

## 7.1 Imagens Discretas

Uma imagem discreta pode ser considerada um reticulado de *pixels*, a cada qual é associada uma luminosidade/brilhança. O *pixel* é a menor unidade entregável por um par de coordenadas inteiras  $[i]^t$ .

Entre as medidas de **distância** entre dois *pixels*  $(i, j)$  e  $(l, m)$ , as mais conhecidas são

**distância Euclideana** :  $D_E((i, j), (l, m)) = \sqrt{(i-l)^2 + (j-m)^2}$ .

**distância de quarteirão** (*city-block distance*):  $D_1((i, j), (l, m)) = |i-l| + |j-m|$ .

**distância de xadrez** (*chessboard distance*):  $D_8((i, j), (l, m)) = \max\{|i-l|, |j-m|\}$

(Ver página 32 do livro-texto de Gonzalez.)

A **conectividade** entre os *pixels* pode ser caracterizada com as distâncias de quarteirão e de xadrez. Dois *pixels*  $p$  e  $q$  são ditos **vizinhos-de-4** se satisfazem  $D_1(p, q) = 1$ . Eles são chamados **vizinhos-de-8** se satisfazem  $D_8(p, q) = 1$ .

(Ver Fig. 2.13 do livro-texto de Gonzalez.)

Dois *pixels* — são **adjacentes** se eles foram conectados. Podemos definir adjacência-de-4 e adjacência-de-8 dependendo do tipo de conectividade especificado.

**Exercício 7.1** Dada uma imagem quadrada

```

3 1 2 1 (a)
2 2 0 1
1 2 1 1

```

```

(p) 1 0 1 2

```

Determine as distâncias  $D_1$  e  $D_8$  entre os *pixels*  $p$  e  $q$ , considerando

1. conectividade de *pixels* com valores 0 e 1.

2. conectividade de *pixels* com valores 1 e 2.

**Exercício 7.2** Dada uma imagem binária

```

0 0 0 0 0 0 0 1 1 0
1 0 0 1 1 0 0 0 1 0
0 1 0 1 0 1 0 1 0 0
1 0 0 1 1 0 0 0 1 1
1 0 0 1 1 0 0 0 1 1
1 1 1 0 1 1 0 1 1 0

```

## Capítulo 7

# Amostragem

Hoje em dia, o problema básico em Computação Gráfica é a síntese de imagens discretas visualizáveis numa tela constituída por um arranjo bidimensional de pontos entregáveis denominados *pixels* (*picture elements*). Portanto, uma imagem contínua  $f(u, v)$  deve ser discretizada em **pontos de amostra** por um **processo de amostragem**. Estes pontos de amostra são usualmente ordenados num plano formando um **reticulado** (*grid*). Na prática, os reticulados mais conhecidos são retangulares e hexagonais. Paralelamente, os reticulados conectados-de-4 são de especial interesse para Computação de Imagens por serem a estrutura da maioria dos dispositivos *rasier*. Na seção 7.1 são introduzidos alguns conceitos relativos imagens discretas.

**Observação 7.1** *Rasters são reticulados com uma relação de adjacência bem definida. A tela de um monitor é um reticulado conectado-de-4.*

Há duas formas para abordar o problema de amostragem em sistemas de informações gráficas. A primeira forma é geométrica (seção 7.2). Ela se ocupa com a conversão de pontos em coordenadas reais de figuras geométricas 2D descritas analiticamente para pontos em coordenadas inteiras. Esta forma é muito utilizada nas funções de saída das primitivas gráficas 2D. A segunda forma é analítica (seção 7.3). Ela trata a imagem como uma integral de Fourier e usa todas as suas propriedades para entender e processar os efeitos visuais. Esta segunda forma nos permite corrigir ou melhorar a qualidade da imagem lançando mão em artifícios fundamentados em Teoria de Sinais, como ilustra os diferentes procedimentos de *antialiasing* na seção 7.4.

Considere a conectividade de pixels com valor 1, as duas regiões  $S_1$  e  $S_2$  são conectadas-de-4? São conectadas-de-8? E em relação à conectividade de pixels com valor 0?

Um **caminho** entre dois pixels  $p$  e  $q$  é uma sequência de pixels distintos  $A_0, A_1, A_2, \dots, A_{n-1}, A_n$ , onde  $A_0 = p$ ,  $A_n = q$  e  $A_i$  é adjacente a  $A_{i+1}$ . Dizemos ainda que  $p$  e  $q$  são **coníguos**. Um **caminho fechado** é um caminho onde o primeiro pixel é igual ao último pixel.

Uma **região** é um conjunto de pixels no qual para qualquer par de pixels da região existe um caminho.

A **borda interna** de uma região é um conjunto de pixels dentro da região que tem um ou mais vizinhos fora da região e a **borda externa** é um conjunto de pixels dentro do complemento da região que tem um ou mais vizinhos dentro da região.

A **aresta** associada a um pixel e a sua vizinhança é um vetor cuja direção é perpendicular à direção de maior variação na luminosidade/brilhança e cuja magnitude é a diferença entre as luminosidades/brilhanças dos dois pixels.

**Exercício 7.3** Dada uma imagem binária

```

0 0 0 0 0 0 0 1 1 0
1 0 0 1 1 0 0 1 1 0
0 1 1 1 1 1 0 1 0 0
1 0 0 1 1 0 1 0 1 1
1 1 1 0 1 1 0 1 1 0

```

Determine as regiões e suas bordas internas e externas considerando

1. conectividade-de-4 de pixels com valor 1 e
2. conectividade-de-8 de pixels com valor 1.

**Observação 7.2** As definições de vizinhança e contiguidade para retículos conectados-de-4, podem levar a certas pontadas como ilustra a seguinte discretização de uma circunferência. O resultado é uma curva conectada-de-8 e não conectada-de-4!



## 7.2 Algoritmos de Conversão

Os primeiros algoritmos de conversão de coordenadas reais para inteiros, conhecidos também como **algoritmos de rasterização** ou *scan-conversion*,

foram desenvolvidos para visualizar as primitivas geométricas analíticas, como pontos, linhas e polígonos, nas telas raster dos dispositivos de exibição. O dispositivo mais difundido é o monitor de vídeo CRT (*cathodic ray tube*).

O monitor CRT consiste de um tubo de raios catódicos com uma tela e um câmbio que produz um ou mais feixes de elétrons controlado por um sistema de focalização. Em cada ponto da tela, também conhecido como *pixel*, há um ou mais tipos de fósforos, que ao serem excitados emitem radiações de ondas espectrais de comprimentos distintos. Para facilitar o endereçamento dos pontos discretos da tela, é definido um sistema de referência de coordenadas inteiras denominado **sistema de coordenadas de dispositivo físico** ou **sistema de coordenadas raster**. Usualmente, a origem deste sistema corresponde ao canto esquerdo superior da tela e cada par de coordenadas entende-se um nó do reticulado que tangencia os pixels.

**Observação 7.3** A relação entre a luminosidade  $L$  num pixel de uma tela CRT e a tensão  $V$  a ser aplicada no feixe de elétrons para excitar os fósforos correspondentes não é linear. Ela depende de um fator  $\gamma$  e de uma constante  $K$  para fazer uma correção  $\gamma$

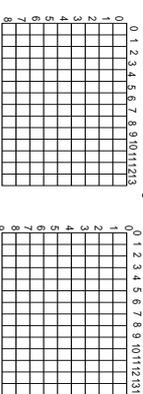
$$L = KV^\gamma$$

Considerando que o valor da luminosidade seja normalizado. O menor valor de luminosidade  $L_0$  que se consegue obter num monitor médio fica entre 0,005 e 0,025.

Nesta seção serão apresentados alguns procedimentos de conversão mais conhecidos.

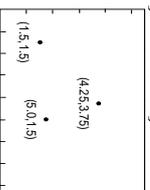
### 7.2.1 Rasterização de Pontos

A conversão se reduz em um problema de arredondamento de valores reais para inteiros, quando as coordenadas do ponto tiverem a parte fracionária diferente de zero. O algoritmo de arredondamento depende da convenção utilizada para o endereçamento dos pixels.



Se for no primeiro caso, basta arredondar para o maior inteiro que seja menor ou igual ao número em consideração. E se for no segundo caso, adicionase às coordenadas do ponto  $[0.5, 0.5]^t$  antes do arredondamento.

**Exercício 7.4** Quais são as coordenadas dos três pontos em termos de pixels para as duas convenções de endereçamento? Esboce o reticulado.



### 7.2.2 Rasterização de Curvas

Para aumentar a eficiência, os algoritmos mais conhecidos são recorrentes: dado o primeiro ponto, os outros pontos subsequentes são obtidos sempre em função do ponto anterior.

Para digitalizar uma reta dada pela expressão

$$y = \frac{\Delta y}{\Delta x}x + b$$

dois algoritmos mais conhecidos são:

- algoritmo de analisador do diferencial digital (DDA) e
- algoritmo de Bresenham's.

No DDA, como o nome já disse, o incremento das coordenadas do ponto anterior  $[x_k, y_k]^t$  para obter o ponto subsequente  $[x_{k+1}, y_{k+1}]^t$  é dado em função do diferencial

$$m = \frac{\Delta y}{\Delta x} = \frac{y_{k+1} - y_k}{x_{k+1} - x_k}.$$

Para  $m \leq 1$ , as coordenadas  $x$  crescem mais rapidamente que as coordenadas  $y$ . Portanto, a amostragem é feita incrementando unitariamente na direção  $x$ . Com isso,

$$y_{k+1} = y_k + m.$$

Se  $m > 1$  faz-se incremento unitário na direção  $y$ . Neste caso temos

$$x_{k+1} = x_k + \frac{1}{m}.$$

(Ver Fig. 3.5 do livro-texto de Foley.)

O algoritmo de Bresenham consegue converter uma linha somente com computações inteiras incrementais. O incremento é condicionado à comparação das “distâncias” entre os pixels vizinhos e a linha (o sinal do “erro”  $p_k$ ).

Considere  $m \leq 1$ . Para cada valor  $y$  correspondente a  $x_{k+1}$ , temos  $d_1 = |y - y_k|$  e  $d_2 = |y_{k+1} - y|$  e

$$d_1 - d_2 = 2 \frac{\Delta y}{\Delta x} (x_k + 1) - 2y_k + 2b - 1$$

Definindo como diferença dos erros

$$p_k = \Delta x (d_1 - d_2),$$

pode-se derivar a seguinte expressão recorrente para  $p_{k+1}$

$$p_{k+1} = p_k + 2\Delta y - 2\Delta x (y_{k+1} - y_k).$$

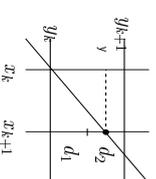
Em especial,

$$p_0 = 2\Delta y - \Delta x.$$

Assim,

$$p_k \geq 0 \rightarrow y_{k+1} = y_k + 1$$

$$p_k < 0 \rightarrow y_{k+1} = y_k$$



(Ver Figs. 3.7-3.11 do livro-texto de Foley.)

**Exercício 7.5** No algoritmo de Bresenham  $d_1$  e  $d_2$  não são exatamente as distâncias dos dois pixels adjacentes em relação à reta. Entretanto, podemos avaliar comparativamente as distâncias por estes dois valores. Por que?

**Exercício 7.6** Derive a expressão de  $p_k$  para os segmentos de inclinação entre  $45^\circ$  e  $90^\circ$ .

**Exercício 7.7** Plote os pontos de um segmento definido por  $P_1 = [1 \ 1]^t$  e  $P_2 = [9 \ 6]^t$  sobre um reticulado retangular com uso do

## 1. algoritmo DDA.

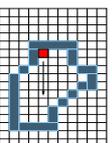
## 2. algoritmo de Bresenham.

**Observação 7.4** O algoritmo de Bresenham é adaptado para converter outros tipos de curvas, como círculos e elipses.

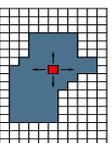
## 7.2.3 Rasterização de Áreas Preenchidas

Preenchimento de uma região é atribuir uma mesma luminosidade/brilhança ao conjunto de *pixels* pertencentes à região. Uma região pode ser definida analiticamente ou em termos de *pixels*.

Quando uma região é definida em termos de *pixels*, algoritmos recursivos com base em **preenchimento a partir de uma semente** são os mais utilizados, como *boundary-fill* e *flood-fill*. No primeiro o algoritmo percorre recursivamente a partir da semente os *pixels* vizinhos até um dos *pixels* da borda ser alcançado.



E no segundo, o algoritmo percorre recursivamente a partir da semente todos os *pixels* que tiverem a mesma luminosidade/brilhança da região original e trocá-la pela nova.



Ambos os algoritmos podem gerar resultados diferentes para os dois tipos de conectividade.

**Exercício 7.8** Escreva um algoritmo recursivo para o procedimento *flood-fill* considerando conectividade de 4.

**Exercício 7.9** Escreva um algoritmo recursivo para o procedimento *boundary-fill* considerando conectividade de 8.

O algoritmo mais conhecido para uma região definida analiticamente, ou seja, definida por seus vértices, é o **algoritmo de varredura por linha** (*scan line*). Este algoritmo explora a coerência espacial e consegue determinar incrementalmente, linha por linha, os *pixels* dentro de uma região. O

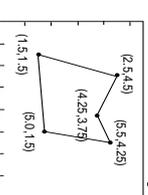
algoritmo inicia definindo os *y-buckets* para cada linha de varredura. Um *y-bucket* associado a uma linha contém informações de uma aresta da borda que a intersecta. São armazenados no *y-bucket* a coordenada *y* do ponto extremo que tem maior valor de *y* ( $y_{max}$ ), a coordenada *x* do outro ponto extremo ( $x_{min}$ ) e a inclinação da reta ( $\frac{1}{m}$ ). O conjunto de *y-buckets* é denominado *edge table*.

(Ver Fig. 3.27 do livro-texto de Foley.)

Com uso das informações contidas nos *y-buckets*, é possível preencher a região à medida que se processa sequencialmente as linhas de varredura começando com a linha 0. Para cada linha, todas as arestas do *y-bucket* correspondente são incluídas numa lista conhecida como lista ativa (*active edge table*). Os *y-buckets* são ordenados de acordo com as coordenadas *x* dos pontos de interseção das arestas que intersectam a linha de varredura corrente. Faz-se, então, o preenchimento da linha de forma alternada em relação à sequência das coordenadas *x* da lista ativa. Finalmente, antes de passar para a próxima linha, remover os *y-buckets* que tiveram  $y+1 > y_{max}$  da lista ativa.

(Ver Fig. 3.28 do livro-texto de Foley.)

**Exercício 7.10** Construa a tabela de arestas para o seguinte polígono



e mostre, passo a passo, o preenchimento dos *pixels* com uso das listas ativas. (Dica: A tabela de arestas deve conter 8 entradas, já que 8 linhas devem ser “varridas” para preencher o polígono.)

**Observação 7.5** O resultado de uma rasterização de polígono não é igual ao resultado de rasterizações das arestas de borda seguidas de preenchimento por *boundary-fill*, porque na discretização de polígonos *leave-se* em consideração a área do polígono.

(Ver Fig. 3.23 do livro-texto de Foley.)

## 7.2.4 Conversão de Modelos 3D

Em Computação Gráfica, nos casos em que as imagens são obtidas sinteticamente a partir dos modelos 3D projetados, dois paradigmas são utilizados: projetivo e anti-projetivo.

O paradigma projetivo é geométrico, no qual os modelos 3D em um espaço contínuo são reduzidos em primitivas gráficas 2D (através de transformações projetivas). Com isso, o problema passa a ser um problema de rasterização de primitivas gráficas.

(Ver Fig. 6.1 do livro-texto de Foley.)

O segundo paradigma é denominado anti-projetivo, porque o processo de amostragem é centrado nos *pixels* da imagem. Neste caso, procura-se os pontos visíveis nos modelos 3D a partir dos *pixels* na direção de visão do observador. O procedimento é chamado em algumas referências por *ray-casting*.

(Ver Fig. 15.55 do livro-texto de Foley.)

### 7.3 Amostragem por Convolução

Uma outra forma de discretizar uma imagem contínua seria considerar que a imagem  $f(x, y)$  seja de banda limitada, ou seja, a sua transformada de Fourier  $F(u, v)$  tenham valores muito pequenos para  $u$  fora do intervalo  $[-U, U]$  e  $v$  fora do intervalo  $[-V, V]$ .

(Ver Figs. 3.2, 3.3, 3.6 e 3.9 do livro-texto de Gonzalez.)

Para obter as amostras uniformes separadas de  $(\Delta x, \Delta y)$ , basta multiplicar a função  $f(x, y)$  por uma função de amostragem bidimensional  $\delta(x, y)$ .

(Ver Fig. 3.20 do livro-texto de Gonzalez.)

Pelo teorema da convolução, esta multiplicação equivale à convolução de  $\mathcal{F}\{\delta(x, y)\}$  e  $F(u, v)$ . Como  $f(u, v)$  é de banda limitada, o resultado da convolução seria um “tren bimensional” de réplicas de  $\mathcal{F}\{f(x, y)\}$  centradas nas frequências  $(n\Delta u, n\Delta v)$ .

(Ver Fig. 3.21 do livro-texto de Gonzalez.)

A complexidade envolvida nesta forma de amostragem torna a impraticável na conversão de imagens contínuas em imagens discretas. No entanto, ela nos permite analisar a adequabilidade das taxas de amostragem numa rasterização.

O inverso do espalhamento entre as duas réplicas no domínio de frequência corresponde ao espalhamento de amostragem entre dois valores de luminosidade adjacentes na imagem. Esta relação nos permite respositer uma das perdas mais frequentes no processo de conversão de primitivas gráficas 2D: quando uma imagem contínua é discretizada numa amostra de  $m \times n$  valores (uma imagem digital correspondente a uma matriz de  $m \times n$ ), onde  $m$  e  $n$  são potências inteiras de dois, quão próxima a imagem discreta é da imagem original?

O teorema de amostragem nos diz que se

$$\frac{1}{\Delta x} \geq 2U \quad \frac{1}{\Delta y} \geq 2V$$

então  $f(x, y)$  pode ser completamente recuperada. Em outras palavras, a imagem discretizada é “fiel” à imagem original.

(Ver Fig. 14.9 do livro-texto de Foley.)

$2U$  e  $2V$  são chamados **limite (de frequência) de Nyquist** nas direções  $u$  e  $v$ , respectivamente.  $\Delta x$  e  $\Delta y$  são, por sua vez, conhecidos como **intervalos de Nyquist**. Podemos sempre recuperar o sinal  $f(x, y)$  integralmente ao multiplicarmos  $F(u, v)$  por uma função de pulso de área igual a  $2U \times 2V$  centrada na origem do domínio de frequência.

(Ver Figs. 14.16 e 14.17 do livro-texto de Foley.)

Quando a taxa de amostragem for menor que a frequência de Nyquist, as altas frequências do sinal podem sobrepor com as baixas frequências. Ao tentarmos recuperar o sinal, as componentes de altas frequências aparecem como se elas fossem (além das) componentes de baixas frequências. Este fenômeno é conhecido como **aliasing**.

(Ver Fig. 14.28 do livro-texto de Foley.)

**Exercício 7.11** Uma imagem com dois níveis de cinza (preto e branco) pode ter componentes de altas frequências? Uma imagem colorida, contendo vários detalhes, pode ter uma banda limitada?

As ferramentas da Teoria de Sinais tornou possível aprimorar o processo de amostragem utilizado em Computação Gráfica para obter imagens de qualidade melhor.

### 7.4 Antialiasing

Os algoritmos de conversão apresentados na seção 7.2 são, de fato, amostragem pontual, ou seja, a luminosidade/brilhança de um *pixel* corresponde à luminosidade/brilhança de um ponto da imagem contínua. Dependendo da frequência de amostragem, pode ocorrer o fenômeno de *aliasing* ou perda de informação.

(Ver Fig. 14.10 do livro-texto de Foley.)

Uma solução seria aumentar a resolução da tela (aumentar o número de *pixels*), e portanto, na taxa de amostragem. Existem, entretanto, várias alternativas para melhorar a aparência das imagens sem alterar o número de *pixels*.

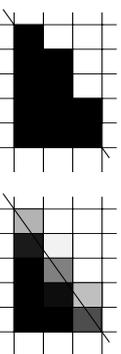
Um procedimento muito popular em Computação Gráfica é amostrar mais de um ponto para cada *pixel* e combinar as luminâncias/brilhanças destas amostras para obter a luminância/brilhança final do *pixel*. Este procedimento é conhecido como **superamostragem**. Ele reconstrói o sinal amostrado com uma taxa de frequência maior e reamostra o sinal reconstruído. (Ver Fig. 14.32 do livro-texto de Foley.)

Algumas implementações deste procedimento:

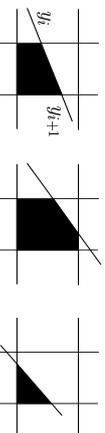
**Amostragem de Área** : consiste em tomar para valor da luminância/brilhança do *pixel*  $P$  a razão da luminância/brilhança total da imagem  $f(x, y)$  dentro do *pixel* e a área  $A$  do *pixel*

$$\frac{\int_P f(x, y) dx dy}{A}$$

Esta luminância/brilhança “suavizada” é então atribuída para o *pixel*. Observe as diferenças na rasterização da borda de um polígono scan (figura esquerda) e com (figura direita) uso da técnica de amostragem de área. A suavização consegue atenuar o padrão serrilhado (*jagged* ou *stair step pattern*).



**Exercício 7.12** Seja um segmento da borda de um polígono. A reta de suporte do segmento é dada por  $y = mx + b$ . Há três possíveis configurações de sobreposição do polígono com os *pixels*



Derive a expressão de percentagem de ocupação do polígono em cada *pixel* em termos da inclinação  $m$  e das coordenadas  $y_i$  e  $y_{i+1}$ .

Como a suavização reduz altas frequências na variação da luminância/brilhança dos *pixels*, a largura da faixa ocupada pelo espectro da imagem diminui. Isso diminui a probabilidade de sobreposição do sinais amostrados.

(Ver Figs. 3.56 e 14.11 do livro-texto de Foley.)

**Superamostragem** : ao invés de uma amostra, um conjunto de amostras  $p_1, p_2, \dots, p_n$  são tomadas para cada *pixel* (usualmente, uma potência de 2) e a luminância/brilhança média destas amostras é calculada. Podemos considerar esta amostragem uma forma discreta da amostragem de área, tendo um custo computacional menor.

**Exercício 7.13** Obter uma imagem cujos *pixels* tem valores iguais à média dos valores de grupos de 9 *pixels* ( $3^2$ ) da seguinte imagem:

```

0 4 0 0 4 4 0 0 4 0
4 3 4 0 4 4 0 4 3 4
0 4 0 0 4 4 0 0 4 0
1 0 0 0 0 0 0 0 0 1
1 1 2 2 2 2 2 2 2 1

```

Dica: Utilize uma matriz  $\frac{1}{9}$  e ignore as linhas e colunas da borda.

**Superamostragem ponderada** : semelhante à superamostragem incluindo apenas pesos a cada amostra de acordo com a sua posição relativa ao centro do *pixel*. O resultado, além de ser uma imagem suavizada (com largura de faixa menor), pode ser distinto para mesmos detalhes diferentemente posicionados dentro de um *pixel*.

(Ver Figs. 3.57, 3.58, 14.12 e 14.13 do livro-texto de Foley.)

**Exercício 7.14** Qual é o resultado da aplicação da matriz  $\frac{1}{10} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix}$  na imagem do Exercício 7.13? Compare-a com o resultado anterior.

**Observação 7.6** Em Computação Gráfica é mais fácil aplicar superamostragem quando se utiliza o paradigma anti-projetivo para obter a luminância/brilhança em cada *pixel* de uma imagem.