

4 Detalhamento de superfícies baseado em imagens

Este capítulo realiza um breve estudo sobre algumas técnicas propostas para detalhamento de superfícies baseado em imagens. Serão explicados os fundamentos de cada abordagem e apontados os critérios que levaram à seleção da técnica de *parallax occlusion mapping* como base para comparação dos resultados deste trabalho.

4.1 Parallax mapping

A técnica de *parallax mapping* (Kaneko, et al., 2001) foi proposta com o objetivo de acrescentar à simulação de rugosidade o *motion parallax*, que consiste no aparente deslocamento de partes de um objeto de acordo com mudanças no ponto de vista do observador (sendo normalmente observado através do bloqueio entre protuberâncias da superfície), por meio de uma aproximação simples do efeito de um mapa de altura. Isto não pode ser alcançado através do algoritmo original de *bump mapping* por não ser levada em consideração a informação de profundidade do modelo.

4.1.1. Algoritmo original

A superfície mostrada na Figura 8 é representada por um polígono sobre o qual foi apenas mapeada uma textura. Quando observada da direção especificada pelo vetor *eye*, é visualizado o ponto A, que corresponde a uma representação incorreta da curvatura real da superfície, a qual, por consequência, parece ter sofrido um achatamento. Nesta situação, o ponto B corresponde à correta representação dessa curvatura, mas requer algum tipo de correção para que possa ser visualizado adequadamente.

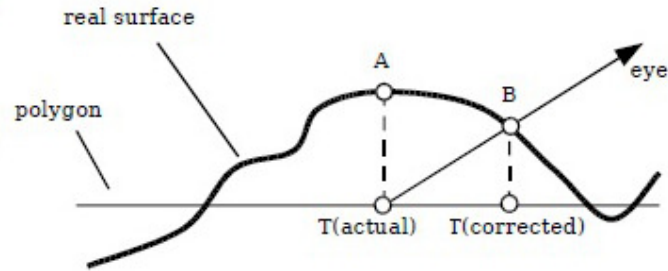


Figura 8: Visualização incorreta da curvatura da superfície (Welsh, 2004).

O algoritmo proposto para tal correção realiza o efeito de paralaxe através da distorção dinâmica da textura mapeada, de forma que ela corresponda à curvatura do objeto representado. Essa distorção não é realizada diretamente sobre a textura, mas através do deslocamento das coordenadas de textura de acordo com um mapa de altura, como mostra a Figura 9. A quantidade de deslocamento depende do ângulo de visão com a superfície e do valor de altura obtido, descrito apenas como um escalar no intervalo [0,0, 1,0].

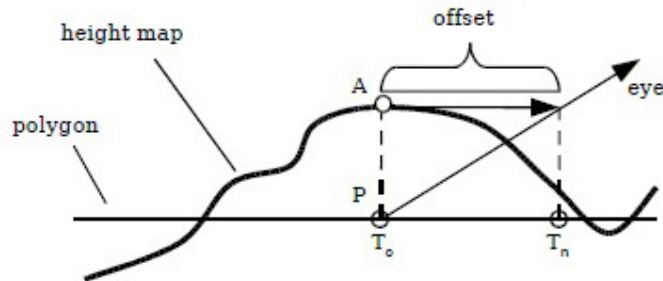


Figura 9: Deslocamento de coordenadas de textura (Welsh, 2004).

O deslocamento das coordenadas de textura para o ponto P é calculado com base no vetor eye normalizado, no valor de altura h obtido do mapa nas coordenadas de textura originais T_o . Traça-se um vetor paralelo ao polígono partindo do ponto A , pertencente ao perfil de curvatura da superfície e localizado diretamente sobre o ponto P , até eye , representado pelas componentes horizontais, eye_x e eye_y e pela altura em relação à superfície, descrita pela componente no eixo z , eye_z . Conforme mostra a Equação 4-1, esse vetor pode ser somado diretamente às coordenadas originais T_o para produzir as coordenadas deslocadas T_n .

$$T_n = T_o + \frac{h \cdot eye_{xy}}{eye_z} \quad (4-1)$$

4.1.2. Limitação de deslocamento

A aproximação simples apresentada na Seção 4.1.1 resulta em efeitos de profundidade razoáveis quando as irregularidades da superfície não são acentuadas, mas introduz deformações consideráveis à medida que o observador se aproxima da superfície. Isto ocorre porque, em ângulos de visualização pequenos, o valor definido para o deslocamento tende a infinito, resultando em um mapeamento que em nada reflete o aspecto da superfície desejada.

Para contornar essa situação, Welsh (Welsh, 2004) propôs a ideia de limitar o deslocamento³ em torno do ponto em que ocorre a interseção do vetor de visualização com o polígono da superfície. A solução consiste apenas em limitar o valor máximo do deslocamento de forma que ele nunca seja superior à altura obtida pelo mapa para esse ponto, como mostra a Figura 10.

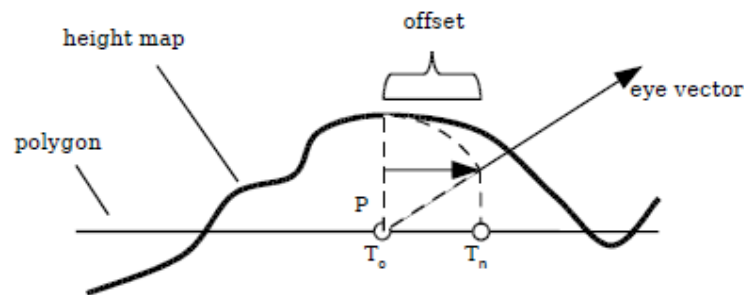


Figura 10: Parallax mapping com limitação de deslocamento (Welsh, 2004).

Observa-se também pela Figura 10 que esta abordagem simplificada pode resultar em uma maior percepção do efeito de achatamento da superfície, já que as novas coordenadas de textura podem não corresponder exatamente ao ponto em que o vetor de visualização intercepta a curvatura da superfície, mas isto evita o descontrole da quantidade de deslocamento quanto menor for o ângulo de visualização com o polígono e melhora a qualidade visual oferecida pela técnica. O novo procedimento para o cálculo para as novas coordenadas de textura é mostrado na Equação 4-2.

$$T_n = T_o + h \cdot eye_{xy} \quad (4-2)$$

³ *Offset limiting* na literatura original.

4.2 Parallax occlusion mapping

O *parallax occlusion mapping* (POM) (Brawley & Tatarchuk, 2004) (Tatarchuk, 2006) surgiu junto a um número de algoritmos para detalhamento de superfícies desenvolvidos no mesmo período, mas por grupos de pesquisa diferentes, os quais também nomearam suas abordagens com nomes distintos: *relief mapping* (Policarpo, Oliveira, & Comba, 2005) e *steep parallax mapping* (McGuire & McGuire, 2005). Na prática, esses algoritmos realizam abordagens muito similares para representar irregularidades em superfícies, mantendo o efeito de *parallax* e adicionando a possibilidade de simular *self-shadowing* e *self-occlusion* em um modelo visto de diferentes ângulos. O princípio dessas técnicas será descrito na Seção 4.2.1 e o algoritmo para amostragem apresentado Tatarchuk (Tatarchuk, 2006) será examinado na Seção 4.2.2.

4.2.1. Visão geral

A ideia dos algoritmos apresentados aqui é baseada na amostragem de uma textura ao longo de um raio projetado sobre a superfície, a qual é descrita por um mapa de profundidade (em que o polígono pertencente à superfície define o valor base na escala [0.0, 1.0]), tendo como objetivo a determinação do ponto de interseção entre raio e perfil de curvatura que esteja mais próximo ao observador, o qual corresponde ao mais próximo ponto visível da superfície, como mostra a Figura 11.

O cálculo do ponto é realizado por meio de uma busca linear pela primeira amostra localizada abaixo do mapa de profundidade. Essa amostra é usada junto à amostra anterior, localizada acima do perfil desse mapa, para resolver um problema de determinação de raiz que define o ponto de interseção desejado.

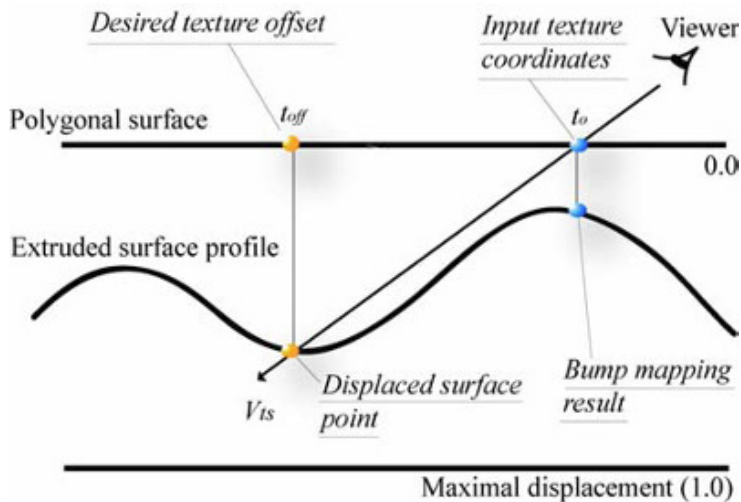


Figura 11: Determinação do ponto de interseção mais próximo (Tatarchuk, 2006).

Para a determinação da raiz, Tatarchuk (Tatarchuk, 2006) utiliza um passo do método secante, permitindo o uso mais eficiente do hardware gráfico, dado que as amostragens da textura podem ser feitas em paralelo. Policarpo (Oliveira & Policarpo, 2005), por outro lado, realiza uma busca binária entre os dois pontos, a qual resulta em menos acessos à textura, mas sofre com a latência causada pela dependência entre cada acesso. Szirmay-Kalos e Umenhoffer (Szirmay-Kalos & Umenhoffer, 2008) realizam um bom estudo sobre esses e outros métodos de busca por interseções.

4.2.2. Amostragem do mapa de profundidade

O procedimento para determinação do ponto de interseção entre raio e mapa de profundidade é iniciado com o cálculo do vetor de deslocamento de *parallax* (*parallax offset vector*) P , o qual determina a direção em que deve ser feita a amostragem e a maior quantidade de deslocamento a ser realizado durante o cálculo da correção das coordenadas de textura. O cálculo do vetor é realizado com base nos vetores de visualização, tangente, binormal e normal para cada vértice pertencente à superfície. Todas as operações são realizadas no espaço tangente, sendo necessário, portanto, transformar para este espaço e normalizar cada vetor de interesse. O procedimento detalhado para o cálculo do vetor de deslocamento de *parallax* é apresentado por Brawley e Tatarchuk (Brawley & Tatarchuk, 2004).

modelos arbitrários, mas esse fator resulta no achatamento das feições da superfície em direção ao horizonte. O *parallax mapping* original é uma técnica de implementação bastante simples e apresenta bons resultados para superfícies com protuberâncias pequenas, mas as distorções são claramente perceptíveis para mapas de altura com altas frequências ou quando a superfície é vista de ângulos rasantes.

Observa-se atualmente a utilização cada vez mais frequente do POM em jogos eletrônicos, que também já vem sendo suportado nativamente em *engines* bastante conhecidos, como CryEngine 3 (Crytek, 2012). A Figura 13 mostra uma comparação entre POM e *normal mapping* no ambiente do jogo Crysis 2 (Crysis 2, 2011), desenvolvido com o *engine* mencionado. Portanto, pelas razões apresentadas aqui, optou-se pela implementação do POM como base para uma análise comparativa dos resultados da implementação utilizando tesselação em *hardware* que é apresentada neste trabalho.



Figura 13: Renderização de cena com normal mapping e parallax occlusion mapping (imagem do jogo Crysis 2) (Crysis 2 DX11, 2011).