5 Medida de Erro para Impostores com relevo

Have you ever had a dream, Neo, that you were so sure was real? What if you were unable to wake from that dream? How would you know the difference between the dream world and the real world? (The Matrix)

5.1 Introdução

No pipeline padrão da GPU todos os objetos poligonais devem ser visualizados a cada frame, mesmo que nada tenha sido alterado em relação aos objetos e ao observador. Como foi discutido na seção 3.3, os impostores convencionais otimizam a visualização porque a geometria dos objetos que representam não precisa ser transformada e projetada a todo instante. De igual maneira, um impostor com relevo não precisa ser recalculado enquanto haja uma coerência visual, podendo assim ser reutilizado durante vários *frames* seguidos. Um impostor com relevo tem seu tempo de vida estendido devido a sua capacidade de corrigir a paralaxe. Divide-se, assim, a validade do impostor com relevo em 3 estágios, separados por dois momentos críticos:

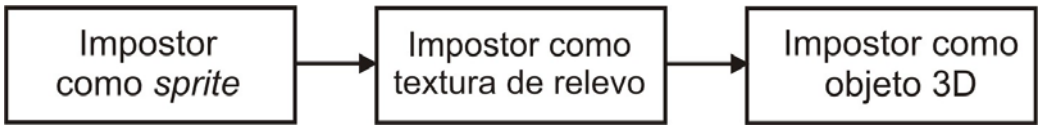

figura 5.1 – Estágios de um impostor com relevo.

Um impostor como *sprite* corresponde aos instantes em que o observador está dentro do limite de tolerância descrita pela equação de Schaufler (Schaufler, 1995). Este impostor é limitado pelo momento crítico em que esta tolerância deixa de ser válida. Neste caso o impostor usa a sua textura com relevo corrente para realizar um *warping* adequado, voltando logo em seguida a comportar-se como um impostor *sprite*. Este ciclo será realizado até o momento em que o *warping* começa a deformar o objeto que está sendo representado. Este momento é descrito

pela equação 5-7, que está descrita na seção 5.5. Neste caso, chega-se ao segundo momento crítico, momento este em que a textura com relevo do impostor não é mais válida, sendo necessário gerar uma nova textura, utilizando para isto os dados da posição do observador no ponto em que se dá este momento crítico e o modelo geométrico do objeto sendo representado.

Em qualquer um destes estágios, pode ocorrer do impostor deixar de ser válido por problemas de amostragem, o que na prática se dá devido a um movimento de aproximação do observador ao objeto. Este problema deve ser resolvido através de uma nova renderização do impostor numa resolução maior.

5.2 Criação do Impostor com Relevo

Num primeiro instante, deve-se determinar qual é a resolução inicial do impostor. Isto depende da distância em que o objeto se encontra em relação ao observador, pretendendo evitar que se desperdice tempo gerando *pixels* que depois são perdidos por motivo de amostragem.

Para determinar o tamanho do impostor a ser criado utiliza-se a seguinte aproximação proposta por Schaufler (1995):

$$resolução\_impostor = resolução\_tela \frac{Tamanho\_do\_objeto}{2 \times dist\_cam \times tg(FOV/2)} \qquad (5\text{-}1)$$

Onde *Tamanho_do_objeto* é a medida da *Bounding Box* paralelo ao plano de projeção da tela, dist_cam é a distância do objeto à câmera e FOV é o ângulo de abertura da lente da câmera.

5.3 Atualização do Impostor com Relevo

Como foi visto, existem basicamente dois movimentos do observador que fazem um impostor estático se tornar obsoleto, sendo portanto necessário um novo cálculo de *warping* ou de visualização para atualizá-lo:

1) O observador se aproximou do objeto, sendo necessário gerar uma textura com relevo para o impostor com uma resolução maior. No presente trabalho, denomina-se este movimento de translação normal ao plano da imagem (*transN*), i.e., na direção correspondente ao *Z* da câmera.

2) O observador se moveu num plano paralelo ao plano de projeção, podendo surgir erro de paralaxe do objeto mapeado. Este erro pode ser corrigido com um *warping* da textura com relevo ou através da geração de uma nova textura, no caso desta já não servir. No presente trabalho, denomina-se este movimento de translação paralela do observador (*transP*).

Para testar o primeiro caso, utiliza-se a aproximação de Schaufler (Schaufler, 1995): toma-se um dos extremos de um *bounding box* criado para o objeto sendo representado pelo impostor, conforme se pode ver na figura 5.2. A idéia é avaliar como estes pontos extremos se projetam no plano do impostor quando o observador se aproxima do mesmo. Tem-se que C_1 corresponde à posição inicial do observador e C_2 à posição que se deseja testar a necessidade de um acréscimo da resolução. Tendo-se que $\beta_{tela} = \frac{FOV}{resolução\_tela}$ e $\beta_{TransN} = B_1\hat{C}_2B_0$, sempre que $\beta_{transN} > \beta_{tela}$ deve-se refinar a textura do impostor. Esta correção pode ser realizada através de um cálculo de visualização apenas para os novos pixels que surgiram, reaproveitando os que já estavam renderizados, de maneira a não recalcular os *texels* já existentes.

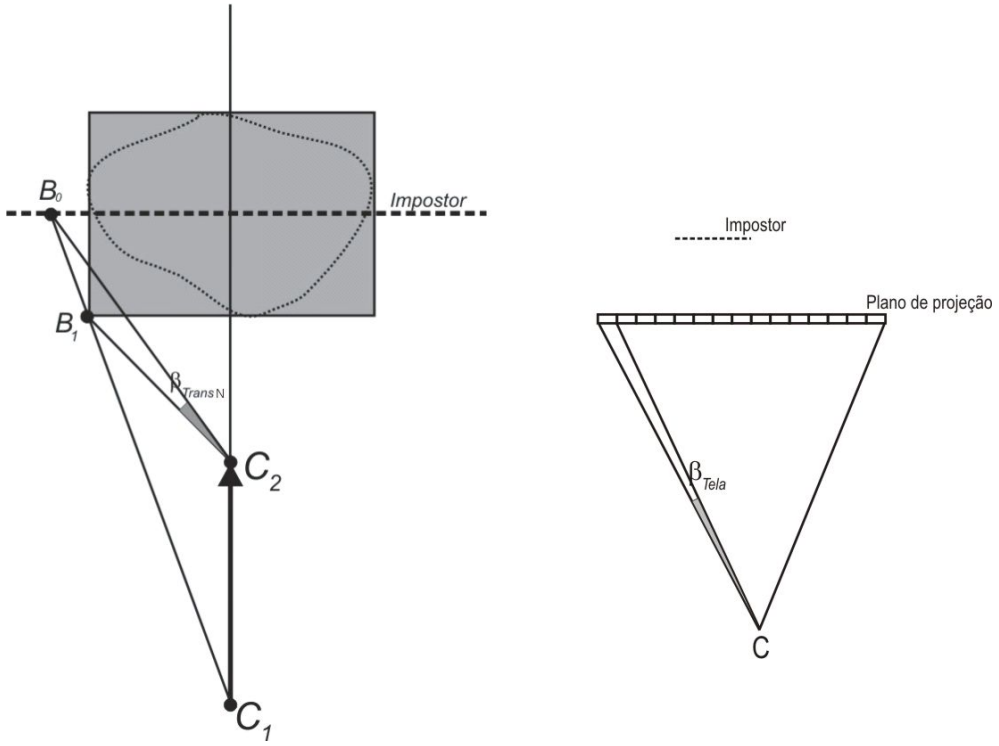

figura 5.2 –Medida de erro de Schaufler para movimento de aproximação do observador ao objeto representado pelo impostor (Schaufler, 1995).

O segundo caso requer mais cuidado: Antes de mais nada, através da aproximação de Schaufler, realiza-se um teste para saber se o impostor ainda é válido como *sprite* (momento crítico 1). Chama-se de ângulo de erro da translação paralela (β_{transP}) o ângulo criado pelos pontos extremos do *bounding box* quando o observador se move (figura 5.3). O deslocamento aparente do objeto devido ao movimento do observador é chamado de paralaxe. Caso $\beta_{transP} > \beta_{tela}$, o impostor não pode mais ser visto como o *sprite* que era, sendo necessário realizar um *warping* de sua textura para corrigir o paralaxe. Antes disto, porém, deve-se fazer uma estimativa para prever se o impostor a ser produzido pelo *warping* é válido, ou seja, se ao realizar o *warping* na sua textura, o *sprite* do objeto não possuirá excessivos erros acumulados provindos de uma extrapolação. Quando isto ocorrer, significa que se chegou ao segundo momento crítico, sendo necessário gerar uma nova textura com relevo para o objeto, utilizando a posição corrente da câmera.

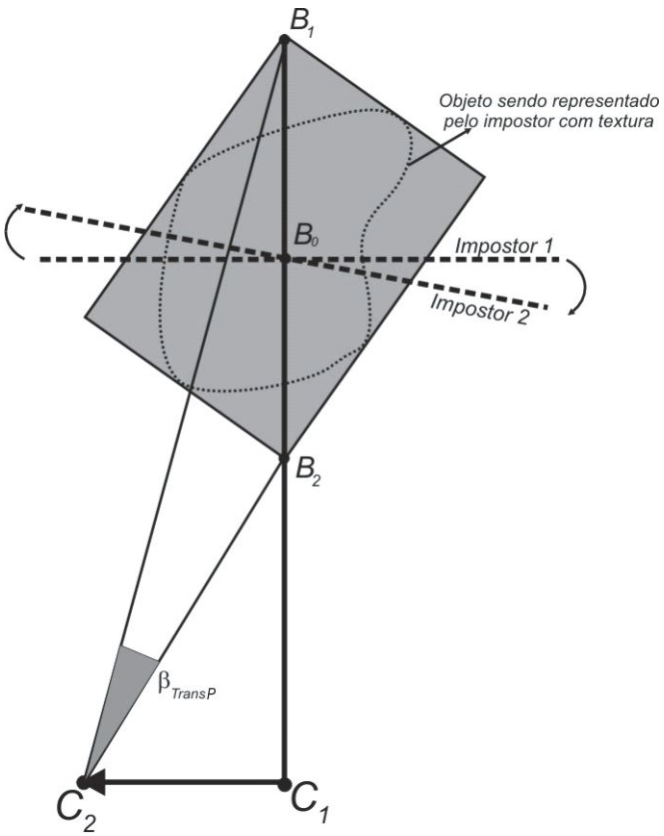

figura 5.3 – Medida de erro para movimento paralelo ao plano de projeção (Schaufler,1995).

Como foi visto na seção 3.4, a interpolação consiste no preenchimento dos buracos que surgiram entre *pixels* que eram vizinhos na figura original. Este

preenchimento gera num erro provocado pela falta de informação da imagem sobre o local. De um modo geral, quanto maior a descontinuidade da superfície que está sendo representada, maior é este erro. No presente trabalho desenvolve-se uma métrica baseada no erro acumulado nesta interpolação para uma textura com relevo, numa dada posição (figura 5.4).

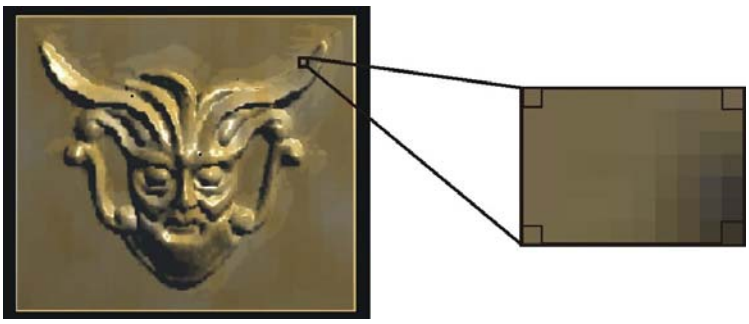

figura 5.4 – Dois *texels* vizinhos, ao se afastarem, criam um "buraco" que é preenchido por *texels* interpolados. Uma possível métrica de erro consiste em monitorar esta interpolação numa região crítica, que são pontos sujeitos a um erro maior.

5.4 Métrica de Erro Acumulado para Impostores com relevo

Para a estimativa desenvolvida é conveniente ver o impostor com relevo conforme ilustra a figura 5.5. Nela pode-se observar a seguinte equivalência de triângulos:

$$\frac{\Delta u}{B_u} = \frac{\partial(u_f, v_f)}{H} \tag{5-2}$$

onde B_u corresponde à distância entre a componente horizontal da projeção do ponto epipolar do impostor com relevo u_e até u_f, que é a projeção de $\dot{X}$ sobre o plano da textura com relevo e $H = h + \partial(u_f, v_f) = \vec{c}.\vec{f} + \partial(u_f, v_f)$, sendo estes parâmetros os mesmos já descritos na seção 3.4.

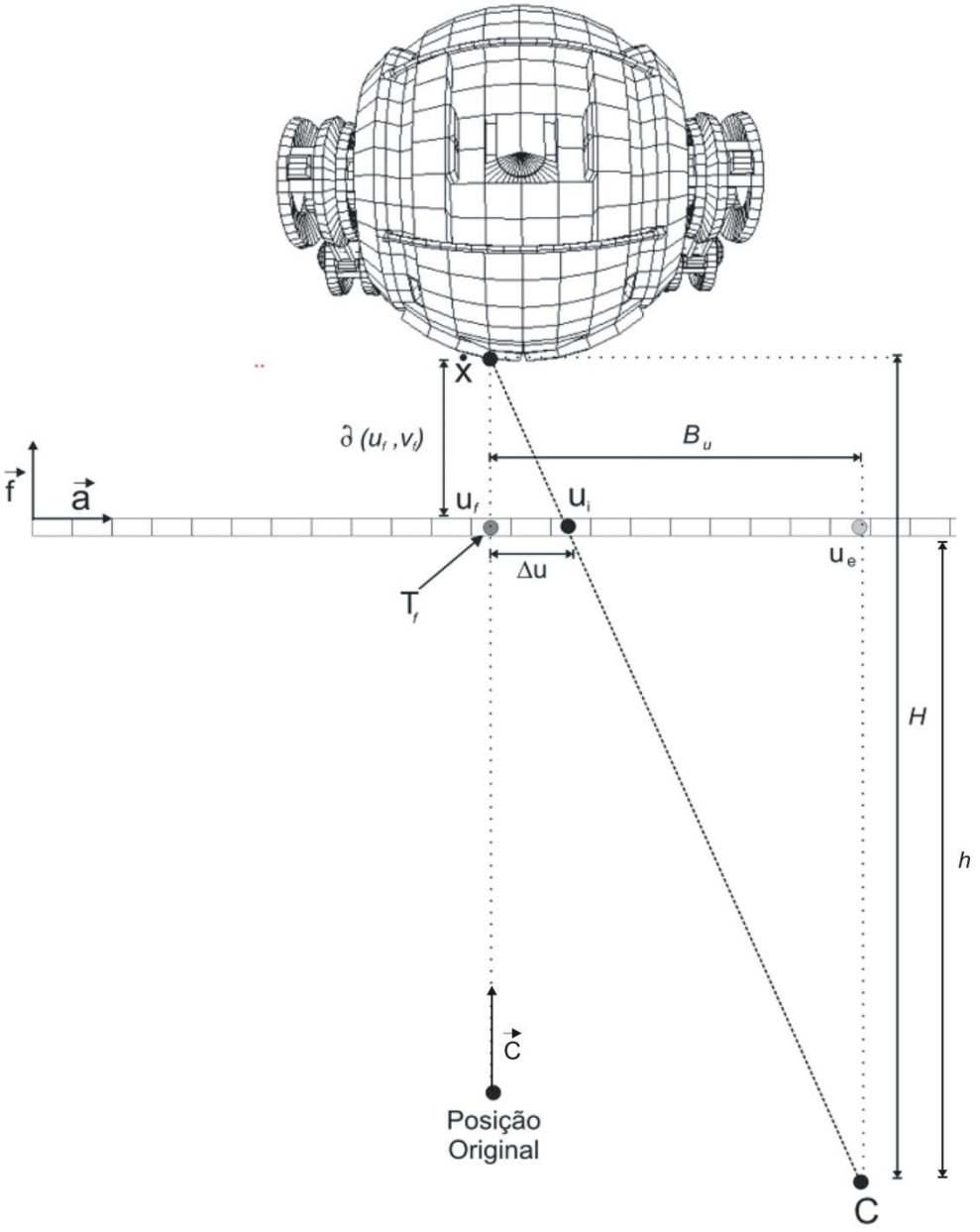

figura 5.5 - O *warping* de uma textura com relevo pode ser descrito da seguinte forma: para cada *pixel* da imagem T_f, realiza-se um deslocamento sobre as colunas da imagem Δv e sobre as linhas da imagem Δu. Estes deslocamentos dão a nova posição v_i e u_i correspondente ao *pixel* para a posição da câmera *C* a partir da sua posição original v_f e u_f, fazendo-se $v_i = v_f + \Delta v$ e $u_i = u_f + \Delta u$. A presente figura ilustra o deslocamento horizontal Δu.

Assim sendo, tem-se que $B_u = u_e - u_f$. Pode-se interpretar através da equação (5-2) que o deslocamento Δu a ser aplicado sobre o *pixel* de $\dot{X}$ é inversamente proporcional à distância de *C* ao plano do impostor, para valores de $\partial(u_f, v_f) \neq 0$.

Desta forma, chega-se à seguinte equação (Oliveira, 2000):

$$\Delta u = \frac{(u_e - u_f)\partial(u_f, v_f)}{\vec{c}.\vec{f} + \partial(u_f, v_f)} \tag{5-3}$$

De forma semelhante, pode-se chegar à seguinte formulação para o deslocamento vertical a ser aplicado sobre o *texel*:

$$\Delta v = \frac{(v_e - v_s)\partial(u_f, v_f)}{\vec{c}.\vec{f} + \partial(u_f, v_f)} \tag{5-4}$$

Intuitivamente, pelas equações (5-3) e (5-4) pode-se notar que quanto maior for o deslocamento do observador *C* em relação a posição original *C'*, maior tendem a ser os valores absolutos de Δu e Δv para um *texel* do impostor e portanto maior a tendência a serem criados *texels* interpolados.

Assim, sejam T_1 e T_2 dois *texels* pertencentes à mesma textura com relevo, tal que T_1 e T_2 são vizinhos e que $|\partial(u_1, v_1) - \partial(u_2, v_2)|$ é máximo para qualquer par nestas condições. Então T_1 e T_2 são fortes candidatos para que $|(u_{f_1} + \Delta u_1) - (u_{f_2} + \Delta u_2)|$ seja máximo em toda a textura.

Esta afirmação pode ser facilmente comprovada com o auxílio da figura 5.6, onde $\Delta h = \partial(u_1, v_1) - \partial(u_2, v_2)$, e pode ser interpretado intuitivamente da seguinte maneira: a região onde pode ocorrer maior descontinuidade no campo de profundidade da textura com relevo é o local onde se acumula o maior erro de interpolação durante o *warping* (figura 5.6).

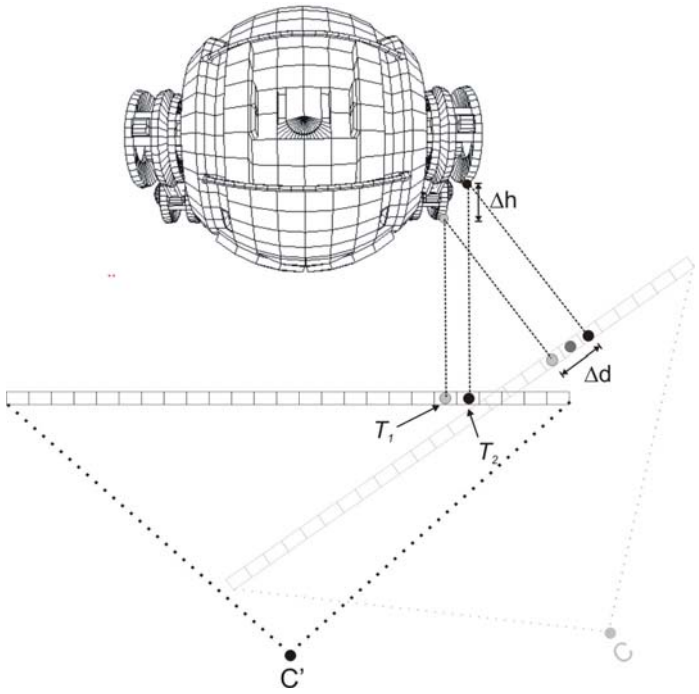

figura 5.6 – Δ*h* corresponde ao valor da maior descontinuidade para a textura com relevo. Neste caso, Δ*d* representa a distância entre os *pixels* da textura depois de sofrerem o *warping*. De acordo com a afirmação apresentada, para a posição de câmera *C,* Δ*d* é o maior valor que pode ocorrer de distanciamento entre *pixels* que são vizinhos no impostor original.

5.5 Métrica de Erro baseado no ponto crítico do Impostor com Relevo

Denomina-se ponto crítico de uma textura com relevo o par de *texels* T_1 e T_2 descritos na afirmação acima. Uma das métricas de erro utilizada neste trabalho consiste em prever como é o erro no ponto crítico, antes de realizar o *warping* para imagem completa. Caso o erro previsto esteja acima de um determinado valor, ao invés de ser feito o *warping* o sistema deve requerer a geração de uma nova textura para o impostor com relevo, apropriado para a posição corrente do observador. O erro deste ponto crítico (Δ*d* na figura 5.6), com relação ao deslocamento horizontal, pode ser calculado da seguinte forma:

$$Erro_u = |(u_{f_1} + \Delta u_1) - (u_{f_2} + \Delta u_2)| = |u_{f_1} - u_{f_2} + \Delta u_1 - \Delta u_2| \qquad (5\text{-}5)$$

Como $u_{f_1} - u_{f_2} = 1$, pois são vizinhos, tem-se que:

$$Erro_u = |1 + \Delta u_1 - \Delta u_2|$$

Substituindo-se Δu_1 e Δu_2 pela equação 5-3 chega-se a

$$Erro_u = \left| 1 + \frac{(k_1 - u_{f_1} k_3)\partial(u_1, v_1)}{1 + k_3 \partial(u_1, v_1)} - \frac{(k_1 - u_{s_2} k_3)\partial(u_2, v_2)}{1 + k_3 \partial(u_2, v_2)} \right|$$

Ou, em termos da equação 5-2:

$$Erro_u = \left| 1 + \frac{\partial(u_1, v_1) B_{u_1}}{h + \partial(u_1, v_1)} - \frac{\partial(u_2, v_2) B_{u_2}}{h + \partial(u_2, v_2)} \right| = \left| \frac{h(K + \Delta h(u_e - u_{f_1}))}{(K - \Delta h)K} \right|$$

Onde $K = h + \partial(u_1, v_1)$. Isto permitiria armazenar em cada texel o maior valor de Δh entre este texel e todos os seus vizinhos.

Como $B_u = u_e - u_f$, pode escrever-se que

$$Erro_u = \left| 1 + \frac{\partial(u_1, v_1)(u_e - u_{f_1})}{h + \partial(u_1, v_1)} - \frac{\partial(u_2, v_2)(u_e - u_{f2})}{h + \partial(u_2, v_2)} \right|$$

Até este ponto assume-se apenas um movimento horizontal. De forma semelhante se pode chegar à seguinte equação:

$$Erro_v = \left| 1 + \frac{\partial(u_1, v_1)(v_e - v_{f_1})}{h + \partial(u_1, v_1)} - \frac{\partial(u_2, v_2)(v_e - v_{f2})}{h + \partial(u_2, v_2)} \right| \quad (5\text{-}6)$$

Finalmente, a previsão do erro para o ponto crítico da textura com relevo é calculado da seguinte forma:

$$Erro = \sqrt{Erro_u^2 + Erro_v^2}$$

Porém, como este erro é apenas usado de forma comparativa, economiza-se uma operação de extração de raiz fazendo-se

$$Erro \cong Erro_u^2 + Erro_v^2 \quad (5\text{-}7)$$

Esta estimativa de erro é adequada para impostores que representam objetos com superfície contínua, onde o erro do ponto crítico não é muito maior do que o erro médio de todos os pontos.

5.6 Métrica de Erro baseado em amostragem de pontos críticos

Caso o objeto possua alguma descontinuidade nos valores de profundidade dos *texels*, o método proposto sobreestima o erro de toda a textura, dando-lhe um tempo de vida muito curto apenas por causa de uma região. É conveniente para estes casos não colocar todo o sistema dependente de apenas um ponto crítico, mas sim de um conjunto destes.

O método proposto sugere que a textura com relevo seja dividida em *N* x *M* regiões. Para cada uma destas regiões existe um ponto crítico baseado nos mesmos critérios da seção 5.5, conforme ilustra a figura 5.7.

O critério para se gerar uma nova textura com relevo para o impostor será o de que a somatória do erro de cada ponto crítico ultrapasse um valor de erro estipulado.

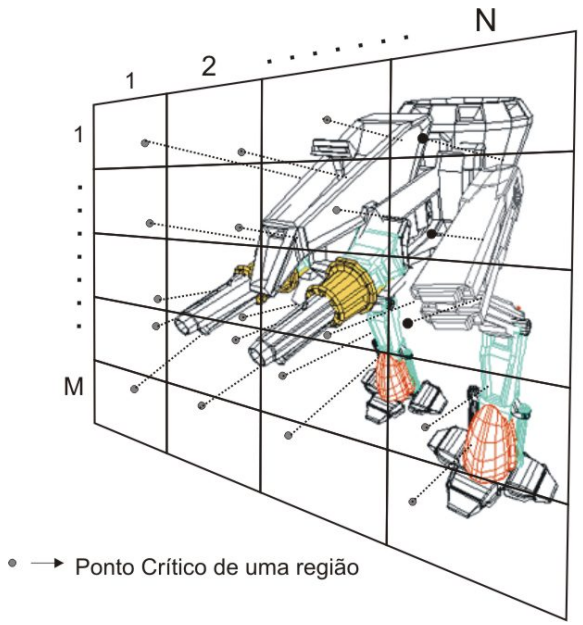

figura 5.7 – Nesta medida de erro uma textura com relevo é dividida em *N* x *M* regiões, tomando-se um ponto crítico para cada região.

5.7 Discussão

A importância da métrica consiste sobretudo em garantir que a textura com relevo dure o máximo possível, sem prejudicar demasiadamente o resultado da visualização. Quanto mais tempo durar a textura, melhor. Este tempo economizado pode ser utilizado para processar as componentes geométricas da cena com mais detalhes, dedicar mais tempo a uma visualização correta do impostor através de *software* e eventualmente realizar a previsão das futuras texturas com relevo com maior precisão. No próximo capítulo é apresentado como se encaixa a GPU no processo de cálculo e visualização destes objetos.