

Capítulo 12

Textura

Embora seja intuitivamente fácil reconhecer texturas numa imagem, é difícil definir uma textura de forma precisa. Além de depender do contexto em que ela é utilizada, a noção de textura é, de certa maneira, relacionada com o nível de detalhes dos objetos numa imagem. Uma superfície rugosa pode aparecer numa imagem com várias elevações e depressões; e numa outra imagem, com veios claros e escuros, aparentando a textura de madeira.

Podemos agrupar algumas definições de textura encontradas na literatura em dois contextos:

no contexto de processamento de imagens: é uma representação que facilita a **identificação** das propriedades geométricas das figuras contidas numa imagem.

- “Variação local da luminância/brilhância entre os *pixels* dentro de uma região pequena da imagem”.
- “Medidas das propriedades da superfície de um objeto”.
- “Consiste de um conjunto de elementos mutuamente relacionados”.

no contexto de computação gráfica: é uma representação das **alterações** que devem ser feitas nos *pixels* das imagens.

- “Consiste de detalhes de uma superfície que podem ser pré-calculados, armazenados em arranjos multidimensionais, e utilizados durante o processo de síntese de imagens”.
- “Consiste de um conjunto de elementos que afeta os atributos de cada *pixel* de uma imagem, com a finalidade de aumentar o realismo sem onerar o procedimento de síntese”.

O processo de identificação pode ser reduzido em particionar a imagem em regiões com propriedades similares usando medidas **estatísticas**, **estruturais**, **espectrais** ou combinação das três. Neste caso, uma textura pode ser representada por uma imagem, por uma representação gráfica da distribuição de frequências de ocorrência dos valores de uma determinada propriedade da imagem, ou por uma sequência de descritores relacionais que associam os elementos primitivos. Vimos, por exemplo, na seção 9.4.1 que a magnitude de uma transformada de Fourier da luminância/brilhância de uma imagem indica a “quantidade” da componente de frequência e o ângulo de fase indica “onde” se encontra tal componente na imagem.

Como modificadores de atributos de cada *pixel* de uma imagem, e conseqüentemente da aparência visual de uma imagem, a textura é modelada como um arranjo multidimensional de *texels* (*texture elements*), pré-computados de forma independente do observador e capazes de melhorar o realismo de uma imagem a um custo computacional adicional relativamente pequeno. Para aumentar a sua reusabilidade, uma textura é modelada num espaço próprio, conhecido como **espaço de textura**. O processo de **texturização** requer, então, o **mapeamento** entre os *pixels* da imagem e os *texels* de um (mapa de) textura.

Mantendo-nos na sequência de **síntese de imagens** a partir de um conjunto de modelos, consideramos neste Capítulo a textura como um arranjo de *texels* capazes de “perturbar”, entre uma infinidade de propriedades, a cor, a especularidade, a normal, a posição do(s) ponto(s) projetado(s) no *pixel*. Veremos na seção 12.1 como se pode modelar uma textura. Em seguida, na seção 12.2 apresentamos algumas técnicas de mapeamento de textura mais conhecidas para obter distintos efeitos visuais. Quando a textura é uma imagem, temos os mesmos problemas que destacamos na seção 11.2: cada *pixel* pode ter uma pré-imagem distinta, podendo resultar em efeitos visuais indesejáveis. Para contornar isso, Williams propôs em 1983 uma técnica de *anti-alias* especificamente para texturização, que detalharemos na seção 12.3.

12.1 Textura

Uma textura é representada num **espaço homogêneo** próprio, conhecido como **espaço de textura**. A dimensão deste espaço pode ser uni-, bi- ou tridimensional, com uso de até quatro **coordenadas homogêneas** s , t , p e q para denotar os pontos deste espaço. Uma textura pode ser representada como um objeto geométrico neste espaço e é passível a todas as

transformações geométricas que vimos no Capítulo 3.

Atualmente, as texturas mais difundidas são as bi-dimensionais. Isso decorre da disponibilidade de uma grande diversidade de dispositivos que permitem capturar imagens de alta resolução, como câmaras fotográficas, vídeos e escaneadores, e de aplicativos para manipular estas imagens, como *gimps*, *photoshop* e *coreldraw*. O valor de cada *texel* é dependente da aplicação.

12.1.1 Imagens

Na grande maioria das aplicações, as texturas são simplesmente imagens sintéticas ou captadas por algum dispositivo. Elas são utilizadas como **decalques**, substituindo integralmente os valores dos *pixels* correspondentes. Estas texturas tem comumente as suas coordenadas normalizadas, ou seja, as coordenadas dos seus quatro vértices são, respectivamente, $(0, 0)$, $(1, 0)$, $(1, 1)$ e $(0, 1)$.

Exercício 12.1 Defina um sistema de referência (de textura) normalizado para as imagens



Marque nas imagens os pontos de coordenadas $(0.25, 0.25)$, $(0.25, 0.75)$, $(0.5, 0.5)$ e $(0.75, 0.75)$.

Outro uso de imagens é criar o efeito de reflexão de um chão bem lustrado ou a de um espelho. Neste caso, podemos utilizar a imagem dos objetos a serem refletidos como textura $C(s, t)$ para **compô-la** (*blend*), de forma ponderada, com os valores $I(u, v)$ originais dos *pixels* correspondentes.

(Ver Fig. 16.38 do livro-texto Foley.)

12.1.2 Textura de Sombra

Os valores $C(s, t)$ de uma textura podem ser utilizados para **modular** os valores originais $I(u, v)$ dos *pixels*, como a **textura de sombra**, cujo *texel* representa a percentagem de iluminação

$$I_n(u, v) = I(u, v)C(s, t).$$

Assim, os *pixels* mapeados nos *texels* de valor elevado terão cores finais claras e os mapeado em *texels* de intensidade menor serão mais escuros. Uma textura de sombra pode ser obtida imageando a cena com o observador localizado na fonte luminosa.

(Ver Fig. 16.33 do livro-texto Foley.)

Exercício 12.2 *Seja uma cena composta por um cubo unitário centrado em $(2, 2, 1, 1)$, uma esfera de raio unitário centrada em $(4, 0, 1, 1)$ e uma fonte do tipo spot em $(0, 4, 1, 1)$ direcionada para o cubo. Defina uma imagem de sombra.*

12.1.3 Textura de Perturbação de Normais

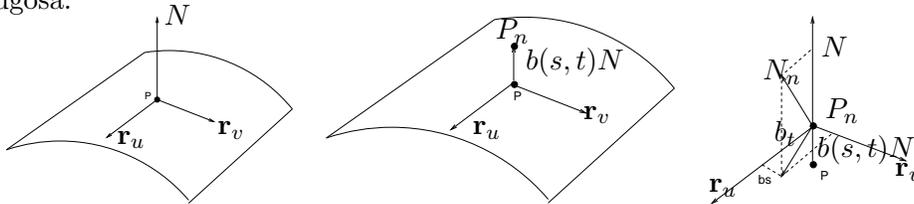
Uma textura bi-dimensional pode ser também definida proceduralmente, como a **textura de perturbação de normais**, ou *bump mapping* proposto originalmente pelo J. Blinn. Esta textura teve como principal motivação reduzir a aparência “plástica” das imagens geradas com o modelo de tonalização de Phong (seção 8.5). A cada ponto (s, t) do mapa de textura é associado um ruído $b(s, t)$, de forma que um ponto $P = \mathbf{r}(x(u, v), y(u, v), z(u, v))$, com normal igual a N , seja deslocado para

$$P_n = P + \frac{b(s, t)N}{|N|}.$$

Blinn mostrou que uma boa aproximação para o vetor normal N_n do ponto deslocado seria

$$N_n = N + \frac{b_s(N \times \mathbf{r}_u) - b_t(N \times \mathbf{r}_v)}{|N|},$$

onde b_s e b_t são derivadas parciais da função de ruído $b(s, t)$ e \mathbf{r}_u e \mathbf{r}_v são as derivadas parciais da superfícies \mathbf{r} . A função $b(s, t)$ controla a aparência rugosa.



Observação 12.1 *As derivadas podem ser aproximadas por diferenças finitas na implementação. Por exemplo,*

$$b_s = \frac{b(s + \Delta s, t) - b(s, t)}{(s + \Delta s) - s}$$

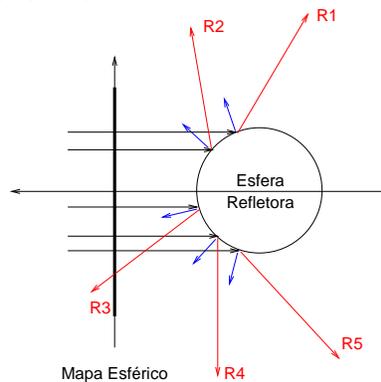
Exercício 12.3 Sejam $b(s, t) = \sin(5s)/5$ um mapa de textura de perturbação das normais e $\mathbf{r}(u, v) = u(1, 0, 0, 0) + v(1, 1, 1, 0)$ um plano. Determine a posição do ponto $\mathbf{r}(1, 1)$ e a normal da superfície neste ponto após a texturização, supondo que o ponto $\mathbf{r}(1, 1)$ seja mapeado ao ponto $(2, 1)$ do mapa de textura.

Exercício 12.4 A textura de perturbação de vetores normais afeta as imagens geradas pelo modelo de tonalização de Gouraud? Justifique.

Observação 12.2 As funções de ruído podem ser utilizadas também para perturbar diretamente nas cores dos pixels. O procedimento mais famoso é o proposto pelo Perlin que permite gerar texturas com aparência de pêlos de animais e de mármore. Conferir no sítio: <http://mrl.nyu.edu/~perlin/doc/oscar.html>;

12.1.4 Mapa Esférico

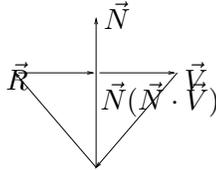
É muito utilizada para **mapeamento de reflexões especulares** (*environment mapping*), com a finalidade de produzir os efeitos de reflexão especular que superfícies metálicas bem polidas apresentam. Ele poderia ser a foto de uma esfera espelhada ou uma foto obtida com uma câmera de lente de peixe. No primeiro caso, a câmera apareceria na foto e no segundo caso, os efeitos são aproximados, já que não há ainda uma lente com um campo de visão de 360° . Uma terceira alternativa seria gerar um mapa esférico, a cada *texel* (s, t) do qual são associado os “atributos” (totalmente) refletidos no ponto $P = (x, y, \sqrt{1 - x^2 - y^2})$ de uma esfera de raio unitário centrada na origem, ou seja, os atributos gráficos do objeto da cena que estiver na linha do raio de reflexão R correspondente.



Como as coordenadas do vetor normal \vec{N} em P coincidem com as coordenadas de posição, podemos substituí-las na seguinte expressão para de-

terminar a direção do raio de reflexão $\vec{R} = (R_x, R_y, R_z)$

$$\vec{R} = \vec{V} - 2\vec{N}(\vec{N} \cdot \vec{V}). \quad (12.1)$$



Observe que com este procedimento é gerado um mapa de *texels* no domínio $[-1, 1] \times [-1, 1]$. Para passar para o espaço de textura usual, são aplicadas ainda as seguintes transformações de coordenadas do espaço (x, y) para o espaço (s, t) :

$$\begin{aligned} s &= \frac{x + 1}{2} \\ t &= \frac{y + 1}{2} \end{aligned} \quad (12.2)$$

12.1.5 Mapa Cúbico

É uma alternativa para mapa esférico. Ao invés de gerar uma imagem com um campo de visão de 360° , um mapa cúbico é constituído por seis imagens/mapas obtidas através de 6 distintos ângulos a partir de um mesmo ponto do ambiente. Estas imagens correspondem às vistas que teríamos, olhando para a direção das quatro paredes, do teto e do chão de um cômodo em forma de um cubo. Cada uma dessas imagens é denominado **mapa de ambiente**.

Observação 12.3 Vale a pena dar uma olhada a história da evolução de *environment mapping* no sítio <http://www.debevec.org/ReflectionMapping/>

12.2 Texturização

Na seção 12.1 vimos que os novos valores dos *pixels* (u, v) dependem dos valores nos *texels* (s, t) correspondentes. Como os dois objetos são representados em espaços distintos, é necessário determinar a correspondência T entre estes dois espaços – daí o uso do termo *texture map*. Nós optamos pela palavra **texturização** para referir esta correspondência. Existe uma variedade de texturizações. A escolha por uma particular técnica depende basicamente dos recursos computacionais e da qualidade de imagem desejada.

Essencialmente, há dois paradigmas: **método direto** e **método inverso**. O método direto é orientado ao espaço de textura. Cada *texel* sofre contrações ou esticamentos, como uma lâmina de borracha, para cobrir os *pixels*. É similar à operação *warping*, que apresentamos na seção 11.2. O método inverso é orientado ao espaço do dispositivo de saída. Ele pode ser prontamente integrado aos algoritmos de varredura e de *z-buffer*, para constituir um eficiente procedimento de síntese de imagens; portanto, é o mais utilizado na prática. A cada *pixel*, é adicionalmente computada a sua **pré-imagem** no espaço de textura e integrada a informação contida nele para determinar a luminância/brilhância final do *pixel*.

Nesta seção apresentamos duas correspondências utilizadas no contexto de texturização.

12.2.1 Função Linear

As coordenadas dos espaços de textura ($s/q, t/q$) são transformadas em coordenadas ($u/w, v/w$) do espaço de dispositivo por uma transformação linear

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} a & b & c \\ e & f & g \\ i & j & k \end{bmatrix} \begin{bmatrix} s \\ t \\ q \end{bmatrix}.$$

Se três pares de correspondências forem especificadas, podemos determinar, com uso de técnicas clássicas de solução de equações lineares, todos os elementos da matriz de transformação inversa e utilizá-la para calcular a pré-imagem de cada *pixel*.

Uma alternativa mais prática é adotar o mesmo esquema de interpolação bilinear que comentamos nas seções 9.3.3 e 10.4 para computar recorrentemente as coordenadas homogêneas de textura.

Exercício 12.5 *Derive a expressão recorrente para computar as coordenadas de textura associadas a cada pixel em uma linha de varredura.*

Exercício 12.6 *Rasterize, com uso do algoritmo de varredura por linha e a tonalização de Gouraud, os dois polígonos especificados pelas coordenadas dos seus vértices no espaço do dispositivo*

1. $(0, 0, 0.4)$, $(6, 0, 0.4)$ e $(3, 8, 0.8)$
2. $(2, 2, 0.6)$, $(8, 1, 0.2)$ e $(5, 7, 0.2)$.

As coordenadas de textura associadas aos vértices são, respectivamente, $(0, 0, 1)$, $(1, 0, 1)$ e $(1, 1, 1)$ para ambas as facetas.

Exercício 12.7 Considere a mesma cena descrita no Exercício 12.2 e um observador posicionado em $(2, 2, 5, 1)$ focando no centro da esfera. Derive a correspondência entre as coordenadas de um pixel (u, v) com as coordenadas (s, t) do mapa de sombra.

12.2.2 Função Esférica

O objetivo desta função é determinar as coordenadas (s, t) do *texel* que deve ser mapeado no ponto $P = (x, y, z)$ do espaço do objeto, para obter efeitos realistas em superfícies altamente refletoras. Esta função pode ser derivada com uso da Eq. 12.1.

Assumindo $\vec{V} = (0, 0, -1)$ e $\vec{N} = (x, y, \sqrt{1 - x^2 - y^2})$, temos

$$\begin{bmatrix} R_x \\ R_y \\ R_z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} - 2 \left(\begin{bmatrix} x \\ y \\ \sqrt{1 - x^2 - y^2} \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} \right) \begin{bmatrix} x \\ y \\ \sqrt{1 - x^2 - y^2} \end{bmatrix}$$

Assim, podemos escrever as coordenadas (x, y, z) em termos do vetor de reflexão R

$$\begin{aligned} x &= \frac{R_x}{2\sqrt{1 - x^2 - y^2}} \\ y &= \frac{R_y}{2\sqrt{1 - x^2 - y^2}} \\ z &= \sqrt{1 - x^2 - y^2} = \frac{R_z + 1}{2\sqrt{1 - x^2 - y^2}} \end{aligned}$$

Como $x^2 + y^2 + z^2 = 1$, essas igualdades podem ser reescritas como

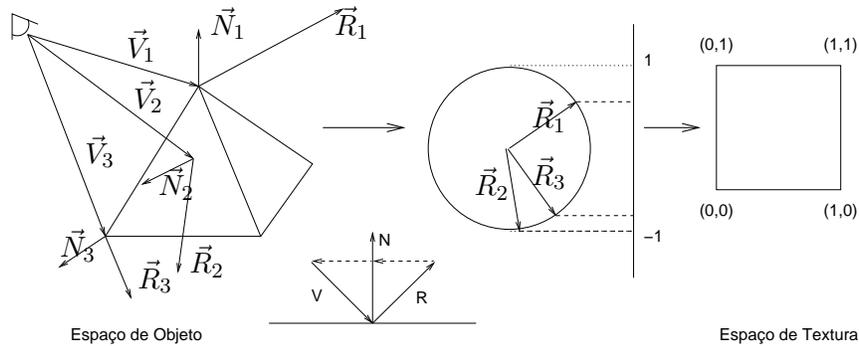
$$\begin{aligned} x &= \frac{R_x}{\sqrt{R_x^2 + R_y^2 + (R_z + 1)^2}} \\ y &= \frac{R_y}{\sqrt{R_x^2 + R_y^2 + (R_z + 1)^2}} \\ z &= \sqrt{1 - x^2 - y^2} = \frac{R_z + 1}{\sqrt{R_x^2 + R_y^2 + (R_z + 1)^2}} \end{aligned}$$

Com uso da Eq. 12.2, chegamos a seguinte relação entre as coordenadas (s, t) do espaço de textura e o raio refletor R em cada ponto (x, y, z) no espaço de objeto

$$s = \frac{R_x}{2\sqrt{R_x^2 + R_y^2 + (R_z + 1)^2}} + \frac{1}{2}$$

$$t = \frac{R_y}{2\sqrt{R_x^2 + R_y^2 + (R_z + 1)^2}} + \frac{1}{2}$$

$$q = 1$$



Exercício 12.8 O vetor normal em $(x, y, z, 1)$ sobre uma esfera centrada na origem é $(x, y, z, 0)$. Qual é o vetor normal em cada ponto sobre uma esfera centrada em $(0, 0, -1, 1)$?

Exercício 12.9 Determine as coordenadas dos pontos do espaço de textura de ambiente correspondentes aos vértices de um cubo unitário com as faces paralelas aos planos xy , yz e xz centrado em $z = -5$ com o observador na origem e o eixo óptico coincidente com o eixo z .

Exercício 12.10 A textura de ambiente para environment mapping é dependente do observador? Ou seja, se for alterada a posição do observador, podemos utilizar ainda o mesmo mapa? Justifique.

12.3 Níveis de Detalhes – Mipmap

O processo de filtragem a ser aplicada numa imagem texturizada para atenuar os artefatos é computacionalmente caro. Milhares de *texels* podem ser mapeados num único *pixel*. Diversas técnicas foram propostas com o objetivo de baratear o processo.

Hoje em dia, a técnica mais popular é a técnica conhecida por *MIP* (*multum in parvo*) que tem como premissa de que a pré-imagem de cada

pixel tenha uma forma quadrada de tamanho igual a uma potência de 2. Ao invés de uma única imagem original $i = 1$, uma sequência de imagens de níveis de resolução monotonicamente decrescente é pré-calculada, de tal sorte que a imagem de nível de resolução $i + 1$ é o resultado da filtragem e subamostragem da imagem de nível de resolução i . Daí o nome sugestivo de **pirâmide de texturas**. Como o tamanho de cada imagem é uma potência de 2, uma memória adicional correspondente a $\frac{1}{3}$ do espaço ocupado pela imagem original é necessário. Observe que filtragens são aplicadas, a priori, em todas as imagens para atenuar as distorções indesejadas. Com isso, durante a rasterização, o complexidade de texturização passa a ser constante.

Durante a rasterização, deve-se selecionar dentre o conjunto de imagens pré-computadas a que tiver uma resolução mais próxima a do *pixel*. William propôs estimar a imagem i a ser utilizada como pré-imagem de um *pixel* através da verificação da variação das dimensões originais ∂s e ∂t da pré-imagem de uma *pixel* $\partial u = \partial v = 1$

$$2^{i-1} = \max\left[\sqrt{\left(\frac{\partial s}{\partial u}\right)^2 + \left(\frac{\partial t}{\partial u}\right)^2}, \sqrt{\left(\frac{\partial s}{\partial v}\right)^2 + \left(\frac{\partial t}{\partial v}\right)^2}\right]$$

(Ver Fig. 17.10 do livro-texto do Foley.)

Exercício 12.11 *Seja uma imagem de 512×512 . Qual seria o nível de resolução do mipmap utilizado, considerando que a pré-imagem de um pixel seja*

1. 4×4 texels
2. 128×128 texels
3. 200×200 texels