Technical Posters Proceedings

XXI Brazilian Symposium on Computer Graphics and Image Processing

October 12-15, 2008, Campo Grande, MS, Brazil

Edited by

Maria Andréia Formico Rodrigues (UNIFOR)
Célia Zorzo Barcelos (UFU)
Rafael Bruno Almeida (UFMS)
Marcelo Ferreira Siqueira (UFMS)

Table of Contents

Image Processing

Computer Graphics, Vision, and Modeling

Preface

The present volume constitutes the proceedings of the Technical Posters of the XXI Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI 2008), held on October 12-15, in Campo Grande, MS, one of the most beautiful cities in Brazil.

This year, 19 papers were accepted out of a total of 29 submissions, which led to an acceptance rate of 65.52%. Each paper was evaluated by three referees. Based on their technical reviews, we selected the accepted articles for exhibition in the SIBGRAPI 2008 Technical Poster Session.

We are very thankful to the authors that chose SIBGRAPI as the place to present their work. Credit also goes to the seventeen Program Committee members, who kindly donated their time and experience to evaluate the submissions. They were chosen based on their extensive expertise and their reputation for good and fair judgement. We also thank the local organization committee, the SIBGRAPI Co-Chairs Marcelo Walter and Claudio Jung, the twenty-seven reviewers for their invaluable collaboration and informative reviews, and all the authors who submitted papers, whose contributions were fundamental to the quality of the technical program. Finally, we would like to thank the University of Fortaleza (UNIFOR) and Federal University of Uberlandia.

We wish all the participants a pleasant and learning journey at SIBGRAPI 2008 and, particularly, during the Oral Presentations of the Technical Posters.

Campo Grande, Brazil, October 12-15, 2008.

Maria Andréia Formico Rodrigues and Célia Zorzo Barcelos
Technical Posters SIBGRAPI 2008 Co-Chairs

Program Committee

Technical Posters Co-Chairs
Maria Andréia Formico Rodrigues (Universidade de Fortaleza – UNIFOR)
Célia Zorzo Barcelos (Universidade Federal de Uberlândia – UFU)

Program Committee

Adilson Gonzaga (Escola de Engenharia de São Carlos – EESC/USP)
Agma Traina (Instituto de Ciências Matemáticas e de Computação – ICMC/USP)
Carla Maria D. S. Freitas (Universidade Federal do Rio Grande do Sul – UFRGS)
Denise Guliato (Universidade Federal de Uberlândia – UFU)
Eraldo Ribeiro (Florida Institute of Technology)
Esteban Walter Gonzalez Clua (Universidade Federal Fluminense – UFF)
Isabel Harb Manssour (Pontifícia Universidade Católica do Rio Grande do Sul – PUCRS)
Léo Pini Magalhães (Univerdidade Estadual de Campinas - UNICAMP)
Luiz Marcos G. Gonçalves (Universidade Federal do Rio Grande do Norte – UFRN)
Marcelo da Silva Hounsell (Universidade do Estado de Santa Catarina – UDESC)
Marcos Craizer (Pontifícia Universidade Católica do Rio de Janeiro – PUC-Rio)
Maria Cristina F. de Oliveira (Instituto de Ciências Matemáticas e de Computação – ICMC/USP)
Maurilio Boaventura (Universidade Estadual Paulista Júlio de Mesquita Filho – UNESP-Rio Preto)
Nina S. T. Hirata (Instituto de Matemática e Estatística – IME/USP)
Olga Regina Bellon (Universidade Federal do Paraná – UFPR)
Thomas Lewiner (Pontifícia Universidade Católica do Rio de Janeiro – PUC-Rio)
Washington Mio (Florida State University)

Image Processing

DICOM Server applied to Medical Image Processing

Barberis, Luís S; Graffigna, Juan P; Lucena, Fabián M;
Gómez, M Eugenia; Romo, Raúl

Abstract

This work describes the design and implementation of a DICOM server that manages medical image studies for their further processing. It is based on connectivity specifications of a massive clinic standard, and has the capacity to manage studies and to run different processing algorithms, according to a decision making system by means of databases. The design provides great flexibility in terms of configuration and operations assignment, and permits compatibility with DICOM images transmission systems. The goal of the development is to apply automatic algorithms of clinical image processing carried out by research groups. By means of this system, it is possible to validate clinically new algorithms and employ them for the diagnosis and patient monitoring.

1. Introduction

The present work is framed within the Medical Image Management and Processing area and aims at managing multi-modal studies for image processing.

Generally, algorithms developed by research groups present inconvenients that make their validation and clinical implementation difficult. On the one hand, a highly trained medical and technical staff are required, who usually work in very demanding environments that make the use of the developed tools impossible. On the other hand, clinic validation of the algorithms requires a sufficient number of studies to achieve strong results. This task is generally performed by selecting studies and processing them manually.

In the above mentioned circumstances, clinic implementation of specific algorithms for different pathologies or modalities becomes impracticable. In order to overcome these difficulties, the design and development of a system that tries to associate processing algorithms research with their clinic implementation are presented in this work, implementing the DICOM standard [1] for the reception, treatment, and sending of such images.

2. Design

This system was designed for image processing management allowing the user to send studies for their processing and to receive the results.

The server, developed as a DICOM communication module, is configured to analyze each received study in order to determine if a set of predefined conditions are fulfilled. If this occurs, the corresponding processing algorithms of the corresponding studies are executed.

The studies can be received directly from a modality or a visualization workstation. The aim is to delegate to the server all complex processing calculations. It allows to speed up diagnosis and to provide new functions that visualization softwares do not generally have. This function solves problems associated with the minimum requirements needed for certain operations.

The system configuration and the state information are stored in data tables. The following three tables are used for the configuration: “Operation”, “Condition” and “Application Entity”. Each line of “Operation” defines a processing type that the server offers clients through different ports. “Condition” contains the list of condition for each operation. “Application Entity” stores the information of the destination entity for the results of the different operations.

The other function of the data system is the recording of the new incoming studies and the storing of information for their later monitoring. This information is also used as a decision making tool due to, the data of stored studies allows to define if the needed conditions for the beginning of a process associated to an operation have been fulfilled.

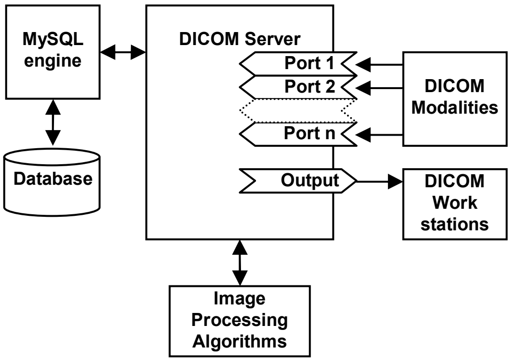


Figure 1 – System Overview

Another aspect to be considered in the design is that the management system must be capable of attending several clients simultaneously. The attention of different listening ports and the answer to the request of different clients are independent processes. In informatics this is implemented by using execution threads. Each reception or sending task is implemented in an independent thread, allowing the system to deal with a request and to immediately wait for others. Figure 1 shows the general structure of the system.

3. Materials and methodology

The methodology employed comprises the following stages: Planning and requirement specification, Analysis, Design, Development, and Experimentation. Unified Modeling Language (UML) diagrams, widely spread, were used for the modeling process.

For the development of the system, GPL software tools were used exclusively. The Code::Blocks application, version 8.02, was chosen as the primary tool for the development of the server, associated with the MinGW compiler version 3.4.5. The database engine chosen is MySQL® version 5.0.

The CTN library (Central Test Node) [2] in its version 3.0.4 was used for the management of DICOM data and for the setting of associations.

The software was programed in C in order to have compatibility with CTN libraries. The query system to the database is carried out by using the API of MySQL included in the software itself.

The processing routines were provided by the people in charge of the project titled “Multimodal Medical Image Fusion and Analysis Applied to Diagnosis and Radiotherapy”, carried out in conjunction with the Medical Technology Cabinet (*Gabinete de Tecnología Médica – GATEME*) from the School of Engineering of the National University of San Juan, Argentina, and the Foundation School of Nuclear Medicine (*Fundación Escuela de Medicina Nuclear – FUESMEN*) in Mendoza, Argentina. In that project, algorithms of high computational cost have been developed to achieve a good quality fusion among intra and inter modality images. These algorithms operate over PROCIMA libraries [3], developed by GATEME.

4. Results

The system has been tested at laboratory level at GATEME, and partially in the clinical environment at FUESMEN. During tests carried out at GATEME to verify the connectivity and the sending of studies, CTN [4] test applications and the eFilm Workstation visualization application, test version 2.1.4, were previously configured for the connection.

Different “Case Studies" based on the characteristics of the imaging processing algorithms developed were performed for the experimentation, namely:

- Intermodality/intrapatient registration for radiotherapy.
- Intramodality/intrapatient registration for treatment monitoring.
- Processing of single studies for diagnosis.
- Others.

The results obtained in the tests were satisfactory in general achieving the management of the study storing and the correct performance of the different treatment algorithms. At FUESMEN, tests for the reception and sending of studies of different modalities have been carried out, and the first algorithm for its clinic validation is been implemented.

5. Conclusions

This work has allowed the development of a Processing Server in accordance with DICOM standard. In this way, it is possible to integrate new processing algorithms to clinical environments. The application allows an automatic management of studies. The design presented in this work offers a system capable of making decisions based on a data set included in a DICOM object. The decision made determines which operation must be executed, which studies take part, and to whom the results must be sent.

The system can receive several studies, execute algorithms, or send results simultaneously using different processing threads. Operations, ports, conditions, etc., can be quickly modified by means of a SQL database.

For the design and the development of the server, GPL software tools have exclusively been used. This permits to improve and expand the research tasks performed in the region.

A weakness of the system is the necessity of employing automatic algorithms that require no user intervention. Even though there is a tendency to this kind of techniques, there is no possibility to interact with the user during processing tasks.

Regarding the implementation of the system in a clinical environment, it is necessary to execute it for a prolonged time in order to evaluate the system and algorithms.

6. References

[1] ACR-NEMA: Digital Imaging and Communications in Medicine (DICOM) Standard. Draft Versión 2007.

[2] Mallinckrodt Institute of Radiology: CTN library.

[3] F.Lucena, A.Valdez, M.Gómez, J.P.Graffigna, O.Nasisi: Diseño de una Librería para Procesamiento de Imágenes Médicas (PROCIMA) . 35 JAIIO. Simposio de Informática y Salud 2006.

[4] D.Beecher, L.Duan, S.Hoffman, S.Moore, P.Weng: CTN Utility Programs - A Guide to Programs for Testing and Demonstrating DICOM Functionality - Mallinckrodt Institute of Radiology Electronic. 1999.

7. Acknowledgments

This work was funded by the FONCYT Project, PICT 2003 No 430. ANPCYT.

Watershed Transform based Interactive Image Segmentation Tool

Bruno Klava and Nina S. T. Hirata
Institute of Mathematics and Statistics - USP
Department of Computer Science
Rua do Matão, 1010, São Paulo, SP - 05508-090 Brazil
{klava, nina}@ime.usp.br

Abstract

Watershed from markers and hierarchical watershed are approaches suitable for interactive image segmentation: in the former, the user can edit markers to control the segmentation result; in the latter, the user can select an image partition from a nested set of partitions. We propose an interactive image segmentation tool that allows transition from one approach to other and thus the combination of the strengths of both.

1 Introduction

The *watershed from markers* and the *hierarchical watershed* are approaches derived from the watershed transform [2]. In spite of being approaches suitable for interaction, they both have shortcomings: when using markers, a considerable interaction effort is demanded if there are a lot of regions to be marked; when using the hierarchy, borders are necessarily a subset of the borders of the primitive regions and there are partitions that can not be obtained due to the way the hierarchy is built.

Previous works that use both approaches together [4, 6, 8] consider the region adjacency graph (RAG) of an initial fine partition as the underlying structure. Markers drawn on the image are mapped to the RAG vertices and operations are performed on the RAG, determining a partition with region level precision (i.e., all regions are unions of primitive regions). In these works, the computation of markers corresponding to a given partition is not considered.

If one desires to place contours beyond those at the border of the primitive regions, it is important that the watershed from markers operates at a pixel level precision. Moreover, computing the markers corresponding to a given partition may be useful, for instance, to segment a sequence with similar frames in videos.

Therefore, a desirable characteristic in a segmentation tool is the ability to work with hierarchies (region level precision) and with markers at a pixel level precision. In the next section we extend our previous work [5], briefly discussing how to map a partition of a hierarchy to a set of markers that recover the same partition and, conversely, how to map a partition corresponding to a set of markers to a hierarchy of partitions. Then, we describe the interaction possibilities that have been implemented in an interactive image segmentation tool. The tool, written in Java, implements the watershed algorithm by the *image foresting transform* [3]. In Section 3 we present the conclusions.

2 Switching back and forth between the watershed approaches

The transition from the hierarchical to the markers approach is addressed through the *minimal seed set* problem, which consists in finding a minimal set of markers that can be used to obtain a partition P by the watershed [1]. If the markers are computed in the RAG and watershed from markers is applied on the pixels graph, the resulting partition P' may differ from P. However, the set of markers computed in the RAG tend to be compact and located more at the center of each region, whereas those computed on the pixels tend to be located at the borders of the regions. We have a proof that P is also an optimal partition, equivalent in cost to P'. For user edition, markers computed in the RAG are more appropriate.

In the inverse transition, as the primitive regions are atomic units in the hierarchy, if a partition P contains borders that crosses some of them, then P can not be represented in the hierarchy. In this case, a possible approach is to set the same label for all the pixels within each region, for example, the most frequent label among them. Furthermore, it is also necessary that each set of primitive regions with same label (that actually make up a region) is a connected set considering the edges of the spanning tree of the RAG from which the hierarchy is constructed.

Details of these transitions will be included in an extended version of this work. Next, we describe the implemented interaction possibilities that explore the strengths of both approaches.

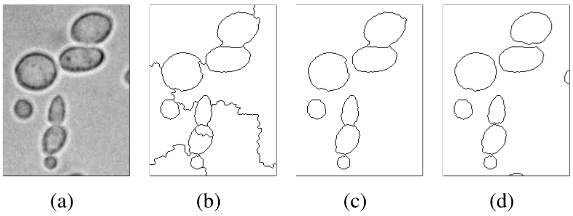


Figure 1. Segmenting using hierarchy and markers: (a) Original image. (b) Partition selected on the hierarchy. (c) Partition after welding regions. (d) Partition with refinements at pixel level precision.

Construction of hierarchies of partitions: by uniform and synchronous floodings, as described in [7], ranking regions by their contrast and/or size. To navigate through the hierarchy, the tool provides a slider control to select a threshold value for the weight of the edges to be suppressed in the RAG. It is also possible to apply local operations (merge and refine) as described in [9].

Merging of adjacent regions with a welding brush: more flexible than the local merge operation over the hierarchy, it can weld regions using edges that are not in a minimum spanning tree of the RAG. This operation was used to obtain the partition of Fig. 1(c), as it cannot be found on the original hierarchies.

Automatic generation of markers: to refine the partition at a pixel level precision, for example, to separate the cells as in Fig. 1(d).

Use of the hierarchy in a selected region: This enables the user to refine (re-segment) a region. For instance, to obtain the primitive regions of the land on a satellite image, one could first design markers by hand in order to separate land from water, then, applying the hierarchy in the land region, select the finest partition, as in Fig. 2.

3 Conclusions

We presented an approach that enables the use of the hierarchical watershed and the watershed from markers interchangeably. Having this possibility in hand, one can take advantage of the best features of each approach, obtaining the desired results with less interaction effort than when using only one of the approaches. The approach has been implemented in an interactive segmentation tool, which will be made publicly available.

Acknowledgements: This work is supported by CNPq.

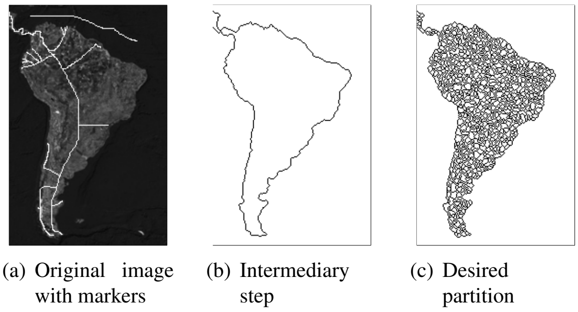

(a) Original image with markers (b) Intermediary step (c) Desired partition

Figure 2. Using hierarchy restricted to a region: (a) Original image. (b) Partition separating land from water. (c) Primitive regions of the land region.

References

[1] R. Audigier and R. Lotufo. Seed-relative segmentation robustness of watershed and fuzzy connectedness approaches. In *Proceedings of SIBGRAPI 2007*, pages 61–70. IEEE Computer Society, 2007.

[2] S. Beucher and F. Meyer. The morphological approach to segmentation: the watershed transformation. In E. Dougherty, editor, *Mathematical morphology in image processing*, chapter 12, pages 433–481. M. Dekker, 1993.

[3] A. X. Falcão, J. Stolfi, and R. de Alencar Lotufo. The image foresting transform: Theory, algorithms, and applications. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(1):19–29, 2004.

[4] H. K. Hahn and H.-O. Peitgen. IWT - interactive watershed transform: A hierarchical method for efficient interactive and automated segmentation of multidimensional gray-scale images. In *Medical Imaging 2003: Image Processing. Proceedings of the SPIE*, volume 5032, pages 643–653, 2003.

[5] B. Klava and N. S. T. Hirata. Watershed segmentation: Switching back and forth between markers and hierarchies. In *Proceedings of ISMM 2007 - 8th International Symposium on Mathematical Morphology*, volume 2, pages 29–30, October 10–13 2007.

[6] B. Marcotegui and F. Zanoguera. Image editing tools based on multi-scale segmentation. In H. Talbot and R. Beare, editors, *Proceedings of ISMM 2002*, pages 127–135, 2002.

[7] F. Meyer. Hierarchies of partitions and morphological segmentation. In *Proceedings Scale-Space '01*, pages 161–182. Springer-Verlag, 2001.

[8] F. Meyer. An overview of morphological segmentation. *International Journal of Pattern Recognition and Artificial Intelligence*, 15(7):1089–1118, 2001.

[9] F. Zanoguera, B. Marcotegui, and F. Meyer. A toolbox for interactive segmentation based on nested partitions. In *Proceedings of ICIP*, volume 1, pages 21–25, 1999.

Marvin – A Tool for Image Processing Algorithm Development

Gabriel Ambrósio Archanjo, Fabio Andrijauskas, Danilo Rossetto Muñoz
Marvin Project
{leirbag.arc,biozit,munozdanilo}@gmail.com

Abstract

This work presents, Marvin, tool for image processing algorithm development. This tool is composed by a framework and an image manipulation program that supports extensibility through plug-ins. The framework, used to develop the plug-ins, provides features for image manipulation, algorithm analysis and integration with the base application. Marvin is suitable for image processing algorithm development and prototyping and for education purpose.

1. Introduction

Image processing is increasingly used for many purposes and usually involves the development of algorithms to filter and extract information from images. Nowadays, many tools for image manipulation are available and they allow users to manipulate images and develop their own algorithms for image processing, such as GIMP [4] and Photoshop [5].

Marvin, as an alternative tool for image manipulation and image processing algorithm development, has an important differential from the majority of these kinds of tools, focused on image manipulation. Marvin is also an image manipulation program, but it is focused on the development of image processing algorithms. For this purpose, Marvin provides an easy-to-use white-box framework, that requires the knowledge of the framework internal structure from the plug-in developers [1]. The framework provides features to facilitate the development of image processing algorithms, and it is open source and multi-platform, considering that it is developed in Java.

2. Plug-ins

Every Marvin feature to manipulate images is externally implemented through plug-ins. The framework defines the interfaces and provides features to manipulate the images. A Marvin plug-in must implement the interface MarvinPluginImage that specifies which methods must be implemented to manipulate an image. Once Marvin plug-ins were implemented in Java, after the plug-in compilation, the class file must be copied to the plug-ins folder that is inside the Marvin folder. Every plug-ins located in that folder will be automatically loaded into Marvin image manipulation program in the next execution.

In many cases, it is interesting to use a plug-in inside another and the Marvin framework allows plug-ins integration. The plug-in that implements an edge detection algorithm, for example, uses the gray scale plug-in in the beginning of the process.

3. Performance Meter

Many image processing algorithms can be divided in few processes. An algorithm, for example, may be divided in three processes to find and segment the interest points of an image: representing the image in gray scale, finding the interest points and segmenting the image[2]. Each process has a different computational cost and the performance meter allows developers to analyses the performance of the entire algorithm and their processes individually, considering the number of steps and the execution spent time. Thus, it is possible a better understanding of the algorithm performance and facilitates optimizations, since is visible what algorithm processes are costly and most interesting to be optimized. Another performance meter application is to determine the complexity of an algorithm, Big-O notation, analysing the number of steps executed for different image resolution.

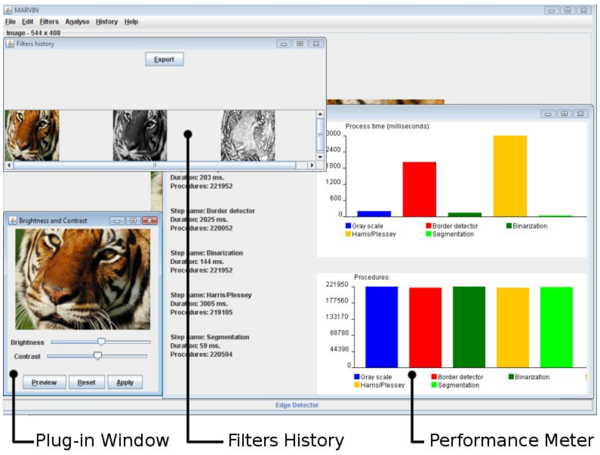
4. Plug-in history

In some cases to achieve some result in an image, it is necessary to use a few processes. In the beginning of an image processing method, compensations can be done, such as noise and blur reduction[3]. Preparing an image for pattern recognition, for example, in some applications it is interesting to remove noise, increase contrast and emphasize edges. For these cases that multiple processes are used, Marvin History stores all plug-ins and their configurations applied to an image. The history can be exported as an image strip containing the configuration and resulting images of each used plug-in. This feature allows future analyses of used processes and facilitates a continuous development of image processing processes.

5. Integrated Graphical User Interface

The plug-ins can have user specified attributes that determines how it will work and the Marvin framework provides features to integrate a GUI (Graphical User Interface) with these attributes. Thus, the plug-in developer sets the relation between the plug-in attributes and the interface components, added to the plug-in window. When the component value is changed, the associated attribute value is changed automatically. The developer does not worry about handling component events.

The Figure 1 shows a plug-in interface using a integrated GUI with a plug-in window, a filters history and the performance meter displaying the result of Interest Point plug-in.



Duration: 203 ms.
Procedures: 221952

Step name: Border detector
Duration: 2025 ms.
Procedures: 220052

Step name: Binarization
Duration: 144 ms.
Procedures: 221952

Step name: Harris/Plessey
Duration: 3005 ms.
Procedures: 219105

Step name: Segmentation
Duration: 59 ms.
Procedures: 220594

Figure 1. Marvin image manipulation program interface.

6. Current plug-ins

Currently, there are plug-ins developed for the Marvin application that provide the following features: image segmentation considering interest points, edge detection, image halftoning, Gaussian blur, image difference, image blending, steganography, color intensity histogram, brightness and contrast manipulation, sepia, mosaic, flipping, cropping, gray scale and negative colors conversion. Some of these plug-ins are contributions of third-party developers.

7. Conclusions and Future Work

Marvin is a tool composed by a framework and an image manipulation program. Features, such as image manipulation methods, integrated graphics user interface, filter history, plug-in integration and performance meter allow analysis, study, prototyping and development of image processing algorithms.

Marvin is developed in Java and it is open source. Every feature for image processing are developed externally through plug-ins, also in Java, by the open source community and are available at http://marvin.incubadora.fapesp.br/.

For the next version, new statistical charts support, new interface components and multi-image processing for motion analysis are expected.

8. References

[1] M.E. Fayad, D.C. Schmidt, “Object-Oriented Application Frameworks”, *Communications of the ACM*, USA, 1997.

[2] C. Harris, M. STEPHENS, “A combined corner and edge detector”, *Proceedings of The Fourth Alvey Vision Conference*, Manchester, pp 147-151. 1988.

[3] GONZALES Rafael, WOODS Richard, *Digital image processing*, Second Edition Pretenci Hall, 2001

[4] The GNU Image Manipulation Program – Official Website. http://www.gimp.org/. Accessed June 26, 2008.

[5] Photoshop CS3 Editions – Official Website. http://www.adobe.com/products/photoshop/. Accessed June 26, 2008.

Color Gradient Map-Oriented Anisotropic Diffusion Filtering

Leandro Coser[1], Antonio C. Sobieranski[1], Adiel Mittmann[1],
Aldo von Wangenheim[1], Eros Comunello[2]
[1]LAPIX – Laboratory for Image Processing and Computer Graphics,
UFSC – Federal University of Santa Catarina, Brazil
[2]Universidade do Vale do Itajaí, Brazil
{leandro.coser,asobieranski,adiel,awangenh,eros}@inf.ufsc.br

Abstract

Anisotropic diffusion filtering is a well-established technique for image enhancement that smooths images without destroying edge information. However, when many filtering iterations are applied, edges gradually fade away and are ultimately smoothed by the process. We propose the adoption of a color gradient map to guide the smoothing so that clearly-defined edges are preserved even after many iterations. Preliminary experiments show good results when compared with the regular anisotropic diffusion filter.

1. Introduction

The anisotropic diffusion filter [2, 3] is a powerful image processing technique for noise removal. The filter works by performing smoothing on the image but at the same time preserving the boundaries between different regions. The process is controlled by a diffusion matrix that measures the color variation on the neighborhood of a hot spot.

When the filter is applied over many iterations, the diffusion matrix slowly becomes adapted to the new local colors. When the original image contains tenuous edges, the smoothing process will gradually erase all edge information as iterations are sequentially applied. In this work, we propose an improvement to the anisotropic diffusion filter that introduces a color gradient map, which behaves as static boundary evidence, enhancing the response of the filter when applied several times to an image containing tenuous edges.

2. Orienting the Anisotropic Diffusion Filter

The anisotropic diffusion filter used in this paper has been extensively discussed [2], and can be regarded as a convolution technique with an adaptive matrix-valued kernel that performs a special smoothing on images: It inhibits the smoothing on edge pixels and stimulates it on internal regions. The basic diffusion equation [1] for an image $I(x, y)$ with M channels and a signal initialized with $u(x, y, 0) = I(x, y)$ is

$$\partial_t u_i = \mathrm{div}\left(D\left(\sum_{k=1}^{M} \nabla u_k \nabla u_k^{\mathrm{T}} \right) \nabla u_i \right), \qquad (1)$$

where D is a matrix-valued function, and $i = 1, \ldots, M$ are the individual channels.

As the process of diffusion is carried on through several iterations, the edges of the original image gradually fade away, because the function D takes into account only the results from the previous iteration. After a certain amount of iterations, even edges that were initially well defined may

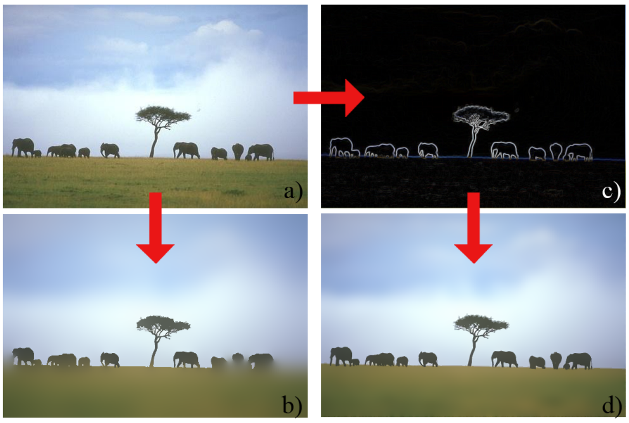


Figure 1. Overview of traditional and gradient map-oriented filtering. a) is the original image; b) is the traditional filtering; c) is the gradient map generated from a); and d) is the filtering oriented by the gradient map.

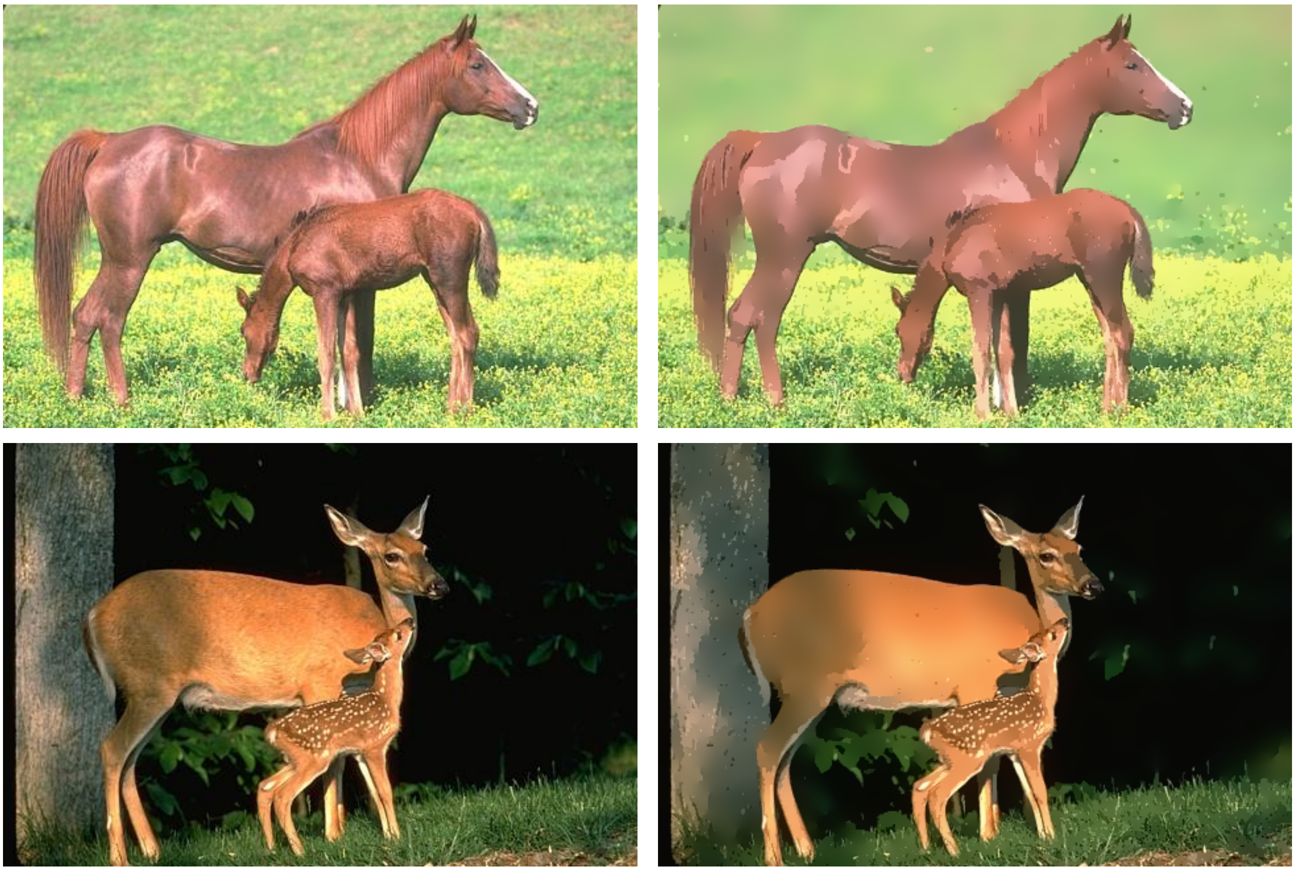


Figure 2. On the left, original images; on the right, images filtered with the approach we propose.

become blurred. In order to keep well-defined edges present in the original image, we propose the introduction of a static factor G_i in Equation 2 that depends on the original image only, thus avoiding the effects introduced by the repeated application of the diffusion filter. This static factor is calculated from a color gradient map produced from the original image and remains constant throughout all iterations. An overview of this process is shown in Figure 1, where both versions of the filter are applied over 300 iterations.

In order to accommodate G_i, Equation 2 can be rewritten as

$$\partial_t u_i = \operatorname{div}\left(D\left(G_i \sum_{k=1}^{M} \nabla u_k \nabla u_k^{\mathrm{T}}\right) \nabla u_i\right). \qquad (2)$$

The gradient map G is calculated by a convolution operation using the following masks:

$$I_x = \frac{1}{4}\begin{pmatrix} -b & 0 & b \\ -a & 0 & a \\ -b & 0 & b \end{pmatrix} \text{ and } I_y = \frac{1}{4}\begin{pmatrix} -b & -a & -b \\ 0 & 0 & 0 \\ b & a & b \end{pmatrix},$$

where $a = 2(\sqrt{2} - 1)$ and $b = (2 - \sqrt{2})$. The modulus of the vector (I_x, I_y) is then used as an estimation of the gradient for each channel.

Figure 2 shows an example of the color gradient map-oriented filter. While the internal regions were greatly smoothed, the distinction between them remains clear.

3. Conclusion and Discussions

The anisotropic diffusion filter is a powerful tool that enhances an image by smoothing regions while adopting a sophisticated border-preserving scheme. We have shown how this filter can be further improved by using a color gradient map that remains unchanged over iterations. This way, we ensure the preservation of edges that were well defined in the original image.

References

[1] T. Brox. *From Pixels to Regions: Partial Differential Equations in Image Analysis*. PhD thesis, Saarland University, 2005.

[2] J. Weickert. *Anisotropic Diffusion in Image Processing*. Teubner-Verlag, 1998.

[3] J. Weickert. Applications of nonlinear diffusion in image processing and computer vision. *Acta Math. Univ. Comenianae*, 70:33–50, 2001.

OCR4JkanjiCards: Exploring Japanese Character Recognition

Oswaldo Bassani Neto, Celso Setsuo Kurashima, Marcelo Zanchetta do Nascimento
Universidade Federal do ABC
{oswaldo.bassani, celso.kurashima, marcelo.zanchetta}@ufabc.edu.br

Abstract

The OCR4JkanjiCards system is presented in this work, as a proposal for taking photos of Japanese characters using a smartphone and have them automatically recognized within a digital Japanese dictionary. Image processing techniques and neural network strategies are necessary, as well as the exploration of several programming tools.

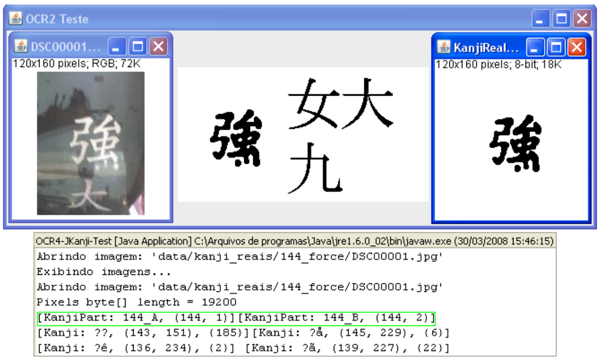
1. Introduction

The increase in processing power and memory available on mobile devices like cell phones, smart-phones or handheld, allow user to have pocket dictionaries providing fast access to information.

The goal of this work is to explore a set of tools that allows joining several features in a unique system. Our OCR4JkanjiCards system is based on character recognition, and on the processing of photos taken from street plates, building facades and other texts in the real world.

2. System Overview

The OCR4JkanjiCards system basically works with an image of a kanji as the input and gives back the kanji identification as the output, through a user interface as shown in Figure 1. Kanji's are the complex characters used in Japanese language.



OCR4-JKanji-Test [Java Application] C:\Arquivos de programas\Java\jre1.6.0_02\bin\javaw.exe (30/03/2008 15:46:15)

```
Abrindo imagem: 'data/kanji_reais/144_force/DSC00001.jpg'
Exibindo imagens...
Abrindo imagem: 'data/kanji_reais/144_force/DSC00001.jpg'
Pixels byte[] length = 19200
[KanjiPart: 144_A, (144, 1)][KanjiPart: 144_B, (144, 2)]
[Kanji: ??, (143, 151), (185)][Kanji: ?å, (145, 229), (6)]
[Kanji: ?ê, (136, 234), (2)] [Kanji: ?ã, (139, 227), (22)]
```

Figure 1: OCR4JkanjiCards User Interface.

The inner window in the upper left part of Figure 1 is a picture, taken with a cell phone camera, of a poster in the street containing a kanji character. After several image processing steps the kanji was clearly separated in the upper right inner window of Figure 1. In the lower part of this figure, the result of recognition process is shown in the line marked by the green box, in the form of ID codes of kanji's. These codes refer to a Japanese dictionary where we can obtain the meanings in English. In this case, we used the "Java Kanji Flashcards 500" [1] data base[1].

We created OCR4JkanjiCards system based on the project called OCR4J [2]. This is an open source library that allows the creation of applications with character recognition like OCR (Optical Character Recognition), but is limited to A-Z characters.

So our system started from the integration of the OCR system with the Kanji education software, and went to the development of image processing techniques for Kanji characters. We are interested in investigating how kanji's can be efficiently processed for recognition tasks, given that the character writing is complex and composed of non-connected lines.

Correlated work can be found in references [3] and [4]. The first one is a project of kanji dictionary for mobile devices.

3. Methodology

Since the purpose of this experiment is the use of images acquired from photographic cameras, we apply some image processing techniques in order to convert input images into binary image format before using the OCR system.

So, the initial step is to convert the input colored image to gray scale and to apply a median filter. Then, it is converted to a binary image, with an appropriate threshold. Finally, this image is passed through an

[1] "Java Kanji Flashcards 500", available in: http://nuthatch.com/java/kanjicards/, accessed in March 2008.

invert function to get it prepared for the kanji recognition process.

However, we have further removed excess of spurious granularity of the image that appeared in our experiments. This was done by applying a region growth to the image and manually removing unwanted regions.

The character recognition sub-system makes use of a neural network of multilayer perceptron type. This neural network can vary its architecture, for example we can change the number of input elements and the number of internal layers as well as the number of neurons on the layers. The only predetermined value is the number of output neurons, which must equal the number of characters we want to recognize.

Before training the neural network, however, we have realized that some kanji's are not fully identified by the neural network because they are composed by two or more sets of connected pixels. In the example shown in Figure 1, the left part of that kanji is disconnected from the right part of it.

For these cases, we propose to break the kanji into two sub-parts. By doing so, these two sub-parts can be easily recognized individually.

4. Results

Our experiments are still in the beginning and were carried out on commodities desktop PC's. We have trained a neural network with 17 kanji's. For each kanji we used two images of training for each one. We also added one kanji with two parts for training the neural network, i.e., with four more training pictures.

In the OCR4JkanjiCards, we have used an image-processing tool called ImageJ [5][6], which was easy to use in the software environment during the process that convert images into binary format.

In Figure 2 we show the example of the result of image processing techniques applied on the kanji with two parts. The leftmost image is the original input image. The other images to the right are the results of the steps of gray scaling, filtering, threshold, binary, invert, and spurious removing.

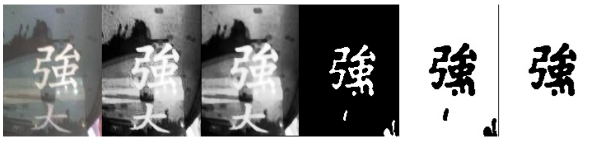


Figure 2: image processing steps of the OCR4JkanjiCards system.

All training images and test images passed through this image processing steps. And, after training the neural network we have got the result shown in Figure 1, where the kanji of two parts was successfully recognized. The same positive result was observed in all testing image for kanji recognition.

5. Conclusions

In this paper we propose the OCR4JkanjiCards system, which is a Japanese character (kanji) recognition tool based on the OCR4J environment. The goal is to use it on mobile devices like cell phones.

We have presented initial experiments with the OCR4JkanjiCards system. This includes our proposal for breaking kanji's into two parts during recognition process on the neural network, as a proof of concept. Consequently, not only the system can run faster but also the recognition becomes easier.

As future work, there are many more experiments to be carried. For example, to expand the quantity of kanji's to be trained on the neural network, and to measure the processing time on mobile devices.

In the image processing steps, we plan to investigate the detection of the base lines to find the characters, as observed by Koga *et al* [3].

6. References

[1] S. Wood Ryner, N. Chikamatsu, H. Nozaki, S. Yokoyama, S. Fukada - "Java Kanji Flashcard 500: Kanji, Java, and the World Wide Web", 16th annual Unicode Conference, Tokyo, 1998.

[2] Project: "Ocr4j", Optical Character Recognition library for Java, available in: http://sourceforge.net/projects/ocr4j/, accessed in March 2008.

[3] M. Koga, R. Mine, T. Kameyama, T. Takahashi, M. Yamazaki, T. Yamaguchi - "Camera-based Kanji OCR for Mobile-phones: Practical Issues", proceedings of the 2005 Eight International Conference on Document Analysis and Recognition (ICDAR'05), 2005.

[4] H. A. Rowley, M. Goyal, J. Bennett - "The Effect of Large Training Set Sizes on Online Japanese Kanji and English Cursive Recognizers", proceedings of the Eighth International Workshop on Frontiers in Handwriting Recognition (IWFHR'02), 2002.

[5] W. Bailer - "Writing ImageJ Plugins - A Tutorial", software documentation.

[6] Software: "ImageJ", available in: http://rsb.info.nih.gov/ij/, accessed in March 2008.

Fusion of Infrared and Visible Spectra Face Recognition Methods

Giovani Chiachia, Aparecido Nilceu Marana
UNESP – Faculdade de Ciências – Departamento de Computação – Bauru – SP – Brazil
giovanichiachia@gmail.com, nilceu@fc.unesp.br

Abstract

In general, face recognition systems are based on visible spectrum images and, in order to have good performance, they need to work in light-controlled environments. However, the performance of such systems decrease significantly when illumination changes. On other hand, Long-Wave Infrared (LWIR) face imagery is insensitive to illumination changes and gives the temperature pattern from the face to be recognized. The purpose of this work is to assess the performance of the fusion of well-known statistical visible and LWIR-based methods for face recognition.

1. Introduction

While humans can easily recognize faces in adverse situations and even after years, machine face recognition remains a big challenge in the fields of Computer Vision and Pattern Recognition [1].

To overcome this challenge, alternative sensor modalities (e.g. 3D range image) have been proposed. One of the new sensing modalities for face recognition is the infrared (IR), which can measure the temperature emitted by the face [1].

Besides being insensitive to light changes, the subsurface anatomical information captured by IR sensors is believed to be unique for each person [1].

Due to all these characteristics and regarding previous researches [2], it is expected that using IR spectrum images together with visible spectrum images can lead to more robust and efficient face recognition systems.

The main goal of this work is to assess the performance of the fusion of well-known statistical visible and LWIR-based methods for face recognition.

2. Database

In our experiments, it was used the University of Notre Dame (UND) time-gap face database [3], which has a large collection of images acquired by visible and LWIR spectrum cameras. The images were acquired once a week, with most of the subjects participating several times, totalizing 2023 images in each spectrum, with neutral and smiling facial expressions. During a given acquisition session, 4 images per subject were taken, being 2 with neutral and 2 with smiling expressions.

In our experiments, 187 subjects from the UND face database were used in the training phase and other 54 subjects were selected for the gallery and the probe sets. Each of these 54 subjects attended at least 7 and at most 10 acquisition sessions. The first session of each subject was used in the gallery set and the remaining 6 to 9 sessions constitute the probe set. Hence, this work also has taken into account the recognition performance over time.

Figure 1 shows an example of visible and IR spectrum images of the same face. In the IR spectrum image (right), the gray level ranges from black (cold) to white (hot).

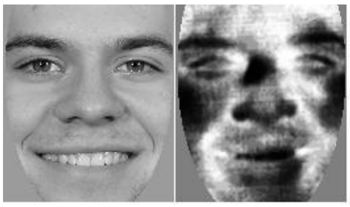


Figure 1. Visible spectrum image (left) and IR spectrum image of the same face [3].

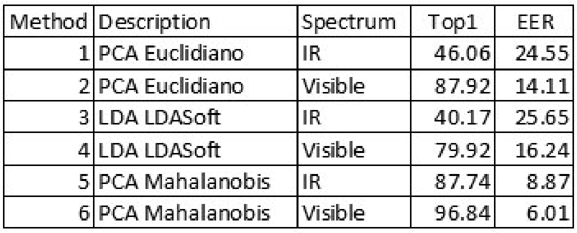
3. Face Recognition Methods and Their Correlation

Three different face recognition methods were used in this work: Principal Component Analysis (PCA) with Euclidean distance, Linear Discriminant Analysis (LDA) with LDASoft distance, and PCA with Mahalanobis Angle [4], as implemented in [5].

Each method was applied individually in both spectra. Table 1 shows the obtained results, where Top 1 means the correct recognition rate considering only the most similar recovered face, and EER means the equal error rate (the false acceptance and the false rejection rates are equal).

In order to predict the performance of the 15 fusion possibilities, it was obtained the *Q statistic* measures of dependence, $Q_{i,k}$, which for two methods i and k, ranges from -1 to 1 [6]. For statistically independent methods, $Q_{i,k}$ is 0. For statistically correlated methods, $Q_{i,k}$ tends to 1, and for inversely correlated methods, $Q_{i,k}$ tends to -1.

Table 1. The six face recognition methods and their individual performances.



Method	Description	Spectrum	Top1	EER
1	PCA Euclidiano	IR	46.06	24.55
2	PCA Euclidiano	Visible	87.92	14.11
3	LDA LDASoft	IR	40.17	25.65
4	LDA LDASoft	Visible	79.92	16.24
5	PCA Mahalanobis	IR	87.74	8.87
6	PCA Mahalanobis	Visible	96.84	6.01

4. Fusion

In this work, the fusion was carried out in the score level. Three score normalization approaches were assessed: Min-Max, Double Sigmoid and Tanh-estimators [7]. The fusion techniques assessed were: sum, max, min, and product.

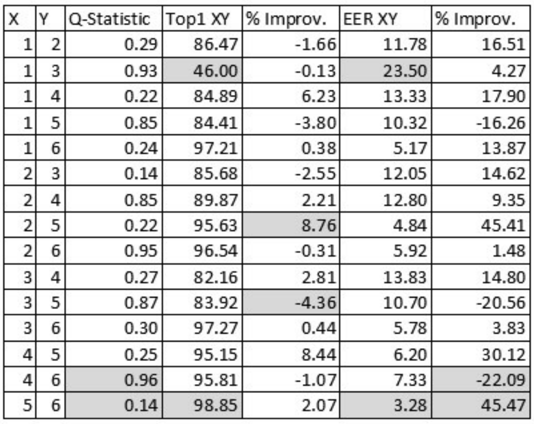
5. Experimental Results

In our experiments, the Double Sigmoid score normalization approach and the product fusion technique have shown to be more regular (better mean improvement) than the others. Therefore, they were chosen to denote the overall fusion performances.

As expected, it can be observed in Table 2 that the correlation between methods applied on different spectra is much smaller than the correlation of methods applied on the same spectrum, which indicates that they hit and fail in different situations for many probes. For instance, the correlation of methods 5 and 6 was 0.14, the lowest for different spectra methods, and the correlation of methods 2 and 4 was 0.85, the lowest for methods on the same spectrum.

We can also observe in Tables 1 and 2 that there is a relationship among the *Q Statistic,* the Top1 individual rates, and the performance of the fusion. When the *Q Statistic* is low (lower than 0.5) and the Top 1 individual rates are high (greater than 50%), the performance of the fusion compared with the best individual rates always increases. Hence, the overall best performance (98.85% for Top 1, and EER=3.28) was obtained with the fusion of two good individual methods, 5 and 6, that present the lowest correlation rate (0.14).

Table 2. Overall performance from the fusion of methods X and Y as identified in Table 1. The best and worst results are highlighted.



X	Y	Q-Statistic	Top1 XY	% Improv.	EER XY	% Improv.
1	2	0.29	86.47	-1.66	11.78	16.51
1	3	0.93	46.00	-0.13	23.50	4.27
1	4	0.22	84.89	6.23	13.33	17.90
1	5	0.85	84.41	-3.80	10.32	-16.26
1	6	0.24	97.21	0.38	5.17	13.87
2	3	0.14	85.68	-2.55	12.05	14.62
2	4	0.85	89.87	2.21	12.80	9.35
2	5	0.22	95.63	8.76	4.84	45.41
2	6	0.95	96.54	-0.31	5.92	1.48
3	4	0.27	82.16	2.81	13.83	14.80
3	5	0.87	83.92	-4.36	10.70	-20.56
3	6	0.30	97.27	0.44	5.78	3.83
4	5	0.25	95.15	8.44	6.20	30.12
4	6	0.96	95.81	-1.07	7.33	-22.09
5	6	0.14	98.85	2.07	3.28	45.47

6. Conclusion

The experimental results obtained in this work suggest that the fusion of IR and visible spectra-based methods may improve significantly the face recognition performance.

Acknowledgments

The authors thank CAPES for financial support.

References

[1] S. G. Kong et. al, “Recent advances in visual and infrared face recognition—a review”, *Computer Vision and Image Understanding 97*, (2005) 103–135;

[2] X. Chen, P. J. Flynn, K. W. Bowyer, "Infra-red and visible-light face recognition", CVIU 99 (3), 332-358, 2005.

[3] Biometrics Database Distribution. The Computer Vision Laboratory, Univ. of Notre Dame, http://www.nd.edu/~cvrl/ , 2002.

[4] A. M. Martinez, A. C. Kak, "PCA versus LDA", IEEE PAMI, 2001.

[5] D. Bolme, R. Beveridge, M. Teixeira and B. Draper,“The CSU Face Identification Evaluation System: Its Purpose, Features and Structure”, *International Conference on Vision Systems*, pp 304-311, Graz, Austria, April 1-3, 2003.

[6] Kuncheva, L.I., Whitaker, C.J., Shipp, C.A., Duin, R.P.W., “Is Independence Good For Combining Classifiers?”, 0-7695-0750-6/00 IEEE, pp. 168-171, 2000.

[7] Jain, A., Nandakumar, K., Ross, A., "Score normalization in multimodal biometric systems", *Pattern Recognition 38*, pp. 2270-2285, 2005.

SAR Image Compression Using Bounded Variation Component Analysis

Regis C. P. Marques* regismarques@cefetce.br
Daniel S. Ferreira* daniels@itti.cefetce.br
Fátima N. S. Medeiros† fsombra@deti.ufc.br
Janaina V. Cruz* janaina@cefetce.br
Maria H. V. Duarte* heveline@cefetce.br

Abstract

*This paper proposes a scheme to compress SAR (Synthetic Aperture Radar) images aiming for preserving fine details such as bright targets. This scheme consists in decomposing a SAR image into bounded variation (BV) and oscillating components. The adaptive compression scheme is a modified version of the well known SPIHT method named in this paper as MSPIHT. The experiments using real SAR images were compared with JPEG*2000 *and the conventional SPIHT. The assessment results indicated a superior performance of method in preserving fine details.*

1 Introduction

Synthetic aperture radar (SAR) images are useful sources of information for several applications in remote sensing of environment. In the last years, the amount of SAR data has increased, and motivated researchers to develop algorithms for SAR image compression. Traditional algorithms fail to compress SAR images, particularly due to speckle phenomena. The algorithm introduced in [1] requires pre-processing steps including speckle filtering, which degrades fine details and edges.

Using the discrete wavelet transform (DWT) efficient compression procedures can be developed in a multiresolution space. Several methods based on wavelets [5] are available in the SAR image compression literature. This approach can be accomplished by decomposing SAR images in subbands and searching for the significant information for lossless compression. Regarding texture, wavelets coefficients present a subband frequency representation, where the significant texture information can be discriminated in middle frequency channels [5].

In this paper we adopted the scheme presented in [4] to decompose the SAR image. Let $f : \Omega \rightarrow \Re$ be the image decomposed into $u \in BV$ component (BV - bounded variation space) and $v \in L^2$ component (L^2 - oscillating space), such that $f := u + v$. The u component which is a cartoon representation of f is a simplified piecewise smooth approximation. Our proposed framework tracks regions with targets in the u component and employs a lossless compression method to these regions, in order to preserve them.

This paper describes the proposed method in Section 2. We present experimental results and some concluding remarks in Section 3.

2 Method

The BV and L^2 decomposition spaces present distinct features. Thus, u (low frequency information) and v (high frequency information) components provide image analysis using distinct bands in the same resolution. Similar to [4], the proposed scheme employs the *Total Variation Minimization (TV)* method to obtain the u component. The oscillating component v is computed as a TV residual information.

To perform target detection, the u component is subdivided into blocks using the quadtree decomposition presented in [5] according to a defined criterion. If a block meets the proposed criterion, it is not subdivided, otherwise it is decomposed into four blocks. The criterion is defined as $\sigma_L < \sqrt{L}(\sigma_{L-1})$, where L is the quadtree decomposition level and σ_L is the standard deviation of the inspected block.

The proposed scheme is shown in Fig.1. Our aim is to allocate to fine details scales more bits in the compression process. Thus, in the highest decomposition scale (16x16 blocks) is used the arithmetic coder to provide a lossless data compression. We compress the other scales (32x32, 64x64,128x128) using the SPIHT algorithm [2] with decreasing bit rates.

The displayed result in Fig. 2(a) is a scene of the coastal zone in RN-Brazil. The image size is 256x256 pixels with

*Departamento de Telemática, Centro Federal de Educação Tecnológica - Laboratório de Processamento de Sinais, Fortaleza, CE, Brazil.

†Departamento de Engenharia de Teleinformática, Universidade Federal do Ceará - Grupo de Pesquisa em Processamento de Imagens, Fortaleza, CE, Brazil.

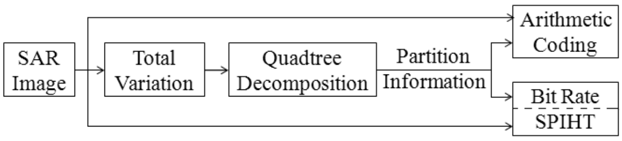
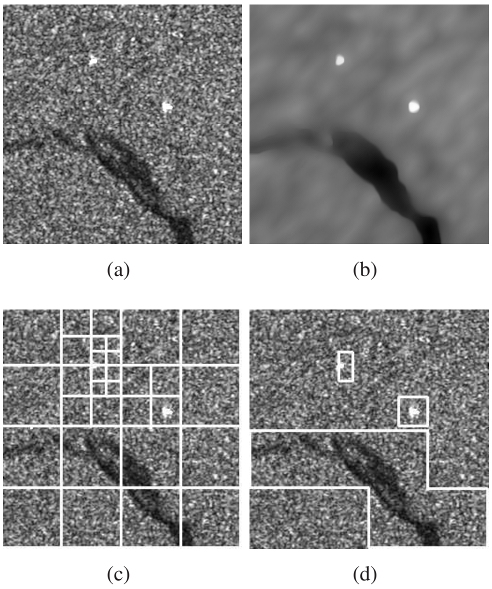


Figure 1. Proposed method

12.5 m resolution. It was acquired in amplitude detection from the RADARSAT system. The decomposition (BV) result can be observed in Fig. 2(b). One can observe preserved targets (bright points in top center and an oil slick down center) in the u component. The quadtree decomposition provides a partition (grid) displayed in Fig. 2(c). Bright targets can be observed in fine scales 16x16 and 32x32, and the oil slick area is divided in 64x64 blocks, as shown in Fig. 2(d).


(a) (b)

(c) (d)

Figure 2. (a) The original SAR image, (b) the u component, (c) the quadtree decomposition (grid) superimposed in the original image and (d) the detected targets.

3 Results and Conclusions

The quantitative measures used to assess the experimental results were **PSNR** (Peak Signal to Noise Ratio) and standard deviation to mean ratio σ_m, as used in [5]. In addition, the **MIE** (Max Intensity Error) [3] was locally computed as a measure of fine details preservation. The modified SPIHT (MSPIHT) outperformed the standard JPEG2000 compression method according to all assessment measures. Furthermore, it was superior to the conventional SPIHT method related to MIE and σ_m.

The results are displayed in Table 1 using two different bit rates, 0.2 bpp and 1.0 bpp. One can observe that the proposed method presented similar PSNR values as SPIHT with a reduction about 0.5 dB.

Table 1. Assessment measures

Bit Rate		0.2 bpp	1.0 bpp
Proposed Method	PSNR(dB)	31.47	37.47
	MIE(dB)	12.30	7.780
	σ_m	0.194	0.277
SPIHT	PSNR(dB)	31.50	38.02
	MIE(dB)	13.80	10.41
	σ_m	0.166	0.231
JPEG2000	PSNR(dB)	31.13	37.34
	MIE(dB)	17.50	10.00
	σ_m	0.154	0.274
Original Image	$\sigma_m = 0.278$		

The results showed that the SPIHT performance related to fine detail preservation can be improved without changing the bit rate significantly.

Acknowledgments

This research was supported by CNPq under grant No. 485070/2006-5. We also would like to thank Renato Feijó for providing the RADARSAT image.

References

[1] X. Hou, G. Liu, and Y. Zou. Sar image data compression using wavelet packet transform and universal-trellis coded quantization. *IEEE Trans. on Geoscience and Remote Sensing*, 42(11):2632–2641, November 2004.

[2] A. Said and W. Pearlman. A new, fast, and efficient image codec based on set partitioning in hierarchical trees. *IEEE Trans. Circuits Syst. Video Technol.*, 6:243–250, June 1996.

[3] M. Sun and C. Li. A new complex sar image compression quality metric. In *Proc. of IGARSS 2005*, volume 7, pages 4697–4700, 2005.

[4] L. A. Vese and S. J. Osher. Image denoising and decomposition with total variation minimization and oscillatory functions. *Journal of Mathematical Imaging and Vision*, 20(1-2):7–18, March 2004.

[5] Z. Zeng and I. G. Cumming. Sar image data compression using a tree-structure wavelet transform. *IEEE Trans. on Geoscience and Remote Sensing*, 39(3):546–552, March 2001.

A Fourier Transform-based Method to Fusion IKONOS Data

Juliana Gouveia Denipote and Maria Stela Veludo de Paiva
Escola de Engenharia de São Carlos – EESC. Universidade de São Paulo – USP
{judeni, mstela}@sel.eesc.usp.br

Abstract

The principal objective of image fusion in remote sensing is to obtain high-resolution multispectral images that can combine the spectral characteristic of the low-resolution multispectral bands with the spatial information of the high-resolution panchromatic band. A method based on Fourier transform is proposed in order to obtain good spatial and spectral resolutions using all bands of IKONOS satellite. Quantitative measurements were applied to compare the results of the proposed method with IHS, IHS enhanced by Fourier transform and wavelet-based methods.

1. Introduction

Modern satellites, like IKONOS, are capable to producing high spatial resolution panchromatic (PAN) band with one meter or less. Simultaneously, they produce good multispectral (MS) bands. Spectral information from MS bands is useful to differentiate land cover classes, like vegetation, bare soil, water, roads and buildings. On the other hand, the spatial information from PAN is necessary for an accurate description of image details, such as shape, contours and features [1]. Image fusion is used to combine both PAN and MS bands to obtain high-resolution multispectral images.

The traditional image fusion methods, such as IHS, can keep almost the same spatial resolution as PAN, but they distort the spectral characteristics of the original MS bands [2]. Image fusion methods using frequency domain processing, like those based on wavelet transform (WT) [3-5], preserve good spectral information, but their spatial visual results are not satisfactory.

IHS fusion methods enhanced by Fourier transform have been very suitable in preserving both spectral and spatial information [1, 6], but they are limited to red (R), green (G) and blue (B) bands, excluding the near-infrared (NIR) band.

2. Proposed FT-based Image Fusion

For IKONOS images, it is necessary to perform some pre-processing, as the pixel sizes from PAN and MS are different (1 meter resolution for PAN and 4 meters for MS). One pixel from MS must be resampled for sixteen pixels.

In a remote sensing image, the details, like objects' edges, are a result of a high contrast between features, for example a light rooftop and dark ground. High contrast in spatial domain is high frequencies in the frequency domain. High frequencies are richer in PAN than in MS. On the other hand, spectral information appears as low frequencies in the frequency domain, which are richer in MS than in PAN.

The proposed method consists in obtain the spectral information from MS band by applying a low pass circular filter in the frequency domain, and obtain the spatial information from PAN by applying a high pass circular filter. No IHS transform is used in order to employ all the four IKONOS MS bands. The steps to fusion resampled R and PAN bands are illustrated in figure 1. The same steps must be applied to G, B and NIR bands.

For the filters, the optimum cutoff frequency must be measured accordingly to the spatial resolution, which depends on the image sampling interval following the Nyquist sampling criterion [1]: the sampling interval is in inverse proportion to the sampling frequency. Therefore the maximum frequency of an image is in inverse proportion to its spatial resolution. If the ratio of pixel resolution between PAN and MS is 1:m, the MS maximum frequency is 1/m PAN maximum frequency, that is, the cutoff frequency is equal to the maximum frequency of MS. But the maximum frequency of an image is $0.5\,\Delta x^{-1}$ (where Δx is its pixel size in meters). IKONOS PAN has one meter spatial resolution, so the maximum frequency is 0.5 cycles per meter. The maximum frequency of MS is 1/4 of that of the corresponding PAN, so the cutoff frequency for an IKONOS image is 0.125 cycles per meters.

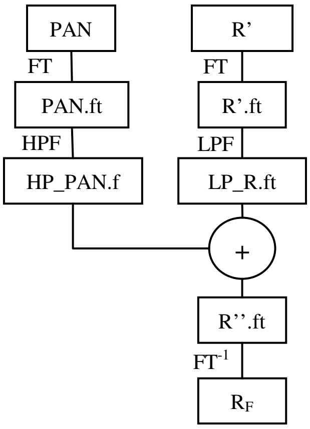


Figure 1 - Schematic diagram for FT-based fusion of PAN and R

The proposed method, using ideal circular filter, results in an image that presents artificial artifacts due to the abrupt combination of low and high frequencies. Some filters like Gaussian, Butterworth and Hanning were tested to smooth the results. The more suitable was Hanning. The high and low pass filter must be complementary to not lose or overlap any information.

Figure 2 shows the result of the proposed fusion method using Hanning filter. It could be noted that figure 2c has high spatial resolution (compare it to PAN in figure 2a), while maintaining the original colors (compare it to RGB composition in figure 2b).

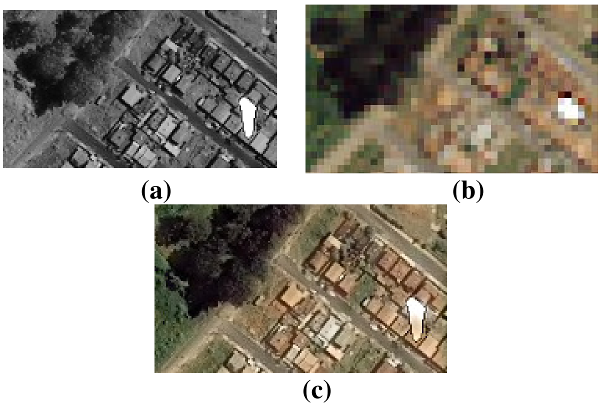


Figure 2 – (a) original PAN, (b) original RGB composition, (c) result of the proposed method

After fusion, the MS bands are expected to be as similar to the original bands as possible. Correlation coefficient indicates how an image is "similar" to another. The proposed method has the higher correlation coefficient (see table 1) followed by IHS+FT method, indicating that these methods attain more spectral information than the others. When comparing correlation coefficient from IHS+FT and the proposed method, R band improves 7%; G band improves 8%; and B band remains almost the same.

Table 1 – Correlation index between the fused bands and their corresponding original bands for different fusion methods.

Method	R	G	B	NIR
IHS	0.34	0.24	0.27	----
WT	0.64	0.61	0.53	0.77
IHS+FT	0.77	0.70	<u>0.75</u>	----
FT (proposed)	<u>0.84</u>	<u>0.78</u>	0.74	<u>0.86</u>

3. Conclusions

A method based on filter in the frequency domain was proposed to the fusion of multiespectral satellite bands. It was compared to other reported methods and has the advantage of using any number of bands. In visual analysis, it can be observed that high information from PAN was added to MS information without distorting the original colors due to the smooth join of the PAN and MS. From a statistical analysis employing correlation coefficient, the proposed method attains more spectral information when compared with reported IHS, WT and IHS+FT methods. Further, differently IHS and IHS+FT, the proposed method can use the NIR band.

Acknowledgments: The authors would like to acknowledge CNPq for the financial support.

4. References

[1] Li, J.; Luo, J.; Ming, D.; Shen, Z., "A new Method for merging IKONOS Panchromatic and Multispectral Image Data", Proceedings of the IEEE International Geoscience and Remote Sensing Symposium, 2005, IGARSS '05, 25-29 July 2005, Vol. 6, page(s): 3916- 3919.

[2] Amolins, K.; Zhang, Y.; Dare, P., "Wavelet-based image fusion techniques – an introduction, review and comparison", ISPRS Journal of Photogrammetry and Remote Sensing, v. 62, issue 4, p. 249-263, September 2007.

[3] Gungor, O.; Shan, J., "Evaluation of Satellite Image Fusion using Wavelet Transform", 20th ISPRS Conference, Istanbul, Turkey, July 12-23, 2004.

[4] Tu, T.M.; Su, S.C.; Shyu, H.C.; Huang, P.S., "A new look at IHS-like image fusion methods, Information Fusion", Volume 2, Issue 3, September 2001, Pages 177-186.

[5] Wang, Z.; Ziou, D.; Armenakis, C.; Li, D.; Li, Q., "Comparative Analysis of Image Fusion Methods", IEEE Transactions on Geoscience and Remote Sensing, vol. 43, no. 6, June, 2005.

[6] Ling, Y.; Ehlers, M.; Usery, E.L.; Madden, M., "FT-enhanced IHS transform method for fusing high-resolution satellite images", ISPRS Journal of Photogrammetry and Remote Sensing, Volume 61, Issue 6, February 2007, Pages 381-392.

A new Content-Based Image Retrieval methodology based on Image Registration algorithms processed in Grid Computing

Marcelo Costa Oliveira[1], Walfredo Cirne[2], José Flávio M. V.Junior[2], Paulo M. de Azevedo-Marques[3]

[1]*Departamento de Ciência da Computação/Universidade Federal de Alagoas (UFAL) – Campus Arapiraca, Arapiraca-AL, Brasil*

[2]*Departamente de Sistemas e Computação/Universidade Federal de Campina Grande (UFCG), Campina Grande – PB, Brasil*

[3]*Departamento de Clínica Médica/Universidade de São Paulo (USP), Ribeirão Preto – SP, Brasil*

Email: oliveiramc@gmail.com

Abstract

The Content-Based Image Retrieval (CBIR) has received great attention in the medical community because it is capable of retrieving similar images that have known pathologies. This work proposes a new methodology based on higher processing provided by the Grid Computing (GC) technology to achieve the CBIR using Registration algorithms. We evaluate the accuracy in retrieving images of the same plane, weighted and acquisition sequence related to a reference image. This works uses a heterogeneous image database and uses two anatomic regions (breast and head) as reference. The IR procedure uses Mean Square Metric (MSM) and Cross Correlation (CC). Both metrics showed a higher efficiency, MSM obtained 83% (breast) and 94% (head) precision average, the CC showed 81% (breast) and 98% (head). The higher computational cost related to the Image Registration were amortized by the GC, showing a potential low cost solution for secure data interchanging and integrating multiple hospitals and clinics.

1. Introduction

In the last 10 years, the CBIR has been one of the most studied computational vision technique. Through CBIR, it is possible to use a reference image to find similar diagnosed images [1]. Although part of the information about the diagnosis can be in the DICOM (Digital Imaging and Communication in Medicine) header, this textual tag frequently shows high error rate. Literature has reported cases with 16% of error [1]. This way, the medical community has emphasized the adoption of alternative methods in relation to the traditional methods based on the data manual insertion into the DICOM files.

The Image Registration (IR) is made of methods capable of find spatial transformations necessary to map homologous points between two images. The transformations are done in an iterative way until the best match is obtained between the reference and target images. However, all this iterative process results in a high computational cost that makes intractable to use the IR to find the best match between a reference image and a big image server [2].

The Grid Computing (GC) technology represents the most recent and promising tool in distributed computing. GC is the integration of many computers distributed geographically, making it possible to create a virtual computing platform, giving to users and institutions a virtually unlimited capacity to solve problems related to the storage and access of data, and also to process applications with high computational costs [3].

The main goal of this work is to develop an algorithm capable of using the GC to make viable the retrieval of similar images through the adoption of the IR techniques based on the Mean Squares Metric (MSM) and Cross Correlation (CC) techniques.

2. Materials and Methods

The application was developed in the operation system GNU/Linux Debian using the Java

1.5 programming language. The database image was developed with PostgreSQL-8.1-3 and Hibernate 3.0 and it followed the DICOM compliances. For its assessment, a heterogeneous image database was used, with 3000 magnetic resonance images from different anatomic regions. The application has two CBIR modules. The first module uses the second-order Texture Analysis (TA), suggested by Halarick [4], to filter the 1000 most similar images into the second module. The second module uses the IR algorithms to find the similarity between an image defined by the user as a reference and the images filtered by the first module. The application starts after the specialist select a reference image. When the first module is finished, the images are classified according to the smallest value of Manhatann Distance between the TA of reference image and the database images. Then, the 1000 most similar images are filtered to be used by the second module. The second module starts when the specialists select one of the two IR techniques implemented in this work: MSM and CC. The second module is processed on the OurGrid computational grid. OurGrid assumes that the parallel applications that run on it are Bag-of-Tasks (BoT) [3]. The application sorts in a list the most similar images according to the MSM or CC and show to the user.

3. Results

The efficacy of the IR algorithms used in CBIR was measured by the Precision versus Recall curves between the reference image and the images selected by the first module (Figure 1 and Figure 2). In all the experiments, the results produced by the IR algorithms were better than the traditional TA. The GC makes affordable for the application the high processing time of the IR algorithms (Table 1).

4. Discussion and Conclusion

Nowadays, the CBIR methods are still limited to detect specific diseases. Therefore, the CBIR methods need that new techniques of image processing be integrated into the traditional methods. The GC was fundamental to amortize the total processing time of the IR algorithms that permitted this work to show a new methodology to evolve CBIR's state of art techniques. The IR applied to the CBIR showed high precision, moreover, their results were better than the ones shown by the traditional TA. This work presents initial results in applying the IR as a CBIR technique where the main focus was to analyze the precision in retrieving different anatomies. Therefore, a deeper study still is necessary to analyze the real capacity of the RI methods in retrieval images from a pathological image database. A study in this way was already started.

	MSM		**CC**	
	Local	GC	Loca l	GC
Breast	118.39	5.31	115.88	4.78
Head	115.09	4.36	106.64	3.56

Table 1 - Compare time (minutes) between the local processing and the processing using the GC.

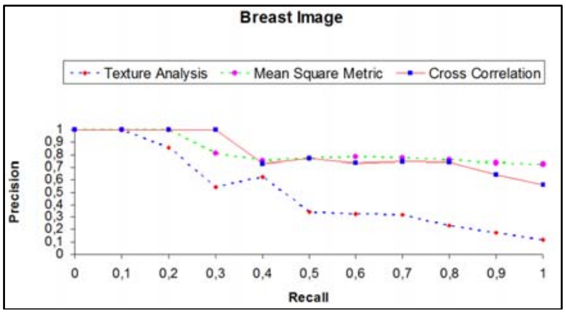


Figure 1 - Precision vs. Recall graph showing the retrieval techniques behavior in retrieve breast images.

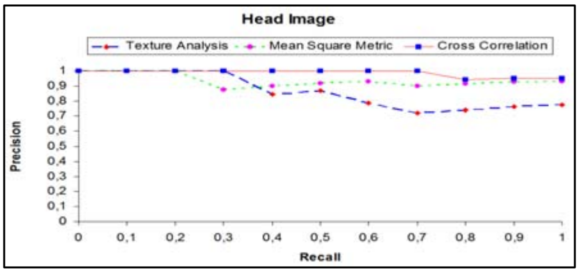


Figure 2 - Precision vs. Recall graph showing the retrieval techniques behavior in head images.

5. References

[1] H. Muller, N. Michoux, D. Bandon, and A. Geissbuhler, "A review of content-based image retrieval systems in medical applications - clinical benefits and future directions," *International Journal of Medical Informatics*, vol. 73, pp. 1-23, 2004.

[2] M. C. Oliveira, W. Cirne, and P. M. D. Marques, "Towards applying content-based image retrieval in the clinical routine," *Future Generation Computer Systems*, vol. 23, pp. 466-474, 2007.

[3] W. Cirne, F. Brasileiro, N. Andrade, L. Costa, A. Andrade, R. Novaes, and M. Mowbray, "Labs of the World, Unite!!!," *Journal of Grid Computing*, vol. 4, pp. 225-246, 2006.

[4] R. M. Haralick, Shanmuga.K, and I. Dinstein, "Textural Features for Image Classification," *IEEE Transactions on Systems Man and Cybernetics*, vol. SMC3, pp. 610-621, 1973.

Improving Content-based Image Retrieval by Combining Growing Hierarchical Self Organizing Map Classifiers for Color, Shape and Texture Features

Saulo de Tarso O. Rodrigues,* Tiago Oliveira de Albuquerque and Herman M. Gomes
Universidade Federal de Campina Grande, Departamento de Sistemas e Computação
Av. Aprígio Veloso, 882, 58109-970 Campina Grande, PB
saulo@dsc.ufcg.edu.br, tiagooa@lcc.ufcg.edu.br, hmg@dsc.ufcg.edu.br

Abstract

This paper presents the development of a content-based image retrieval system that combines a set of GHSOM (Growing Hierarchical Self Organizing Map)[4] classifiers taking as input color, shape and texture features. Experiments demonstrated combination scenarios that produced better accuracy in image classification.

1. Introduction

Automated search in large image databases is still an open research problem [6]. Most applications found in the literature still present low precision and recall, typically below the levels of 50% and 70% respectively. Moreover, some of the works that obtained better results either considered only a few number of very diverging groups on a large image set or were evaluated using a small image set (e.g. 100 different images) [1, 5]. Such limitations were the starting point for this research, which aimed at studying feature extraction techniques and a classification scheme based on the combination of individual classifiers.

2. Content-based Image Retrieval (CBIR)

A typical CBIR system contains three modules: Feature Extraction; Indexing; and Retrieval. There are several ways to extract image features; this research considered the most common features, which are color, shape and texture.

On current CBIR systems the color characteristic is frequently represented as histograms. For color images, the histograms are commonly calculated for each color component. The color spaces considered in this work were RGB, HSV and YCbCr, each color space provides a different set of features for the same image. Another feature extraction technique involves the description of shapes. In some areas, such as pattern recognition, shapes are important features to identify and distinguish objects [7]. Other than color and texture, the shape is extracted after the image has been segmented in regions or objects. A very used technique in the area of CBIR, which is related to the description of shapes, are Hu's moment invariants [2]. There are several different methods of extraction and representation of textures. LBP (Local Binary Pattern) [3] is a very good example which is invariant to rotation. Another well referenced method for texture representation are the Wavelets, which are capable of representing textures in multiple resolutions and scales.

The classification method adopted in this work was based on GHSOMs, which is a data structure similar to a B-Tree, but it is not usually balanced, nor its nodes have the same size. The nodes of a GHSOM point towards several other similar nodes and hold the tree property. To train each GHSOM, the same image base was used, which had all the color, shape and texture features extracted for each image in the set. Once GHSOM training ends, a tree-like map is created, where images that are similar, relative to a specific features, are represented in nearby neurons. During the recall phase, the specific features extracted from the query image are fed into the corresponding GHSOM. There will be a propagation of the input through the GHSOM up to its leaf nodes, where the most similar group of images to the input image are stored, relative to a given feature.

3. Proposed Approach

Classifier combination is a set of techniques that is proving useful in many application scenarios. Generally, the main advantage of using a combination method is because, since the classification methods are able to overcome the deficiencies of one another, it is possible to improve the overall system's accuracy. The classifier combination method adopted in this work is based on a voting process, where for every image returned by each selected classifier there will be a value associated to it, this value will be used to sort the final result, so the most voted image will be up front

*Saulo de Tarso O. Rodrigues is supported by CAPES

in the result, and considered to be the most similar image to the searched one. To calculate the vote for each image, the quantization error [4] produced by the GHSOM to the image is considered. This value is between zero and one, the closest to zero the more similar the image is. The layer of the GHSOM where the image appears is also used in the calculation. The following equation is used to calculate the vote for each returned image:

$$v_i = \sum_{g}^{G} (1 - QE_{g,i}) \cdot (\frac{LC_{g,i}}{LT_g}) \qquad (1)$$

where i is the image, G is the count of used GHSOM's, $QE_{g,i}$ indicates the quantization error of the image for a given GHSOM, $LC_{g,i}$ indicates the layer of the image for a given GHSOM and LT_g is the total number of layers of a given GHSOM, which results on a double precision value related to every retrieved image.

4. Experimental Results and Analysis

In order to test the proposed approach and classifier combination, part of Microsoft Research Cambridge Object Recognition Image database, version 1.0, has been used. This is a labeled database with 800 images, divided in 16 sets of 50 images each. Six GHSOM networks that were chosen to evaluate classifier combination were the ones which presented the best individual results. The tests first considered all possible combinations from $\mathrm{C}(6,1)$ (isolated results) to $\mathrm{C}(6,6)$, which gives a total of 63 tests. The crude combination does not exclude any results, so the bad images from all the combined GHSOMs will be returned as a valid result. To overcome this deficiency of the combination, a result filtering strategy has been adopted, which consisted of gradually varying the number of images recovered using the classifier combination and recalculating the precision for each number of returned images tested. This showed that, as the allowed number of returned images is reduced, more bad images start being rejected and better images start being accepted.

In comparison to the simple classification methods, classifier combination resulted on a considerable improvement for precision (see Table 1). The simple classification method that obtained the best precision was the one using the color feature on HSV space and 32 levels per component. Initially the simple classification has a better precision, when considering more than 70 images, but with a reduction on the resulting images, there is a natural reduction in the images that are not considered to be similar and the combination results are greatly superior to the ones of the simple classification. The best combination used HSV 16, RGB 32, LBP and Wavelets, increasing the mean precision in over 17% when the number of returned images is set to five. Even with the simpler combinations, the results are superior when the amount of returned images is under 40. The classification combination showed superior results to the simple classification, where only one feature of the image is used. Table 1 also shows that even the best individual result is not superior to the worst combined result.

Table 1. Experimental Results

Classifier	Precision (%)
WAV	12.50
LBP	18.38
Hu	10.78
RGB 32	17.05
HSV 16	20.33
YCbCr 32	19,55
RGB 32 - LBP	29.38
HSV 16 - RGB 32 - LBP	33.13
HSV 16 - RGB 32 - LBP - WAV	**33.52**
HSV 16 - RGB 32 - YCbCr 32 - LBP - WAV	31.48
HSV 16 - RGB 32 - YCbCr 32 - Hu - LBP - WAV	30.08

5. Conclusion

The main contribution of this research was the development of a combination technique for image classifiers based on GHSOM neural networks, improving the overall precision of a content based image retrieval system. The presented experimental results confirm that using a classifiers combination technique yields a considerable improvement of precision and recall for image retrieval.

References

[1] P. B.G., G. S.K., and B. K.K. Color and shape index for region-based image retrieval. *International Workshop on Visual Form*, 2001.

[2] T. S. Huang and Y. Rui. Image retrieval: Past, present, and future. *Symposium on Multimedia Information Processing*, 1997.

[3] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987, 2002.

[4] A. Rauber, D. Merkl, and M. Dittenbach. The growing hierarchical self-organizing map: exploratory analysis of high-dimensional data. *Neural Networks, IEEE Transactions on*, 13(6):1331–1341, 2002.

[5] E. Stian. Cbir distance measures and genetic programming. *Undergraduate Theses from Norwegian Classification of Images using Color*, 2006.

[6] R. Torres. Recuperao de imagens por contedo. *Revista Computao Brasil - Sociedade Brasileira de Computao*, 2006.

[7] D. Zhang and G. Lu. Review of shape representation and description techniques. *Pattern Recognition*, 37(1):1–19, 2004.

Computer Graphics, Vision, and Modeling

Surface Equalization

Francisco José Benavides Murillo [(1)] Douglas Navarro Guevara [(2)]
Francisco Torres-Rojas [(3)] Alexander Ramírez Gonzalez [(1)]
[(1)]Universidad de Costa Rica [(2)]Universidad Nacional de Costa Rica
[(3)]Instituto Tecnológico de Costa Rica

Abstract

Subdivision is a method to create smooth surfaces through a refinable schema of polygonal or triangular meshes. From a mathematical point of view, this technique is an application of a second generation wavelet analysis. At the same time, this analysis is based on the lifting schema and does not use any frequency criterion to define scaling functions or wavelets. This simplifies computational cost. Yet, a frequency analysis applied to a polygonal mesh, can provide an intuitive method to modify surface characteristics by creating an analogy to one-dimensional sound equalization. These foundations can be useful not only as a surface generation tool, but they are also naturally associated to finite element techniques and can be applied to a wide variety of simulation problems.

Keywords: subdivision surfaces, frequency analysis, Fourier, multirresolution, equalization, radiosity

1. Introduction

Subdivision techniques take a control mesh or polyhedron and add new polygons to it by refining its current faces. These new vertices are then perturbed according to some stencil rule, adding details and, in most cases, increasing surfaces's smoothness. Subdivision techniques only need a good and fast structure to preserve adjacencies between the polygons that approximate a surface. These polygons can be triangles, but many subdivision schemas can be adapted to other kinds of polygons [3].

As NURBS and Splines [8], subdivision techniques have a very desirable property: perturbation of one vertex can be made locally. By using subdivision and an appropriate stencil rule (such as Loop's [5]), more triangles can be added to an arbitrary mesh preserving continuity. This gives a nice tool to generate and manipulate smooth arbitrary surfaces, topologically equivalent to a basic shape.

An example of this process is shown in figure 1.

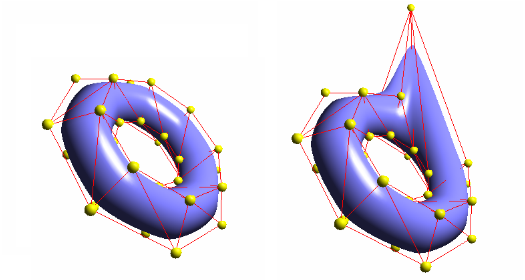


Figure 1. A deformed torus. Yellow spheres represent control points.

2. Equalizing Images and Polyhedra

From a mathematical point of view, equalizing an image is equivalent to applying an operator to a bidimensional array of numbers (a matrix). A frequency interpretation of this process is given by discrete bidimensional Fourier basis, which can be considered as a set of eigenvectors of a given operator [4]. From this discrete approach, it is relatively easy to define a multirresolution analysis (defined in [7]) for vector spaces of matrices, as an increasing set of finite-dimensional subspaces generated by a well chosen basis. These bases are called wavelets. Interpreting these subspaces and projecting the original matrix on them, we get a criterion to select coefficients in a basis expansion. As this criterion is based on a frequency filtering concept, we have a multirresolutive image equalizer. This is shown in figure 2.

Equalizing images is a somehow straightforward task because they can be sampled as regular grids and matrices. Extending this concept to general polyhedra is not trivial because they are not necessarily regular and cannot be generally represented by matrices.

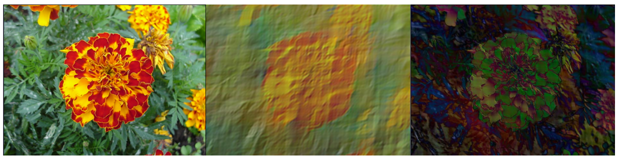


Figure 2. Original image. Low-pass filter and high-pass filter.

Nevertheless, in 1997, it was shown that subdivision techniques are a particular case of general multirresolution analysis on arbitrary topologies [6]. The method used to build wavelets with this degree of freedom is an application of the more general lifting method for second generation wavelets [9]. This technique has no need to use any frequency criterion to build a semiorthogonal multirresolution analysis. It is generally used to smooth a surface (which is equivalent to apply a low-pass filter).

3. Vibration Modes of a Polyhedron

In our research, we are trying to find a frequency criterion to control de degree of smoothness produced through a subdivision schema. As described above, Fourier analysis on matrices can be considered as a method to exploit conceptually an eigenvector basis of certain matrices. As described in [10], eigenvectors of a connectivity matrix can be considered the vibration modes of a polyhedron. Two characteristics of this technique is that the number of faces remains unchanged and it is applied globally. So, being able to link this approach with the lifting schema, could produce an equalizing subdivision stencil that could locally add smooth details or sharp details to a polyhedron. This could be done according to a criterion defined by the user. A global sharp surface subdivision process is shown in figure 3. As we pretend to have a better equalizing user's control of this, the figure is shown only for illustrative purposes.

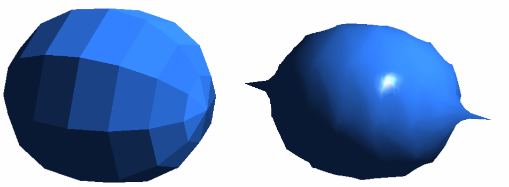


Figure 3. A non-smooth subdivision surface

A lifting schema based on frequency can be computationally slower than Lounsberry's [6], as it implies to get eigenvalues of large matrices (as large as the number of vertices of the polyhedron). Our hypothesis is that eigenvalues of a polyhedron's connectivity matrix should be naturally associated to its degree of smoothness in the lifting schema. We believe that the method described in [10] can be extended into an intuitive frequency-based method to design second generation wavelets. Equalization to design wavelets can be used directly on regular grids (such as figure 2) in a somehow straight application of classical wavelet theory. Yet, as far as we know, there is no general method to use the lifting schema with a frequency criterion to create a subdivision technique based on Fourier coefficients.

4. Future Research

As the lifting schema and subdivision methods can be used to define functions on a polyhedron, the techniques described above can be used to define an adaptive finite element set [1], which can be used in simulation problems such as radiosity [2]. It gives an intuitive method to construct wavelets based on discrete digital filters. This can be useful to naturally deal with discontinuities of functions defined on arbitrary topologies.

References

[1] S. C. Brenner and L. R. Scott, *The mathematical theory of finite element methods*, Springer-Verlag, Ed. Springer, 2000.

[2] M. F. Cohen and J. R. Wallace, *Radiosity and realistic image synthesis*, H. Brace and Company, Eds. Academic Press Professional, 1993.

[3] P. S. Denis Zorin, "Subdivision for modeling and animation," *Siggraph 2000 Course Notes*, vol. Dowonloadable from http://mrl.nyu.edu/ dzorin/sig00course/, 2000.

[4] M. Frazier, *An introduction to wavelets through linear algebra*, Springer-Verlag, Ed. Springer, 1999.

[5] C. T. Loop, "Smooth subdivision surfaces based on triangles," Ph.D. dissertation, University of Utah, 1987.

[6] M. Lounsberry, T. D. DeRose, and J. Warren, "Multirresolution analysis for surfaces of arbitrary topologial type," *ACM Transactions of graphics*, vol. 16, issue 1,, pp. 34–73, Juanuary, 1997.

[7] S. Mallat, *A wavelet tour of signal processing*, P. Ecole Poolytechnic, Ed. Academic Press, 1999.

[8] D. Solomon, *Curves and surfaces for computer graphics*, S. S.-B. Media, Ed. Springer, 2006.

[9] W. Sweldens, "The lifting schema: A construction of second generation wavelets," *Mathematics Subject Classification SIAM Journal on Mathematical Analysis*, vol. 42C15, 1995.

[10] G. Taubin, "A signal processing approach to fair surface design," *International Conference on Computer Graphics and Interactive Techniques Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, vol. IBM T.J. Watson Research Center, P.O.Box 704, Yorktown Heights, NY, pp. pp: 351 – 358, 1995.

Efficient Computation of Global Illumination for Image Synthesis

Anamaria Gomide, Danillo Roberto Pereira, Jorge Stolfi
Institute of Computing – University of Campinas (UNICAMP)
13083-970 , Campinas, SP, Brazil
anamaria@ic.unicamp.br,danillo.pereira@students.ic.unicamp.br,stolfi@ic.unicamp.br

Abstract

We are exploring various functional approximation schemes (such as radial bases and Shepard's interpolation) for precomputed radiance transfer (PRT), a finite element approach for real-time radiosity of complex scenes.

1. Introduction

Radiosity [1] is a general method for realistic rendering that uses finite element modeling to solve the Kajiya's *rendering equation* [3]. In this formulation, the light flow in the scene is found by solving a large system of linear equations $(I - R)\lambda = \epsilon$, where the vector ϵ gives the spontaneous light emission and λ gives the total emission (spontaneous plus scattered) of each scene element.

Implementation of the radiosity method requires choosing a finite element scheme to model the light flow. The goal of our project is to compare various schemes, such as radial bases [2] and Shepard's interpolation [5] basis - especially for use in real-time animation through *precomputed radiance transfer* [4, 6].

2. The rendering equation

The rendering equation can be written as $L = E + \mathcal{R}L$, where:

- $L(x, \vec{u})$ is the (unknown) *radiance function*, the total amount light emitted or scattered by the scene near the surface point x along directions near the unit vector $\vec{u}$;
- $E(x, \vec{u})$ is the *spontaneous emission function*, the amount of light emitted by the scene near x and along $\vec{u}$;
- $\mathcal{R}$ is the *light transfer operator*, that represents the transport and interaction of the light with the scene's objects. Its effect on an arbitrary function $F(x, \vec{u})$ is

$$(\mathcal{R}F)(x, \vec{u}) = \int_{\mathcal{S}} \rho(x, \vec{v}, \vec{u}) F(x', -\vec{v}) G(x, x', \vec{n}(x)) d\vec{v} \quad (1)$$

- $\mathcal{S}$ is the set of all directions (i.e. the unit sphere).
- $x' = x^*(x, \vec{v})$ is the first point on the scene's surface found by a ray that leaves x towards $\vec{v}$.
- $\rho(x, \vec{v}, \vec{u})$ is the *scattering coefficient*, the fraction of the incident light at the point x, coming from the direction $\vec{v}$, that is re-emitted along directions near $\vec{u}$;
- $G(x, x', \vec{n}(x))$ is the *geometric factor* that depends on the orientation of the surface at x.

In favorable circumstances, the solution of the rendering equation is
$L = (\mathcal{I} - \mathcal{R})^{-1}E$, where $\mathcal{I}$ is the identity operator. The *rendering operator* $(\mathcal{I} - \mathcal{R})^{-1}$ can be computed by Neumann's formula

$$(\mathcal{I} - \mathcal{R})^{-1} = (\mathcal{I} + \mathcal{R} + \mathcal{R}^2 + \mathcal{R}^3 + \cdots) \quad (2)$$

Each term $\mathcal{R}^k$ accounts for light that interacted k times with the surface of the scene before being observed.

3. Finite element radiosity

We use the term *site* to mean a pair $p = (\dot{p}, \vec{p})$ where $\dot{p}$ is a point on the scene's surface and $\vec{p}$ is the corresponding unit normal vector. For Lambertian radiosity, a *finite element* is a function ϕ defined on the scene's surface sites, such that $\phi(p)$ is nonzero only for a relatively small and compact set of sites (the *support* of ϕ).

In *point-based finite element methods*, the basis ϕ is defined by selecting a number of *sampling sites* $P = \{p_1, p_2, ...p_n\}$ on the scene's surface (see Figure 1(a)) and then choosing for each p_i a corresponding finite element ϕ_i that has p_i as its centroid (see Figure 1(b)).

3.1 Basis functions

In a *radial basis* each element ϕ_i is derived from fixed *mother function* ϕ by the formula $\phi_i(p) = \Phi\left(||p, p_i||/\alpha_i\right)$; where $||p, q||$, denotes the distance between the sites p and q, and α_i is a parameter that determines the "mean radius" of the element. See figure 2(a).

In *Shepard's basis* [5], each element is

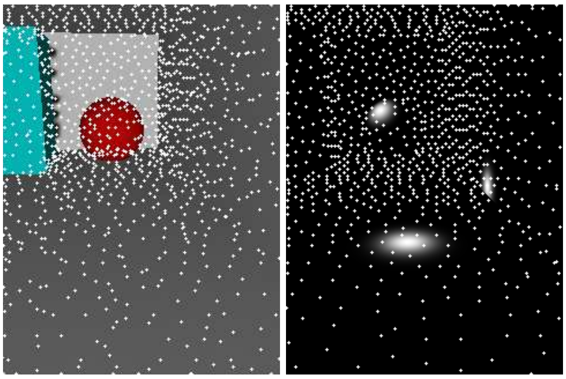

(a) (b)

Figure 1. The point-based FE method.

$$\phi_i(p) = \frac{w(p, p_i)}{\sum_{j=1}^{n} w(p, p_j)}$$

Here $w(p, q)$ is a non-negative function that tends to infinity when p tends to qe. g. $1/||p, q||$. See figure 2(b).

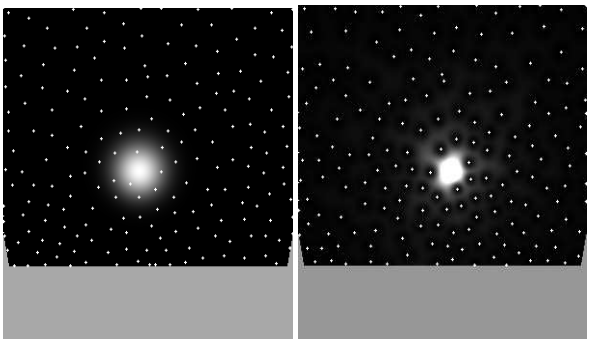

(a) (b)

Figure 2. Radial and Shepard elements.

4. Discretizing the rendering equation

In finite element radiosity, the rendering equation becomes a linear equation system $\lambda = \epsilon + R\lambda$, where $\lambda = (\lambda_1, \lambda_2, ..., \lambda_n)$ are the coefficients of L in the chosen basis, and $\epsilon = (\epsilon_1, \epsilon_2, ..., \epsilon_n)$ are the coefficients of E. Each element R_{jk} represents the influence of the light emitted by ϕ_k on the light scattered by ϕ_j. It is determined by the formula

$$R_{jk} = G(\dot{p}_k, \dot{p}_j, \vec{p}_j)\, \rho(p_j)\, V(\dot{p}_k, \dot{p}_j) \qquad (3)$$

Once the matrix R is available, the coefficients λ_i of the radiance function L can be computed by seting $\lambda \leftarrow (0, ..., 0)$ and then iterating $\lambda \leftarrow \epsilon + R\lambda$ until convergence.

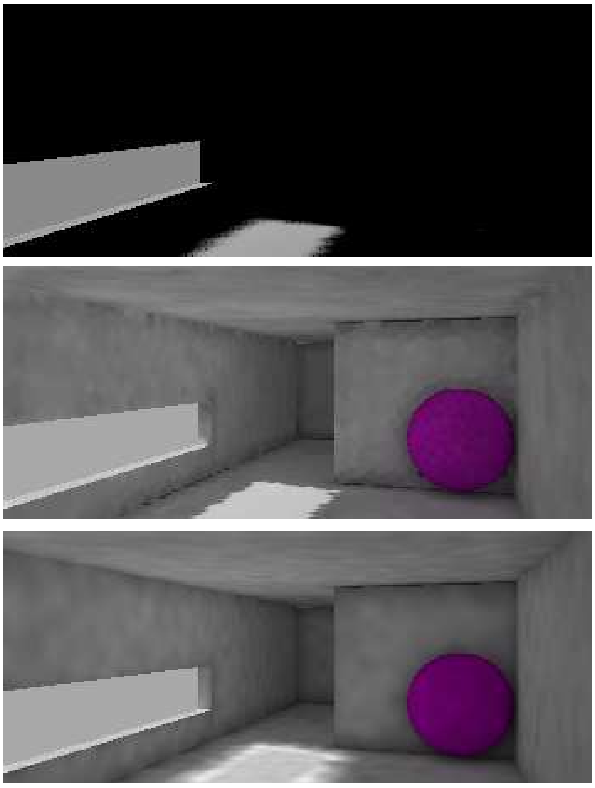


Figure 3. Rendered scenes.

5. Results

Figure 3 shows a test scene rendered without radiosity (top) and with ten radiosity iterations, using Shepard interpolation (middle) and a radial basis (bottom).

References

[1] M. F. Cohen and J. R. Wallace. *Radiosity and Realistic Image Synthesis*. Academic Press, 1993.

[2] M. S. Floater and A. Iske. Multistep scattered data interpolation using compactly supported radial basis functions. *J. Comput. Appl. Math.*, 73(1-2):65–78, 1996.

[3] J. T. Kajiya. The rendering equation. In *Proc. SIGGRAPH 1986*, pages 143–150, 1986.

[4] J. Lehtinen, M. Zwicker, J. Kontkanen, E. Turquin, F. Sillion, and T. Aila. Meshless finite elements for hierarchical global illumination. Technical Report TML-B7, Helsinki Univ. of Technology, 2007.

[5] D. Shepard. A two-dimensional interpolation function for irregularly-spaced data. In *Proc. 23rd ACM National Conference*, pages 517–524, 1968.

[6] P. P. Sloan, J. Kautz, and J. Snyder. Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. *ACM Trans. on Comp. Graphics*, 21(3):527–536, 2002.

Using Metaprogrammed Functors to Implement Double-Dispatch for Collision Handling

Tiago H. C. Nobrega, Diego D. B. Carvalho, Aldo von Wangenheim
LAPIX - Laboratory for Image Processing and Computer Graphics
UFSC - Federal University of Santa Catarina, Brazil
{tigarmo,diegodbc,awangenh}@inf.ufsc.br

Abstract

We provide a strategy to the management of collisions between multiple objects with different types employing multiple-dispatch and both Object-Oriented and Generic Programming concepts. The solution scales well to the number of object types, with a fixed, constant-time cost to arrive at the proper interference-detection routine. Additionally, it helps the application programmer by allowing them to implement only the functionality required by their own program, removing the need for common solutions such as Abstract Base Classes full of pure virtual methods.

1. Introduction

The design of a collision detection engine or framework is a difficult task riddled with decisions and trade-offs on important design goals such as efficiency, modularity, size, dependencies, and others. The C++ language is a mature language with support for both Object-Oriented (which provides runtime polimorphism and aims at easing the development and maintenance of large systems, such as 3D engines) and Generic (with templates, offering run-time speed benefits) programming. In this work we explore the combination of these methodologies to implement the collision detection mechanism between different entities. We aim to expose the benefits this combination provides to both the system designer and the application programmer.

2. Bounding Hierarchies with Double-Dispatch and Meta Programming

Bounding Hierarchies[2] are often employed by collision detection algorithms to speed up interference detection between models with complex geometry, and common choices for bounding objects include Spheres, Oriented Bounding Boxes (OBBs) and Axis-Aligned Bounding Boxes (AABBs). Classes are often designed so that most functionality is placed in a base class, here named BoundingTree, and then object-specific details are delegated to other classes that inherit from it.

Table 1. A matrix of functions.

	AABBTree	SphereTree	OBBTree
AABBTree	*f1*	–	*f2*
SphereTree	–	–	–
OBBTree	*f3*	–	*f4*

The problem arises when a scene is composed of many distinct kinds of entities and a manager object has to decide if two entities are colliding. The manager only has knowledge of BoundingTrees. The problem is such: given two instances of BoundingTree subclasses, how can we efficiently determine if they are intersecting?

Since the choice of algorithm to determine the intersection depends on the types of *both* entities, this problem can be solved with Multimethods, which are “the mechanism that dispatches a function call to different concrete functions depending on the dynamic types of multiple objects involved in the call” (Alexandrescu[1]). To cope with the fact that the C++ language has no built-in support for Multimethods Alexandrescu proposes a few alternatives, one of which is used here - a constant-time matrix of pointer to functions, with some adaptations.

The method consists of assigning a numerical value (an *id*) to each BoundingTree subclass, retrievable at runtime through a virtual function. Alexandrescu uses these ids to populate a matrix of function pointers, at setup time, and to dispatch a function based on the types of two objects.

There are several advantages to this approach: Because the matrices become the sole point of interaction, no class involved in the BoundingTree hierarchy needs to be aware of the others, reducing build times on changes and potentially diminishing coupling — this is a benefit over common implementations of double-dispatch like the Visitor design pattern[3]. Additionally, the cost to arrive at the correct

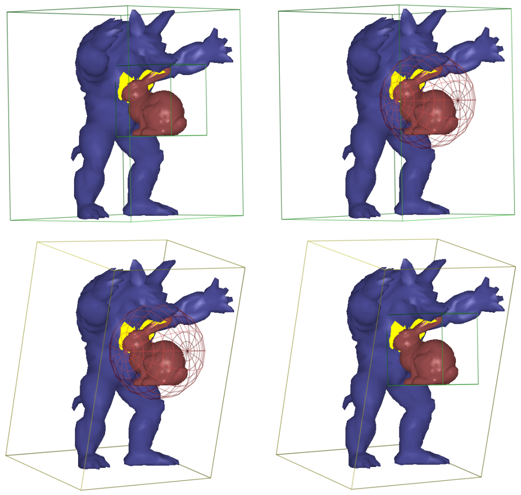


Figure 1. Employing different bounding hierarchies at runtime on the Stanford Armadillo and Bunny.

function is constant regardless of the number of subclasses and the matrix needs to be populated (and thus, functions implemented) only for the desired types, e.g., if the programmer knows their application won't need SphereTrees they can disregard their intersection tests with the other hierarchy types entirely (table 1). Since many collision detection algorithms are commutative, the number of implemented functions can be further reduced.

We extend the matrix to hold Functors (function objects) instead of raw function pointers. The added cost of one virtual call is the downside to the greater flexibility, as functions are poor at affecting the external world and managing state. The functors are small pieces of code written separately from the tree hierarchies and dedicated exclusively to handling the intersection between two types of bounding objects.

2.1. Functor Generation with Metaprogramming

It can be cumbersome to populate the matrices individually, cell-by-cell, because a matrix of n types requires n^2 functions, or $\binom{n}{2} + n$ if the algorithms are commutative. The matrices and the functors are actually class templates and the number of BoundingTree subclasses of interest is usually known at compilation time, so we developed a set of utility functions that work on *Typelists*[1] to create appropriate Functor classes and populate matrices through metaprogramming.

For instance, assume we're interested in managing Sphere, AABB and OBB trees. We have code to perform the intersection tests for all combinations between these three types of geometric objects, and we wrap all of them inside an utility class called OverlapDetection. The signature for all intersection functions is similar such that the general functor to perform the tests could look like:

```
template<class TreeA, class TreeB>
struct OverlapTester{
  bool operator()(const TreeA& a, const TreeB& b)
  {
    return OverlapDetection::overlaps(
      a.boundingObject(), b.boundingObject());
  }
};
```

Next, we implement a function to receive the list of *types* of trees we're interested in and perform all pairwise combinations of types from that list, at compilation time. Each pair fully defines an OverlapTester functor. If a particular pair doesn't follow the OverlapDetection convention, the OverlapTester can be trivially specialized. As long as this specialization is visible to the compiler it'll choose it when forming the specialized pair from the typelist.

3. Conclusions and Future Work

By our initial evaluations the time spent finding the appropriate method to dispatch takes up to 10% of the overall intersection process in the worst case. The scene in figure 1 reaches about 400 full intersection processings per second on an standard Athlon X2 3800+ with both models at around 3 thousand triangles. A next step is to compare this implementation with a more traditional approach and locate the source of the overhead.

We believe the idea of generic functor matrices can be expanded beyond the double-dispatch case. Any time a new kind of operation has to be performed on a BoundingTree a matrix can be employed instead of further extending the basic BoundingTree contract to support new dependencies. For example, employing bounding trees for Raytracing[4] would require line segment-bounding tree intersection tests. A matrix for such an application could have one row for each BoundingTree subclass, and the LineSegment class for its single column.

References

[1] A. Alexandrescu. *Modern C++ design: generic programming and design patterns applied.* Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2001.

[2] C. Ericson. *Real-Time Collision Detection (The Morgan Kaufmann Series in Interactive 3D Technology).* Morgan Kaufmann, December 2004.

[3] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design patterns: elements of reusable object-oriented software.* Addison-Wesley Professional, 1995.

[4] I. Wald, S. Boulos, and P. Shirley. Ray tracing deformable scenes using dynamic bounding volume hierarchies. *ACM Trans. Graph.*, 26(1):6, 2007.

A Geometric Bisection Algorithm for Automatic Page Layout

Alexandre Seki, Mauricio Klein, Ricardo Piccoli,
Isabel Manssour, João B. Oliveira
FACIN - Faculdade de Informática - PUCRS
{alexandre.seki, mauricio.klein, ricardo.piccoli}@cpph.pucrs.br,
{manssour, oliveira}@inf.pucrs.br

Abstract

Content distribution in a page designed for printing is a difficult task performed by human editors. There are applications and tools to help the editor in this task, but it is still a manual, iterative process that is made more difficult when content is distributed in columns, as is the case of newspapers. This work describes an algorithm to automatically distribute content in a printable page. Our implementation tries to use all the page, dividing it among the items to be placed according to their area.

1. Introduction

One of the problems in the design of magazines and newspapers is to distribute elements on a page. A designer is responsible for creating the page layout, distributing elements to achieve some degree of aesthetic quality. To reach a high-quality final result, that professional usually edits the layout several times, changing the news positions and format. There are tools to help the designer to format the page layout, but these tools do not generates the layout automatically and only support human intervention.

A typical bisection algorithm is an algorithm that repeatedly divides an interval in half and recursively processing both halves. Here, such an algorithm is used to distribute elements into a page, a task that will help many areas of digital publishing. Besides document designers, common users may also produce printable pages with a RSS feed content or pictures.

Jacobs *et al.* [3, 5] present an algorithm to generate layout automatically, using templates to guide the layout construction. Harrington *et al.* [4] propose a genetic algorithm to distribute elements on a page, but the computational cost of a genetic algorithm is very high for massive production. The distribution of photos on a page was researched by Atkins [1]. Another solution for this problem using genetic algorithms was presented by Geigel and Loui [2].

The main goal of this work is to describe and present results of an algorithm to automatically distribute elements into a page. This algorithm was developed as part of the APL (Automatic Page Layout) project. The current version is able to produce documents akin to newspapers covers. In our case, a user selects some news and pictures to be placed on the page and the algorithm distribute these elements. The remaining of the poster is organized as follows: the algorithm and the obtained results are described in Section 2; future work and conclusions are presented in Section 3.

2. The APL project

The main goal of the APL Project is to develop ways of distributing content automatically on a page. In our algorithm, the elements of a page are news, pictures and groups of such elements. News are composed by a title, the content text and optional attributes, such as author or the page number to the full news. Pictures are placeholders for an image or picture, and groups are used to aggregate them. For example, a group of two images and a piece of news will have its elements placed together, either side by side or one above the other, thus keeping elements organized.

The algorithm is based on a divide and conquer strategy: it receives the page width and height and a list of elements to be placed, and through consecutive bisections it divides the page area into regions, trying to allocate the required area of a news into this region. In our case, bisection is made by a horizontal or vertical line dividing a region into two new regions and, similarly, the list of elements are also divided. We can see an example of the defined regions for some news in Fig. 1a and 1b. In these figures, we have used the same news and only the number of columns was changed.

The algorithm tries to keep the elements in reading order (as given by the input sequence), from the top left to the bottom right. This strategy reduces the amount of possibilities to be examined and also reduces the surprise of the user, as news are not placed randomly. The first tests

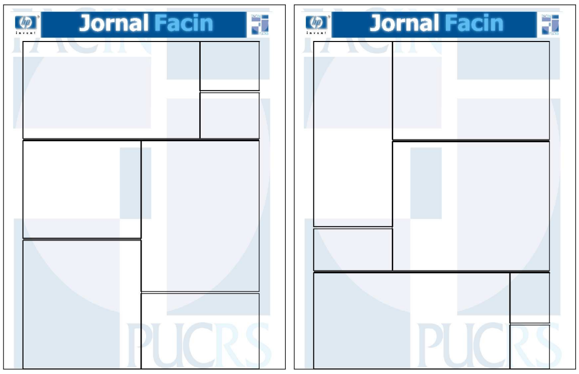


Figure 1. Specified regions in a page (a) with 4 and (b) with 6 columns.

were executed, automatically generating newspaper covers, the content were pictures and news. We have high-quality results in a short processing time. Fig. 2a shows an example of a generated newspaper cover that has as input four news and three pictures.

Our algorithm can be used to produce pages from RSS feeds. In this case, most of news are short and have a link to the full news or other related pages. Another peculiarity of RSS feeds is that usually there are no pictures. An example of a cover generated from RSS feeds is shown in Fig. 2b. Finally, as in Fig. 3, our algorithm can also place pictures, images or photos on a page.

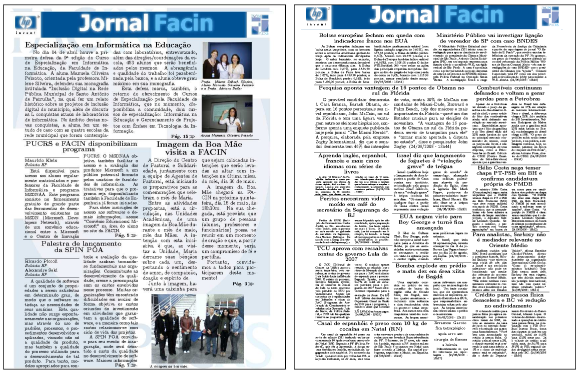


Figure 2. Examples of generated pages (a) from newspaper content and (b) from RSS feed.

The project is being developed using Java as the main programming language. The input file to the algorithm is an extension of XML obtained by converting RSS feeds and/or Wiki formats into our XML extension. The program can generate different outputs, such as PDF, LaTeX and dvi.

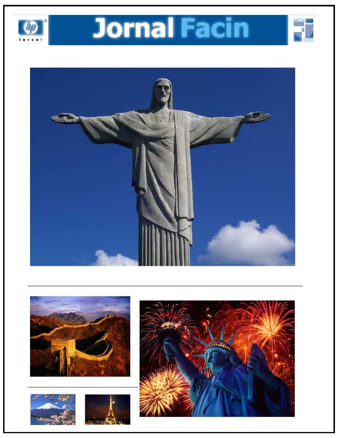


Figure 3. Generated page with pictures

3. Conclusions and Future Work

This poster presents a new algorithm to distribute content automatically in a printable page divided into columns, preserving reading order and taking into account the area of each news. Future works include finishing the prototype to make it available for tests and evaluation; extending the page layout from one to several pages and the development of a user interface for simpler use.

Acknowledgements

This work was (partially) developed in collaboration with HP Brazil R&D.

References

[1] C. B. Atkins. Adaptive photo collection page layout. In *ICIP*, pages 2897–2900, 2004.

[2] J. Geigel and A. C. P. Loui. Automatic page layout using genetic algorithms for electronic albuming. volume 4311, pages 79–90. SPIE, 2000.

[3] C. Jacobs, W. Li, E. Schrier, D. Bargeron, and D. Salesin. Adaptive grid-based document layout. In *SIGGRAPH '03: ACM SIGGRAPH 2003 Papers*, pages 838–847, New York, NY, USA, 2003. ACM.

[4] L. Purvis, S. Harrington, B. O'Sullivan, and E. C. Freuder. Creating personalized documents: an optimization approach. In *DocEng '03: Proceedings of the 2003 ACM symposium on Document engineering*, pages 68–77, New York, NY, USA, 2003. ACM.

[5] E. Schrier, M. Dontcheva, C. Jacobs, G. Wade, and D. Salesin. Adaptive layout for dynamically aggregated documents. In *IUI '08: Proceedings of the 13th international conference on Intelligent user interfaces*, pages 99–108, New York, NY, USA, 2008. ACM.

Enabling Interactive Brain Fiber Tracking with the GPU

Adiel Mittmann, Tiago H. C. Nobrega, Eros Comunello,
Diego D. B. Carvalho, Aldo von Wangenheim
LAPIX – Laboratory for Image Processing and Computer Graphics
UFSC – Federal University of Santa Catarina, Brazil
{adiel,tigarmo,eros,diegodbc,awangenh}@inf.ufsc.br

Abstract

Brain fiber tracking allows a neurosurgeon to find fiber bundles in any region of the human brain, rendering it a useful aid to preoperative planning. However, this process requires a significant amount of computation, which limits the interactivity of current applications. We show how executing this process on graphics processing units can significantly improve the user's experience by computing the fiber tracts in real time.

1. Introduction

Brain fiber tracking [2] is a computational procedure which finds the fiber bundles in a patient's brain, thus allowing a neurosurgeon to make a better preoperative planning. Implementations of fiber tracking have to cope with a great amount of mathematical operations, and for that reason they are typically slow at finding the fiber tracts, resulting in a poor interactive experience.

We have previously shown how fiber tracking can be run on the graphics processing units (GPUs) of computers [1] using the Cg language. The speed obtained by executing fiber tracking on GPUs has enabled us to consider an approach to fiber tracking that was not previously possible: find new fiber tracts whenever the volume of interest is dragged by the user or when fiber tracking parameters are changed. While the approach itself does not imply the use of GPUs, they provide the computational power required for a reasonable frames per second rate. This differs from previous work by the way fiber tracking is experienced by the user, who previously had access to a static environment, where the dragging of volumes of interest did not cause the fiber tracts to be updated in real time.

In this paper, we show how this higher level of interactivity can be achieved by executing fiber tracking on GPUs by means of the CUDA (Compute Unified Device Architecture) technology [3]. We compare the interactivity obtained by the execution of fiber tracking on the GPU and on the CPU (central processing unit) by measuring the frames per second (FPS) rate obtained with both implementations.

2. Fiber Tracking with CUDA

CUDA is a set of technologies recently introduced by NVIDIA with the aim of making it easy for developers to write GPU programs. It provides a C compiler with a few extensions and a relatively simple API (application programming interface) for interacting with the GPU. This is a progress over previous models, which had domain-specific languages and typically required the use of complex combination of APIs.

The process of fiber tracking, which in our approach is executed on the GPU, works by looking for fibers at each of a set of seed points, typically a few thousand. Each fiber consists of a set of line segments, and each of these points is calculated by the fourth-order Runge–Kutta method. Despite the great amount of computation involved in finding all points for each fiber, they can be determined independently from each other. Thus, the fact that fiber tracking entails a big amount of complex calculations and high level of data parallelism makes a GPU implementation very attractive.

All fibers are located by the GPU, but they are displayed on the screen by our radiological workstation running on the CPU, that is, the CPU is responsible for the actual visualization of data, whereas the GPU generates raw results. Ultimately, however, the CPU will delegate at least part of the rendering process to the GPU, though in an indirect manner, i.e., with no direct access through CUDA.

The workstation employs a two-phase fiber rendering strategy. While the user is dragging the volume of interest, fibers are shown as simple lines. When the user releases the mouse button, all fibers are immediately rendered as cylinders, which provides for a better visualization of the fibers.

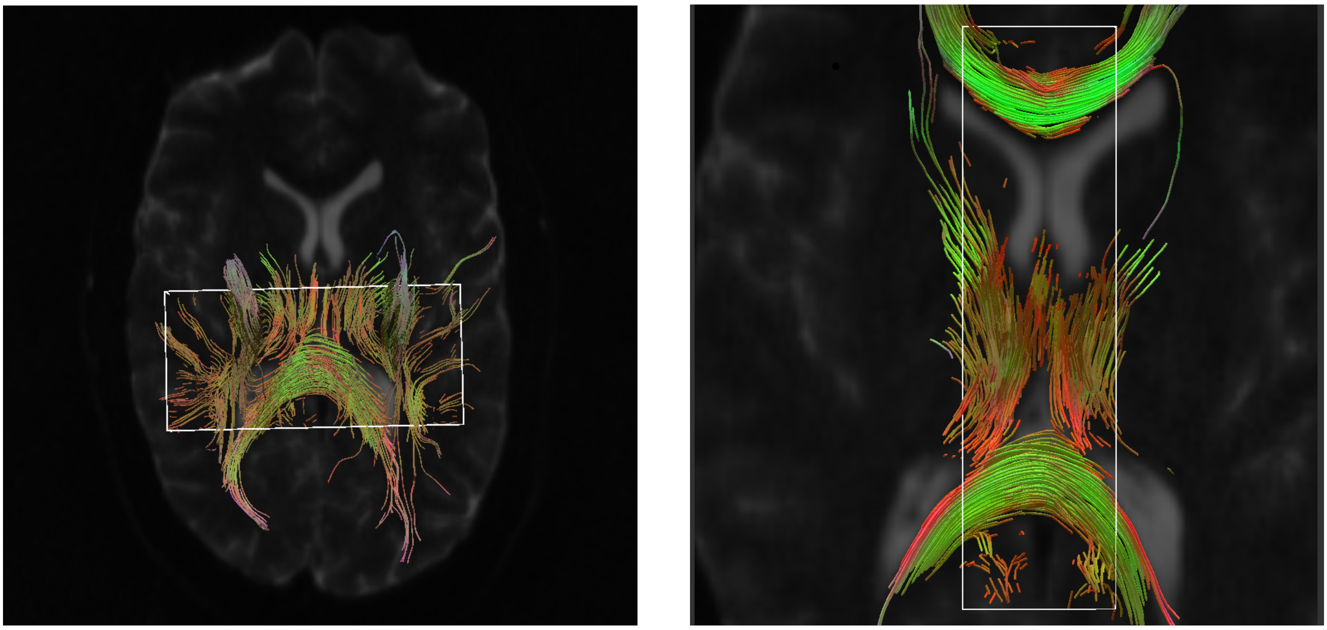


Figure 1. Results of fiber tracking being explored. The white lines define the limits of a volume of interest from which fiber tracts are determined.

3. Results

Although our current CUDA module is not as fast as our previous Cg implementation, it shows promising results. Figure 1 shows two screen shots of a fiber tracking exploration session. To better assess the level of interactivity achieved by our approach, please visit `http://www.lapix.ufsc.br/videos/`, which contains an uncut video showcasing the features of our approach. All experiments were run on a desktop computer, with an Athlon 64 3500+ processor, 2 GB of RAM and a GeForce 8800 GTS video card.

In order to estimate how fast our fiber tracking application is, the following method was used: A cube of dimensions 10 mm × 10 mm × 10 mm is centered at the bottom of the volume and then swept to its top in a total of 100 steps, as though the user were dragging the cube. The number of seeds within the volume is $20 \times 20 \times 5 = 2000$. After each move, fiber tracking takes place and the new trajectories are displayed, thus effectively rendering a new frame on the screen. At the end of the process, the mean and minimum rate of frames per second (FPS) can be found.

Table 1 shows results of this experiment for the GPU and the CPU. The GPU was more than 10 times faster than the CPU on average, and more than 16 times faster on the worst case. Qualitatively, this means the GPU gives the user a better level of interactivity, which we expect to be further improved as GPUs continue to outperform CPUs.

Table 1. Mean and minimum frames per second obtained on both the GPU and the CPU.

	Mean FPS	Minimum FPS
CPU	0.99	0.43
GPU	10.56	7.03

4. Conclusion and Future Work

GPUs enable fiber tracking application to take on a new level of interactivity. The user's experience shifts from a limited and slow process to a real-time, highly interactive exploration of a patient's brain. We expect to further improve the results obtained so far and to find new ways to take advantage of the GPU's processing power.

References

[1] A. Mittmann, E. Comunello, and A. von Wangenheim. Diffusion tensor fiber tracking on graphics processing units. *Comput Med Imaging Graph*, 2008. DOI: 10.1016/j.compmedimag.2008.05.006.

[2] S. Mori and P. C. M. van Zijl. Fiber tracking: principles and strategies - a technical review. *NMR Biomed*, 15(7-8):468–480, 2002.

[3] J. Owens, M. Houston, D. Luebke, S. Green, J. Stone, and J. Phillips. Gpu computing. *IEEE J Proc*, 96(5):879–899, 2008.

Physically Based Simulation Using Particle Systems

Yalmar Ponce Atencio   Claudio Esperança
Federal University of Rio de Janeiro
Computer Graphics Laboratory
{yalmar, esperanc}@lcg.ufrj.br

Abstract

This work consists of extending the rigid body simulation method presented by Harada [3]. Firstly, the Newtonian formulation is replaced by impulse based physics [2]. Secondly, deformable body simulation is achieved by using two different approaches: non-zero volume objects are handled by an adapted meshless shape matching technique[5], whereas objects such as cloths and ropes are simulated using Jakobsen's approach [4].

1. Introduction

Physically based simulation of rigid and deformable bodies has been intensively researched during the last decade. The idea is to obtain a fast and realistic, although not (in general) physically correct, animation. Its main area of application is in entertainment, such as animation films or computer games.

In this work we propose a unified physically based simulation technique, which supports many kinds of objects: rigid, deformable with volume, cloths and ropes. Since all objects are represented by particle systems, they share the same collision detection engine. Further, collision response is computed by integrating collided particle reaction forces for inferred displacements.

2. Collision detection

As described in [3], objects are represented by particle sets, where each particle is a sphere with small diameter. Thus, the collision detection process essentially detects collisions between small spheres. Complex objects can be handled by employing a voxelization method in order to place particles onto the object boundary (see the figure 1).

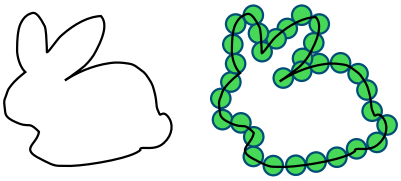


Figure 1. *Discretizing the boundary of a complex object.*

3. Collision response

Once all colliding particles are determined, Harada estimates the force F_B and torque τ_B that will be applied to a given body B by adding up all contributions computed for its associated particles, i.e.,

$$F_B = \sum_{i \in B} f_i \quad \tau_B = \sum_{i \in B} r_i \times f_i,$$

where r_i and f_i are the relative position and the force of particle i respectively. In contrast, we employ a slightly different formulation, using impulses for collided particles associated with rigid bodies. Applying an impulse at a collided point (the midpoint between two collided particles) changes the linear and angular velocities of the body according to

$$\begin{aligned} v^{'} &= v + J/m \\ \omega^{'} &= \omega + I^{-1}(r \times J). \end{aligned}$$

For a particle associated with a deformable object, rather than submit it to a force or impulse, the particle is merely pushed to a non-colliding state.

4. Integration

Once the new states of colliding particles are computed, a new state for the object to which they belong must be estimated. This is done by integrating all particle contributions. For rigid bodies, the integration follows the approach of Guendelman et al. [2]. In the case of deformable objects with volume, we use the meshless shape matching technique presented by Mueller et al. [5], which uses a semi-implicit integration method (Figure 2 illustrates collision response for both rigid and deformable bodies). Cloths and ropes are simulated using the Jakobsen method [4], which uses a Verlet integrator supported by a constraint relaxation scheme.

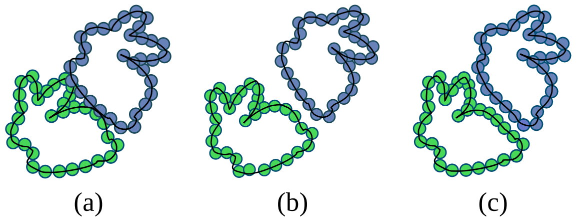


Figure 2. *A collided state (a) and the state after collision response for rigid (b) and deformable bodies (c).*

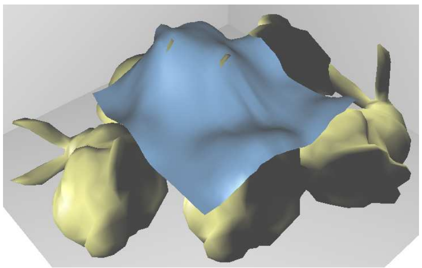


Figure 4. *A Cloth falling on top of bunnies.*

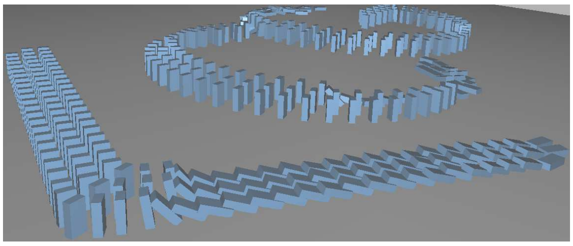


Figure 5. *Simulating a domino configuration.*

5. Preliminary results

Our prototype is implemented using C++ and OpenGL/GLSL. Some results are shown in the figures 3, 4 and 5. In the examples, we use deformable cubes with 386 particles, bunnies with 402 particles, cloths with 625 particles and each domino piece has 15 particles. All the examples run at interactive rates.

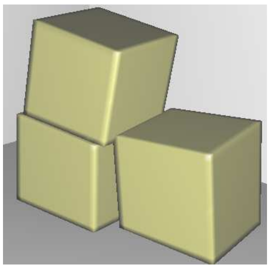


Figure 3. *Three deformable cubes in contact.*

6. Conclusions and future work

As shown by Harada, the method can be executed entirely in GPU by storing particle data in textures. Since modern GPUs support 4096x4096 textures, thousands of objects can be simulated. Since our current prototype is still CPU programmed, our next step is to reimplement it in GPU using the CUDA [6] technology..

The particle-based method works well for small time steps. On the other hand, fast moving particles or large time steps may lead to an ill-conditioned simulation. We are currently at work trying to improve the collision detection process in order to support larger time steps and objects with different particle sizes by adapting the technique presented by Le Grand [1].

Notice that although objects are represented by coarsely geometry for quick collision detection, rendering can be performed using more detailed representations. Our prototype, at this time, does not support interaction with fluids yet, but the extension is natural.

References

[1] S. L. Grand. *GPU Gems 3*, chapter Broad-Phase Collision Detection with CUDA, pages 697–721. Addison Wesley, 2007.

[2] E. Guendelman, R. Bridson, and R. Fedkiw. Nonconvex rigid bodies with stacking. In *SIGGRAPH '03: ACM SIGGRAPH 2003 Papers*, pages 871–878, New York, NY, USA, 2003. ACM.

[3] T. Harada. *GPU Gems 3*, chapter Rigid Bodies on GPU, pages 611–632. Addison Wesley, 2007.

[4] T. Jakobsen. Advanced character physics. In *Proceedings, Game Developer's Conference 2001*, SJ, USA, 2001. GDC Press.

[5] M. Muller, 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.

[6] NVIDIA™. CUDA Environment – Compute Unified Device Architecture, 2007. http://www.nvidia.com/object/cuda_home.html.

3D Texture Painting of Point Models

José Ricardo Mello Viana, Ricardo Marroquim, Cláudio Esperança
Universidade Federal do Rio de Janeiro - PESC/COPPE

Abstract

We propose a technique for painting 3D models based on a rendered image of the model and a 2D map of normal vectors. Texture generation and coordinate mapping is then produced on-the-fly, making it possible to paint in broad strokes or in small detail. The minimalistic requirements of this technique make it suitable for painting both regular meshes and point-based models.

1. Introduction

Most 3D painting systems follow the so-called *UV-mapping* paradigm. Thus, they require an initial step where an "unwrapping" map of the model surface is established. This is then followed by mapping the user's painting strokes on the obtained flattened canvas. Such a system is heavily dependent on the quality and resolution of the unwrapping map which ideally should contain as little angle distortion as possible [2]. The map should also minimize the extent of the texture seams which will be responsible for discontinuities on the map. Another problem is the fact that the method is not concerned with the areas of the model which will actually be painted: the texture resolution is fixed once the unwrapping map is established, making it hard to paint small detail in some parts of the model without making the texture bigger as a whole.

In this work we follow the ideas of the Chameleon system [1], where textures and maps are created dynamically, and only for affected parts of the model. Given a view of the model, the artist paints it as though it were a regular 2D canvas. Only after the view is changed – rotating or moving the model, for instance – is the painting transformed into a texture and a corresponding map.

2. Painting Interface

The painting interface is similar to that of other 3D painting systems. A 3D model is loaded and rendered, after which the user chooses a color for painting. The painting strokes are realized with the aid of a 3D cursor in the shape of a regular polygon with size and number of sides specified by the user. The location of the 3D cursor is specified with the mouse, but its projection on the screen follows the curvature of the model. By clicking and dragging the mouse over the model, the parts of the model traveled by the 3D cursor have their color changed. These paint strokes are in fact drawn on a temporary buffer which will later be used to build a proper texture.

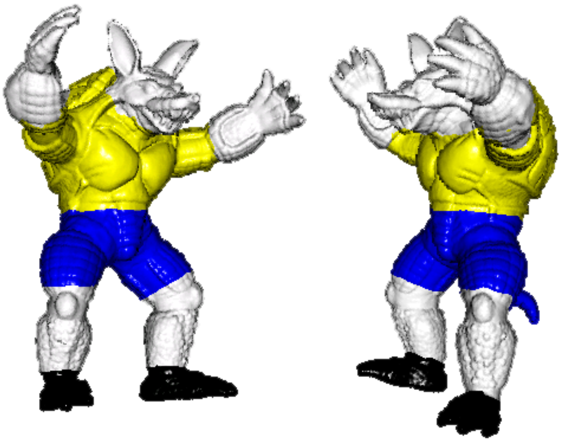


Figure 1. *Views of a model painted with our system.*

Despite the fact that no texture is generated at this stage, the user is shown a proper visual feedback by recomputing the illumination function of the screen pixels modified by the painting (see Figure 1). In order to compute the 3D cursor projection and the illumination function, a *Normal Buffer* is used, that is, a buffer which associates a normal vector with each color buffer pixel. The Normal Buffer also makes it possible to distinguish background pixels from pixels covered by the model (see Figure 2).

3. Texture Generation

Once the model view is zoomed, rotated or translated, a new texture is created containing all paint strokes input by the user since the previous view change. This corresponds to (1) a texture image and (2) a set of texture coordinates which will be assigned to the affected model points.

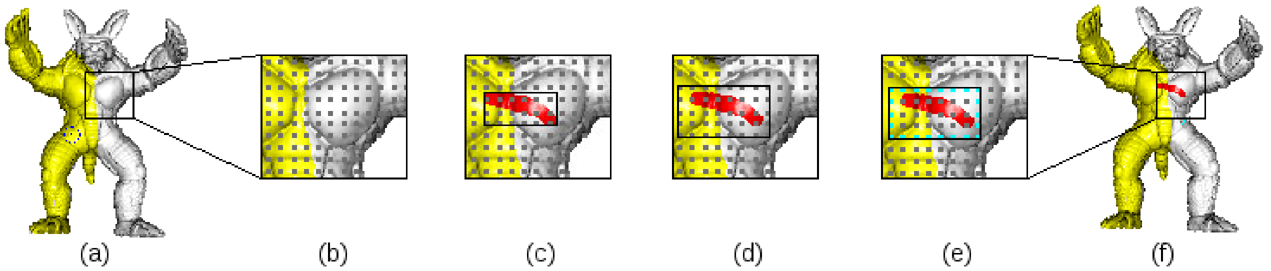


Figure 3. *Drawing texture passes: (a) input model for paint stroke, (b) points of affected area, (c) bounding box of new stroke, (d) expanded bounding box containing all points related with the stroke, (e) critical points (green) possibly associated with more than one texture, (f) model with new stroke.*

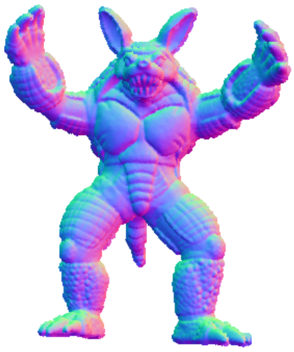


Figure 2. *Normal Buffer.*

The texture image should contain not only the newly painted strokes, but also the background pixels in the neighborhood of the painted areas. In fact, the texture image corresponds to all color buffer pixels inside a bounding box B large enough to contain all strokes, plus a safety margin. This safety margin is the maximum edge length in the case of a mesh model, or the maximum distance between neighbor samples in the case of a point-based model. For efficiency considerations, texture images are not handled individually, but rather copied as patches in a large texture atlas. This, however, makes it necessary to adopt some sort of management of the atlas space such as the one proposed by Igarashi [1].

For a point of the model to be assigned a new set of texture coordinates, it must satisfy the following conditions: (a) it must be visible in the current view, and (b) it must fall inside bounding box B. The selection of these points is performed with the aid of an *Id Buffer*, which maps screen pixels to model points. Notice, however, that a given point may participate in more than one texture image. This may occur for vertices close to the border of B. As a consequence, drawing the textured model may require several passes (see Figure 3).

4. Point-based rendering

As explained above, the system uses two auxiliary buffers (Normal and Id) which must be supplied along with a rendered view of the model. Although these could be easily obtained for regular mesh models, our proof-of-concept prototype implements texture painting on point-based models. This is accomplished with the help of the point-based renderization technique described in [3].

5. Conclusions

Although model rendering is mostly implemented in GPU, the texture painting algorithms are still processed entirely in CPU. We are currently at work implementing a better-performing prototype where texture painting is mostly implemented in GLSL, like [4].

An important topic not addressed by Igarashi is that of texture patch reuse. We are experimenting with an algorithm for counting texture patches in which each point participates and eliminating unused patches.

References

[1] T. Igarashi and D. Cosgrove. Adaptive unwrapping for interactive texture painting. In *I3D '01: Proceedings of the 2001 symposium on Interactive 3D graphics*, pages 209–216, New York, NY, USA, 2001. ACM.

[2] B. Lévy, S. Petitjean, N. Ray, and J. Maillot. Least squares conformal maps for automatic texture atlas generation. *ACM Trans. Graph.*, 21(3):362–371, 2002.

[3] R. Marroquim, M. Kraus, and P. R. Cavalcanti. Efficient point-based rendering using image reconstruction. In *Symposium on Point-Based Graphics 2007, Prague-Czech Republic*, September 2007.

[4] T. Ritschel, M. Botsch, and S. Müller. Multiresolution gpu mesh painting. In *Eurographics 2006 Short Papers*, pages 17–20, 2006.

Level of Detail for Point Model Rendering

Felipe Carvalho   Antonio Oliveira   Ricardo Marroquim
Universidade Federal do Rio de Janeiro - UFRJ/COPPE
{fmc,oliveira,ricardo}@lcg.ufrj.br

Abstract

Point-Based representation became a popular alternative to polygonal meshes for representing 3D geometric models. 3D photography and scanning systems acquire the geometry and appearance of real-world objects as point samples. In this work we present a method for efficiently creating a hierarchical multiresolution structure for point models. A variant of the octree is used to partition the space, while the merging of samples is driven by two error metrics.

1. Introduction

Point Based representation has been proposed as an alternative to polygonal meshes (usually triangular) for 3D surfaces, offering a number of benefits [2]. One of the main advantage of discrete point primitives over polygonal meshes is the lack of topological information; only a set of points, with additional attributes such as normal and color, have to be stored and processed. This allows for a simple and compact representation, ideal for efficient rendering and editing.

Today's technology for 3D surface acquisition has reached sampling densities that makes interactive visualization of the acquired data sets a difficult task[5]. With the increase of scene complexity, projected triangles become smaller than a single pixel. In this case, triangle based scanline rendering wastes time in superfluous sub-pixel computation. Level of detail (LOD) methods [3] can remove such tiny geometric details dynamically, but at the expense of a large CPU load for on-the-fly tree traversal and retriangulation. However, the enormous processing power of modern graphics hardware is underused.

2. Space-Partitioning Hierarchy

The multiresolution point representation used is a *point-octree*, which partitions the space adaptively according to the sample distribution (data driven), rather than regularly in space (space-driven), such as *region-octrees* [4].

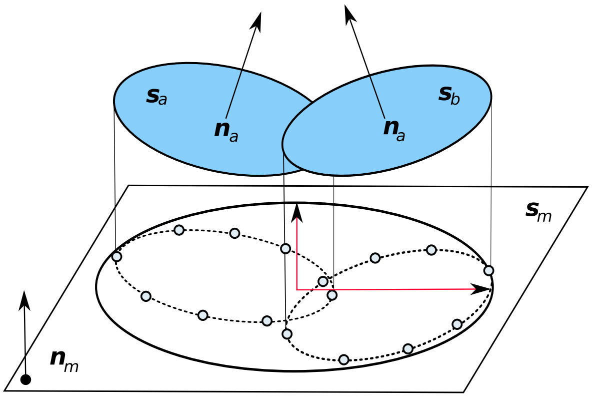


Figure 1. Merging splats

Each leaf node L_n of the hierarchy H, contains a set of k splats $s_c = \{s_1...s_k\}$, while each internal node I_n has a representative elliptical splat covering the extent of its children. Elliptical splats were chosen over circular splats, since the same of amount of surface can be covered with less samples [2]. Furthermore, for efficient back face culling, each $I_n \in H$ also includes a normal-cone with semi-angle θ bounding all samples in s_c.

3. Merging splats

Given a set of splats where each splat has center c_i and normal n_i, the new merged splat s_m has center and normal defined as [6]:

$$c_m = \frac{\sum_i |s_i| \cdot c_i}{\sum_i |s_i|} \quad (1)$$

$$n_m = \frac{\sum_i |s_i| \cdot n_i}{\sum_i |s_i|}, \quad (2)$$

where $|s_i|$ stands for the area of splat s_i. To compute the extent of s_m, we sample n (8 is enough) points p_j^i on the boundary of each splat s_i and project them to the splat plane of s_m. Then we apply principal component analysis (PCA) to the set of points p_j^i to find the main axis directions and proper scaling.

4. Error Measures

A good error measure should be able to distinguish smooth regions, that can be well handled by a single elliptical disk, from detailed or boundary regions. To this end, we use two error measures: a *perpendicular error* [1] and a *tangential error* . The *perpendicular error* e_p is the minimum distance between two planes orthogonal to the node's normal, which define a region enclosing all children in the node's descendent leaves. The computation of e_p is illustrated in Figure 2, and can be expressed as:

$$\begin{aligned} e_p &= \max\{a_i + d_i\} - \min\{a_i - d_i\} \qquad (3) \\ \text{with} \quad & d_i = r_i\sqrt{1 - (n_i \cdot n)^2} \\ & a_i = (c_i - p) \cdot n, \end{aligned}$$

where r_i is half the length of the major axis of s_i.

For the *tangential error*, s_m and its respective children are rendered offscreen in a small viewport. s_m is first rendered in blue, while its children are all rendered in red. The *tangential error* e_t (Figure 3) is proportional to the ratio between the area in red and the total painted area. Specifically, it is given by:

$$e_t = r_m\sqrt{\frac{n_{blue}}{n_{blue} + n_{red}}}, \qquad (4)$$

where n_{blue} and n_{red}, represent the number of pixels with the respective colors. If none of these two errors exceeds a predefined threshold ε, the s_m is rendered following the approach of [1]. Otherwise the tree of its node decendants must be explored .

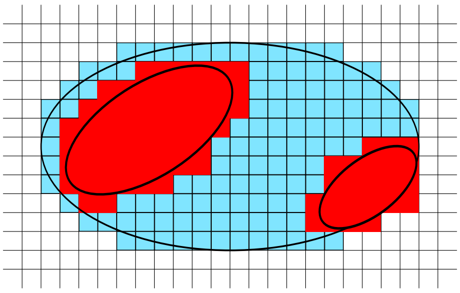


Figure 3. Tangential error

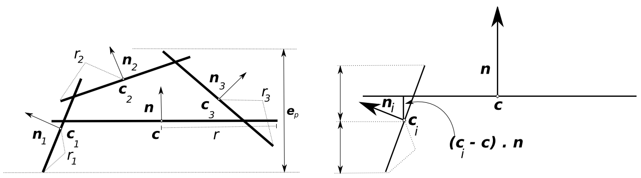


Figure 2. Perpendicular error

5 Conclusion and Future Work

We present a multiresolution hierarchy for point samples based on a space partitioning data structure, i.e., the point-octree. Elliptical splats are created in each internal node by analyzing the local surface properties. The two error metrics are used to define which splats should be projected for a given viewpoint, taking into account distance to the eye as well as surface smoothness, i.e, finer resolution splats along the silhouettes and detailed regions.

As future work, a more sophisticated partitioning refinement shall be investigated, such as curvature driven methods. This will allow for a more coherent clustering based on surface properties, instead of spatial distribution. Furthermore, the hierarchical data structure should only be used to create the coarser resolution splats, while an array-like structure that maps well to the GPU, should be used for efficient rendering. In addition, the perpendicular error can be improved for elliptical splats, since it was derived for circular splats [1, 4, 5]. In this case, the error is overestimated using the larger ellipse's axis.

References

[1] C. Dachsbacher, C. Vogelgsang, and M. Stamminger. Sequential point trees. In *SIGGRAPH '03: ACM SIGGRAPH 2003 Papers*, pages 657–662, New York, NY, USA, 2003. ACM.

[2] L. Kobbelt and M. Botsch. A survey of point-based techniques in computer graphics. *Computers Graphics*, 28(6):801–814, Dec. 2004.

[3] D. Luebke, B. Watson, J. D. Cohen, M. Reddy, and A. Varshney. *Level of Detail for 3D Graphics*. Elsevier Science Inc., New York, NY, USA, 2002.

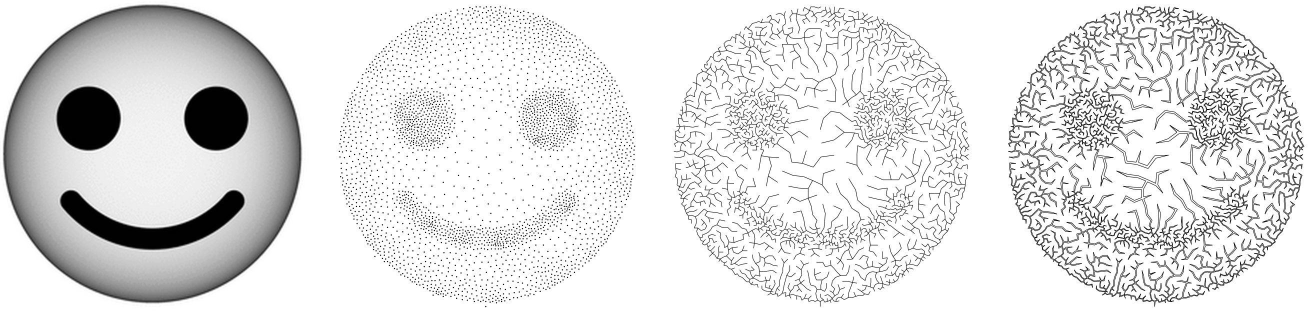
[4] R. Pajarola. Efficient level-of-details for point based rendering. In *Proceedings IASTED Invernational Conference on Computer Graphics and Imaging*, Calgary, AB, Canada,, 2003. ACTA Press.

[5] S. Rusinkiewicz and M. Levoy. Qsplat: a multiresolution point rendering system for large meshes. In *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 343–352, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.

[6] J. Wu, Z. Zhang, and L. Kobbelt. Progressive splatting. *Symposium on Point-Based Graphics*, 0:25–142, 2005.

Fracture Patterns for Non-Photorealistic Image Rendering

Anna Regina Corbo   Luiz Henrique de Figueiredo
IMPA – Instituto Nacional de Matemática Pura e Aplicada, Rio de Janeiro, Brazil
{corbo,lhf}@impa.br



Abstract

We describe a method for non-photorealistic, "artistic" rendering of images that uses lines to depict fracture (or crack) patterns based on image features. The method starts by sampling the image with a set of points that is good for stippling. These points are then used as vertices of a minimum spanning tree which guides the crack propagation in image space.

Surface crack patterns occur on a variety of materials, including glass, mud, and ceramic glaze. The generation of realistic crack patterns on the surface of 3D objects can be done with physically based methods [3, 4]. In this work, we follow a non-photorealistic approach for artistic rendering of images with crack patterns that are generated based on image features.

Several recent papers have presented methods for reproducing drawing art techniques (e.g., [5]). We considered the reproduction of cracks as an artistic line position problem. This approach is used by Wyvill et at. [6] and Mould [2]; in their individual way, these authors interpreted fracture patterns as lines or edges of some graph.

Our work is closely inspired by the work of Kaplan and Bosch [1]. They developed a technique of non-photorealistic image rendering using image point samples and a continuous line between them: the solution of the Traveling Salesman Problem on the set of point samples.

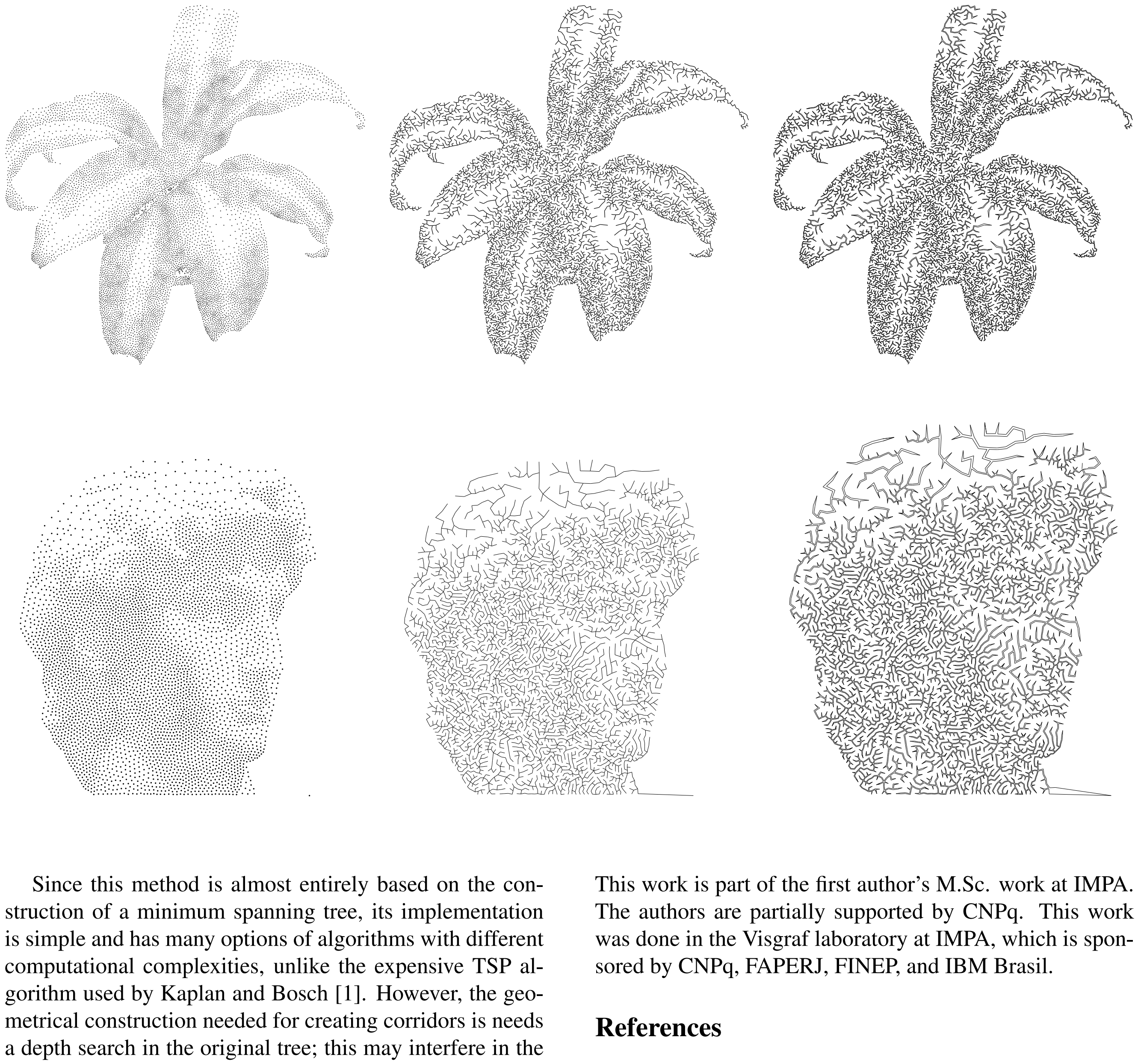
Here, we extended this method for drawing multiple lines based on some set of image point samples. We aimed to draw lines so that their distribution are similar to the design of a fracture. With this objective, our method connects, in a simple way, the sample points as vertices of a minimum spanning tree. This results in short edges for dark regions and long edges for lighter areas. Thus, the tree edges can represent the original gray levels by their concentration, length, or thickness shown in a given region.

This method has produced interesting results for images with relatively few gray levels. Not surprisingly, our tests have shown that having good initial sample set is very important for good visual results [5]. However, our main contribution is an unusual and artistic filter for digital image reproduction.

As shown in the banner picture above, the method has three steps: (1) generate a point sample of an image; (2) connect the sample points with minimum spanning tree; (3) create a crack pattern from the tree.

There are many methods for generating image points samples. We used a sampling method by Centroidal Vononoi Diagram such as the one by Secord [5] for the *stippling* problem. This method computes good samples in a short number of iterations.

The points are connected by an Euclidean minimum spanning tree, which can be rendered directly or by rendering tree edges as corridors of variable width. Smaller edges imply narrow corridors. This naturally controls the simulation of area gray levels, because the perceptible effect of near lines is the same effect of thick lines. In addition to variable width, we can also add small disturbances to the corridors.



Since this method is almost entirely based on the construction of a minimum spanning tree, its implementation is simple and has many options of algorithms with different computational complexities, unlike the expensive TSP algorithm used by Kaplan and Bosch [1]. However, the geometrical construction needed for creating corridors is needs a depth search in the original tree; this may interfere in the method's interactivity for large sample sets.

The figure above shows some examples of the artistic effects achieved with the method. More results and details on the method can be found at

`http://w3.impa.br/~corbo/diss/`

Our tests showed that good results can be obtained without any image information besides an initial good sample set of image's density. On the other hand, the less points we have, the more emphasized the fracture will be.

As a future work, we intend to create simulations to process the growth (of fractures or plants, for example) in two or three-dimensional objects, from arbitrary intensity maps. We also intend to extend this technique for realistic rendering of 3D objects.

This work is part of the first author's M.Sc. work at IMPA. The authors are partially supported by CNPq. This work was done in the Visgraf laboratory at IMPA, which is sponsored by CNPq, FAPERJ, FINEP, and IBM Brasil.

References

[1] C. S. Kaplan and R. Bosch. TSP Art. In *Proc. of Bridges 2005, Mathematical Connections in Art, Music and Science*, pages 301–308, 2005.

[2] D. Mould. Image-guided fracture. In *Proc. of Graphics Interface 2005*, pages 219–226.

[3] J. F. O'Brien, A. W. Bargteil, and J. K. Hodgins. Graphical modeling and animation of ductile fracture. In *Proc. of ACM SIGGRAPH 2002*, pages 291–294.

[4] J. F. O'Brien and J. K. Hodgins. Graphical modeling and animation of brittle fracture. In *Proc. of ACM SIGGRAPH 1999*, pages 137–146.

[5] A. Secord. Weighted Voronoi stippling. In *Proc. of NPAR '02*, pages 37–43.

[6] B. Wyvill, K. van Overveld, and S. Carpendale. Rendering cracks in Batik. In *Proc. NPAR '04*, pages 61–149.