

Geração de ilustração a partir de modelo 3-D texturizado com imagem vetorial

Haroldo Watson Teodósio da Silva, Rafael Beserra Gomes (Orientador) e Luiz Marcos Garcia Gonçalves (Co-Orientador)

Laboratório Natalnet
Universidade Federal do Rio Grande do Norte
<http://www.natalnet.br/>
haroldowatson@gmail.com

Proposta

Trabalhar com textura vetorial para gerar imagens vetoriais não-fotorrealísticas sobre modelos 3D de objetos representados numa cena, sem perder resolução. Um método para utilizar imagens SVG como textura, recortando os polígonos através do algoritmo de Sutherland-Hodgman.

1 – Introdução

Existem vários algoritmos para gerar ilustrações a partir de modelos 3D, mas poucos lidam com modelos texturizados e menos ainda formam polígonos fechados. É proposta, neste trabalho, uma técnica para gerar imagens vetoriais não-fotorrealistas que correspondem ao mapeamento de texturas vetoriais sobre modelos 3D projetados na tela.

O algoritmo de Sutherland-Hodgman é adaptado para recortar os objetos geométricos da textura e projetá-los na tela. A principal inovação é a geração de uma ilustração vetorial, invariante à resolução, que representa a visualização de uma malha de triângulos 3-D com mapeamento de uma textura, também vetorial. Assim, uma imagem vetorial pode ser renderizada diretamente como textura para uma ilustração que representa o objeto 3-D, sem rasterizar a textura, sem utilizar pipeline gráfico e tendo como resultado um gráfico vetorial, não uma matriz de pixels.

2 – Trabalhos Relacionados

Há técnicas para gerar ilustrações a partir de objetos tridimensionais (Markosian, 1997; DeCarlo, 2003, Judd, 2007; Rusinkiewicz et al., 2008) incluindo comparação das técnicas, com foco ao desenho de linhas e estilização, mas não geração de polígonos fechados. Karsch et al. (Karsch, 2011) usam *snaxels* para definir polígonos fechados na imagem de saída, gerando resultados no formato SVG. Há poucos estudos com mapeamento de texturas vetoriais, incluindo trabalhos que gerem ilustrações invariantes à resolução usando texturas vetoriais. Winkenbach e Salesin (1994) geram ilustrações com estilo de traçado artístico a partir de texturas do mesmo tipo, mas não envolvem polígonos fechados.

Estratégia

O algoritmo de Sutherland-Hodgman é adaptado para recortar os objetos geométricos da textura e projetá-los na tela. A única restrição é que os polígonos de corte sejam convexos, mas isso não traz dificuldades, já que os polígonos de corte aqui utilizados são todos triângulos.

3 – Método proposto

A ideia é recortar os caminhos na imagem da textura, usando uma versão modificada do algoritmo de Sutherland-Hodgman (1974), e transformar os caminhos resultantes para o espaço da tela.

3.1 – Textura vetorial

Uma imagem vetorial pode ser ampliada e rotacionada sem perda de sua qualidade visual ou de informação. Além disso, a natureza descritiva das imagens vetoriais permite que elas armazenem informação em um alto nível de abstração, relacionando-a com os aspectos visuais específicos de seus elementos, o que resulta em uma riqueza de dados impossível quando se tem apenas valores de cores.

3.2 - A transformação final de um elemento, resultante da multiplicação de sua matriz individual pelas de seus ancestrais, é chamada de Current Transformation Matrix, ou CTM. Assim, um ponto X_e descrito no espaço de um elemento C , cujo ancestral direto é B , o qual tem apenas um ancestral, A , será mapeado para o ponto X_i da imagem.

3.3 – Caminho

O formato SVG disponibiliza um conjunto de tipos de elementos gráficos para compor as imagens. Alguns desses elementos são mais primitivos e outros são mais complexos, mas para os fins deste trabalho, nos concentraremos em uma primitiva chamada caminho. Um caminho SVG é uma sequência de comandos de traçado de desenho que orientam sobre a aparência de um certo elemento visual.

Comentários

O método está sendo testado no momento. A vantagem das imagens vetoriais é o seu aspecto invariante à resolução. O mesmo arquivo, com o mesmo tamanho, pode ser renderizado em detalhes sem perda de qualidade.

4 - Recorte de caminhos

Cada polígono de corte é subdividido em planos de corte, cada um definido por seu vetor normal e um de seus pontos. Como o processo é feito no plano 2-D, poderia se esperar que os objetos de corte fossem retas, cada uma definida por dois pontos, e essa visão é reforçada pelo fato de que cada plano de corte é calculado a partir de uma aresta de um triângulo, mas essa representação não é eficaz.

Se um ponto não está na área visível, então ele está sobre o plano ou na área não visível. Nota-se a vantagem da representação para o objeto de corte, como um plano definido por uma normal e um ponto nele, em vez de uma reta definida por dois pontos. No segundo caso, seria preciso determinar valores adicionais para se determinar em qual área um ponto está. A interseção das três áreas visíveis dos planos de corte corresponde ao interior do triângulo de corte. A ideia principal é que os segmentos do polígono a ser recortado são submetidos ao teste de intersecção com cada um dos planos de corte, sendo a saída de cada etapa repassada como entrada à etapa seguinte. Cada segmento k pode ser entendido como uma função paramétrica $f_k(t)$.

5 – Resultados parciais

Os triângulos de corte, e seus planos de corte, são descritos em coordenadas no espaço da imagem. Cada caminho está descrito por coordenadas no espaço do elemento, que está sujeito à sua matriz de transformação atual. Em geral, tentar fazer o recorte sem levar em conta essa transformação produz resultados incorretos. Uma abordagem que deve funcionar, nesse caso, é transformar todos os pontos de cada caminho antes do recorte. No entanto, uma ideia ainda mais eficiente é transformar cada plano de corte com a inversa da CTM de cada elemento antes de iniciar o procedimento de recorte. As ideias e os algoritmos estão sendo ainda implementadas, não tendo sido formalizados resultados práticos.

Conclusão

A técnica aumenta substancialmente o controle do artista sobre a aparência da imagem final, quando comparada a métodos que não usam texturas, sem perder a característica de ser invariante à resolução. O formato SVG, no entanto, é muito flexível, e algumas nuances do seu uso como textura ainda não estão resolvidas. Deixamos para a continuação dos nossos trabalhos uma modificação do algoritmo de recorte para curvas Bézier quadráticas e cúbicas, bem como a otimização para animação.