well-know Otsu method [18]), where the within-class variance is the total squared error of each histogram class with respect to its mean value (*i.e.*, the brightness). That is, the decomposition aim is to find the optimal threshold set $T^k = \{t_1^k, \dots, t_{k-1}^k\}$ that minimizes the decomposition error of the histogram of the image into k histogram classes, and decomposes the image $I[0, L-1]$ into k sub-images $I[l_s^{1,k}, l_f^{1,k}], \dots, I[l_s^{k,k}, l_f^{k,k}]$. $l_s^{j,k}$ and $l_f^{j,k}$ are the lower and upper gray-level boundaries of each sub-image j when the image is decomposed into k sub-images, and are defined as: $l_s^{j,k} = t_{j-1}^k$, if $j > 1$, and $l_s^{j,k} = 0$ otherwise, and $l_f^{j,k} = t_j^k + 1$, if $j \neq k$, and $l_f^{j,k} = L-1$ otherwise. The discrepancy function for decomposing the original image into k sub-images following the minimization of within-class variance can be expressed as

$$Disc(k) = \sum_{j=1}^k \sum_{l=l_s^{j,k}}^{l_f^{j,k}} (l - l_m(I[l_s^{j,k}, l_f^{j,k}]))^2 P_l^{I[0, L-1]}. \quad (1)$$

The method conceived with this discrepancy function will be called Minimum Within-Class Variance MHE method (MWCVMHE). Note that the mean gray-level (*i.e.*, the brightness) of each sub-image processed by the CHE method is theoretically shifted to the middle gray-level of its range, *i.e.*, $l_m(O[l_s, l_f]) = l_{mm}(I[l_s, l_f]) = l_{mm}(O[l_s, l_f]) = (l_s + l_f)/2$. As we want to minimize the brightness shift of each processed sub-image such that the global processed image has its contrast enhanced and its brightness preserved (creating a natural looking output image), we focus our attention on the brightness of the output image. Hence, instead of using the mean $l_m(I[l_s, l_f])$ of each sub-image $I[l_s, l_f]$ in the discrepancy function, we propose to use its middle level $(l_s + l_f)/2$, since every enhanced sub-image

$O[l_s, l_f]$ will theoretically have its mean value (brightness) on the middle level of the image range - thanks to the specification of a uniform histogram distribution. Therefore, a new discrepancy function is proposed and it is expressed as

$$Disc(k) = \sum_{j=1}^k \sum_{l=l_s^{j,k}}^{l_f^{j,k}} (l - l_{mm}(I[l_s^{j,k}, l_f^{j,k}]))^2 P_l^{I[0, L-1]}, \quad (2)$$

where $l_{mm}(I[l_s^{j,k}, l_f^{j,k}])$ stands for the middle value of the image $I[l_s^{j,k}, l_f^{j,k}]$ and it is defined as $|(l_s + l_f)/2|$. The method conceived with this discrepancy function will be called Minimum Middle Level Squared Error MHE method (MMLSEMHE).

2.2. Finding the Optimal Thresholds

The task of finding the optimal $k - 1$ threshold levels which segment an image into k classes can be easily performed by a dynamic programming algorithm with $O(kL^2)$ time complexity [7]. Algorithm 1 shows the pseudocode of this algorithm, where $\varphi(p, q)$ stands for the “discrepancy contribution” of the sub-image $I[p, q]$, i.e.,

$$\varphi(p, q) = \sum_{l=p}^q (l - \gamma)^2 P_l^{I[0, L-1]}, \quad (3)$$

where γ stands for $l_m(I[p, q])$ or $l_{mm}(I[p, q])$, depending on the discrepancy function used (see Equations 1 and 2).

Once Algorithm 1 is run, the optimal threshold vector T^k can be obtained through a back-searching procedure on PT , i.e.,

$$t_j^k = PT(j + 1, t_{j+1}^{k*}), \quad (4)$$

where $1 \leq j < k$, $t_{j+1}^{k*} = L - 1$ if $j + 1 = k$, and $t_{j+1}^{k*} = t_{j+1}^k$ otherwise.

2.3. Automatic Thresholding Criterion

This section presents an approach to automatically choose in how many sub-images the original image should be decomposed on. This decision is a key point of our work, which has three main aims: 1) contrast enhancement; 2) brightness preserving; 3) natural appearance. Nonetheless, these goals cannot be all maximize simultaneously. We take into account that as the number of sub-images in which the original image is decomposed increases, the chance of preserving the image brightness and natural appearance also increases. However, the chances of enhancing the image contrast decrease. To decide in how many sub-images the original image should be decomposed on, a tradeoff between brightness, natural appearance and contrast should be considered. Hence, we

Algorithm 1: Computing $Disc(k)$ and $PT(k, L - 1)$

Data: $\varphi(p, q)$ - discrepancy of sub-image $I[p, q]$

Result: $D(p)_q$ - discrepancy function $Disc(p)$ up to level q

Result: PT - optimum thresholds matrix

```

1 for  $q \leftarrow 0 ; q < L ; q++$  do  $D(1)_q \leftarrow \varphi(0, q) ;$ 
2 for  $p \leftarrow 1 ; p \leq k ; p++$  do
3    $D(p+1)_p \leftarrow D(p)_{p-1} + \varphi(p-1, p-1) ;$ 
4    $PT(p+1, p) \leftarrow p-1 ;$ 
5   for  $q \leftarrow p+1 ; q \leq L-k+p ; q++$  do
6      $D(p+1)_q \leftarrow -\infty ;$ 
7     for  $l \leftarrow p-1 ; l \leq q-1 ; l++$  do
8       if  $(D(p+1)_q > D(p)_l + \varphi(l+1, q))$  then
9          $D(p+1)_q \leftarrow D(p)_l + \varphi(l+1, q) ;$ 
10         $PT(p+1, q) \leftarrow l ;$ 

```

propose to use a cost function, initially used in [23], to automatically select the number of decomposed sub-images. This cost function takes into account both the discrepancy between the original and processed images (which is our own aim decomposition function) and the number of sub-images to which the original image is decomposed, and it is given as

$$C(k) = \rho(Disc(k))^{1/2} + (\log_2(k))^2, \quad (5)$$

where ρ is a positive weighting constant. The number of decomposed sub-images k is automatically given as the one which minimizes the cost function $C(k)$. It is shown in [23] that the cost function presented in Equation 5 has a unique minimum. Hence, instead of finding the value k which minimizes $C(k)$ throughout k values range, it is enough to search for k from 0 up to the point $C(k)$ starts to increase.

3. Fast Hue-Preserving Histogram Equalization Methods for Color Image Contrast Enhancement

This section presents a generic method that, in contrast with the classical method presented in [5] (from now on C1DHE method) and the one in [20] (from now on TV3DHE method), is both hue-preserving and has time and space complexities which complies with real-world and real-time applications. We propose two variants of this generic method, which are characterized by the histogram dimensions used to estimate the 3D probability functions, i.e., 1D or 2D histograms.

3.1. Generic Hue-preserving Histogram Equalization Method

Our generic hue-preserving HE method is divided in three phases. Let I and O be the input and output images, respectively. Let the input $\#D$ histograms and probability functions be defined as in [15, Section 2] and [10, Section 4.1] (omitted due to space constraints), where $\#$ is the histogram dimension used (the variant point of our method). The first phase of our method consists of computing the $\#D$ histograms of I . Although the proposed method works with $\#D$ histograms and probability functions, we do not equalize the $\#D$ histograms per say, but a $3D$ pseudo-histogram, $H^{I^{RGB}}$. Indeed, the equalization of $H^{I^{RGB}}$ is based on a pseudo $3D$ cumulative density function, built through probability density functions.

The computation of this cumulative density function, $C^{I^{RGB}}$, which constitutes the second phase of our method, is performed as the product of the three $\#D$ cumulative functions. We show in details the variant methods in Sections 3.2 and 3.3.

The third phase works as follows. Let $H^{O^{RGB}}$ be the uniform histogram of the output image, where any entry (R_o, G_o, B_o) has the same amount of pixels, since such output histogram is desired, *i.e.*,

$$H_{R_o, G_o, B_o}^{O^{RGB}} = \frac{1}{L^3}(mn), \quad (6)$$

or any entry (R_o, G_o, B_o) in $P^{O^{RGB}}$ has the same density, *i.e.*,

$$P_{R_o, G_o, B_o}^{O^{RGB}} = 1/L^3. \quad (7)$$

Hence, any entry (R_o, G_o, B_o) in $C^{O^{RGB}}$ is directly computed using $P^{O^{RGB}}$, *i.e.*,

$$C_{R_o, G_o, B_o}^{O^{RGB}} = (R_o + 1)(G_o + 1)(B_o + 1)/L^3. \quad (8)$$

To yield the output enhanced image, for every input pixel $(x, y) \in X$, where $(R_i, G_i, B_i) = I^{RGB}(x, y)$, we obtain the smallest $(R_o, G_o, B_o) = O^{RGB}(x, y)$ for which the inequality

$$|C_{R_i, G_i, B_i}^{I^{RGB}} - C_{R_o, G_o, B_o}^{O^{RGB}}| \geq 0, \quad (9)$$

holds.

However, this step of calculating the output pixel value presents an ambiguity, mainly because there are many possible solutions for (R_o, G_o, B_o) which satisfy Equation 9. This ambiguity is remedied as follows. Unlike the method described in [20] (TV3DHE method), which iteratively increased or decreased the values of R_o , G_o , and B_o in order to minimize Equation 9, we propose to find the output triplet (R_o, G_o, B_o) for any image pixel in a single step, *i.e.*, $O(1)$. Thus, from Equations 8 and 9, we have

$$C_{R_i, G_i, B_i}^{I^{RGB}} - \frac{(R_o + 1)(G_o + 1)(B_o + 1)}{L^3} = 0. \quad (10)$$

If we take R_o , G_o , and B_o as $R_i + k$, $G_i + k$, and $B_i + k$, respectively, where k is the number of iterations required for minimizing Equation 9, we obtain

$$\begin{aligned} & k^3 + \\ & k^2[R'_i + G'_i + B'_i] + \\ & k[R'_i \times G'_i + R'_i \times B'_i + G'_i \times B'_i] + \\ & R'_i \times G'_i \times B'_i - L^3 \times C_{R_i, G_i, B_i}^{I^{RGB}} = 0. \end{aligned} \quad (11)$$

where R'_i , G'_i , and B'_i mean $R_i + 1$, $G_i + 1$, and $B_i + 1$, respectively. By solving this cubic equation in function of k , we obtain the desired output triplet (R_o, G_o, B_o) as the input one plus the displacement k , *i.e.*, $(R_i + \langle k \rangle, G_i + \langle k \rangle, B_i + \langle k \rangle)$, where $\langle k \rangle$ stands for the nearest integer to $k \in \mathbb{R}$.

Equation 11 can be easily solved by [17] or by the classical Cardan's methods which use transcendental functions. As the former method is faster and mathematically simpler than the latter, we chose to use it.

Observe that any image pixel is enhanced following a shift transformation by a k factor, *i.e.*, from (R_i, G_i, B_i) to $(R_o, G_o, B_o) = (R_i + k, G_i + k, B_i + k)$, which makes our generic method hue-preserving [16].

Having described this generic method, the next subsections show our variant methods, which differ only on the histogram dimension used. By respecting the chronology's conception of our methods, the method based on RG , RB , and GB $2D$ histograms [12] (from now on HP2DHE method), is described first in Section 3.2. Then, the method based on $1D$ histograms [13] (from now on HP1DHE method) is presented in Section 3.3.

3.2. Hue-preserving 2D Histogram Equalization

In this section, we present our HP2DHE method, initially introduced in [12]. It uses $2D$ histograms (as defined in [15, Section 2] and [10, Section 4.1]) and is based on the joint probability distribution functions of channels two-at-a-time to perform HE. The cumulative probability density function (or the probability distribution function), $C_{R_i, G_i, B_i}^{I^{RGB}}$, is computed as the product of the three $2D$ cumulative functions for any entry (R_i, G_i, B_i) , *i.e.*,

$$C_{R_i, G_i, B_i}^{I^{RGB}} = C_{R_i, G_i}^{I^{RG}} C_{R_i, B_i}^{I^{RB}} C_{G_i, B_i}^{I^{GB}}. \quad (12)$$

For details on how to directly calculate $C_{R_i, G_i}^{I^{RG}}$, $C_{R_i, B_i}^{I^{RB}}$, and $C_{G_i, B_i}^{I^{GB}}$ through the probability density function $P_{R_i, G_i}^{I^{RG}}$, $P_{R_i, B_i}^{I^{RB}}$, and $P_{G_i, B_i}^{I^{GB}}$ see [15, Section 2] or [10, Section 4.1]. The main rationale for computing this pseudo-cumulative density function as the product of three $2D$ cumulative density functions is that the three channels in an image are usually not simultaneously correlated.

Image	HE	BBHE	DSIHE	MMBEBHE	BPHEME	RMSHE	MWCVME	MMLSEMHE
arctichare	8.11	16.63	13.09	23.55	22.95	30.74	31.44	40.27
bottle	12.88	18.68	17.53	28.44	25.72	29.68	35.99	36.71
copter	10.61	15.95	14.20	25.50	23.20	25.62	33.83	34.77
couple	7.57	13.18	11.61	19.86	38.54	19.65	30.59	40.16
Einstein	15.08	15.15	15.58	18.91	16.21	19.51	31.42	34.53
F16	11.92	20.69	16.02	20.32	21.61	22.72	24.43	37.10
girl	13.03	13.30	12.99	14.03	13.19	28.00	29.39	33.03
hands	4.36	19.58	17.76	19.99	17.18	30.93	24.49	35.82
house	10.82	14.27	14.07	21.41	19.93	21.36	31.81	36.37
jet	9.51	22.50	14.37	30.78	23.99	27.85	29.14	31.74
U2	6.99	15.06	10.94	19.87	27.32	22.12	26.21	31.08
woman	17.83	17.73	18.25	21.60	19.23	23.67	28.83	34.53

Table 1. $PSNR = 10 \times \log_{10} (L - 1)^2 / MSE$

Note that, in [13], we proposed to solve Equation 9 iteratively, as done in [20], by using a non hue-preserving transformation. Here, we modify the method originally proposed in [13] to use the hue-preserving shift transformation and the solution of Equation 9 described in the previous subsection. These two modifications make the HP2DHE method presented here hue-preserving, and reduces its time complexity from $O(\max(mnL, L^2))$ to $O(\max(mn, L^2))$.

3.3. Hue-preserving 1D Histogram Equalization

In this section, we present a hue-preserving HE method based on the RGB color space for image contrast enhancement, which uses 1D histograms, and is also a variant of the generic method described in Section 3.1. The method is based on the independence assumption of color channels, which is used for computing the cumulative density function.

We use 1D histograms to estimate a 3D probability distribution function (or a cumulative density function), and then equalize the conceived histogram through the estimated function. Hence, the function $C^{I^{RGB}}$ is estimated for any entry (R_i, G_i, B_i) as the product of every probability distribution function $C_{R_i}^{I^R}$, $C_{G_i}^{I^G}$, and $C_{B_i}^{I^B}$, following the rule, *i.e.*,

$$C_{R_i, G_i, B_i}^{I^{RGB}} = C_{R_i}^{I^R} C_{G_i}^{I^G} C_{B_i}^{I^B}. \quad (13)$$

For details on how to directly calculate C^{I^R} , C^{I^G} , and C^{I^B} through P^{I^R} , P^{I^G} , and P^{I^B} see [15, Section 2] or [10, Section 4.1]. Note that, in Equation 13, $C^{I^{RGB}}$ is defined with a correct dimensional meaning, *i.e.*, $C^{I^{RGB}}$, a 3D cumulative function, is computed as the product of three 1D cumulative functions, while in Equation 12 $C^{I^{RGB}}$ is defined with a wrong dimensional meaning, *i.e.*, $C^{I^{RGB}}$

is computed as the product of three 2D cumulative functions. Nevertheless, the images processed by the HP2DHE method produce similar results to the HP1DHE method, as the experiments reported in Section 4.

As we use 1D histograms, this method has a smaller time complexity than the HP2DHE method, *i.e.*, $O(\max(mn, L))$, and the space complexity is linear, *i.e.*, $O(L)$. Moreover, the time and space complexities of HP1DHE are exactly the same of the C1DHE method, which are the best to our knowledge.

A complete description of the methods presented in this section can be found in [15, 10].

4. Experiments

In this section, we report experiments performed to compare and evaluate the methods proposed in Sections 2 and 3. Section 4.1 presents the experiments involving methods proposed for handling gray-level images, and Section 4.2 analyzes and discusses the experimental results concerning the methods proposed for handling color images.

4.1. Experiments with Gray-level Images

In this section, we report results of experiments comparing our proposed methods with other HE methods (HE, BBHE, DSIHE, MMBEBHE, and RMSHE ($r = 2$)) and the method proposed in [21]. The input images used in the experiments were the ones previously used in [6, 22, 2, 1, 21]. They are named as they were in the works where they first appeared: arctic hare, bottle, copter, couple, Einstein, F16, girl, hands, house, jet, U2, woman (girl in [21]). Images were extracted from the CVG-UGR database [4] and provided by the authors of [2, 1].

Besides an analyzes of brightness (the mean) and contrast (the standard deviation) values of the original and out-

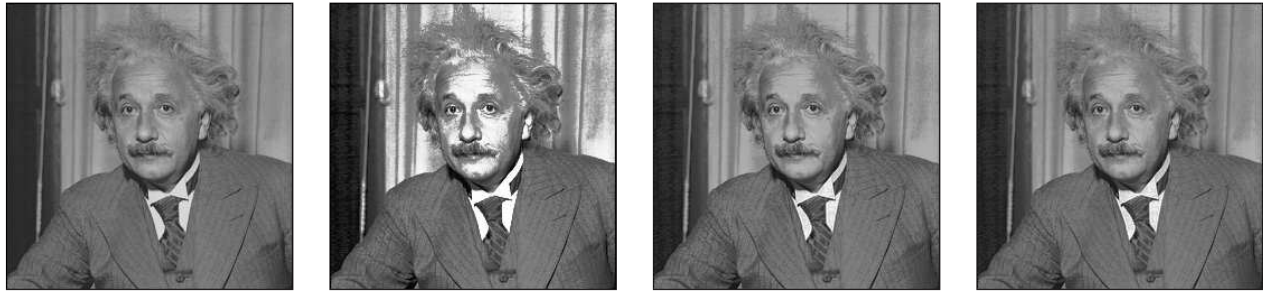


Figure 1. From left to right: the Einstein original, enhanced RMSHE ($r = 2$), MWCVMHE ($k = 6$), and MMLSEMHE ($k = 7$) images.

put images, in order to assess the appropriateness of the processed images for consumer electronics products, for each image, we compute the *PSNR* measure [19]. In image processing literature, the *PSNR* has been used as a standard measure to evaluate and compare compression and segmentation algorithms [19]. It is well-known that a processed image with good quality (with respect to the original one) presents *PSNR* values within 30 db and 40 db [19].

Due to space constraints, the analysis of brightness and contrast of the original and the output images obtained by the HE methods was omitted and it is presented in [14, 10]. The values of *PSNR* obtained for each image are presented in Table 1. This table is divided into three parts: 1) The names of original images; 2) The data values obtained by the Uni- and Bi-HE methods, *i.e.*, HE, BBHE, DSIHE, MMBEBHE, and BPHEME; 3) The values obtained by the MHE methods, *i.e.*, RMSHE ($r = 2$), and our proposed MWCVMHE and MMLSEMHE.

For each image in Table 1, we highlight the best data values in the second and third parts of the table in either dark or light gray. We then compare these best values in the second and third parts of the table against each other (*i.e.*, Uni- and Bi-HE methods against MHE methods). The best value is dark-grayed, the worst light-grayed. Recall that the greater the value of the *PSNR*, the better it is.

Analyzing the data presented in Table 1, we observe that the images processed by the MMLSEMHE method produces the best *PSNR* values, as they are within the range [30 db, 40 db]. Hence, we can argue that the MMLSEMHE method performs image contrast enhancement and brightness preserving while still producing images with a natural looking. Moreover, this result corroborates, in practice, our hypothesis that the MMLSEMHE method, using the discrepancy function in Equation 2, yields images with the best *PSNR* values among all HE methods.

Besides this *PSNR* analysis, we also perform an image visual assessment. Remark that all the 12 input images, their histograms, their respective enhanced images and equalized

histograms (obtained by all the method listed in Table 1), adding up more than 200 images, can be seen in [9]. Here, we show the images obtained by the image Einstein.

Figure 1 shows the Einstein image and the resulting images obtained by the MHE methods, *i.e.*, RMSHE (with $r = 2$), MWCVMHE, and MMLSEMHE. By observing the processed images, it is noticeable that our proposed methods are the only ones among the MHE methods that can produce natural looking images.

After analyzing the data presented in Table 1 and visually observing the processed images, we can conclude that the MMLSEMHE method produces images with better quality than the other methods with respect to the *PSNR* measure. By further analyses made in [14, 10], we can also conclude that: 1) A better image contrast enhancement can be obtained by the MWCVMHE method, which also presents satisfactory brightness preserving and natural looking images; 2) The RMSHE method ($r = 2$) should be employed if even more contrast enhancement than offered by the the MMLSEMHE and MWCVMHE methods is desired. However, in this case, the processed image may present some annoying and unnatural artifacts (for instance Figure 1-RMSHE ($r = 2$)).

4.2. Experiments with Color Images

The majority of image enhancement methods found in the literature, including our previous works [12, 13], assesses the contrast improvement of the output image by visually comparing it to the original one. In [12, 13], we claimed that it is difficult to judge a processed enhanced image using a subjective assessment. Hence, in this work, we use two types of quantitative measures to assess the original and processed images produced by the C1DHE and TV3DHE method and ours (presented in Section 3), and then perform an objective comparison among them.

The first quantitative measure used is a color image quality measure (CIQM) [24], defined by the image color nat-

Method	L^*	L^{RGB}		Q		CNI		CCI	
Original	12.53 ± 3.98	31.13	± 9.90	0.68	± 0.02	0.81	± 0.03	0.80	± 0.12
C1DHE	18.38 ± 3.78	47.11	± 9.76	0.68	± 0.01	0.78	± 0.03	1.03	± 0.13
HP1DHE	18.14 ± 3.71	46.73	± 9.61	0.66	± 0.02	0.78	± 0.04	0.78	± 0.07
HP2DHE	18.55 ± 3.91	47.02	± 10.01	0.67	± 0.02	0.78	± 0.04	0.91	± 0.10
TV3DHE	13.30 ± 2.89	36.44	± 7.72	0.58	± 0.02	0.72	± 0.02	0.49	± 0.05

Table 2. Contrast for the images in the $CIELUV$ and RGB color spaces and Color Image Quality Measures

uralness and colorfulness indexes, and applied to verify if the HE methods preserve the quality of the images. The second measure refers to the contrast in the $CIELUV$ and in the RGB color spaces, and aims to show how much the HE methods improve the contrast of the original image.

This section presents and discusses the numerical results obtained by using the CIQMs and the contrast measure above mentioned and detailed described in [15, Section 5.1] and [10, Section 5.2.2.1] to evaluate our proposed methods (HP1DHE and HP2DHE) and the others (C1DHE and TV3DHE) in a data set of 300 images taken from the University of Berkeley [8].

We compute, for both the original and the processed images, the contrast in both the $CIELUV$ and RGB color spaces and the CIQMs, as showed in Table 2. Table 2 is divided in three parts. In the first part, we present the image source name, *i.e.*, the original or the methods that originated the image. In the second and third parts, we show the values obtained for the contrast and CIQMs, respectively. Note that the values in the table are presented in the form $\mu \pm \sigma$, *i.e.*, the mean and standard deviation of the measures computed on the data set of 300 images. All images used in this experiment can be seen in [11].

From the second part of Table 2, we observe that the images processed by our methods, *i.e.*, HP1DHE and HP2DHE, have the value of contrast increased, in average, about 50% in both the $CIELUV$ and RGB color spaces. The values of the contrast of images processed by the C1DHE method also increase in a similar fashion. In contrast, the TV3DHE method is the one that increases the less the contrast. Remark that, in general, the improvement of the value of contrast in the $CIELUV$ color space is proportional to the one in the RGB space (the range of the $CIELUV$ luminance is $[0, 100]$ and the RGB luminance is $[0, 255]$ (with $L = 256$)). From this first analysis, we state that our methods and the C1DHE are effective in yielding significant increasing in the value of image contrast.

In the third part of Table 2, we find the Q , CNI , and CCI measures for the original and processed images. Note that the third numerical column in this table reports the Q measure values, which are a weighting function of the CNI and CCI measures. We observe that, in average, the im-

ages processed by our methods have preserved values of Q in the processed images close to the value in the original ones. This means that our methods produce images with quality similar to the original images. Also note that the images enhanced by the C1DHE method have obtained similar Q values to the ones obtained by our methods. In contrast, the images produced by the TV3DHE method have Q values quite smaller than the ones calculated from the original images. This shows that the TV3DHE method yields images with deteriorated color quality.

On the fourth numerical column of Table 2, we have the values for the CNI measure. Observe that, in average, our methods and the C1DHE keep the naturalness of the produced images close to the one in the original image, whereas the images produced by the TV3DHE method have CNI values significantly smaller than the ones obtained from the original images.

On the fifth numerical column of Table 2, we report the values for the CCI measure. Observe that the CCI measure is based on the mean and standard deviation of the saturation of the image in the $CIELUV$ color space. The results reported show that, in average, the C1DHE method is the one that more frequently increases the value of the CCI measure from the original to the processed images. The C1DHE method achieves such result because it equalizes the three R , G , and B 1D histograms freely and separately. On the other hand, it has the well-known drawback of not being hue-preserving, which will be discussed and illustrated further in this section. The images produced by the TV3DHE method, in average, do not preserve both the CNI and CCI values and, consequently the Q value, close to the values of the original images. The fact that the TV3DHE method produces images with CCI values quite different from the ones in the original images corroborates the hypothesis subjectively stated in [12] and [13] that the TV3DHE method produces overenhanced / under-saturated images (*i.e.*, brighter images). That is, in general the saturation values of the images produced by the TV3DHE method are smaller than the saturation values of the images produced by the other methods, and so are their variances.

From the analysis regarding the contrast and the CIQMs, we claim that: 1) The contrast of the images processed by our methods is in average 50% greater than the contrast of

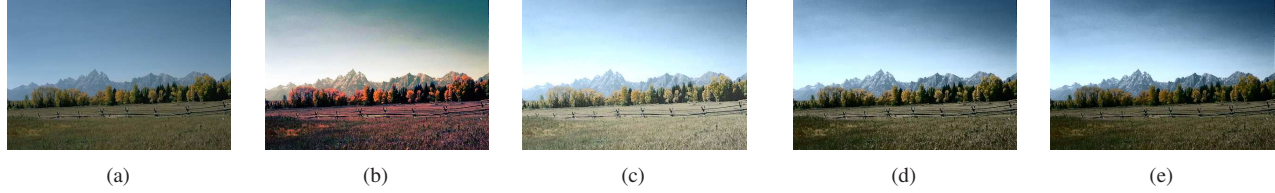


Figure 2. Results for the landscape image: (a) original image; (b) $C1DHE$; (c) $TV3DHE$; (d) our $HP1DHE$; (e) our $HP2DHE$.

Method	Color Quality			Contrast	
	Q	CNI	CCI	$CIELUV$	RGB
Original	0.7038	0.8540	0.7196	7.00	17.03
$C1DHE$	0.7681	0.9292	0.8089	12.09	30.32
$HP1DHE$	0.7210	0.8725	0.7575	11.50	28.98
$HP2DHE$	0.6504	0.7688	0.8381	11.00	27.59
$TV3DHE$	0.7140	0.9004	0.4392	8.76	23.68

Table 3. Color Image Quality and Contrast Measures for the Images in Figure 2.

the original images, whilst the color quality, measured by the naturalness and colorfulness indexes, of the processed images are close to the ones of the original image; 2) The $TV3DHE$ method is the one that show the smaller improvement on the contrast of the original image. Moreover, it produces images overenhanced, deteriorating the color quality of the images; 3) The results achieved for contrast enhancement and color quality preservation by the $C1DHE$ method are as good as our methods.

Note that we could perform changes in the $TV3DHE$ method in order to make it faster and hue-preserving, by applying our shift hue-preserving transform. Nonetheless, even after these modifications, the images enhanced by the $TV3DHE$ method would continue to be overenhanced and the contrast improvement would not be significant.

Despite the good results that our numerical analysis attributed to the $C1DHE$ method, and the fact that it is six times faster than our methods, the $C1DHE$ is not suitable for real-world applications: the images produced by this method do not preserve the hue of the original image. As a result, the images produced by the $C1DHE$ method may have unnatural colors, even though the CNI , CCI , and, consequently, Q , indicate that the images produced by the $C1DHE$ method have image color quality close to the ones of the original images. These contradictory results show that the CQIMs used in this work have a drawback. They can quantitatively represent the color quality of a image by means of the naturalness and colorfulness indexes, but they do not take into account simultaneously the original and processed images in such assessment.

In order to exemplify the conclusions reached, we will carefully analyze one example of an image extracted from the 300 presented in the data base, named “landscape”. Table 3 shows the contrast and the CNI , CCI , and Q values for the original and processed landscape images in Figure 2. Figure 2(b) shows the landscape image processed by the $C1DHE$ method, and highlights the fact that it is not hue-preserving. We observe that the colors present in the image in Figure 2(b) look unnatural with respect to the original image in Figure 2(a), even though the CNI , CCI , and Q values of the processed image are close to the ones in the original image. We can also observe that the image produced by the $TV3DHE$ method in Figure 2(c) is overenhanced, *i.e.*, the colors are undersaturated, as explained before in this section. Moreover, we can see that the increase in the value of the image contrast produced by the $TV3DHE$ method is the smallest among the compared methods, as shown in Table 3.

Finally, the claims about our methods are verified in the images in Figures 2(d) and 2(e) and confirmed in Table 3. As observed, the images have their contrast value increased by, in average, 50%, while their color quality measures are kept close to the ones of the original image. Besides, our methods are hue-preserving.

5. Conclusions

In the first part of this work, we proposed and tested a new framework called the MHE for image contrast enhancement and brightness preserving which generated natural looking images. The experimental results showed that our methods are better at preserving the brightness of the processed image (in relation to the original one) and yield images with natural appearance, at the cost of contrast enhancement. The contributions of this part of the work are threefold: 1) An objective comparison among all the HE methods using quantitative measures such as the $PSNR$, brightness, and contrast (the comparison of these last two measures can be seen in [14, 10]); 2) An analysis showing the boundaries of the HE technique and its variations (*i.e.*, Bi- and Multi-HE methods), for contrast enhancement,

brightness preserving and natural appearance; 3) Our proposed methods.

In the second part of this work, we presented two fast hue-preserving HE methods based on 1D and 2D histograms of the *RGB* color space for image contrast enhancement. The HP1DHE and HP2DHE methods have time and space complexities that comply with real-time application requirements. Although the C1DHE method is six times faster than ours, it is not hue preserving. We evaluated the resulting images objectively by using measures of contrast, naturalness and colorfulness [24] on a data set composed of 300 images, such that a quantitative comparison could be performed. The experimental results showed that the value of the contrast of the images produced by our methods is in average 50% greater than the original value. Simultaneously, our methods keep the quality of the image in terms of naturalness and colorfulness close to the quality of the original image. In practice, our methods enhance 512×512 image pixels in 100 milliseconds on a Pentium 4 - 2GHz.

Recall that both proposed methodologies are suitable for real-time and real-world applications.

Acknowledgment

This work was supported by the CNPq/MCT and CAPES-COFECUB/MEC, Brazilian Government's research support agencies and by French Government grant ANR SURF NT05-2_45825. The first author is very grateful to Prof. Dr Hugues Talbot, for financial providing support during part of his stay in France, where a substantial part of this work was developed.

References

- [1] S.-D. Chen and A. Ramli. Contrast enhancement using recursive mean-separate histogram equalization for scalable brightness preservation. *IEEE TCE*, 49(4):1301–1309, Nov. 2003.
- [2] S.-D. Chen and A. Ramli. Minimum mean brightness error bi-histogram equalization in contrast enhancement. *IEEE TCE*, 49(4):1310–1319, Nov. 2003.
- [3] D. Coltuc, P. Bolon, and J.-M. Chassery. Exact histogram specification. *IEEE TIP*, 15(5):1143–1152, May 2006.
- [4] CVG-URG. Image database, 2007. <http://decsai.ugr.es/cvg/dbimagenes/>.
- [5] R. Gonzalez and R. Woods. *Digital Image Processing*. Prentice Hall, 3rd edition, Jan. 2008.
- [6] Y.-T. Kim. Contrast enhancement using brightness preserving bi-histogram equalization. *IEEE TCE*, 43(1):1–8, Feb. 1997.
- [7] M. Luessi, M. Eichmann, G. Schuster, and A. Katsaggelos. New results on efficient optimal multilevel image thresholding. In *IEEE ICIP*, pages 773–776, 2006.
- [8] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *IEEE ICCV*, volume 2, pages 416–423, July 2001.
- [9] D. Menotti. Test image, 2007. <http://wavelet.dcc.ufmg.br/MHE/>.
- [10] D. Menotti. *Contrast Enhancement in Digital Imaging using Histogram Equalization*. Phd thesis, UFMG - Universidade Federal de Minas Gerais, DCC - Departamento de Ciência da Computação, Apr. 2008. <http://wavelet.dcc.ufmg.br/thesismenotti.pdf>.
- [11] D. Menotti. Test image, 2008. <http://wavelet.dcc.ufmg.br/FHPHEM/>.
- [12] D. Menotti, A. P. Melo, A. A. Araújo, J. Facon, and E. M. Sgarbi. Color image enhancement through 2D histogram equalization. In *IWSSIP*, pages 235–238, 2006.
- [13] D. Menotti, L. Najman, A. A. Araújo, and J. Facon. A fast hue-preserving histogram equalization method for color image enhancement using a bayesian framework. In *IWSSIP*, pages 414–417, 2007.
- [14] D. Menotti, L. Najman, J. Facon, and A. A. Araújo. Multi-histogram equalization methods for contrast enhancement and brightness preserving. *IEEE TCS*, 53(3):1186–1194, Aug. 2007.
- [15] D. Menotti, L. Najman, J. Facon, and A. A. Araújo. Fast hue-preserving histogram equalization methods for color image contrast enhancement. *omitted*, 2008. submitted in August 2008 - <http://wavelet.dcc.ufmg.br/fhphem.pdf>.
- [16] S. Naik and C. Murthy. Hue-preserving color image enhancement without gamut problem. *IEEE TIP*, 12(12):1591–1598, Dec. 2003.
- [17] R. Nickalls. A new approach to solving the cubic: Cardan's solution revealed. *The Mathematical Gazette*, 77:354–359, 1993.
- [18] N. Otsu. A threshold selection method from grey-level histograms. *IEEE TSMC*, 9(1):41–47, Jan. 1979.
- [19] M. Rabbani and P. Jones. *Digital Image Compression Techniques*. Society of Photo-Optical Instrumentation Engineers (SPIE), Bellingham, WA, USA, 1st edition, 1991.
- [20] P. Trahanias and A. Venetsanopoulos. Color image enhancement through 3-d histogram equalization. In *IEEE ICPR*, pages 545–548, 1992.
- [21] C. Wang and Z. Ye. Brightness preserving histogram equalization with maximum entropy: A variational perspective. *IEEE TCE*, 51(4):1326–1334, Nov. 2005.
- [22] Y. Wang, Q. Chen, and B. Zhang. Image enhancement based on equal area dualistic sub-image histogram equalization method. *IEEE TCE*, 45(1):68–75, Feb. 1999.
- [23] J.-C. Yen, F.-J. Chang, and S. Chang. A new criterion for automatic multilevel thresholding. *IEEE TIP*, 4(3):370–378, Mar. 1995.
- [24] S. Yendrikhovskij, F. Blommaert, and H. de Ridder. Perceptually optimal color reproduction. In *Proceedings of the Human Vision and Electronic Imaging III*, volume 3299, pages 274–281. SPIE, 1998.

Fusão de Métodos Baseados em Minúcias e em Cristas para Reconhecimento de Impressões Digitais *

¹Fernanda Pereira Sartori Falguera e ²Aparecido Nilceu Marana

UNESP – Faculdade de Ciências – Departamento de Computação – Bauru – SP – Brasil

¹fernanda.sartori@gmail.com

²nilceu@fc.unesp.br

Resumo

Biometria é uma das maiores tendências na identificação humana e a impressão digital é a característica biométrica mais amplamente utilizada. Entretanto, considerar o reconhecimento automático de impressões digitais um problema totalmente resolvido é um erro comum. Imagens de baixa qualidade e contendo impressões digitais parciais ainda representam um grande desafio para os métodos baseados em minúcias, os mais extensivamente utilizados. A Multibiometria é considerada uma das chaves para superar as limitações e aprimorar a acurácia de sistemas biométricos. Esta dissertação de mestrado apresenta a fusão de métodos baseados em cristas e em minúcias para reconhecimento de impressões digitais. Os resultados obtidos (redução média na Taxa de Erro Igual de mais de 30% e o aumento na Taxa de Recuperação Correta de 75%) mostram que a fusão de métodos baseados em cristas e em minúcias pode representar uma melhoria significativa na acurácia dos sistemas de reconhecimento de impressões digitais.

Abstract

Biometrics is one of the major tendencies in human identification and fingerprints are the most widely used biometrics trait. However, considering the automatic fingerprint recognition a completely solved problem is a common mistake. The most extensively used methods, the minutiae-based methods, do not perform well on poor-quality images and when just a small area of overlap between the template and the query image exists. The Multibiometrics is considered one of the keys to overcome the weakness and to

improve the accuracy of biometrics systems. This master thesis presents the fusion of minutiae-based and ridge-based methods. The achieved results (mean reduction of the Equal Error Rate of more than 30% and an increase of 75% in the Correct Retrieval Rate) have showed that the fusion of minutiae-based and ridge-based methods can provide a significant accuracy improvement of the fingerprint recognition systems.

1. Introdução

Reconhecimento biométrico refere-se ao uso de características físicas (face, impressão digital, etc) ou comportamentais (assinatura, voz, etc) para estabelecer a identidade de uma pessoa.

Para que uma característica humana possa se tornar um identificador biométrico ela deve possuir alguns requisitos como universalidade, unicidade, permanência, entre outros. Devido a isso, nenhum identificador biométrico é considerado o melhor.

As impressões digitais, no entanto, são os identificadores biométricos mais amplamente utilizados devido ao fato de praticamente todo ser humano as possuírem, delas serem distintas até mesmo em gêmeos idênticos e mesmo diante de mudanças temporárias, como cortes e cicatrizes, elas se regenerarem voltando a sua forma original [9].

Os métodos baseados em minúcias são os mais extensivamente utilizados em sistemas automáticos de reconhecimento de impressões digitais. Todavia, é um erro comum considerar o reconhecimento automático de impressões digitais um problema totalmente resolvido. A extração de pontos de minúcias ainda representa um problema complexo em imagens de baixa qualidade [9]. Além disso, o casamento de pontos de minúcias, quando existe apenas uma pequena área de

sobreposição entre a imagem de consulta e a imagem *template*, também representa uma dificuldade.

Estes cenários, porém, estão se tornando cada vez mais comuns na medida em que sensores de impressões digitais estão se popularizando em celulares, notebooks, pendrives e outros dispositivos móveis.

Atualmente, o uso de sistemas multibiométricos é considerado uma das chaves para contornar as limitações e melhorar a acurácia dos sistemas biométricos. De acordo com Ross, Nandakumar e Jain [14], é extremamente difícil aumentar significativamente a acurácia dos sistemas utilizando apenas um identificador biométrico, uma única forma de representação e um único algoritmo de casamento.

Neste contexto, o presente trabalho vem contribuir ao apresentar um estudo sobre a multibiometria, uma das principais tendências na área de reconhecimento biométrico, e ao buscar formas de melhorar a acurácia de sistemas baseados na impressão digital, identificador biométrico que estará cada vez mais presente no cotidiano das pessoas.

1.1. Objetivo

O objetivo geral deste trabalho é aprimorar a acurácia dos sistemas atuais de reconhecimento de impressões digitais por meio da fusão de métodos baseados em minúcias e em cristas, possibilitando, dessa forma, uma adoção mais segura e confiável dessa tecnologia.

1.2. Justificativas

Estudos realizados pelo *International Biometric Group* (IBG), empresa de consultoria do setor de Biometria dos Estados Unidos, apontam que as vendas globais dos equipamentos de Biometria vão saltar de US\$ 2,1 bilhões em 2006 para US\$ 5,7 bilhões em 2010. O reconhecimento de impressões digitais, o mais difundido e barato dos sistemas biométricos, deve responder por 44% do mercado global do setor [15].

Tais estudos apenas confirmam que o uso de sistemas biométricos deixou de fazer parte somente dos filmes de ficção científica e passou a estar presente na vida real, sendo utilizados até mesmo em aplicações cruciais para a sociedade.

Vários exemplos corroboram este fato, como o Sistema de Eleição Brasileiro com o uso da foto e da impressão digital estar em teste em três cidades brasileiras [2]. O novo passaporte, incluindo a foto digital do rosto e as impressões digitais. Além disso, o

uso da biometria também já pode ser visto em várias situações do cotidiano, como em bancos, planos de saúde, locadoras, academias, etc.

A primeira pergunta a ser respondida, portanto, deve ser: O uso da Biometria em larga escala é suficientemente seguro?

Para responder a essa pergunta, é necessário entender a real extensão dos problemas que podem ser causados por uma margem de erro aparentemente insignificante. Por exemplo, uma taxa de erro de 1% em um banco de dados de 47 milhões de impressões digitais, como o instalado pelo *Federal Bureau of Investigation* (FBI) aproximadamente no ano 2000, significaria 470 mil impressões que poderiam ser falsamente reconhecidas.

No entanto, margens de erro até mesmo acima de 1% não são raras, mesmo em sistemas biométricos considerados eficientes. Esses erros são geralmente ocasionados por algumas dificuldades inerentes à característica biométrica utilizada. No caso das impressões digitais, por exemplo, de acordo com Maltoni et al. [9], a extração dos pontos de minúcias ainda é um problema complexo em imagens de baixa qualidade, enquanto que a extração de outras informações, como as cristas e suas características, apresenta-se de forma mais confiável. Além disso, segundo Jain et al. [7], em aproximadamente 3% da população, as imagens das impressões digitais são de baixa qualidade devido à presença de cortes ou simplesmente pelas condições da pele (dedos secos ou oleosos demais).

Desse modo, a pergunta a ser respondida então passa a ser: Como essa melhoria na acurácia pode ser alcançada?

O uso de sistemas multibiométricos pode ser a resposta para essa questão. A Multibiometria é considerada uma das chaves para melhorar a acurácia dos sistemas biométricos por ser capaz de explorar os diferentes pontos fortes e contornar as diferentes limitações dos métodos biométricos [14].

Neste contexto, a justificativa para o presente trabalho é a necessidade de aprimorar a acurácia dos sistemas de reconhecimento de impressões digitais. A fusão de métodos baseados em minúcias e em cristas, proposta neste trabalho, vem ao encontro dessa necessidade, contribuindo ao mesmo tempo para a área acadêmica, ao apresentar um estudo sobre a multibiometria, uma das maiores tendências da área de reconhecimento biométrico e para a sociedade, ao buscar formas de melhorar a acurácia de sistemas

baseados no identificador biométrico que estará cada vez mais presente em seu cotidiano: a impressão digital.

2. Impressões Digitais

Impressões digitais são os desenhos formados pelas cristas papilares (elevações da pele), presentes nas polpas dos dedos das mãos [13]. As cristas são formadas no feto através da combinação de fatores ambientais e genéticos de cada indivíduo. Conseqüentemente, as impressões digitais são uma marca exclusiva de cada indivíduo, mesmo entre gêmeos idênticos.

Além das cristas, outras características estruturais evidentes são encontradas nas impressões digitais, as minúcias. Minúcia, no contexto de impressões digitais, se refere aos vários modos pelos quais uma crista pode se tornar descontínua [9]. Em sistemas automáticos de reconhecimento de impressões digitais dois tipos de minúcias são utilizados: terminações e bifurcações. Na Figura 1 é possível observar uma minúcia do tipo terminação e outra do tipo bifurcação bem como suas coordenadas (x_0 , y_0) e sua orientação (θ).

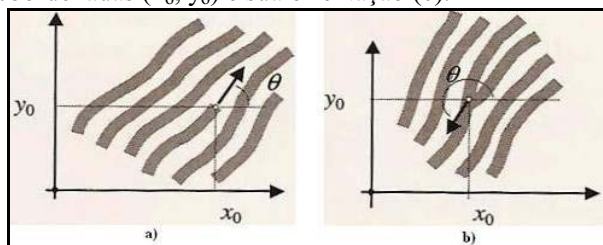


Figura 1: a) Terminação e b) Bifurcação [9].

2.1. Reconhecimento de Impressões Digitais

De acordo com Maltoni et al. [9], as abordagens utilizadas no casamento automático de impressões digitais podem ser divididas em duas grandes categorias:

- Baseadas em minúcias: abordagem mais amplamente utilizada. A fase de casamento consiste em encontrar o alinhamento entre o conjunto de minúcias da imagem *template* e da imagem de consulta que resulta no número máximo de pares de minúcias. Exemplos da abordagem baseada em minúcias são os métodos propostos por Jain et al. [6] e o *software NIST Biometric Image Software (NBIS)* do *National Institute of Standards and Technology (NIST)* [12]. Ambos os métodos são descritos na seção 2.1.1;
- Baseadas nas características das cristas: as abordagens pertencentes a esta família comparam as

impressões digitais em termos das características extraídas das cristas como orientação local, frequência, forma da crista, informação da textura, etc. Uma das pesquisas mais recentes desta abordagem foi o método proposto por Marana e Jain [10], que faz uso da transformada de Hough. Este método é apresentado na seção 2.1.2.

2.1.1. Métodos Baseados em Minúcias

Esta seção apresenta uma breve descrição dos métodos baseados em minúcias objetos de estudo do presente trabalho.

Método de Jain

No algoritmo proposto por Jain et al. [6], para cada minúcia detectada, além das coordenadas x, y e da orientação (θ), é armazenado também o segmento associado à crista (aproximadamente dez pontos equidistantes uns dos outros pela média da distância intercristas). Esse segmento é utilizado para obter o alinhamento (rotação e translação) entre a imagem de consulta e a imagem *template*.

Uma vez obtidos os parâmetros de alinhamento, na etapa de casamento, os conjuntos de pontos de minúcias são representados como uma cadeia no sistema de coordenadas polares. Para ser mais tolerante a erros provocados pelas deformações e pelos ruídos, é utilizada uma caixa de tamanho ajustável dentro da qual pontos coincidentes são procurados. Um algoritmo de casamento de seqüências fornece a pontuação final de casamento. A Figura 2 mostra um exemplo de casamento obtido por este método. O algoritmo completo pode ser encontrado em [6].

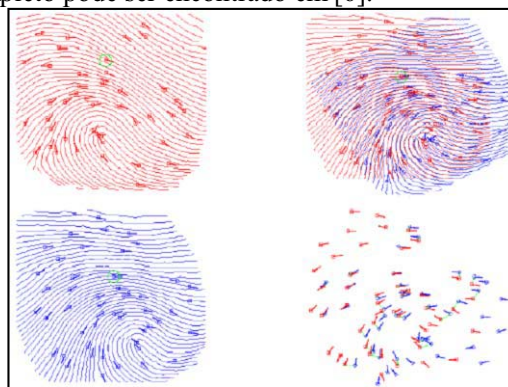


Figura 2: Exemplo de casamento obtido por este método [6].

Método do *software* NBIS

O *software* NBIS [12] é composto de dois programas principais que são utilizados no reconhecimento de impressões digitais:

- **Mindtct**: responsável pela detecção das minúcias. Esse programa tem como entrada uma imagem de impressão digital e como saída um arquivo texto com o conjunto de pontos de minúcias detectados com seus descritores: coordenadas x,y e a orientação θ da minúcia.
- **Bozorth3**: é uma versão modificada do algoritmo proposto por Allan S. Bozorth [17]. Esse programa é responsável pelo casamento de dois conjuntos de minúcias. Ele tem como entrada o arquivo texto gerado pelo programa Mindtct e gera como saída uma pontuação de casamento. Para isso, esse programa computa para a imagem de consulta e de *template*, as medidas relativas de cada minúcia em relação a todas as outras minúcias e armazena em uma tabela. São essas medidas relativas que fazem com que esse algoritmo seja invariante à rotação e translação. A Figura 3 ilustra um exemplo das medidas relativas obtidas para um par de minúcias da impressão digital de consulta e um par de minúcias da impressão *template*.

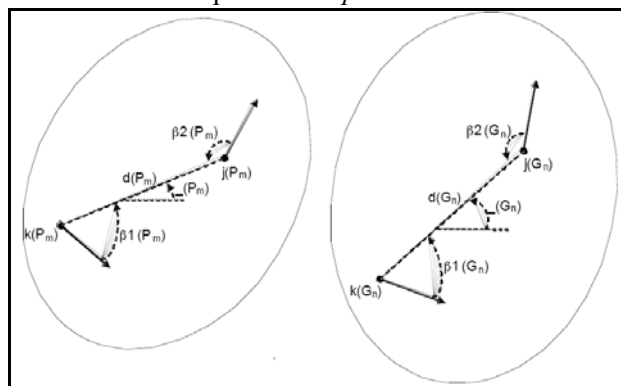


Figura 3: Exemplo de duas impressões digitais do mesmo dedo com variações de rotação e escala [17].

Na etapa de casamento, são procuradas as entradas compatíveis da tabela da imagem de consulta em relação à imagem *template*. Essas associações compatíveis entre pares de minúcias são consideradas como um grafo a ser percorrido para encontrar o caminho mais longo (caminho com maior número de associações).

Os detalhes da implementação dos programas *Mindtct* e *Bozorth3* podem ser encontrados em [17].

2.1.2. Método Baseado em Cristas

O método baseado em cristas proposto por Marana e Jain [10] e adotado neste trabalho extrai as retas que passam pelas cristas utilizando a Transformada de Hough e depois utiliza essas retas para estimar os parâmetros de rotação e translação. A Figura 4 ilustra

as cristas detectadas e as retas extraídas para uma determinada crista.

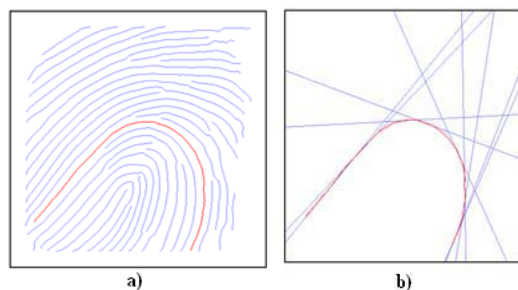


Figura 4: a) Cristas detectadas; b) Retas extraídas a partir de uma das cristas [10].

Para calcular a pontuação de casamento é utilizada uma matriz que identifica quantos pixels de cada crista das duas impressões digitais casam. A Figura 5 ilustra um casamento de impressões digitais obtido por este método.

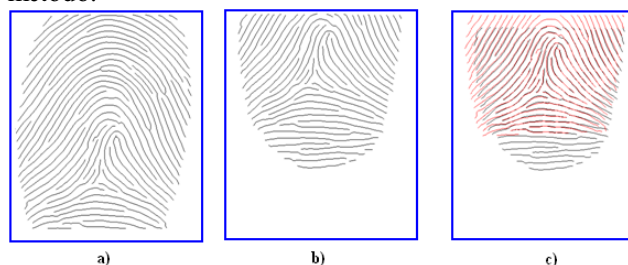


Figura 5: a) Imagem de Consulta; b) Imagem *Template*; c) Alinhamento obtido pelo método proposto por Marana e Jain [10].

A descrição completa deste método pode ser encontra em [10].

2.2. Desafios do Reconhecimento de Impressões Digitais

O reconhecimento de impressões digitais ainda apresenta desafios, principalmente devido à grande variabilidade presente em diferentes impressões do mesmo dedo. Os principais fatores que contribuem para este problema são [9]: (i) deslocamento; (ii) rotação; (iii) pequena área de sobreposição entre a imagem de *template* e a imagem de consulta; (iv) deformação e (v) outros problemas como as condições da pele (cicatrizes, oleosidade), os ruídos devido à sujeira no sensor, etc.

3. Multibiometria

Multibiometria significa a fusão de diferentes tipos de informação em um sistema biométrico. Na seção 3.1 são apresentados os diferentes níveis e as técnicas de fusão utilizadas e avaliadas no presente trabalho. Na

seção 3.2 são apresentados os desafios dos sistemas multibiométricos.

3.1. Níveis e Técnicas de Fusão

Um sistema multibiométrico pode ter os seguintes níveis de fusão [14]:

- Fusão antes do casamento: nível de sensor e nível das características;
- Fusão após o casamento: nível de pontuação, nível de decisão e nível de classificação.

Dentro de cada nível, existem várias técnicas de fusão.

3.1.1. Nível de Classificação

Para este nível, as técnicas de fusão propostas por Ho, Hull e Srihari [5] podem ser utilizadas: (i) Classificação mais Alta: a melhor classificação é associada à amostra; (ii) Contador de Borda: utiliza a soma das classificações obtidas pelos métodos que serão combinados e (iii) Regressão Logística: refere-se a uma generalização do método de Borda no qual um peso é associado para cada método.

3.1.2. Nível de Decisão

Para a fusão em nível de decisão, a técnica Decisão Bayesiana estudada por Xu et al. [18] pode ser utilizada. Esta técnica consiste em transformar os rótulos de decisão (classes) em valores de probabilidade. Sendo que para isso, é necessária uma fase de treinamento onde a matriz de confusão de cada método é calculada.

3.1.3. Nível de Pontuação

Na maioria das vezes, para este nível de fusão é necessário incluir uma etapa de normalização das pontuações, uma vez que as pontuações de casamento obtidas pelos métodos a serem combinados podem estar em intervalos diferentes. As técnicas de normalização estudadas e avaliadas no presente trabalho foram:

- Min-Max: técnica mais utilizada quando os valores dos limites das pontuações são conhecidos [9];
- Função Sigmóide Dupla: técnica utilizada por Cappelli, Maio e Maltoni [1] e consiste em realizar uma transformação linear das pontuações na região de sobreposição entre os casamentos genuínos e impostores, enquanto que as pontuações fora dessa região são transformadas de maneira não-linear;
- Estimadores Tanh: técnica introduzida por Hampel et al. [4] que objetiva reduzir a influência das pontuações nas extremidades da distribuição.

Uma vez normalizadas as pontuações, as técnicas de fusão que podem ser aplicadas são: Soma, Produto, Pontuação Máxima, Pontuação Mínima e Soma com Pesos [9].

3.2. Desafios de Sistemas Multibiométricos

Segundo Nandakumar [11], o projeto de sistemas multibiométricos é um problema desafiador, pois é muito difícil prever a estratégia de fusão ótima. Estas dificuldades se devem, principalmente, aos seguintes fatores:

- **Fusão antes do casamento:** (i) incompatibilidade: a fusão no nível de sensor e no nível de características nem sempre é possível devido à incompatibilidade das informações capturadas; (ii) complexidade: a complexidade de se projetar um algoritmo de fusão visando desenvolver um novo método para extrair as características e realizar o casamento dos dados combinados pode anular as vantagens da fusão em termos do custo-benefício;
- **Fusão após o casamento:** (i) habilidade de discriminação variada: existem medidas que são menos discriminativas que outras. Por exemplo, as cristas de uma impressão digital são medidas menos discriminativas que as minúcias; (ii) correlação entre os métodos: em geral, a fusão de métodos independentes tende a gerar uma melhoria mais significativa na acurácia do que métodos correlacionados.

De uma perspectiva acadêmica, pesquisas em multibiometria apresentam inúmeros desafios que vão desde determinar o tipo de informação que será fundida, projetar uma metodologia ótima de fusão considerando o cenário, o nível e a técnica de fusão até avaliar e comparar as diferentes metodologias de fusão e construir um sistema multibiométrico automatizado [14]. Tal cenário representa o objeto de estudo do presente trabalho e seu desenvolvimento incluindo seus materiais e métodos serão expostos nas seções seguintes.

4. Material

Para analisar o desempenho dos métodos baseados em minúcias e em cristas individualmente bem como as técnicas de fusão implementadas foi utilizada a base de dados DB1A da FVC2002 (Fingerprint Verification Competition 2002) [3], uma base de dados gratuita e utilizada como referência para avaliação de métodos de impressões digitais.

Essa base de dados é composta por 800 imagens (100 indivíduos, 8 imagens por indivíduo) de 388 x 374 pixels e 500 dpi de resolução com as seguintes características [54]:

- As impressões foram coletadas em três sessões com pelo menos uma semana separando cada sessão;
- Nenhum esforço foi despendido para controlar a qualidade da imagem e o sensor não foi sistematicamente limpo;
- Durante a segunda sessão, os indivíduos foram instruídos para exagerar no deslocamento e na rotação;
- Durante a terceira sessão, os dedos foram alternadamente secos e umedecidos.

Para a realização da fusão de métodos de reconhecimento de impressões digitais baseados em cristas e em minúcias foram escolhidos os seguintes métodos:

- Método Baseado em Minúcias de Jain et al. [6];
- Método Baseado em Cristas de Marana e Jain [10];
- Método Baseado em Minúcias do *software* NBIS [12].

5. Metodologia

Esta seção apresenta a metodologia proposta para a fusão dos métodos de reconhecimento de impressões digitais baseados em minúcias e em cristas, bem como a forma de avaliação dos resultados.

5.1. Estratégia de Fusão

O objetivo do presente trabalho é aprimorar a acurácia dos sistemas atuais de reconhecimento de impressões digitais por meio da fusão de métodos baseados em minúcias e em cristas. Para que fosse possível avaliar de forma consistente as técnicas de fusão, as seguintes fusões foram realizadas:

- Fusão 1: Método baseado em cristas proposto por Marana e Jain [10] com o método baseado em Minúcias proposto por Jain et al. [6].
- Fusão 2: Método baseado em cristas proposto por Marana e Jain [10] com o método baseado em minúcias do *software* NBIS [12].
- Fusão 3: Método baseado em minúcias proposto por Jain et al. [6] com o método baseado em minúcias do *software* NBIS [12].
- Fusão 4: Método baseado em cristas por Marana e Jain [10] com os métodos baseados em minúcias: proposto por Jain et al. [6] e implementado no *software* NBIS [12].

Embora a viabilidade (melhoria na acurácia) da fusão pudesse ser validada com apenas umas das três primeiras fusões acima citadas, a estratégia definida permitiu analisar de forma mais abrangente a fusão de métodos baseados em minúcias e em cristas, possibilitando verificar o comportamento da fusão de métodos baseados nas mesmas características (minúcias) em relação à fusão de métodos que utilizam diferentes características (minúcias versus cristas), bem como o comportamento da fusão de três métodos em comparação com dois métodos, além de verificar a viabilidade da fusão quando um dos métodos isoladamente apresenta uma alta acurácia, como acontece com o *software* NBIS.

No entanto, conforme apresentado na seção 3.2, um dos fatores que impactam um sistema multibiométrico é o grau de dependência entre os métodos utilizados na fusão. Diante disso, visando identificar a viabilidade da estratégia de fusão definida, foi aplicada uma técnica para medir o grau de dependência entre os métodos a serem combinados.

As medidas escolhidas para calcular o grau de dependência entre os métodos foram a Q-Statistic e a Q-Average. A escolha dessas medidas foi baseada nas pesquisas realizadas por Kuncheva e Whitaker [8], que avaliaram dez técnicas estatísticas para medir a diversidade entre classificadores e recomendaram o uso da medida Q-Statistic e, quando mais de dois métodos são combinados, da medida Q-Average.

Pelo fato do grau de dependência dos métodos ter resultado em valores que indicaram que para todas as fusões pretendidas uma melhoria na acurácia poderia ser obtida, (atingindo, portanto, os objetivos do presente trabalho), as quatro fusões propostas foram realizadas.

5.2. Níveis e Técnicas de Fusão

Para que fosse possível avaliar o desempenho da fusão em diferentes níveis, foram utilizadas e implementadas as técnicas de fusão explicadas na seção 3.1.

5.3. Protocolo de Testes

Para avaliar o desempenho dos métodos baseados em minúcias e em cristas individualmente, bem como as técnicas de fusão implementadas, os experimentos foram conduzidos de acordo com o protocolo de teste FVC-2002 [3], a saber:

- Reconhecimento Genuíno: cada impressão digital foi testada com as impressões restantes do mesmo indivíduo, mas evitando casamentos simétricos;
- Reconhecimento Impostor: cada impressão digital de um indivíduo foi testada com a impressão digital de mesma ordem dos indivíduos restantes (por exemplo, a primeira impressão digital de cada usuário foi testada com a primeira impressão digital de todos os outros usuários)

5.4. Medidas de Desempenho

As medidas de desempenho calculadas para cada método de reconhecimento de impressões digitais foram: (i) Taxa de Erro Igual (EER); (ii) Taxa de Recuperação Correta; e (iii) Taxa de Identificação.

A avaliação de desempenho das técnicas de fusão implementadas varia de acordo com o nível de fusão, pois, em cada nível, uma informação diferente é trabalhada. Para a fusão em nível de pontuação, o trabalho é realizado sobre as pontuações de casamento, e, portanto, foi utilizada a EER. Para a fusão em nível de classificação, a lista de classificação (*rank*) de cada imagem é utilizada, e por isso, foi utilizada a Taxa de Recuperação Correta. Para a fusão em nível de decisão, apenas a decisão final do sistema (identidade do indivíduo com maior pontuação de casamento) é considerada e, diante disso, a Taxa de Identificação foi utilizada.

5.5. Abordagens Utilizadas

Durante a implementação das técnicas de fusão foi necessário estabelecer algumas abordagens para a seleção das imagens da fase treinamento e para o estabelecimento de pesos. Tais abordagens foram condensadas nessa subseção para possibilitar uma organização melhor do texto.

Estabelecimento de Pesos

A técnica de fusão Regressão Logística no nível de classificação e a técnica Soma com Pesos no nível de pontuação requerem que pesos sejam estabelecidos para os métodos utilizados na fusão. Para determinar esses pesos, duas abordagens foram utilizadas:

1. A abordagem proposta por Wang, Tan e Jain [16] que faz com que cada método receba um peso proporcional as suas taxas de erro FAR (*False Acceptance Error*) e FRR (*False Rejection Rate*).
2. A segunda abordagem utilizada consistiu no cálculo empírico dos pesos. Para isso, os pesos foram

variados e os resultados foram calculados seguindo as medidas de desempenho utilizadas em cada um dos níveis de fusão.

Seleção das Imagens de Treinamento

Para a técnica de fusão Decisão Bayesiana no nível decisão é necessário uma fase de treinamento para se calcular a matriz de confusão. Duas abordagens foram utilizadas para selecionar as imagens do conjunto de treinamento:

1. Abordagem 1: A base de dados foi dividida de tal forma que das 8 imagens de cada usuário, mais de 50% fizessem parte da etapa de treinamento (4, 5, 6 e 7 imagens). Para isso, foram gerados randomicamente 10 conjuntos de treinamento para cada pacote de imagens. Os resultados de todos os conjuntos da mesma quantidade de imagens (por exemplo, os 10 conjuntos de 4 imagens por usuário) foram calculados e depois se obteve a média.
2. Abordagem 2: Semelhante à abordagem anterior, foram utilizados conjuntos de treinamento de 4 a 7 imagens por usuário com a diferença que nem todas as imagens foram escolhidas randomicamente, 4 imagens foram fixadas para cada usuário por representarem mais completamente a variabilidade intraclasse de cada usuário.

6. Resultados Experimentais

Esta seção apresenta os resultados obtidos individualmente pelos métodos utilizados bem como os melhores resultados obtidos pelas técnicas de fusão realizadas.

6.1. Resultados Obtidos pelos Métodos de Reconhecimento de Impressões Digitais

A Tabela 1 apresenta os resultados obtidos pelos métodos de reconhecimento de impressões digitais utilizados nas fusões.

É possível observar que para o método baseado em cristas é necessário recuperar 13 imagens do banco de dados para garantir uma recuperação correta (uma imagem do mesmo indivíduo) para 100% do banco de dados (*rank* 13). Para o método baseado em minúcias do método de Jain, é necessário recuperar 8 imagens (*rank* 8) e para o método baseado em minúcias do *software* NBIS, recuperando 1 imagem foi garantido para 100% das imagens do banco um casamento genuíno (*rank* 1).

Além disso, quando um sistema biométrico retorna a imagem de *template* mais similar a uma dada imagem de consulta e essa imagem de *template* é uma imagem do mesmo usuário, significa que a identificação foi bem sucedida. A taxa de identificação geral de cada método foi obtida calculando a porcentagem das imagens pertencentes ao *rank 1*, ou seja, a porcentagem de imagens que entre todos os casamentos genuínos e impostores possíveis, um casamento genuíno obteve maior pontuação.

Os resultados obtidos mostram que o método do *software* NBIS apresenta acurácia máxima tanto para a Taxa de Recuperação Correta quanto para a Taxa de Identificação na base de dados utilizada e, portanto, a fusão no nível de classificação e a fusão no nível de decisão que são avaliadas por essas duas medidas não são necessárias para este método.

Tabela 1: Resultados obtidos por cada um dos métodos.

Métodos	Medidas de Desempenho		
	EER	Recuperação Correta 100 % Casamento Genuíno	Taxa de Identificação
Método Baseado em Cristas	8,27%	Rank 13	98,75%
Método Baseado em Minúcias (Jain)	4,09%	Rank 8	99,25%
Método Baseado em Minúcias (<i>software</i> NBIS)	3,08%	Rank 1	100%

6.2. Resultados da Fusão

Esta seção apresenta os melhores resultados obtidos pelos níveis e técnicas de fusão aplicadas.

6.2.1. Nível de Classificação

Considerando as técnicas utilizadas, a técnica Classificação Mais alta alcançou o melhor resultado. Na análise pela Taxa de Recuperação Correta, com *rank 2*, pelo menos um casamento genuíno foi obtido para 100% das imagens. Isso representou um aumento de 75% na acurácia quando comparado ao método mais preciso. A Tabela 2 apresenta os resultados da fusão no nível de classificação e a Figura 6 apresenta uma representação gráfica da técnica que obteve melhor resultado.

6.2.2. Nível de Decisão

A técnica Decisão Bayesiana foi aplicada utilizando duas abordagens para a escolha das imagens da fase de

treinamento. A Taxa de Identificação da segunda abordagem apresentou uma melhoria na acurácia passando de 99,25% para 99,75%. A primeira abordagem, no entanto, apresentou uma queda de acurácia em todos os conjuntos de imagens. O gráfico da Figura 7 apresenta os resultados obtidos comparados entre si.

Tabela 2: Resultados da fusão no nível de classificação.

Nível de Classificação	Rank	% Melhoria
Classificação Mais Alta	2	75
Contador de Borda	4	50
Regressão Logística - Abordagem 1 para estabelecimento de pesos	6	25
Regressão Logística - Abordagem 2 para estabelecimento de pesos	4	50

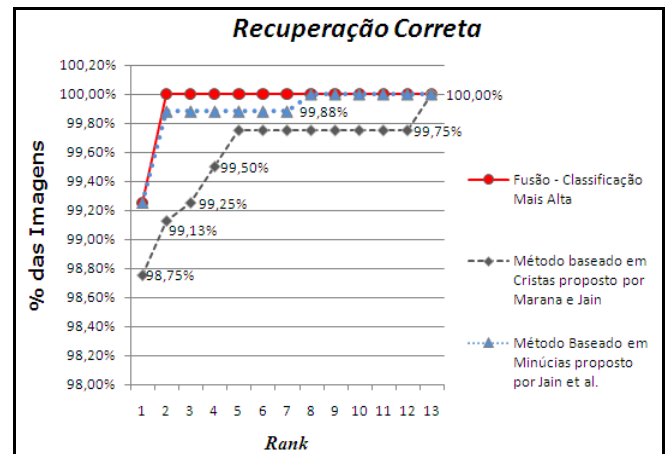


Figura 6: Taxa de recuperação correta obtida pela técnica classificação mais alta na fusão do método baseado em cristas com o método baseado em minúcias proposto por Jain et al. [6].

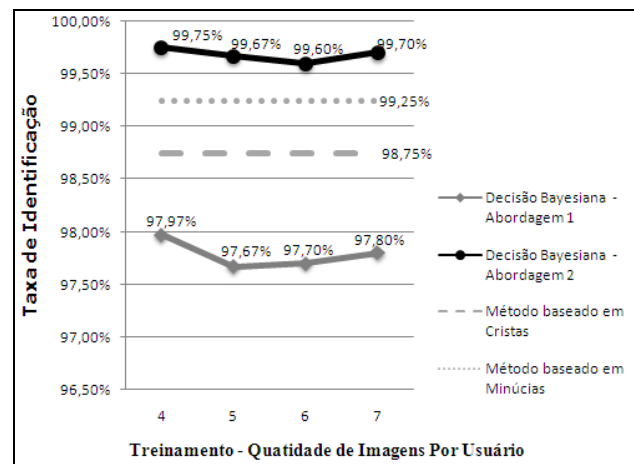


Figura 7: Taxa de identificação obtida pela técnica de fusão decisão bayesiana na fusão do método baseado em cristas com o método baseado em minúcias proposto por Jain et al. [6].

6.2.3. Nível de Pontuação

O nível de pontuação foi utilizado nas quatro fusões realizadas e explicadas na seção 5.1. Sendo que para todas as fusões, a técnica de normalização de melhor desempenho foi a Estimadores Tanh e a melhor técnica de fusão variou entre o Produto e a Soma com Pesos. A porcentagem de melhoria média na Taxa de Erro Igual foi de mais de 32%, sendo que a melhoria mais significativa chegou aos 42,53%. A Tabela 3 apresenta os resultados obtidos pelas quatro fusões realizadas no nível de pontuação bem como a melhor técnica de normalização e de fusão de cada uma delas. Na Figura 8 é possível observar o comportamento da fusão em nível de pontuação para a fusão que obteve a maior porcentagem de melhoria na acurácia.

Tabela 3: Resultados da fusão no nível de pontuação.

Nível de Pontuação	Fusão 1	Fusão 2	Fusão 3	Fusão 4
Melhor Técnica de Normalização	Estimadores Tanh	Estimadores Tanh	Estimadores Tanh	Estimadores Tanh
Melhor Técnica de Fusão	Produto	Soma com Pesos	Soma com Pesos	Produto
Resultado Obtido	EER De: 4,09 Para: 2,79	EER De: 3,08 Para: 2,36	EER De: 3,08 Para: 1,77	EER De: 3,08 Para: 2,13
% de Melhoria	31,78%	23,38%	42, 53%	30,84%

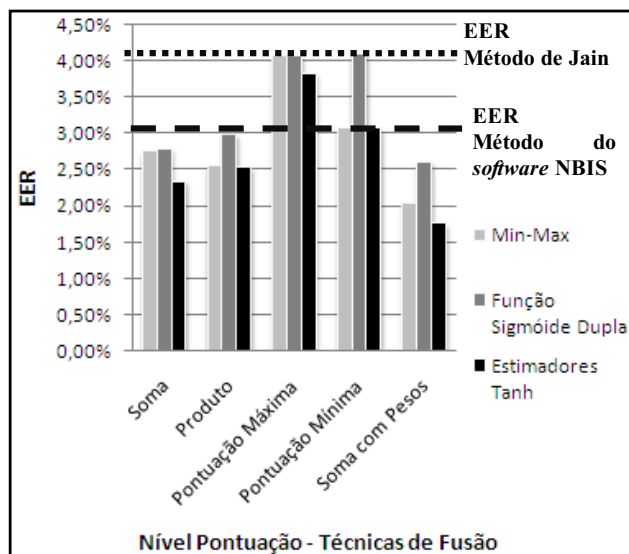


Figura 8: EER obtidas pelas técnicas de fusão em nível de pontuação na fusão do método proposto por Jain et al. [6] com o método do *software* NBIS [12].

7. Conclusões

Os indícios obtidos pelas medidas *Q-Statistic* e *Q-Average* de que as estratégias de fusão definidas no presente trabalho poderiam obter um ganho de desempenho, foram comprovados na prática em todas as fusões propostas e em todos os níveis de fusão.

Neste contexto, algumas conclusões importantes puderam ser obtidas dentro de cada nível de fusão:

i. Fusão no Nível de Classificação

- A técnica de fusão Classificação mais Alta é a mais indicada quando os métodos apresentam diferentes níveis de acurácia por ser capaz de fazer uso eficiente dos pontos fortes de cada método;
- As técnicas Contador de Borda e Regressão Logística não são recomendadas quando os métodos têm uma diferença de acurácia significativa, por representam uma medida do grau de concordância entre os diferentes métodos, sendo dependente do desempenho de ambos.

ii. Técnica de Fusão Decisão Bayesiana no Nível de Decisão

- A escolha das imagens que farão parte da fase de treinamento é crucial. Isso se torna ainda mais crítico quando apenas um número limitado de imagens por usuário está disponível;
- Quando a fase de treinamento é realizada de forma a representar a variabilidade intraclasse de cada usuário, essa técnica de fusão pode obter uma melhoria significativa na acurácia.

iii. Fusão no Nível de Pontuação

- A técnica de normalização é um ponto importante a ser considerado. A técnica de normalização Estimadores Tanh é mais indicada quando comparada às técnicas Min-Max e Função Sigmóide Dupla;
- As técnicas de fusão Produto e Soma com Pesos são as mais indicadas para as fusões no nível de pontuação. Em todas as fusões a técnica Soma e Produto apresentaram melhoria na acurácia, sendo que, na média geral, a Soma obteve 30% de melhoria e o Produto 20%.

Pode-se observar, no entanto, que a fusão dos dois métodos baseados em minúcias propiciou para a base de dados utilizada (DB1A-FVC2002) uma melhoria mais representativa na acurácia em relação às fusões que utilizaram o método baseado em cristas. Acreditamos que para uma base de dados obtida com

pequenos sensores, cada vez mais comuns principalmente nos dispositivos móveis, a quantidade reduzida de minúcias pode comprometer o desempenho de métodos baseados apenas em pontos de minúcias e favorecer o desempenho do método baseado em cristas.

Ao analisar as taxas de melhoria na acurácia mais significativas (**75%** de melhoria obtida pela fusão em nível de classificação, **0,50%** de melhoria obtida pela fusão em nível de decisão fazendo que a taxa de identificação alcançasse o patamar de **99,75%, 42,53%** de melhoria obtida pela fusão em nível de pontuação), pode-se concluir que esse aumento na acurácia dificilmente seria possível de se obter utilizando uma única forma de representação, um único algoritmo de extração de características e um único algoritmo de casamento.

Esses resultados evidenciam que a fusão é capaz de explorar os diferentes pontos fortes e contornar as diferentes limitações de cada método e apresentar um ganho significativo na acurácia.

Referências

- [1] CAPPELLI, R.; MAIO, D.; MALTONI, D. **Combining Fingerprint classifiers**. Proceedings of First International Workshop on Multiple Classifier Systems, p. 351-361, 2000.
- [2] FOLHA ON-LINE. **Eleitores serão identificados pelas impressões digitais em 2008**. Outubro de 2007. Disponível em <<http://www1.folha.uol.com.br/folha/informatica/ult124u338365.shtml>>. Acesso em: 27 de abril de 2008.
- [3] **FVC 2002 – Fingerprint Verification Competition**. Disponível em: <<http://bias.csr.unibo.it/fvc2002/databases.asp>>. Acesso em: 18 de Janeiro de 2008.
- [4] HAMPEL, F.R.; ROUSSEUW, P.J.; RONCHETTI, E.M.; STAHEL, W.A. **Robust Statistics: The Approach Based on Influence Functions**. Wiley, Nova York, 1986, 536 p.
- [5] HO, T. K.; HULL, J. J.; SRIHARI, S. N. **Decision Combination in Multiple Classifier Systems**. IEEE Transactions on pattern analysis and machine intelligence. v. 16, n. 1, p.66-75, 1994.
- [6] JAIN, A.K.; HONG, L.; PANKANTI, S; BOLLE, R. **An Identity-Authentication System Using Fingerprints**. Proceedings of IEEE, v. 85, n. 9, p. 1365-1388, 1997.
- [7] JAIN, A.; PANKANTI, S.; PRABHAKAR, S.; HONG, L.; ROSS, A.; WAYMAN, J. **Biometrics: A Grand Challenge**. Disponível em: <http://biometrics.cse.msu.edu/Presentations/Jainetal_BiometricsGrandChallenge_ICPR04.pdf>. Acesso em: 27 de abril de 2008
- [8] KUNCHEVA, L.; WHITAKER, C. **Measures of Diversity in Classifier Ensembles and Their Relationship with the Ensemble Accuracy**. Machine Learning, v. 51, p. 181-207, 2003.
- [9] MALTONI, D.; MAIO, D.; JAIN, A. K.; PRABHAKAR, S. **Handbook of Fingerprint Recognition**. Usa, Springer, 2003.
- [10] MARANA, A. N.; JAIN, A. K. **Ridge-Based Fingerprint Matching Using Hough Transform**. IEEE Proceedings of the Sibgrapi, Natal, Brasil, p. 112-119, 2005.
- [11] NANDAKUMAR, K. **Multibiometric Systems: Fusion Strategies and Template Security**. 2008, 249 p., Dissertação (Doutor em Filosofia), Universidade Estadual de Michigan, Departamento de Ciência da Computação e Engenharia.
- [12] **NBIS - Nist Biometric Image Software**. Disponível em: <<http://www.itl.nist.gov/iad/894.03/nigos/nbis.html>>. Acesso em: 27 de abril de 2008.
- [13] PASQUALI, L.; ARAÚJO, M. E. C. **Histórico dos processos de identificação**. Disponível em: <<http://papioscopistas.org/historico.doc>>. Acesso em: 04 de Fevereiro de 2007.
- [14] ROSS, A.; NANDAKUMAR, K.; JAIN, A.K. **Handbook of Multibiometrics**. USA, Springer, 2006.
- [15] SPOSITO, R. **A biometria dá um gás na segurança das empresas de qualquer tamanho**. Info SMB, 2006. Disponível em: <<http://info.abril.com.br/infosmb/edicoes/001/18-3.sh>>. Acesso em: 27 de abril de 2008.
- [16] WANG, Y.; TAN, T.; JAIN, A. K. **Combining Face and Iris Biometrics for Identity Verification**. Proc. AVBPA, 2003.
- [17] WATSON, C.; GARRIS, M.; TABASSI, E.; WILSON, C. L.; MCCABE, R. M.; JANET, S.; KO, K. **User's guide to NIST Biometric Image Software (NBIS)**. National Institute of Standards and Technology, Outubro de 2004.
- [18] XU, L.; KRZYZAK, A.; SUEN, C. Y. **Methods of Combining Multiple Classifiers and their Applications to Handwriting Recognition**. IEEE Transactions on Systems, Man, and Cybernetics, v. 22, n. 3, p. 418-435, 1992.

High-resolution image reconstruction using the Discontinuity Adaptive ICM algorithm

MARTINS, A.L.D. and MASCARENHAS, N.D.A.

Universidade Federal de São Carlos, Departamento de Computação
Via Washington Luís, Km 235, CP 676, CEP: 13.565-905, São Carlos, SP, Brazil
ana_martins@dc.ufscar.br, nelson@dc.ufscar.br

Abstract

Super-Resolution reconstruction methods intend to reconstruct a high-resolution image from a set of low-resolution observations. For that, the observed images must have sub-pixel displacements between each other. This requirement allows the existence of different information on each of the low-resolution images. This paper discusses a Bayesian approach for the super-resolution reconstruction problem using Markov Random Fields (MRF) and the Potts-Strauss model for the image characterization. Since it is difficult to maximize the joint probability, the Iterated Conditional Modes (ICM) algorithm is used to maximize the local conditional probabilities sequentially. For the oversmoothness inherent to Maximum a Posteriori (MAP) formulations using MRF prior models, we adopt a discontinuity adaptive (DA) procedure for the ICM algorithm. The proposed method was evaluated in a simulated situation by the Peak signal-to-noise ratio (PSNR) method and the Universal Image Quality Index (UIQI). Also, video frames with sub-pixel displacements were used for the visual evaluation. The results indicate the effectiveness of our approach both by numerical and visual evaluation.

1. Introduction

High-resolution images are usually necessary in a great number of applications: medical imaging; image surveillance methods; digital television picture technology; and high-precision classification in remote-sensing, just to name a few.¹ This kind of images can be acquired using high-resolution acquisition devices. However, there are several

cost and hardware limitations. For instance, the size of each detector of a charged-couple device (CCD) camera cannot be made arbitrarily small because of the shot noise effect [17]. Also, in positron emission tomography (PET) technology, besides the problem with the detector size, spatial resolution is limited by several factors such as patient motion and positron range of the isotope of interest [13]. Thus, it is of great interest to reconstruct a high resolution image using signal processing techniques. The super-resolution image reconstruction approach uses a set of low-resolution observations of the same scene to reconstruct an image in a high-resolution grid. For that, the low-resolution observations must have sub-pixel displacements between each other. This characteristic allows the existence of different information on each of the low-resolution images and the exceeding information can be used to increase the spatial resolution of the image. According to Park et al. [17], images with this characteristic can be acquired in many ways: from a single camera with several captures; from multiple cameras located in different positions; by scene motions or local objects movements; by vibrating imaging systems.

Tsai and Huang [20] were the first to address the problem of reconstructing a high-resolution image from a set of low-resolution observations of the same scene. They used a frequency domain approach based on the shifting property of the Fourier transform to model global translational scene motion. More recently, several algorithms were proposed to solve the same problem, most of them in a spatial domain context. In fact, despite the simplicity of frequency domain approaches, there are several disadvantages on this formulation [2]. For instance, it does not usually permit more general motion models. Spatial domain approaches allow arbitrary motion models, complex degradation models and, mainly, the inclusion of *a priori* constraints. Note that super-resolution reconstruction is considered an ill-posed problem. Thus, regularized solutions using *a priori* constraints are usually required. Several methods that follow this idea were proposed, including the following: projection onto convex sets (POCS) based

¹ This paper contains excerpts from the M.Sc. thesis named "Uso do Algoritmo ICM Adaptativo a Descontinuidades para o Aumento da Resolução de Imagens Digitais por Técnicas de Reconstrução por Super Resolução" by the first author. This research project was financially supported by CAPES.

approaches [19][24][22]; deterministic regularized methods [10][9][3]; and probabilistic reconstruction techniques imposing the prior probability density functions (MAP reconstruction methods) [18][8]. A remarkable point is that MAP formulations using MRF prior models are considered the most flexible and realistic approaches. Even under limited prior information, these formulations allow the imposition of usual images characteristics using just neighborhood relationships [2].

In the super-resolution context, there are usually only low-resolution noisy observations. Therefore, a common constraint to be imposed is smoothness. In a MAP-MRF approach, this constraint is expressed as the prior probability of the high-resolution image, which is uniquely determined by its local conditional probabilities [1]. Besides, in MRFs, only neighboring pixels have direct interaction. Thus, the smoothness constraint can be imposed just by considering that in a neighborhood the pixel values do not change abruptly. However, despite this facility, the maximization of the joint probability usually demand high computational power. Besides, the global optimization is difficult to be computed exactly and an approximation has to be used [15]. In this context, the ICM algorithm is an interesting alternative. It is a deterministic algorithm proposed by Besag [1], which maximizes the local conditional probabilities sequentially. A remarkable point about this algorithm is the very fast convergence rate. It is important to note that image models based on a MAP-MRF formulation usually implies uniform smoothness of the image. This oversmoothness do not respect discontinuities, where abrupt changes occur. Therefore, we adopt a DA procedure for the ICM algorithm. In this way, when a discontinuity is detected, the degree of interaction is adjusted not to smooth the area.

This work presents a coherent image formation model and a MAP-MRF approach for super-resolution image reconstruction using the DA ICM algorithm. We discuss the use of the Potts-Strauss model for the *a priori* probability density function of the high-resolution image.

The PSNR method and the UIQI were used for the numerical evaluation of the proposed approach in a simulated situation. Besides, video frames with sub-pixel displacements were used for the visual evaluation. The results indicate the considerable effectiveness of our approach. Section 2 discusses some super-resolution approaches. Section 3 briefly reviews some MRF concepts and the Potts-Strauss model. Section 4 presents the proposed method. Section 5 shows some results obtained in the simulation and in the real situation. Finally, Section 6 presents some discussions about the proposed method.

2. Super Resolution Reconstruction

The super-resolution image reconstruction methodology uses a set of low-resolution observations of the same scene to reconstruct a high-resolution image. It has proved to be useful in many practical situations in which high resolution images are required. In fact, super-resolution methods use signal processing techniques to obtain the high-resolution image. Thus, it overcomes the resolution limitation inherent to acquisition devices.

Each pair of low-resolution observations must present sub-pixel displacements between them. This requirement guarantees that each low-resolution image contains additional information that can be used to increase spatial resolution. If these displacements were multiples of the low-resolution pixel size, images would contain redundant information. Figure 1 illustrates two low-resolution grids with integer displacements between each other. Note that both images contain the same information. Images presenting sub-pixel displacements can be acquired from a single camera with several captures; from multiple cameras located in different positions; by scene motion or local object movements; by vibrating imaging systems; using video frames.

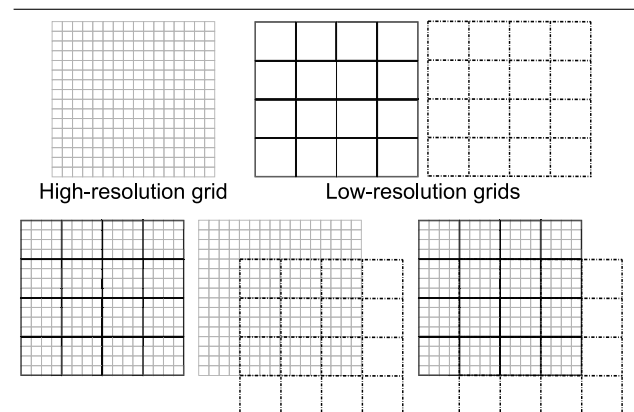


Figure 1. Integer displacements between two low-resolution grids.

According to Borman and Stevenson [2], the super-resolution methodology emerged from image interpolation methods applied for single frame resolution enhancement. As illustrated in Figure 2, more general problems have evolved from the actual methodology: spatial resolution enhancement of video sequences; temporal resolution enhancement of video sequences; and spatial-temporal resolution enhancement of video sequences. In this work we focus on the problem of reconstructing a high-resolution image, based on a set of low-resolution observations.

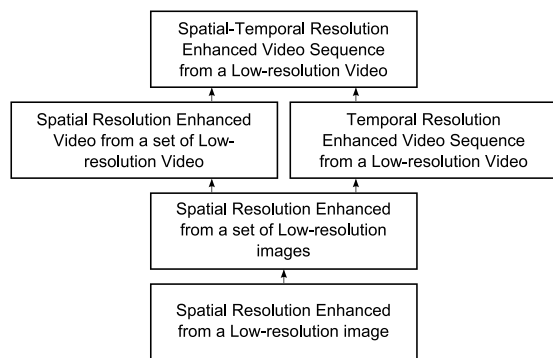


Figure 2. The hierarchy of the spatial and temporal resolution enhancement techniques [2].

Figure 3 illustrates the process of increasing the spatial resolution based on the observations. Note that to increase resolution, first of all it is necessary to identify the sub-pixel displacements among the low-resolution images. Image registration methods intend to map points in one image to the corresponding points in another image [4]. In this way, these images can be aligned with one another so that differences can be detected. Several sub-pixel image registration methods were proposed in the literature. For instance, Irani and Peleg [11] proposed a sub-pixel image registration and resolution enhancement method based on the back-projection procedure used in tomography; Xing et al. [23] proposed a multi-sensor and multi-resolution remote sensing image sub-pixel registration scheme; Guizar-Sicairos et al. [6] proposed a sub-pixel displacement estimation procedure based on the upsampled cross correlation of the reference image and each of the other observations.

Several algorithms were proposed in the last years for super-resolution image reconstruction [2][17]. In general,

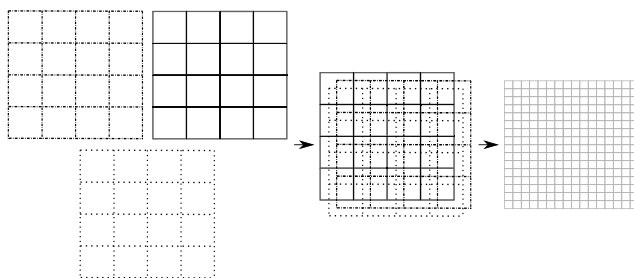


Figure 3. Spatial resolution increased based on three low-resolution grids with sub-pixel displacements.

they can be classified into two main classes: spatial and frequency domain approaches. Despite the simplicity of frequency domain approaches, they usually do not permit more general motion models. Spatial domain methods are able to work with more general observation models. Besides, they allow the inclusion of *a priori* constraints. In fact, there are usually only degraded observations and, besides increasing spatial resolution, super-resolution methods intend to reconstruct a high-quality digital image. In this sense, similar to restoration problems, super-resolution reconstruction is considered an ill-posed problem. Thus, some kind of regularization using *a priori* constraints is usually required.

POCS based approaches [19][24][22] impose prior knowledge by convex sets. This is a computational approach based on the theory that iterative projections onto convex sets converge to the intersection of all sets. In this way, given an initial estimation, the solution will respect all of the constraints represented by the sets. Despite the simplicity and flexibility of this approach, if the intersection of the sets is not a single point, there will be more than one solution. Thus, the result depends on the initial estimation. Besides, this approach demands high computational power. On the other hand, deterministic regularization methods use desired information about the solution to stabilize the inversion of the problem [10][9][3]. Smoothness is the most common constraint imposed. It assumes that, in general, images present limited high-frequency activity. However, in many cases, other priors would preserve high-frequency information in a better way. In this way, this approach models the prior information in an unfavorable way. It only includes a regularization term in the optimization function.

Probabilistic reconstruction techniques usually includes prior knowledge in a more natural way. The Bayesian MAP estimation is the most promising method. This approach uses the prior probability density function of the image to impose constraints to the solution. In this context, MRFs prior models are considered the most flexible and realistic because they allow the inclusion of images characteristics using only neighborhood relations.

3. Markov Random Fields

MAP formulations using MRF prior models are considered the most flexible and realistic approaches in the super-resolution context [2]. The MRF theory is based on contextual dependencies of physical phenomena. Let $F = \{F_1, F_2, \dots, F_m\}$ be a set of random variables defined on a regular lattice S , and each F_i takes a value $f_i \in L$. $(F_1 = f_1, F_2 = f_2, \dots, F_m = f_m)$ denotes the joint event that each F_i takes the value $f_i, i = 1, \dots, m$. In this way, F is called a random field and $f = \{f_1, f_2, \dots, f_m\}$ is a realization of the field.

The sites in S are related by a neighborhood system η , defined by

$$\eta = \{\eta_i | \forall i \in S\}, \quad (1)$$

where η_i is the set of i 's neighbors. This neighborhood has the following properties:

1. $i \notin \eta_i$;
2. $i \in \eta_{i'} \iff i' \in \eta_i$.

In the first order neighborhood system, every site has four neighbors. On the other hand, in the second order neighborhood system, every site has eight neighbors. Considering the site a , Figure 4 illustrates these neighborhood systems together with higher order systems indicated by the numbers in the third figure.

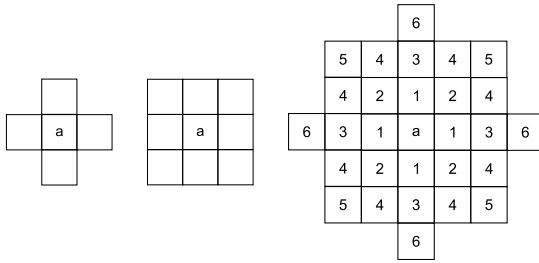


Figure 4. Neighborhood systems on a regular lattice.

A clique c is defined as a subset of sites in S that are neighbors to one another. Thus, it can consist of a single site $c = i$, a pair of sites $c = i, i'$, a triple of sites $c = i, i', i''$, and so on.

The collection of all cliques in a MRF is

$$C = C_1 \cup C_2 \cup C_3 \dots \quad (2)$$

where

$$C_1 = \{i | i \in S\}$$

$$C_2 = \{i, i' | i' \in \eta_i, i \in S\}$$

$$C_3 = \{i, i', i'' | i, i', i'' \in S \text{ are neighbors to one another}\}$$

and so on. In this way, considering the first order neighborhood system, the possible cliques are the first three cliques presented in Figure 5. The second order neighborhood system presents all the cliques showed in Figure 5.

For a discrete label set L , $P(f)$ denotes the joint probability of the realization f . F is called a MRF on S with respect to the neighborhood system η , if and only if

1. $P(f) > 0, \forall f = \{f_1, f_2, \dots, f_m\}, f_i \in L$;
2. $P(f_i | f_{S-\{i\}}) = P(f_i | f_{\eta_i})$,

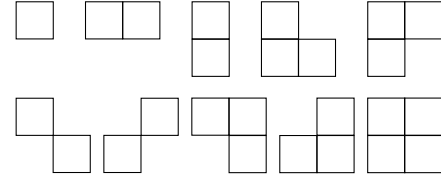


Figure 5. Cliques on a regular lattice.

where $S - \{i\}$ is the set of sites in S without i and $f_{S-\{i\}}$ is the set of labels of the sites in $S - \{i\}$.

The second property is called *Markovianity*. It depicts that in MRFs, only neighboring labels have direct interactions. In this way, a MRF is uniquely determined by its local conditional probabilities.

3.1. The MRF-Gibbs Equivalence

A random field F is said a Gibbs random field (GRF) on S according to η , if it is characterized by the Gibbs distribution

$$P(f) = Z^{-1} e^{-\frac{1}{T} U(f)} \quad (3)$$

where

$$Z = \sum_f e^{-\frac{1}{T} U(f)}. \quad (4)$$

Z is called partition function, and the energy function $U(f)$ is given by

$$U(f) = \sum_{c \in C} V_c(f), \quad (5)$$

where $V_c(f)$ are the clique potentials that characterize the interaction between neighbors.

The Hammersley-Clifford theorem established the MRF-Gibbs equivalence [7]. It stated that F is a MRF on S according to a neighborhood system η if and only if F is a GRF on S according to η . This theorem provides a simple way of specifying the joint probability.

3.2. The Potts-Strauss Model

In general, since the Hammersley-Clifford theorem, the Gibbs distribution is adopted for the image characterization in MAP-MRF approaches. However, the maximization of the joint probability usually demand high computational power and the global optimization is difficult to be computed exactly. The Potts-Strauss model can be defined by the set of all the local conditional distributions $p(f_i | f_{\eta_i})$, which are defined as

$$p(f_i | f_{\eta_i}) \sim e^{\beta \sum_{t \in \eta_i: f_t = f_i} 1} \forall i \in S. \quad (6)$$

The parameter β is often referred to as the attraction or repulsion parameter whether it is positive or negative, respectively [16]. We believe the estimation process could be op-

timized by this model together with the maximization of the local conditional probabilities sequentially.

4. The Proposed Method

In a typical MAP reconstruction context, first of all it is necessary to formulate the image formation model that relates the desired high-resolution image to the low-resolution observations and the *a priori* distribution to be adopted. Also, for the sub-pixel displacements estimation, we must define how image registration will be applied.

4.1. Image Formation Model

Consider $f[i, j]$, $0 \leq i, j \leq M$, the ideal undegraded image sampled above the Nyquist rate from the continuous scene of interest $f : \mathbb{R}^2 \rightarrow \mathbb{R}$. Following a lexicographic ordering of $f[i, j]$, an undersampled version $d[k, l]$, $0 \leq k, l \leq N$, $N \leq M$, can be modeled by

$$d = Df, \quad (7)$$

where $D[u, v]$, $0 \leq u \leq N^2$ and $0 \leq v \leq M^2$, is the down-sampling operator. In this sense, a low-resolution pixel is modeled as a weighted sum of the high-resolution pixels, in which the weights are given by the elements of operator D . Note that, according to the position of the acquisition sensors, this operator can be responsible for the presence of sub-pixel displacements among the low-resolution observations. As illustrated in Figure 6, the second low resolution grid is displaced from the previous grid by a fraction of the low-resolution pixel dimension in both horizontal and vertical directions.

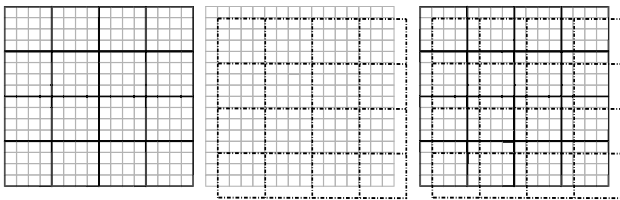


Figure 6. Sub-pixel displacements caused by the downsampling operator.

In a realistic situation, the digital image is usually blurred by the optical system during acquisition and also corrupted by noise. The blurring operator is often considered a linear space-invariant operator H , which elements are given by samples of the optical system point spread function (PSF). Thus, the blurred low-resolution image b is

given by $b = Hd$ and H is a $N^2 \times N^2$ block-circulant matrix. In this sense, a low resolution degraded version of the high-resolution image f , can be modeled by

$$g = HDf + n, \quad (8)$$

where n is additive noise.

4.2. Image Registration

The proposed method uses an initial high-resolution estimation to perform super-resolution reconstruction. This estimation can be derived according to the sub-pixel displacements between the low-resolution observations. Note that, in the alignment of the low resolution images with the high-resolution grid, several low-resolution pixels lay over one high-resolution pixel. Figure 7 illustrates the superposition of two low-resolution pixels over a fixed high-resolution pixel. Thus, in an intuitive way, each high-resolution pixel can be modeled as a composition of the low-resolution pixels influencing it. In this context, we implement the high-resolution pixel as a weighted sum of each low-resolution pixel that lay over it. The weights are defined according to the position of the high-resolution pixel under the low-resolution one.

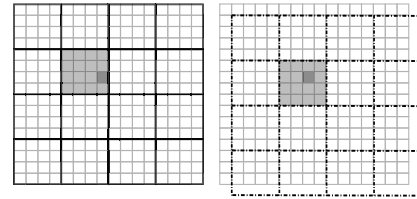


Figure 7. Low resolution pixels that lay over a fixed high-resolution pixel.

Figure 8 illustrates six of sixteen simulated images formed undersampling the a high-resolution image by four in each direction, each time starting from a different pixel within the first 4x4 block [19]. In this way, the low-resolution simulated images present sub-pixel displacements between each other. Figure 9 shows a 256x256 image used for this purpose (a); one of the 64x64 low-resolution simulated images (b); the high-resolution image reconstructed modeling each high-resolution pixel as a weighted sum of the low-resolution pixels that lay over it (c); and the bilinear interpolation of the low-resolution image (d).

For the sub-pixel displacements estimation, all observed images are compared with a reference image $g_0(x, y)$. The displacements x_0 and y_0 are estimated by minimizing the

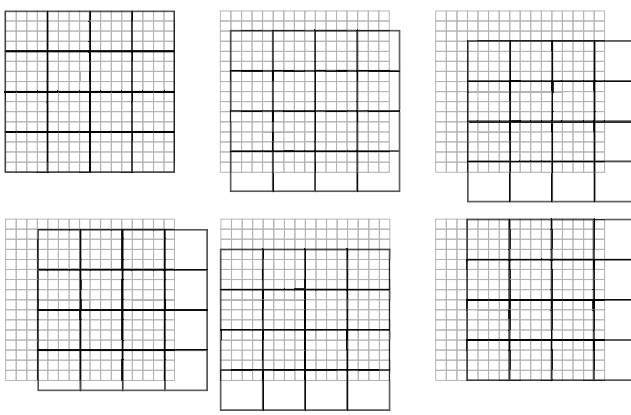


Figure 8. Six low-resolution simulated grids.

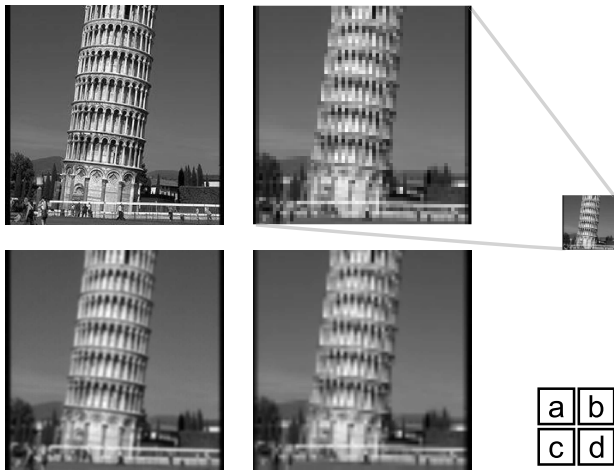


Figure 9. High-resolution image used to simulate sixteen low-resolution observations (a); first simulated image (b); reconstructed image (c); and bilinear interpolation of the low-resolution image (d).

similarity function

$$s(x_0, y_0) = \sum_{x \in X} \sum_{y \in Y} [g_0(x, y) - g_1(x - x_0, y - y_0)]^2, \quad (9)$$

where X and Y are finite sets of points.

Expanding $g_1(x, y)$ to the first term of its Taylor series, it is easy to show that x_0^* and y_0^* that minimize Equation (9) are given by

$$x_0^* = \frac{\sum_x \sum_y [(g_0 - g_1)(x, y) - x_0^* g_{1x}(x, y)] g_{1x}(x, y)}{\sum_x \sum_y g_{1x}^2(x, y)} \quad (10)$$

and

$$y_0^* = \frac{\sum_x \sum_y [(g_0 - g_1)(x, y) - x_0^* g_{1x}(x, y)] g_{1y}(x, y)}{\sum_x \sum_y g_{1y}^2(x, y)} \quad (11)$$

where $g_{1x}(x, y)$ and $g_{1y}(x, y)$ are the first derivatives of $g_1(x, y)$ in relation to x and y , respectively. This method only considers a set of globally translated and low-resolution observations. However, this assumption has been proved not to be so restrictive since we are considering very slight displacements between the images [11].

4.3. Bayesian Formulation for Super-Resolution

It is well known that high-resolution image reconstruction is an ill-posed problem. Thus, some kind of regularization is required to reach a good approximation of the original scene. In fact, a high-resolution estimation, $\hat{f}$, must be reconstructed considering a set of low-resolution observations g_k , $k = 1, \dots, q$, each of them modeled by Equation (8). A Bayesian formulation usually provides a flexible and realistic way of imposing *a priori* constraints to the estimation. In this sense, the prior information is expressed as the prior probability of the high-resolution image and the MAP solution decides for the estimation that maximizes the conditional probability density of f given all the observations,

$$\hat{f} = \arg \max_f \{p(f|g)\}, \quad (12)$$

where g is composed by all of the low-resolution observations g_k , $k = 1, 2, \dots, q$.

4.4. The Discontinuity Adaptive Approach

MAP-MRF formulations usually imply a uniform smoothness of the image. According to Li [14], since Geman and Geman [5] introduced the *line fields* idea, the application of the smoothness constraint while preserving discontinuities has been an active research issue in the image processing context. Discontinuity adaptive methods control the interaction between neighbors in such a way that the degree of interaction is adjusted when a discontinuity is detected.

The discontinuity detection is performed by an adaptive interactive function (AIF). Let k be the difference between a pixel and each of its neighbors. According to the value of this difference, the AIF has to distinguish noise from the presence of a discontinuity. For that, this function is convex in the interval $[-B, B]$, increasing monotonically with k to smooth out the noise. Outside this interval, the function is non-convex, decreasing as k increases and becoming zero as $k \rightarrow \infty$ [14].

We adopt an AIF parameterized by γ given by

$$A_\gamma(x) = \gamma - \frac{\gamma}{1 + \frac{x^2}{\gamma}}, \quad (13)$$

which is convex in the interval $B_\gamma = (-\sqrt{\frac{\gamma}{3}}, \sqrt{\frac{\gamma}{3}})$. Figure 10 shows the qualitative shape of the adopted AIF function.

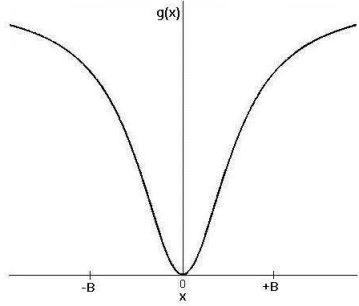


Figure 10. AIF adopted to the discontinuities detection.

4.5. The Discontinuity Adaptive ICM Algorithm

The Iterated Conditional Modes algorithm was proposed by Besag [1] as a computationally feasible alternative for the MAP estimation. It uses a *greedy* strategy to maximize the local probabilities sequentially. The method is based on the local posterior distribution

$$p(f_i|g, f_{\eta_i}) \sim p(g_i|f_i) \cdot p(f_i|f_{\eta_i}), \quad (14)$$

given the observations g and the current values in the neighborhood η_i . As discussed in Section 3.2, the Potts-Strauss model is adopted as the prior distribution $p(f_i|f_{\eta_i})$.

We assume that the conditional probability density function $p(g_i|f_i)$ is given by

$$p(g_i|f_i) = \frac{1}{\sigma\sqrt{2\pi}} \cdot e^{-\frac{(g_i - m_i)^2}{2\sigma^2}}, \quad (15)$$

where g_i is the weighted sum of the low-resolution pixels that lay over the high-resolution pixel, $m_i = \frac{1}{C} \left(\sum_{j \in \eta_i} f_j \right) + \frac{1}{C} f_i$ and $C = \sharp\eta_i + 1$.

In each iteration, the ICM algorithm updates f_i by the value that maximizes $p(g_i|f_i) \cdot p(f_i|f_{\eta_i})$, for each $i \in S$. In our DA formulation, first of all, the AIFs $A_\gamma(i)$, for all $i \in S$, are calculated. Then, $p(g_i|f_i) \cdot p(f_i|f_{\eta_i})$ is maximized considering only the values $i' \in \eta_i$ which $A_\gamma(i')$ is close to $A_\gamma(i)$.

In this context, the DA ICM algorithm is given by:

1. Choose an initial high-resolution estimation;

2. For i from 1 to M^2 , update f_i by one of the values $i' \in \eta_i$ that maximizes

$$p(g_i|f_i)p(f_i|\hat{f}_{\eta_i})$$

and $A_\gamma(i')$ is close to $A_\gamma(i)$.

3. Repeat item (2) τ_{iter} times.

τ_{iter} is the maximum number of iterations.

We used the image reconstructed using the low-resolution observations as discussed in Section 4.2, as the initial high-resolution estimation.

5. Results

Peak signal-to-noise ratio (PSNR) method and the Universal Image Quality Index (UIQI) were used for the numerical evaluation of the proposed approach in a simulated situation. Besides, in a real situation, video frames with sub-pixel displacements were used for the visual evaluation.

5.1. Simulation Description

The proposed method was evaluated in a simulated situation where sixteen low-resolution images were generated according to the image formation model discussed on Section 4.1. The image adopted as the high-resolution image to be reconstructed was undersampled as discussed on Section 4.2. In this way, the simulated observations present sub-pixel displacements between each other. Then, the images were convolved with a 3x3 uniform rectangular kernel to simulate the blur due to the image process acquisition and corrupted by additive and independent Gaussian noise at 30 dB. Figure 11 shows the 512x512 high-resolution image used (a) and the reference 128x128 low-resolution observation (b).

Figure 12 shows the high-resolution registered image as discussed on Section 4.2 (a); the bilinear interpolation of the reference low-resolution image (b); the high-resolution estimation reconstructed using the proposed method without the DA procedure (c); and the result using the DA constraint (d). As one can see, the registration procedure is able to give better results when compared with the interpolated image. Although in this simulation we have knowledge of the actual displacements between each low-resolution image and the reference image, we have estimated the displacement values. We note that the proposed method for sub-pixel registration has demonstrated to be very accurate in all conducted experiments.

In the simulations, the algorithm was initialized with the registered image in Figure 12 and the β parameter in Equation (6) was found following the procedure proposed in [12] for the second order neighborhood system. Also, in this experiment we do not take into account the blur from the optical system in the restoration process. From the presented



Figure 11. High-resolution image used in the simulation (a) and one of the low-resolution images (b).

results, we can see that the algorithm was able to improve the quality of the initial high-resolution estimation. We also note that in most of the experiments, the algorithm had fast convergence rate, where 5 or 6 iterations were sufficient to produce good results.

The oversmoothness inherent to the MAP-MRF formulation can be identified by uniform areas in the image. Figure 13 shows a zoom of the image reconstructed without imposing the DA constraint (a) and the image reconstructed with the DA procedure (b). Note that the DA approach avoids the formation of uniform areas in the image, preserving the details.

We used the Universal Image Quality Index proposed by Wang and Bovik [21] for the numerical evaluation. This index is given by

$$Q = \frac{4\sigma_{fq}\bar{f}\bar{q}}{(\sigma_f^2 + \sigma_q^2)[(\bar{f})^2 + (\bar{q})^2]}, \quad (16)$$

where q is the image to be compared with the original image f and

$$\bar{f} = \frac{1}{M} \sum_{i=1}^M f_i, \bar{q} = \frac{1}{M} \sum_{i=1}^M q_i, \quad (17)$$

$$\sigma_f^2 = \frac{1}{M-1} \sum_{i=1}^M (f_i - \bar{f})^2, \quad (18)$$

$$\sigma_q^2 = \frac{1}{M-1} \sum_{i=1}^M (q_i - \bar{q})^2, \quad (19)$$

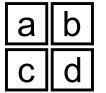


Figure 12. Registered image used in the simulation (a); Bilinear interpolation of the reference low-resolution image (b); high-resolution estimation reconstructed without the DA procedure (c); and the result using DA constraint (d).

$$\sigma_{fq} = \frac{1}{M-1} \sum_{i=1}^M (f_i - \bar{f})(q_i - \bar{q}). \quad (20)$$

Q assumes values in the interval $[-1, 1]$. We believe this quality index is more appropriate to the super-resolution context since it models any distortion as a combination of three different factors: loss of correlation, luminance distortion, and contrast distortion.

Irani and Peleg's method [11], which presented the most important results in the super-resolution context, is used for comparison purpose. Since the proposed method does not include the deblurring process, the quality index was calculated using the blurred image as the original one. Table 1 shows the resulting quality indices. Indeed, the values shown are in agreement with a visual evaluation of the images. The proposed algorithm with the DA procedure presented a very similar evaluation compared with Irani and Peleg's method.

We also used the normalized mean squared error



Figure 13. Zoom of the image reconstructed without the imposition of the DA constraint (a) and the image reconstructed with the DA procedure (b).

(NMSE), given by

$$NMSE = \frac{\|f - q\|^2}{\|f\|^2} = \frac{\sum_{i=1}^M (f_i - q_i)^2}{\sum_{i=1}^M f_i^2} \quad (21)$$

to evaluate the same images. Table 1 shows that the results were very similar to the evaluation by the UIQI.

	UIQI	NMSE
Registered Image	0.9660	0.1439
ICM	0.9331	0.1709
DA ICM	0.9875	0.0577
Bilinear interpolation	0.9668	0.1445
Irani-Peleg	0.9897	0.0209

Table 1. UIQI and NMSE using the blurred image.

5.2. Case Study

In a real situation, we used video frames with sub-pixel displacements between each other to visual evaluate the proposed method. Figure 14 shows four 128x128 video frames. Considering the upsampling/downsampling factor 2, Figure 15 shows the 256x256 reconstructed image.

6. Concluding Remarks

We have presented an efficient algorithm for super-resolution image reconstruction based on a Markov random field where we used a DA procedure with the Iterated



Figure 14. Frames of a video, containing sub-pixel displacements between each other.

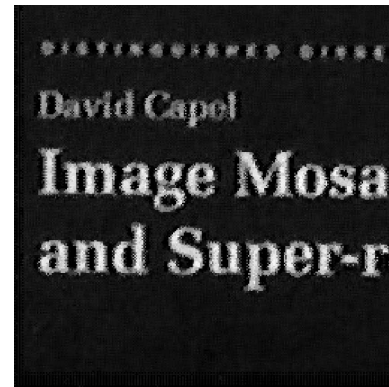


Figure 15. High-resolution reconstructed image.

Conditional Modes algorithm for computing the maximum *a posteriori* conditional probability. Indeed, the results demonstrate that the algorithm can be extremely efficient in a super-resolution reconstruction framework where the method has demonstrated good performance both in visual accuracy and computational cost. We also note that, although we do not address the image deblurring procedure in this work, it can be easily incorporated into the proposed algorithm. In future works, we intend to make additional experiments in order to verify the accuracy of the proposed method when compared with the Irani-Peleg algorithm and also considering different levels for the signal to noise ratio in the observations. We also intend to test the algorithm with other models for the *a priori* probability density function of the actual image.

References

- [1] J. BESAG. Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society B*, 36(2):192–236, 1974.
- [2] J. L. BORMAN and R. STEVENSON. Spatial resolution enhancement of low-resolution image sequences - A comprehensive review with directions for future research. Technical report, University of Notre Dame, Notre Dame, IN, 1998.
- [3] N. K. BOSE, S. LERTRATTANAPANICH, and J. KOO. Advances in superresolution using L-curve. *Proc. Int. Symp. Circuits and Systems*, 2:433–436, 2001.
- [4] L. G. BROWN. A survey of image registration techniques. *ACM Computing Surveys*, 24(4):325–376, 1992.
- [5] S. GEMAN and D. GEMAN. Stochastic relaxation, Gibbs distribution, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741, 1984.
- [6] M. GUIZAR-SICAÍROS, S. T. THURMAN, and J. R. FIENUP. Efficient subpixel image registration algorithms. *Optics Letters*, 33(2):156–158, 2008.
- [7] J. M. HAMMERSLEY and P. CLIFFORD. Markov field on finite graphs and lattices. 1971.
- [8] R. C. HARDIE, K. J. BARNARD, and E. E. ARMSTRONG. Joint MAP registration and high-resolution image estimation using a sequence of undersampled images. *IEEE Trans. Image Processing*, 6:1621–1633, 1997.
- [9] R. C. HARDIE, K. J. BARNARD, J. G. BOGNAR, E. E. ARMSTRONG, and E. A. Watson. High-resolution image reconstruction from a sequence of rotated and translated frames and its application to an infrared imaging system. *Opt. Eng.*, 37(1):247–260, 1998.
- [10] M. C. HONG, M. G. KANG, and A. K. KATSAGGELOS. A regularized multichannel restoration approach for globally optimal high resolution video sequence. *SPIE VCIP*, 3024:1306–1317, 1997.
- [11] M. IRANI and S. PELEG. Improving resolution by image registration. *CVGIP: Graphical Models and Image Processing*, 53:231–239, 1991.
- [12] A. L. M. LEVADA, N. D. A. MASCARENHAS, and A. TANNÚS. Pseudolikelihood Equations for Potts MRF Model Parameter Estimation on Higher Order Neighborhood Systems. *IEEE Geoscience and Remote Sensing Letters*, 5(3), 2008. (accepted for publication).
- [13] C. L. LEVIN and E. J. HOFFMAN. Calculation of positron range and its effect on the fundamental limit of positron emission tomography system spatial resolution. *Phys. Med. Biol.*, 44:781–799, 1999.
- [14] S. Z. LI. On discontinuity-adaptive smoothness priors in computer vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAM-17(6):576–586, june 1995.
- [15] S. Z. LI, Y. H. HUANG, and J. S. FU. Convex MRF potential functions. *Proceedings of the IEEE International Conference on Image Processing*, 2:296–299, 1995.
- [16] N. D. A. MASCARENHAS and A. C. FRERY. SAR image filtering with the ICM algorithm. *International Geoscience and Remote Sensing Symposium*, pages 2185–2187, August 1994.
- [17] S. C. PARK, M. K. PARK, and M. G. KANG. Super-resolution image reconstruction: a technical overview. *IEEE Signal Processing Magazine*, 20(3):21–36, 2003.
- [18] R. R. SCHULTZ and R. L. STEVENSON. Improved definition image expansion. *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 3:173–176, 1992.
- [19] H. STARK and P. OSKOU. High-resolution image recovery from image-plane arrays, using convex projections. *Journal of Optical Society America A*, 6(11):1715–1726, 1989.
- [20] R. Y. TSAI and T. S. HUANG. Multi-frame image restoration and registration. *Advances in Computer Vision and Image Processing*, pages 317–339, 1984.
- [21] Z. WANG and A. C. BOVIK. A Universal Image Quality Index. *IEEE Signal Processing Letters*, 9(3):81–84, 2002.
- [22] F. WHEELER, R. HOCTOR, and E. BARRETT. Super-resolution image synthesis using projections onto convex sets in the frequency domain. In *Computational Imaging III. Edited by Bouman, Charles A.; Miller, Eric L. Proceedings of the SPIE*, volume 5674, pages 479–490, 2005.
- [23] S. XING, Q. XU, and D. MA. Multi-sensor Satellite Image Sub-pixel Registration. *Fourth International Conference on Image and Graphics, ICIG 2007*, pages 713–718, 2007. IEEE Computer Society.
- [24] S. YEH and H. STARK. Iterative and One-step Reconstruction from Nonuniform Samples by Convex Projections. *Journal of the Optical Society of America A*, 7:491–499, 1990.

Mapeamento e Monitoramento Ambiental Usando Imagens Aéreas de Pequeno Formato

Natal Henrique Cordeiro

Bruno Motta de Carvalho

Luiz Marcos Garcia Gonçalves

Universidade Federal do Rio Grande do Norte, Laboratório Natalnet

Campus Universitário, Lagoa Nova, Natal, RN, Brasil

natalhenrique@gmail.com, motta@dimap.ufrn.com, lmarcos@dca.ufrn.br

Resumo

No presente trabalho ¹ descrevemos uma metodologia para a construção de mosaicos georreferenciados a partir de imagens aéreas de pequeno formato, ou SFAI, consideradas não controlada. Técnicas de Estéreo-Fotogrametria e de Visão Computacional são aplicadas para auxiliarem em resultados de qualidade e confiabilidade. As imagens são obtidas através de uma câmera digital comum acoplada a um helicóptero aeromodelo. Técnicas para eliminar distorções comuns são aplicadas e a orientação relativa dos modelos é realizada baseada na geometria perspectiva. Pontos reais do terreno são usados para executar a orientação absoluta, além de uma definição de escala e de um sistema de coordenadas que relaciona a imagem medidas no terreno. O mosaico será manipulado em um SIG (Sistema de Informação Geográfica), fornecendo informação útil para diferentes tipos de usuários tais como pesquisadores, instituições governamentais, pescadores e empresas de turismo. Resultados são mostrados, ilustrando a aplicabilidade do sistema. A principal contribuição é a geração de mosaicos georreferenciados usando imagens aéreas de pequeno formato, o que tem sido pouquíssima explorada em projetos de cartografia. A arquitetura proposta apresenta uma solução viável e bem barata quando comparada com sistemas que utilizam imagens controladas.

Abstract

We propose a technique that uses small format aerial images, or SFAI, considered as not controlled, and stereo-photogrammetry techniques to construct georeferenced mo-

saics. Images are obtained using a simple digital camera coupled to a radio controlled (RC) helicopter. Techniques for removing common distortions are applied and the relative orientation of the models are recovered using perspective geometry. Ground truth points are used to get absolute orientation, plus a definition of scale and a coordinate system which relates image measures to the ground. The mosaic is read into a GIS system, providing useful information to different types of users, such as researchers, government officers, fishers and tourism enterprises. Results are reported, illustrating the applicability of the system. The main contribution is the generation of georeferenced mosaics using SFAIs, what has not been widely explored previously in cartography projects. The proposed architecture presents a viable and much less expensive solution, when compared to systems using controlled pictures.

1. Introdução

Neste trabalho, propomos uma metodologia visando produzir mosaicos georreferenciados usando imagens aéreas de pequeno formato (small format aerial images - SFAI). Um mosaico nada mais é que uma composição de várias fotos ou imagens adjacentes, visando ter uma visão mais ampla (e na mesma escala) de uma determinada cena, no caso, com mais realismo que os mapas. Imagens aéreas de pequeno formato resumem-se à imagens capturadas com uma câmera digital comum, de baixa resolução.

Esta metodologia envolve a realização de uma série de etapas, desde a retirada de erros e distorções causados pelo processo (ou pelo sistema) de aquisição, até a colagem final das imagens. No processo, devem ser aplicadas transformações, visando reconstruir a orientação relativa das imagens, umas às outras, e absoluta, em relação à cena; Finalmente,

¹ Este documento contém informações da dissertação de mestrado denominada "Construção de Mosaicos Georreferenciados Usando Imagens Aéreas de Pequeno Formato".

a definição e a adoção de uma escala e de um sistema de representação, em relação à cena (terreno) imageada.

A metodologia para SFAI, em linhas gerais, segue aproximadamente o mesmo processo tradicional de cartografia, e do caso de se usarem imagens de grande formato (big format aerial images - BFAI), sendo que esta última, permite uma precisão significativamente melhor, uma vez que o sistema de aquisição é controlado.

Quando mencionamos sobre um sistema de aquisição controlado, significa dizer que as imagens podem ser tomadas em um momento desejado e a uma altura significativa, em que permite diminuir substancialmente as distorções geradas pelas variações de vôo e de vista perspectiva comparado com as obtidas usando uma câmera digital comum acoplada a um helicóptero aeromodelo. É importante ressaltar que estas imagens(BFAI) possuem alta resolução e não geram distorções oriundas da curvatura da lente como comparado à uma câmera digital comum. Dessa forma, existe a possibilidade de se produzir orto-imagens, ou imagens muito próximas de uma projeção ortográfica.

Isso facilita muito o processo de correção das distorções. Neste trabalho, o uso de imagens aéreas de pequeno formato implica o cálculo de todos os coeficientes das transformações(orientação relativa e absoluta) citadas acima, antes da geração do mosaico. Um dos principais problemas na geração de mosaicos georreferenciados refere-se à qualidade das imagens adquiridas.

No processo cartográfico tradicional, imagens adquiridas por técnicas controladas são utilizadas, sendo que este processo, no entanto, mesmo com a introdução de etapas automatizadas pelo uso de tecnologia recente, é ainda caro, pois envolve também o uso de câmeras especiais e o emprego de aviões específicos para aquisição dos dados.

O uso de imagens de satélite é uma alternativa, porém menos precisa que as imagens obtidas a partir de aviões, por serem estas últimas adquiridas de altitudes menores, com os sensores mais próximos da cena, portanto em escalas maiores.

O uso de câmeras comuns e helicópteros ou aviões aeromodelos tornam o processo muito mais barato. Claro, ocorre também uma perda na precisão do controle de georreferenciamento do produto final, mas há o apelo de que a precisão visual é muito melhor que as outras duas, sendo por isso aceitável em algumas aplicações, tais como em monitoramento ambiental, inspeção de locais, turismo, entre outras. No presente trabalho, buscamos justamente uma nova abordagem para a geração de mosaicos georreferenciados, que se encaixe nesta terceira categoria.

Usamos imagens aéreas de pequeno formato (Small Format Aerial Images - SFAI), obtidas por uma câmera digital Sony comum, acoplada ao helicóptero aeromodelo mostrado na Figura 1. O helicóptero é guiado remotamente, sobrevoando áreas de interesse para a aquisição de seqüên-

cias de vídeo, que são posteriormente quebradas em quadros que, finalmente processados, produzem o mosaico final.



Figura 1. Helicóptero Aeromodelo.

2. Trabalhos Relacionados

Neste trabalho, como são disponibilizadas apenas imagens aéreas de pequeno formato sem um maior controle, torna-se indispensável adaptar e/ou criar metodologias adequadas para trabalhar com este formato de imagem. Sistemas como o ArcView da ESRI, ERDAS da Leica Geosystem, Regeemy e Spring do INPE, entre outros permitem gerar mosaicos de qualidade, usando a técnica de registro em imagens aéreas de grande formato (BFAI).

Convém ressaltar que, por definição, imagens do tipo BFAI são controladas. Imagens controladas não apresentam distorções, como as geradas pelo processo usando câmera digital comum e pelas orientações do vôo, principalmente quando se obtêm imagens de escala grande. Ainda equipamentos que adquirem BFAI permitem gerar conhecimentos prévios, tais como de que forma e em que momento adquirir uma imagem. Desta maneira, as imagens de grande formato, controladas, podem possuir algumas distorções. Mas são mínimas e, em alguns casos consideradas livres de distorções e até mesmo já georreferenciadas, dependendo da aplicação. Isso facilita substancialmente o processo, o que não ocorre no tipo de dados que estamos tratando.

As imagens obtidas com câmeras digitais comuns têm o ângulo de abertura maior em relação às fotografias aéreas controladas ou às imagens de satélite, o que causa maior distorção radial e radiométrica.

Ainda devido à baixa altitude, geralmente necessita-se de inúmeras imagens a partir de diferentes pontos de vista, sendo que a projeção perspectiva caracteriza as imagens

dessas câmeras, em contraposição à projeção ortogonal, ou muito próxima desta, conseguida usando câmeras fotogramétricas ou imagens de satélite,

Notamos que o processo de mosaicamento será utilizado, justamente quando a área de estudo é maior que a imagem disponibilizada pela câmera, visando cobrir melhor a área de interesse. Além dos sistemas comerciais e outros acima citados, vimos na literatura alguns que projetos usam técnicas de registro de imagem para correção geométrica antes do processo de geração do mosaico [6, 17, 8, 7, 16].

As técnicas de registro de imagens propiciam corrigir pequenas distorções geométricas, presentes em imagens de grande formato, principalmente pela baixa resolução, mesmo com distorções irrelevantes, usando uma variedade de pontos de controle (pontos no terreno). Observamos que este modelo é diferente para as imagens de pequeno formato, cujos sistemas de aquisição não têm os mesmos requisitos.

No presente projeto, estaremos empregando SFAIs não controladas e com vários tipos de distorções e, o que complica mais, usaremos poucos pontos de controle devido à região de trabalho ser marinha, dificultando a sua aquisição. Observamos que o mosaico final é mais rico de detalhes, por usarmos imagens com escala maior, obtidas mais próximas da superfície terrestre. Por outro lado, projetos que utilizam BFAI permitem monitorar áreas maiores, apesar do custo de aquisição ser maior.

O Global Rain Forest Mapping Project (GRFM) [6] é um projeto gerenciado pela Agência Nacional de Desenvolvimento do Espaço do Japão (NASDA). Tem por objetivo gerar mosaicos georreferenciados a partir de BFAIs adquiridas por satélites. São aplicadas técnicas para correção radiométrica nas imagens, a fim de melhorar a qualidade e de não perder informações nas regiões de sobreposição do mosaico. O trabalho apresentado por Xiao e seus colaboradores [16] também se preocupa com problemas ocorridos devido à iluminação, criando um método que combina ruídos na busca de qualidade. O método criado por Albrecht e Michaelis [1] apresenta técnicas de estéreo-fotogrametria com melhorias na resolução espacial dos pares estereoscópicos, a fim de reduzir erros em regiões com relevo acentuado.

Já Nicolas [10] criou um método de mosaicamento, utilizando um critério de minimização analítica, para determinação de coeficientes baseados na variação temporal das imagens de fundo, com foco em corrigir distorções geométricas. Zomet [18] propõe novos métodos com melhores formas de correção geométrica de distorções resultantes da transformação 2D para 3D. Este projeto utiliza cálculos analíticos para resolução do problema, e a sua relação com o nosso projeto (NatalGIS) é a busca de disponibilizar informações confiáveis após correção das distorções.

Syuhei [12] descreve uma nova abordagem para o posicionamento de helicópteros aeromodelos operados de forma

autônoma. Métodos para obter a terceira dimensão e a altitude do helicóptero são realizados através de sensores, GPS (Global Positioning System), IMU (Inertial Measurement Unit) e imagens, que são utilizadas para permitir que o sistema de controle do helicóptero seja autônomo. Metodologias para a detecção de pontos no terreno são usadas para a solução do problema de posicionamento.

No trabalho em questão [12], o helicóptero aeromodelo não é utilizado para mapeamento da área ou para gerar mosaicos georreferenciados. Entretanto as técnicas utilizadas são interessantes para aplicações que podem usar helicóptero aeromodelo e para futuros trabalhos relacionados ao presente projeto.

3. Construção de mosaicos georreferenciados usando SFAIs

Como visto anteriormente, o processo de construção de mosaicos georreferenciados é dificultado pelo uso de imagens aéreas de pequeno formato (SFAI). Além disso, o ambiente escolhido (região marinha) para mapeamento neste projeto é muito dinâmico, sofrendo mudanças devido a vento, maré e iluminação (sol), aumentando a probabilidade de ocorrência de erros nas imagens. Também ocorrem variações de posição e orientação do helicóptero durante o voo, que podem gerar erros imprevisíveis. Outras distorções geodésicas, como as geradas pela curvatura da terra e a variação do relevo, são insignificantes neste projeto, podendo ser desprezadas. Porém distorções causadas pelos parâmetros intrínsecos da câmera digital usada, tais como a radial e radiométrica devem ser retiradas.

Gonçalves [5] esclarece que, no processo de cartografia tradicional, as condições de clima (vento, etc) e pilotagem não permitem que o avião mantenha sua altitude constante, bem como a velocidade e horizontalidade das asas. Esses, entre outros fatores, afetam a obtenção de um modelo "ideal" para a estéreo-fotogrametria. Ou seja, mesmo usando dados supostamente controlados (no caso tradicional), o que se obtém é um modelo mais ou menos próximo do ideal, com pequenas distorções. Isso se agrava ao usar dados não controlados, ou seja, imagem sem nenhum controle na sua aquisição como é o caso no presente trabalho.

As câmaras de pequeno formato podem produzir imagens instáveis em relação à iluminação e à geometria, não somente pelas distorções radial e radiométrica, mas também pela instabilidade do voo, freqüente no helicóptero aeromodelo, no momento da aquisição de imagens. O fato de a câmara digital comum não se manter ortogonalmente à superfície terrestre é um outro problema que produz mais distorções. Desta forma, são obtidas várias imagens, em vista perspectiva, o que gera outras distorções geométricas na transformação da imagem de mundo (3D) para a imagem digital (2D).

O processo de criação do mosaico resume-se à realização das fases descritas a seguir: Calibração de câmera, correção das distorções radial e radiométrica, reconstrução a partir de estéreo-fotogrametria, geração do mosaico georreferenciado;

3.1. Calibração da câmera e correção da distorção radial e radiométrica

Quando se pretende obter informação tridimensional de uma cena ou objeto utilizando Visão Computacional, é geralmente necessário determinar um modelo matemático da câmera utilizada. No presente projeto, usamos o método Tsai [14] para a etapa de calibração de câmera. O método usa um esquema de transformação do sistema de coordenadas de mundo para o sistema de coordenadas de imagem. No método tradicional, são determinados 11 parâmetros, sendo 5 deles intrínsecos (f (distância focal), κ_1 (coeficiente de distorção radial da lente de 1ª ordem), C_x e C_y (coordenadas de centro de distorção radial da lente), S_x (fator de escala)) e 6 extrínsecos (R_x , R_y e R_z (ângulos de rotação para a transformação entre as coordenadas nos frames de mundo e de câmera), T_x , T_y e T_z (componentes de translação para a transformação entre as coordenadas nos frames de mundo e de câmera)) [14]. Além destes parâmetros, neste trabalho, determinamos também o coeficiente de distorção radial de 2ª ordem da lente, conhecido como κ_2 , para o processo de retirada de distorção radial e também o ângulo α (para retirada de distorção radiométrica, vista adiante). Após realizar a calibração da câmera, podemos corrigir a distorção radial (equação 1) e radiométrica (equação 2) usando as equações abaixo [13].

$$\begin{aligned} x &= x_d(1 + \kappa_1 r^2 + \kappa_2 r^4), \\ y &= y_d(1 + \kappa_1 r^2 + \kappa_2 r^4), \\ r &= \sqrt{x_d^2 + y_d^2}, \end{aligned} \quad (1)$$

onde, x_d e y_d são pontos na imagem distorcida. A incógnita r é a distância do centro da imagem até o pixel. O κ_1 e κ_2 são os coeficientes de distorção descobertos na etapa de calibração de câmera.

$$E(p) = L(p) \left[\frac{\pi}{4} \left(\frac{d}{z} \right)^2 \cos^4 \alpha \right]. \quad (2)$$

Convém ressaltar que a iluminação na imagem P cresce o mesmo que a quarta potência do cosseno do ângulo formado entre o raio principal que chega a P e o eixo ótico. Em caso de pequena abertura, este efeito pode ser negligenciado; então, a irradiância na imagem pode ser entendida como uniformemente proporcional à radiância da cena sobre todo o plano da imagem [13].

3.2. Estéreo-Fotogrametria

Na aplicação em monitoramento ambiental, foco deste trabalho, temos recobrimento tanto longitudinal (pelo menos 70%) quanto lateral (pelo menos 30%) entre as imagens que farão parte do mosaico e cada imagem é adquirida de uma posição diferente. Isso propicia que técnicas de reconstrução estéreo possam ser empregadas visando melhorar ainda mais a qualidade do mosaico final. O principal problema das técnicas de reconstrução a partir de imagens estéreo é descobrir quais pontos em cada imagem correspondem às projeções de um mesmo ponto da cena. Este problema é mais conhecido como *matching* [9], e o denominaremos simplesmente de correspondência, sendo ele a etapa mais demorada e uma das mais estudadas em Visão Estéreo.

Em nossa aplicação, depois de determinadas as correspondências de todos os pixels das imagens, esta informação é utilizada na construção do mosaico. A profundidade de cada pixel pode ser determinada em relação a um referencial fixo, por triangulação, em relação às câmeras. Essa profundidade pode ajudar a distinguir as características ou atributos de um dado pixel que aparece em mais de uma imagem. Note que, no pior caso, uma média entre os atributos pode ajudar a minimizar problemas de erros das imagens devidos ao processo de aquisição.

A correspondência entre as imagens pode ser feita por área [4] ou atributo [9]. Neste trabalho, a correspondência por área poderia ser usada com algumas simplificações observadas adiante. Esse tipo de operação é realizado com aplicação de operadores de correlação cruzada normalizada (ou simplesmente correlação) ou então, pela soma do quadrado das diferenças (SSD) [2]. A SSD é mais rápida de ser calculada do que a correlação, mas não é imune a variações de contraste e brilho nas imagens, problemas que não afetam a correlação cruzada normalizada, que pode ser dada pela Equação 3:

$$r_{x,y} = \frac{n \sum (x_i y_i) - \sum (x_i) \sum (y_i)}{\sqrt{n \sum (x_i^2) - (\sum x_i)^2} \sqrt{n \sum (y_i^2) - (\sum y_i)^2}}. \quad (3)$$

onde n é o número de amostras em cada sinal. Para o *matching*, a correlação é restrita a uma região (janela de comparação) de cada imagem, sendo n o número de pixels dessa janela.

3.3. Orientação Relativa

Uma simplificação adotada neste trabalho, com o objetivo de reduzir a complexidade do modelo acima, é considerar que o relevo da região é plano. Essa simplificação se

justifica pelo fato de ser uma região marinha com diferenças mínimas de nível do fundo (variando entre zero e dois metros), o que é desprezível em relação à altitude de voo.

Dessa maneira, distorções causadas pelo relevo podem ser desconsideradas. Isto simplifica substancialmente o processo de reconstrução, permitindo realizar a orientação relativa entre cada par de imagens, [15] usando 6 pontos apenas, não exigindo, portanto, que a correspondência estéreo, o gargalo do processo de reconstrução estéreo [9] seja determinada para todos os pixels de cada par de imagens. Mais especificamente, os princípios de estéreo-fotogrametria são empregados na fase de orientação relativa dos modelos produzidos por cada par de imagens consecutivas, visando determinar as relações espaciais que o helicóptero possuía no momento da tomada de cada imagem. A posição aproximada do momento da tomada das imagens é dada pelo GPS de bordo, facilitando a determinação de uma primeira aproximação para os parâmetros de orientação exterior.

Com essas simplificações, com apenas 6 pares de pontos conhecidos (pontos de Von Grüber) em cada modelo (entre cada par de imagens), uma boa precisão pode ser conseguida na determinação de coeficientes das transformações que deverão retirar as distorções causadas pelo posicionamento e orientação não conhecidos do helicóptero. Dadas duas imagens da mesma cena, supondo-se uma à direita (r) e outra à esquerda (l) sendo p_l e p_r pixels nessas imagens, o que se deseja é determinar a transformação que leva os pixels de uma imagem p_r na outra p_l . A princípio, esta transformação envolve mais que uma transformação 2D, uma vez que os pontos homólogos são projeções de pontos 3D na cena.

3.4. Orientação Absoluta

Para o georreferenciamento em si (determinação de escala e referência a um sistema de coordenadas) são determinados a priori, na região a ser imageada, pontos de controle, cujas coordenadas são determinadas por GPS. Neste projeto, poucos pontos de apoio de campo serão determinados, o que dificulta o processo de georreferenciamento usando SFAI. Essa complexidade é existente por várias destas imagens não presenciarem os pontos previamente escolhidos no terreno. Usando técnicas de aerotriangulação [15], as coordenadas (Geográficas, UTM) conhecidas (dos Pontos de controle) são estendidas para os pontos determinados pelo processo de orientação relativa.

A partir desses, em caso de não assunção de um modelo de relevo plano, seria possível estender a todos os outros pontos de todas as imagens, gerando assim coordenadas de terreno, referenciadas em relação a um sistema de coordenadas, para todo o mosaico [5]. Note que cada modelo (par de imagens) pode ser ligado ao posterior via uma das imagens comum a dois modelos adjacentes.

Estender as coordenadas dos pontos de controle significa usar este recobrimento para extrapolar as coordenadas de uma imagem a outra. Note que um erro inerente ao processo de digitalização aparece aqui. Esse erro pode ser minimizado através de outros cálculos, a partir de um outro ponto de terreno (Obtido pelo GPS) presente em uma outra imagem selecionada para o mosaico. Dessa forma, extrapolaríamos novamente as coordenadas das imagens.

A técnica tradicional de aerotriangulação é readaptada neste trabalho, usando mínimos quadrados, para minimizar erros no processo de determinação de coordenadas para os pontos de cada modelo. Ao final, obtêm-se, por uma técnica de ajuste de bloco, as coordenadas georreferenciadas de todos os pontos. Usando essas, podem-se determinar quais os coeficientes das transformações necessárias a serem aplicadas em cada imagem para geração do mosaico final.

4. Implementação

O sistema foi desenvolvido em C++, usando a biblioteca QT para criação de interfaces gráficas com o usuário. A estrutura geral do sistema é mostrado na Figura 2. Note que o módulo de calibração de câmera está localizado externo à região com linhas tracejadas. Isto ocorre devido ao processo de calibração de câmera ser realizado apenas uma vez, em que determina-se os parâmetros intrínsecos para correção da distorção radial e radiométrica.

Na fase de calibração, computamos todos parâmetros necessários usando método de Tsai, em que relaciona-se coordenadas do mundo com coordenadas da imagem. Assim, a correção radial e radiométrica foram realizadas aplicando a Equação 1 e a Equação 2, seguido por interpolação de pixels, resultando em uma imagem corrigida. Após a correção radial e radiométrica, a orientação relativa é determinada. Para calcular esta, definimos pontos homólogos entre pares estéreos e computamos as transformações de mapeamento de uma imagem em relação a outra através do método de mínimos quadrados. As transformações, contém translação, rotação e escala, em que estas operações são usadas para mapear os pixels correspondentes em diferentes imagens, então, estes são ajustados no mosaico final.

Para gerar mosaicos georreferenciados, precisamos executar a orientação absoluta, assim, obteremos coeficientes para realizar o processo de aero-triangulação e ajustar o mosaico com as coordenadas do mundo (terreno).

No módulo de correção geométrica, transformações afim (Equação 4) ou projetiva (Equação 5) são aplicadas (usando interpolação bilinear ou bicúbica). A transformação afim é formalizada a seguir:

$$\begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} A_0 + A_1X' + A_2Y' \\ B_0 + B_1X' + B_2Y' \end{bmatrix} \quad (4)$$

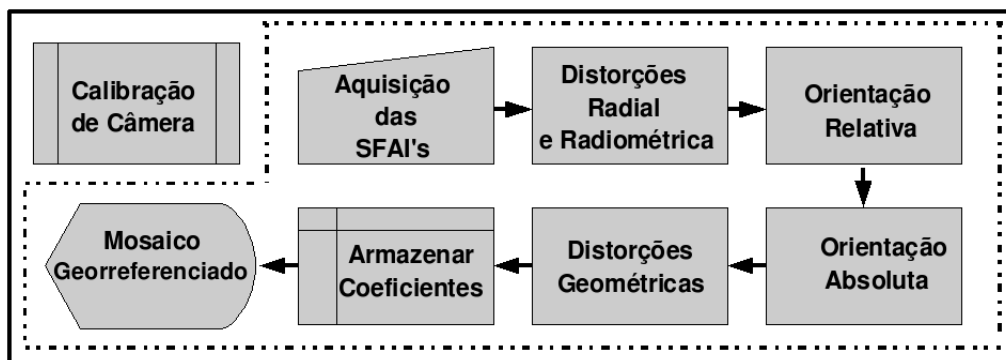


Figura 2. Estrutura do sistema.

onde $A_0, A_1, A_2, B_0, B_1, B_2$ são parâmetros correspondentes a dois fatores de escala, um de rotação e um de não ortogonalidade(cisalhamento), com i e $j = 1$ ou 2 ; X e Y são coordenadas a serem transformadas no espaço; X' e Y' são coordenadas no espaço transformado; X_0 e Y_0 são parâmetros de translação na direção X e Y [11].

A transformação projetiva é dada por:

$$\begin{aligned} X' &= \left[\frac{a_{11}X + a_{12}Y + a_{13}}{a_{31}X + a_{32}Y + 1} \right] \\ Y' &= \left[\frac{a_{21}X + a_{22}Y + a_{23}}{a_{31}X + a_{32}Y + 1} \right] \end{aligned} \quad (5)$$

onde a_{ij} são os parâmetros das transformações geométricas, com i e $j = 1, 2$ ou 3 ; X e Y são os valores medidos no sistema de referência; X' e Y' são os valores calculados para o sistema de ajuste [3].

A implementação dessas duas transformações é de suma importância para que possamos comparar os mosaicos gerados após as suas aplicações nas imagens. O objetivo é retirar distorções geométricas referentes às imagens que farão parte do mosaico, principalmente da área de sobreposição. No caso, distorções geradas pela variação da orientação do helicóptero aeromodelo na captação das imagens em pontos diferentes.

Essa diferença de resultados entre as transformações afim e projetiva pode ser justificadas, porque o paralelismo na transformação afim é mantido, enquanto que, na transformação projetiva pode não se manter. As duas transformações são semelhantes nos aspectos de posição, escala e forma. Esses itens são alterados na transformação da imagem. Contudo a colinearidade não é alterada.

5. Experimentos e Resultados

Com intuito de verificar nossa metodologia, primeiramente realizamos testes para validar os módulos do sistema

desenvolvido, passo a passo. Na Figura 3(a) mostra-se uma imagem aérea de pequeno formato obtida com a câmera digital comum acoplada ao helicóptero aeromodelo. Na Figura 3(b) é ilustrado a correção radial e radiométrica, obtida após aplicar a distorção "pincushion". É interessante ressaltar que esta correção foi realizada justamente com os coeficientes gerados no processo de calibração de câmera (Tsai). Nestes resultados pode observar-se uma sutil diferença geométrica e de iluminação, principalmente nas áreas colaterais da imagem.

Posteriormente, vários experimentos foram executados no processo de orientação relativa, usando ambas as transformações, e observamos visualmente que a transformação projetiva produziu mosaicos com melhor qualidade. Essa diferença de resultados entre as transformações afim e projetiva pode ser justificadas, porque o paralelismo na transformação afim é mantido, enquanto que, na transformação projetiva pode não se manter. As duas transformações são semelhantes nos aspectos de posição, escala e forma. Esses itens são alterados na transformação da imagem. Contudo a colinearidade não é alterada e essas informações são visíveis na figura 4.

Na Figura 4 é ilustrado o módulo de correção geométrica e orientação relativa, além dos resultados das transformações oriundas desse módulo. Pode ser observado que neste caso, as transformações foram geradas somente com a transformação projetiva, justamente pela geração de resultados melhores. Na Figura 5 é mostrado dois exemplos de mosaicos produzidos usando os mesmos módulos e as mesmas transformações. Finalmente, na Figura 6 ilustramos o resultado de um mosaico georreferenciado. Para este processo foi necessário conhecer as coordenadas geográficas dos pontos de controle (terreno) usando GPS. Assim, através de uma interface gráfica, o usuário insere estas informações, clicando nos pontos de controle planejados no mosaico (imagem) e associando com as coordenadas geográficas colhidas no terreno.

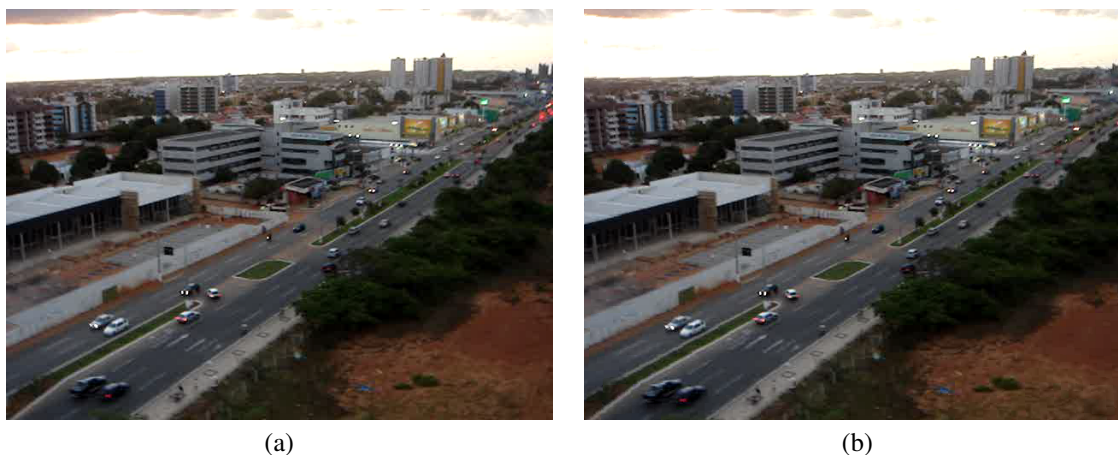


Figura 3. Imagem Original (a), e imagem corrigida da distorção "Pincushion" $\kappa_1 = -2$, $\kappa_2 = -0.025$ e $\alpha = 0.45$ (b)

6. Conclusão

Propomos, neste trabalho, um sistema completo para gerar mosaicos georreferenciados, usando imagens aéreas de pequeno formato, obtidas por uma filmadora digital comum acoplada a um helicóptero aeromodelo. Por meio de técnicas da estéreo-fotogrametria e da visão computacional, retiramos distorções, corrigimos as imagens visando a sua ortogonalização e construímos o mosaico final.

Então, com base em coordenadas de terreno (reais) de pontos de controle, previamente escolhidos, o georreferenciamento foi realizado. O mosaico final está, então, pronto para ser usado em um SIG, correntemente em desenvolvimento na UFRN.

Uma das principais contribuições do presente trabalho reside no fato de que esse tipo de imagem tem sido ainda muito pouco explorado na geração de mosaicos georreferenciados; especificando ainda mais, com o uso de um helicóptero aeromodelo não foi encontrado nenhum trabalho, talvez em função de uso de imagens de grande formato, e controladas (BFAI) ser geralmente a técnica adotada nos projetos de cartografia, solicitados pelos órgãos governamentais e empresas.

Note que o uso de BFAI encarece o mapeamento, principalmente em projetos como o nosso, onde um monitoramento constante da região de interesse é necessário. Ainda projetos dessa natureza em regiões costeiras, que implicam sobrevôos rotineiros, visando checar determinadas características, tornam-se factíveis com a nossa técnica.

Assim, como principal contribuição, com a implementação dessas técnicas, mostramos ser possível desenvolver uma metodologia de baixíssimo custo, com o uso de um helicóptero aeromodelo e câmera digital comum, voltada a projetos de monitoramento ou a projetos onde a coleta de dados deve ser constante. No caso, uma câmera Sony sem

controle fotogramétrico foi usada.

Ainda como trabalho futuro, faremos mais alguns vôos na região dos Parrachos de Maracajaú, com o helicóptero já operacional (vários vôos experimentais já foram realizados em terra). Depois de adquiridos novos dados, estes servirão de base para a construção do mosaico e, conseqüentemente, a alimentação do SIG de monitoramento ambiental em desenvolvimento na UFRN.

Referências

- [1] P. Albrecht and B. Michaelis. Stereo photogrammetry with improved spatial resolution. In *Proceedings of the 14th International Conference on Pattern Recognition*, Otto-von-Guericke University Magdeburg Institute for Measurement Technology and Electronics P. O. Box 4120, D-39016 Magdeburg, Germany, 1998. IEEE Computer Society Washington, DC, USA.
- [2] D. Ballard and C. Brown. *Computer Vision*. Prattice-Hall, Englewood Cliffs, New Jersey, 1982.
- [3] L. Fonseca, E. Xavier, J. Alge, and E. Castejon. Implementação da transformação projetiva na terralib: uma análise comparativa com a transformação afim. In *XIII Simpósio Brasileiro de Sensoriamento Remoto*, pages 6283–6290, Florianópolis/SC, 21-26 Abril 2007. INPE.
- [4] M. Galo. Automação dos processos de correspondência e orientação relativa em visão estéreo. Master's thesis, UNICAMP - Departamento de Engenharia Elétrica e de Computação, Campinas, SP, Dezembro 2003.
- [5] L. M. G. Gonçalves. *Reconstrução a partir de estéreo fotogrametria*. Rio de Janeiro - BRASIL, 1995.
- [6] G. Grandi, P. Mayaux, Y. Rauste, A. Rosenqvist, M. Simard, and S. Saatchi. The global rain forest mapping project jers-1 radar mosaic of tropical africa: Development and product characterization aspects. *IEEE Transactions On Geoscience and Remote Sensing*, 38(5), Setembro 2000.



(a)



(b)



(c)



(d)



(e)



(f)

Figura 4. Módulo de orientação relativa e correção geométrica (a), SFAI Original (Primeira) (b), SFAI Original (Segunda) (c), SFAI Original (Terceira) (d), SFAI Corrigida (Segunda) com transformação projetiva (e), SFAI Corrigida (Terceira) com transformação projetiva (f).



(a)



(b)

Figura 5. Mosaico com três SFAs da Figura 4 (a), Outro exemplo de mosaico com cinco SFAs (b).



(a)



(b)

Figura 6. Exemplo de mosaico com cinco SFAls (a), Mosaico da Figura (a) georreferenciado com pontos dado pelo GPS (Coordenadas Geográficas) (b).

- [7] S. Hsu. Geocoded terrestrial mosaics using pose sensors and videos registrations. In *ICCV. Proceedings. Eighth IEEE International Conference on Computer Vision*, pages 834–841. IEEE Computer Society, Julho 2001.
- [8] M. Lhuillier, L. Quan, H. Shum, and H. Tsui. Relief mosaics by joint view triangulation. In *ICCV. Proceedings. Eighth IEEE International Conference on Computer Vision*, pages 785–790. IEEE Computer Society Washington, DC, USA, Julho 2001.
- [9] D. Marr and T. Poggio. A computational theory of human stereo vision. In *PROC*, volume 204, pages 301–328. Royal Society Publishing, 1979.
- [10] H. Nicolas. New methods for dynamic mosaicking. *IEEE Transactions On Image Processing*, 10(9), Agosto 2001.
- [11] F. Nogueira. Geração automática de mapas de disparidade em visão estéreo. Master's thesis, UNICAMP - Faculdade de Engenharia Elétrica e de Computação - FEEC, Campinas, SP, Abril 1998.
- [12] S. Syuhei, B. Yue, and M. Takahito. Autonomous flight control for rc helicopter using camera image. In *SICE*, Kagawa University, September 2007. IEEE Computer Society.
- [13] E. Trucco and A. Verri. *Introductory Techniques for 3-D Computer Vision*. Prentice-Hall, Upper Saddle River, New Jersey, 1998.
- [14] R. Tsai. An efficient and accurate camera calibration technique for 3d machine vision. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 364–374, 1986.
- [15] P. Wolf. *Elements of Photogrammetry*. McGraw-Hill Book Company, Singapore, 1983.
- [16] N. Xiao, Z. Wu, L. Xjao, Y. Tang, and W. Junma. Auto method for ambient light independet panorama mosaics. In *Proceedings of International Conference on Machine Learning and Cybernetics*, pages 3851–3854. IEEE Computer Society, Agosto 2004.
- [17] Z. Zhu, A. Hanson, and E. Riseman. Parallel-perspective stereo mosaics. In *ICCV. Proceedings. Eighth IEEE International Conference on Computer Vision*, pages 345–352, Department of Computer Science, University of Massachusetts at Amherst, MA 01003, Julho 2001. IEEE Computer Society.
- [18] D. Zomet. Generating mosaics with minimum distortions. In *Proceedings of the Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'04)*, School of Computer Science and Engineering - The Hebrew University of Jerusalem, Junho 2004. IEEE Computer Society.

Reconhecimento Semi-Automático de Sinus Frontais para Identificação Humana Forense Baseado na Transformada Imagem-Floresta e no Contexto da Forma *

¹Juan Rogelio Falguera e ²Aparecido Nilceu Marana

UNESP – Faculdade de Ciências – Departamento de Computação – Bauru – SP – Brasil

¹juan.falguera@gmail.com

²nilceu@fc.unesp.br

Resumo

Diversos métodos biométricos baseados em características físicas como impressão digital, face e íris têm sido propostos para identificação humana. No entanto, para a identificação post-mortem, tais características podem não estar disponíveis. Nestes casos, partes do esqueleto do corpo humano podem ser utilizadas. Investigações anteriores mostraram que os padrões dos sinus frontais são únicos para cada indivíduo. Esta dissertação de mestrado propõe um método computacional para o reconhecimento de sinus frontais. Para tanto, foram avaliados métodos de segmentação de imagens de radiografias anteroposteriores de sinus frontais. O método baseado na Transformada Imagem-Floresta demonstrou ser eficiente para segmentação dos sinus frontais das imagens de radiografias. Foram também investigadas e implementadas técnicas para extração de descritores geométricos e baseados nas formas dos sinus frontais. Os resultados obtidos nos experimentos corroboram os encontrados na literatura sobre a individualidade dos sinus frontais e sua viabilidade em termos de precisão e usabilidade para a identificação humana post-mortem.

Abstract

Several methods based on Biometrics such as fingerprint, face and iris have been proposed for person identification. However, for postmortem identification such biometric measurements may not be available. In such cases, parts of the human skeleton can be used. Previous investigations showed that frontal sinus patterns are unique for each individual. The objective of this master thesis is to propose a computational method for frontal sinus recognition for

postmortem human identification. In order to achieve this, methods for frontal sinus segmentation from anteroposterior radiographs were evaluated. The method based on Image-Foresting Transform has shown itself efficient in frontal sinus segmentation from radiograph images. Techniques for extracting frontal sinus geometrical and shape-based descriptors were investigated and implemented as well. The results obtained in our experiments confirm the outcomes described in literature about the individuality of the frontal sinus and its feasibility in terms of precision and usability for postmortem human identification.

1. Introdução

A identificação biométrica de indivíduos refere-se ao uso de atributos físicos ou comportamentais (impressões digitais, face, íris, sinus frontais, assinatura, voz, etc.) para estabelecer suas identidades. Para que um atributo humano seja utilizado como um identificador biométrico, ele deve satisfazer alguns requisitos como: universalidade, unicidade, permanência, aceitabilidade, entre outros [1].

No caso de identificação de cadáveres há a possibilidade desses atributos físicos não estarem disponíveis. Nestes casos, partes do esqueleto do corpo humano tais como: dentes, tórax, vértebras, ombros e os sinus frontais podem ser utilizadas.

Os sinus frontais, também conhecidos como seios frontais da face, são cavidades loculadas e irregulares localizadas no osso frontal do crânio e que se comunicam com a fossa nasal através do infundibulum [2].

Diversas investigações realizadas anteriormente mostram que os padrões dos sinus frontais são altamente variáveis e únicos para cada indivíduo

*Artigo completo de dissertação de mestrado

[2,3,4,5] sendo, portanto, uma característica física que pode ser utilizada em sistemas biométricos forenses.

1.1. Objetivo

O objetivo do presente trabalho de mestrado é propor, implementar e avaliar um método para automatização do reconhecimento de sinus frontais para identificação humana, de modo que sistemas biométricos forenses baseados neste método sejam viáveis em termos de precisão e usabilidade.

1.2. Justificativas

Mais do que uma necessidade, a identificação post-mortem de pessoas é uma obrigação legal para a resolução de diversas questões como herança e sucessão de propriedade, cobrança de apólices de seguro e pensões, reparações e indenizações em casos de acidentes, entre outras. Além disso, a identificação rápida e precisa de um indivíduo é também uma questão humanitária, uma vez que pode abreviar o sofrimento de sua família em caso de desaparecimento ou de acidentes envolvendo dezenas ou centenas de pessoas.

Os sistemas biométricos computacionais podem ser importantes aliados no processo de identificação post-mortem de indivíduos, acelerando e facilitando o trabalho dos peritos forenses. No entanto, a Biometria aplicada à medicina legal pode apresentar sérias limitações, principalmente no que diz respeito à indisponibilidade de certas características físicas, devido ao estado de conservação precário do corpo do indivíduo que se deseja identificar.

Neste contexto, o sinus frontal é um excelente candidato à característica física a ser utilizada em sistemas biométricos para a identificação de cadáveres [2,3,4,5], não só por sua unicidade e permanência, mas também por estar localizado em uma parte da estrutura óssea muito resistente (osso frontal do crânio).

Em comparação com outras características físicas geralmente utilizadas na medicina legal, os sinus frontais podem apresentar algumas vantagens. Com relação à identificação de indivíduos baseada no formato e disposição dos dentes, muito utilizada pela medicina forense, os sinus frontais podem apresentar certa vantagem quanto à característica de permanência, uma vez que seu formato se estabiliza em torno dos 20 anos de vida do indivíduo e permanece inalterado, a não ser em casos de acidentes que ocasionem traumas no osso frontal do crânio ou em caso de algumas doenças que possam alterar seu formato. No caso do formato e disposição dos dentes, estes estão mais

sujeitos a alterações durante a vida do indivíduo, devido a doenças, extrações dentárias, restaurações odontológicas, tratamentos ortodônticos e traumas, que podem dificultar a identificação do indivíduo ao se comparar radiografias post-mortem com radiografias antemortem.

Os sinus frontais podem ser também uma ótima alternativa aos exames de DNA, uma vez que este, apesar de ser extremamente preciso, pode ser demorado, altamente custoso e estar sujeito à contaminação.

No entanto, apesar da literatura registrar diversos casos de identificação positiva de pessoas por meio dos sinus frontais [4,5,8] são raros os métodos para o reconhecimento destes que utilizam recursos computacionais para a automatização ou facilitação dos processos, o que se configura em importante justificativa para os esforços empreendidos neste trabalho.

2. Sinus Frontais

Os sinus frontais (ou seios frontais) da face estão localizados dentro do osso frontal do crânio e consistem em um par de cavidades loculadas e irregulares que se comunicam com a fossa nasal através do infundibulum [2].

A Figura 1 exibe uma radiografia anteroposterior (AP) (a) e outra em perfil (b) de um indivíduo, com as bordas dos sinus frontais no destaque.

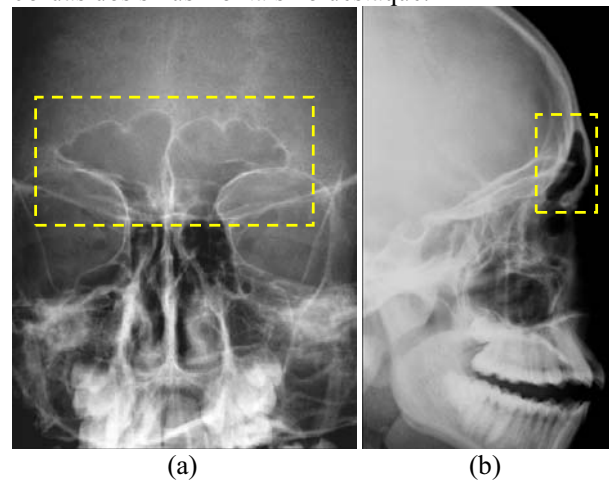


Figura 1. (a) Radiografia anteroposterior e (b) perfil com as bordas dos sinus frontais no destaque.

Os sinus frontais se desenvolvem embrionariamente de uma célula etmoidal e não são visíveis ao nascimento. Eles começam a se desenvolver a partir do segundo ano de vida e atingem seu tamanho máximo

por volta do vigésimo ano, permanecendo estáveis a partir de então [9,10].

A configuração dos sinus frontais é considerada única para cada indivíduo [2,3,4,5] e é controlada por fatores ambientais, embora a raça, o sexo e algumas doenças também afetem o seu desenvolvimento [4]. Variações dos sinus frontais foram registradas na literatura até mesmo para gêmeos monozigóticos [2,5]. A Figura 2 apresenta imagens de radiografias em incidência AP de quatro indivíduos, onde é possível observar as peculiaridades dos sinus frontais para cada um deles.

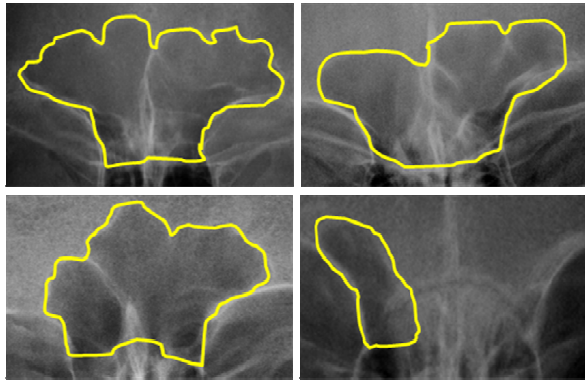


Figura 2. Imagens de radiografias de sinus frontais de quatro indivíduos, onde é possível observar as diferenças entre seus formatos.

2.1. Reconhecimento de Sinus Frontais

O reconhecimento de sinus frontais é composto por três etapas principais: (i) segmentação; (ii) extração das características e (iii) casamento.

A segmentação é o processo pelo qual, dada uma radiografia AP do crânio de um indivíduo, apenas os pontos pertencentes à borda que separa a área externa e interna dos sinus frontais são extraídos.

Na etapa de extração das características, métodos para análise de formas são utilizados para se obter informações a partir das bordas dos sinus frontais, de maneira que se possa compor um descritor para cada borda. O descritor é, em geral, um vetor de características que armazena algum tipo de medida relacionada à forma do objeto, de modo compacto, que pode ser armazenado em um banco de dados e, posteriormente, utilizado na realização do casamento.

Na etapa de casamento, os descritores extraídos dos sinus frontais são comparados de algum modo a se obter uma medida de similaridade entre eles. Esta medida pode ser obtida, por exemplo, através de uma equação matemática que exprima o grau de alinhamento entre os vetores de características. Dessa

forma, se a medida de similaridade entre duas imagens de sinus frontais estiver acima de um determinado limiar, entende-se que estas pertencem ao mesmo indivíduo (genuínas), caso contrário, consideram-se imagens pertencentes a indivíduos diferentes (impostores).

2.2. Desafios no Reconhecimento de Sinus Frontais

Os maiores desafios no reconhecimento de sinus frontais estão concentrados na etapa de segmentação, na qual os pontos da borda dos sinus frontais devem ser extraídos a partir de uma radiografia AP do crânio. Esta etapa é considerada complexa uma vez que a detecção das bordas dos sinus frontais pode ser dificultada pela presença de ruídos na imagem, como os descritos a seguir.

2.2.1. Ruídos Causados pela Projeção de Outras Estruturas Craniais

No momento em que uma radiografia AP do crânio é feita, as estruturas de todo o crânio, que estão dispostas em três dimensões, são projetadas em um filme de duas dimensões.

Uma estrutura da parte traseira do crânio, denominada lâmina interna, pode ser projetada durante a geração da radiografia e aparecer próxima aos sinus frontais ou até mesmo atravessando-os de forma longitudinal.

O mesmo acontece com três outras estruturas do crânio: a crista gali e a crista frontal interna, que atravessam os sinus frontais de forma latitudinal, e as paredes superiores das órbitas oculares, que podem atravessar os sinus frontais pelos dois lados. No caso das paredes, estas podem ajudar a delimitar os sinus frontais, pois suas bordas geralmente coincidem com as bordas laterais inferiores dos sinus. No entanto, como é possível observar na Figura 3, a borda direita inferior dos sinus frontais invade o interior da órbita ocular de modo que a parede superior da órbita direita cria a impressão de uma falsa borda dos sinus.

A Figura 3 apresenta um sinus frontal com suas bordas e com as estruturas citadas demarcadas.

2.2.2. Ruídos Durante a Geração da Radiografia

A geração da radiografia no aparelho de Raios-X pode sofrer interferências que prejudicam a visualização dos sinus frontais, por diversas razões, como por exemplo, a falta de imobilização do paciente ou ainda por questões estruturais do próprio aparelho.

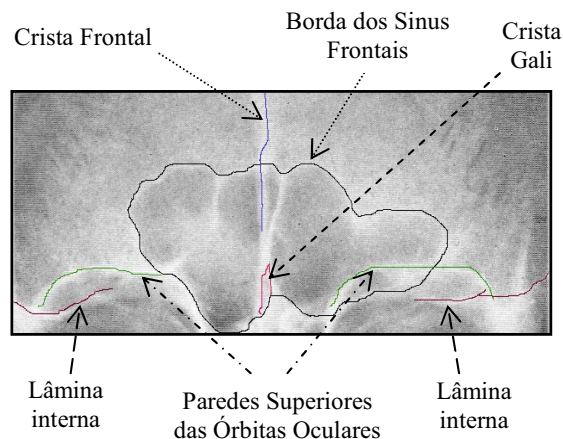


Figura 3. Sinus frontal com algumas estruturas do crânio em destaque e com suas bordas demarcadas.

No caso específico do banco de imagens de sinus frontais utilizado neste trabalho, todas as radiografias apresentam uma faixa vertical no centro da parte superior do crânio, sobrepondo-se aos sinus frontais. Esta faixa é causada por um suporte do aparelho de raios-X, que tem por objetivo centralizar e imobilizar o crânio do paciente. A Figura 4 exibe a imagem digitalizada a partir de uma das radiografias do banco de imagens citado, na qual é possível observar a faixa vertical sobrepondo-se ao sinus frontal, dificultando a definição de parte da borda superior.

2.2.3. Ruídos Causados por Secreção na Área Interna dos Sinus Frontais

Por se tratarem de câmaras de ar, os sinus frontais devem apresentar na radiografia, em sua região interna, níveis de cinza menores (mais escuros) do que a região externa adjacente. Este contraste entre as regiões interna e externa aos sinus frontais contribui para a determinação precisa dos pontos de suas bordas.

No entanto, devido à comunicação existente com a fossa nasal [2], os sinus frontais muitas vezes armazenam secreções que causam na radiografia um clareamento de sua região interna. Caso este clareamento aconteça próximo à região de fronteira entre as regiões interna e externa, a detecção dos pontos da borda pode ser dificultada ou comprometida.

A Figura 5 exibe uma imagem na qual não é possível observar parte da bordas do sinus frontal devido a um clareamento na sua área interna, causado por excesso de secreção. Observe no destaque uma pequena porção da borda na qual a visualização de seus pontos fica prejudicada pelo acúmulo de secreção na região de fronteira.

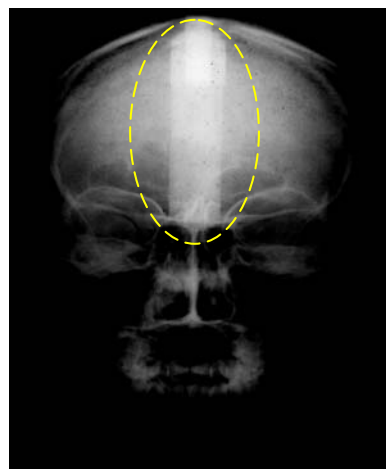


Figura 4. Radiografia AP na qual é possível observar uma interferência causada pelo suporte do aparelho de raios-X.

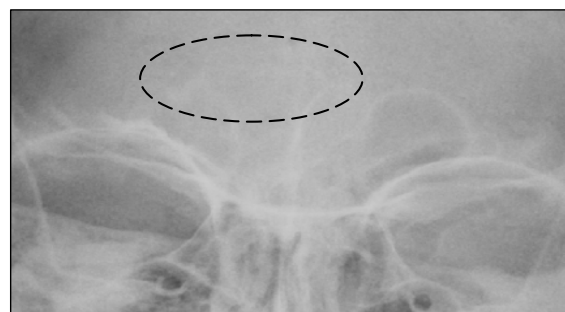


Figura 5. Imagem na qual não é possível observar parte da borda do sinus frontal devido a um clareamento na sua área interna, causado por excesso de secreção.

3. Material

Nesta seção são apresentados o banco de imagens e o *ground truth* utilizados nos experimentos para a comprovação da efetividade do método proposto no presente trabalho.

3.1. Banco de Imagens

As técnicas para segmentação e reconhecimento de sinus frontais foram avaliadas utilizando-se um banco contendo 90 imagens de radiografias AP de 29 indivíduos com mais de 20 anos de idade. Nesse conjunto de radiografias há pelo menos duas imagens de radiografias para cada indivíduo.

As imagens encontram-se na mesma escala e orientação, uma vez que o aparelho de raios-X utilizado na obtenção das radiografias contava com um

suporte para centralização e imobilização do crânio do paciente

As radiografias foram obtidas dos arquivos de radiografias do Denver Growth Study, mantidos pela Faculdade de Odontologia da Universidade de Michigan, localizada em Ann Arbor, Michigan, EUA.

3.2. Ground Truth

Com o objetivo de prover uma avaliação consistente do método de segmentação proposto, realizou-se, sob a supervisão de um profissional médico da área de Radiologia do Hospital Estadual de Bauru, a segmentação manual dos sinus frontais do banco de imagens utilizado. Ao final do processo obteve-se um banco de bordas “verdadeiras” dos sinus frontais, o *ground truth*, com o qual é possível realizar a avaliação supervisionada do método de segmentação.

Para a segmentação manual, foi utilizada a ferramenta “paint brush” do software de edição de imagens Serif Photoplus [11]. Sobre uma nova camada na imagem original, a borda foi manualmente traçada com a utilização do mouse e os pontos da borda puderam ser extraídos diretamente da camada superior.

4. Métodos

Nesta seção, é apresentado o método proposto para o reconhecimento de sinus frontais, composto pelos métodos Transformada Imagem-Floresta Diferencial (DIFT) e *Shape Context*, para as etapas de segmentação e extração de características.

4.1. Método da Transformada Imagem-Floresta Diferencial

O método baseado na Transformada Imagem-Floresta Diferencial (DIFT - Differential Image-Foresting Transform) para segmentação de imagens foi proposto por Falcão e Bergo [6]. Esse método reduz o problema de segmentação de imagens ao cálculo de florestas de caminhos de custo mínimo no grafo derivado da imagem. A busca por caminhos de custo mínimo é restringida a caminhos que se originam em um conjunto de pixels iniciais, denominados sementes [6,12].

A função que determina o custo do caminho de um vértice do grafo até outro é a diferença entre os níveis de cinza dos pixels que são representados pelos vértices. Dessa forma, pixels com níveis próximos de cinza terão custos menores de caminho e tenderão a se conectar ao caminho com o qual tenham maior grau de similaridade.

Portanto, com a utilização desse algoritmo, a intervenção do usuário durante a segmentação da imagem é reduzida à escolha de algumas sementes dentro e fora da área de interesse. As sementes competem por vértices no grafo a cada iteração e geram, ao final, regiões internas e externas à área de interesse cujos pixels terminais delimitam o objeto [12].

A Figura 6(a) apresenta um grafo de uma imagem bidimensional em tons de cinza, com vizinhança 4. Os números correspondem às intensidades $I(p)$ dos pixels e os círculos maiores denotam duas sementes: uma interna ao objeto que se deseja segmentar (um retângulo no centro da imagem) e outra externa ao objeto. A Figura 6(b) mostra uma floresta de caminhos de custo mínimo para o grafo da Figura 6(a), usando como função de custo, $C(p,q) = I(q) - I(p)$, onde $I(p)$ é a intensidade de um pixel e $I(q)$ é a intensidade do seu pixel predecessor. Os números mostrados na Figura 6(b) são os valores de custo.

A segmentação do retângulo no centro da imagem é obtida pela árvore enraizada na semente interna. As setas da Figura 6(b) apontam para o predecessor do vértice no caminho ótimo.

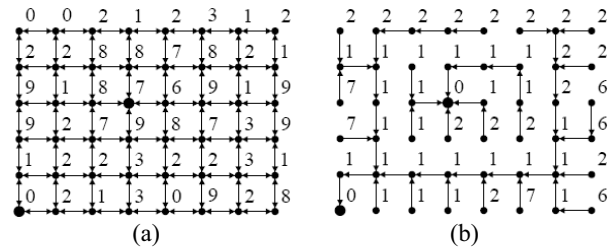


Figura 6. (a) Grafo de uma imagem 2D em tons de cinza com vizinhança 4. (b) Floresta de caminhos de custo mínimo para o grafo da Figura 6(a) [12].

Caso o resultado da segmentação não seja ideal, o usuário pode acrescentar mais sementes dentro ou fora da área de interesse e executar novamente o algoritmo, que ao invés de recalculer uma nova floresta de caminhos ótimos toda vez que o conjunto de sementes é alterado, somente atualiza o resultado da segmentação de forma diferencial, em tempo proporcional ao número de vértices nas regiões modificadas da floresta [12].

4.2. Método Shape Context

O Contexto da Forma (*Shape Context*) é um descritor de formas proposto por Belongie et al. [7]. Dado um ponto p pertencente a uma borda, esse descritor captura a distribuição polar-logarítmica dos

demais pontos da borda em relação a p , proporcionando uma caracterização global e discriminativa do contexto da forma da borda. Desse modo, pontos correspondentes em duas bordas semelhantes, apresentam contextos de forma semelhantes [7].

A Figura 7(a) apresenta a imagem de uma borda de sinus frontal, na qual se ilustra o cálculo do histograma polar-logarítmico de um dos pontos da borda. Para tanto, conta-se o número de pontos da borda em cada compartimento. A Figura 7(b) ilustra o histograma completo do ponto da borda selecionado na Figura 7(a).

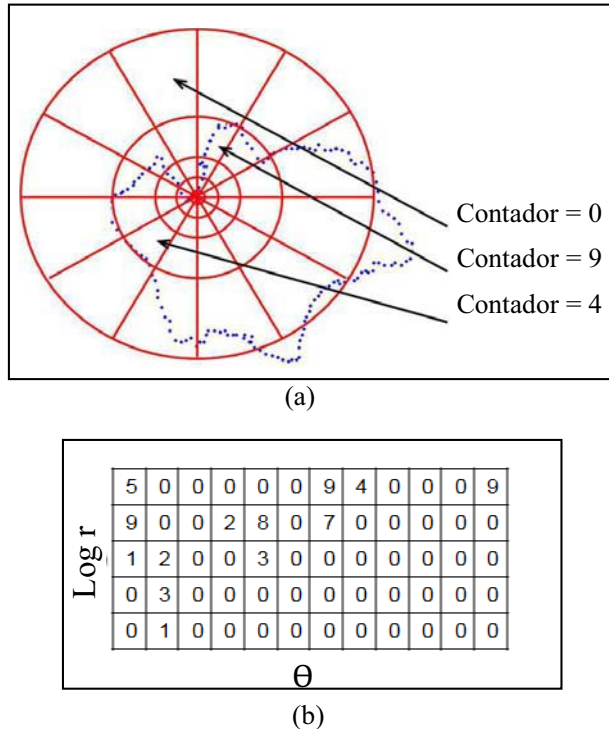


Figura 7. (a) Ilustração do cálculo do histograma polar-logarítmico de um dos pontos da borda de um sinus frontal; (b) Histograma computado para o ponto da borda selecionado em (a).

A Figura 8 ilustra a comparação entre histogramas de três pontos de bordas de duas imagens obtidas em momentos distintos do sinus frontal de um indivíduo. Note que os histogramas dos dois pontos das bordas localizados em regiões semelhantes apresentam histogramas semelhantes. Note também que o terceiro ponto da borda localizado à direita, em uma região bastante diferente das anteriores, apresenta um histograma completamente diferente dos outros dois pontos. Para facilitar a comparação entre os histogramas, seus valores foram substituídos por níveis

de cinza: quanto mais escura a cor da posição no histograma, maior é a ocorrência de pontos da borda naquele ângulo e distância.

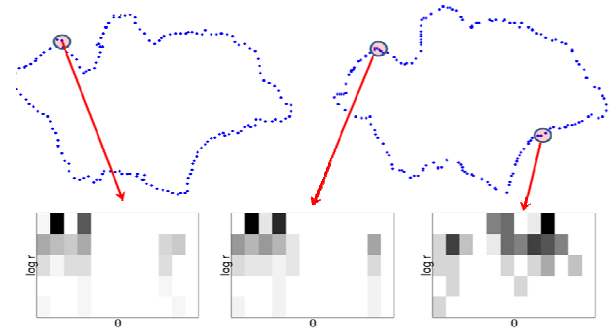


Figura 8. Comparação entre histogramas de três pontos distintos de duas bordas obtidas dos sinus frontais do mesmo indivíduo.

Para o método *Shape Context*, o casamento entre dois sinus frontais é feito através do cálculo da similaridade entre eles. A similaridade é determinada pelo custo do melhor alinhamento obtido entre os dois sinus, utilizando os descritores dos contextos de forma para cada um dos pontos pertencentes às suas bordas. Quanto menor o custo de alinhamento, maior o grau de similaridade entre os sinus frontais.

Dados dois pontos p_i e p_j , pertencentes a cada uma das bordas sendo comparadas, o custo de casamento C_{ij} entre os pontos p_i e p_j , é dado pela equação (1), onde $h_i(k)$ e $h_j(k)$ denotam os histogramas normalizados dos pontos p_i e p_j , respectivamente, e K representa a quantidade de compartimentos do histograma [7]. Como os contextos das formas associados a cada ponto são distribuições representadas por histogramas, é natural utilizar-se o teste estatístico Chi-quadrado.

$$C_{ij} = \frac{1}{2} \sum_{k=1}^K \frac{[h_i(k) - h_j(k)]^2}{h_i(k) + h_j(k)} \quad (1)$$

4.3. Fases de um Sistema Biométrico Baseado no Método Proposto

Os sistemas biométricos são compostos de duas fases principais: registro e reconhecimento. Desse modo, as duas fases do sistema biométrico baseado no método proposto, são apresentadas a seguir.

4.3.1. Fase de Registro

O objetivo principal da fase de registro é obter descritores de sinus frontais a partir de imagens digitais

de radiografias AP antemortem de indivíduos que se deseja identificar, de forma que possam ser armazenados, de forma compacta, em um banco de dados e, posteriormente, na fase de reconhecimento, possam ser submetidos a um processo de análise de similaridade contra sinus frontais obtidos de radiografias AP post-mortem.

A Figura 9 exibe uma visão geral da fase de registro de um sistema biométrico baseado no método proposto.

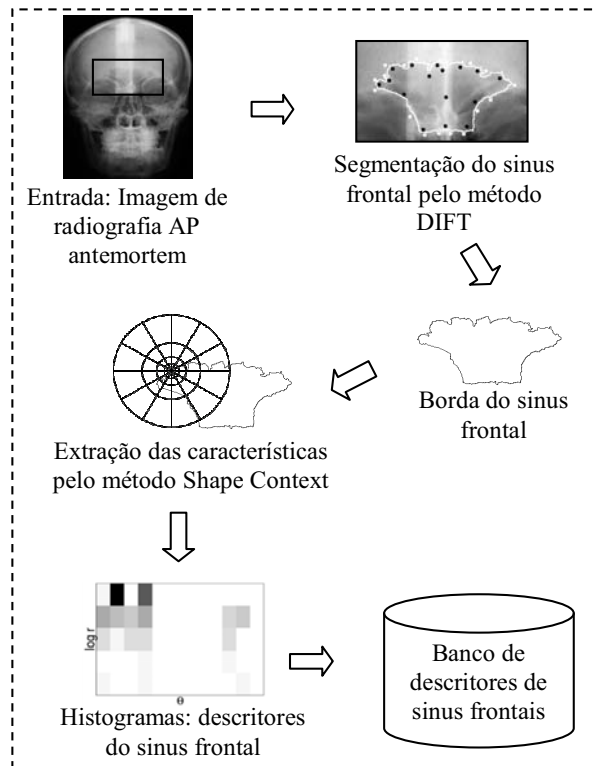


Figura 9. Fase de registro de um sistema biométrico baseado no método proposto.

As etapas que compõem a fase de registro são:

- 1. Entrada de dados:** São apresentadas ao sistema, imagens digitais de radiografias AP antemortem de indivíduos que se deseja cadastrar.
- 2. Segmentação:** Os sinus frontais presentes nas imagens das radiografias são segmentados utilizando-se o método da Transformada Imagem-Floresta Diferencial (DIFT). Ao final do processo de segmentação, imagens binárias contendo apenas as bordas dos sinus frontais são obtidas e submetidas à extração de características.
- 3. Extração de características:** As características das bordas dos sinus frontais segmentados são

extraídas por meio do método *Shape Context* para análise de formas. O processo de extração resulta em um histograma polar-logarítmico para cada um dos pontos das bordas dos sinus frontais.

4. Armazenagem na Base de Dados: Os histogramas, que são os descritores dos sinus frontais para o método *Shape Context*, são armazenados em um banco de dados (de templates) para serem utilizados na fase de reconhecimento.

4.3.2. Fase de Reconhecimento

Na fase de reconhecimento, a imagem digital de uma radiografia AP post-mortem de um indivíduo (imagem de consulta) é apresentada ao sistema biométrico a fim de se determinar a identidade deste, através do cálculo de similaridade dos descritores de seu sinus frontal em relação aos descritores dos sinus frontais armazenados em um banco de dados na fase de registro.

A Figura 10 exibe uma visão geral da fase de reconhecimento de um sistema biométrico baseado no método proposto.

As etapas que compõem a fase de reconhecimento são:

- 1. Entrada de dados:** A entrada do sistema de reconhecimento de sinus frontais é uma imagem de uma radiografia AP, denominada imagem de consulta. A imagem de consulta é apresentada ao sistema e os descritores do sinus frontal são obtidos de modo idêntico ao descrito na fase de registro.
 - 2. Casamento:** Os graus de similaridade entre os descritores do sinus frontal de consulta e todos os descritores dos sinus frontais armazenados no banco de dados (os *templates*) são obtidos através da equação (1).
 - 3. Decisão:** Se o sistema biométrico estiver operando no modo de autenticação, ou seja, se existir uma hipótese prévia para a identificação do cadáver, o sistema aceitará ou rejeitará a identidade presumida, dependendo do valor do limiar estabelecido para o grau de similaridade entre o sinus frontal de consulta e os templates.
- De outro modo, se o sistema biométrico estiver operando no modo de identificação, será retornada uma lista ordenada contendo as identidades associadas aos m templates do banco de dados que apresentam os maiores graus de similaridade com o sinus frontal de consulta.

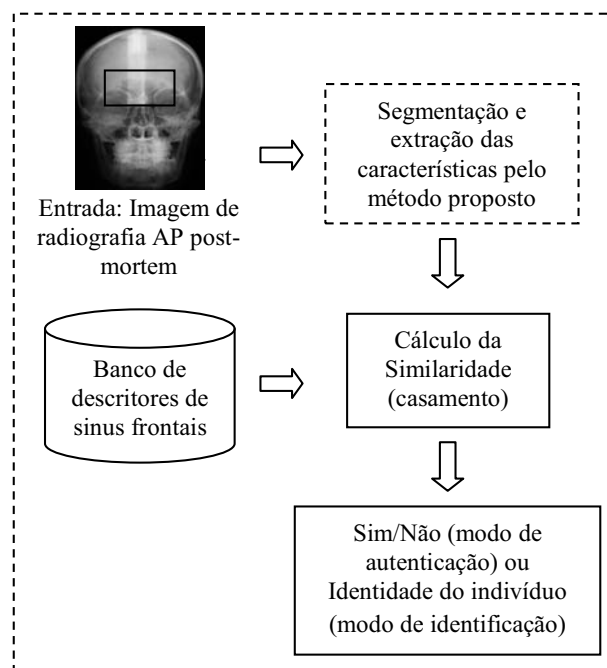


Figura 10. Fase de reconhecimento para o método proposto.

4.4. Avaliação do Método Proposto

Para realizar uma avaliação de desempenho consistente do método proposto para reconhecimento de sinus frontais, além dos experimentos realizados com os métodos de segmentação de imagens e análise de formas que o compõem (DIFT e *Shape Context*), foram realizados experimentos com outros métodos de segmentação e análise de formas.

Além do método manual de segmentação de imagens, o método *Watershed* [13] foi utilizado para fins de comparação de desempenho com o método DIFT, por se tratar de um método amplamente utilizado para a segmentação de diversos tipos de imagens, inclusive de imagens médicas. No entanto, devido aos ruídos apresentados nas imagens de sinus frontais, descritos anteriormente, o método *Watershed* não obteve resultados satisfatórios sendo, portanto, descartado.

Portanto, o desempenho do método semi-automático de segmentação dos sinus frontais baseado na DIFT pôde ser avaliado quantitativamente por meio de comparações com a segmentação manual (o *ground truth*).

Os métodos de análise de formas escolhidos para fins de comparação com o método *Shape Context* foram: (i) método de análise por descritores geométricos, proposto por Marana et al. [14] e (ii) método de análise por estatísticas dos ângulos dos raios

(BAS), proposto por Arica e Vural [15]. Os dois métodos, assim como o método baseado no contexto da forma, foram implementados em Matlab [16].

Os experimentos realizados com estes métodos seguiram as mesmas etapas de registro e de reconhecimento descritas nas seções 4.3.1 e 4.3.2, respectivamente.

Para cada método de análise de formas, foram utilizadas como entrada as bordas dos sinus frontais do banco de imagens, segmentadas pelo método manual, com o auxílio de um software gráfico de propósito geral e pelo método DIFT, por meio do software Mflow [17]. O Mflow apresenta uma implementação deste método que permite a marcação por parte do usuário de sementes internas e externas à região de interesse. Após a definição das sementes, executa-se o método DIFT e caso a segmentação não esteja satisfatória, é possível determinar novas sementes externas ou internas e executar o software novamente (de modo incremental), até que uma segmentação considerada adequada seja obtida.

Após a obtenção dos descritores por meio dos três métodos de análise de formas utilizados para as bordas dos sinus frontais segmentados manualmente e pelo método DIFT, foram realizados casamentos entre estes descritores a fim de se obter os graus de similaridade entre todos eles. Dessa forma, nove experimentos foram realizados.

Após a realização de cada experimento foram calculadas as taxas de erro igual (EER) e de recuperação correta (CRR). Desse modo, foi possível analisar, a partir de diversos cenários, o desempenho do método proposto para reconhecimento dos sinus frontais utilizando-se a transformada imagem-floresta e os descritores do contexto da forma (*Shape Context*), comparativamente à segmentação manual (*ground truth*) e com os descritores de formas alternativos.

5. Resultados Experimentais

Nesta seção são apresentados os resultados dos experimentos realizados na etapa de segmentação e os resultados gerais para os métodos de reconhecimento de sinus frontais, compostos pela combinação dos métodos de segmentação com os métodos de análise de formas para a extração das características.

5.1. Segmentação de Sinus Frontais

A avaliação das segmentações dos sinus frontais pelo método DIFT foi realizada de duas maneiras: (i) análise qualitativa, por meio de sobreposição com as segmentações manuais obtidas com a supervisão de um

profissional médico da área de Radiologia; e (ii) análise quantitativa, por meio do cálculo da similaridade entre as segmentações obtidas pelo método DIFT e pelo método manual, por meio dos métodos de análise de formas implementados (*Shape Context*, BAS e Medidas Geométricas) e a subsequente geração das taxas EER e CRR.

Na análise qualitativa, o profissional médico da área de Radiologia que supervisionou a segmentação manual atestou que as segmentações obtidas utilizando-se o método DIFT estavam muito próximas do ideal, pois não foi possível notar diferenças significativas em relação às segmentações manuais, como mostra a Figura 11.

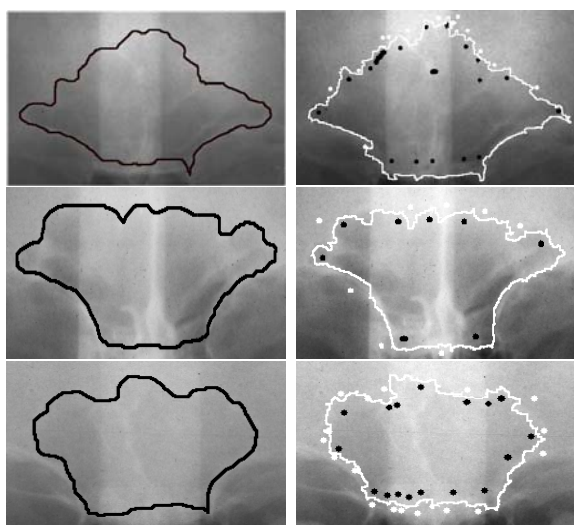


Figura 11. Segmentações de 3 sinus frontais, utilizando o método manual (à esquerda) e por DIFT (à direita). Os pontos pretos e brancos representam, respectivamente, as sementes internas e externas definidas pelo usuário.

Os resultados da análise quantitativa para o casamento entre as bordas segmentadas pelo método DIFT versus o método manual (*ground truth*) por meio dos três métodos de análise de formas, estão mostrados na Tabela 1. As taxas de erro igual (EER) e de recuperação correta (CRR mostram o alto grau de similaridade entre as bordas segmentadas pelo método DIFT e as bordas segmentadas manualmente (*ground truth*).

O profissional médico da área de Radiologia, que supervisionou a segmentação manual para obtenção do *ground truth* realizou também algumas segmentações de sinus frontais manualmente, com o auxílio de um software gráfico de propósito geral (Serif Photo Plus) e utilizando o software Mflow. Desse modo foi possível obter o relato da experiência do usuário.

O profissional atestou que se sentiu muito mais confortável em realizar as segmentações pelo método DIFT do que pelo método manual uma vez que no método DIFT há apenas a necessidade de se selecionar alguns pontos dentro e fora da área de interesse.

5.2. Reconhecimento de Sinus Frontais

Para a avaliação do desempenho dos métodos para reconhecimento de sinus frontais implementados, foram realizados nove experimentos. Para cada método de análise de formas, os seguintes casamentos entre os bancos de bordas foram feitos: (i) casamento e cálculo da similaridade entre as bordas segmentadas manualmente; (ii) casamento e cálculo da similaridade entre as bordas segmentadas pelo método DIFT e (iii) casamento e cálculo da similaridade entre as bordas segmentadas manualmente versus as bordas segmentadas pelo método DIFT. Os casamentos entre as bordas dos sinus frontais dos indivíduos foram feitos no modo de identificação, ou seja, cada borda foi comparada com todas as bordas restantes do banco, calculando-se a similaridade para cada casamento. Ao final do processo foram obtidas as taxas de erro para os casamentos entre genuínos e impostores e calcular a taxa de erro igual (EER- Equal Error Rate) e a taxa de recuperações corretas (CRR – Correct Retrievals Rate).

Os resultados obtidos para cada método de reconhecimento de sinus frontais estão descritos na Tabela 1, onde estão destacados os resultados para o método de proposto no presente trabalho.

Tabela 1. Resultados dos experimentos realizados para avaliar os métodos de reconhecimento de sinus frontais.

Descritores da Forma	Método de Segmentação	EER	CRR Top 1
Geométrico	Manual	12,22%	36,7%
	DIFT	12,63%	26,7%
	Manual vs. DIFT	12,39%	52,22%
Shape Context	Manual	3,85%	100%
	DIFT	3,73%	95,5%
	Manual vs. DIFT	7,37%	95,5%
BAS	Manual	8,28%	91,1%
	DIFT	8,81%	83,33%
	Manual vs. DIFT	10,52%	75,6%

6. Conclusões

O método proposto para reconhecimento de sinus frontais, que combina os métodos DIFT e *Shape Context* para segmentação e extração das características obteve a melhor taxa de erro igual (EER = 3,73%) entre todos os outros métodos para reconhecimento de sinus frontais. Porém, apesar do método proposto ter obtido uma taxa de recuperações corretas muito boa (CRR = 95,5%), ele ficou atrás do método que combina a segmentação manual com o método *Shape Context* (CRR = 100%).

No entanto, a segmentação manual com auxílio de software exige esforço e destreza muito maiores por parte do usuário do que o método DIFT.

Sendo assim, uma vez que a segmentação manual foi realizada neste trabalho por um usuário com boa destreza manual, é lícito acreditar que usuários com pouca experiência de uso de computadores ou com pouca destreza manual venham a obter melhores resultados através da segmentação pelo método DIFT do que pela segmentação manual.

Desse modo, os resultados experimentais obtidos permitem concluir que o método proposto para o reconhecimento de sinus frontais, que combina a segmentação pelo método DIFT e a extração de características pelo método *Shape Context* é adequado para a identificação forense de indivíduos, baseada nas características de sinus frontais obtidos a partir de radiografias AP antemortem e post-mortem. Além disso, estes resultados corroboram os descritos na literatura sobre a viabilidade do uso das informações dos sinus frontais como uma alternativa para a identificação humana post-mortem em aplicações forenses.

Agradecimentos

Os autores agradecem a Dra. S. L. Brooks e o Dr. J. McNamara, da Universidade de Michigan, EUA, pelas imagens de radiografias de sinus frontais, e a Dra. Flavia Casadei, do Hospital Estadual de Bauru, por sua colaboração durante a segmentação manual dos sinus frontais.

Referências

- [1] JAIN, A. K.; ROSS, A.; PRABHAKAR, S. **An Introduction to Biometric Recognition**. IEEE Transactions on Circuits and Systems for Video Technology, Special Issue on Image- and Video-Based Biometrics. Vol. 14. No. 1, pp. 4-20, 2004.
- [2] SAUNDERS, W.B. **Dorland's Illustrated Medical Dictionary**. 25th ed.. Philadelphia: 1974.

- [3] BROGDON, B. G. **Forensic Radiology**. CRC Press, 1998.
- [4] KIRK, N. J.; WOOD, R. E.; GOLDSTEIN, M. **Skeletal Identification Using the Frontal Sinus Region: A Retrospective Study of 39 Cases**. J Forensic Sci 2002, 47(2):318-323.
- [5] SCHULLER, A. **A note on the identification of skulls by x-ray pictures of the frontal sinuses**. Med. J. Australia, 1943, 1:554-7.
- [6] FALCÃO, A. X.; BERGO, F. P. G. **Interactive volume segmentation with differential image foresting transforms**. IEEE Trans. on Medical Imaging, 23(9):1100-1108, 2004.
- [7] BELONGIE, Serge; MALIK, Jitendra; PUZICHA, Jan. **Shape context: A new descriptor for shape matching and object recognition**. In NIPS, p. 831-837. November 2000.
- [8] RIBEIRO, F. A. Q. **Standardized measurements of radiographic films of the frontal sinuses: An aid to identifying unknown persons**. Ear, Nose and Throat Journal, Jan 2000, v79, p26.
- [9] HAJEK, M. **Normal anatomy of the frontal sinus**. In: Pathology and Treatment of the Inflammatory Diseases of the Nasal Accessory Sinuses. St. Louis: C.V. Mosby, 1926:35-43.
- [10] SIEUR, JACOB, O. **Des sinus frontaux**. In: Recherches Anatomiques, Cliniques et Operatoires sur les Fosses Nasales et Leur Sinus. Paris: J. Rueff, 1901:83-91.
- [11] SERIF Photo Plus. **Software**. Disponível em: <http://www.freerifsoftware.com/>.
- [12] MIRANDA, P. A. V. **Segmentação de Imagens pela Transformada Imagem-Floresta**. Dissertação de Mestrado. IC-UNICAMP, 2006.
- [13] VINCENT, L.; SOILLE, P. **Watersheds in digital spaces: An efficient algorithm based on immersion simulations**. IEEE Transactions on Pattern Analysis and Machine Intelligence, 13(6):583-598. 1991.
- [14] MARANA, A. N.; OURA, D. Y.; YAMAMOTO, T.; FALGUERA, F. P. S.; FALGUERA, J. R. **Towards an Automatic Frontal Sinus Identification System**. Anais do II Workshop de Visão Computacional, p. 93-98. USP - São Carlos, 2006.
- [15] ARICA, Nafiz; VURAL, Fatos T. Y. **BAS: a perceptual shape descriptor based on the beam angle statistics**. Pattern Recognition Letters. Vol. 24. Number 9. p. 1627-1639(13). Elsevier. 2003.
- [16] MATLAB 7.0. **Software**. Disponível em: http://www.mathworks.com/products/featured/embeddedmatlab/index.html?s_cid=HP_MI_EmbeddedMATLAB.
- [17] MFLOW. **Software**. Disponível em: <http://www.ic.unicamp.br/~afalcao/download/gift.tar.gz>.

Shape Descriptors based on Tensor Scale

Fernanda A. Andaló, Ricardo da S. Torres, and Alexandre X. Falcão
Institute of Computing – State University of Campinas (Unicamp)
CEP 13084-851 – Campinas – SP – Brazil
{feandalo,rtorres,afalcao}@ic.unicamp.br

Abstract

Tensor scale is a morphometric parameter that unifies the representation of local structure thickness, orientation, and anisotropy, which can be used in several computer vision and image processing tasks. We exploit this concept for binary images and propose two shape descriptors – Tensor Scale Descriptor with Influence Zones and Tensor Scale Contour Saliences. It also introduces a robust method to compute tensor scale, using a graph-based approach – the image foresting transform. Experimental results are provided, showing the effectiveness of the proposed methods, when compared to other relevant methods with regard to their use in content-based image retrieval tasks.

1. Introduction

The recent growth of the World Wide Web and the new technologies that became available for image acquisition and storage have increased the demand for image retrieval systems based on image properties. In content-based image retrieval (CBIR) systems, image processing techniques are used to describe the image content, encoding image properties – shape, color, or texture – that are relevant to the query. These properties are processed by image descriptors that can be characterized by two functions: a feature vector extraction function and a similarity function that computes the similarity between images based on their feature vectors [7].

The shape of an object is an important and basic visual feature for describing image content [16]. Shapes are often the archetypes of objects belonging to the same pattern class, and can be used in a wide range of practical problems, such as document analysis (optical character recognition), internet (content-based image retrieval), security (fingerprint detection), etc [6].

This work focus on shape feature extraction and description for CBIR systems and, for this purpose, we need a parameter for characterizing the structures presented

in the images. In [15], Saha introduces a new concept called *tensor scale* – a local morphometric parameter that yields a unified representation of structure size, orientation, and anisotropy. We extend the application of tensor scale, proposing two shape descriptors – Tensor Scale Descriptor with Influence Zones (TSDIZ) and Tensor Scale Contour Saliences (TSCS). We also present a much faster tensor scale computation, as compared to previous methods [15, 14], by exploiting the Euclidean Image Foresting Transform (Euclidean IFT) [10]. This new tensor scale algorithm can also be applied to the solution of other problems not related to image description, such as clustering, classification, image filtering and image registration.

The complete work¹ is available at [1] and its related publications are [2, 3].

This paper briefly overviews the methods as follows. Section 2 describes the previous methods for computing tensor scale and also the new method proposed for binary images. Section 3 describes the proposed shape descriptors and Section 4 shows the related experiments and results. Finally, Section 5 concludes this paper.

2. Tensor scale

In [15], Saha introduced the *tensor scale* of a pixel p in a gray-scale image as the largest ellipse within the same homogeneous region, centered at p . The homogeneous region is defined based on small differences between the pixels' intensity.

Tensor scale defines the ellipse by three factors:

- *Orientation*(p) = angle between $t_1(p)$ and the horizontal axis;
- *Anisotropy*(p) = $\sqrt{1 - \frac{|t_2(p)|^2}{|t_1(p)|^2}}$;
- *Thickness*(p) = $|t_2(p)|$;

¹ This paper contains extracts of the M.Sc. dissertation named "Descritores de forma baseados em tensor scale" by Fernanda A. Andaló.

where $|t_1(p)|$ and $|t_2(p)|$, with $|t_1(p)| \geq |t_2(p)|$, denote the length of the two semi-axis of the ellipse centered at p . Figure 1 illustrates the components to compute each one of these factors.

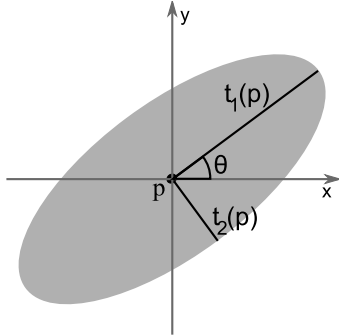


Figure 1. Tensor scale factors.

In the following subsection, we describe two previously proposed tensor scale computation methods for gray-scale images [15, 14]. After that, we provide a faster tensor scale computation for binary images.

2.1. Tensor scale for gray-scale images

In Saha's approach [15], a tensor scale ellipse is calculated from sample lines that are traced around a given pixel, from 0 to 179 degrees (Figure 2(a)). The axes of the ellipse are determined by computing the image intensities along each of the sample lines and the localization of two opposite edge points on these lines (Figure 2(b)). The next step consists of repositioning the edge locations to points equidistant to that given pixel, following the axial symmetry of the ellipse (Figure 2(c)). The computation of the best-fit ellipse to the repositioned edge locations is done by Principal Component Analysis (PCA) (Figure 2(d)).

These computations are performed for every pixel of the image and a critical drawback is that the computational cost of the algorithm makes the method quite prohibitive for more complex tasks, such as image description for content-based image retrieval. For this reason, Miranda et al. [14] proposed an efficient implementation of the original method, which differs in the following aspects.

The first change was in the localization of the edge points. Miranda's approach proposes to go along each pair of opposite segments, alternately and at the same time, instead of going along one entire segment by turn. By doing this, the reposition phase is no longer necessary. The second change was the use of two connected thresholds to improve and simplify the original method of detecting edges. The third and final change was the improvement of the ellipse computation phase. Miranda et al. proposed a func-

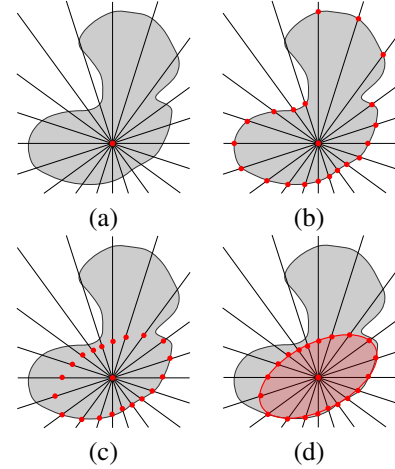


Figure 2. Tensor scale Computation.

tion g (Equation 1) that gives the angle of the ellipse directly, instead of using PCA. The ellipse orientation is obtained from the value of γ that minimizes the function

$$g(\gamma) = \sum_{i=1,2,\dots,2m} [x_{i_\gamma}^2 - y_{i_\gamma}^2], \quad (1)$$

where $x_{i_\gamma} = x_i \cos(\gamma) - y_i \sin(\gamma)$, $y_{i_\gamma} = x_i \sin(\gamma) + y_i \cos(\gamma)$, (x_i, y_i) are the relative coordinates of the edge points with respect to the center pixel $p = (x_p, y_p)$ of the ellipse, and $(x_{i_\gamma}, y_{i_\gamma})$ are the new coordinates after applying a rotation by the angle γ .

In the next subsection, we provide a faster tensor scale computation, as compared to the described approaches [15, 14], by exploiting the Euclidean IFT.

2.2. Tensor scale for binary images

The improvements achieved by Miranda et al. algorithm are important for computing tensor scale in gray-scale images. However, for binary images the tensor scale computation can be further improved.

The first simple change consists of the elimination of the thresholds used for edge detecting. These thresholds are not necessary, because the images have already been segmented. Furthermore, the method can incorporate techniques to easily find the edges in the directions of the sample lines. Such techniques comprehend the use of Euclidean Distance Transform, computed by the Image Foresting Transform, that is described in the next subsection.

2.2.1. Euclidean Distance Transform via Image Foresting Transform

The Image Foresting Transform (IFT) is a graph-based approach to the design of image processing operators based on connectivity, in which the images are represented by graphs – the pixels are considered as nodes and

the arcs are defined by an adjacency relation between pixels. For a given seed set (roots), the seeds compete with each other, defining influence zones. Each influence zone consists of pixels that are “more closely connected” to a seed than to any other, according to a path-cost function [10]. We use a path-cost function that assigns the closest Euclidean distance between object pixels and contour pixels to each pixel inside the object (Euclidean IFT).

In the Euclidean IFT (Algorithm 1), the path-cost function is such that the cost of a path from a seed s to a pixel t in the forest is the Euclidean distance between s and t . The algorithm also needs an Euclidean relation A defined as

$$q \in A(p) \Rightarrow (x_q - x_p)^2 + (y_q - y_p)^2 \leq \rho^2$$

where ρ is the adjacency radius and (x_i, y_i) are the coordinates of a pixel i in the image.

Algorithm 1 assigns to each object pixel p three attributes: the squared Euclidean distance $C(p)$ between p and its closest point s in the contour (forming an optimum cost map), its closest seed $R(p) = s$ (forming a root map), and the label $L(p) = L(s)$ of the segment that contains s (forming a label map).

The advantages of computing the Euclidean Distance Transform via IFT is that label propagation is executed on-the-fly and in linear time. The Euclidean IFT is used for two purposes in the proposed methods: faster tensor scale computation, that is described in the next subsection, and tensor scale orientation mapping (Section 3.1).

2.2.2. Tensor scale computation via Image Foresting Transform A considerable speed up in the computation of the tensor scale for binary images is possible by exploiting the following aspect: if we have the shortest distance between a pixel p and the contour, there is no need to search for edge points inside the circle with radius $\sqrt{C(p)}$ (Figure 3(a)). For every pixel p , this distance can be obtained from the cost attribute $C(p)$ returned by Euclidean IFT.

According to Miranda’s algorithm, edge points are searched along opposite sample lines, alternately. However, in our approach, the algorithm jumps along the lines and visits the pixels q and r at the same time (Figure 3(b)). The searching for edge points continues outside the area defined by the cost $\sqrt{C(p)}$ in Figure 3(b), and the minimum between $\sqrt{C(r)}$ and $\sqrt{C(q)}$ indicates the location for the next jump. These jumps may continue iteratively until the closest edge point along the sample line is found.

In the example, the edge is found at the pixel $R(r)$ (i.e., at the contour point r' nearest to r). The algorithm defines that the two edge points in this sample line are at r' (coordinate of $R(r)$ relative to p) and at q' (coordinate of the point diametrically opposite to r' , relative to p), as shown in Figure 3(c).

Algorithm 1:

Input: A binary image I , a set S of contour points or segments in I (seeds), an Euclidean adjacency relation A , and a labeling function $\lambda(p)$ that assigns a label to each point or segment p in S .

Output: The cost map C , the root map R , and the label map L .

Auxiliary data structure: A priority queue Q .

```

foreach  $p \in I$  do
     $C(p) \leftarrow +\infty$ ;
     $R(p) \leftarrow NIL$ ;  $L(p) \leftarrow NIL$ ;
foreach  $p \in S$  do
     $C(p) \leftarrow 0$ ;
     $R(p) \leftarrow p$ ;
     $L(p) \leftarrow \lambda(p)$ ;
    insert  $p$  in  $Q$ ;
while  $Q$  is not empty do
    remove from  $Q$  a pixel  $p = (x_p, y_p)$  such that  $C(p)$ 
        is minimum;
    foreach  $q = (x_q, y_q)$  such that  $q \in A(p)$  and
         $C(q) > C(p)$  do
         $C' \leftarrow (x_q - x_{R(p)})^2 + (y_q - y_{R(p)})^2$ , where
             $R(p) = (x_{R(p)}, y_{R(p)})$  is the root pixel of  $p$ ;
        if  $C' < C(q)$  then
            if  $C(q) \neq +\infty$  then
                remove  $q$  from  $Q$ ;
             $C(q) \leftarrow C'$ ;
             $R(q) \leftarrow R(p)$ ;
             $L(q) \leftarrow L(p)$ ;
            insert  $q$  in  $Q$ ;

```

By performing this procedure for all sample lines, the algorithm defines all edge points and uses the same formula defined by Miranda et al. (Equation 1) for finding the orientation of the ellipse.

The localization of the edge points is formalized in Algorithm 2.

3. Shape descriptors

The key idea of the proposed methods is to compute the tensor scale ellipse for every object point and to map the orientations onto the object’s contour. The tensor scale computation is done by the proposed method described in the previous section. The orientation mapping is described in the next subsection.

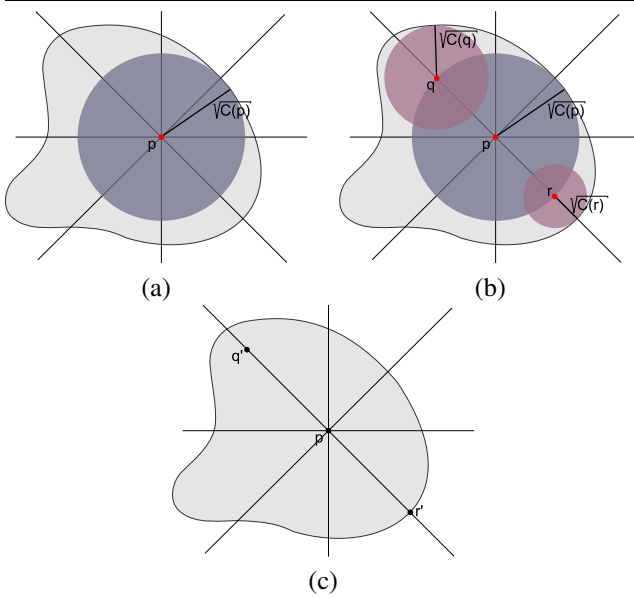


Figure 3. Example of optimization by using Euclidean IFT.

3.1. Orientation mapping

The orientation mapping (Algorithm 3) consists in mapping the tensor scale ellipse orientation (computed for all the object pixels) onto the object's contour. Although the tensor scale orientation mapping is different for each proposed method, they both exploit the discrete Voronoi regions (influence zones) of contour points inside the object. The influence zones can be efficiently obtained by label propagation using the Euclidean IFT [10].

It is necessary to compute the discrete Voronoi regions inside the object and summarize, in some way, the orientation information contained in each of these regions. The introduced methods require two different types of orientation mapping: one that assigns information to each contour point and one that assigns information to contour segments. The former can be understood as an instance of the latter with one-point length segments.

The orientation mapping (Algorithm 3) requires label map L returned by the Euclidean IFT (Algorithm 1), using the segments as seeds and a labeling function that assigns a different label to each segment. Map L groups pixels and their ellipses in the influence zone of each segment.

The meta-function $Summary(V[i])$ returns a summarization of ellipses information inside the correspondent influence zone. Each method has its own summary function and they will be discussed in the next subsections.

Algorithm 2:

Input: A pixel $p = (x_p, y_p)$, the number m of sample lines, and the cost map C returned by Algorithm 1.

Output: The vector *edge* that contains m pairs of edge points localized at the sample lines.

```

for  $\theta \leftarrow 0^\circ$  to  $179^\circ$ , with increments  $\frac{180}{m}$  do
     $v \leftarrow \sqrt{C(p)}$ ;
     $p_1 \leftarrow NIL$ ;
     $p_2 \leftarrow NIL$ ;
     $q_1 \leftarrow 0$ ;
     $q_2 \leftarrow 0$ ;
    while  $p_1 \neq 0$  and  $p_2 \neq 0$  do
         $x \leftarrow v * \cos(\theta)$ ;
         $y \leftarrow v * \sin(\theta)$ ;
        if  $q_1 = 0$ 
             $temp \leftarrow (x_p + x, y_p + y)$ ;
             $p_1 \leftarrow \sqrt{C(temp)}$ ;
        if  $q_2 = 0$ 
             $temp \leftarrow (x_p - x, y_p - y)$ ;
             $p_2 \leftarrow \sqrt{C(temp)}$ ;
         $d \leftarrow \min(p_1, p_2)$ ;
         $v \leftarrow v + d$ ;
         $q_1 \leftarrow p_1 - d$ ;
         $q_2 \leftarrow p_2 - d$ ;
     $edge[\theta] \leftarrow ((x, y), (x', y'))$ , where  $(x', y')$  is the co-
    ordinate of the point diametrically opposite to
     $(x, y)$ , relative to  $p$ ;
```

3.2. Tensor Scale Contour Saliences (TSCS)

The saliences of a shape are defined as the higher curvature points along the shape contour [9], or vertex points along the contour with first derivative discontinuity [6]. Their detection is the key to various applications in image processing (e.g., image registration, polygonal approximation, motion analysis, and shape description [8]).

The TSCS method consists of three steps: the tensor scale computation for all pixels inside a given object, followed by the mapping of such tensor scale orientations onto each contour point, as described in Algorithm 3, and finally the detection of saliences based on the mapped orientations. These saliences are used to form the feature vector of the TSCS descriptor.

In this case, the $Summary(V[i])$ function of Algorithm 3 returns the orientation of the ellipse with maximum anisotropy inside the influence zone with label i :

Algorithm 3:

Input: A binary image containing an object O , the number n_s of contour segments, the label map L returned by the Euclidean IFT, and the vectors Ani and Ori that contain the anisotropies and the orientations of the tensor scale ellipses computed for all pixels of O , respectively.

Output: A vector M that contains the mapped orientations for each contour segment.

Auxiliary data structure: A vector V of n_s lists to store ellipse information in each influence zone of segment.

foreach pixel $p \in O$ **do**

insert $(Ani[p], Ori[p])$ in list $V[L(p)]$, where $L(p)$ is the label of the influence zone in which p is contained;

foreach $i \in [1, \dots, n_s]$ **do**

$M[i] = Summary(V[i]);$

$$Summary(V[i]) = \underset{p}{\operatorname{argmax}} Ani[p], (Ori[p], Ani[p]) \in V[i],$$

where the vectors Ani and Ori contain the anisotropies and the orientations of the tensor scale ellipses computed for all the object pixels.

Therefore, the vector M contains n_s values, where n_s is the number of points in the object's contour. Contour points with no influence zone inside the object borrow the orientations of the neighbors.

In order to localize the salience points, the method calculates the differences between adjacent mapped orientations in M . The difference value at $p \in M$ is

$$Difference(p) = AngularDist(M(p-1), M(p+1)),$$

where the function $AngularDist(\alpha, \beta)$ gives the smallest angle between the orientations α and β .

Now, the method uses a threshold value to eliminate low values of difference along the contour. Figure 4(a) shows the adopted threshold and the difference values computed for every contour point of the shape illustrate in Figure 4(b). Figure 4(c) shows another example of the detected saliences (dots) using threshold 16, i.e., saliences related to angle differences lower than 16° were not represented.

After the salience detection phase, the descriptor is formed by the feature extraction and metric functions used in CS [8].

3.3. Tensor Scale Descriptor with Influence Zones (TSDIZ)

The key idea of the Tensor Scale Descriptor with Influence zones (TSDIZ) is to map the tensor scale orientations

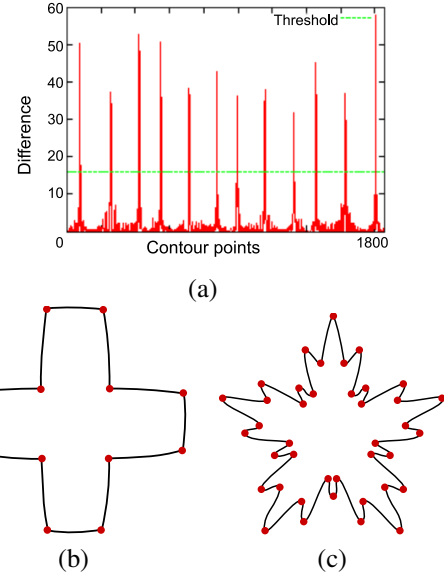


Figure 4. Difference values along a contour and detected saliences.

inside an object onto a few segments of its contour, and use this information for shape description.

First, the TSDIZ approach computes tensor scale for all pixels inside an object. Next, it divides the object's contour into segments and maps the tensor scale orientation onto each contour segment, as described in Algorithm 3.

For TSDIZ, the $Summary(V[i])$ function returns the weighted angular mean [13] of the ellipses orientations contained in the influence zone with label i , considering the anisotropies as the weights:

$$Summary(V[i]) = \arctan(K),$$

$$K = \frac{\sum_{(Ori[p], Ani[p]) \in V[i]} Ani[p] * \sin(2Ori[p])}{\sum_{(Ori[p], Ani[p]) \in V[i]} Ani[p] * \cos(2Ori[p])}.$$

The vector M returned by Algorithm 3 is used as TSDIZ feature vector and contains n_s values, where n_s is the number of contour segments.

The similarity function has to determine the rotation difference of the orientations between two TSDIZ vectors. This function also has to determine the segment in which the feature vectors must be lined up to obtain the best matching between the underlying shapes.

The exhaustive algorithm (Algorithm 4) consists of the registration between the orientation feature vectors. For this purpose, the algorithm computes, for each rotation α , where $\alpha = 0^\circ, \dots, 179^\circ$, and for each shift j in the feature vector, where $j = 1, \dots, n_s$ and n_s is the size of the vectors,

the difference between the vectors, after rotating all orientations of one vector by α and circular shifting the same vector by j . The minimum difference obtained corresponds to the distance between the vectors.

Algorithm 4:

Input: Two feature vectors F_A and F_B .

Output: Distance $dist$ between F_A and F_B .

```

 $dist \leftarrow \infty;$ 
foreach  $j \in [1, \dots, n_s]$  do
    foreach  $\alpha \in [0, \dots, 179]$  do
        foreach  $i \in [1, \dots, n_s]$  do
             $dist_{aux} \leftarrow \text{AngularDistance}(\{F_B[(j - i) \bmod n_s] + \alpha\} \bmod 180, F_A[i]);$ 
            if  $dist_{aux} < dist$  then
                 $dist \leftarrow dist_{aux};$ 

```

The complexity of this algorithm is $O(c * n_s^2)$, where c is a constant (in this case, 179°). Although it is an exhaustive search, small values of n_s (e.g., $n_s < 70$) makes it still fast. Figure 5 illustrates the registration between two TSDIZ vectors. An orientation curve is computed for each object and then, applying the matching algorithm, these curves can be matched.

4. Experimental results

Experiments were conducted using two databases: Fish-shape² and MPEG-7³ part B.

4.1. TSCS results

The experiments are based on comparisons between the TSCS and the Contour Saliency (CS) method [8], because of its interesting previous results. But, before comparing the different approaches, we need to find the best threshold for our method. For this purpose, we constructed a database consisting of 42 shapes of the Fish-shape database and 112 shapes of the MPEG-7 Part B database, resulting in 2835 saliencies. The images were chosen by taking into account the obviousness of the contour saliency points location. Then, a set of ground truth images were constructed with the location of the saliency points.

This experiment relies on counting the true positive saliencies (T_+) and false positive saliencies (F_+) for the

² <http://www.ee.surrey.ac.uk/research/vssp/imagedb/demo.html>

³ <http://www.chiariglione.org/mpeg/>

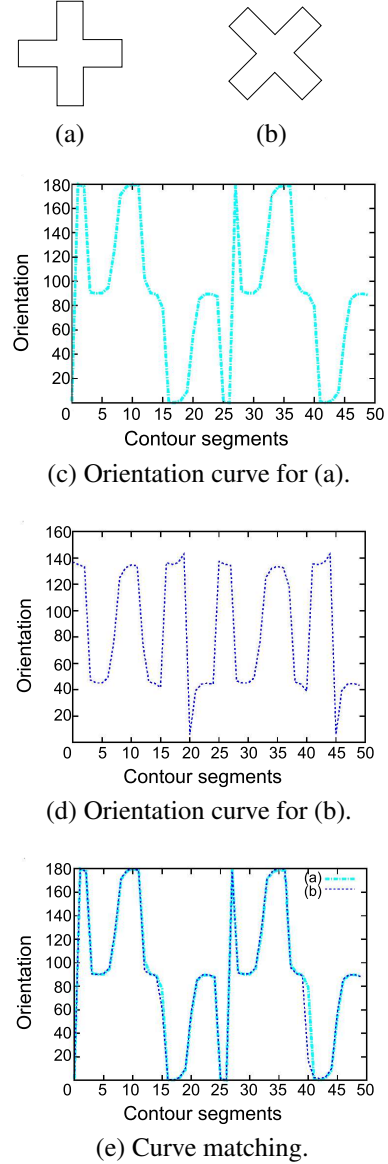


Figure 5. Examples of TSDIZ curves and registration.

ground truth images. After this counting, three effectiveness measures were calculated: recall, precision, and accuracy. Recall (Rec) and precision (Prec) are computed as

$$Rec = \frac{T_+}{T_+ + T_-}$$

and

$$Prec = \frac{T_+}{T_+ + F_+}$$

where T_- is the number of true negatives, and $(T_+ + T_-)$ represents the total number of points. The accuracy (Acc) is calculated as

Measures	10	12	14	16	18
Recall	0.964	0.964	0.963	0.963	0.962
Precision	0.889	0.923	0.946	0.963	0.968
Accuracy	0.840	0.862	0.874	0.875	0.867

Table 1. Effectiveness measures.

$$Acc = \frac{T_+ + T_-}{T_+ + T_- + F_+ + F_-}$$

where F_- (false negatives) represents the number of miss-detections.

The results with different threshold values (from 10 to 18) are presented in Table 1. Note that the method is robust to the choice of the threshold. However, the accuracy was maximized with threshold value 16 and this is the value adopted for this method in further experiments.

The first consideration made between the approaches was related to performance issues. Our method was twice faster (speedup of 2.04), on average, than the CS approach, when executed for the entire Fish database (on an AMD 64 3000+ with 1GB of memory).

The second consideration is that our method is computed locally, looking for each mapped orientation and for its neighbors along the contour. The CS method is more global, because it uses the internal and external skeletons of the whole shape for salience detection. This difference in granularity also makes the detection of saliences less robust in the CS approach, because the multiscale skeletons have to be thresholded to obtain salience points. This threshold represents a smoothing of the contour and, consequently, loss of some important saliences. In order to detect these saliences, we would have to reduce the threshold. Our method is also dependent of a threshold, but it is much easier to fix a single threshold for the entire database, which is the case of our approach, than to find the best threshold for every single image in the database, which is the case of the CS approach.

The last consideration is about the impact of a better salience estimation in shape description. Corners and high curvature points concentrate more information than other points of the shape [5]. For this reason, it is intuitive to conceive that curvature is an important key for the identification of many geometric aspects. Based on this, we use the saliences as key points for shape description.

We compared the descriptors using the multiscale separability (MS separability) effectiveness measure. Separability indicates the discriminatory ability between objects that belong to distinct classes [8]. The TSCS and CS descriptors were computed for Fish-shape database and the MS separability curves for the descriptors are shown in Figure 6. Higher is the curve, better is the method.

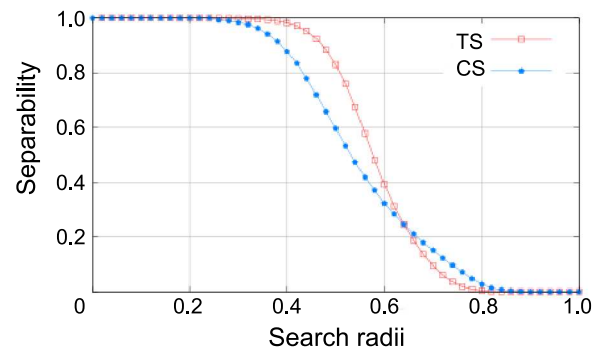


Figure 6. Multiscale separability curve for Fish database.

By analyzing Figure 6, we observe that TSCS is more effective or equal to CS in 80% of the search radii.

4.2. TSDIZ experiments

In [8], Torres et al. showed that MS separability represents better than precision vs. recall (PR) curves the separation among clusters (groups of relevant images) in the feature space. However, PR is still the most popular effectiveness measure in CBIR. For this reason, we present the results with both measures.

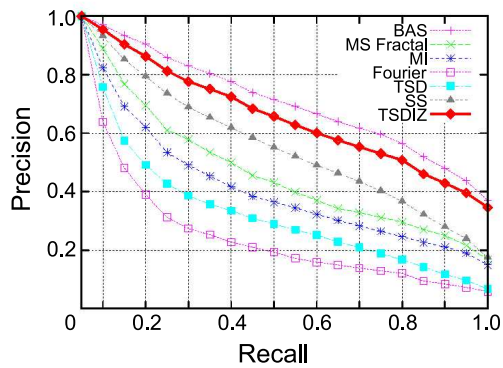
Precision is defined as the fraction of retrieved images that are relevant to the query. In contrast, recall measures the proportion of relevant images among the retrieved images. The Precision vs. Recall curve, or PR curve, indicates the commitment between the two measures and, generally, the highest curve in the graph indicates better effectiveness.

In this experiment, TSDIZ is compared with the following shape descriptors: Beam Angle Statistics [4] (BAS), Multiscale Fractal Dimension [9] (MS Fractal), Moment Invariants [12] (MI), Fourier Descriptor [11] (Fourier), Tensor Scale Descriptor [14] (TSD), and Segment Saliences [8] (SS).

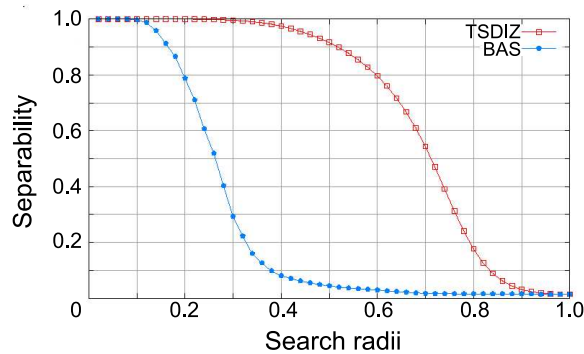
Figure 7(a) presents the PR curves for the evaluated descriptors and TSDIZ with 60 contour segments. We have tested different number of segments (30 to 120) and the method is quite robust to this choice.

TSDIZ descriptor has the second better PR curve among the tested descriptors. As TSDIZ has outperformed all other descriptors for MS separability as well, we show in Figure 7(b) the MS separability curves of TSDIZ and BAS only.

TSDIZ and BAS present equivalent effectiveness for search radii less than 10% of their maximum distance. From this point on, the BAS separability curve decreases quickly,



(a) PR curve.



(b) MS separability.

Figure 7. TSDIZ experiments.

indicating that this descriptor is neither robust nor effective for search radii greater than 20%.

Table 2 shows a visual CBIR example for a query image. The images that are not in the same class of the query image and should not be returned by the query are shown with a border around them.

5. Conclusions

This paper describes a faster algorithm for tensor scale computation in binary images using Image Foresting Transform (IFT), and two shape descriptors based on tensor scale.

For the TSCS descriptor, the experimental results showed that it is faster and more robust than the CS descriptor. The experiments for TSDIZ indicate that this descriptor has better PR curve than all relevant shape descriptors (except BAS) and the best separability among them, making it the most robust and effective according to this metric.

Acknowledgments

Authors are grateful to FAEPEX, FAPESP, CAPES, CNPq, and Microsoft.

Query image		
		
Rank	TSDIZ	BAS
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		

Table 2. Visual CBIR example.

References

- [1] F. A. Andaló. Descritores de forma baseados em tensor scale. IC – Unicamp, 2007.
<http://www.liv.ic.unicamp.br/~feandalo/pubs/msc.pdf>.
- [2] F. A. Andaló, P. A. V. Miranda, R. da S. Torres, and A. X. Falcão. Detecting contour saliences using tensor scale. In *Proceedings of the 14th IEEE International Conference on Image Processing*, volume 6, pages 349–352, September 2007.
- [3] F. A. Andaló, P. A. V. Miranda, R. da S. Torres, and A. X. Falcão. A new shape descriptor based on tensor scale. In *Proceedings of the 8th International Symposium on Mathematical Morphology*, pages 141–152, October 2007.
- [4] N. Arica and F. T. Y. Vural. BAS: a perceptual shape descriptor based on the beam angle statistics. *Pattern Recognition Letters*, 24(9–10):1627–1639, 2003.
- [5] F. Attneave. Some informational aspects of visual perception. *Psychological Review*, 61(3):183–193, 1954.
- [6] L. da F. Costa and J. R. M. Cesar. *Shape analysis and classification: theory and practice*. CRC Press, Florida, USA, 2001.
- [7] R. da S. Torres and A. X. Falcão. Content-based image retrieval: theory and applications. *Revista de Informática Teórica e Aplicada*, 13(2):161–185, 2006.
- [8] R. da S. Torres and A. X. Falcão. Contour salience descriptors for effective image retrieval and analysis. *Image and Vision Computing*, 25(1):3–13, 2007.
- [9] R. da S. Torres, A. X. Falcão, and L. da F. Costa. A graph-based approach for multiscale shape analysis. *Pattern Recognition*, 37(6):1163–1174, 2004.
- [10] A. X. Falcão, J. Stolfi, and R. A. Lotufo. The image foresting transform: theory, algorithms, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(1):19–29, 2004.
- [11] R. C. Gonzalez and R. E. Woods. *Digital image processing*. Addison-Wesley, 2nd edition, 2001.
- [12] M. K. Hu. Visual pattern recognition by moment invariants. *IEEE Transactions on Information Theory*, 8(2):1163–1174, 1962.
- [13] K. V. Mardia and P. E. Jupp. *Directional statistics*. John Wiley and Sons, 1999.
- [14] P. A. V. Miranda, R. da S. Torres, and A. X. Falcão. TSD: A shape descriptor based on a distribution of tensor scale local orientation. In *Proceedings of the Brazilian Symposium on Computer Graphics and Image Processing*, 2005.
- [15] P. K. Saha. Tensor Scale: A local morphometric parameter with applications to computer vision and image processing. *Computer Vision and Image Understanding*, 99:384–413, 2005.
- [16] D. Zhang and G. Lu. Review of shape representation and description techniques. *Pattern Recognition*, 37(1):1–19, 2004.