

Combinando Textura Com Luzes

por Chris Bentley
Tradução: 3DZone

O que é Mapeamento de Textura?

Mapeamento de textura é um método de colar imagens em objetos poligonais arbitrários. Essa técnica vem se tornando muito popular porque permite que superfícies sejam cobertas com complexos designs, sem precisar de um modelo poligonal complexo. Por exemplo, um simples modelo de um cubo pode ter fotografias do interior de uma sala mapeado em suas faces, criando a ilusão de um modelo arquitetural de alta complexidade.

No mapeamento de texturas, uma imagem é usada essencialmente como uma tabela de consulta: se um pixel da projeção está para ser desenhado, uma consulta é feita na imagem. A cor que estiver na imagem na posição consultada, torna-se a cor que será usada no modelo. A cor descoberta checando-se a imagem pode ser usada sem modificações, ou pode ser misturada com a cor da iluminação do modelo. Também é comum a utilização de características da imagem para modular os parâmetros das equações que calculam a iluminação. Por exemplo, a normal da superfície em um ponto do modelo pode ser alterada devido à presença de uma certa característica encontrada na imagem.

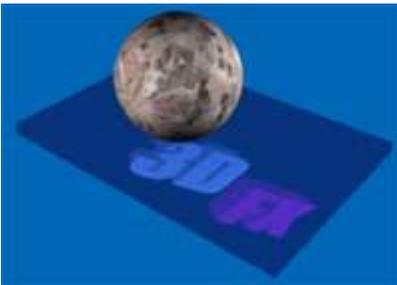


Fig 1. Esfera mapeada com mármore

Coordenadas da Textura

Na texturização uma associação é formada entre os vértices do modelo 3D e os pixels da imagem. A imagem é vista como se estivesse limitada entre as coordenadas, (u,v) , que vão de 0.0 à 1.0 . O parâmetro u é normalmente a abscissa da imagem, e o v , é a ordenada da imagem. Em outras palavras: $0.0 \leq u \leq 1.0$, e $0.0 \leq v \leq 1.0$, e um valor (u,v) de $(0.5,0.5)$ corresponde à meia distância horizontal e meia distância vertical, o que chamamos de centro. O mapeamento de texturas, então, é um mapeamento 3D para imagens 2D: $O(x,y,z) \rightarrow T(u,v)$.

Os Passos do Mapeamento de Textura

Há dois passos seguidos pela Mapeamento de Textura:

1. As coordenadas **(u,v)** são associadas à cada vértice do modelo. Esses parâmetros "grudam" a imagem ao objeto.
2. Uma vez que os vértices têm coordenandas **(u,v)** é necessário que seja interpolados por pontos no interior dos polígonos.

Designando Coordenadas **(u,v)** aos Vértices

Existem vários métodos, descritos por Watt [WATT92], de associar coordenandas **(u,v)** para cada vértice. Um método simples é o chamado "mapeamento por latitude/longitude". Outro método designa valores **(u,v)** no momento em que os polígonos do objetos são construídos. Nessa técnica, o conhecimento da forma do objeto pode ser utilizado para se designar parâmetros mais inteligentes ao mapeamento.

No mapeamento por latitude/longitude, a normal da superfície de cada vértice é usada para calcular valores **(u,v)** para esse vértice. A normal é um vetor de três componentes: $\mathbf{N} = (x,y,z)$. O problema é representar essa trinca na forma de um vetor de apenas dois componentes. A resposta é converter as coordenadas cartesianas 3D em coordenadas esféricas, como em latitude e longitude. Coordenadas esféricas podem expressar qualquer ponto em uma esfera através de dois ângulos. O ângulo da longitude dá a localização horizontal e o da latitude a vertical.

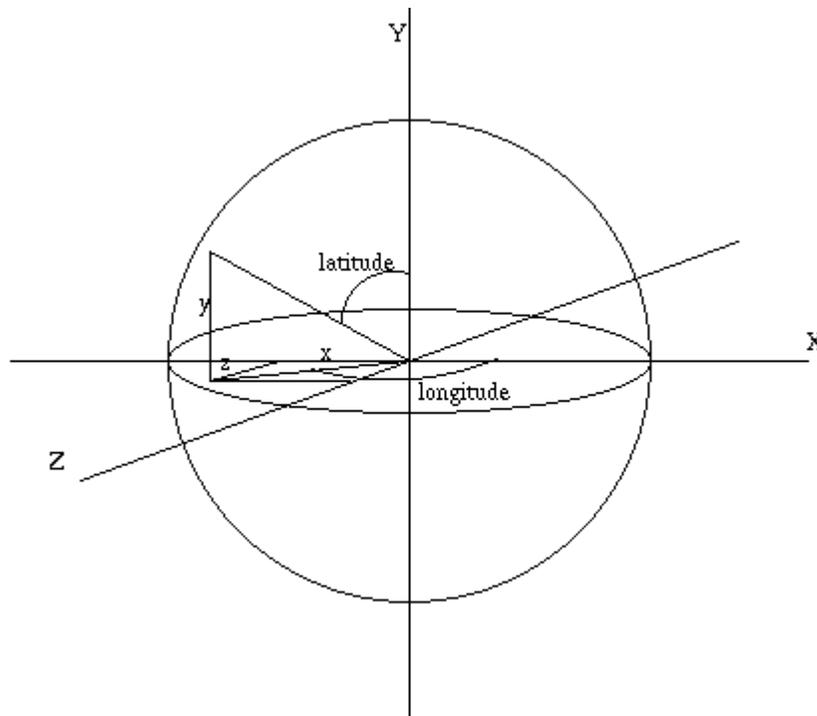


Fig 2.Mapeamento por Latitude/Longitude

Ao imaginarmos a normal como um ponto específico dessa esfera podemos converter cada normal $\mathbf{N} = (x,y,z)$, para coordenadas esféricas, e então usar os valores (u,v) de latitude/longitude para fazer o mapeamento.

Cada ponto \mathbf{P} na esfera, pode sr expresso em coordenadas esféricas como a seguir:

```
longitude = arctan( z/x );  
latitude = arccos( y );
```

Por exemplo: um valor u de **0.3** seria traduzido como um pixel da imagem localizado na posição correspondente a 30% da distância horizontal da imagem.

Implementando o mapeamento por Latitude/Longitude

```
/*  
*****  
*/  
/* norm_uv - convert normal vector to u, v coordinates */  
/*  
*****  
*/  
static void norm_uv( norm, u, v )  
VECT norm;  
double *u;  
double *v;  
{  
    *u = (FLT_ZERO(norm[0])) ? 0.5 : atan2pi(norm[2], norm[0]);  
    *v = (asinpi(norm[1]) + 0.5);  
}  
  
/*  
*****  
*/  
/* map - find image color for a given u, v pair */  
/*  
*****  
*/  
static unsigned char map( u, v, image )  
double u;  
double v;  
IMAGE *image;  
{  
    unsigned int x, y, index;  
  
    x = (unsigned int)(u * image->width);  
    y = (unsigned int)(v * image->height);  
  
    index = (y * image->width) + x;  
  
    return( image->data[index] );  
}
```

}

Interpolando Valores (u,v) Para Pixels Sem Polígono

Uma vez que os vértices de um modelo tiveram seus valores (u,v) designados, ainda resta a tarefa de interpolar essas coordenadas para derivar valores (u,v) para pixels no interior do polígono. Isso pode ser feito usando interpolação bilinear, que é muito parecido com a interpolação de valores de intensidade usados no sombreamento Gouraud. Esse método sofre do mesmo problema dos algoritmos usados para o scanearamento das linhas para gerar o sombreamento em geral: as coordenadas (u,v) são interpoladas uniformemente no espaço da tela. Porém, por causa da distorção da perspectiva, o ponto central entre dois vértices na tela **não** é realmente o ponto central entre esses dois vértices! Portanto, a interpolação uniforme na verdade não é correta.

No sombreamento Gouraud, quando uma linha é renderizada, somente os valores z e a **intensidade** são interpolados, e ao longo da linha y os valores são constantes. No mapeamento de textura, quando uma linha é renderizada, ambos u e v precisam ser implementados para cada pixel, pois em geral cada linha horizontal do polígono projetado será transportada caminho diagonal através da textura. Como pode ser visto na figura 3 abaixo, no momento em que cada linha é renderizada pixel por pixel, ambos u e v deverão ser interpolados.

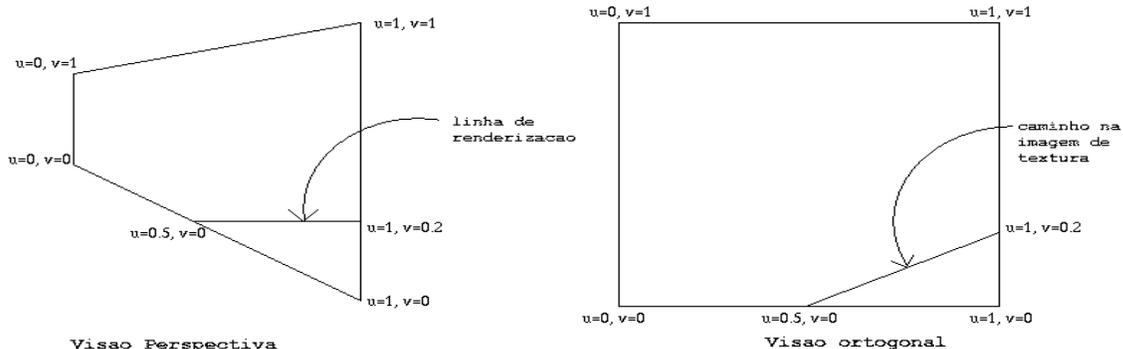


Fig 3. Linha de textura renderizada

Combinando Mapeamento de Textura com Iluminação

O mapeamento de textura pode "colar" uma imagem em qualquer objeto com faces. A imagem de textura é usada para consulta de cores que serão aplicadas no objeto na renderização. Porém, também é possível misturar a cor da textura com a cor calculada através das equações de iluminação. Similarmente, a cor da textura pode ser usada para modular os componentes ambient, diffuse ou specular do reflexo dos objetos. A figura 4 abaixo mostra a cor da textura misturada com a iluminação, com um valor igual a **0.4** para a cor da textura, enquanto que a figura 5 dá à textura um valor de **0.8**.



Fig 4. Misturando cores da textura e iluminação



Fig 5. Cores da textura com um valor mais alto

Conclusão

Existem várias áreas de mapeamento de textura avançado que não são explorados aqui: **mip-mapping** e **area sampling** para reduzir o **anti-aliasing**; **perspective correction**; e outras técnicas de mapeamento. O Mapeamento de textura é um método poderoso para se obter efeitos foto-realísticos em imagens geradas por computador, sem a necessidade de se criar modelos poligonais muito complexos.



Fig 6. Um Teapot mapeado com mármore.

Referências

[WATT92]

Watt, Alan, and Watt, Mark, "Advanced Animation and Rendering Techniques", Addison Wesley, 1992