

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

Laurindo S. Britto Neto*

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

Bruno M. Carvalho (Orientador)

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

Abstract

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

Resumo

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

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

1. Introdução

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

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

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

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

2. Trabalhos Relacionados

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

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

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

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

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

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

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

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

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

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

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

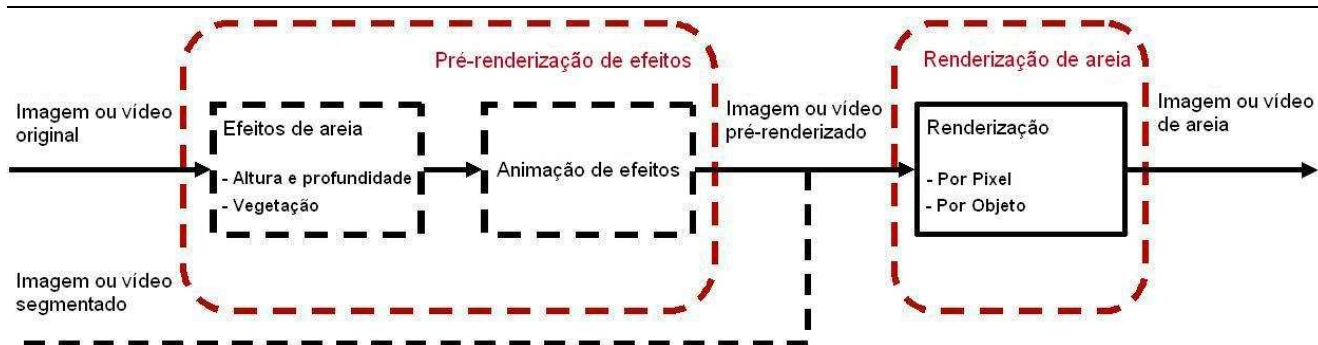


Figura 1. Pipeline de renderização Csand.

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

3. A Garrafa com Areia Colorida

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

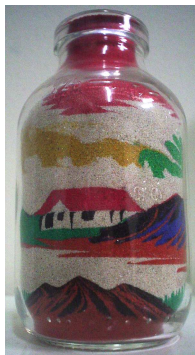


Figura 2. Garrafa com areia colorida.

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

4. Renderização Estilizada de Areia

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

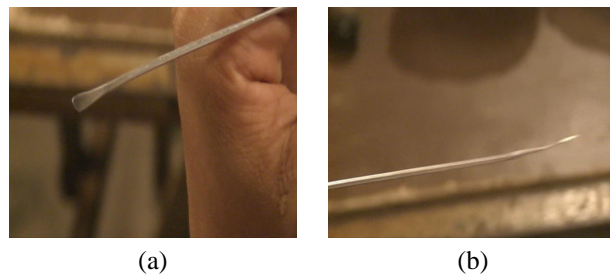


Figura 3. A ponta chata (a) e a ponta aguda (b) da ferramenta usada pelos artesãos.

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

4.1. Texturas Procedurais de Areia

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

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

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

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

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

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

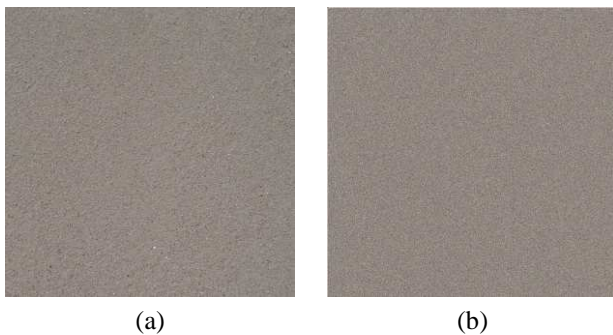


Figura 4. Fotografia real de areia em (a) e textura procedural de areia em (b).

4.2. Renderização da Areia

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

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

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

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

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

5. Simulando Efeitos de Pinceladas de Areia

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

5.1. Efeito de Altura e Profundidade

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

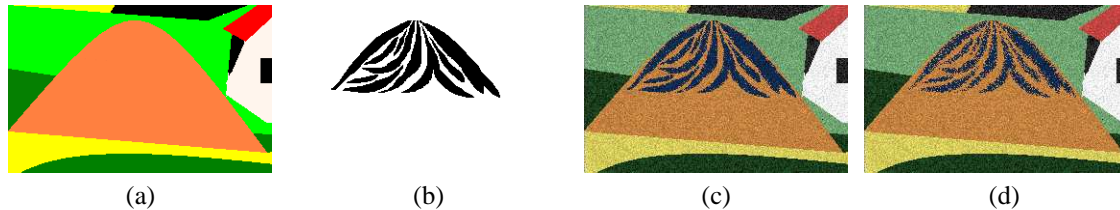


Figura 5. Simulação do efeito de altura e profundidade, onde são exibidos: a imagem original (a), máscara (b), efeito simulado com $\alpha = 20^\circ$ (c) e $\alpha = 40^\circ$ (d).

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



Figura 6. Efeito de altura e profundidade.

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

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

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

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

5.2. Efeito de Vegetação

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



Figura 7. Efeito de vegetação

Depois das máscaras geradas, é computado um mapa

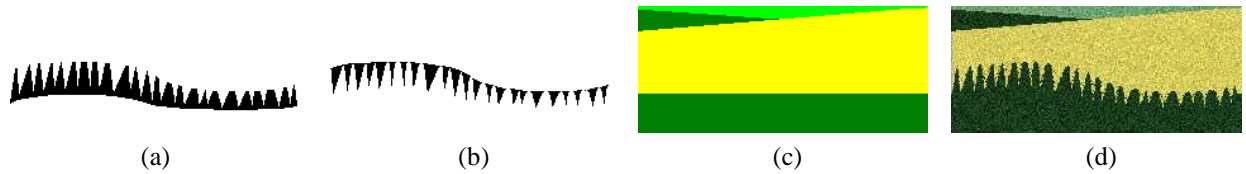


Figura 8. Máscaras original (a) e complementar (b) aplicados na imagem original (c) para gerar o efeito de vegetação visto em (d).

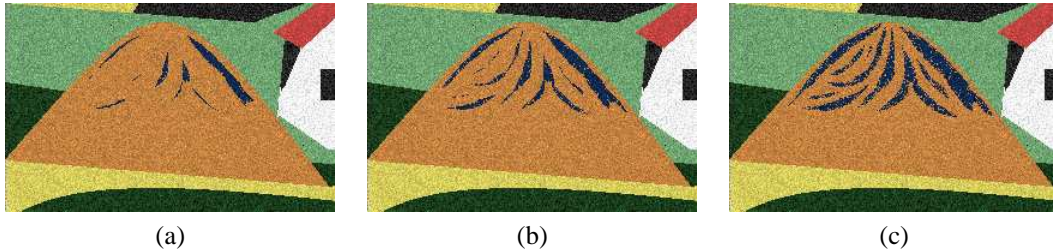


Figura 9. Três quadros da animação de um efeito.

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

5.3. Animando Efeitos de Pinceladas de Areia

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

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

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

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



Figura 10. Um quadro do vídeo de exemplo.

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

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

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

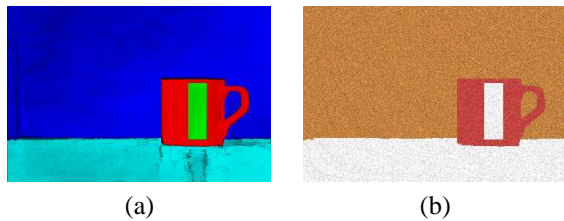


Figura 11. Quadro do vídeo segmentado (a) e quadro do vídeo estilizado com areia colorida (b).

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

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

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

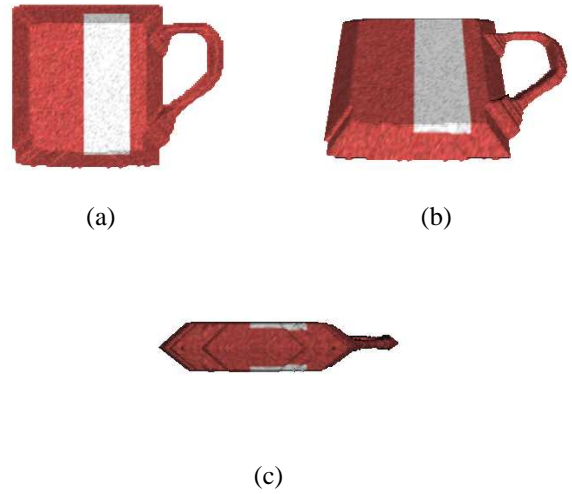


Figura 12. Forma 3D abstrata do objeto. Visão superior (a), visão em 45° (b) e visão de fundo (c).

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

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

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

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

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

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

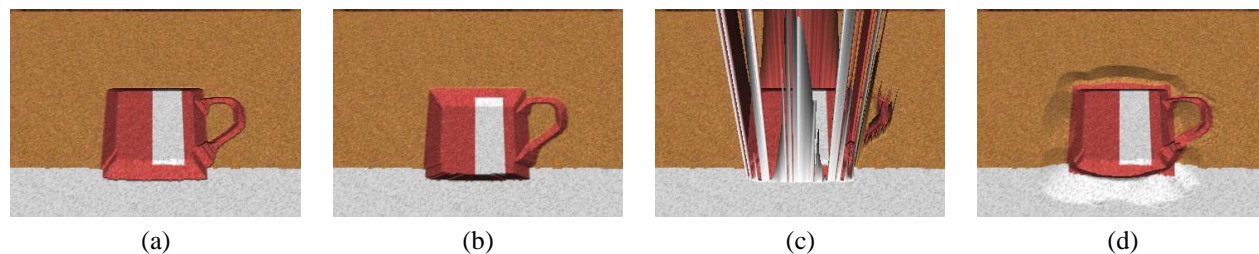


Figura 13. Modelagem 3D (a), compressão (b), deslocamento (c) e erosão (d).

7. Experimentos

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

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

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

8. Conclusão

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

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

Referências

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

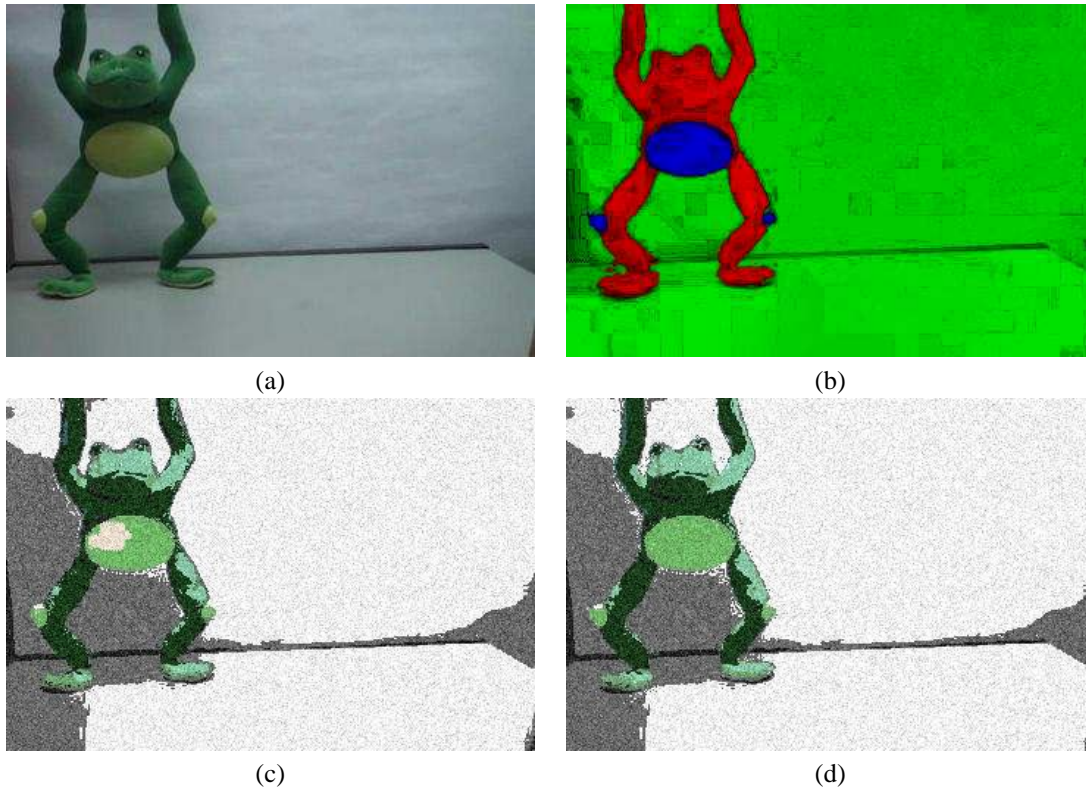


Figura 14. Um quadro do vídeo do sapo (a), o mapa de segmentação associado (b), a renderização por pixel do quadro (c), e o quadro onde o objeto azul de (b) foi renderizado usando o método de renderização por objeto (d).

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



(a)



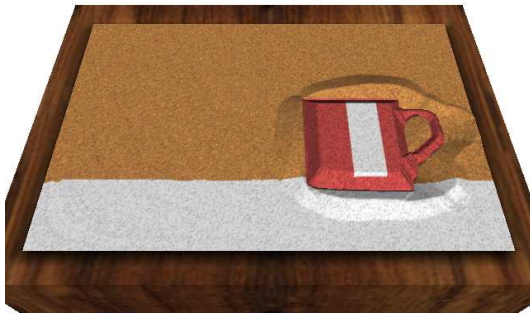
(b)



(c)



(d)



(e)



(f)



(g)



(h)

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