

Utilização de arte conceitual para geração de modelos 3D

Renato D. Viana
Renato A. Celso Ferreira
Erickson R. Nascimento

Departamento de Ciência da Computação - UFMG

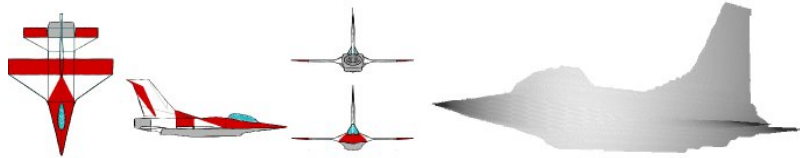


Figura 1: Arte Conceitual utilizada para geração de modelo em 3D utilizando Visual Hull (imagem à esquerda) e o modelo 3D gerado pelo algoritmo (imagem à direita).

Resumo

Este trabalho apresenta uma ferramenta que automatiza o processo de criação de modelos 3D a partir de esboços conceituais. Na abordagem tradicional para a criação de modelos 3D, um artista 2D desenvolve a arte conceitual dos objetos e, mais tarde, um modelador 3D cria o modelo manualmente utilizando esses esboços. Isto é um processo lento e oneroso. Nós desenvolvemos uma aplicação que recebe como entrada as imagens da arte conceitual, gerando como saída o modelo 3D. O aplicativo usa técnicas derivadas dos algoritmos Visual Hull e Marching Intersects. Nossos resultados são preliminares, porém muito promissores, já que a aplicação é capaz de gerar modelos precisos a partir de esboços bastante simplificados.

Palavras-chave:: Sketching, arte conceitual, modelos 3D, visual hull, voxel, marching intersections

Emails:

{renato.viana, renato, erickson}@dcc.ufmg.br

1 Introdução

Tarefas com alto grau de complexidade, tais como a criação de um cenário virtual ou objetos tridimensionais ricos em detalhes geométricos e visuais, normalmente, são executadas criando um rascunho da ideia geral daquele cenário ou objeto primeiramente. Esse rascunho provê um modelo simplificado, por exemplo, uma versão em duas dimensões de um objeto, que será utilizada para a criação de uma versão final mais detalhada, como a geração de um modelo tridimensional do mesmo objeto.

A técnica de criação de um modelo simplificado contendo os traços mais significantes de um objeto é chamada de *Sketching*. Essa técnica é largamente utilizada em diversas aplicações, e.g. criação de desenhos conceituais, esboços de paisagens e criação de cenários e objetos tridimensionais.

Neste trabalho, é proposto uma nova técnica para gerar modelos 3D a partir de esboços em duas dimensões. Normalmente, no processo de se criar modelos em 3D, utilizados na indústria de entretenimento digital, um artista 2D esboça o desenho de um personagem ou objeto em um programa gráfico de desenho como o Adobe Photoshop [Adobe 1989]. Esse tipo de desenho é o chamado *Concept Art* (arte conceitual) de um determinado objeto. Após o artista finalizar a criação do desenho conceitual, um modelador 3D, utilizando ferramentas, tais como: Maya [AutoDesk 2001b], 3D Studio Max [AutoDesk 2001a] ou Blender [Blender 2006], gera um modelo em três dimensões baseado no desenho conceitual.

Essa metodologia além de demandar longo trabalho, possui custos elevados para a indústria de entretenimento, pois requer ao menos dois artistas altamente especializados (um artista para criar a

arte conceitual e o modelador 3D). Por meio de uma ferramenta de sketching que seja capaz de analisar os esboços de desenhos criados pelo artista 2D, pode-se eliminar a segunda etapa da modelagem do objeto em 3D, tornando possível assim, a criação automática de pelo computador. A Figura 1 ilustra as 4 imagens 2D de um pequeno avião, e o modelo 3D produzido a partir delas.

2 Trabalhos relacionados

Extrair características tridimensionais a partir de imagens 2D para a criação de modelos 3D tem sido uma estratégia amplamente explorada nos últimos anos. Trabalhos como [Grimm and Joshi 2012], [Olsen et al. 2011], [Barequet and Sharir 1994] e [Markosian et al. 1999] utilizam imagens geradas por meio de projeção perspectiva, para estimar, por exemplo, o ponto 3D correspondente a um determinado pixel. Nesses trabalhos são utilizadas imagens em perspectiva de um objeto visualizado de distintos pontos de visão, bem como características como sombras [Grimm and Joshi 2012] para a geração do modelo final em três dimensões. Essa abordagem pode simplificar o trabalho de um artista 2D pois é necessário fazer somente um desenho 2D para que o modelo 3D seja gerado. Porém, em vários tipos de objetos tal abordagem se revela imprecisa visto que vários objetos 3D possuem características que não são perceptíveis por apenas uma visualização.

Em [Igarashi et al. 1999] é apresentado o sistema Teddy, desenvolvido para gerar modelos 3D baseados em formas 2D criadas pelo usuário. As formas são providas pelo usuário por meio de uma interface de desenho. Essa interface fornece ao usuário ferramentas para criar, utilizando traços 2D, a silhueta do objeto, e um conjunto de operações, e.g. criação, extrusão, suavização e corte, para fazer a modelagem do objeto 2D em três dimensões. A aplicação das operações é realizada de maneira iterativa, facilitando o aprendizado de uso do usuário e a correção do modelo 3D. A malha produzida pelo sistema é criada em tempo de execução sendo possível também corrigi-la. A Figura 2 (a) demonstra a forma de se utilizar o desenho 2D para a geração dos modelos 3D no Teddy. Na Figura 2 (b) são mostrados alguns modelos gerados por meio da interface do Teddy.

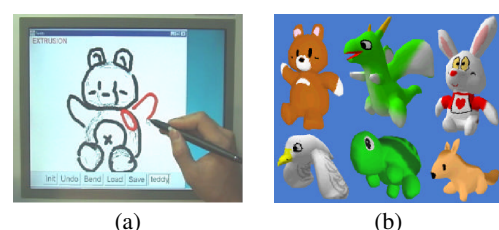


Figura 2: Utilização da interface do sistema Teddy (a) e alguns exemplos de modelos gerados (b).

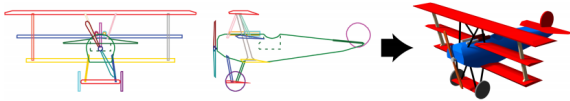


Figura 3: Exemplos de objetos 3D gerados por meio da utilização da técnica chamada Visual Hull e baseada em silhuetas (Imagem extraída do trabalho [Rivers et al. 2010]).

Similar aos softwares de modelagem CAD, [Systemes 2009b], [Systemes 2009a] e [AutoDesk 1998], este projeto propõe trabalhar com até seis visões de um objeto providas por uma câmera ortográfica: frontal, traseira, esquerda, direita, inferior e superior. Contudo, diferente desses softwares de modelagem, as visões não são imagens de um objeto 3D, mas visões diferentes de um mesmo desenho 2D representado em diversos ângulos e desenhados por um artista 2D. A esse tipo de desenho é dado o nome de desenho conceitual. Em nenhum momento é necessário fazer qualquer edição no modelo em 3D. A câmera em visão perspectiva tem, tão somente, o objetivo de visualizar o modelo 3D gerado a partir dos desenhos em 2D.

Uma abordagem similar com a proposta deste trabalho foi utilizada no projeto [Rivers et al. 2010]. Tal abordagem se sobressai às outras, pelo baixo custo computacional e pela capacidade de gerar modelos 3D mais precisos e próximos dos esboços em 2D originais. Por outro lado, sua principal desvantagem é a necessidade do artista prover o desenho do objeto em diversos ângulos. No entanto, tendo em vista que esse projeto de pesquisa visa a atender a indústria do entretenimento, esse fato não é desvantagem tão significativa, uma vez que os desenhos são usualmente feitos em diversos ângulos de visão antes de serem modelados em 3D. A Figura 3 mostra um modelo gerado pela metodologia apresentada em [Rivers et al. 2010].

A Figura 4 mostra o software de sketching desenvolvido pelo [Google 2009]. Essa aplicação utiliza técnicas de sketching para fazer alterações em modelo 3D. A abordagem utiliza uma câmera perspectiva que possibilita a visualização da cena e também possui ferramentas capazes de alterar a malha do modelo por meio de desenhos.

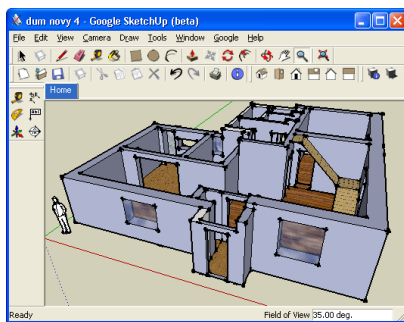


Figura 4: Tela do Sketch Up, software para geração de objetos 3D desenvolvido pelo Google.

3 Metodologia

Para fazer a geração de modelos em 3D de forma automatizada, utilizando arte conceitual, este trabalho usa uma abordagem baseada na estratégia de Visual Hull [Matusik et al. 2000]. Nessa estratégia são utilizadas duas ou mais visualizações diferentes de um mesmo objeto para reconstruir o objeto em outra visualização. A geração dessa nova visualização é baseada na interseção dos cones de visualização de cada uma das câmeras. A Figura 5 ilustra a interseção dos cones de visualização de três câmeras diferentes.

Um dos algoritmos mais tradicionais para reconstrução de modelos 3D utilizando imagens de pontos de vistas distintos utiliza o conceito de elementos de volumes (*Voxels*) para implementar uma solução de Visual Hull. O algoritmo se baseia na utilização de um *grid* de voxels. Após posicionar as câmeras em locais diferentes, a

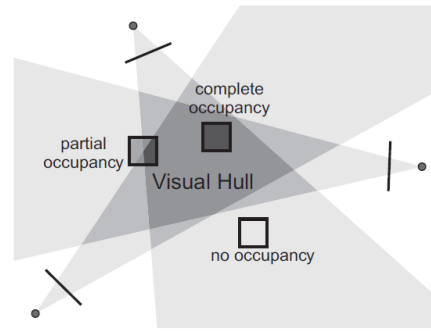


Figura 5: Imagem ilustrativa da abordagem de Visual Hull. Utilizando a interseção de cones de visualização, essa abordagem estima a estrutura geométrica de um objeto.

metodologia constrói o objeto em 3D utilizando as interseções dos cones das câmeras. A Figura 6 ilustra essa idéia.

Todos os voxels coloridos de vermelho representam uma aproximação do modelo real. Para construção do modelo em 3D, utilizando as imagens de diversas câmeras, o algoritmo processa a imagem de cada câmera separadamente e molda o objeto retirando os voxels que não está contido na visualização. Quanto mais voxels forem utilizados para fazer a aproximação do desenho 2D em um modelo 3D, melhor é a aproximação - porém, com maior custo computacional e uso de memória.

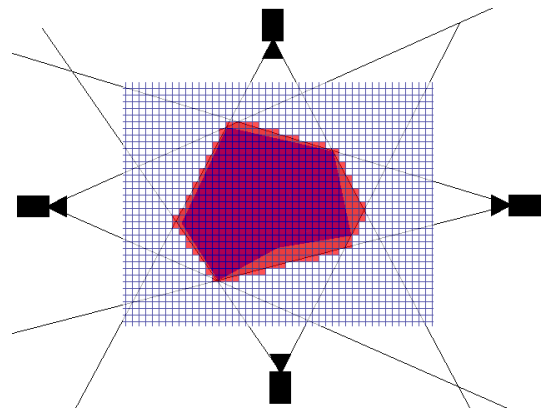


Figura 6: Interseção no grid de voxels da diversas visualizações da câmeras

As principais vantagens dessa abordagem são:

1. Tempo de execução do algoritmo controlado pelo tamanho da grid de voxels;
2. Complexidade da implementação baixa, já que não necessita de cálculos matemáticos complexos.

Porém, a geração de modelos em 3D a partir de desenho em 2D possui alguns problemas como:

1. Grande quantidade de polígonos gerados para se conseguir boa definição do modelo;
2. Não é possível detectar cavidades côncavas nas imagens;
3. Necessita que todos os desenhos conceituais tenham bordas com proporções iguais ou muito próximas.

Marching Intersects

Nossa metodologia utiliza uma variação do algoritmo tradicional de Visual Hull chamada de Marching Intersects. Esse algoritmo possui uma solução que melhora a aproximação da reconstrução do modelo em 3D, utilizando a idéia de fazer interseções entre a borda dos objetos nas imagens com os voxels.

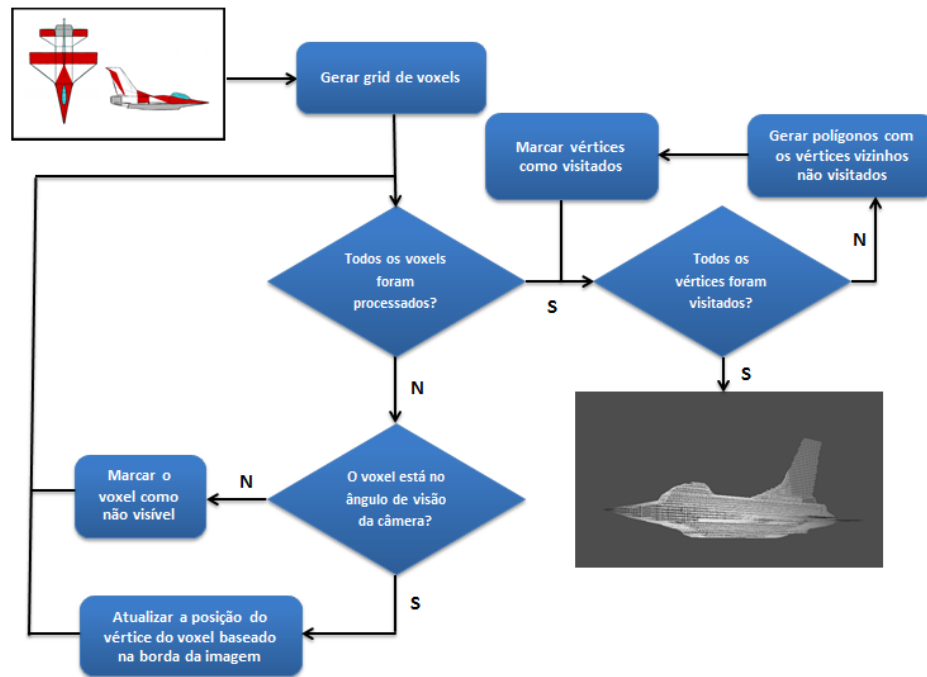


Figura 7: Fluxograma com as principais etapas da metodologia. Após determinar a visibilidade e estimar a posição de cada voxel, nossa abordagem cria uma malha de triângulos baseada nos centróides dos voxels e sua vizinhança.

O algoritmo percorre as bordas da imagem e procura encontrar o local onde a borda da imagem colide com a uma superfície do voxel. Esse procedimento é ilustrado na Figura 8. Dessa maneira, o algoritmo atualiza os vértices do voxel para posicioná-lo em relação às bordas da imagem. Apesar de produzir resultados melhores que o algoritmo tradicional, essa abordagem ainda mantém os alguns dos problemas enumerados anteriormente.

A Figura 7 mostra o fluxograma de nossa metodologia. Nós adotamos uma abordagem que utiliza no mínimo 3 visualizações do objeto: frente, lado e superior. Sendo que, se necessário, pode-se utilizar outras visualizações como: trás, visualização inferior, esquerda e direita diferentes.

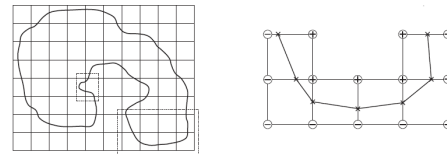


Figura 8: A figura acima mostra a silhueta de um desenho projetado sobre um grid de voxels, além disso, mostra os pontos de contatos da borda da imagem com os voxels.

```

cria a grid de voxel;
for para cada arte conceitual do
  for para cada voxel do
    if voxel está dentro da visualização then
      atualiza a posição dos vértices do voxels baseado
      na borda da imagem;
    else
      marca o voxel como não visível;
    end
  end
end
for lado da grid de voxel do
  for para cada vértice com index (i,j) em cada lado da grid
  do
    if vértice não foi visitado e o vértice pertence a um
    voxel visível then
      cria triângulo com os vértices vizinhos;
      marcar os vértice como visitado;
    end
  end
end

```

Algoritmo 1: Pseudo-código da metodologia proposta.

O pseudo-código no Algoritmo 1 apresenta de forma simplificada nossa metodologia. A primeira parte do algoritmo processa cada imagem da arte conceitual e atualiza os voxels da grid baseado na visualização câmera. As artes conceituais representam visualizações ortográficas do objeto, assim, o algoritmo processa a

visualização como uma câmera ortográfica e não perspectiva. Isso simplifica a implementação, pois o algoritmo pode processar o pixel diretamente no grid sem a necessidade de calcular a projeção do objeto em relação à câmera. Por exemplo, ao pegar a imagem frontal do objeto, o algoritmo calcula para cada pixel da imagem, qual voxel aquele pixel corresponde. Assim, se aquele pixel X, Y está vazio então o algoritmo percorre todos os voxels em Z no eixo X, Y do grid de voxel correspondente e marca todos os voxels em Z como não visíveis. Caso o pixel esteja preenchido e seu vizinho seja vazio então, esse pixel é uma borda da imagem, sendo necessário atualizar a posição do vértice lateral do voxel baseado na posição do pixel em relação ao voxel.

Esse procedimento é executado para todas as visualizações disponibilizadas para o objeto (artes conceituais). Porém, é preciso alguns cuidados na execução desse passo. Por exemplo, as imagens podem estar invertidas em relação ao eixos X, Y, Z dependendo do lado da imagem, então é necessário inverter o índice para acessar o voxel correto de acordo com a visualização. Para a visualização traseira o índice X, Y (0, 0) começa da direita e para a visualização frontal o índice X, Y (0, 0) começa da esquerda.

A segunda etapa do algoritmo tem como objetivo a criação da malha triangular. Para criar os polígonos o algoritmo deve procurar em cada lado da grid: frente, trás, inferior, superior, esquerda e direita pelos voxels visíveis mais externos, e, após encontrado cada voxel externo, o algoritmo cria triângulos entre os vértices do voxel e seus vizinhos. Por exemplo: Ao criar os triângulos pela visualização frontal da grid, é preciso para cada índice X, Y percorrer o Z de 0..n pelo primeiro voxel visível, isso é necessário porque visualizações

como superior, esquerda ou direita podem marcar os voxels mais a frente com não visível. É necessário realizar esse passo para todos os lados do voxel.

Por fim, é preciso tomar cuidado para não criar triângulos duplicados, já que, visualizações vizinhas como frente e superior podem tentar gerar polígonos em voxels iguais em suas bordas. Para evitar que isso ocorra é preciso marcar o vértice como já visitado, evitando assim a criação de polígonos repetidos.

4 Experimentos

Para avaliar a metodologia proposta neste trabalho, foram realizados experimentos utilizando artes conceituais de dois objetos: um avião com quatro pontos de vistas diferentes e uma ovelha com três pontos de vistas. As artes conceituais para cada um dos objetos é mostrada na Figura 9 (a) e (b).

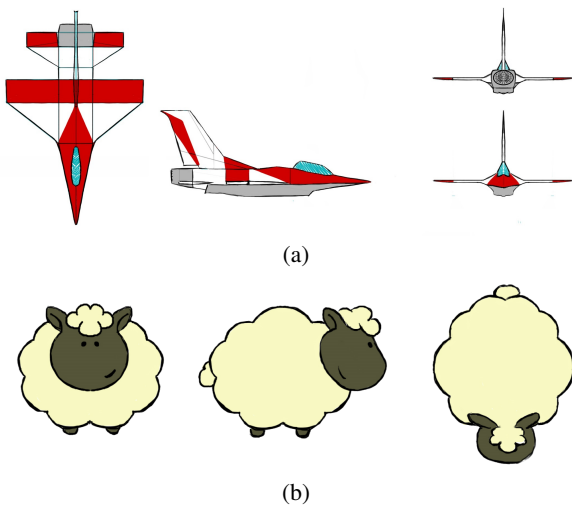


Figura 9: Arte conceitual de um avião (imagem superior) e arte conceitual de uma ovelha (imagem inferior) utilizadas nos experimentos para geração dos modelos 3D.

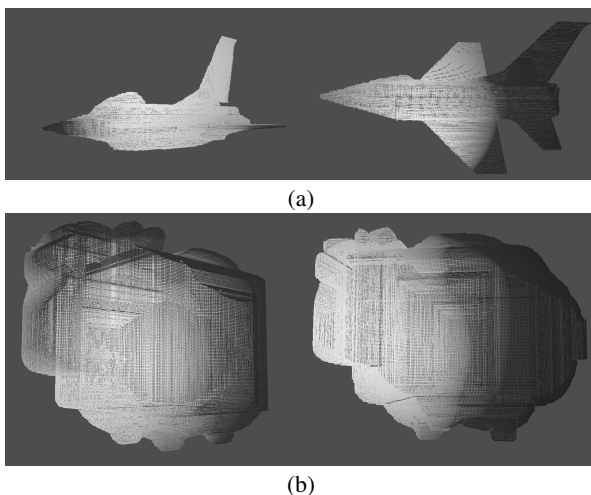


Figura 10: Modelos 3D gerados a partir da metodologia utilizando a arte conceitual da ovelha e do avião, visualizados em ângulos diferentes.

As artes conceituais desses dois objetos foram utilizadas como dados de entrada para a metodologia gerar um modelo 3D de cada objeto. A Figura 10 apresenta os modelos 3D em Wireframe gerados a partir da arte conceitual da Figura 9. Podemos perceber que a topologia das superfícies dos dois objetos foi estimada corretamente, mesmo para um objeto com uma geometria mais complexa como é o caso do avião.

5 Conclusão e Trabalhos Futuros

O algoritmo de marching intersects é uma solução muito interessante para automatizar o processo de geração de modelos 3D, pois não possui um tempo de processamento elevado e é de fácil implementação. Porém, alguns problemas, como quantidade de polígonos gerados, superfícies côncavas e tratamento de proporções diferentes das bordas das artes conceituais criadas pelos artistas 2D, ainda precisam ser melhorados para que se tenha uma solução robusta que atenda à indústria do entretenimento digital.

Como trabalhos futuros, nós planejamos incluir um parâmetro responsável para controlar a quantidade de polígonos gerados, aplicar textura extraída das imagens no modelo gerado, realizar outros experimentos quantificando a qualidade das malhas geradas pela metodologia e comparar os resultados com modelos gerados manualmente por artista 3D.

Referências

- ADOBE, 1989. Photoshop. <http://adobe.com/photoshop>.
- AUTODESK, 1998. Auto cad. <http://usa.autodesk.com/autocad/>.
- AUTODESK, 2001. 3d studio max. <http://www.autodesk.com/3dsmax>.
- AUTODESK, 2001. Maya. <http://www.autodesk.com/maya>.
- BAREQUET, G., AND SHARIR, M. 1994. Piecewise-linear interpolation between polygonal slices. *ACM 10th Computational Geometry Proceedings 1*, 93–102.
- BLENDER, 2006. Blender foundation. <http://www.blender.org/>.
- GOOGLE, 2009. Google sketchup. <http://www.sketchup.com/>.
- GRIMM, C., AND JOSHI, P. 2012. An Interface for Sketching 3D Curves. *EUROGRAPHICS Proceedings of the International Symposium on Sketch-Based Interfaces and Modeling 1*, 121–130.
- IGARASHI, T., MATSUOKA, S., AND TANAKA, H. 1999. Teddy: A Sketching Interface for 3D Freeform Design. *ACM SIGGRAPH 1*, 1–8.
- MARKOSIAN, L., COHEN, J., CRULLI, T., AND HUGHES, J. 1999. An interface for sketching 3d curves. *Symposium on Interactive 3D Graphics 1*, 17–21.
- MATUSIK, W., BUEHLER, C., RASKAR, R., AND MCMILLAN, L. 2000. Image based visual hulls. *ACM Transactions on Graphics 27*, 369–374.
- OLSEN, L., SAMAVATI, F., AND JORGE, J. 2011. Naturasketch: Modeling from images and natural sketches. *IEEE Computing Graphics Appl. 31 1*, 24–34.
- RIVERS, A., DURAND, F., AND IGARASHI, T. 2010. 3D Modeling with Silhouettes. *ACM Transactions on Graphics 29*, 109.
- SYSTEMES, D., 2009. Catia. <http://www.3ds.com/products/catia/>.
- SYSTEMES, D., 2009. Solid works. <http://www.solidworks.com/>.