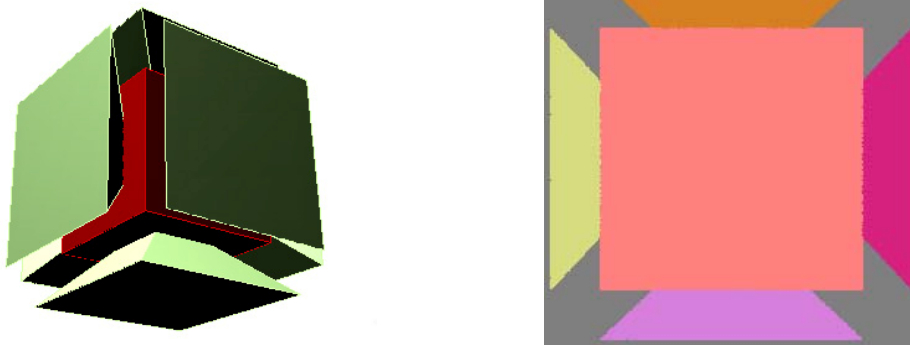


4 Impostores com Relevo

*I just wish... I wish I knew what I'm supposed to do.
That's all. I just wish I knew. (The Matrix)*

4.1 Introdução

Os objetos modelados por texturas com relevo, por corrigirem o problema de paralaxe, possuem um tempo de vida extremamente maior do que objetos modelados por simples *sprites*. Entretanto, como foi visto na seção 3.4, apenas uma textura não é suficiente para representar um objeto por inteiro. Ao representar um objeto através de um paralelepípedo de texturas com relevo, conforme descrito em 3.4.3, tem-se uma limitação do tipo de objetos possíveis de serem representados. A figura 4.1 mostra um exemplo de topologia extrapolada e incorreta para tal método.



(a)

(b)

figura 4.1 – A parte vermelha do objeto representado em (a) não pode ser vista utilizando o método proposto em 3.4.3. Todas as 6 texturas com relevo são iguais ao mapa de normais e profundidade representado em (b).

De forma geral, partes de objetos que não são visíveis por nenhuma das 6 vistas ortogonais, desaparecerão na sua representação por texturas com relevo. Tal caso se dará com frequência em objetos com características côncavas. A figura 4.2 (d) mostra um personagem com os polígonos que não são alcançados através deste método.

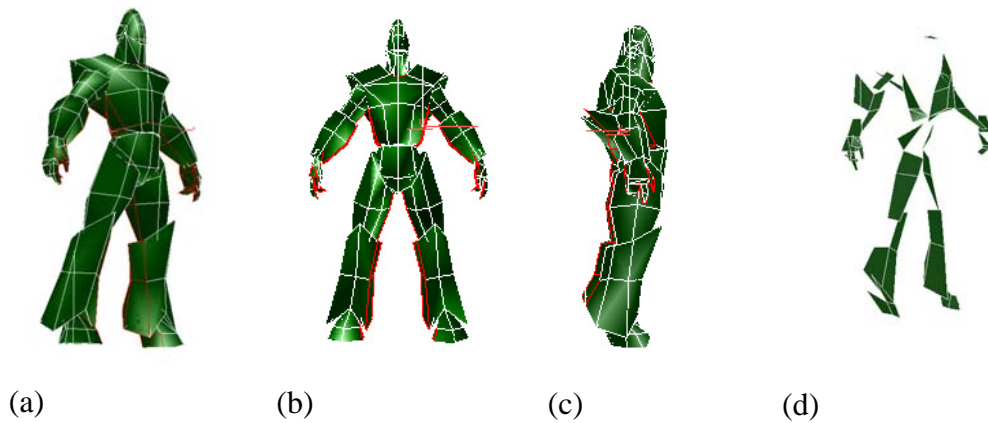


figura 4.2 – Nas vistas ortogonais muitos dos polígonos do objeto não podem ser vistos e são incorretamente interpolados pelo mapeamento de texturas com relevo padrão para uma posição de câmera, como a de (a). Em (d) mostra-se quais são estes polígonos: de um total de 768 faces, 107 estão dentro deste grupo.

A idéia inicial dos *impostores com relevo* consiste em representar um objeto através da textura com relevo correta, para a posição corrente do observador. Para cada movimento de câmera ou do objeto, verifica-se se esta textura ainda é válida. Para fazer este teste, desenvolve-se uma métrica, no presente trabalho, onde se estima o erro a partir da quantidade de *texels* que se devem interpolar, escolhendo-se para isto um *texel* especial, denominado de ponto crítico. Esta métrica está detalhadamente discutida no capítulo 5 desta tese. Diferentemente da modelagem apresentada em 3.4.3, o objeto é representado por apenas um polígono e uma textura com relevo para cada instante.

Quando se verifica que uma textura com relevo está prestes a se tornar obsoleta, o sistema proposto gera uma nova textura, tendo em conta os dados correntes da câmera e do objeto, procurando por outro lado não prejudicar a visualização em andamento dos demais elementos da cena.

A criação destas novas texturas pode ser feita por *software* (CPU) ou por *hardware* (GPU). Gerá-la por *software* pode ser conveniente para sintetizar objetos em tempo real com algoritmos pouco adequados para a GPU ou para explorar e aproveitar melhor o tempo livre existente na CPU. Gerar as texturas com relevo por GPU pode servir para se fazer uma abordagem de renderização distribuída para placas gráficas.

Atualizar as texturas com relevo equivale a calcular um novo mapa de profundidade, um mapa de normais e um mapa de textura, dizendo a cor de cada *texel*. Na seção 8.8 discute-se com mais detalhes como isto é implementado.

Excetuando os casos de objetos com transparência, este método não possui restrições quanto ao tipo de objeto geométrico capaz de ser representado. Entretanto, modelos com geometria complexa fazem com que o tempo de validade da textura seja menor do que em objetos com topologia mais simples (ver figura 8.11). Este fato intrigante está explicado na seção 8.8. A figura 4.3 mostra diversas regiões onde, para cada uma, um único impostor é suficiente para representar o objeto.

No caso de se fazer o cálculo da nova textura com relevo por *software*, pode-se utilizar modelos de visualização impossíveis de serem implementados por *hardware*, tais como o ray-tracing, que exige recursão na sua implementação, ou volume-rendering, que exige um volume grande de dados. Também pode servir para tratar objetos cuja natureza geométrica seja inadequada para a GPU que se dispõe (objetos volumétricos, por exemplo).



figura 4.3 – Cada cor representa uma região onde uma mesma textura com relevo pode ser utilizada. Através da ilustração, pode-se notar que o tempo de vida de cada textura deste tipo é muito maior que o tempo de vida de um *sprite*.

Quanto à classificação apresentada na seção 2.3, pode-se dizer a respeito aos impostores com relevo que:

- a) A autoria que os impostores requerem é equivalente às dos demais elementos geométricos da cena, já que o usuário não interfere no processo de criação e atualização das texturas com relevo. Pode-se, no entanto, permitir que alguns parâmetros sejam controlados: resolução máxima da textura, condição para que um objeto seja transformado num impostor com relevo, constante de erro tolerada pela métrica, algoritmo de visualização por software a ser usado para o objeto, etc.
- b) Os graus de liberdade para objetos deste tipo correspondem a cinco, ou seja, permite-se um movimento livre pela cena, bem como uma variação do vetor de azimute e elevação do observador em relação ao mesmo.
- c) A mudança da posição do observador é contínua para qualquer região do espaço, exceto para o caso do observador entrar na *bounding box* do elemento. Neste caso é necessário tratá-lo como geometria e não mais como textura com relevo.
- d) A quantidade de amostras para visualizar o objeto corretamente se resume a uma textura para cada instante. Entretanto, conforme se apresenta no capítulo 8, havendo um gerenciamento de *cache* de texturas com relevo ou um sistema de previsão para gerar texturas futuras, mais de uma textura com relevo pode estar presente na memória.
- e) Pode-se dizer que os impostores com relevo são do tipo *View Dependent Texture Mapping* e portanto estão dentro da área de modelagem baseada em imagens (*image-based modeling*). No entanto, a extração dos dados geométricos é feita de forma automática e transparente para o usuário, comportando-se, neste caso, como um objeto puramente imagem.
- f) A variável tempo pode vir a ser inserida nos impostores com relevo, sendo necessária uma extensão da pesquisa neste sentido. Para tanto, deve-se ter uma sequência de texturas com relevo, que representem o objeto animado. Uma discussão mais detalhada sobre esta possibilidade pode ser encontrada no capítulo 9 e nas páginas 107-108 de Oliveira (2000a).

4.1.1 Multi-resolução para Impostores com relevo

Em algumas situações pode ser conveniente diminuir a resolução original do impostor com relevo, de forma a não ter que calcular *pixels* que são

posteriormente desperdiçados por problemas de amostragem. A figura 4.4 mostra um objeto sendo representado por um impostor que possui uma resolução de 256 x 256, sendo projetado numa tela na resolução de 800 x 600. Para a câmera em questão, o objeto ocupa aproximadamente 120 x 120 *pixels* da tela. Assim sendo, desperdiça-se tempo ao realizar o *warping* para todos os 256² *texels* da textura com relevo original. É conveniente, portanto, reescalar esta textura.

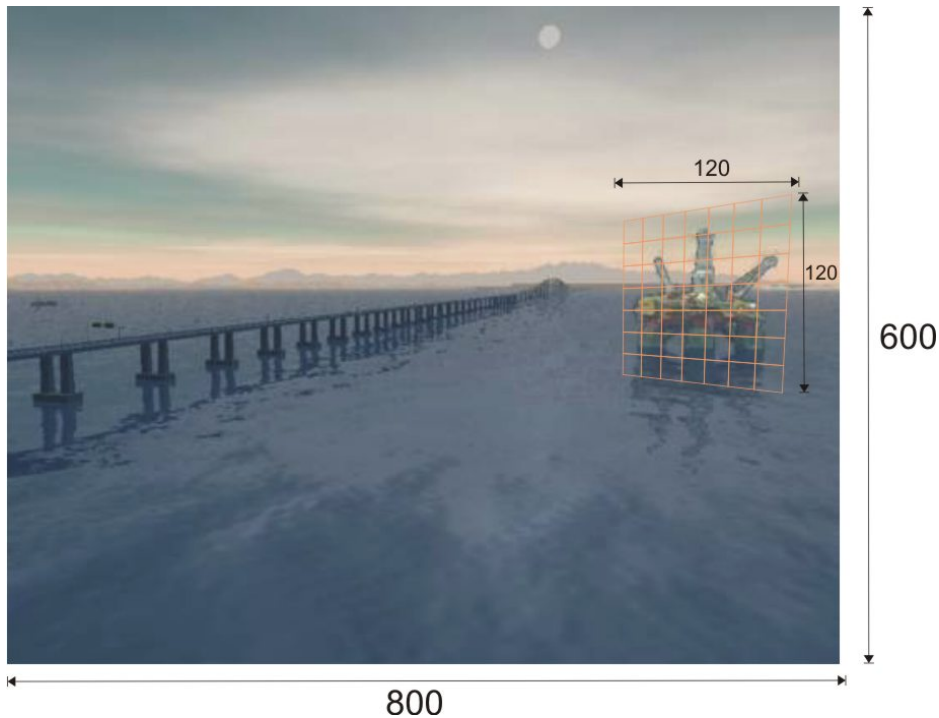


figura 4.4 – Neste exemplo, dos 256² *texels* do impostor, apenas 120² estão sendo aproveitados para a imagem final. Assim sendo, ao se fazer o *warping* da imagem no tamanho original, se está desperdiçando mais de ¾ do tempo de processamento com *texels* que não são vistos.

Para realizar este reescalonamento, é possível testar se um *texel* do impostor se tornou maior que um *pixel* da imagem gerada do observador, realizando-se o seguinte cálculo (ilustrado na figura 4.5):

$$\beta_{\text{impostor}} > K \cdot \beta_{\text{tela}} \quad (4-1)$$

Sendo que

$$\beta_{\text{impostor}} = \frac{FOV}{\text{resolução_impostor}} \quad \text{e} \quad \beta_{\text{tela}} = \frac{FOV}{\text{resolução_tela}} \quad (4-2)$$

A constante K permite que se coloque uma margem de erro, permitindo que o *texel* do impostor possa ser um pouco maior do que o *pixel* da tela, uma vez que a diferença pode ser imperceptível.

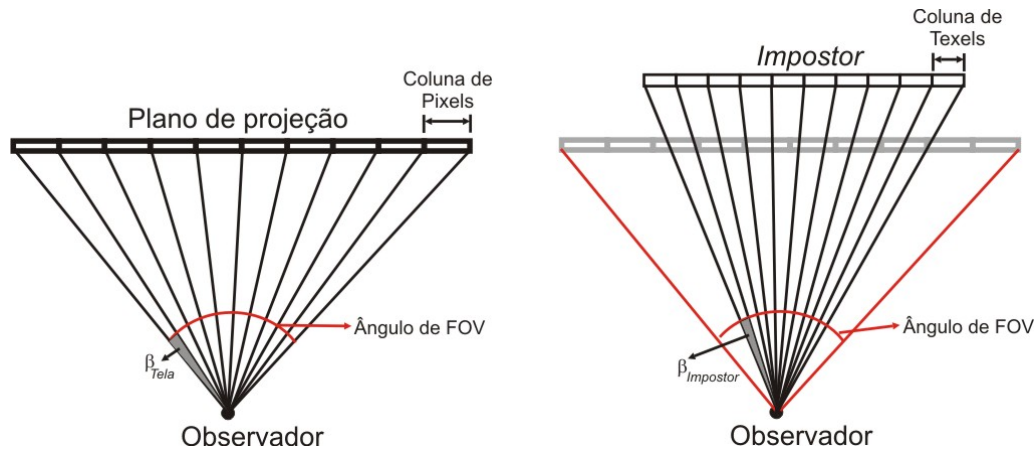


figura 4.5 – $\beta_{impostor}$ deve ser obtido a cada *frame*, para determinar se o impostor se tornou obsoleto em relação à resolução da tela e β_{tela} apenas em caso de alteração do FOV ou da resolução da imagem.

4.2 Discussão

Neste capítulo, após uma breve discussão sobre as possíveis formas de se representar modelos de objetos através de imagens, introduziu-se o conceito de impostores com relevo, como uma extensão na capacidade de modelagem através de texturas com relevo. Este método, diferentemente que a maioria das técnicas de modelagem baseada em imagem e da própria técnica texturas com relevo, apresentada inicialmente por Oliveira (1999), mantém dependência da geometria do objeto. Esta dependência se origina do próprio conceito de impostores, que consiste numa atualização dinâmica da textura. No capítulo seguinte apresenta-se uma forma de medir quando estas atualizações são necessárias. Quanto mais acurada for esta medida, menos texturas devem ser geradas e, portanto, mais eficiente é o impostor com relevo.