Capítulo 11

Operações Básicas de Imagens

Uma vez rasterizadas as imagens vetoriais podemos ainda modificar as imagens discretas resultantes, aplicando sobre elas operações a nível de *pixel*, isto é, gerar uma nova imagem cujos valores são funções dos valores dos *pixels* da imagem original. Entendemos pelo **processamento de imagens** o conjunto de técnicas e algoritmos que conseguem mapear os (multi-)valores $f_k(i_k, j_k)$ do *pixel* (i_k, j_k) de k imagens para o (multi-)valor $g(i_f, j_f)$ do *pixel* (i_f, j_f) de uma outra imagem, através de uma transformação T

$$g(i_f, j_f) = T[\sum \mu_k f_k(i_k, j_k)],$$

onde μ_k é um fator de ponderação. Tais métodos desempenham um papel importante em Sistemas de Informações Gráficas sob vários aspectos:

- por proverem recursos para facilitar a percepção e a interpretação humana das informações gráficas, como **segmentacão**, **realce** de contraste;
- por possibilitarem efeitos visuais especiais, como *warping* e *morphing*;
- por viabilizarem o melhoramento automático da qualidade das imagens, como **superamostragem**, e **restauração** de imagens degradadas;
- por permitirem fazer certas modificações no espaço 2D, como mudar a orientação ou aplicar *zooming* nas imagens, sem repercorrer o fluxo de síntese de imagens que envolve transformações geométricas e projetivas

Esses métodos se aplicam tanto às imagens sintéticas quanto às imagens reais, como as capturadas pelas câmeras fotográficas, pelos satélites, e pelos equipamentos hospitalares.

Entende-se por **realce** de uma imagem a redistribuição das luminâncias/brilhâncias de forma a acentuar contrastes do objeto de interesse da aplicação. A **restauração**, por sua vez, tem o propósito direcionado para reconstruir ou recuperar uma imagem "degradada", usando-se algum conhecimento a priori do processo de degradação. Enquanto a **segmentação** consiste em distinguir numa imagem as partes de interesse.

A transformação T consiste essencialmente de uma sequência de **operações lógico-aritméticas**, *pixel* por *pixel*, e/ou de **transformações geométricas** sobre a posição de cada *pixel*. A grosso modo, podemos dizer que as primeiras operações alteram o valor da função associada aos *pixels* e a segunda classe de operações afeta a relação espacial entre os *pixels*.

As operações lógico-aritméticas são aplicadas, normalmente, em imagens de dimensões iguais. Na seção 11.1 apresentamos um conjunto de operadores lógico-aritméticos. Estas operações, com vasta aplicação em processamento de imagens, são simples e podem ser implementadas de forma extrememente eficiente com a tecnologia disponível. Em seguida, mostramos na seção 11.2 que transformções geométricas apresentadas no Capítulo 3 podem também ser aplicadas nos pontos discretos das imagens, com a diferença de que novos endereços nem sempre são os endereços válidos no espaço discreto e os endereços contíguos nem sempre são mapeados em endereços contíguos. Portanto, soluções complementares são necessárias para assegurar uma nova imagem com mínima distorção possível. Diversas soluções baseadas em técnicas de **filtragem** tem sido propostas, algumas das quais são detalhadas na seção 11.3.

11.1 Operações Lógico-Aritméticas

As operações lógico-aritméticas são operações locais envolvendo somente os **valores** de um *pixel* por imagem, indenpendentemente dos valores dos *pixels* restantes. O endereçamento dos *pixels* é preservado.

As operações aritméticas entre dois *pixels* (i, j) de duas imagens distintas f_1 e f_2 são definidas como

Adição :

$$g(i, j) = f_1(i, j) + f_2(i, j),$$

ou seja, os valores dos *pixels* são adicionados, componente por componente, na forma convencional de soma. Observe que f_2 pode ser uma

função constante C. Neste caso, todos os valores dos *pixels* sofrem o mesmo incremento C. Uma das aplicações da adição de imagens é composição de imagens.

Subtração :

$$g(i,j) = f_1(i,j) - f_2(i,j),$$

ou seja, os valores dos *pixels* são subtraídos, componente por compoenente, na forma convencional de subtração. Quando f_2 é uma função constante C. todos os valores dos *pixels* sofrem o mesmo decremento C. Uma das aplicações da subtração é diminuir a intensidade luminosa do fundo de uma imagem para facilitar a segmentação dos objetos da cena.

Diferença absoluta :

$$g(i,j) = |f_1(i,j) - f_2(i,j)|,$$

ou seja, a diferença absoluta entre os valores dos *pixels* são computados componente por compoenente. Uma possível aplicação desta operação é para identificar a mobilidade dos objetos de uma cena.

Multiplicação :

$$g(i,j) = f_1(i,j) * f_2(i,j),$$

ou seja, os valores dos *pixels* são multiplicados, componente por compoenente, na forma convencional de multiplicação. Quando f_2 é uma função constante C, todos os valores dos *pixels* sofrem o mesmo fator de escala C. Uma das aplicações da multiplicação é aumentar o contraste entre os objetos de uma cena.

Divisão :

$$g(i,j) = f_1(i,j)/f_2(i,j),$$

ou seja, os valores dos *pixels* são divididos, componente por compoenente, na forma convencional de divisão. Quando f_2 é uma função constante C, todos os valores dos *pixels* sofrem o mesmo fator de escala $\frac{1}{C}$. Uma das aplicações da divisão é a deteção da mobilidade dos objetos de uma cena.

Exercício 11.1 *Explique melhor como cada uma das operações podem ser aplicadas em distintas situações. Consultar o sítio* `http://homepages.inf.ed.ac.uk/rbf/HIPR2/arthops.htm`.

A quantidade de *bits* utilizada para representar um valor do *pixel* é limitada. Se o resultado de uma operação aritmética for superior ao maior valor representável ou inferior ao menor valor representável, o efeito visual é dependente da implementação. Há essencialmente duas estratégias: **saturação** (o valor é saturado no maior/menor valor representável) e **wrapping** (o módulo do maior/menor valor representável é utilizado).

Observação 11.1 *A média (adição) das imagens ruidosas pode resultar numa imagem com uma definição melhor.*

(Ver Fig. 4.18 do livro-texto de Gonzalez.)

Exercício 11.2 *Dadas as duas imagens em níveis de cinza, cuja tonalidade de cinza é representada por um byte*

115	100	103	107	104	96	105	96	92	97	103	93
89	89	90	94	106	93	101	98	95	88	88	88
40	49	47	47	68	56	80	88	87	90	87	75
77	93	92	67	51	138	159	146	125	157	170	172
96	93	105	105	100	102	98	93	99	54	46	50
51	46	50	51	47	37	35	45	44	34	28	31
22	24	27	30	75	95	89	90	82	21	21	33
161	173	166	165	174	184	172	173	155	174	169	138
126	120	129	159	140	105	190	190	185	171	109	107

Imagem 1

170	200	88	98	155	120	105	130	130	140	120	100
89	99	90	94	106	93	101	98	105	105	88	88
40	70	47	80	68	56	80	88	87	90	87	75
77	140	92	68	51	140	159	146	115	157	170	172
96	93	105	105	100	102	98	93	99	54	46	50
51	46	68	68	200	200	200	188	188	120	120	120
22	24	27	30	75	130	188	120	82	145	21	33
161	173	166	165	174	184	172	173	99	174	150	160
126	170	129	159	140	105	190	190	185	171	109	107

Imagem 2

1. *Qual á a imagem resultante para cada uma das seguintes operações aritméticas aplicadas sobre as duas imagens: adição, subtração, diferença absoluta, multiplicação e divisão?*

2. *Em qual das operações pode resultar valores fora da faixa representável?*

3. *Em qual das operações pode resultar valores negativos? Qual é o efeito visual deste resultado?*

4. *Qual é o efeito visual da técnica de saturação e da técnica de wrapping?*

Diferentemente das operações aritméticas, as operações lógicas são aplicadas a nível dos *bits* que representam os valores f_1 e f_1 de cada *pixel* (i, j) das duas imagens de entrada. Portanto, elas são tipicamente utilizadas em imagens binárias ou em níveis de cinza. As mais utilizadas em processamento de imagens são

E :

$$g(i,j) = f_1(i,j) \wedge f_2(i,j),$$

ou seja, o valor $g(i,j)$ é o resultado da operação boolena E entre os valores f_1 e f_2, *bit* a *bit*. Esta operação é muito utilizada para detectar a sobreposição de duas imagens. Quando f_2 é um valor constante C, dizemos que ela é uma **máscara** que pode remover certos objetos de uma imagem.

OU :

$$g(i,j) = f_1(i,j) \vee f_2(i,j),$$

ou seja, o valor $g(i,j)$ é o resultado da operação boolena OU entre os valores f_1 e f_2, *bit* a *bit*. Esta operação pode ser utilizada para composição de duas imagens binárias.

XOR :

$$g(i,j) = f_1(i,j) \oplus f_2(i,j),$$

ou seja, o valor $g(i,j)$ é o resultado da operação boolena XOR entre os valores f_1 e f_2, *bit* a *bit*. Esta operação pode também ser utilizada para detecção da mobilidade dos objetos contidos numa cena.

Complemento :

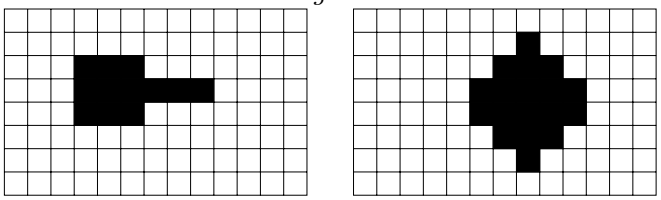
$$g(i,j) = \neg f_1(i,j),$$

é equivalente à operação booleana NOT convencional. Ela é uma operação básica em várias técnicas de processamento de imagens para separar os objetos do fundo de uma imagem.

(Ver Fig. 2.14 do livro-texto de Gonzalez.)

Exercício 11.3 *Explique melhor como cada uma das operações podem ser aplicadas em distintas situações. Consultar o sítio* `http://homepages.inf.ed.ac.uk/rbf/HIPR2/arthops.htm`.

Exercício 11.4 *Dadas as duas imagens binárias*



1. *Qual á a imagem resultante para cada uma das seguintes operações lógicas aplicadas sobre as duas imagens: E, OU, XOR?*

2. *Qual é a imagem resultante da operação lógica NOT sobre cada uma das imagens?*

11.2 Transformações Geométricas

Transformações geométricas são operações que alteram a posição dos pontos no espaço, como vimos no Capítulo 3. Certamente, estas transformações podem também ser aplicadas sobre os pontos de uma imagem discreta para, por exemplo, deslocar, rotacionar e aumentar a escala dos objetos contidos numa imagem. Além disso, é comum utilizar funções não lineares, porém contínuas, para alterar as distâncias entre as amostras, como a imagem fosse feita de borracha. Este tipo de distorção, em que a posição relativa entre as amostras é preservada, é denominado **warping**.

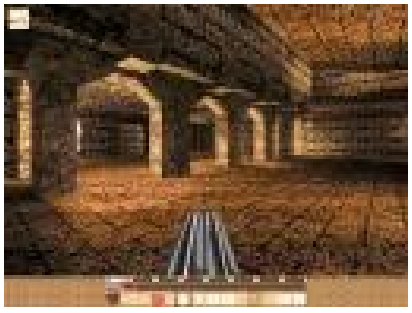
Neste caso, é comum utilizar as coordenadas polares. A transformação de coordenadas cartesianas (i, j) para coordenadas polares (r, ϕ) se dá por

$$r = \sqrt{i^2 + j^2} \qquad \phi = arctg\frac{j}{i}.$$

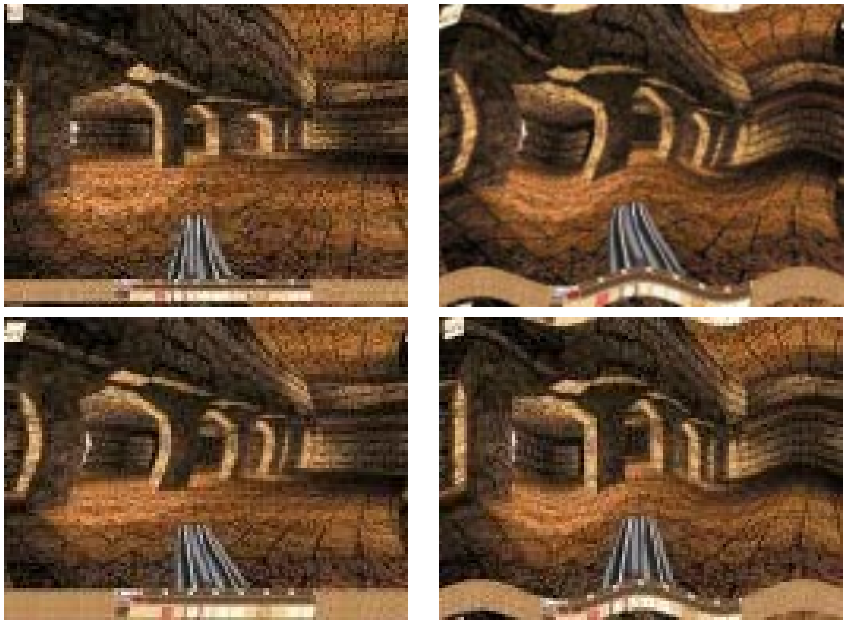
Após a transformação, as coordenadas polares devem ser convertidas novamente para coordenadas cartesianas para endereçar corretamente os *pixels*

$$i = r\cos(\phi) \qquad j = r\sin(\phi).$$

Exemplo 11.1 *As seguintes imagens distorcidas (warping images a partir da imagem original*



foram extraídas do sítio `http://www.gamers.org/dEngine/quake/wavy_image/warp.html`

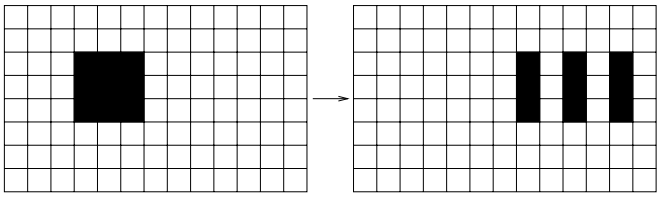


Na primeira imagem superior esquerda foi aplicado esticamento em ambas direções x e y e na superior direita, ondulações senoidais em ambas as direções. Nas duas imagens inferiores, ondulações senoidais são aplicadas nas direções y e x, respectivamente.

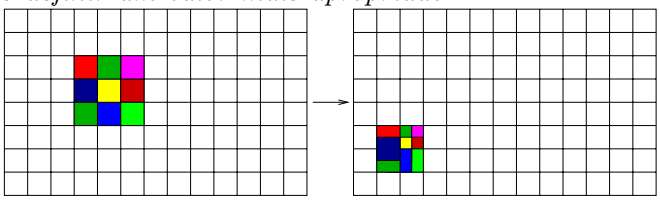
Transformações geométricas e *warping* no espaço de imagem são operações simples que permitem criar efeitos visuais interessantes sem onerar o custo de processamento. Entretanto, pelo fato das imagens serem discretas, o reticulado de *pixels* transformado não se alinha necessariamente com o reticulado original. Isso pode gerar dois problemas:

- dois *pixels* adjacentes se afastam demasiadamente, criando vazios no reticulado original e
- dois ou mais *pixels* se aproximam demais, sobrepondo em cima de um ponto do reticulado original.

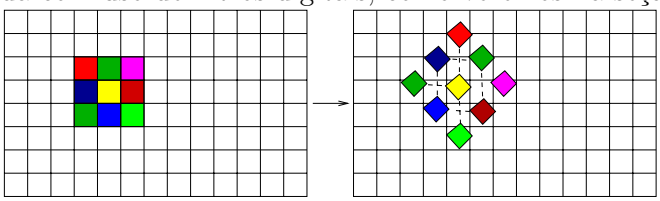
Exemplo 11.2 *Ao escalonar um quadrado constituído por 9 pixels na direção x, podemos gerar sulcos entre os pixels cujos valores precisam ser interpolados para que o resultado seja visualmente um retângulo.*



Se multiplicarmos as coordenadas de um quadrado por um fator de escala igual a $\frac{1}{2}$, teremos mais de um valor associado a um pixel. Neste caso, precisaremos definir um valor mais apropriado.



No primeiro caso, precisaremos calcular os valores para preencher os buracos entre os *pixels* de valores conhecidos. A solução mais trivial é preencher com o valor do *pixel* mais próximo com o valor conhecido. Outra alternativa, muito difundida hoje em dia, é utilizar a **soma ponderada** dos valores dos *pixels* mais próximos. Esta última alternativa pode ser eficientemente implementada com uso de filtros digitais, como veremos na seção 11.3



No segundo caso, é comum considerar que houve uma superamostragem e aplica-se um filtro de suavização para obter o valor do novo *pixel* (seção 9.5).

Exercício 11.5 *Quais seriam as imagens resultantes se aplicarmos na imagem $n \times m$ do Exemplo 11.2 as seguintes transformações:*

1. *rotação de 45^0 em torno do centro do quadrado,*
2. *deslocamento* $[1.5\ 1.5]^t$,
3. *aplicação de um fator de escala* $[0.5\ 0.5]^t$,
4. *warping pela função*

$$x \quad = \quad round(8.0 * \sin(2.0 * PI * i/128.0))$$

$$\begin{aligned} y &= round(8.0 * \cos(2.0 * PI * j/128.0)) \\ i_f &= (i + x + n)\%n \\ j_f &= (j + y + m)\%m \end{aligned}$$

Aplique a técnica de "amostra mais próxima" para preencher os buracos.

Observação 11.2 *Algumas funções de warping podem ser encontradas em* `http://local.wasp.uwa.edu.au/~pbourke/projection/imagewarp/` *e* `http://www.gamers.org/dEngine/quake/wavy_image/warp.html`.

11.3 Filtragem

Vimos na seção 9.4 que uma imagem contínua pode ser discretizada num reticulado $f(i\Delta x, k\Delta y)$ de $N \times M$ *pixels* $x = i\Delta x$ e $y = k\Delta y$ para $i \in \{0, \cdots, N-1\}$ e $k \in \{0, \cdots, M-1\}$. A sua **transformada discreta** de Fourier

$$\begin{bmatrix} f(0,0) & \cdots & f(0,i\Delta y) & \cdots & f(0,(N-1)\Delta y) \\ f(\Delta x,0) & \cdots & f(\Delta x,i\Delta y) & \cdots & f(\Delta x,(N-1)\Delta y) \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ f((M-1)\Delta x,0) & \cdots & f((M-1)\Delta x,i\Delta y) & \cdots & f((M-1)\Delta x,(N-1)\Delta y) \end{bmatrix}$$

é dada por

$$F(u\Delta u, v\Delta v) = \frac{1}{NM} \sum_{i=0}^{N-1} \sum_{k=0}^{M-1} f(i\Delta x, k\Delta y) e^{-j2\pi(\frac{iu}{N} + \frac{kv}{M})}$$

e a inversa desta transformada por

$$\mathcal{F}^{-1}[F(u\Delta u, v\Delta v)] = f(i\Delta x, k\Delta y) = \sum_{u=0}^{N-1} \sum_{v=0}^{M-1} F(u\Delta u, v\Delta v) e^{j2\pi(\frac{iu}{N} + \frac{kv}{M})},$$

onde $\Delta u = \frac{1}{N\Delta x}$ e $\Delta v = \frac{1}{M\Delta y}$.

Mostramos ainda que a convolução das duas funções no domínio espacial (x, y) equivale à transformada inversa do produto das suas transformadas no domínio espectral

$$g(x,y) = h(x,y) * I(x,y) = \mathsf{F}^{-1}[\mathcal{H}(u,v)\mathcal{I}(u,v)] = \sum_{i=0}^{N-1} \sum_{k=0}^{M-1} \mathcal{H}(x - i\Delta x, y - k\Delta y)\mathcal{I}(i\Delta x, k\Delta y).$$

Se $\mathcal{I}(u,v)$ for a transformada da função de luminância/brilhância de uma imagem, podemos considerar $\mathcal{H}(u,v)$ uma **função de filtragem** de $\mathcal{I}(u,v)$, pois são zeradas todas as componentes de $\mathcal{I}(u,v)$ de frequências em que $\mathcal{H}(u,v)$ se anula. Em decorrência disso, teremos uma nova imagem com uma nova distribuição espectral, podendo conter somente frequências altas ($\sim$ uma nova imagem com alto contraste, se $\mathcal{H}(u,v)$ for um filtro passa-alto) ou somente frequências baixas ($\sim$ uma nova imagem suavizada, se $\mathcal{H}(u,v)$ for um filtro passa-baixo).

(Ver Fig. 4.19 do livro-texto de Gonzalez)

Esta análise espectral nos permite concluir que se quisermos suavizar as cores de uma imagem, podemos convoluí-la com um filtro passa-baixo e, se quisermos uma imagem com elevado contraste, uma simples solução é convoluí-la com uma filtro passa-alto. O problema se reduz, então, à escolha adequada da **função de convolução** $h(x,y)$ no domínio espacial. Decidida a função, ela é discretizada numa área retangular que é sempre um múltiplo da área de *pixels* da imagem de interesse. Esta área é conhecida por **área de suporte do filtro**. Um subconjunto de pontos não nulos nesta área de suporte é denominado **máscara de convolução** espacial. Para que se garanta que a filtragem não altere a luminância/brilhância média da imagem, é imposta ainda que os valores de uma máscara de ordem w sejam tais que

$$\frac{1}{NM}\sum_{i=1}^{w}\sum_{k=1}^{w} h_{ik} = 1$$

(Ver Fig. 4.20–4.28 do livro-texto de Gonzalez)
(Ver Fig. 17.7 do livro-texto de Foley)

Vale ainda observar que no uso de uma máscara de convolução para filtragem, devemos tomar devidos cuidados para processar os *pixels* da borda da imagem onde a máscara não consegue cobrir todos os elementos. Estes pontos são usualmente tratados de forma diferenciada dos pontos interiores. Entre as propostas existentes para tratar os *pixels* da borda temos:

- reduzir o tamanho das amostras a serem convoluídas, de forma que a máscara nunca ultrapassa da borda da imagem;
- atribuir um valor constante pré-definido para os pontos da máscara que ficam fora da imagem em processamento;
- replicar os valores dos *pixels* da borada para os pontos da máscara que ficam fora da imagem em processamento.

Observação 11.3 *Detalhes da derivação de uma máscara de convolução a partir de um filtro $H(u, v)$ no domínio de frequência podem ser encontrados na seção 4.5 do livro-texto de Gonzalez.*

Nesta seção apresentamos alguns filtros suavizadores mais utilizados.

11.3.1 Filtro *Box* ou Filtros por Média

É um filtro de suavização que atenua as altas frequências da imagem fora do domínio D. O operador de transformação é dado por

$$h(x, y) = \begin{cases} (\frac{1}{2a}) \cdot (\frac{1}{2a}), \text{se} -a \leq x, y \leq a \\ 0, \text{caso contrário} \end{cases} .$$

Observação 11.4 *O operador binário $\cdot$ indica que $H(u, v)$ pode ser obtida em dois passos por aplicações sucessivas da transformada de Fourier unidimensional. Isso é garantido pela propriedade de separabilidade das transformadas de Fourier.*

Observação 11.5 $\mathcal{F}[\frac{1}{2a}] = sincu = \frac{senu}{u}$.

Uma discretização desse filtro numa máscara espacial 3×3, ou **filtro espacial**, é dada por:

$$\frac{1}{9}\begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

(Ver Fig. 4.21 do livro-texto de Gonzalez)

O fator $\frac{1}{9}$ é um fator de normalização para garantir que a luminância total da imagem seja preservada.

Um dos principais problemas deste filtro de suavização é que ele "borra" bordas e pequenos detalhes.

(Ver Fig. 4.22 do livro-texto de Gonzalez)

Exercício 11.6 *Aplique o filtro box nas imagens do Exercício 11.2.*

11.3.2 Filtros por Mediana

Estes filtros tem como objetivo "borrar" menos as bordas e os detalhes. O nível de cinza de cada pixel é substituído pela mediana dos níveis de cinza na vizinhança. O tamanho da vizinhança fica a critério de cada aplicação.

A principal desvantagem de um filtro por mediana é poder "corromper" linhas finas e cantos pontiagudos. Um método para evitar isso é escolher máscaras de geometria mais apropriada.

(Ver Fig. 4.23 do livro-texto de Gonzalez)

Exercício 11.7 *Aplique o filtro por mediana nas imagens do Exercício 11.2.*

11.3.3 Filtro de Bartlett

É um filtro de suavização que atenua mais as altas frequências em relação ao filtro *box*. A função no domínio espacial é dada por

$$h(x,y) = \begin{cases} (1-|x|)\cdot(1-|y|), se x \leq 1 \ e \ y \leq 1 \\ 0, caso\ contrário \end{cases} .$$

Observação 11.6 $\mathcal{F}[1-|x|] = sinc^2 u = \frac{sen^2 u}{u^2}$.

Uma discretização deste filtro pode ser obtida em duas etapas:

1. máscara unidimensional, tomando três pontos $\{-\frac{1}{2}, 0, \frac{1}{2}\}$, cinco pontos $\{-\frac{2}{3}, -\frac{1}{3}, 0, \frac{1}{3}, \frac{2}{3}, \}$ ou sete pontos $\{-\frac{3}{4}, -\frac{2}{4}, -\frac{1}{4}, 0, \frac{1}{4}, \frac{2}{4}, \frac{3}{4}\}$ do intervalo de suporte. Substintuindo-os na função $h(x)$ e normalizando-os, teremos as seguintes máscaras, respectivamente,

$\frac{1}{4}$

1	2	1

$\frac{1}{9}$

1	2	3	2	1

$\frac{1}{16}$

1	2	3	4	3	2	1

2. Pela propriedade de separabilidade, podemos obter uma máscara bidimensional, multiplicando as máscaras unidimensionais antes de normalizar os valores da máscara. Por exemplo, para um filtro de Bartlett de ordem 5 teremos

$\frac{1}{81}$

1	2	3	2	1
2	4	6	4	2
3	6	9	6	3
2	4	6	4	2
1	2	3	2	1

(Ver Fig. 14.23(c) do livro-texto de Foley)

Exercício 11.8 *Aplique o filtro de Bartlett nas imagens do Exercício 11.2.*

11.3.4 Filtro Gaussiano

A função de transformação é dada por uma função gaussiana na forma

$$h(x,y) = \frac{1}{2\sigma^2\pi} e^{\frac{-(x^2+y^2)}{2\sigma^2}}.$$

(Ver Fig. 14.25(c) do livro-texto de Foley)

Pode-se mostrar que ela é equivalente à convolução dos filtros *box*. Portanto, convoluindo sucessivamente os filtros *box* podemos obter uma aproximação para o filtro gaussiano. No caso de ordem 5, temos o seguinte filtro espacial

$$\frac{1}{256}\begin{array}{|c|c|c|c|c|}\hline 1 & 4 & 6 & 4 & 1 \\ \hline 4 & 16 & 24 & 16 & 4 \\ \hline 6 & 24 & 36 & 24 & 6 \\ \hline 4 & 16 & 24 & 16 & 4 \\ \hline 1 & 4 & 6 & 4 & 1 \\ \hline \end{array}$$

(Ver Fig. 14.33(d) do livro-texto de Foley)

Exercício 11.9 *Aplique o filtro gaussiano nas imagens do Exercício 11.2.*