

Iluminação de Cartoons Baseada em uma Árvore de Regiões

Allan Carlos Rocha
Instituto de Computação-UFAL
allancarlosar@gmail.com

Renata Thomaz Lins
Instituto de Computação-UFAL
renatinhalins@gmail.com

Leandro Carnaúba
Instituto de Computação-UFAL
leandrocarnaubaleite@gmail.com

Adelailson Peixoto (orientador)
Instituto de Matemática-UFAL
adelailson@pos.mat.ufal.br

Vinícius Mello (orientador)
Instituto de Matemática-UFAL
vinicius.moreira.mello@gmail.com

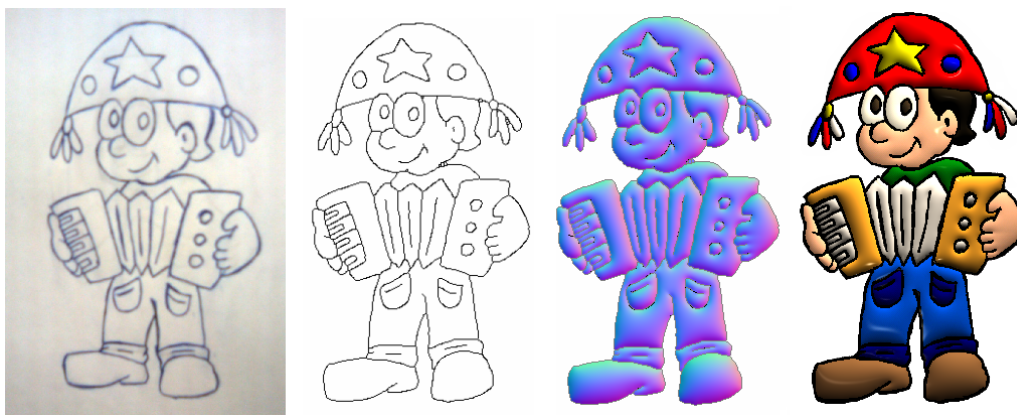


Figura 1. Da esquerda: Cartoon Original, Esqueletonização, Mapa de Normais e Cartoon Renderizado.

Resumo

Este trabalho apresenta uma técnica para iluminar cartoons 2D baseada em uma árvore de regiões que contém informações geométricas e topológicas dos objetos. Cada região é composta por um conjunto de pixels internos e um conjunto de curvas. Esta estrutura permite rapidamente a busca de relações de adjacência e inclusão entre regiões, o que facilita operações como colorização e iluminação durante a animação de cartoons. A partir desta estrutura, o trabalho apresenta um método de iluminação que permite o cálculo preciso das normais dos objetos.

1. Introdução

Três importantes etapas na produção de uma seqüência de cartoons são *animação*, *colorização* e *iluminação*. A primeira calcula os quadros intermediários da seqüência a partir do conjunto de quadros iniciais. A colorização transfere as cores de um quadro para os quadros subsequentes. A iluminação calcula a interação do cartoon com as luzes presentes em um ambiente 3D. Este trabalho apresenta um método de iluminação calculado a partir de uma *árvore de regiões* que representa o cartoon. Esta árvore possibilita acesso rápido às relações de adjacência e de inclusão entre regiões, o que permite o cálculo da interpolação das normais no interior de cada região durante a etapa de iluminação.

Alguns trabalhos empregam estruturas hierárquicas que permitem agrupar regiões em diferentes escalas e obter representações da imagem em diversas resoluções [2, 4, 3],

ideais para aplicações como segmentação e compactação de imagens e vídeos. Como o alvo do método aqui apresentado são imagens cartoon, a estrutura empregada para representar as regiões pode ser simplificada, uma vez que o cálculo da iluminação apresentado requer apenas as relações de adjacência e inclusão das regiões. Outros trabalhos empregam estruturas ainda mais simples para segmentar as regiões [5, 11, 12, 1] e realizar operações como transferência de cor durante a animação. Embora estas técnicas funcionem bem na colorização de uma animação de cartoons, as representações de regiões contêm informação suficiente para realizar a iluminação, sendo necessário, para isto, recorrer a outros métodos de iluminação como [7].

Algumas técnicas para iluminar objetos 2D fornecem ferramentas interativas para reconstruir modelos 3D a partir das informações 2D disponíveis [6, 13] e em seguida aplicam técnicas tradicionais de iluminação [9]. Embora estas técnicas produzam bons resultados, elas não são adequadas para cartoons, já que qualquer alteração no número de linhas do desenho pode alterar significativamente a geometria do objeto 3D. Assim, Johnston [7] apresenta uma técnica para aproximar as normais diretamente no próprio desenho do cartoon, sem a necessidade de reconstrução 3D do modelo. Ao contrário de Johnston, o método de iluminação aqui proposto dispõe de um conjunto de informações topológicas e geométricas armazenadas na árvore de regiões que permite o cálculo preciso do mapa de normais. No caso de se ter uma região circular, será demonstrado matematicamente na seção 3.1 que a reconstrução do mapa de normais aqui apresentada corresponde exatamente aos vetores normais de uma esfera.

2. Árvore de Regiões

Essa seção descreve uma estrutura topológica que contém a topologia e geometria dos objetos presentes no cartoon. Assim como em outros métodos, aplica-se inicialmente uma segmentação para extrair um conjunto de curvas e regiões [1]. Em um estágio adicional, foram exploradas outras informações que serão utilizadas para a construção da árvore de regiões. Em [1] tem-se uma descrição completa do conjunto de operações usadas para construir uma estrutura topológica típica. Essas operações incluem: esqueletonização e detecção de regiões.

Esqueletonização A esqueletonização ou *thinning* procura extrair um conjunto de pontos redundantes em uma imagem. Os pontos que não serão removidos constituem o *esqueleto* do objeto. A idéia principal é simplificar a representação do objeto, mantendo sua topologia e a conectividade das linhas. Neste trabalho, foi aplicado o algoritmo de Zhang-Suen [14].

Detecção de Regiões Esse estágio procura segmentar a imagem resultante do *thinning*, identificando todas as re-

giões que a compõem. Cada região é definida pelo conjunto de curvas (bordo e curvas internas) e pelo conjunto de pixels internos à região. Estes dois conjuntos são calculados separadamente. Primeiramente são calculados todos os pixels internos a cada região e em seguida aplica-se um rótulo a cada pixel, como uma cor ou um caracter, que identifica a região. Assim, a imagem pode ser vista como um conjunto de regiões.

Numa próxima etapa os pixels que definem o bordo e as curvas internas a cada região R devem ser ordenados para formar o conjunto de curvas poligonais de R . Embora as curvas internas não sejam necessárias para a definição de uma região, elas são importantes, pois contêm características relevantes. Em desenhos artísticos, por exemplo, as curvas internas podem fornecer uma variedade de detalhes que são diretamente responsáveis por dar um efeito mais realístico ao desenho. Para obter uma representação poligonal das curvas de cada região, pode-se aplicar o algoritmo de *chain-code* com vizinhança 8-conectado [8]. Após esta etapa cada região possuirá: uma área interna (conjunto de pixels rotulados), uma curva de bordo e um conjunto de curvas internas.

2.1. Geração da Árvore de Regiões

Estruturas topológicas usuais utilizadas em alguns métodos de colorização ajudam a rastrear regiões correspondentes de quadros consecutivos da animação [5, 11, 12, 1]. Tais estruturas exploram basicamente as relações de adjacência entre as regiões para manter essa coerência temporal, porém elas não exploram suficientemente as informações locais que estão em um mesmo quadro, como as relações de inclusão entre regiões do desenho. Assim sendo, a estrutura aqui descrita engloba a relação de subconjunto entre as regiões e explora esse fato, por exemplo, para aplicar novas operações no cartoon, como a iluminação apresentada na seção 3.

A estrutura aqui proposta é definida como uma árvore, onde cada nó contém uma região R_i e um conjunto de nós descendentes que representam as regiões internas (contidas) à R_i . Além de ser uma estrutura simples e fácil de ser implementada, ela contém mais informações do que as estruturas topológicas usuais (Figura 2).

A árvore de regiões proposta provê o controle total no acesso à cada região: curva de bordo e curvas internas, pixels interiores, regiões adjacentes e regiões contidas. Isso permite que sejam definidos atributos separadamente a cada região do cartoon como: propriedade dos materiais, cor, transparência, textura, etc. Da mesma forma, podem ser definidos atributos para cada curva como: cor, estilo de linha, espessura, etc. Os atributos das regiões podem ser armazenados em um frame buffer, numa estratégia inspirada no OpenGL [10], o que permite que os dados sejam proces-

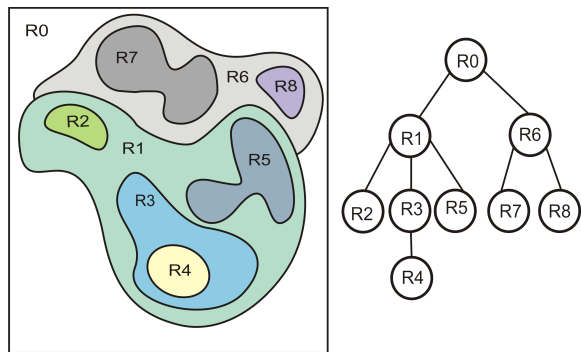


Figura 2. Hierarquia de Regiões

sados na GPU. No contexto deste trabalho, as normais são os atributos mais importantes relacionado às regiões ou curvas. A próxima seção trata desses atributos.

3. Iluminação

Esta seção apresenta uma técnica para calcular o mapa de normais baseada na árvore de regiões proposta. O cálculo das normais é realizado primeiro nas curvas de cada região e em seguida as normais são interpoladas nos pixels internos de cada região. Como as curvas de cada região estão ordenadas, as normais de cada curva foram calculadas como em [1]: dados dois pontos consecutivos $p_i = (x_i, y_i)$ e $p_{i+1} = (x_{i+1}, y_{i+1})$ da curva, calcula-se a normal como $n_i = (y_{i+1} - y_i, x_i - x_{i+1}, 0)$. A coordenada $z = 0$ indica que o plano da imagem é o plano de projeção em relação ao observador e que a curva faz parte da silhueta do objeto. Como há apenas oito possíveis direções para os vetor n_i (já que foi usada a vizinhança 8-conectado) é possível suavizar o mapa de normais, substituindo n_i por uma média ponderada entre n_i e as normais de seus vizinhos: $n_i = (n_{i-2} + 4n_{i-1} + 6n_i + 4n_{i+1} + n_{i+2})/16$.

3.1. Interpolação das Normais

Aqui é apresentada a idéia básica empregada no método de iluminação proposto. Seja C uma curva fechada parametrizada pelo comprimento de arco, ou seja, $C(s) = (x(s), y(s))$, e seja R a região delimitada por C . A normal $n(s)$ em cada ponto $C(s)$ é dada por $n(s) = (n_x(s), n_y(s)) = (y'(s), -x'(s))$. O objetivo é calcular uma normal 3D $n(p) = (n_x(p), n_y(p), n_z(p))$ para cada ponto p em R . Para calcular as componentes x e y de $n(p)$, integra-se cada contribuição da normal ao longo da curva, utilizando um peso proporcional ao quadrado do inverso da distância de p a $C(s)$. Assim, tem-se

$$n_x(p) = \frac{\int_C \frac{n_x(s) ds}{|p - C(s)|^2},$$

$$n_y(p) = \frac{\int_C \frac{n_y(s) ds}{|p - C(s)|^2},$$

onde

$$w(p) = \int_C \frac{ds}{|p - C(s)|^2}.$$

De modo a manter o o vetor normal unitário, a componente z é definida como

$$n_z(p) = \sqrt{1 - n_x(p)^2 - n_y(p)^2}.$$

Uma análise desta fórmula quando C é um círculo, isto é, $C(s) = (\cos s, \sin s)$ mostra que as normais reconstruídas equivalem às normais de uma esfera: graças à simetria radial, será considerado apenas o caso $p = (x, 0)$, com $-1 < x < 1$ (caso contrário pode-se simplesmente girar o sistema de coordenadas). Tem-se que $n_y(p)$ deve ser zero, pois a contribuição de y de cada $C(s)$ é cancelada pela contribuição de $C(-s)$. Resolvendo a integração com ajuda de um sistema de álgebra computacional, tem-se

$$w(p) = \int_0^{2\pi} \frac{ds}{(x - \cos s)^2 + \sin^2 s} = \frac{2\pi}{1 - x^2},$$

$$N_x(p) = \int_0^{2\pi} \frac{\cos s ds}{(x - \cos s)^2 + \sin^2 s} = \frac{2\pi x}{1 - x^2}.$$

Como $N_y(p) = 0$ por simetria, $N_z(p) = \sqrt{1 - x^2}$. Assim, percebe-se que o campo de normal é exatamente o mesmo de uma esfera de raio 1 centrada na origem.

Diferentemente de [7], que faz uma propagação aproximada para o cálculo das normais, a formulação aqui explicitada é precisa. Além disso, devido à flexibilidade da estrutura proposta, é possível escolher quais curvas podem contribuir na interpolação de cada região e obter os efeitos propostos em [7] (Figura 3).

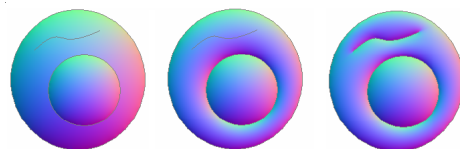


Figura 3. Diferentes Cálculos das Normais

A orientação das normais das curvas também pode ser controlada para obter outros resultados. Como cada região e curva da estrutura topológica aqui apresentada pode ser acessada independentemente, pode-se definir e aplicar diversos operadores aos atributos. Foram implementados dois operadores de normais: *operador de escala* e o *operador de profundidade*. Dada uma curva C , o operador de escala consiste de uma escala na coordenada z das normais do atributo selecionado (Figura 4).

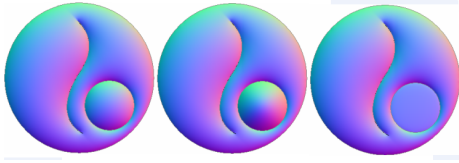


Figura 4. Operador de Escala

O operador de profundidade trabalha com a idéia de levantar/afundar uma dada curva C em uma região. O usuário controla os efeitos do operador selecionando uma curva desejada (Figura 5).

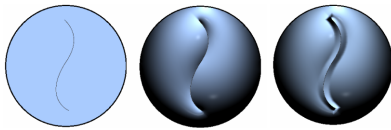


Figura 5. Operador Profundidade

As Figuras 1 e 6 mostram alguns resultados.



Figura 6. Cartoon Renderizado

4. Conclusões e Trabalhos Futuros

Este trabalho apresentou um método de iluminação de cartoons baseado em uma estruturação do cartoon como uma árvore de regiões. Diferentemente de outras estruturas, a representação aqui proposta explora a coerência espacial local de cada frame, verificando relações de subconjunto de regiões. Além disso mantém todas as informações de vizinhança presentes nas estruturas topológicas usuais. O método de iluminação proposto realiza o cálculo preciso do

mapa de normais e apresenta uma grande flexibilidade na obtenção de diferentes efeitos na visualização do cartoon.

Etapas futuras incluem: verificar a estrutura em cartoons animados; definir novos operadores de atributos; verificar quais curvas de cada região são invisíveis a um ponto interno, de modo a evitar sua contribuição no cálculo do vetor normal ao ponto; utilizar outras estruturas de dados mais eficientes para otimizar o cálculo do mapa de normais; permitir que as operações de atributos sejam enviadas e processadas na GPU a fim de acelerar o processamento.

Referências

- [1] H. Bezerra, L. Velho, and B. Feijó. A computer-assisted colorization algorithm based on topological difference. *XIX Brazilian Symposium on Computer Graphics and Image Processing*, pages 121–127, 2006.
- [2] M. De Carvalho, M. Couprie, and R. De Alencar Lotufo. Image segmentation by analysis of scale-space. *Computer Graphics and Image Processing, 2002. Proceedings. XV Brazilian Symposium on*, pages 403–, 2002.
- [3] M. de Carvalho and R. de Alencar Lotufo. Hierarchical regions matching in image sequences through association graph. *Computer Graphics and Image Processing, 2001 Proceedings of XIV Brazilian Symposium on*, pages 396–, Oct 2001.
- [4] L. Garrido, P. Salembier, and J. R. Casas. Representing and retrieving regions using binary partition trees. In *IEEE International Conference on Image Processing, ICIP 99*, 1999.
- [5] S. S. Hock and F. Tian. Computer-assisted coloring by matching line drawings. *The Visual Computer*, pages 289–304, 2000.
- [6] T. Igarashi, S. Matsuoka, and H. Tanaka. Teddy: a sketching interface for 3d freeform design. *Computer Graphics*, pages 409–416, 1999.
- [7] S. F. Johnston. Illumination for cel animation. *NPAP - 2RD International Symposium on Nonphotorealistic Animation and Rendering*, pages 45–FF, 2002.
- [8] J. Parker. *Practical Computer Vision using C*. John Wiley and Sons, Inc, New York, 1994.
- [9] B. T. Phong. Illumination for computer generated pictures. *Graphics and Image Processing*, pages 311–317, 1975.
- [10] D. Shreiner, M. Woo, J. Neider, and T. Davis. *OpenGL(R) Programming Guide : The Official Guide to Learning OpenGL(R), Version 2 (5th Edition)*. 2005.
- [11] D. Sykora, J. Burianek, and J. Zara. Segmentation of black and white cartoons. *Spring Conference on Computer Graphics*, pages 245–254, 2003.
- [12] D. Sykora, J. Burianek, and J. Zara. Unsupervised colorization of black-and-white cartoons. *NPAP - 2RD International Symposium on Nonphotorealistic Animation and Rendering*, pages 121–127, 2004.
- [13] R. C. Zeleznik, K. P. Herndon, and J. F. Hudhes. Sketch: an interface for sketching 3d scenes. *Computer Graphics*, pages 163–170, 1996.
- [14] Zhang and Suen. *A fast parallel algorithm for thinning digital patterns*. Comm, ACM, 1984.