Capítulo 9

Amostragem

Hoje em dia, os dispositivos de exibição são dominantemente de tecnologia *raster*, ou seja, podem ser modelados como um arranjo bidimensional de pontos endereçáveis denominados *pixels* (*picture elements*). Portanto, uma vez determinada, para cada ponto do modelo geométrico, a intensidade dos três componentes primários que determinam o estímulo cromático $\vec{I}$, precisamos ainda associar estes pontos do modelo aos *pixels* (*picture elements*) se quisermos visualizá-los. Para isso, devemos discretizar, por um processo conhecido como **amostragem**, a imagem contínua (u, v), com u e v reais, em **pontos de amostra** (i, j), onde i e j são valores inteiros.

Observação 9.1 *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 a amostragem em Sistemas de Informações Gráficas. A primeira forma é geométrica (seção 9.3). Ela se ocupa com a seleção das amostras e a conversão das coordenadas cartesianas (em ponto flutuante) destas amostras para coordenadas inteiras dos *pixels*. A segunda forma é física (seção 9.4). Ela trata a imagem como uma integral/um somatório de componentes de todas as possíveis frequências de variações de intensidades ao longo das coordenadas u e v. Esta segunda forma nos permite, com base na Teoria de Informações, corrigir ou melhorar a qualidade da imagem. Na seção 9.5 são apresentadas três técnicas práticas fundamentadas nesta Teoria.

Antes de apresentarmos alguns algoritmos de amostragem espacial, é conveniente introduzirmos na seção 9.1 os conceitos elementares de imagens discretas e na seção 9.2 uma visão elementar dos monitores CRT, que ainda são muito utilizados para reprodução das imagens.

9.1 Imagens Discretas

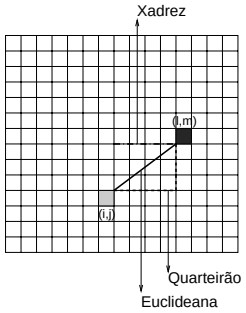
Uma imagem discreta pode ser considerada um reticulado de *pixels*, a cada qual é associada uma brilhância $\vec{I}$ capaz de produzir um certo estímulo cromático no olho humano. O *pixel* é a menor unidade endereçável por um par de coordenadas inteiras $[ij]^t$.

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*): $D_4((i,j),(l,m)) = |i-l| + |j-m|$.

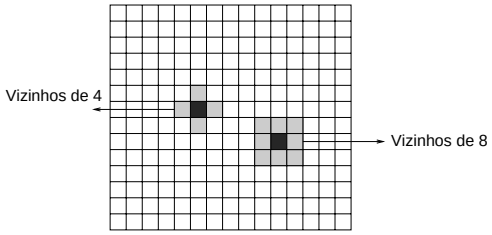
distância de xadrez (*chessboard distance*): $D_8((i,j),(l,m)) = max\{|i-l|, |j-m|\}$



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 $D_4(p,q) = 1$. 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 forem conectados. Podemos definir adjacência-de-4 e adjacência-de-8 dependendo do tipo de conectividade especificado.

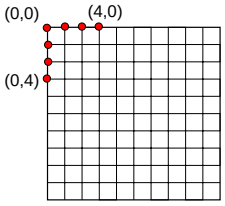


Observação 9.2 *As definições de vizinhança e contiguidade para reticulados conectados-de-4 podem levar a certos paradoxos como ilustra a seguinte discretização de uma circunferência. O resultado é uma curva conectada-de-8 e não conectada-de-4!*

9.2 Monitores CRT

O dispositivo de exibição de tecnologia *raster* que reinou nas últimas décadas é o monitor de vídeo CRT (*cathodic ray tube*).

O monitor CRT consiste de um tubo de raios catódicos com uma tela e um canhão 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 comprimentos de onda 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 endereça um nó do reticulado que tangencia os *pixels*.



A relação entre a luminância/brilhância desejada 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 γ e de uma constante K

$$L = KV^{\gamma}$$

Para que a relação fique linear de forma a produzir uma "imagem correta", condizente com a original, pode ser feita uma **correção** γ na tensão aplicada – ao invés de V, excita-se com $V^{\frac{1}{\gamma}}$ os feixes de elétrons.

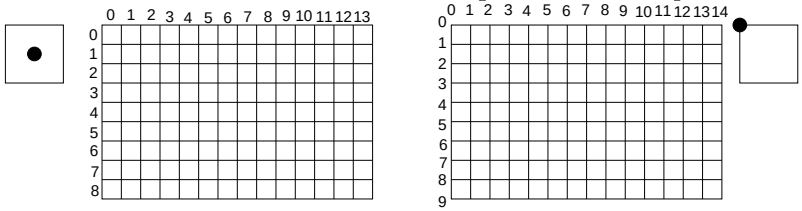
9.3 Algoritmos de Conversão

Os primeiros algoritmos de conversão de coordenadas reais para inteiras, 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.

Nesta seção serão apresentados alguns procedimentos de conversão mais conhecidos.

9.3.1 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*. Podemos associar as coordenadas inteiras ao cento do *pixel*, ao seu canto superior esquerdo, ao canto inferior esquerdo, ou a qualquer um outro ponto. O importante é ter em mente que se deve optar por uma conveção que assegura maior semelhança entre a figura vetorial original e a figura discretizada. Por este motivo, quando o sistema referencial do dispositivo for um que tem a sua origem no canto inferior esquerdo e os seus eixos x e y alinhados, respectivamente, com o lado inferior e esquerdo da tela, é comum utilizar a convenção de associar as coordenadas inteiras ao canto inferior esquerdo de cada *pixel*.



Exercício 9.1 *Considere que o referencial do dispositivo tem os eixos x e y alinhados, respectivamente, com o lado inferior e esquerdo da tela e a sua origem no canto inferior esquerdo. Desenhe um retângulo, cujas coordenadas reais (representação vetorial) e inteiras (representação raster) são (0,0), (4,0), (4,4), (0,4), num reticulado de pixels cujo endereçamento é por*

- *centro do pixel,*
- *canto esquerdo inferior,*
- *canto esquero superior,*

- *canto direito inferior, ou*
- *canto direito superior.*

Em quais casos, a área do retângulo é preservada após a rasterização? Em quais casos, o canto inferior esquerdo do retângulo, de coordenada (0,0) coincide com o canto inferior esquerdo da tela?

Exercício 9.2 *A correspondência entre as coordenadas do espaço raster e as do espaço vetorial é 1:1 (biunívoca), 1:m, m:1, ou m:n? Há perda de informação, ou "degradação" da qualidade de uma imagem, no processo de rasterização?*

9.3.2 Rasterização de Curvas

A rasterização de curvas, além do problema de arrendodamento, deve-se preocupar com a posição relativa dos *pixels* adjacentes para que o conjunto seja uma boa aproximação da sua forma original. Para aumentar a eficiência, os algoritmos mais conhecidos são **recorrentes**: dado o primeiro ponto, os outros pontos subsequentes são obtidos em função do ponto anterior, com o passo mínimo de um *pixel*, para evitar a geração de um conjunto de pontos de mesmos atributos gráficos (por exemplo, mesma cor) mapeado num mesmo *pixel*.

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 – *digital differential analyzer*) e
- algoritmo de **Bresenham**.

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,

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

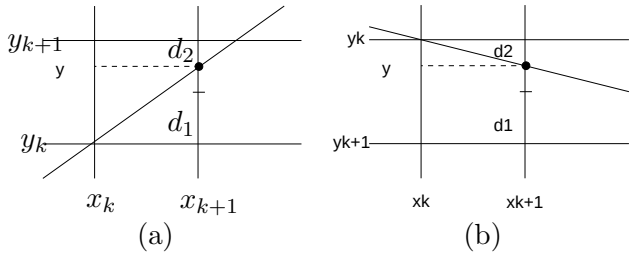


Figura 9.1: Inclinação de reta (a) positiva e (b) negativa.

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 Breseham consegue converter uma linha somente com computações inteiras incrementais. O incremento é condicionado à comparação das "distâncias" entre os *pixels* vizinhos e a reta.

Se $m \leq 1$, ou seja $|\Delta y| < |\Delta x|$, e $\Delta y > 0$, o ponto $(x_{k+1}, y) = (x_k + 1)$, subsequente ao ponto (x_k, y_k), deve estar entre (x_{k+1}, y_k) e (x_{k+1}, y_{k+1}) (Figura 9.1.(a)).

Definindo $d_1 = |y - y_k|$ e $d_2 = |y_{k+1} - y|$, temos

$$\begin{aligned} d_1 - d_2 &= (\frac{\Delta y}{\Delta x}x + b) - y_k - y_{k+1} + (\frac{\Delta y}{\Delta x}x + b) \\ &= 2\frac{\Delta y}{\Delta x}(x_k + 1) - 2y_k + 2b - 1 \end{aligned}$$

Como $\Delta x > 0$, basta analisarmos o sinal da **diferença dos erros**

$$\begin{aligned} p_k &= \Delta x(d_1 - d_2) = 2\Delta y x_k + 2\Delta y - 2\Delta x y_k - \Delta x + 2b\Delta x \\ &= 2\Delta y x_k - 2\Delta x y_k + (2\Delta y + 2b\Delta x - \Delta x). \end{aligned}$$

para concluirmos se devemos incrementar ou não a coordenada y, ou seja,

$$\begin{aligned} p_k \geq 0 &\rightarrow y_{k+1} = y_k + 1 \\ p_k < 0 &\rightarrow y_{k+1} = y_k \end{aligned} \tag{9.1}$$

Agora, vamos ver uma forma eficiente para determinar p_k para cada *pixel* k, envolvendo somente somas e subtrações de "constantes", Δx, $2\Delta x$, $2\Delta y$. Fazendo $c = (2\Delta y + 2b\Delta x - \Delta x)$, que é um valor constante, temos

$$p_k = 2\Delta y x_k - 2\Delta x y_k + c$$

e para p_{k+1}, podemos derivar uma expressão recorrente

$$\begin{aligned} p_{k+1} &= 2\Delta y x_{k+1} - 2\Delta x y_{k+1} + c \\ &= 2\Delta y(x_k + 1) - 2\Delta x y_{k+1} + 2\Delta x y_k - 2\Delta x y_k + c \\ &= (2\Delta y x_k - 2\Delta x y_k + c) - 2\Delta x(y_{k+1} - y_k) + 2\Delta y \\ &= p_k + 2\Delta y - 2\Delta x(y_{k+1} - y_k). \end{aligned} \tag{9.2}$$

Em especial,

$$\begin{aligned} p_0 &= 2\Delta y x_0 - 2\Delta x y_0 + c \\ &= 2\Delta y x_0 - 2\Delta x(\frac{\Delta y}{\Delta x} x_0 + b) + 2\Delta y + 2b\Delta x - \Delta x \\ &= 2\Delta y - \Delta x \ . \end{aligned} \tag{9.3}$$

Pela Eq. 9.1 podemos simplificar ainda a Eq. 9.2, pois

$$\begin{aligned} p_k \geq 0 \ &\rightarrow \ y_{k+1} = y_k + 1, \text{ então } p_{k=1} = p_k + 2\Delta y - 2\Delta x \\ p_k < 0 \ &\rightarrow \ y_{k+1} = y_k, \text{ então } p_{k=1} = p_k + 2\Delta y. \end{aligned} \tag{9.4}$$

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

Exercício 9.3 *Rasterize o segmento definido pelos pontos* $(9, 4)$ *e* $(18, 8)$ *com uso do*

1. *algoritmo DDA*
2. *algoritmo de Bresenham*

Compare o tipo de operações envolvidas e o número de operações dos dois algoritmos. Qual deles "independe" do procedimento de arredondamento utilizado? Qual deles é mais eficiente? Justifique a sua resposta.

Exercício 9.4 *Observe que para a situação apresentada na Figura 9.1.(b),* $y_{k+1} = y_k - 1$ *ao invés de* $y_{k+1} = y_k + 1$*. Neste caso, Eqs. 9.4 e 9.3 devem ser modificadas ligeiramente. Quais são as alterações? Justifique a sua resposta.*

Exercício 9.5 *Mostre que se $m > 1$, ou seja $|\Delta y| > |\Delta x|$, as expressões para computar as diferenças dos erros em cada iteração são semelhantes às Eqs. 9.4 e 9.3, quando $\Delta x > 0$, e às equações derivadas no Exercício 9.4, quando $\Delta x < 0$.*

Exercício 9.6 *Utilize o algoritmo de Bresenham para rasterizar os seguintes segmentos*

1. *(15,7) – (10,10)*
2. *(10,10) – (15,1)*
3. *(10,1) – (15,10)*

Observação 9.3 *O algoritmo de Bresenham é adaptado para converter outros tipos de curvas planas, como círculos e elipses, e segmentos 3D.*

9.3.3 Rasterização de Áreas

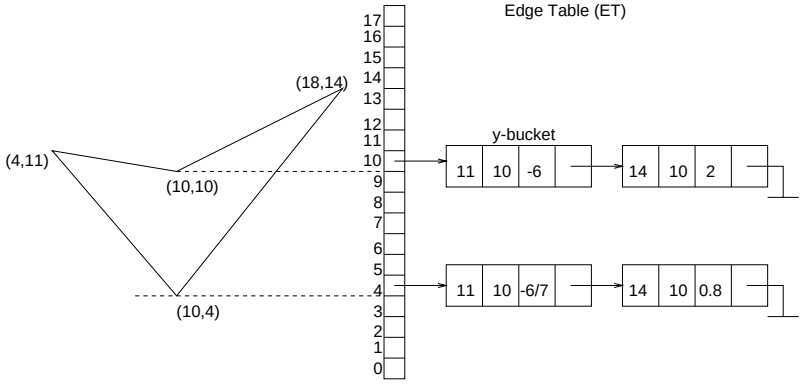
Rasterização de áreas consiste em transformar uma região plana delimitada por uma sequência fechada de segmentos em um conjunto de *pixels* conexos. Tipicamente, os algoritmos para rasterização de áreas exploram certas coerências entre os *pixels* dentro da área de interesse, a fim de reduzir o número de processamentos.

O algoritmo mais conhecido é o **algoritmo de varredura por linha** (*scanline*). Este algoritmo explora vários tipos de **coerência espacial** (Seção 10.1 e consegue determinar **incrementalmente**, linha por linha, os *pixels* dentro de uma região, com uso dos seguintes fatos:

1. os *pixels* das arestas da borda da região podem ser obtidos com uso dos algoritmos recorrentes de rasterização de linhas (seção 9.3.2);
2. numa linha de varredura os *pixels* podem ser divididos em distintas extensões (*extent*) separadas por *pixels* correspondentes às arestas da borda. A classificação de pertinência das extensões à área de interesse só muda nestes *pixels* "divisórios"; e
3. a classificação de pertinência das extensões identificadas em cada linha se alterna ao longo de uma linha de varredura.

O algoritmo inicia definindo os *y-buckets* para cada linha de varredura. Um *y-bucket* associado a uma linha contém informações de uma aresta do polígono 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 ponto extermo que tem menor valor de y (x_{min}) e a inclinação da reta ($\frac{1}{m}$). O conjunto de *y-buckets* é denominado *edge table*.



Observação 9.4 *Observe que em edge table são armazenados os dados necessários para rasterizar recorrentemente as arestas da borda da área de interesse com o algoritmo DDA.*

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 de varredura y, os *y-buckets* ativos são ordenados de acordo com os valores x no seu campo x_{min} para poder fazer o preenchimento da linha de forma alternada entre os segmentos delimitados por estes pontos. O conjunto de *y-buckets* ativos associados à linha de varredura corrente é conhecida como **lista ativa** (*active edge table*). Finalmente, antes de passar para a próxima linha, remover da lista ativa os *y-buckets* que tiverem $y+1 > y_{max}$. Para os *buckets* remanescentes, substituir o valor x no seu campo x_{min} por $x+\frac{\Delta x}{\Delta y}$. Ao passar para a próxima linha, é adicionada à lista ativa os *y-buckets* da linha, e o procedimento é repetido até que a tabela de arestas fique vazia.

Observação 9.5 *No sítio* `http://www.cse.unsw.edu.au/~cs3421/slides/bres/ScanLine.html` *encontra-se um applet do algoritmo de scanline, através do qual vocês podem verificar o procedimento, passo a passo. Para quem entende alemão, vale a pena dar uma olhada no sítio* `http://www.iam.unibe.ch/~fcglib/special_www/cg_lecture/lectContent/polygon_filling/applets/scanline_applet/scanline_applet_intro.php`.

Exercício 9.7 *Rasterize o seguinte polígono definido pela seguinte sequência de vértices: (1.5,1.5), (5.0,1.5), (5.5, 4.25), (4.25,3.75), (2.5,4.5).*

Podemos integrar ao algoritmo de varredura por linha a tonalização de Gouraud. O cômputo da cor em cada *pixel* pode ser feito incrementalmente, se as cores nos pontos $P_1 = (x_1, y_1, z_1, 1)$ e $P_2 = (x_2, y_2, z_2, 1)$ forem conhecidas. Sejam $C_1 = (R_1, G_1, B_1)$ e $C_2 = (R_2, G_2, B_2)$ as cores em P_1 e P_2, respectivamente. Pela interpolação linear, obtém-se a cor

$$C(t) = (1 - t)C_1 + tC_2$$

para o ponto

$$P(t) = (1 - t)P_1 + tP_2.$$

Numa mesma linha de varredura, o valor t_{k+1} deve ser em função do incremento de X

$$X_k + 1 = x_1 + t_{k+1}(x_2 - x_1)$$

ou seja

$$(x_1 + t_k(x_2 - x_1)) + 1 = x_1 + t_{k+1}(x_2 - x_1).$$

Daí podemos chegar a

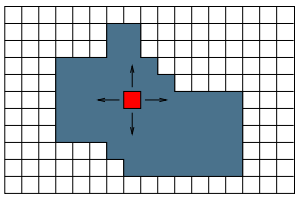
$$t_{k+1} = t_k + \frac{1}{\Delta x}.$$

E, de forma análoga, podemos derivar o valor t_{k+1} na linha de varredura $k + 1$ em relação ao valor t_k da linha anterior k

$$t_{k+1} = t_k + \frac{1}{\Delta y}.$$

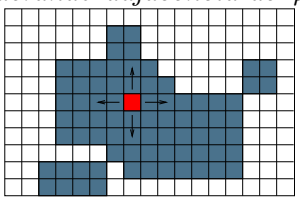
Exercício 9.8 *Considere que as cores nos vértices do polígono dado no Exercício 9.7 sejam, respectivamente, (0,1,1), (0,1,0),(1,1,0),(0,0,1) e (0,1,0). Rasterize o polígono utilizando a tonalização de Gouraud.*

Quando uma região é rasterizada e queremos trocar, por exemplo, as sua cor de preenchimento por uma outra, podemos utilizar algoritmos recursivos com base em **preenchimento a partir de uma semente**. Um exemplo desta classe de algoritmos é o algoritmo *flood-fill*. O algoritmo percorre recursivamente a partir da semente todos os *pixels* que tiverem a cor antiga e trocá-la pela nova.



Este algoritmo pode gerar resultados diferentes para os dois tipos de conectividade, adjacência-de-4 e adjacência-de-8.

Exercício 9.9 *Utilize o algoritmo de flood-fill para trocar a cor do polígono para a cor amarela, considerando adjacência-de-4 e adjacência-de-8.*



Compare os resultados obtidos.

9.3.4 Conversão de Modelos 3D

Após as transformações projetivas que convertem os modelos geométricos 3D para 2D, como vimos no capítulo 5, rasterização dos modelos 3D pode ser reduzida em rasterização 2Dpara a qual podemos utilizar os algoritmos comentados nas seções 9.3.1, 9.3.2 e 9.3.3.

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

Uma outra alternativa para rasterização "direta" de modelos 3D é lançar um ou mais raios projetivos a partir do observador na direção de **cada** *pixel* da imagem e determinar a "área" do modelo 3D que está mais próxima do observador na direção do(s) raio(s). A cor da área alcançada é "transferida" para o *pixel*. Este procedimento de conversão é conhecido por *ray-casting*.

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

9.4 Análise Espectral

Um dos problemas encontrados nos processos de rasterização apresentados é a presença de **bordas serrilhadas** (*jagged* ou *stair step pattern*) nas imagens, devido às decisões binárias (acende–apaga) em relação à luminância/brilhância de cada *pixel* da tela e às possíveis transições abruptas de luminância/brilhância entre dois pontos vizinhos. É natural perguntar-se se há alguma forma de eliminar, ou pelo menos atenuar, tais efeitos. Veremos nesta seção que esta pergunta pode ser respondida com os conceitos da Teoria de Informações.

Uma imagem pode ser representada como uma **função contínua** $I(u, v)$ de luminância/brilhância num domínio espacial $[u_{min}, u_{max}] \times [v_{min}, v_{max}]$, contendo uma infinidade de possibilidades de variações nas intensidades entre dois pontos vizinhos. No processo de rasterização, esta função é amostrada em alguns pontos (u, v), transformando-se numa **função discreta**. Esta função discreta contém somente as variações entre as intensi-

dades dos pontos amostrados, podendo haver perda de informações. Existe, então, uma frequência de amonstragem para a qual não há nenhuma perda de informações? Esta pergunta pode ser respondida mais facilmente com uso da **transformada de Fourier** de $I(u,v)$, que transforma esta função do domínio espacial para o domínio de frequência (de variação das luminâncias/brilhâncias) w_u e w_v

$$\mathcal{I}(w_u, w_v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} I(u,v)[\cos 2\pi(w_u u + w_v v) - i \sin 2\pi(w_u u + w_v v)]dudv. \tag{9.5}$$

Uma breve revisão sobre a transformada de Fourier é dada na seção 9.4.1.

Sejam $[-U, U]$ e $[-V, V]$ as faixas espectrais para as quais $\mathcal{I}(w_u, w_v)$ não se anula. O **teorema de amostragem** nos diz que se

$$\frac{1}{\Delta u} \geq 2U \qquad \frac{1}{\Delta u} \geq 2V, \tag{9.6}$$

$I(u,v)$ pode ser completamente recuperada. Em outras palavras, a imagem discretizada é "fiel" à imagem original. $2U$ e $2V$ são chamados **limite (de frequência) de Nyquist** nas direções u e v, respectivamente. Δu e Δv são, por sua vez, conhecidos como **intervalos de Nyquist**, como veremos mais detalhadamente na seção 9.4.2.

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

Para visualiza as amostras num dispositivo de saída que operam de forma analógica como um monitor CRT, precisamos **reconstruir** a partir delas sinais analógicos através de um conversor D/A (veremos no Capítulo 12 que as informações de cada amostra enviadas ao monitor são, de fato, digitais), usualmente pelo esquema *sample and hold* por um período de 1 *pixel*. No caso de um monitor CRT, o valor da lumiância/brilhância associado a cada *pixel* é convertido em tensão a ser aplicada no canhão dos elétrons. Como a frequência de Nyquist da maioria das imagens tende a ser infinita, veremos que é difícil assegurar a reconstrução exata da imagem original. Na seção 9.4.3 mostraremos, sob o ponto de vista teórico, como se pode atenuar as distorções e aproximar melhor as imagens geradas das imagens originais.

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

9.4.1 Transformada de Fourier

A **transformada de Fourier** pode ser considerada com um processo de **transformação de funções**. A partir de uma função $f(x)$ definida no domínio de x ela nos dá uma outra função

$$\mathcal{F}(w) = \mathsf{F}(f(x)) = \int_{-\infty}^{+\infty} f(x)[\cos 2\pi wx - i \sin 2\pi wx)dx]$$

que é uma integral de Fourier no domínio de w.

Duas condições são suficientes para existência de uma transformada de Fourier:

1. $f(x)$ é contínua por parte em todo o intervalo,
2. $f(x)$ é integrável.

Exercício 9.10 *Determine a transformada de Fourier de uma função-pulso*

$$f(x) = \begin{cases} 1, & |x| \leq 1 \\ 0, & |x| > 1 \end{cases}$$

No caso de imagens, que contém duas variáveis, é aplicada uma transformada de Fourier bi-dimensional. Considerando que todas as imagens $I(u,v)$ satisfazem a condição de continuidade e de integrabilidade, a sua transformada de Fourier $\mathsf{F}(I(u,v)) = \mathcal{I}(w_u, w_v)$ é dada pela Eq. 9.5. Observe que $\mathcal{I}(w_u, w_v)$ é um valor complexo cuja amplitude/magnitude é dada por

$$|\mathcal{I}(w_u, w_v)| = \sqrt{(\Re^2(\mathcal{I}(w_u, w_v)) + Imag^2(\mathcal{I}(w_u, w_v)))}$$

e o ângulo de fase por

$$\phi(w_u, w_v) = tg^{-1} \frac{Imag(\mathcal{I}(w_u, w_v))}{\Re(\mathcal{I}(w_u, w_v))},$$

com $\Re$ e $Imag$ designando a sua parte real e a parte imaginária, respectivamente.

Observação 9.6 *A magnitude de uma transformada de Fourier de uma imagem indica a "quantidade" da componente de frequência (w_u, w_v) e o ângulo de fase indica "onde" se encontra tal componente na imagem.*

A versão discreta da função $\mathcal{I}(w_u, w_v)$ dada pela Eq. 9.5 para $N \times M$ amostras igualmente espaçadas é

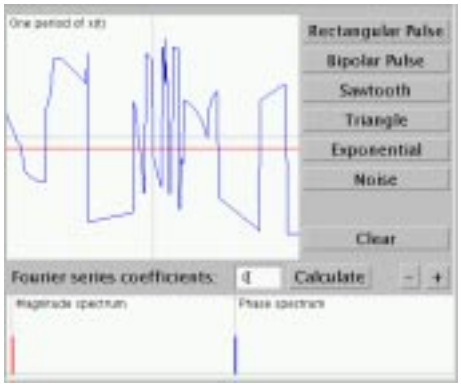
$$\mathcal{I}(w_u, w_v) = \frac{1}{NM} \sum_{u=0}^{N-1} \int_{v=0}^{v=M-1} I(u,v)[\cos 2\pi(\frac{w_u u}{N} + \frac{w_v v}{M}) - i \sin 2\pi(\frac{w_u u}{N} + \frac{w_v v}{M})] du dv, \tag{9.7}$$

que equivale a expandir a função de imagem $I(u,v)$ em um somatório de função constante e uma série de funções trigonométricas de distintas frequências (harmônicas).

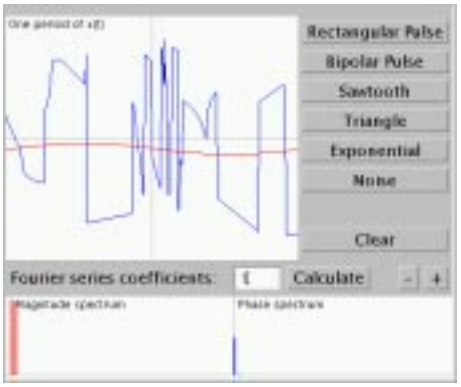
Exercício 9.11 *Dadas as quatro amostras* $(0.25, 2)$, $(0.5, 3)$, $(0.75, 4)$, $(1.0, 3)$ *da função* $f(x)$. *Determine a magnitude da sua transformada de Fourier discreta* $\mathcal{F}(0)$, $\mathcal{F}(1)$, $\mathcal{F}(2)$ *e* $\mathcal{F}(3)$.

Exercício 9.12 *Qual das imagens tem mais harmônicas? Imagens em dus cores, preto e branco, ou imagens com tons de cinza na qual não se consegue distinguir transições abruptas entre os tons? Justifique a sua resposta com o espectro de magnitude.*

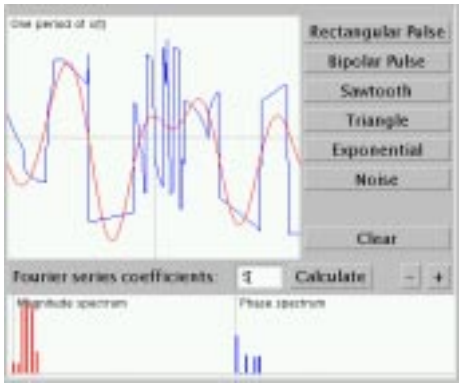
Observação 9.7 *No sítio* `http://www.jhu.edu/~signals/fourier2/` *encontra-se um applet que ilustra a expansão de uma função* $f(x)$ *em uma série de Fourier. A seguinte sequência de ilustra as diferentes aproximações da função* $f(x)$ *plotada em azul¿*



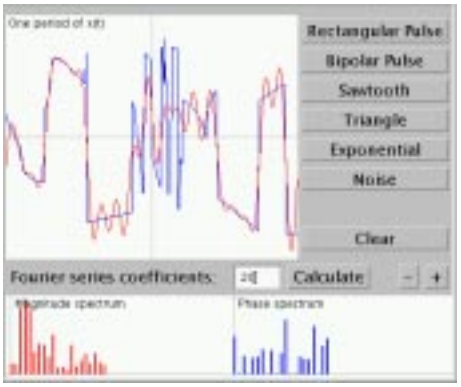
componente dc



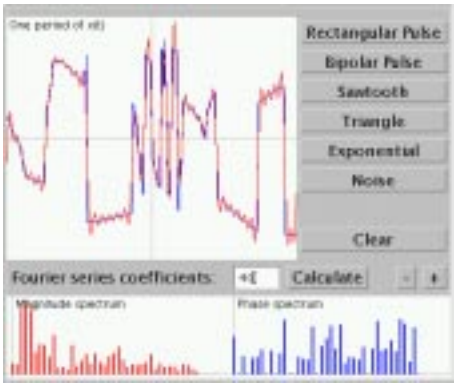
1 harmônica



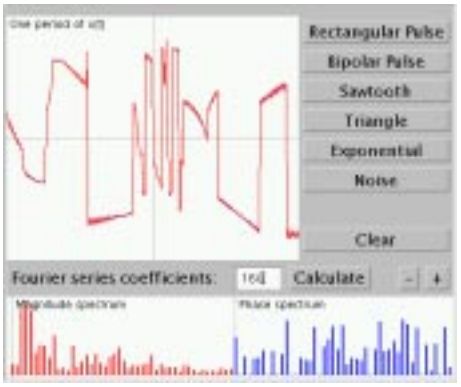
5 harmônicas



20 harmônicas



40 harmônicas



160 harmônicas

9.4.2 Amostragem

Para obter as amostras uniformes separadas de $(\Delta u, \Delta v)$, basta multiplicar a função $I(u, v)$ por uma **função de amostragem** bidimensional $\delta(u, v)$

(*comb function*).

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

Pelo **teorema da convolução**, a transformada deste produto equivale à convolução das suas transformadas correspondentes, $\mathsf{F}\{\delta(u,v)\}$ e $\mathcal{I}(w_u, w_v)$, ou seja,

$$\mathsf{F}(I(u,v)\delta(u,v)) = \mathsf{F}\{\delta(u,v)\} * \mathcal{I}(w_u, w_v),$$

isto é, integrar as cópias, "transladadas" e ponderadas por $\mathcal{F}\{\delta(u,v)\}$, de $\mathcal{I}(w_u, w_v)$.

(Ver Fig. 14.22 do livro-texto de Foley.)

Exercício 9.13 *Determine a convolução de duas funções de pulso*

$$f(x) = g(x) = \begin{cases} 1, & |x| \leq 1 \\ 0, & |x| > 1 \end{cases}$$

Lembrete: Para obter a função $h(x) = f(x) * g(x)$ deve-se dividir a reta x em 4 intervalos: $(-\infty, -1)$, $(-1, 0)$, $(0, 1)$ e $(1, \infty)$.

No sítio `ttp://www.ju.edu/signals/convolve/index.html` *encontra-se um applet que deomonstra graficamente a convolução de duas funções.*

Exercício 9.14 *Esboce o gráfico da convolução de*

1. *duas funções-pulso dadas no Exercício 9.13.*
2. *uma função-pulso, como no item anterior, e uma função triangular*

$$g(x) = \begin{cases} x, & x \in [0, 1] \\ 2 - x, & x \in [1, 2] \end{cases}$$

3. *um trem de impulsos (função de amostragem) distanciados de 2.5 e uma função triangular, como definido no item anterior.*

Se a função $\mathcal{I}(w_u, w_v)$ for de **banda limitada**, ou seja, ela tem $|\mathcal{I}(w_u, w_v)|$ nulos para u fora do intervalo $[-U, U]$ e v fora do intervalo $[-V, V]$ e o espaçamento de amostragem satisfizer Eq. 9.6, o resultado da convolução seria um "trem bidimensional" de réplicas de $\mathcal{I}(w_u, w_v)$, sem distorções, centradas nas frequências $(2mU, 2nV)$. Em outras palavras, o sinal poderá ser recuperado integralmente.

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

(Ver Figs. 14.27 do livro-texto de Foley.)

Caso o inverso do espaçamento entre as duas réplicas no domínio de frequência (usualmente, em termos de unidades de amostras por *pixel*), que correponde ao espaçamento espacial (usualmente, em termos de unidades de *pixels*) entre duas amostras, for menor que a frequência de Nyquist, as componentes de altas frequências de uma réplica podem sobrepor as outras réplicas e serem confundidas com as componentes de baixas frequências, como se elas fossem (aliás das) compoenentes de baixas frequências. Este fenômeno é conhecido por *aliasing*.

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

Exercício 9.15 *Se aumentarmos a taxa de amostragem, podemos reduzir o efeito de aliasing? Justifique a sua resposta.*

Exercício 9.16 *Se filtramos uma imagem com larga faixa espectral por um filtro passa-baixo, pode-se reduzir o efeito aliasing. Explique a afirmação.*

9.4.3 Reconstrução

No domínio espectral, a reconstrução de um sinal original a partir do sinal amostrado equivale a extrair uma réplica do trem de réplicas da transformada do sinal amostrado $\mathsf{F}(I(i,j))$. No domínio espectral, isso corresponde ao produto de $\mathsf{F}(I(i,j))$ pela transformada da função de pulso $\mathsf{F}(Box(u,v))$. Pelo teorema de convolução, o procedimento correponde à aplicação da transformada de Fourier sobre a convolução dos sinais amostrados $I(i,j)$ pela função-pulso $Box(u,v)$

$$\mathsf{F}(I(i,j) * Box(u,v)) = \mathsf{F}\{Box(u,v)\}\mathsf{F}(I(i,j)),$$

Observe que para reconstruir fielmente o sinal original a largura do espectro de $\mathsf{F}(Box(u,v))$ deve cobrir todo o espectro de uma réplica.

Vale observar que a distribuição da intensidade luminosa do(s) feixe(s) de elétrons que incide(m) sobre um ponto da tela é aproximadamente Gaussiana, ou seja, tal feixe convolui, de certa forma, o sinal de tensão aplicada no canhão CRT, suavizando as variações nas cores.

(Ver Figs. 14.27(e) e 14.28(e) do livro-texto de Foley.)

Observação 9.8 *Funções frequentemente utilizadas na amostragem e na reconstrução dos sinais de uma imagem são esboçadas na Fig. 14.25 de Foley. Duas versões são apresentadas: no domínio espacial e no domínio espectral.*

9.5 Antialiasing

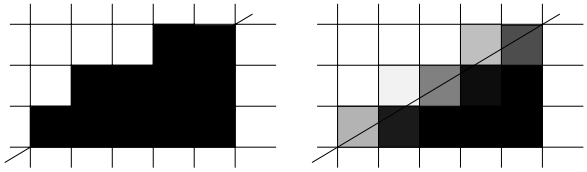
Como já comentamos, os algoritmos de conversão apresentados na seção 9.3 são passíveis à geração de bordas serrilhadas. Na seção 9.4 mostramos, com base em Teoria de Informações, que estas bordas serrilhadas é um efeito de *aliasing*. Uma alternativa seria estreitar a função de reconstrução, removendo as componentes de frequências sobrepostas e a outra seria aumentar o número de amostras. Ambas soluções estão relacionadas com o *hardware* do dispositivo de saída: a modificação do processo de reconstrução implementado e resolução do dispositivo. Nesta seção apresentamos três soluções a nível de *software*. O princípio básico destas soluções é eliminar as componentes de frequências altas ou atenuar as fortes transições de luminâncias/brilhâncias nas bordas, com uso de mais de uma amostra para computar a luminância/brilhância de cada *pixel*.

(Ver Fig. 14.29 e 14.32 do livro-texto de Foley.)

Amostragem por Área . Esta solução se baseia no fato de que qualquer primitiva rasterizada ocupa, no mínimo, um *pixel*. Ela consiste em tomar para valor da luminância/brilhância do *pixel* P a razão da luminância/brilhância total da imagem $f(x, y)$ dentro do *pixel* e a área A do *pixel*

$$\frac{\int_P f(x,y)dxdy}{A}.$$

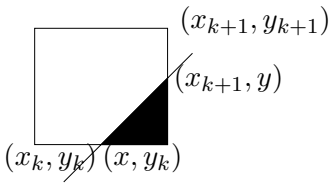
Esta luminância/brilhância "suavizada" é então atribuída para o *pixel*. Observe as diferenças na rasterização da borda de um polígono sem (figura esquerda) e com (figura direita) uso da técnica de amostragem de área. A suavização consegue atenuar o padrão serrilhado.



Como a suavização reduz altas frequências na variação da luminância/brilhância dos *pixels*, a largura da faixa ocupada pelo espectro da imagem diminui. Isso diminui a probabilidade de *aliasing*.

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

Exercício 9.17 *Determine, por método de amostragem por área, a cor a ser atribuída a uma amostra, sabendo que ela pertence a um polígono que cobre parcialmente o canto inferior direito de um pixel*



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

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 uma faixa espectral mais estreita), 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.)