

Técnicas de Melhorias e Reconhecimento de Formas por Imagens

Aura Conci e Rafael Heitor Correia de Melo

Depto de Ciências de Computação, Instituto de Computação, Universidade Federal Fluminense
aconci@ic.uff.br, rmelo@ic.uff.br - <http://www.ic.uff.br/~aconci/AI.html>

Resumo: *Este mini curso apresenta diversas técnicas de melhorias, segmentação e identificação de imagens de uso na engenharia e ciências. Os tópicos abordados podem ser agrupados em:*

- *Técnicas tradicionais baseadas no histograma para segmentação da imagem.*

- *Técnicas de filtragem baseadas na Transformada de Fourier.*

- *Técnicas de identificação de formas genéricas baseadas na Transformada de Hough.*

Não se tem a menor pretensão de esgotar assuntos tão vastos, mas apenas apresentar, ao interessado, os pontos fundamentais de cada técnica.

Palavras-chave: processamento de imagens, histograma, transformadas, filtragem.

Introdução

Sistemas baseados em imagens são importantes para uma série de aplicações do dia a dia. Embora haja no mercado diversos produtos que oferecem sistemas genéricos, a complexidade de muitas aplicações da área precisa de abordagens mais específicas. Nas próximas seções, apresentamos técnicas que embora não sejam encontradas facilmente nos sistemas comerciais fazem muita diferença em uma grande gama de aplicações de imagens digitais.

O texto deste mini curso é dividido em 3 partes. Na primeira destas, fala-se no uso do histograma para melhoria e segmentação de imagens. A parte 2 considera a teoria da Transformada de Fourier e a filtragem por convolução. Sua utilização na melhoria e processamento de imagens é abordada. A metodologia relacionada à transformada de Hough é descrita na última parte.

Este texto foi planejado para ser de leitura fácil, possibilitando que o interessado aprenda sozinho (mesmo sem qualquer conhecimento anterior da área) os aspectos básicos e possa evoluir por si próprio, posteriormente, nos assuntos que lhe forem de maior interesse.

Parte 1:

Modificações baseadas no histograma

Um fator de grande ajuda no processamento e análise de imagens é o seu histograma. Ele mostra a estatística de ocorrência de cada nível de tonalidade da imagem, isto é, indica quantos pixels existem na imagem conforme o tom.

Veja por exemplo a imagem "Degradê" na figura abaixo:

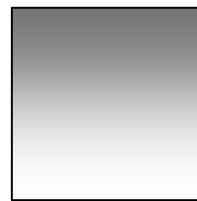


Figura 1: Imagem "Degradê"

A imagem "Degradê" tem 100 x 100 pixels, e o mesmo tom em cada linha. Os tons variam de linha para linha a incrementos de 2 tons, ou seja, uma certa linha tem o mesmo tom da anterior mais 2.

Se você olhasse o "bitmap" da imagem "Degradê" veria a seguinte distribuição tonal (obs.: *bitmap* ou Mapa de Bits da imagem é o mapeamento que define a intensidade ponto a ponto de uma imagem. Não confunda com o tipo de arquivo de mesmo nome que tem a terminação BMP) :

0	0	0	0	0	0	0	0
2	2	2	2	2	2	2	2
4	4	4	4	4	4	4	4
6	6	6	6	6	6	6	6
8	8	8	8	8	8	8	8

.....

.....

Para calcular o histograma dessa imagem você poderia fazer a seguinte análise: ela tem 100 colunas, logo terá 100 pixels de cada tom. Como a imagem tem 100 linhas, na linha 0 teve tom 0, na linha 1 teve tom 2, na linha 2 teve tom 4 e na linha 3 teve tom 6. Obviamente, na linha 4, terá tom 8, e **assim** sucessivamente de modo que a linha 99 terá o tom $2 \times 99 = 198$.

Logo o histograma do "bitmap" da imagem "Degradê" irá do tom 0 ao tom 198, com 100 pixels em cada tom. De modo que terá 100 pixels para cada tom, do tom 0 ao 198 e 0 pixel nos demais tons. Esse histograma tem uma envoltória com a aparência de um degrau em linha reta, como mostrado na figura abaixo.

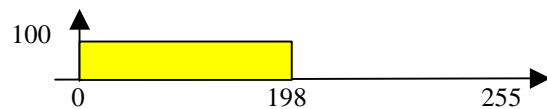


Figura 2: Histograma da imagem "Degradê"

Repare que, sob o ponto de vista da análise desse histograma, essa imagem não apresenta um bom contraste, a imagem não usa toda a sua possibilidade de contraste já que não há qualquer pixel com tons acima de 198: 200....a 255.

Ela poderia ser melhorada usando funções lineares ou não lineares (como o método de **equalização** [6]) para **aumentar o contraste**.

O uso de funções lineares é bem mais simples, bastando encontrar a função adequada para aumentar sua amplitude (*range*) de 198 para 255. Tal função poderia ser obtida por "regra de 3":

$$tom_novo = tom_atual \times 255/198$$

$$tom_novo = tom_atual \times 1.287878787879$$

Usando a expressão anterior, nos valores tonais da imagem, ela ficará com um novo "bitmap", que agora, salvo alguns erros de arredondamentos, irá até 255:

0	0	0	0	0	...
3	3	3	3	3	...
5	5	5	5	5	...
8	8	8	8	8	...
10	10	10	10	10	...
:	:	:	:	:	:
255	255	255	255	255	...

Será que houve alguma perda de dados neste processo? Como os tons na forma final devem ser inteiros, a multiplicação pela constante real acima nem sempre resultará no mesmo incremento no nível do tom. Esse incremento era constante e igual a 2 na imagem original. A imagem processada, devido aos arredondamentos, poderá ter incrementos não constantes. Isso não é perceptível por olhos humanos, mas, em um processo de detecção de incrementos em visão de máquina, poderia ser notado. Assim, essa constância dos incrementos tonais é a única "perda" no processo.

Técnica de *stretching*

A técnica de usar funções lineares para **aumentar o contraste** é denominada de *stretching* do histograma e pode ser generalizada através de uma

fórmula que pode ser facilmente memorizada, como exemplificado a seguir.

Nas figuras abaixo são mostrados vários histogramas que descrevem imagens com características diferentes no que se refere ao contraste.

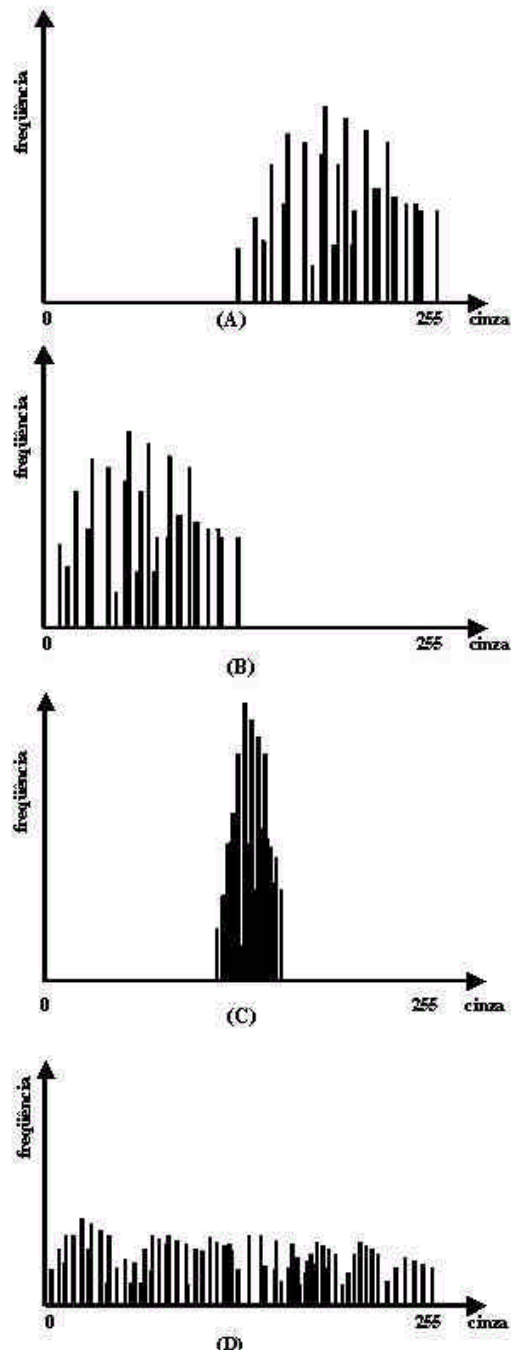


Figura 3: Caracterizando o contraste pelo histograma

Sob ponto de vista da análise do histograma, como já deve ter ficado claro pela seção anterior, uma imagem que apresenta um bom contraste é aquela que exhibe um histograma onde as colunas mostram-se

distribuídas de forma a terem valores dentro de toda a escala de níveis de cinza. Dessa forma, podemos dizer em relação aos histogramas nas figuras A, B, C e D que:

(1) O histograma em A apresenta concentração de pixels no final da escala, ou seja, há ali a representação de uma imagem muito clara;

(2) Em B há uma imagem escura, que apresenta alta concentração de pixels no início da faixa de cinzas;

(3) Em C tem-se uma imagem com pixels concentrados em tons intermediários e baixo contraste;

(4) O histograma em D é o que mostra melhor contraste, nele aparece uma distribuição de tons de forma mais balanceada.

O histograma, assim, permite que você identifique como vai parecer uma figura na tela sem que ela seja vista. O que é muito útil para técnicas automáticas de análise e visão de máquina.

Como devem aparecer as figuras na tela em cada caso? Na imagem A, a figura que aparecerá na tela terá características predominantemente claras ou "muito brancas". Já na figura que aparecerá em B será uma imagem com características de tons mais pretos, imagens escuras. Em C, a imagem aparecerá toda muito cinza, com baixa variação na iluminação. Na D, a imagem aparecerá na tela com melhor nitidez. As figuras 4 a 7, que seguem, mostram essas imagens.



Figura 4 : Imagem referente ao histograma 3A

Para melhorar a imagem A, dever-se-ia fazer o tom mais escuro ser zero, mas o mais claro permanecer ainda em 255. Se essa imagem tivesse tom mínimo, T_{min} , o ideal é que esse tom passasse a ser 0, ao mesmo tempo em que o intervalo total que ela cobria, que antes era $(255 - T_{min})$ agora seria de 255. Isso é feito pela equação :

$$t_{novo} = 255 (t - T_{min}) / (255 - T_{min})$$

onde t representaria o tom de cinza de cada pixel inicial e t_{novo} o tom depois da multiplicação.

A imagem B para ser melhor deveria estar com o tom máximo, T_{max} , em 255; como ela inicia em zero, basta ampliar seu intervalo de T_{max} , para 255. De maneira semelhante ao que fizemos na imagem A, isso pode ser feito pela equação :

$$t_{novo} = 255 t / T_{max}$$

onde t e t_{novo} representariam o tom de cinza de cada pixel anterior e posterior ao processo.

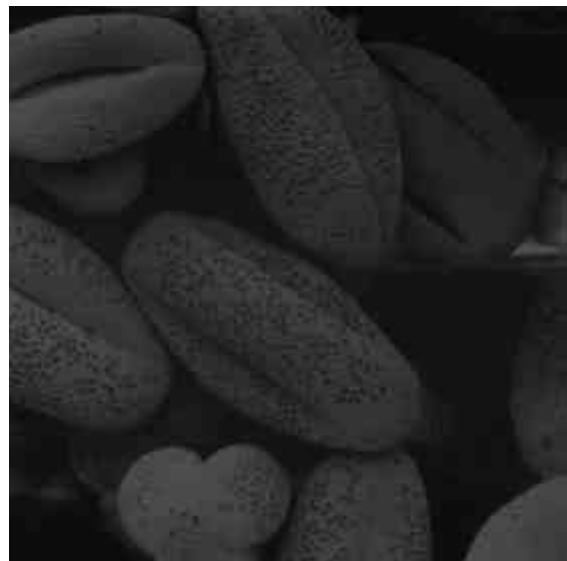


Figura 5: Imagem referente ao histograma 3B

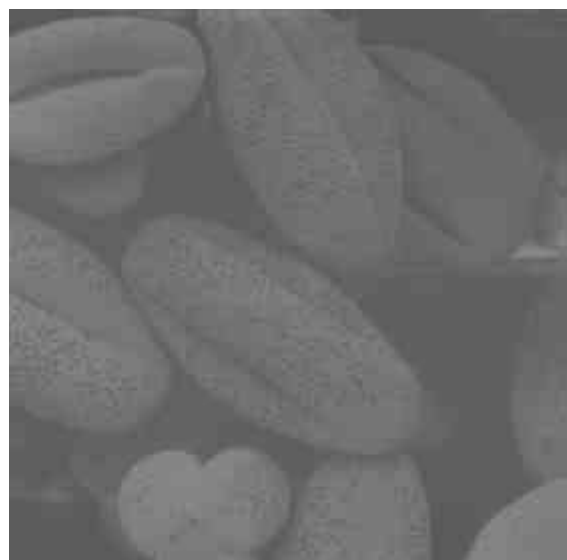


Figura 6: Imagem referente ao histograma 3C

Vamos chamar o menor tom de cinza da imagem C de T_{min} e o maior de T_{max} . Para ela ser melhorada, T_{min} deveria estar em zero e o tom

máximo, T_{max} , em 255. Isto é, ao invés de apenas ser representada no pequeno intervalo $(T_{max}-T_{min})$, deveria cobrir todo o intervalo 255. Isso é feito pela equação :

$$t_{novo} = 255 (t - T_{min}) / (T_{max} - T_{min})$$

onde t representaria o tom de cinza inicial de cada pixel e t_{novo} o tom final.

Repare que essa última expressão inclui os casos específicos dos histogramas anteriores e pode ser usado como uma fórmula geral para a técnica .

As imagens depois de processadas por essas expressões terão um histograma como o da figura 3D, e ficarão muito mais nítidas (figura 7).



Figura 7: Imagem referente ao histograma 3D

Aqui foi discutida apenas uma técnica de melhoria de imagens usando operações ponto a ponto e o histograma da imagem. Esta é uma técnica linear. Existem diversas outras. Um grupo de técnicas não lineares usa um histograma de referência ou desejado para aproximar o histograma da imagem tratada no processo de sua melhoria. Dois tipos de histogramas desejados são principalmente usados: constantes e gaussianos (figura 8).

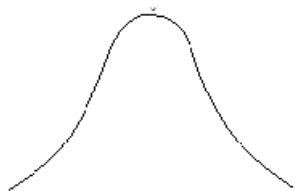


Figura 8 : Aspecto de uma curva gaussiana.

O histograma gaussiano (em homenagem à curva estudada por Karl Friedrich Gauss, matemático e astrônomo alemão, que viveu de 1777 a 1855) ou uma aproximação desse, como geralmente é mais usado, produz imagens mais suaves já que se

aproxima mais de uma distribuição normal, que seria a distribuição de um único objeto em cena bem iluminado. Quando um histograma constante é usado para aproximar o histograma da imagem a ser tratada a técnica é denominada de Equalização [6].

Segmentação de objetos por tom

A conversão de uma imagem com níveis de cinza para uma imagem com representação binária (dois tons) é geralmente feita a partir do histograma. Essa conversão, chamada binarização, é importante para uma série de objetivos, tais como:

- identificação e segmentação de objetos na imagem ;
- na análise das formas que compõem a imagem ;
- na armazenagem mais eficiente; ou
- na apresentação em dispositivos com somente um bit de resolução de intensidade.

A binarização ou limiarização (*thresholding*) para preto-e-branco é o método mais simples de segmentação de imagens. Consiste em transformar uma imagem, originalmente em muitos tons de cinza (ou em tons de diversas cores, no caso de imagens coloridas), em outra composta por apenas pixels pretos e brancos.

A decisão da nova cor que terá o pixel da imagem é feita de acordo com a escolha do ponto de corte T (limiar ou *threshold*). Qualquer pixel com intensidade menor ou igual ao ponto de corte passa a ser branco (ou preto). Se o pixel tem intensidade maior que o ponto de corte, passa a ter a cor preta (ou branca, de acordo com o que foi convencionado).

Como uma imagem digital pode ser descrita como uma função $f(x,y) = t$, a resposta de uma binarização é:

$$g(x,y) = \begin{cases} R_1 & \text{se } f(x,y) \leq T \\ R_2 & \text{se } f(x,y) > T \end{cases} \quad (1)$$

onde R_1 e R_2 são os valores estipulados para os dois níveis de cinza da imagem binária.

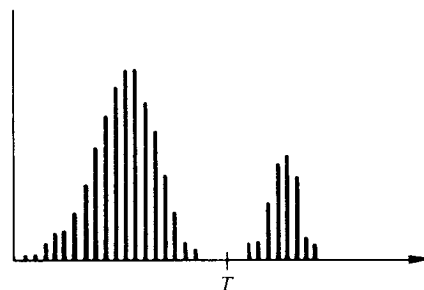


Figura 9: Histograma bimodal e valor de limiarização, T .

O diagrama da figura que segue representa a transformação das intensidades da imagem para apenas dois níveis distintos, no caso usamos 0 e 255

(preto e branco) ou através da aplicação da função $s = T(r)$.

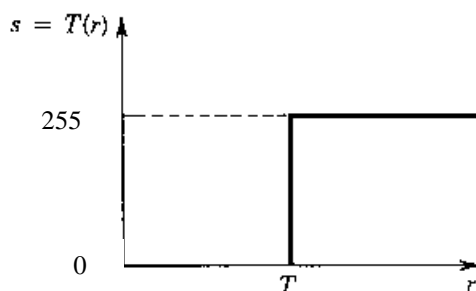


Figura 10: Função de binarização

O histograma da imagem, após sua binarização, terá apenas dois tons com número de pixels diferentes de zero.

Alguns autores se referem ao método de binarização como um método para separar os objetos do fundo (*background*) da imagem. Por exemplo, na figura 9, qualquer valor de T na região sem pixels entre as duas modas do histograma separa a imagem em fundo e objeto. Em estatística, denomina-se moda ao valor que expressa a concentração de uma variável: no caso, o tom de cinza mais freqüente em um histograma, isto é, o que tem mais pixels. Um histograma é bimodal quando apresenta duas modas ou dois picos de concentração nítidos.

É fundamental, no uso deste método para segmentação, a escolha do ponto de corte T . A ferramenta auxiliar usual nesta escolha é o histograma da imagem (gráfico onde se obtém o número de pixels da imagem associado a cada nível de cinza, como discutido na seção: "Modificações baseadas no histograma" no início deste texto).

A observação do histograma permite a localização do melhor valor de T para a imagem. Esta localização, no caso de segmentação entre fundo e objeto, é fácil se não houver tons de cinza em comum e existir uma região no histograma de nítida separação entre ambos. A localização, não havendo tal região, do melhor valor de T para a imagem é ainda possível e tão mais fácil quanto menos tons em comum tiverem ambos e quanto mais profundo e bem definido for, no histograma, o vale entre as modas.

O processo para determinação do ponto de limiarização (*threshold*) de um histograma pode ser **interativo** (escolhido pela interação com o usuário) ou **heurístico** (baseado em hipóteses sobre a melhor forma de separação entre os objetos).

A primeira forma sempre produz melhores resultados. Ao usá-la, devemos, por exemplo, examinar a curva do histograma à procura de um mínimo significativo. Em muitos sistemas de análise de imagens, um *preview* de como será a binarização quando o usuário escolhe determinado nível do

histograma se encontra disponível, o que facilita a escolha do melhor tom possível.

Limiarização sem uso do histograma

Como alternativa podemos definir o limiar (*threshold*) sem observar o histograma usando a seguinte estratégia.

a) Assume-se que os pixels dos quatro cantos da imagem pertencem ao fundo. Assim, a partir do valor do tom de cinza destes é definida uma média inicial para a intensidade do fundo. Para intensidade do objeto é assumido que todos os pixels centrais pertencem ao objeto. Considerando os pixels centrais da imagem, faz-se uma média inicial, que será considerada a intensidade do objeto.

b) A partir destes valores iniciais dos tons de fundo e objeto, entramos em um processo de limiarização iterativa, que consiste em modificar o ponto de limiarização T sucessivamente até existir uma diferença suficientemente pequena entre o novo valor do limiar, T^{i+1} , e o valor anterior, T^i .

$$T^{i+1} = (\mathbf{m}_{fundo}^i + \mathbf{m}_{objeto}^i) / 2 \quad (2)$$

onde: \mathbf{m}_{fundo}^i e \mathbf{m}_{objeto}^i são as médias para a intensidade do fundo e intensidade do objeto calculados como descritos em (a) inicialmente.

Após cada iteração são agregados mais pixels vizinhos aos iniciais para o cálculo da média. Ao se aplicar o tom de cinza T como limiar é importante algum processo de decisão sobre haver ou não uma boa separação entre os objetos (os melhores resultados neste sentido são também com a participação do usuário).

A base heurística deste processo se concentra na hipótese de que o objeto se encontra centrado na cena em análise. Esta técnica falhará sempre que isso não ocorrer, como é o caso das imagens das figuras 1 e 7 anteriores e da imagem "Detalhes" que segue. No caso da imagem "Lua" das páginas seguintes (figura 14), essa técnica produz resultados bem razoáveis.

Possibilidades de erros

É óbvio que o uso de segmentação por limiarização (*threshold*) pode produzir erros de classificação. A extensão desse erro depende de quantos tons em comum os objetos têm. No histograma da imagem, como pode ser visto na figura 11, isso corresponde a não existência de tons sem pixels entre as modas do objeto e fundo.

O histograma de um objeto sobre um fundo é aproximadamente a soma de dois histogramas: o histograma do fundo e o histograma do objeto individualmente. Se eles exibem intensidades de pixels em comum individualmente, essas serão somadas no histograma da imagem. Estes tons em comum serão impossíveis de serem separados pela limiarização. Seja lá qual for o valor escolhido para identificar o que é objeto e o que é fundo, sempre

haverá algum pixel do lado “errado” do valor de limiarização (*threshold*) escolhido. De modo que a seleção de um simples valor para limiar pode ser sempre inadequado, independente de qual método seja escolhido.



Figura 11: Histograma de uma cena como resultado da soma dos elementos presentes na cena

Diversas técnicas unem a observação do histograma com a regionalização ou identificação da área em análise na imagem para facilitar essa definição.

Quando a imagem tem dois (ou mais) objetos com cinzas bem diferentes sobre um fundo (*background*) e este é de cor bastante diferente dos objetos, podem ser usadas técnicas de binarização multinível (*multilevel thresholding*). Quando a imagem não tem objetos e fundo em tons característicos é mais difícil estabelecer os pontos de corte que isolam as regiões de interesse, a segmentação dos objetos, baseados apenas no histograma, é bem mais difícil ou até mesmo impossível.

A figura 12 mostra um esboço de como seria a aparência de um histograma tri-modal. São indicados aproximadamente que valores teriam as modas: neste histograma. As modas seriam os valores na horizontal correspondentes aos picos verticais, ou seja, os 3 valores indicados no gráfico pelas 3 setas horizontais. Será que neste histograma seria possível uma boa limiarização entre os objetos presentes na imagem?

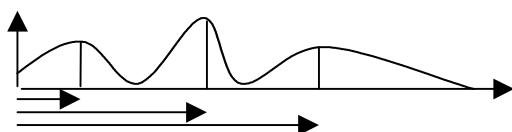


Figura 12: Um histograma tri-modal.

Como entre as normas deste histograma há regiões com quase nenhum pixel é possível uma boa

separação entre os 3 objetos, ou seja, uma boa limiarização.

Estratégia de partição para histogramas complexos

Uma extensão óbvia e simples para melhorar e mesmo possibilitar a limiarização é assumir que podemos usar ainda uma aproximação de histogramas separáveis se a imagem for dividida em regiões, cada uma representada pelo seu próprio histograma e ponto de limiarização.

Essa técnica é chamada de partição (*windowing*) e consiste em aplicar qualquer forma de limiarização já conhecida ou implementada apenas em uma parte da imagem por vez. Para cada parte tem-se, então, um limiar (ou *threshold*) diferente a ser aplicado a sub-regiões da imagem, tomando diferentes valores de região em região.

Assumindo que temos em cada parte o fundo e um objeto da cena, nós dividimos a imagem em regiões cada vez menores à procura de um histograma bimodal em cada região. Precisamos identificar, também, regiões que não exibem duas modas e assumirmos que estas são de todo objeto ou de todo fundo. Estas regiões não devem ser mais divididas.

A figura 13 mostra um exemplo de imagem que, se tratada regionalmente, poderá ser separável muito mais adequadamente.

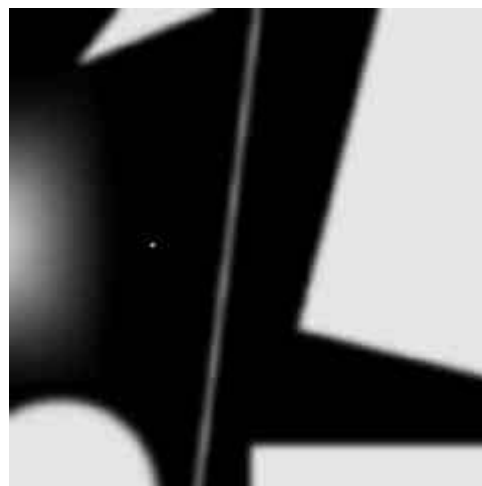


Figura 13: Imagem "Detalhes": exemplificando a separabilidade por regiões

Outra característica importante é o conhecimento do possível tamanho do objeto de maneira a não “forçar” mais subdivisões, onde a segmentação não deve existir, baseada em variações e tamanho de janelas insignificantes (o que só aumenta o tempo e custo de processamento).

Nesta técnica, a forma de partição da imagem pode ir de uma simples divisão da imagem em determinado número de partes iguais até técnicas de

subdivisão inteligentes e/ou baseadas em partições fractais.

Método Chow e Kaneko

Uma variação da técnica da seção anterior é o método proposto por Chow e Kaneko em 1972, que consiste em dividir a imagem em blocos de 64x64 pixels sem sobreposição e testar a bimodalidade do histograma, tentando associar um par de curvas Gaussianas (com médias m_1 e m_2 e desvio padrões S_1 e S_2) ao histograma em cada região [5]

Se a região não for bimodal, assume-se ser pertencente totalmente ao objeto ou ao fundo, sendo fixado um *threshold* para a região idêntico ao de uma região vizinha, bimodal. A estratégia final envolve agrupar regiões com mesmo *threshold*.

Estratégia de subtração para problemas de iluminação não constantes

Se o fundo e o objeto forem iluminados de maneira não uniforme, de modo que o tom de ambos durante a captura da imagem varie com a posição, a técnica de *threshold* particionada anterior pode ajudar, mas não daria o resultado desejado, pois, nunca haveria tons constantes em cada partição.

Pode-se tentar resolver este problema de variações na imagem pela captura e análise apenas da imagem do fundo sob mesma iluminação. Depois da obtenção da mesma imagem sem os objetos pode-se usar uma aproximação melhorada da imagem original. Esta aproximação melhorada é obtida pela **subtração** desta da imagem original e seu uso posterior no processo de segmentação.

A figura 14 apresenta a imagem "Lua" que é tipicamente um caso onde o processo de limiarização global não fornece bons resultados mas onde o objeto é separável do fundo depois de um processo de subtrair, da original, uma imagem apenas com o fundo sob mesmo tipo de iluminação (isto é sem o objeto).

Ao aplicar esta técnica encontramos algumas dificuldades pela diminuição do contraste nas regiões de borda. Muitas vezes, para resolver este problema empregamos métodos de ampliação do contraste pelo uso do histograma (como os discutidos no início desta seção), após a subtração e antes dos cálculos relacionados ao método de limiarização a ser usado.

Thresholding Automático

Existem várias maneiras de se escolher automaticamente o limiar para binarização ao invés de sugerir que o usuário defina esse limiar.

Existem métodos automáticos para a escolha e ajuste do ponto de corte [21,23] que usam o histograma da imagem ou a imagem propriamente dita como referência. Métodos que comparam conhecimentos prévios sobre parâmetros da imagem e obtêm vários pontos de corte também são conhecidos

[30,34]. Aqui, veremos apenas algumas dessas maneiras, selecionadas por serem as mais freqüentemente encontradas na literatura ou oferecerem os melhores resultados.



Figura 14: Imagem "Lua": um exemplo de iluminação não uniforme

Método de Bernsens

Esse método usa para valor de limiarização de um pixel a média entre o menor tom de cinza e o maior em uma vizinhança quadrada deste pixel desde que este valor seja maior que um valor de contraste preestabelecido [39,34].

Se o maior tom e o menor tom da vizinhança forem muito próximos, isto é, menor que o valor de contraste preestabelecido, é suposto que toda a vizinhança pertence a uma classe única: objeto ou imagem de fundo.

É preciso uma análise prévia do tipo de imagem a ser tratada para que estes parâmetros (tamanho da vizinhança e menor contraste) sejam adequadamente definidos.

Método de Otsu

Este método baseia-se em uma análise discriminante e na otimização do tom escolhido para limiar [28].

A operação de limiarização pode ser vista como a repartição dos pixels da imagem em duas classes C_0 e C_1 (que serão, por exemplo, o objeto e o fundo) em um nível de cinza t . Assim a classe C_0 será composta por níveis de cinza de 0 até t , ou seja :

$$C_0 = \{ 0, 1, 2, \dots, t \}$$

A classe C_1 será composta pelos níveis de cinza $t+1$, $t+2$... até $l-1$, onde " l " é o nível de cinza da imagem:

$$C_1 = \{ t+1, t+2, \dots, l-1 \}$$

Supondo-se que $l = 256$, tem-se:

$$C_l = \{t+1, t+2, \dots, 254, 255\}$$

Se a imagem for multimodal, os pixels da imagem são divididos em classes C_0, C_1, \dots, C_{n-1} , onde n é o número de modas da imagem e identifica-se os tons dos pixels nessas classes, por exemplo

$$C_0 = \{0, 1, 2, 3, 4, \dots, t_1\},$$

$$C_2 = \{t_1+1, t_1+2, \dots, t_2\}, \dots,$$

$$C_{n-1} = \{t_{n-1}, t_{n-1}+1, t_{n-1}+2, \dots, M-1\},$$

onde M é o número de diferentes intensidades na escala de cinza da imagem (geralmente 256).

Por simplificação, inicialmente podemos admitir um histograma bimodal, onde teremos apenas um valor de *threshold* e, portanto, apenas duas classes: C_0, C_1 . Para melhor entendimento do método utiliza-se os seguintes símbolos:

T = Toda a imagem (classe C_0 e C_1)

W = Valores dentro (*within*) de determinada classe (C_0 , ou C_1)

B = Valores entre (*between*) classes

Lembrando que a variância, denotada por σ^2 , é definida como:

$$\sigma^2 = S(x - m)^2 p(x) \quad (2)$$

onde m é a média e $p(x)$ é a função densidade de probabilidade da variável x , um limiar ótimo é obtido minimizando uma das funções abaixo (os resultados são equivalentes),

$$I = \frac{S_B^2}{S_W^2}, h = \frac{S_B^2}{S_T^2}, T = \frac{S_T^2}{S_W^2},$$

onde: S_T^2 = Variância Total, S_W^2 = Variância dentro da classe (internas das classes) e S_B^2 = Variância entre classes). Nestas h é a mais simples de ser minimizada e o *threshold* ótimo pode ser obtido através de:

$$t^* = \min h (t \in G),$$

sendo $G = \{\text{conjunto dos tons de cinza, "gray levels"}\}$.

A função densidade de probabilidade da variável x , $p(x)$, pode ser obtida pensando em cada coluna do histograma como uma probabilidade.

Fazendo-se $n_i = h(i)$, onde $h(i)$ é a frequência do tom de cinza i , obtida do histograma, podemos ter o número total de pixels fazendo um somatório em todos os tons i , de 0 até 255, $N = \sum n_i$ ou seja N = total de pixels da imagem.

Se esta soma parar em um determinado tom, t , (teremos o número total de pixels da imagem até um nível de cinza t , dividido pelo número total de pixels) e a fração $p_i = (n_i / N)$ corresponde à probabilidade de ocorrência na imagem de tons menores ou iguais ao tom t .

A fração $p_i = (n_i / N)$ representa o percentual de cada tom de cinza no total da imagem, onde p_i é a

probabilidade de ocorrer a intensidade de i , ou seja, o número de pixels daquele tom presente na imagem dividido pelo número total de pixels N .

Abrangências do Método Otsu

A segmentação de objetos em uma imagem, usando a determinação dos valores de *threshold* adequados não deve ser restringida pelas possibilidades do método. Há métodos que só são adequados na definição de um limiar. Outros se baseiam em informações espaciais ou estatísticas, que requerem histogramas constituídos apenas de distribuições Gaussianas. As propriedades importantes, que caracterizam o método de Otsu, são ele não fazer qualquer hipótese sob a forma da distribuição e poder ser usado para qualquer número de limiares (e portanto objetos).

Uma boa técnica de thresholding deve ser capaz de segmentar em uma imagem os diferentes objetos presentes. Diversos métodos são extensões do método de Otsu e permitem segmentar recursivamente a imagem. Nas imagens digitais, a uniformidade dos objetos tem papel importante na sua separabilidade do fundo. O método de Otsu para o thresholding de imagens é eficiente em função da medida de uniformidade das duas classes presentes.

Automatização baseada na moda do histograma e critérios discriminantes

Essa automatização é baseada na decomposição modal utilizando o conceito de otimização. Dessa forma, a decomposição modal separa o histograma de acordo com os tons que representam os seus picos (ou modas), de forma a podermos achar entre os picos relevantes o melhor ponto de thresholding.

Para que possamos trabalhar matematicamente com esses valores, é necessário que modelemos funções matemáticas capazes de representar com fidelidade a forma do histograma que representa as intensidade de cinza e suas frequências. Isso pode ser feito através da parametrização do histograma por um modelo auto-regressivo, por exemplo, utilizando-se a transformação inversa discreta de Fourier do histograma.

A transformação inversa de Fourier do histograma pode ser aproximada, utilizando-se a série de Taylor, que fornece um valor aproximado para $W = e^{-i(2\pi/M)}$, possibilitando que encontremos os valores W^k , variando-se o valor k de 0 a 255 (representando os 256 tons de cinza) para cada valor de M (entre 0 e 255)..

Um par complexo representa um senóide da moda (um pico do histograma), sendo que os valores mais significativos estão sempre no início da série, tornando-se rápida a associação e a identificação das modas relevantes.

Essa modelagem matemática consegue dar bons resultados, fornecendo funções que expressam corretamente os contornos do histograma em relação a seus picos; porém, não há um casamento entre a curva e o histograma real nos vales, o que de certa forma acaba por dificultar esse processo de thresholding.

Com interferência humana, torna-se mais fácil resolver o problema de quantas modas ou picos significativos de fato há no histograma; no entanto, a intenção desse método é inferir automaticamente o número de picos relevantes e definir os limiares correspondentes.

Outras metodologias

Outros métodos de localização dos pontos de thresholding utilizam medidas desse espalhamento de intensidades da escala de cinzas da imagem (variância), medindo a maior variância de dados em relação a um pico relevante e a seu vizinho adjacente, encontrando entre essas duas medidas o fundo do vale, ponto onde deve ser feito o thresholding. Todos os dados necessários ao *thresholding* automático são calculados a partir daí.

Por exemplo, se for desejado separar uma imagem em duas classes, define-se duas variáveis:

$W_0 = S(p_i)$ na classe C_0 , ou seja, variando-se i de 1 até t ; e

$W_1 = S(p_i)$ na classe C_1 , ou seja, variando-se i de $t+1$ até $M-1$.

Fazendo $m_0 = S(ip_i)$, com i variando de 1 a t , teremos a média, ou valor esperado para a classe C_0 . Fazendo $m_1 = S(ip_i)$, com i variando de $t+1$ a $M-1$, teremos a média, ou valor esperado para a classe C_1 . O valor esperado ($E(x)$) para toda a série será expresso por:

$$m_i = S(ip_i), \text{ com } i \text{ variando de } 1 \text{ a } M.$$

A variância entre classes, será expressa por:

$$s_B^2 = W_0(m_0 - m_i)^2 + W_1(m_1 - m_i)^2$$

A fração $p_i = (n_i / N)$ representa o percentual de cada tom de cinza no total da imagem, onde p_i é a probabilidade de ocorrer a intensidade de i , ou seja, o número de pixels daquele tom presente na imagem dividido pelo número total de pixels. N é o número total de pixels considerando todos os níveis de cinza.

Uma extensão é fazer uma estimativa de duas distribuições de probabilidade, uma do objeto, H_o , e uma do fundo, H_f . O limiar é escolhido de forma que a soma $H = H_f + H_o$ seja máxima.

Uma maneira de estimar essas funções, que foi proposta por Pun é definir essas funções como as entropias do objeto (H_o) e do fundo da imagem (H_f). Na proposta de Pun elas são definidas em função de um nível de cinza t como:

$$H_f(t) = - \sum_{i=0}^t p_i \ln p_i$$

$$H_o(t) = - \sum_{i=t+1}^{M-1} p_i \ln p_i$$

Kapur estendeu a proposta de Pun, fazendo com que as entropias fossem definidas como:

$$H_f(t) = - \sum_{i=0}^t \frac{p_i}{P_t} \ln \left(\frac{p_i}{P_t} \right)$$

$$H_o(t) = - \sum_{i=t+1}^{M-1} \frac{p_i}{1 - P_t} \ln \left(\frac{p_i}{1 - P_t} \right)$$

$$P_t = \sum_{i=0}^t p_i$$

Os picos do histograma são chamados de modas do histograma (*histogram modes*), sendo cada pico relevante e suas redondezas chamado de agrupamento ou cluster (*histogram cluster*). O agrupamento ou cluster fornece informações importantes sobre a imagem.

A amplitude de um pico pode ser modelada de forma empírica, sendo uma boa medida ser considerada entre 10% e 20%.

Para valores inferiores ao limite de 10%, teríamos limiares em excesso, isto é, muitas classes diferentes, possivelmente encontrando falsos vales no histograma, enquanto que para valores superiores ao limite estabelecido poderíamos estar desprezando vales existentes e agrupando dados pouco homogêneos, dado que 20% de poder de contribuição é um valor significativo.

Para procedermos nesta técnica, um caminho é achar o centro do pico, isto é, seu valor mais alto, que indica de fato a moda, sendo este o valor para o qual devem convergir os outros valores desse agrupamento (*cluster*) da imagem após o thresholding.

A variância indica o espalhamento dos dados em torno da média, ou valor esperado; em outras palavras, mede a dispersão de determinados dados em relação a um ponto. Para o caso de imagens, uma pequena variância indica uma maior homogeneidade de intensidades (em uma escala de cinza), enquanto que uma grande variância (histograma bastante espalhado) indica várias intensidades diferentes, ou seja, menos tons homogêneo, em torno de uma intensidade mais freqüente (isto é a intensidade média).

Considerando uma imagem bimodal, o thresholding automático seria feito entre dois picos ou modas, no fundo do vale entre esses picos (normalmente uma imagem é multimodal, possui ruídos, vales mal definidos e quase planos, dificultando o trabalho)

Para podermos encontrar o ponto de thresholding, inicialmente devemos definir o agrupamento ideal (*histogram cluster*), que é representado através de um conjunto de níveis de cinza,

$$G_{Sl} = \{h_l^L, \dots, h_l^H\}$$

Os valores limites \mathbf{h}_1^L e \mathbf{h}_1^H são encontrados, tomando-se uma distância correspondente à metade da largura do pico relevante a partir de seu centro. O que agiliza computacionalmente esse método, pois o domínio de t está compreendido entre \mathbf{h}_1^H e \mathbf{h}_2^L , isto é, entre o valor mais alto do conjunto $G_{S1}=\{\mathbf{h}_1^L, \dots, \mathbf{h}_1^H\}$, que representa um dos *histogram clusters* (com níveis baixos de intensidade na escala de cinza – próximo ao branco) e o valor mais baixo do conjunto $G_{S2}=\{\mathbf{h}_2^L, \dots, \mathbf{h}_2^H\}$, que representa o outro *histogram cluster* (com níveis altos de intensidade na escala de cinza – próximo ao preto).

Para obtermos o thresholding t^* , achamos t que satisfaça a condição de $\mathbf{S}_B^2(t^*)=\text{máxima } \mathbf{S}_B^2(t)$, calculando-se o valor de $\mathbf{S}_B^2(t)$ para os pontos $t \in G_S^*$, tal que $G_S^*=\{\mathbf{h}_1^H, \dots, \mathbf{h}_2^L\}$.

Essa última colocação, em relação à restrição do conjunto utilizado estar entre as duas bordas dos picos para se encontrar o ponto de thresholding pode ser entendido como o diferencial deste método, que o torna viável computacionalmente para múltiplos thresholdings.

O exemplo anterior para um histograma bimodal, com apenas um thresholding pode ser estendido para múltiplos thresholdings; no entanto os valores a serem variados serão tantos quantos forem os números de thresholdings necessário, t_1, t_2, \dots, t_n que variarão entre os limites definidos por $S_1^*, S_2^*, \dots, S_n^* = \{\mathbf{h}_n^H, \dots, \mathbf{h}_{n+1}^L\}$, para as $n+1$ classes ou regiões consideradas homogêneas da imagem.

Esse método se mostra eficaz em muitos casos. A fidelidade e confiabilidade do modelo está relacionada com a largura dos vales do histograma.

Dessa forma, para o caso em questão, comparando esse método com o de Otsu, observa-se que pode haver falha em encontrar os histogram clusters. Também, o método de Otsu torna-se menos confiável à medida que aumenta o número de modas significativas. Neste método, a segmentação mostra-se eficiente na definição dos agrupamentos do histograma (*histogram clusters*) e também na obtenção dos limiares entre os grupos (*thresholdings*), conforme constatou-se em implementações e exemplos testados.

Parte 2:

Transformada de Fourier

A representação das funções por séries de Fourier (desenvolvidas por Jean Baptiste Joseph Fourier, matemático e físico francês, que viveu de 1768 a 1830), através da expansão em senos e cossenos são importantes para a solução analítica e numérica de equações diferencial, bem como, para a análise e tratamento de sinais.

Nas filtragens usadas em processamento de imagens a transformada de Fourier é uma ferramenta fundamental. Aqui veremos alguns conceitos

importantes para o entendimento da transformada de Fourier. Começaremos pela transformada de uma função contínua de uma ou duas dimensões, passando por Transformada Discreta de Fourier, até chegar em Transformada Rápida de Fourier.

Dada uma função $f(x)$ contínua e definida em todo o espaço dos reais, sua transformada de Fourier, $\mathfrak{F}\{f(x)\} = F(u)$ é definida como:

$$\mathfrak{F}\{f(x)\} = F(u) = \int_{-\infty}^{\infty} f(x) \exp[-j 2\pi u x] dx$$

onde $j = \sqrt{-1}$ ou $j^2 = -1$. (3)

Dada uma função $F(u)$ contínua e definida em todo o espaço dos complexos, sua transformada inversa de Fourier, $\mathfrak{F}^{-1}\{F(u)\} = f(x)$, é definida pela expressão:

$$\mathfrak{F}^{-1}\{F(u)\} = f(x) = \int_{-\infty}^{\infty} F(u) \exp[j 2\pi u x] du$$

como anteriormente $j = \sqrt{-1}$ ou $j^2 = -1$. (4)

A transformada de Fourier, $\mathfrak{F}\{f(x)\}$ muda o espaço de domínio dos parâmetros dos números reais para os complexos (de $x \in \mathbb{R}$ passa-se para $u \in \mathbb{C}$).

Os números complexos são números com uma parte real e outra imaginária, na forma: $z = a + bj$, onde a e b são números reais. Como os reais, os números complexos podem ser somados, subtraídos, multiplicados e divididos, resultando dessas operações outro número complexo. Essas operações podem ser representadas pelas expressões:

$$(a + bi) + (c + di) = (a + c) + (b + d)i$$

$$(a + bi) - (c + di) = (a - c) + (b - d)i$$

$$(a + bi)(c + di) = (ac - bd) + (ad + bc)i$$

$$\frac{ac + bd}{c^2 + d^2} + \left(\frac{bc - ad}{c^2 + d^2} \right) i$$

Podemos representar os números complexos graficamente pelo Diagrama de Argand-Gauss, que mostra o Plano Complexo, figura 15. Essa também é chamada forma cartesiana dos complexos com a parte real a (no eixo horizontal) e imaginária b (no eixo vertical ou imaginário).

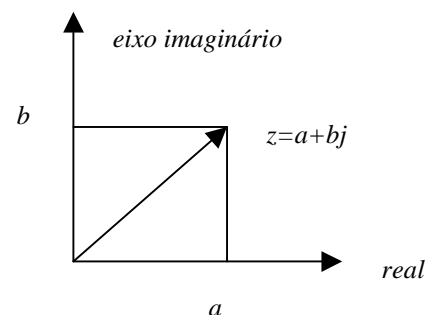


Figura 15: Plano Complexo

A forma polar de um número complexo é descrita pelo seu módulo ou amplitude e seu ângulo de fase, sendo:

$$r = \sqrt{a^2 + b^2} \rightarrow \text{módulo do número ou amplitude}$$

$$q = \arctg\left(\frac{b}{a}\right) \rightarrow \text{ângulo de fase}$$

$$a = r \cos q$$

$$b = r \sin q$$

Lembrando das relações trigonométricas:

$$\cos(a + b) = \cos a \cos b - \sin a \sin b$$

$$\sin(a + b) = \sin a \cos b + \cos a \sin b$$

a multiplicação na forma polar pode ser representada como:

$$\begin{aligned} r_1(\cos q_1 + j \sin q_1) \cdot r_2(\cos q_2 + j \sin q_2) \\ = r_1 r_2 (\cos(q_1 + q_2) + j \sin(q_1 + q_2)) \end{aligned}$$

A divisão na forma polar fica:

$$\begin{aligned} \frac{r_1(\cos q_1 + j \sin q_1)}{r_2(\cos q_2 + j \sin q_2)} = \\ \frac{r_1}{r_2} (\cos(q_1 - q_2) + j \sin(q_1 - q_2)) \end{aligned}$$

O número complexo conjugado do complexo $a + bj$ é definido como o número $a - bj$.

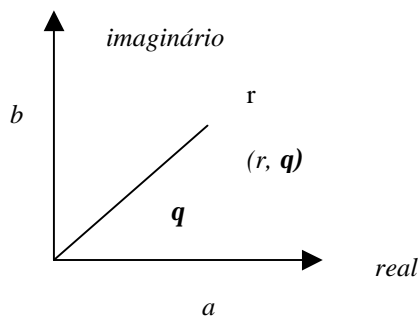


Figura 16: Um número complexo na forma polar (amplitude r e ângulo)

O Teorema de Moivre é um resultado importante para as operações de radiciação e potenciação de complexos, ele diz que se $p=1/n$ é real e n é um inteiro positivo, então:

$$\begin{aligned} [r(\cos q + j \sin q)]^{\frac{1}{n}} = \\ r^{\frac{1}{n}} \left(\cos \frac{q + 2kp}{n} + j \sin \frac{q + 2kp}{n} \right) \end{aligned}$$

onde na expressão acima k é um inteiro qualquer. Observe que, por esta expressão, a n -ésima raiz de um número complexo pode ser obtida fazendo-se $k=0, 1, 2, \dots, n-1$.

Pelas **Identidades de Euler** temos ainda:

$$e^{jq} = \cos q + j \sin q$$

$$e^{-jq} = \cos q - j \sin q$$

ou

$$\sin q = \frac{e^{jq} - e^{-jq}}{2j}$$

$$\cos q = \frac{e^{jq} + e^{-jq}}{2}$$

e

$$e^{j\pi} = -1$$

onde e representa a função exponencial, $\exp[\]$, ou seja, potências usando como base o número e . Esse número é a base dos logaritmos naturais ou Neperianos. Os logaritmos comuns tem base 10, mas nos Neperianos a base é:

$$e = \lim_{n \rightarrow \infty} \left(1 + \frac{1}{n} \right) = 2,71828 \dots$$

A forma polar dos números complexos, usando as expressões anteriores pode ser escrita mais simplificadamente da forma:

$$r(\cos q + j \sin q) = re^{-jq}$$

Essa última forma facilita a representação das operações de multiplicação, divisão, potenciação e radiciação entre os números complexos, que podem ser representadas, como:

$$r_1 e^{jq_1} \cdot r_2 e^{jq_2} = r_1 \cdot r_2 e^{j(q_1 + q_2)}$$

$$\text{pois } x^p \cdot x^q = x^{p+q}$$

$$\frac{r_1 e^{jq_1}}{r_2 e^{jq_2}} = \frac{r_1}{r_2} e^{j(q_1 - q_2)}$$

$$\text{pois } \frac{x^p}{x^q} = x^{p-q}$$

$$(re^{jq})^p = r^p e^{jpq}$$

$$\text{pois } (x^p)^q = x^{p \cdot q}$$

$$(re^{jq})^{\frac{1}{n}} = [re^{j(q+2kp)}]^{\frac{1}{n}} = r^{\frac{1}{n}} e^{\frac{j(q+2kp)}{n}}$$

$$\text{pois } (\sqrt[n]{x}) = x^{\frac{1}{n}}$$

A transformada de Fourier transforma a imagem para o domínio da frequência. Os valores dentro da função exponencial, $\exp[\]$, considerando as Identidades de Euler, podem ser escritos como senos e cossenos:

$$\exp[j 2\pi u x] = e^{j 2\pi u x} = \cos(2\pi u x) + j \sin(2\pi u x)$$

$$\exp[-j 2\pi u x] = e^{-j 2\pi u x} = \cos(2\pi u x) - j \sin(2\pi u x)$$

A exponencial acima, usada na transformada de Fourier, combina senos e cossenos de $[2\pi u x]$ em

uma só expressão. Como estas funções, senos e cossenos são, na realidade ondas e nos fenômenos ondulatórios a frequência é a quantidade de ondas que passa em uma região em um determinado tempo (cada onda senoidal e cossenoidal se repete a cada 360° ou 2π). O que acaba caracterizando a frequência destas ondas são os valores que estão sendo multiplicados a 2π , ou seja, o u ou o x nas expressões da Transformada de Fourier.

Chama-se **par de Fourier** o conjunto definido pelas equações (3) e (4): $(f(x), F(u))$. A utilidade da transformada de Fourier consiste em possibilitar que se analise uma função no domínio do espaço através da sua forma no domínio da frequência. A transformada inversa funciona levando uma função do domínio da frequência, novamente para o do espaço. Ao ir de uma representação para a outra nada da informação é modificada. A função pode ser analisada então pela forma que for mais adequada.

Para exemplificar como se encontra esse par suponha que se tenha a função degrau mostrada na figura que segue, ou seja, suponha $f(x)$ dada por: $f(x) = A$ (uma constante real), para $x < X$

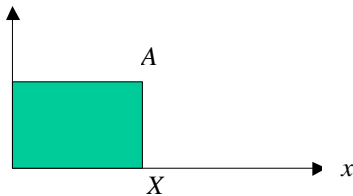


Figura 17: Função degrau usada para exemplificar o cálculo da transformada de Fourier

Usando a equação (3) temos:

$$\mathfrak{F}\{A\} = F(u) = \int_{-\infty}^{\infty} A \exp[-j 2\pi u x] dx$$

$$\mathfrak{F}\{A\} = \int_0^X A \exp[-j 2\pi u x] dx$$

$$\mathfrak{F}\{A\} = \frac{-A}{j 2\pi u} \exp[-j 2\pi u x]_0^X$$

$$\mathfrak{F}\{A\} = \frac{-A}{j 2\pi u} [\exp[-j 2\pi u X] - \exp[0]]$$

$$\mathfrak{F}\{A\} = \frac{-A}{j 2\pi u} \exp[-j 2\pi u X] - 1$$

Lembrando que a função exponencial tem periodicidade 2π :

$$\exp j [2k\pi + q] = \exp j[q]$$

e que: $a^p a^q = a^{p+q}$ ou $a^{p-p} = a^0 = 1$. Lembrando ainda que $a^p a^p = a^{2p}$ podemos escrever a expressão acima de forma mais facilmente interpretável:

$$\begin{aligned} \mathfrak{F}\{A\} &= \frac{-A}{j 2\pi u} [e^{-j 2\pi u X} e^{-j 2\pi u X} - 1] \\ &= \frac{-A}{j 2\pi u} [e^{-j 2\pi u X} [e^{-j 2\pi u X} - e^{j 2\pi u X}]] \\ &= \frac{-A}{j 2\pi u} e^{-j 2\pi u X} \text{sen}(2\pi u X)(-2j) \\ &= \frac{A}{\pi u} e^{-j 2\pi u X} \text{sen}(2\pi u X) \end{aligned}$$

neste desenvolvimento foi usado também que:

$$\text{sen} q = \frac{-1}{2j} [e^{-jq} - e^{jq}]$$

Assim o **par de Fourier** da função constante A é: $(A, \hat{A}/A)$, com \hat{A}/A definida pela expressão anterior. Isso nos permite dizer que a mesma função tem dois modos de representação, um no **domínio do espaço**: $x \times f(x)$, e o outro no **domínio da frequência**: $u \times F(u)$.

No domínio da frequência a função é em geral complexa, tem uma parte Real e outra Imaginária, isto é, passa a ser descrita como um número complexo. Ou seja, um número que tem duas partes: uma real e outra imaginária (multiplicada por j). Como o mostrado na figura 15. Chamamos a parte real de $R(u)$ e a imaginária (multiplicada por j), de $I(u)$.

Como já mencionado, amplitude de um número complexo é o seu módulo, o r no gráfico da figura 16 anterior. O ângulo de fase é o ângulo que o ponto, no plano complexo (x, y) , faz com a horizontal, ou eixo real, o q no gráfico da figura 16. No caso da Transformada de Fourier essa **amplitude** e esse **ângulo de fase** são dados por:

$$|F(u)| = \sqrt{R^2(u) + I^2(u)}$$

$$q(u) = \tan^{-1} \left[\frac{I(u)}{R(u)} \right]$$

Estas expressões são úteis no caso de se calcular a transformada de Fourier de uma função de uma variável como nas expressões acima.

Espectro de amplitude ou espectro de Fourier de uma função é o gráfico de sua amplitude *versus* cada frequência. Ou seja, a descrição de qual é a amplitude $|F(u)|$ para cada frequência u . Assim o Espectro de amplitude ou espectro de Fourier da função degrau, do exemplo anterior acima, é obtida da expressão:

$$|F(u)| = \left| \frac{A}{p u} \sin(p u X) \exp[-j p u X] \right|$$

cujo gráfico é mostrado na figura 18. Repare que é **na origem dos eixos de freqüência onde a maior intensidade das informações se concentra** (isso é **geralmente** o que ocorre em qualquer função quando transformada, sendo importante principalmente em imagens).

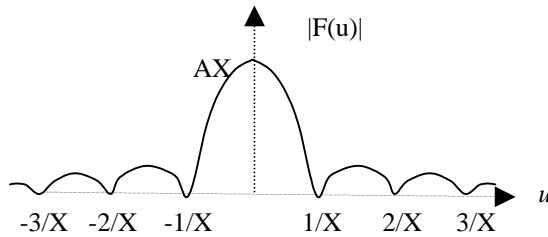


Figura 18: Espectro de Fourier da função Degrau

Transformadas Bidimensionais

Quando passa-se para funções de duas variáveis a integral simples, que define a transformada de Fourier, torna-se dupla:

$$\begin{aligned} \mathfrak{F}\{f(x, y)\} &= F(u, v) \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \exp[-j 2\pi (ux + vy)] dx dy \end{aligned} \quad (5)$$

o mesmo ocorrendo com a expressão da transformada inversa:

$$\begin{aligned} \mathfrak{F}^{-1}\{F(u, v)\} &= f(x, y) = \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(u, v) \exp[j 2\pi (ux + vy)] du dv \end{aligned} \quad (6)$$

e também para as funções que definem a amplitude e o ângulo de fase:

$$\begin{aligned} |F(u, v)| &= \sqrt{R^2(u, v) + I^2(u, v)} \\ q(u, v) &= \tan^{-1} \left[\frac{I(u, v)}{R(u, v)} \right] \end{aligned}$$

Espectro de amplitude ou espectro de Fourier de uma função de duas variáveis, é o gráfico de $|F(u, v)|$ para cada par de freqüência (u, v) , isto é, o gráfico de sua amplitude *versus* as duas variáveis. Neste último caso, o gráfico deve ser representado em 3D ou em 2D com tons de cinza.

Como exemplo em 2D, vamos considerar uma extensão do exemplo anterior. A função mostrada na figura 19, é a função degrau em duas direções, isto é, a função : $f(x, y) = A$ para $x < X$ e $y < Y$.

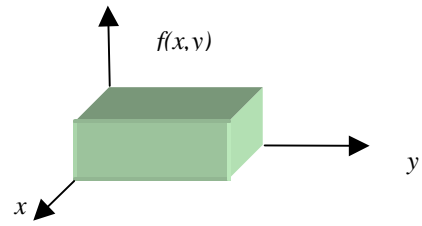


Figura 19: Função degrau 2D usada para exemplificar o calculo da transformada de Fourier

Usando as expressões da transformada de Fourier 2D teremos:

$$\begin{aligned} F(u, v) &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \exp[-j2\pi(ux + vy)] dx dy \\ &= A \int_0^X \exp[-j2\pi ux] dx \int_0^Y \exp[-j2\pi vy] dy \\ &= A \left[\frac{e^{-j2\pi ux}}{-j2\pi u} \right]_0^X \left[\frac{e^{-j2\pi vy}}{-j2\pi v} \right]_0^Y \\ &= \frac{A}{-j2\pi u} [e^{-j2\pi uX} - 1] \frac{1}{-j2\pi v} [e^{-j2\pi vY} - 1] \\ &= AXY \left[\frac{\sin(\pi uX) e^{-j\pi uX}}{(\pi uX)} \right] \left[\frac{\sin(\pi vY) e^{-j\pi vY}}{(\pi vY)} \right] \end{aligned}$$

O Espectro de Fourier da função Degrau 2D é obtido de forma semelhante ao do caso 1D anterior e pode ser visto na figura 20 na forma 3D, onde $|F(u, v)|$ é mostrado como terceiro eixo:

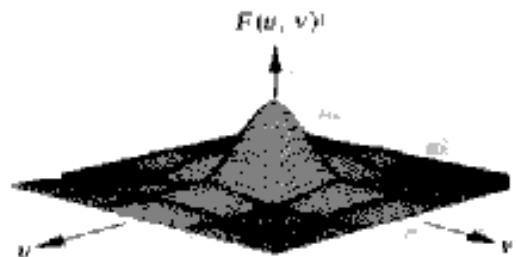


Figura 20: Espectro de Fourier da função Degrau 2D, na forma tridimensional

Outra forma muito usual, principalmente em imagens, de mostrar o gráfico do espectro de Fourier de funções 2D, é utilizando ao invés do terceiro eixo, tons de cinza para mostrar $|F(u, v)|$. Esta forma é mostrada na figura 21.

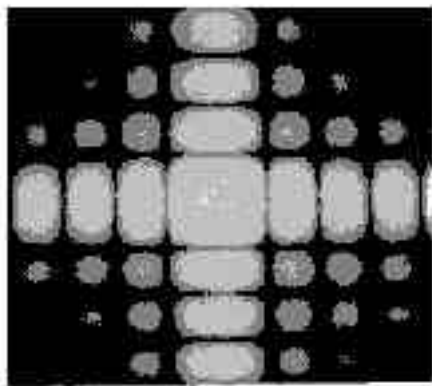


Figura 21: Espectro de Fourier da função Degrau 2D, na forma de tons de cinza para o valor $|F(u,v)|$.

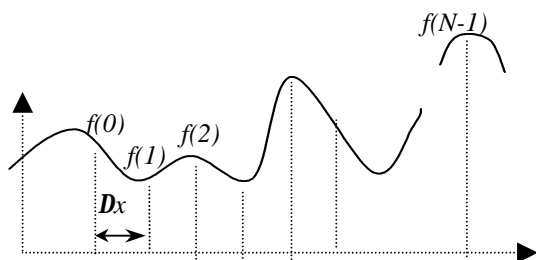
No exemplo anterior o espectro tem valores que vão de mais a menos infinito em todas as direções. Mas há muitos casos onde os valores significativos apenas ocorrem entre um conjunto de frequências mínima e máxima: (u_{\min}, u_{\max}) e (v_{\min}, v_{\max}) , ou (u_{\min}, v_{\min}) e (u_{\max}, v_{\max}) . Esta diferença entre os valores máximos e mínimos é que se chama **largura de banda**: $|u_{\max} - u_{\min}|$ ou $|v_{\max} - v_{\min}|$ ou ainda $|v_{\max} - u_{\min}|$.

No exemplo acima, o espectro teve uma banda infinita em todas as direções, mas longe da origem $(0, 0)$ a intensidade de $|F(x, y)|$ decaiu rapidamente. Este resultado (como no caso 1D) é bastante típico, de modo que é na origem dos eixos de frequência onde geralmente a maior parte das informações se concentra, principalmente em imagens.

Assim, se a Figura 19 anterior representasse o tom de uma imagem formada por todos os pixels de uma única cor, por exemplo 255, seu espectro de Fourier seria a forma mostrada. Como a intensidade de $|F(x,y)|$ decai exponencialmente, seria visto apenas um ponto na origem. Por essa razão, muitas vezes na representação em 2D em tons de cinza transforma-se o valor da intensidade por alguma função amplificadora, de modo a haver diferença entre imagens.

Transformada de Funções Discretas

Se funções discretas, ou seja, definidas apenas a intervalos constantes de Dx , da variável forem usadas: $f(x) = f(n) = f(x_0 + n Dx)$ como mostrado no esquema que segue:



Teremos uma sequência ou uma tabela de valores onde a função é definida: $f(0), f(1), f(2), \dots, f(N-1)$, de modo que poderemos usar só estes valores conhecidos no cálculo da Transformada de Fourier e passaremos a ter um par discreto de Fourier dado por:

$$F(u) = \frac{1}{N} \sum_{n=0}^{N-1} f(n) e^{-2\pi j (un/N)}$$

para $u=0, 1, 2, \dots, N-1$ e (7)

$$f(n) = \sum_{u=0}^{N-1} F(u) e^{2\pi j (un/N)}$$

para $n=0, 1, 2, \dots, N-1$. (8)

Os incrementos ou intervalos em u e x se relacionam por:

$$N Dx = 1/Du$$

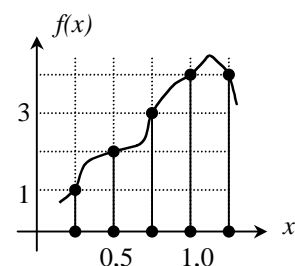
A expressão (7) descreve a Transformada de Fourier de um função discreta e a expressão (8) sua inversa. As expressões (7) e (8) definem o par da Transformada Discreta de Fourier (DFT) em uma dimensão.

Para se avaliar quantas amostras o domínio de frequência u terá em relação ao domínio espacial x na forma discreta n , analisa-se a sequência $F(0)$, $\{Complexo \{F(u)\}, u = 1, \dots, N/2-1\}$, $\{Complexo \{F(u)\}, u = N/2+1, \dots, N-1\}$, $F(N/2)$ que define completamente a DFT da sequência real $f(x)$. na forma discreta, isto é, $f(n)$.

A sequência acima mostra que, na forma unidimensional o espectro de Fourier necessita de N posições para armazenamento, mesmo tamanho da sequência original. Os coeficientes $a_u + ib_u$, onde $0 \leq u \leq N-1$ são os conteúdos da DFT e u é a unidade de frequência, ou seja, faz parte das frequências dos senos e cossenos de $-2\pi j ux/N$ da DFT

Um aspecto importante é que diferentemente do caso contínuo, no caso discreto a transformada de Fourier de uma **função sempre existe**.

Novamente, para ficar claro, vejamos um exemplo concreto. Imagine que se queira calcular a transformada de Fourier da função discreta mostrada no esquema que segue, que é definida a intervalos de $1/4=0,25$:



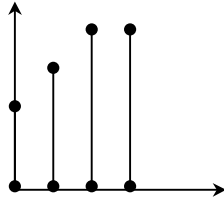


Figura 22: Exemplo de uma função discreta

Reorganizando a ordem da sequência, podemos dizer que a $f(x) = f(n)$ é definida nos 4 pontos: 0, 1, 2 e 3, onde tem os valores $f(0)=2$, $f(1)=3$, $f(2)=4$, e $f(3)=4$ (figura 22).

De modo que, usando as expressões da transformada discreta 1D teremos:

$$\begin{aligned} F(0) &= \frac{1}{4} \sum_{n=0}^3 f(n) e^{0} \\ &= \frac{1}{4} (f(0) + f(1) + f(2) + f(3)) \\ &= \frac{1}{4} (2 + 3 + 4 + 4) = 3,25 \end{aligned}$$

de maneira semelhante:

$$\begin{aligned} F(1) &= \frac{1}{4} \sum_{n=0}^3 f(n) e^{-2\pi j(n/4)} \\ &= \frac{1}{4} (2 - 3j - 4 + 4j) = \frac{1}{4} (-2 + j) \\ F(2) &= \frac{1}{4} \sum_{n=0}^3 f(n) e^{-\pi j n} \\ &= \frac{1}{4} (2e^0 + 3e^{-j\pi} + 4e^{j2\pi} + 4e^{j3\pi}) = \frac{1}{4} (-1) \end{aligned}$$

e

$$F(3) = -\frac{1}{4} (2 + j)$$

Para o caso de duas variáveis discretas pode-se ampliar o conceito da DFT unidimensional [3]. Se u e v são as variáveis de frequência, na forma bidimensional teremos, para o par de Fourier :

$$F(u, v) = \frac{1}{NM} \sum_{n=0}^{M-1} \sum_{m=0}^{N-1} f(n, m) e^{-2\pi j (un/M + vm/N)} \quad (9)$$

sendo $u=0, 1, 2, \dots, M-1$ e $v=0, 1, 2, \dots, N-1$, e

$$f(n, m) = \frac{1}{NM} \sum_{n=0}^{M-1} \sum_{m=0}^{N-1} F(n, m) e^{2\pi j (un/M + vm/N)} \quad (10)$$

onde $m=0, 1, 2, \dots, M-1$ e $n=0, 1, 2, \dots, N-1$. A relação entre os incrementos das amostras é

$$N \Delta x = 1/\Delta v \quad M \Delta y = 1/\Delta u$$

No caso de funções com mesmo número de amostras nas duas direções: $M=N$ e $1/N$ aparece em $f(n, m)$ e $F(u, v)$:

$$F(u, v) = \frac{1}{N} \sum_{n=0}^{N-1} \sum_{m=0}^{N-1} f(n, m) e^{-2\pi j (un+vm)/N} \quad (11)$$

$$f(n, m) = \frac{1}{N} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} F(u, v) e^{2\pi j (un+vm)/N} \quad (12)$$

Para imagens digitais, a $f(x, y)$ é descrita apenas nos valores inteiros n, m que correspondem aos pixels da imagem: $f(n, m)$, assim para o cálculo da Transformada de Fourier de Imagens, $F(u, v)$, usa-se sempre a forma digital a DFT definida pela equação (9) ou (11). O primeiro índice do domínio de frequência é 0 e N^*M é o número de pixels da imagem.

Espectro de amplitude ou espectro de Fourier de uma função de duas variáveis discretas é o gráfico de $|F(u, v)|$ versus as duas variáveis. Como $|F(u, v)|$ é periódica, não há necessidade de se encontrar mais frequências, além da primeira. Mesmo porque, por exemplo, em uma imagem de baixa resolução, como 128×128 pixels, tanto a imagem original quanto o espectro de Fourier terão $2^7 \cdot 2^7 = 2^{14}$ pontos ($2^7=128$) para serem usados. Isto é, a função original é decomposta em 2^{14} elementos, que são o somatório de 2^{14} senos e 2^{14} cosenos.

Na transformada inversa de imagens digitais são usado as mesmas 2^{14} funções avaliadas nos mesmos pontos. Esta quantidade de funções é mais do que suficiente para se obter com exatidão os valores da função original. As posições do domínio de frequência estão relacionadas às posições (n, m) do domínio espacial.

Na definição do espectro de uma imagem, geralmente para melhor visualizar devido ao grande range, é muito usada a forma logarítmica:

$$\log(1 + |F(u, v)|).$$

A figura 23 a seguir mostra uma imagem e seu espectro de Fourier em duas formas.

Transformada Rápida de Fourier (FFT)

Para aproximar uma função através das amostras, e aproximar a integral de Fourier pela transformada discreta de Fourier, necessitamos aplicar uma matriz cuja ordem é o número n de amostras. Como multiplicar uma matriz $n \times n$ por um vetor de ordem n emprega n^2 operações de aritmética, o problema aumenta com o aumento no número de pontos da amostra. Porém, se as amostras são uniformemente espaçadas, então a matriz de Fourier pode ser fatorada em um produto de poucas matrizes esparsas, e os fatores resultantes podem ser aplicados a um vetor com um total de $n \log n$ operações

aritméticas, diminuindo bem a complexidade da operação. Esta é idéia fundamental da transformada rápida de Fourier (FFT - *Fast Fourier Transform*) .

Antes de aprofundar neste sentido precisamos destacar algumas **propriedades da Transformada de Fourier**. Na tabela 1, observamos a simetria e a correspondência entre os domínios do espaço e da frequência e algumas propriedades para o par transformado de Fourier em uma variável (1D) . Nesta tabela F_1 e F_2 se referem as Transformadas de Fourier das funções f_1 e f_2 ., respectivamente. Os símbolos: a , a_1 e a_2 representam constantes reais arbitrárias.

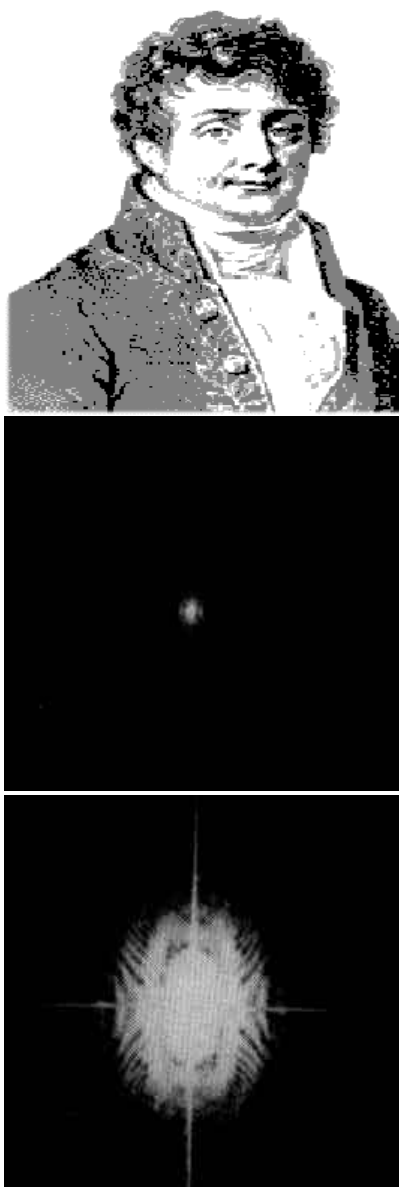


Figura 23: Imagem "Mosieu Fourier", seu espectro de Fourier e seu espectro na forma $\log (1+|F(u, v)|)$.

A **linearidade** indica que a transformada de Fourier da função oriunda da soma de duas funções tem sua transformada igual a soma das transformadas dessas funções. Além disso, a multiplicação de uma função por uma constante, faz sua transformada ser multiplicada pela mesma constante.:

$$\mathfrak{F}\{a_1 f_1(x) + a_2 f_2(x)\} = a_1 F_1(u) + a_2 F_2(u)$$

Como o parâmetro multiplicativo pode ser qualquer inclusive menor que 1 ou negativo, tem-se como decorrência que :

$$\mathfrak{F}\{f(-x)\} = F(-u)$$

A mudança de escala ou propriedade de **escalamento** (terceira linha da tabela 1) relaciona a multiplicação do parâmetro a no domínio do espaço pela divisão do mesmo parâmetro no domínio da frequência. Esta propriedade mostra que a expansão no domínio do espaço é equivalente a compressão no domínio da frequência e vice-versa.

$$\mathfrak{F}\{f(ax)\} = |1/a| F(u/a)$$

A **mudança de origem** (0,0) no domínio do espaço ou frequência é tratada nas linhas da tabela que consideram o deslocamento no espaço e na frequência. Considerar a imagem da transformada de Fourier inicialmente posicionada na origem transladada para $(N/2, N/2)$ auxilia muitas vezes sua percepção.

Girar uma imagem de um determinado ângulo, faz com que sua transformada seja também rotacionada deste angulo, isso pode ser melhor entendido pela figura 24.

Tabela 1 : Propriedades da Transformada de Fourier

Operação	$f(x)$	$F(u)$
Linearidade	$a_1 f_1(x) + a_2 f_2(x)$	$a_1 F_1(u) + a_2 F_2(u)$
Escalamento	$f(ax)$	$\frac{1}{ a } F\left(\frac{u}{a}\right)$
Deslocamento no espaço	$f(x - x_0)$	$F(u) e^{-i u x_0}$
Deslocamento no frequência	$f(x) e^{i u_0 x}$	$F(u - u_0)$
Convolução no espaço	$f_1(x) * f_2(x)$	$F_1(u) F_2(u)$
Convolução na frequência	$f_1(x) f_2(x)$	$\frac{1}{2\pi} [F_1(u) * F_2(u)]$

Outro resultado importante pela sua aplicação em diversos filtros (como o de Gabor, por exemplo) em imagens é que a Transformada de Fourier de uma função Gaussiana é também uma função Gaussiana, mais precisamente:

$$\mathfrak{F}(\exp[-t^2/2]) = (2\mathbf{p})^{-1/2} \exp[-u^2/2]$$

Assim, exceto pelo fator multiplicativo $\mathbf{\tilde{0}2p}$ a transformada de Fourier da função $\exp[-t^2/2]$ é a mesma.

Na Tabela 1 não se encontra a propriedade de simetria e separabilidade.

A propriedade de **simetria** diz que:

$$\mathfrak{F}\{F(x)\} = 2\mathbf{p}f(-u)$$

A **separabilidade** só ocorre em transformadas de Fourier de mais de uma variável. Essa propriedade permite fazer a integração das variáveis de forma independente em uma e depois em outra direção. Sendo um dos pontos que fundamentam a FFT.

Para exemplificar, vamos considerar uma função discreta em 2D, que poderia ser vista como uma imagem mínima, descrita pelos valores abaixo:

$$f(n,m) = \{f(0,0)=0, f(0,1)=0, f(0,2)=0, f(0,3)=0, \\ f(1,0)=2, f(1,1)=3, f(1,2)=4, f(1,3)=4, \\ f(2,0)=0, f(2,1)=0, f(2,2)=0, f(2,3)=0\}$$

Considerando a separabilidade na função $f(n,m)$ dada, podemos considerar que temos 3 funções discretas em n :

$$f1(n) = \{f1(0)=0, f1(1)=0, f1(2)=0, f1(3)=0\} \\ f2(n) = \{f2(0)=2, f2(1)=3, f2(2)=4, f2(3)=4\} \\ f3(n) = \{f3(0)=0, f3(1)=0, f3(2)=0, f3(3)=0\}$$

e 4 funções discretas em m :

$$f4(m) = \{f4(0)=0, f4(1)=2, f4(2)=0,\} \\ f5(m) = \{f5(0)=0, f5(1)=3, f5(2)=0,\} \\ f6(m) = \{f6(0)=0, f6(1)=4, f6(2)=0,\} \\ f7(m) = \{f7(0)=0, f7(1)=4, f7(2)=0,\}$$

Neste caso de duas variáveis discretas teremos: $u = 0,1,2,...M-1$, $v = 0,1,2,...N-1$, $m = 0,1,2,...M-1$ e $n = 0,1,2,...N-1$. Fica mais fácil calcular estas **transformadas separadamente** e depois combiná-las por multiplicação. Cada uma delas sendo dada pela equação (7).

$$F(u) = \frac{1}{N} \sum_{n=0}^{N-1} f(n)e^{-2\mathbf{p}j(un/N)}$$

Para $f1, f2$ e $f3$, $N=4$ e n no somatório varia de 0 a 3. Para $f1$ e $f3$ $f(n) = 0$ para todo n , de modo que $F1=F3=0$ para todo u

Para $f2$ temos:

$$F(0) = \frac{1}{4} \sum_{n=0}^3 f(n)e^0 \\ = \frac{1}{4}(f(0) + f(1) + f(2) + f(3)) \\ = \frac{1}{4}(2 + 3 + 4 + 4) = 3,25$$

$$F(1) = \frac{1}{4} \sum_{n=0}^3 f(n)e^{-2\mathbf{p}j(n/4)} \\ = \frac{1}{4}(2 - 3j - 4 + 4j) = \frac{1}{4}(-2 + j)$$

$$F(2) = \frac{1}{4} \sum_{n=0}^3 f(n)e^{-\mathbf{p}jn} \\ = \frac{1}{4}(2e^0 + 3e^{-j\mathbf{p}}j + 4e^{j2\mathbf{p}} + 4e^{j3\mathbf{p}}) = \frac{1}{4}(-1)$$

$$F(3) = -\frac{1}{4}(2 + j)$$

Para $f4, f5, f6$ e $f7$, $N=3$ e n no somatório varia de 0 a 2.

Como $fk(0)$ e $fk(2) = 0$ para todas ($k=4,5,6,7$), só precisamos calcular a parte exponencial para $n=1$ para todas as $f4, f5, f6$ e $f7$. Essa será:

$$F(u) = 1/3 f(1) \exp[-j2\mathbf{p}u/3]$$

Para $f4$ teremos:

$$F4(u) = 2/3 \exp[-j2\mathbf{p}u/3]$$

de modo que:

$$F4(0) = 2/3$$

$$F4(1) = 2/3 \exp[-j2\mathbf{p}/3]$$

(como $2\mathbf{p}/3 = 120$ graus, usando a identidade de Euler:)

$$F4(1) = 2/3 \exp[-j2\mathbf{p}/3] = \\ 2/3(\cos[2\mathbf{p}/3] - j \sin[2\mathbf{p}/3]) = \\ 2/3(\cos[120^\circ] - j \sin[120^\circ]) = \\ 2/3(-1/2 - j\mathbf{\tilde{0}3}/2) = -2(1 + \mathbf{\tilde{0}3}j)/6 = \\ F4(1) = -(1 + \mathbf{\tilde{0}3}j)/3$$

$$F4(2) = 2/3 \exp[-j4\mathbf{p}/3]$$

($4\mathbf{p}/3 = 240$ graus, assim pela identidade de Euler:)

$$F4(2) = 2/3(\cos[4\mathbf{p}/3] - j \sin[4\mathbf{p}/3]) = \\ 2/3(\cos[240^\circ] - j \sin[240^\circ]) = \\ 2/3(-1/2 - j\mathbf{\tilde{0}3}/2) = -(1 - \mathbf{\tilde{0}3}j)/3$$

Para $f5$ teremos:

$$F5(u) = 3/3 \exp[-j2\mathbf{p}u/3]$$

de modo que:

$$F5(0) = 1$$

$F5(1) = 3/3 \exp[-j2p/3]$
 $(2p/3 = 120 \text{ graus}, \text{ assim pela identidade de Euler:})$

$$\begin{aligned} F5(1) &= \exp[-j2p/3] = \\ &= (\cos[2p/3] - j \sin[2p/3]) = \\ &= (\cos[120^\circ] - j \sin[120^\circ]) = \\ &= (-1/2 - j\sqrt{3}/2) = -(1 + j\sqrt{3})/2 \end{aligned}$$

$F5(2) = 3/3 \exp[-j4p/3]$
 $(4p/3 = 240 \text{ graus}, \text{ assim pela identidade de Euler:})$

$$\begin{aligned} F5(2) &= (\cos[4p/3] - j \sin[4p/3]) = \\ &= (\cos[240^\circ] - j \sin[240^\circ]) = \\ &= (-1/2 - j\sqrt{3}/2) = -(1 + j\sqrt{3})/2 \end{aligned}$$

Para $f6$ teremos:

$$F6(u) = 4/3 \exp[-j2pu/3]$$

de modo que :

$$F6(0) = 4/3$$

$F6(1) = 4/3 \exp[-j2p/3]$
 $(2p/3 = 120 \text{ graus}, \text{ assim pela identidade de Euler:})$

$$\begin{aligned} F6(1) &= 4/3 \exp[-j2p/3] = \\ &= 4/3 (\cos[2p/3] - j \sin[2p/3]) = \\ &= 4/3 (\cos[120^\circ] - j \sin[120^\circ]) = \\ &= 4/3 (-1/2 - j\sqrt{3}/2) = -2(1 + j\sqrt{3})/3 \end{aligned}$$

$F6(2) = 4/3 \exp[-j4p/3]$
 $(4p/3 = 240 \text{ graus}, \text{ assim pela identidade de Euler:})$

$$\begin{aligned} F6(2) &= 4/3 (\cos[4p/3] - j \sin[4p/3]) = \\ &= 4/3 (\cos[240^\circ] - j \sin[240^\circ]) = \\ &= 4/3 (-1/2 - j\sqrt{3}/2) = \\ &= -(1 + j\sqrt{3})/6 \end{aligned}$$

Para $f7$ teremos:

$$F7(u) = 4/3 \exp[-j2pu/3]$$

e valores idênticos ao de $F6(u)$

A transformada de Fourier desta mini imagem será definida por esses 12 valores resultantes da multiplicação dos resultados das linhas e colunas e $F(u,v)$ fica:

$$\begin{aligned} F(u,v) &= \{F(0,0)=0, \\ &F(0,1)=0, \\ &F(0,2)=0, \\ &F(0,3)=0, \\ &F(1,0)=(13/4)(-1 + j\sqrt{3})/3, \\ &F(1,1)=(-2+j)/4[-(1 + j\sqrt{3})/2], \\ &F(1,2)=[-1/4][-2(1 + j\sqrt{3})/3], \\ &F(1,3)=[(-2+j)/4][-2(1 + j\sqrt{3})/3], \\ &F(2,0)=0, \\ &F(2,1)=0, \\ &F(2,2)=0, \\ &F(2,3)=0\} \end{aligned}$$

A **parte Real** de $F(u,v)$ da imagem são os valores que não estão sendo multiplicados por j . A

parte Imaginária de $F(u,v)$ são os valores que estão sendo multiplicados por j . As partes reais e imaginárias dos primeiros quatro valores e dos últimos quatro é nula. Para os valores da segunda linha ambas, parte real e imaginária, existem, sendo que para:

$$F(1,0)=(13/4)(-1 + j\sqrt{3})/3,$$

a parte real é: $(-13/12)$ e a parte imaginária $-13\sqrt{3}/12$. Para:

$$\begin{aligned} F(1,1) &= [(-2+j)/4][-(1 + j\sqrt{3})/2] = \\ &= (-2+j)(1 + j\sqrt{3})/8 = \\ &= (-1/8)[(-2+j) + (-2\sqrt{3}j - \sqrt{3})] \end{aligned}$$

(lembre-se que $j^2 = -1$), temos que a parte real de $F(1,1)$ é $(2 + \sqrt{3})/8$ e a parte imaginária $(2\sqrt{3} - 1)/8$.

Finalmente para:

$$F(1,2)=[-1/4][-2(1 + j\sqrt{3})/3],$$

a parte real é: $(1/6)$ e a parte imaginária $\sqrt{3}/6$.

O **espectro de Fourier** será obtido achando-se o módulo de cada um dos valores $F(u,v)$ acima. Isto é, elevando a parte real ao quadrado, somando com a parte imaginária elevada ao quadrado de cada (u,v) e depois extraindo a raiz quadrada deste número.

Os incrementos em u e v se relacionam com os incrementos em n e m por: $NDn = 1/Dv$, $MDm = 1/Du$. Se no domínio espacial os incrementos em n e m forem de $0,25 = 1/4$, em u serão $4/M$ e em v serão $4/N$. No caso em análise $M=4$ e $N=3$, então: $\Delta u = 1$ e $\Delta v = 4/3$.

Considerando este exemplo, um **algoritmo** que calculasse o espectro da transformada de Fourier de uma imagem poderia ser proposto como:

- Decomponha a imagem por linhas com N valores :
- Decomponha a imagem por colunas com M valores :
- Defina os valores u de cada linha e para cada linha calcule

$$F(u) = \frac{1}{N} \sum_{n=0}^{N-1} f(n) e^{-2\pi j(un/N)}$$

- Armazene estes valores.
- Defina os valores v de cada coluna e para cada l coluna calcule:

$$F(v) = \frac{1}{M} \sum_{n=0}^{M-1} f(n) e^{-2\pi j(vn/m)}$$

- Armazene estes valores.
- Multiplique os valores correspondentes e obtenha F

Este algoritmo **não é nada otimizado**. E muito pode ser feito para melhorar sua performance. Por exemplo, a estratégia conhecida como *Método de Decomposição Linha Coluna*, consiste em varrer a matriz de *pixels* sucessivamente em linhas e colunas [17] extraindo dela em cada etapa um vetor 1D para o

qual é calculado a *Transformada Rápida de Fourier* para 1D (TRF-1D). Portanto, o núcleo do cálculo da transformada para uso em imagens (2D) é o cálculo da TRF-1D. Reduzir a complexidade computacional do cálculo da Transformada Discreta de Fourier para 1D acaba por reduzir o cálculo das Transformadas de Fourier em 2D.

Método de Decomposição Linha Coluna

A complexidade computacional envolvida no cálculo da Transformada Discreta de Fourier é do tipo $O(n^2)$, pois podemos considerar que para cada um dos N valores de u são realizadas N multiplicações de $f(x)$ pela exponencial e $(N-1)$ adições dos resultados multiplicados. Os termos internos da exponencial são calculados apenas uma vez, e podem ser armazenados após o primeiro cálculo para uso posterior, por isto, não são computados no cálculo da complexidade computacional. O objetivo da Transformada Rápida de Fourier é reduzir o esforço computacional para $O(N \log_2 N)$.

Um método que auxilia o entendimento dos passos que levam a Transformada Rápida de Fourier para 2D é o *Método de Decomposição Linha-Coluna*. Para entender suas bases suponha que queremos achar a transformada de Fourier de $f(n_1, n_2)$, função que representa uma matriz de *pixels* de uma imagem. Primeiramente é feita a varredura por linha da matriz para obter a transformada intermediária $f(k_1, n_2)$, a seguir "varre-mos" a matriz intermediária por colunas para produzir a TRF-2D final $F(k_1, k_2)$. Cada varredura de linha (ou de coluna) da matriz original (ou intermediária) implica extrair uma linha (coluna) para calcular a respectiva TRF-1D analisada na seção anterior deste texto. A demonstração deste algoritmo e a respectiva análise de complexidade podem ser encontradas em [17].

Método da Duplicação Sucessiva

O *Método da Duplicação Sucessiva*, representa uma das possibilidades adicionais de reduzir o esforço computacional para o cálculo da Transformada Discreta de Fourier para uma dimensão (1D). Chamando $N = 2M$, com M representando a metade dos elementos do qual se quer achar a transformada 1D, a equação (7) pode ser escrita como.

$$F(u) = \frac{1}{2M} \sum_{n=0}^{2M-1} f(n) e^{-2\pi j (un/2M)}$$

$$= \frac{1}{2M} \left[\sum_{n=0}^{2M-1} f(n) e^{-2\pi j (un/2M)} + \sum_{n=0}^{2M-1} f(2n+1) e^{-2\pi j (un/2M)} \right]$$

*onde o primeiro termo entre colchetes se refere aos termos pares da transformada, enquanto o último

termo se refere aos termos ímpares da transformada. A exigência deste método é que N seja uma potência de 2 ($N = 2^n$). É claro que $n = (\log N)/2$, e, se chamamos m o número de multiplicações e a o número de adições:

$$m(n) = 2m(n-1) + 2n-1$$

$$a(n) = 2a(n-1) + 2n-1$$

Estes resultados nos permitem definir o número de operações de modo recursivo, e mostra que a complexidade computacional deste método é $O(n \log_2 n)$, portanto, menor que a complexidade da Transformada Discreta de Fourier [36].

Convolução e Transformada de Fourier

As últimas propriedades da Tabela 1, se referem as relações entre Convolução e Transformada de Fourier. Convolução entre duas funções $f(x)$ e $g(x)$ é a operação entre elas definida como:

$$f(x) * g(x) = \int_{-\infty}^{\infty} f(t) g(t-x) dx$$

Essa operação pode ser descrita no domínio do espaço ou da frequência. Isto é, existe uma definição idêntica se for usado $F(u)$ e $G(u)$.

Um caso importante, especialmente em imagens é o das funções $f(x)$ e $g(x)$ não existirem para valores positivos, isto é: $f(x)=0$ e $g(x)=0$ se $x < 0$, neste caso a integral terá limites de 0 e t e não mais de menos a mais infinito:

$$f(x) * g(x) = \int_0^t f(t) g(t-x) dx \quad (13)$$

A convolução é distributiva em relação a soma de duas funções, comutativa e associativa. A convolução de uma função qualquer com a função impulso unitário resulta na própria função.

O teorema da Convolução diz que a Transformada de Fourier da convolução de duas funções no espaço é igual a multiplicação de suas transformadas na frequência:

$$\mathfrak{F}(h(x) * f(x)) = \mathfrak{F}(h(x)) \mathfrak{F}(f(x)) = H(u) F(u)$$

O teorema da Convolução afirma também que a Transformada de Fourier Inversa da convolução de duas funções na frequência é igual a multiplicação de suas transformadas inversas no espaço:

$$\mathfrak{F}^{-1}(H(u) * F(u)) = 2\pi h(x) f(x)$$

Ou, em outras palavras, a transformada de Fourier da multiplicação de duas funções é a multiplicação de suas transformadas:

$$\mathfrak{F}(h(x) f(x)) = 1/2\pi H(u) * F(u)$$

A transformada de Fourier facilita a convolução de funções, transformando a integração

acima em um domínio, em multiplicação no outro domínio. No caso de funções de duas variáveis, como em imagens, temos:

$$g(x,y)=h(x,y)*f(x,y) \quad \hat{U} \quad G(u,v)=H(u,v)F(u,v)$$

ou

$$G(u,v)=H(u,v)*F(u,v) \quad \hat{U} \quad g(x,y)=h(x,y)f(x,y)$$

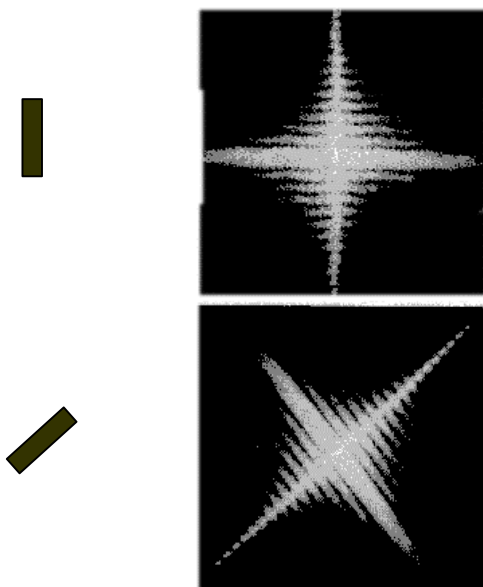


Figura 24: Imagem "Retângulo" original (acima), rotacionada de 45 graus (abaixo), e seus respectivos espectros de Fourier.

Uma característica muito útil é que a maior parte do espectro de Fourier, $|F(u,v)|$, de imagens se concentra nas componentes de baixa frequência. A figura 25 mostra a energia ou $|F(u,v)|^2$, de uma imagem típica. Nesta imagem, são falsamente adicionados raios de 8, 18, 43, 78 e 152 pixels que contém 90 %, 93 %, 95 %, 99% e 99,5 % da potência da imagem, respectivamente.

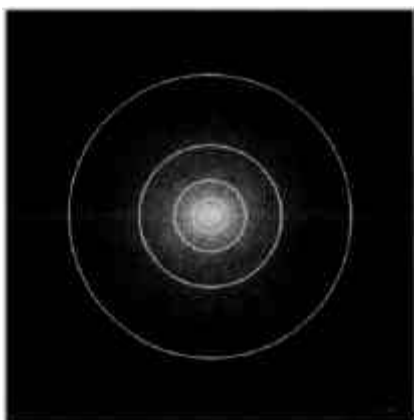


Figura 25: No espectro de Fourier de uma imagem típica com falsos círculos a 90 %, 93 %, 95 %, 99% e 99,5 % de sua potência.

Por definição, uma **função impulso** é: "uma função que tem um determinado valor em apenas um ponto": $h(x) = a$ se $x=x_0$ e $h(x) = 0$ para os outros valores de x . A função impulso unitário é muitas vezes identificada pela letra grega $\delta(x)$ e é aquela que tem valor 1 em apenas um ponto. A figura que segue mostra a função impulso na origem.

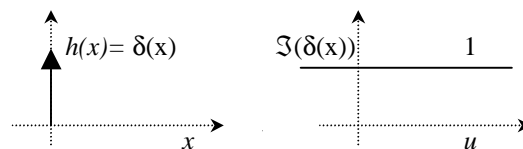


Figura 26: Função Impulso unitário e sua Transformada de Fourier

A Transformada de Fourier da Função Impulso unitário é a função constante unitária, $F(u)=1$ sobre todo domínio da frequência. Por sua vez a Transformada de Fourier da Função constante $f(x)=A$, sobre todo domínio do espaço x é a função Impulso $\mathcal{Z}(A)=2\pi A \delta(u)$. Esses pares transformados ilustram a afirmação de que um ponto no domínio da frequência contém informação sobre toda a imagem. O valor de um ponto na frequência indica uma distribuição com certa frequência na imagem espacial. Assim, se na imagem houver um certo ruído ou textura constante, isso irá se refletir como um componente de alta frequência na transformada de Fourier desta imagem.

Convolução em Imagens

A figura 27 mostra uma imagem degradada por um ruído comum em transmissões pela adição de uma frequência espúria ao sinal. O Espectro de Fourier desta imagem, mostrado na figura 28, apresenta intensidades altas de frequência bem definidas fora da região central (normalmente a maior parte da energia da imagem está concentrada próximo a origem). Retirar essas frequências na imagem original com ruídos equivale a eliminar impulsos, no domínio da frequência, na sua forma transformada e a torná-la, no domínio do espaço, na forma mostrada na figura 29, isto é sem os ruídos em formato de listas a 45° vistos na figura original.

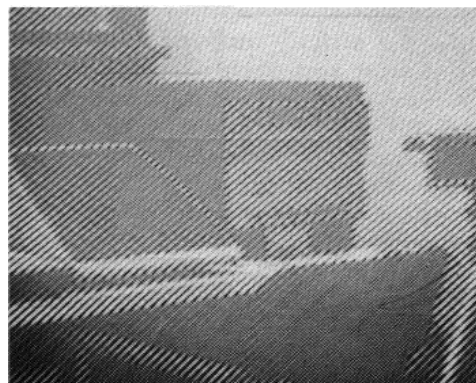


Figura 27: Imagem com ruído



Figura 28 :Espectro de Fourier da imagem da figura anterior

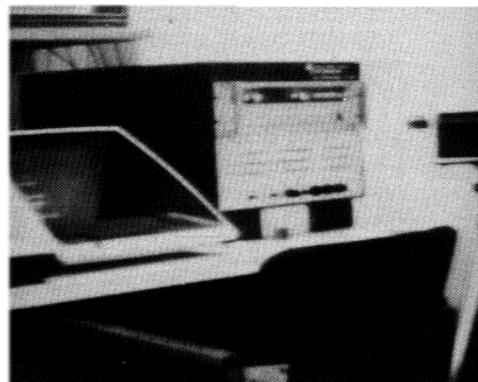


Figura 29: Imagem depois de retiradas as frequências secundárias.

Muitas vezes é muito complexo descobrir os valores de frequência a serem eliminados. Como exemplo tem-se a imagem da figura 30 que se apresenta degradada por uma combinação de ruídos que formam uma textura adicional, padrões ou *moiré*. Seu espectro de Fourier, figura 31, indica a existência de diversas intensidades fora do centro onde seria concentrada a informação fundamental da imagem. Considerando no espectro apenas as frequências de alta intensidade que têm raio maior que 1. Ao passar para o domínio do espaço tem-se como resultado uma imagem que é apenas de uma textura resultante destas frequências como mostrado na figura 32. Ao fazer uma filtragem da imagem original usando um filtro rejeita banda de frequências maiores ou iguais que as correspondentes ao menor destes raios, tem-se a imagem mostrada na figura 33. Essa imagem, embora ainda com diversas características a serem tratadas, se encontra livre do ruído que causava a textura de fundo da imagem original.

Os processos exemplificados consistiram então de transformar uma imagem para o domínio de Fourier, verificar seu espectro e usá-lo para descobrir frequências correspondentes a ruídos que se desejasse eliminar. A imagem é processada neste domínio eliminando determinadas frequências que correspondem a esses ruídos. Depois retorna-se a imagem processada para o domínio espacial onde ela é entendida pelo nossos olhos.

Filtros Passa e Rejeita Banda

O processo discutido anteriormente pode ser generalizado pelos filtros que consideram a eliminação de faixas de frequência. Denomina-se filtro **rejeita banda** ao que não deixar passar sinais que forem de determinado intervalo de frequência, ou seja transformará para zero os valores que estiverem dentro de um limite definido. Denomina-se filtro **passa banda** ao que só deixar passar sinais que forem de determinado intervalo de frequências, ou seja, transformará para zero os valores que estiverem fora deste limite.

Deconvolução é um processo que é usado para restaurar imagens. Consiste na divisão, no domínio de Fourier, ou na Multiplicação pela inversa. É também chamada de filtragem inversa.

Se uma imagem $I(x,y)$ for degradada por um sinal ou por uma função $h(x,y)$, ficando $g(x,y)$:

$$g(x,y) = I(x,y) * h(x,y)$$

No domínio de Fourier essa convolução ficará:

$$G(u,v) = F(u,v) H(u,v),$$

se $H(u,v)$, for a transformada de $h(x,y)$. A filtragem inversa será: $F(u,v) = G(u,v) / H(u,v)$. **Filtros de Wiener** são formas de obter a restauração de imagens usando melhorias desta técnica.

De maneira semelhante, podem ser projetados filtros no domínio da frequência, que depois são transformados para poderem ser aplicados no domínio do espaço. A arte da operação de filtragem consiste então em definir adequadamente a função usada na filtragem em frequência: $H(u,v)$.

Os filtros **passa alta** e **passa baixa** frequências ideais são definido por:

$$H(u,v) = \begin{cases} 0 & \text{se } D(u,v) \leq D_0 \\ 1 & \text{se } D(u,v) > D_0 \end{cases}$$

$$H(u,v) = \begin{cases} 1 & \text{se } D(u,v) \leq D_0 \\ 0 & \text{se } D(u,v) > D_0 \end{cases}$$

onde D_0 é um valor específico não negativo e $D(u,v)$ é a distância do ponto (u,v) a origem do plano de frequência, isto é:

$$D = (u^2 + v^2)^{1/2}$$

O filtro **rejeita banda** suprime todas as frequência em uma vizinhança de raio D_0 em torno de um ponto (u_0, v_0) :

$$H(u, v) = \begin{cases} 0 & \text{se } D(u, v) \leq D_0 \\ 1 & \text{se } D(u, v) > D_0 \end{cases}$$

onde

$$D(u, v) = [(u - u_0)^2 + (v - v_0)^2]^{1/2}.$$



Figura 30: Imagem original degradada



Figura 31: Espectro de Fourier da imagem da figura 30

Devido a simetria da transformada de Fourier se não for em torno da origem deve ser feita em pares simétricos, para manter o sentido:

$$H(u, v) = \begin{cases} 0 & \text{se } D_1(u, v) \leq D_0 \text{ ou } D_2(u, v) \leq D_0 \\ 1 & \text{para outros valores} \end{cases}$$

Estes filtros tem uma variação abrupta nas frequências eliminadas. Muitas vezes podem ser adequados cortes mais suaves. Isso é feito pelos **Filtros de Butterworth**, que são definidos pela seguintes equações:

$$H(u, v) = 1 / (1 + (D/D_0)^{2n})$$

$$H(u, v) = 1 / (1 + (D_0/D)^{2n})$$

Filtragem por convolução no espaço da imagem

Como visto na seção anterior a convolução na forma digital bidimensional inteira é muito utilizada em processamento de imagens. Dada uma imagem $f(m, n)$, a operação de um filtro gera uma nova imagem $g(m, n)$, determinada por

$$g(m, n) = \sum_{n_1=-\infty}^{+\infty} \sum_{m_1=-\infty}^{+\infty} h(m_1, n_1) f(m - m_1, n - n_1)$$

onde (m, n) são usados ao invés de (x, y) nas funções para lembrar que estão na forma digital [24]. A variação de n_1 e m_1 geralmente é não infinita e restrita a poucos valores, cuja faixa de restrição é denominada máscara. Na prática, o conceito de filtragem por convolução é aplicado no domínio de frequência, usando-se as propriedades da Transformada de Fourier, mencionadas anteriormente.

A operação de aplicar uma máscara *pixel* a *pixel* sobre uma imagem e calcular a quantidade predefinida em cada *pixel*, é o fundamento da convolução. Formalmente, a convolução discreta unitária de duas funções $f(n, m)$ e $h(n, m)$ de tamanho $M \times N$ é denotado por $f(n, m) * h(n, m)$ e definida como:

$$f(n, m) * h(n, m) = \frac{1}{MN} \sum_{n'=0}^{M-1} \sum_{m'=0}^{N-1} f(n', m') h(n - n', m - m')$$

O sinal negativo significa que a função é espelhada sobre a origem. Isto é inerente na definição da convolução. A equação acima é a implementação de (1) uma reflexão vertical de uma função sobre a origem; (2) deslocar aquela função com respeito a outra pelo incremento dos valores de (n, m) ; e (3) calcular a soma dos produtos sobre todos os valores de m e n , para cada deslocamento. Os deslocamentos (n, m) são incrementos inteiros que param quando as funções não mais se sobrepõem [17].

Supondo que $F(u,v)$ e $H(u,v)$ denotem as transformadas de Fourier de $f(x,y)$ e $h(x,y)$, respectivamente, o teorema da convolução diz que $f(x,y) * h(x,y)$ e $F(u,v)H(u,v)$ constituem um par de transformadas de Fourier. Isto é formalmente escrito como:

$$f(x,y) * h(x,y) \Leftrightarrow F(u,v)H(u,v)$$

Se a imagem já estiver descrita no domínio de frequência, a multiplicação da equivalência acima permite que a convolução seja realizada mais rapidamente, pois a integral no domínio do espaço

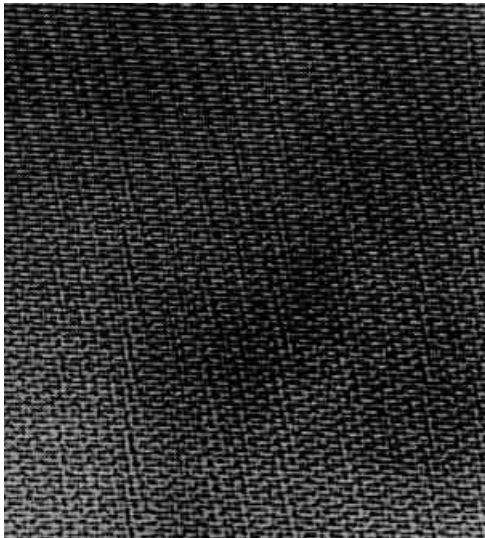


Figura 32: Textura devido aos pontos fora do centro

mais demorada. Deve-se notar, que a multiplicação no domínio das frequências não é realizado segundo o conceito de multiplicação de matrizes. Esta multiplicação é realizada elemento a elemento, tal como um peso aplicado aos valores da imagem. [24].

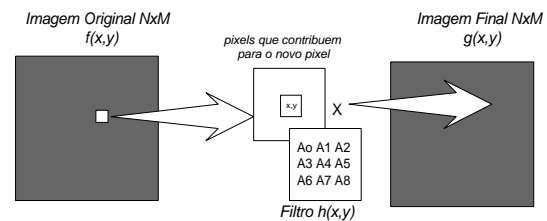


Figura 33: Imagem da figura 29 depois de processada

Combinando Filtros no espaço da imagem

De uma forma geral, o processo de filtragem, tanto linear quanto não linear, tem um papel importante na área de processamento de imagens, inclusive no pré-processamento. A filtragem permite suavizar a imagem ou aumentar o seu contraste. Contornos podem ser detectados por filtragem linear usando a distribuição espacial dos níveis de cinza da imagem.

A filtragem linear mais usada é a convolução da imagem digital com um *kernel* ou “máscara” (*mask*). O processo de convolução calcula um novo valor de intensidade para cada pixel da imagem, tendo como base os tons de cinza dos pixels da vizinhança. Assim, cada pixel contribui, com uma porcentagem do seu valor, para o cálculo do nível de cinza do novo pixel. Um esquema representativo deste processo pode ser visto na figura 34.



Se $h(x,y)$ for definida pelo *Filtro de Sobel horizontal* os $A_0...A_8$ serão definidos como:

-1	-2	-1
0	0	0
1	2	1

Figura 34: Esquema do Processo de Convolução Filtragem Espacial Usando um *Kernel* 3x3

O processo comumente empregado em processamento de imagens para detecção de contorno é o uso de filtragem por convolução com um *filtro passa alta*. Para esta tarefa, diversos filtros, deste tipo (Sobel, Roberts e Laplacianos de várias dimensões $m \times n$), se encontram disponíveis na literatura. Como exemplo, segue a máscara do Laplaciano do filtro Gaussiano (*Laplacian of Gaussian - LoG*) em 5×5 :

0	0	-1	0	0
0	-1	-2	-1	0
-1	-2	-16	-2	-1
0	-1	-2	-1	0
0	0	-1	0	0

Usando a representação de uma imagem digital como uma função $f(x,y)$, podemos descrever a convolução (usando a notação da figura 34) dessa imagem por uma máscara $h(i,j)$ pela expressão que segue:

$$g(x, y) = \sum_{i,j \in h} f(x-i, y-j) \cdot h(i, j) \quad (14)$$

onde os parâmetros x e y representam um passeio por todos os pixels da imagem: $0 \leq x-i < N$ e $0 \leq y-j < M$. O resultado é uma nova imagem: $g(x, y)$.

A expressão 14 descreve as operações necessárias para a obtenção do novo valor do pixel (x, y) , não a filtragem inteira. Os somatórios dos índices i e j se realizam ao longo dos elementos da máscara, nesta expressão suposta quadrada (sendo esse o significado de " $i, k \in h$ ").

Os valores dos elementos $h(i, j)$, são fundamentais na determinação do que ocorrerá na imagem. A imagem dependendo destes valores poderá ter maior ou menor contraste, ter seus ruídos amplificados ou suavizados, formas básicas (pontos, linhas e contornos) que poderão ser detectadas ou diluídos.

Para o cálculo de cada pixel da imagem final, $g(x, y)$, (na equação 14) soma-se o valor na imagem original, $f(x, y)$, com os valores vizinhos multiplicados pelo peso definido no *kernel* da convolução $h(x, y)$. Para *kernels* 3x3 a convolução definida na equação 14 se reduz a:

$$\begin{aligned} g(x, y) = & A_0 f(x-1, y-1) + \\ & A_1 f(x, y-1) + A_2 f(x+1, y-1) + \\ & A_3 f(x-1, y) + A_4 f(x, y) + \\ & A_5 f(x+1, y) + A_6 f(x-1, y+1) + \\ & A_7 f(x, y+1) + A_8 f(x+1, y+1) \end{aligned}$$

Por exemplo, o filtro **Laplaciano do Gaussiano** (já comentado), é útil no realce de bordas. É um filtro obtido pela combinação de dois filtros lineares. Esse filtro calcula a derivada segunda da imagem depois dela ser processada pelo filtro de Gaussiano em uma só operação, fazendo a derivada segunda da máscara Gaussiana a ser usada, e aplicando essa máscara já derivada na imagem. Dependendo da forma como esta combinação é feita diversos filtros deste tipo podem ser gerados.

Geralmente nos filtros Laplaciano do Gaussiano, a soma dos pesos da máscara é igual a zero, e a máscara sempre terá um alto valor central, cercado de valores negativos nas direções N-S e E-W do centro da máscara. Como esses filtros não têm direção preferencial, eles não consideram a direção das bordas detectadas, sendo mais adequado quando as bordas não são direcionais.

Vejamos como este processo de combinação pode ser obtido. O filtro Gaussiano em 2D é uma aproximação digital da figura 8. O filtro Gaussiano pode ser usado como um filtro de suavização ou

passa-baixa (ou "Low Pass"), ele usa a função Gaussiana para obter valores de uma máscara a ser definida.

O filtro de Gaussiano em 1-D tem diversas formas dependendo de quantos elementos terá a máscara, ou seja de como é aproximada a forma contínua da curva gaussiana ao se fazer a conversão para o mundo digital. Isto é, se a máscara final aproximará a função com apenas 3 pixels ou 30 temos diversos resultados. A forma digital sempre será uma aproximação da equação da curva de normal:

$$G(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}$$

onde σ é o desvio padrão considerado, na representação da expressão da normal considerada. Assume-se que a distribuição é centrada em $x=0$, de modo que a máscara será simétrica e o ponto central da máscara terá valor máximo. Como a imagem é armazenada como uma coleção bidimensional de pixels discretos, para obter-se uma máscara de filtros para imagens, deve-se usar a expressão do Gaussiano para duas variáveis, ou da superfície gaussiana mostrada na figura 35.

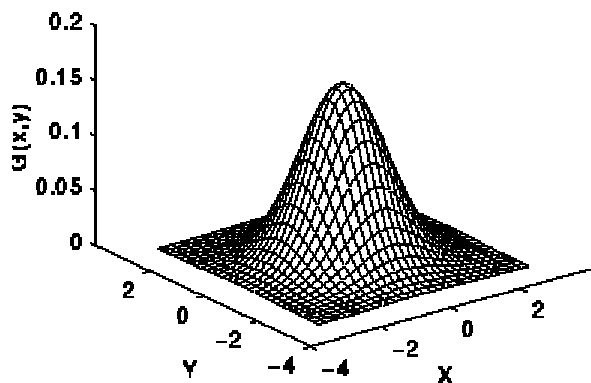


Figura 35: Representação da superfície gaussiana.

A distribuição Gaussiana perfeita iria requerer um núcleo de elementos muito grande para uma convolução perfeita, mas, na prática ela pode ser aproximada por máscaras em diferentes resoluções desde 3x3 pixels.

Uma forma adequada de projetar um filtro de suavização Gaussiano é avaliar o desvio padrão da imagem de entrada, ou de uma região desta, e usar este valor como o parâmetro da largura da Gaussiana bidimensional. Dependendo da forma que se quer suavizar, este parâmetro pode ser obtido e relacionada com a forma alvo, por exemplo, se linhas de determinado número de pixels precisarem ser suavizadas, pode-se usar o número médio de pixels das linhas das imagens como parâmetro.

O Laplaciano é o equivalente em 2D da segunda derivada parcial de uma função qualquer. Esse filtro pode ser usado como um filtro passa-alta

("High Pass"), ele usa as formas digitais das derivadas parciais de uma função para obter valores de uma máscara a ser definida, que se torna equivalente a derivar essa imagem.

O Filtro Laplaciano de Gaussiano é obtido então considerando uma aproximação da derivada parcial de uma aproximação digital da superfície gaussiana (figura 35).

A fórmula da função do Laplaciano (em homenagem a Pierre Simon, Marquês de Laplace, matemático e astrônomo francês, que viveu de 1749 a 1827), de uma função de duas variáveis, $f(x,y)$ é:

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

As derivadas de $f(x,y)$ são aproximadas usando equações de diferença:

$$\begin{aligned} \frac{\partial^2 f}{\partial x^2} &= \frac{\Delta_x}{\Delta x} \\ &= \frac{\partial (f[i, j+1] - f[i, j])}{\partial x} \\ &= \frac{\partial (f[i, j+1])}{\partial x} - \frac{\partial (f[i, j])}{\partial x} \\ &= (f[i, j+2] - f[i, j+1]) - (f[i, j+1] - f[i, j]) \\ &= f[i, j+2] - 2f[i, j+1] + f[i, j] \end{aligned}$$

Essa aproximação é feita sobre o pixel $[i, j+1]$. Então, substituindo j por $j-1$ obtêm-se:

$$\frac{\partial^2 f}{\partial x^2} = f[i, j+1] - 2f[i, j] + f[i, j-1],$$

que é a aproximação desejada da segunda derivada parcial centrada em $[i, j]$. Similarmente temos para a direção y :

$$\frac{\partial f}{\partial y^2} = f[i, j+1] - 2f[i, j] + f[i, j-1],$$

Fazendo a combinação das duas direções, temos as duas equações em um único filtro, a máscara por aproximação do Laplaciano de uma imagem ou função pode ser dada por:

0	1	0
1	-4	1
0	1	0

Às vezes é importante fornecer mais peso para os pixels do centro, no desenvolvimento anterior das derivadas, assim, uma outra aproximação muito usada para o Laplaciano pode ser:

0	4	0
4	-20	4
0	4	0

Ao ser usada essa máscara diretamente na imagem, tem-se a segunda derivada da imagem ou Laplaciano. Neste exemplo, o Laplaciano foi aproximado por uma máscara de 3x3, mas pode ser usado critério semelhante para obtenção de máscaras 2x2, 5x5, etc.

Uma forma de derivada primeira em linha isso é como 1x2 seria:

$$\begin{matrix} 1 & -1 \\ 1 & -1 \end{matrix}$$

Em uma matrix quadrada 2x2 seria :

$$\begin{matrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{matrix}$$

em 3x3 poderia-se ter:

$$\begin{matrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{matrix}$$

e assim por diante.

Para a derivada segunda seria mais inteligente aplicar a segunda diferença no filtro do que usá-lo duas vezes na imagem, o que resulta nos filtros *Laplacianos* já discutidos. Outras máscaras de derivadas segundas são as abaixo.

$$\begin{matrix} -1 & -1 & -1 & -1 & -2 & -1 \\ -1 & 8 & -1 & -2 & 12 & -2 \\ -1 & -1 & -1 & -1 & -2 & -1 \end{matrix}$$

Se, ao invés de ser aplicado na imagem, a derivada parcial for feita na aproximação digital da curva normal ou gaussiana, chega-se às diversas formas do filtro Laplaciano de Gaussiano. O Filtro Laplaciano do Gaussiano se usado com um conjunto de desvios padrões para detecção de contorno é também conhecido como Método de Marr-Hildreth. Esse filtro também ser obtido direto do cálculo analítica do Laplaciano da função Gaussiana 2D, isto é, pode ser obtido da discretização da função:

$$LoG(x,y) = -\frac{1}{\pi\sigma^4} \left[1 - \frac{x^2 + y^2}{2\sigma^2} \right] e^{-\frac{x^2 + y^2}{2\sigma^2}}$$

O **filtro de Gabor** é outro filtro resultante da combinação de processos (desenvolvido em 1946 por Dennis Gabor, físico Britânico inventor da holografia, nascido na Hungria em 1900 e falecido em 1979). Esse filtro é muito aplicado na melhoria e detecção de texturas, devido às suas características, especialmente as relacionadas as representações de frequência e orientação, que são similares ao do sistema visual humano [9,11,23]. Este filtro no domínio espacial é largamente utilizado na literatura [25], sendo sua forma geral :

$$G(x,y,f,\mathbf{q},\mathbf{s}_x,\mathbf{s}_y) = e^{2\pi j f x_q - \left[\frac{x_q^2}{2s_x^2} + \frac{y_q^2}{2s_y^2} \right]}$$

onde:

$x_q = x \cos \mathbf{q}_k + y \sin \mathbf{q}_k$, $y_q = -x \sin \mathbf{q}_k + y \cos \mathbf{q}_k$ e x, y as coordenadas retangulares no domínio espacial.

O filtro de Gabor tem como parâmetro a frequência, f , e o desvio padrão nas direções horizontal e vertical \mathbf{s}_x e \mathbf{s}_y . [11].

Essa f é tem as seguintes peculiaridades: se f é muito grande, ruídos são criados na imagem filtrada; se f é muito pequena, linhas parecerão entrelaçadas. Este parâmetro pode ser obtido de $f=1/k=0,5$, onde k é a média da largura das linhas ou outro elemento constante (em *pixels*) que se quiser processar.

O parâmetro \mathbf{q}_k refere-se a orientação do filtro. \mathbf{s} é o desvio padrão da distribuição normal (ou Gaussiana) bidimensional [9], sendo relacionado à largura significativa da Gaussiana que modula o filtro de Gabor.

O filtro de Gabor é projetado no domínio espacial da imagem, modulado por uma Gaussiana bidimensional, e no domínio de frequências tem o mesmo formato que no domínio espacial (já que a transformada da Gaussiana é também uma função Gaussiana). A largura da Gaussiana bidimensional é determinada pelo desvio padrão. Se \mathbf{s} é muito grande, o filtro é mais robusto a ruídos, mas não conseguirá capturar os detalhes das linhas. Se \mathbf{s} é muito pequeno, o filtro não consegue remover ruídos, mas consegue capturar os detalhes das linhas [18].

Este é um filtro passa-banda, a frequência f estabelecida no domínio espacial da imagem é o pico da Gaussiana bidimensional no domínio da frequências. Isto faz com que o filtro atenua ou rejeite as frequências com distância maior que o desvio padrão no domínio das frequências, com centro na frequência f . Logo, encontrar valores ideais para estes parâmetros para cada imagem é muito importante.

Utilizar a origem no centro do filtro, faz com que o processo de filtragem seja independente do domínio a ser usado. Por exemplo, usar os domínios de -15 a 15 e -7 a 7 para ambas as direções horizontais e verticais. Isto devido à simetria no processo de convolução.

Testes mostram grande relação do ângulo utilizado como parâmetro no filtro de Gabor com linhas presentes na imagem. Dependendo do ângulo x_θ na primeira parte da expressão é possível e retirar inconsistências horizontais em imagens, como lacunas ou falhas. Utilizar o filtro de Gabor com o valor adequado de y_θ permite retirar inconsistências com direções verticais das imagens.

O filtro Sobel é usado para realçar bordas nas direções horizontais ou verticais. O tamanho da máscara é sempre (3 x 3). Considerando uma vizinhança do pixel (i, j) mostrado na figura 36. O operador Sobel é dado pela combinação dos gradientes em x e y :

$$\sqrt{S_x^2 + S_y^2}$$

onde:

$$S_x = (a_2 + c a_3 + a_4) - (a_0 + c a_7 + a_6),$$

$$S_y = (a_0 + c a_1 + a_2) - (a_6 + c a_5 + a_4),$$

com a constante $c = 2$. As duas máscaras: S_x e S_y detectam as derivadas horizontais e verticais respectivamente. S_y é obtida da rotação de 90 graus de S_x , que é apresentada na figura 34.

a_0	a_1	a_2
a_7	$[i,j]$	a_3
a_6	a_5	a_4

Figura 36: Variáveis usadas no operador Sobel

O filtro de Prewitt usa as equações do filtro de Sobel, a única diferença é que ele usa para a constante c o valor 1. Assim duas máscaras apresentadas abaixo detectam as derivadas horizontais e verticais respectivamente pelo filtro de Prewitt.

$$\begin{array}{ccccc} 1 & 1 & 1 & & 1 & 0 & -1 \\ S_x = 0 & 0 & 0 & S_y = 1 & 0 & -1 \\ -1 & -1 & -1 & & 1 & 0 & -1 \end{array}$$

O método **de Roberts** é outra filtragem não-linear parecida com a de Sobel e representa uma aproximação da função de Roberts. O tamanho da máscara (2 x 2) não pode ser mudado. Esse filtro tem a característica de realçar bordas direcionalmente, mas pode produzir muitas vezes bordas artificiais, que podem confundir o usuário.

O operador Roberts provem de uma aproximação do gradiente por:

$$G[f[i, j]] = \frac{1}{2} R_x \frac{1}{2} + \frac{1}{2} R_y \frac{1}{2}$$

onde R_x e R_y são calculados usando as seguintes máscaras:

$$R_x = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \quad R_y = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

O resultado final das filtrações usando as máscaras de Sobel, Robert e Prewit consiste em gerar uma nova imagem, onde cada ponto corresponde a uma soma dos produtos termo a termo das matrizes, resultado da passagem dos filtros que correspondem as derivadas nas direções x e y elevadas ao quadrado. Assim calcula-se $S_x^2 + S_y^2$. Depois, pode-se fazer a extração da raiz quadrada em cada ponto. Mas, geralmente, nesta etapa é feita uma aproximação, onde esta operação é substituída pela soma dos valores absolutos de cada máscara $||S_x|| + ||S_y||$.

Parte 3:

A transformada de Hough

As transformadas são maneiras alternativas de mostrar a mesma informação, elas buscam usar as características que a imagem deixará mais evidente em **determinado domínio** e, assim, facilitar a análise da imagem.

A transformada de **Fourier** (ou de senos e cossenos) passa a mostrar a imagem decomposta em suas componentes de frequência, o que permite identificar os aspectos mais frequentes e comuns da imagem (por exemplo suas texturas). Ela representa, em cada ponto, a informação daquela frequência da imagem inteira. Isso permite filtrar, eliminar ou ampliar mais facilmente uma faixa específica de frequência (como discutido na seção "Convolução em Imagens").

A transformada de **Cossenos** decompõem a imagem em séries de cossenos, sendo muito usada para compressão de dados.

As transformadas de **Wavelets** passam a representar a imagem como coeficientes que serão multiplicados por diversas funções de base (Haar, Daubechies, etc). São usadas para filtragem e compressão de dados.

A transformada de **Distância**, substitui uma região por sua distância a um ponto. Serve como passo intermediário em diversas outras técnicas.

A transformada de **Hough** leva a imagem para o domínio dos parâmetros que descreveriam as figuras que você gostaria de achar nas imagens. Ela foi desenvolvida por Paul Hough em 1962 e patenteada pela IBM. Originalmente, foi elaborada para detectar características analíticas representáveis em imagens binárias, assim como linhas, círculos e elipses. Na última década, tornou-se uma ferramenta de uso comum na visão computacional para o reconhecimento destas características.

Forma geral

A transformada de Hough é aplicável quando se possui informações precisas das formas presentes na imagem. A transformada de Hough (HT) é um método padrão para detecção de formas parametrizadas em imagens computacionais. Em geral, a transformada é aplicada após a imagem sofrer o pré-processamento de detecção de bordas e da limiarização (discutida na Parte 1 deste texto).

A idéia é aplicar na imagem uma transformação tal que todos os pontos pertencentes a uma mesma curva ou reta sejam transformados num único ponto de um espaço dos parâmetros da curva ou reta procurada. O conceito principal da HT está em definir um mapeamento entre o espaço de imagem e o espaço dos parâmetros relevantes. Cada borda de uma imagem é transformada pelo mapeamento para determinar células no espaço de parâmetros, indicadas

pelas primitivas definidas através do ponto analisado. Essas células são incrementadas, e indicarão no final do processo, através dos máximos do acumulador, quais os parâmetros correspondentes a forma especificada.

Hough é um método de acumulação de requisitos muito geral. Permite detectar qualquer curva ou reta, mesmo pouco visível ou fortemente ruidosa. Como a imagem computacional (digital) é formada por um conjunto de pontos, os pixels, que possuem determinadas características pode-se dizer que a transformada de Hough consiste em mapear um *pixel* da imagem em uma curva no espaço de parâmetros, organizado em forma de um acumulador *n* dimensional, onde *n* corresponde ao número de parâmetros.

Detecção de Retas

No caso de uso desta transformada para detectar linhas retas nas imagens, há várias parametrizações possíveis para o espaço de parâmetros. Por exemplo, a equação das retas na forma declive/interseção, definida por $y = ax + b$, onde *a* representa a tangente da reta e *b* seu ponto de interseção com o eixo vertical como mostrado na figura 37. O uso desta forma de parametrização conduz no entanto a dificuldade prática de um espaço de parâmetro, devido ao fato do parâmetro *a* ter valor ilimitado para linhas que são paralelas ao eixo *y*, ou seja, essas linhas terão tangentes infinitas.

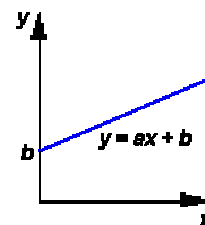


Figura 37: Representação da equação da reta na forma cartesiana de explícita

O algoritmo de Hough requer um acumulador de dimensão igual ao número de parâmetros que irão descrever a equação da família de curvas que são buscadas. Por exemplo, achar segmentos de linhas usando a equação $y = ax + b$, equivale a achar dois parâmetros para cada segmento de reta: *a* e *b*. Assim, as duas dimensões da matriz acumuladora para esta família correspondem aos valores 'quantizados' para *a* e *b*. Se o objetivo fosse encontrar os parâmetros de uma parábola do segundo grau, esse acumulador teria mais um elemento, de modo que sua dimensão passaria a ser 3.

Usando uma matriz acumuladora *A*, o procedimento de Hough examina cada pixel e calcula os parâmetros da curva (equação) especificada que passa pelo pixel. Caso uma imagem que não foi pré-

processada com algoritmo de detecção de bordas, esteja sendo analisada, fato incomum na transformada de Hough, será examinado o pixel e sua vizinhança na imagem, para determinar se há evidência de extremidade naquele pixel. Somente se houver, realizar-se-á o cálculo dos parâmetros.

Após calculados os parâmetros de um determinado pixel, eles são ‘quantizados’ para um valor correspondente de a e b , e o acumulador na célula correspondente $A(a, b)$ é incrementado. Cada par (x, y) no espaço da imagem dará origem a uma reta no espaço dos parâmetros. O conjunto dos pixels sobre a mesma reta no espaço (x, y) dará origem a um conjunto de retas que se interceptam em um ponto do espaço dos parâmetros.

Quando todos *pixels* tiverem sido processados, são procurados no acumulador A os maiores valores (picos). Eles indicam os parâmetros de prováveis linhas na imagem. A seguir é mostrado um pequeno exemplo utilizando essa técnica, cujo cálculo é realizado apenas para dois pixels pertencentes a uma linha (reta) de uma imagem, apresentada na figura 38, (plano x - y ou espaço real). Na figura 39 (plano a - b ou espaço de Hough), pode-se verificar as linhas geradas pelos parâmetros calculados, onde a linha verde refere-se ao ponto (1,1) e a azul ao ponto (2,2) do plano x - y .

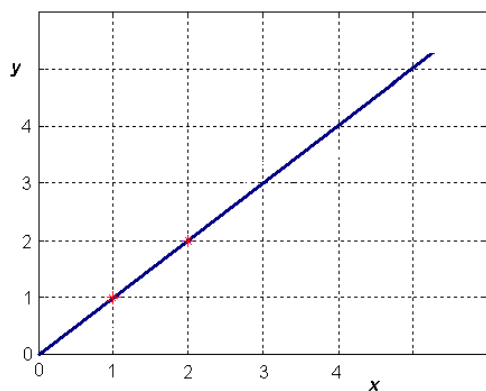


Figura 38: Plano x - y (imagem)

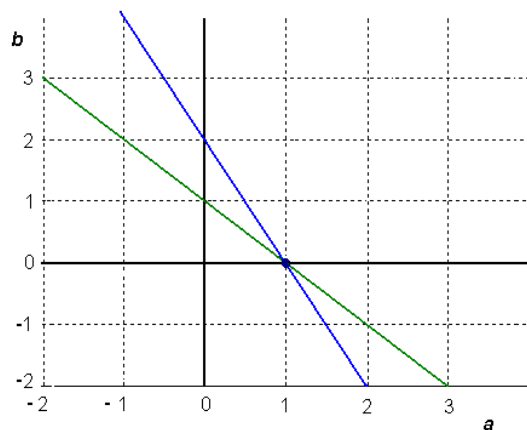


Figura 39: Plano a - b (parâmetros)

Para este exemplo (figura 38) foi estipulado o intervalo $[-2, 3]$ para a , e empregados os valores de x e y do *pixel* que está sendo examinado na equação $b = -ax + y$, possibilitando o cálculo do valor de b . Os valores são utilizados para incrementar o referido elemento do acumulador $A(a, b)$ e traçar uma reta no plano a - b . A Tabela 2 mostra os valores utilizados e os parâmetros obtidos.

Tabela 2: Cálculo do parâmetro b em função de cada a e indicação do elemento incrementado no acumulador A .

x	y	a	$b = -a.x + y$	Acumulador $A(a, b)$ incrementado
1	1	-2	3	$A(-2, 3)$
1	1	-1	2	$A(-1, 2)$
1	1	0	1	$A(0, 1)$
1	1	1	0	$A(1, 0) *$
1	1	2	-1	$A(2, -1)$
1	1	3	-2	$A(3, -2)$
2	2	-2	6	$A(-2, 6)$
2	2	-1	4	$A(-1, 4)$
2	2	0	2	$A(0, 2)$
2	2	1	0	$A(1, 0) *$
2	2	2	-2	$A(2, -2)$
2	2	3	-4	$A(3, -4)$

* único elemento do acumulador que é incrementado 2 vezes, indicando colinearidade entre esses pontos.

Ao procurar o máximo no acumulador $A(a, b)$, verifica-se que este indicará o elemento referenciado por $a = 1$ e $b = 0$. Isto significa que a equação $y = 1.x + 0$, ou simplesmente $y = x$, é a equação que melhor define uma reta que passa sobre os pontos (1,1) e (2,2) da imagem. Tendo esta informação faz-se um pós-processamento para determinar onde começa e termina a reta encontrada, caso haja interesse. Isso será muito importante para definir as retas de nossos moldes.

Um limiar pode ser utilizado quando procura-se o(s) máximo(s) no acumulador, a fim de determinar um valor mínimo de pontos colineares. Se o valor do acumulador não for superior ao do limiar então será considerado um ruído [18]. As detecções de outras formas utilizando a transformada de Hough usam o mesmo princípio, havendo alteração no número de parâmetros da equação que será empregada, e em consequência na dimensão do acumulador.

Retas na forma polar

Duda e Hart [10] utilizaram coordenadas polares para representação de uma linha, nesta forma, as linhas podem ser completamente parametrizadas

usando a distância r , e a orientação q do vetor normal da linha na imagem original (figura 40). Usando esta parametrização, todo o ponto (x,y) na linha satisfará a equação:

$$r = x \cdot \cos(q) + y \cdot \sin(q).$$

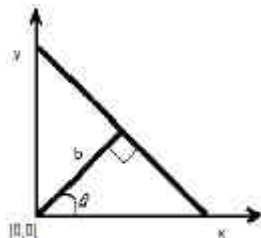


Figura 40: Representação da equação da reta em coordenadas polares

Calcula-se a transformada de Hough dos *pixels* da imagem, registrando os resultados em um acumulador (matriz) bidimensional. Cada *pixel* (x,y) do espaço real produz uma linha senoidal no espaço de Hough em coordenadas polares como descrito anteriormente. Todas as retas possíveis passando por um *pixel* (x,y) serão mapeados numa senóide de modo que, para a representação de Duda-Hart, a linha no espaço de Hough parecerá como mostrado na figura 41. É claro que existe uma infinidade de retas passando possivelmente por esse ponto, assim pode-se, por exemplo, considerá-lo um incremento de 10 em 10 graus ou $0,05\pi$. A reta comum a 3 *pixels* será dada pela interseção das 3 senóides que passam por esses *pixels* (figura 41). A interseção das 3 senóides representa os parâmetros da reta que passa pelos 3 pontos na forma cartesiana.

O acumulador torna-se complicado quando a imagem contiver muitas linhas, porque muitas senóides podem não correspondem à linhas, mas a ruídos. Assim deve-se interativamente descobrir a maior linha da imagem, remover sua contribuição do acumulador, e então repetir o processo para achar sucessivamente a maior das linhas restantes. A cada repetição acha-se o máximo global e obtém-se a equação da linha correspondente. Um pós-processamento pode ser usado para achar os pontos de começo e fim de cada linha.

Deteção de Círculos

Para círculos a implementação sugerida por Davies [18] é uma das mais interessantes. Círculos são parametrizados por 3 parâmetros x , y , e r , onde (x,y) referem-se a posição de centro e r ao raio, conforme figura 42. Neste caso, o espaço dos parâmetros é 3D.

Para evitar muito custo computacional nesta transformada de Hough 3-D, o problema é organizado em duas fases:

1. achar o centro de todos círculos;

2. achar o raio de cada círculo.

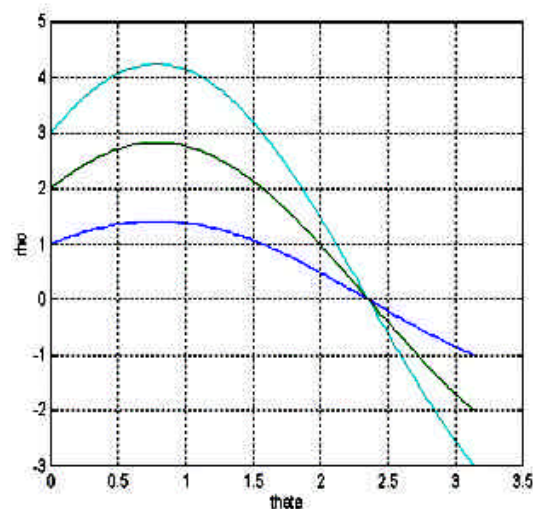
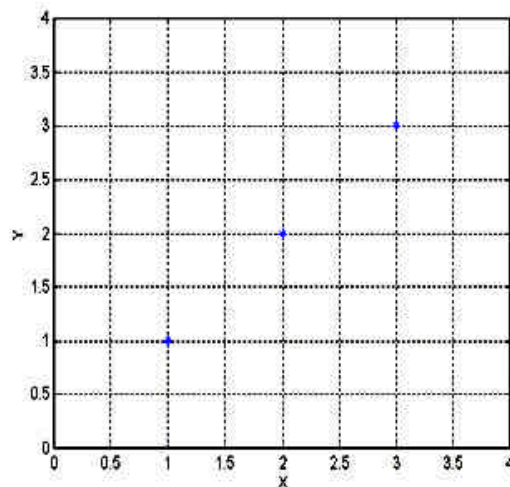


Figura 41: Acima imagem original com 3 pontos $(1,1)$, $(2,2)$ e $(3,3)$ no plano x,y . Abaixo cada um destes pontos descritos na forma como uma senóide.

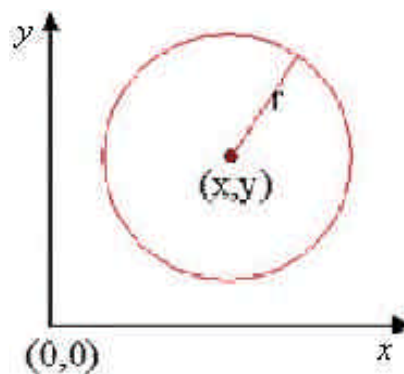


Figura 42: Representação geométrica de um círculo.

Observe que dado qualquer *pixel* em uma

circunferência, o centro do círculo localiza-se sobre a normal da tangente ao círculo naquele *pixel*. Assim, dado vários *pixels* do mesmo círculo, as normais deles se cruzarão no centro do círculo. Ou seja, definindo-se um acumulador dos valores de (x,y) , no espaço de parâmetros, deve-se primeiro achar o centro que correspondente à imagem. Para cada *pixel* pode-se calcular a tangente como a linha que se ajusta melhor a todos *pixels* de uma vizinhança pequena (usando o método dos mínimos-quadrados). Isto permite calcular a normal e registrá-la no histograma. Os máximos do acumulador indicam os locais de centros dos possíveis círculos. O próximo passo é achar o raio correspondente ao centro (ou raios para os círculos múltiplos, centrados no mesmo local). Faz-se isso através de um histograma para cada centro. Ou seja, para todo *pixel* na imagem é calculado sua distância até o centro de determinado círculo. Esta distância é registrada em um histograma unidimensional (vetor). O máximo dos histogramas correspondem aos raios dos círculos. Finalmente deve-se aplicar um passo de pós-processamento para decidir se os parâmetros encontrados correspondem a um círculo na imagem.

Detecção de Elipses

Uma elipse é definida por cinco parâmetros, porém cinco parâmetros na Transformada de Hough é inviável computacionalmente. Então será mostrado abaixo a aproximação de Yuen que separa a tarefa em duas passagens [35]:

1. identificação do centro da elipse, isto requer dois parâmetros da Transformada de Hough.
2. avaliação dos outros três parâmetros (referentes aos eixos) usando uma implementação focada da Transformada de Hough.

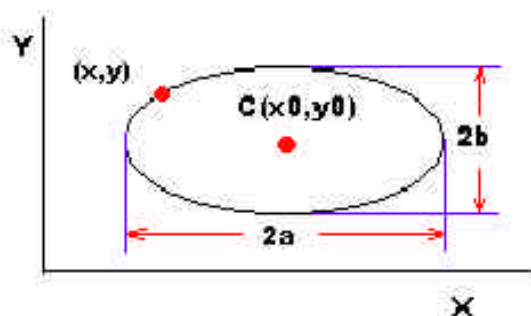


Figura 43: Parâmetros da elipse paralela aos eixos x e y.

Em coordenadas retangulares os pontos sobre a elipse devem satisfazer a equação:

$$\frac{(x - x_0)^2}{a^2} + \frac{(y - y_0)^2}{b^2} = 1$$

onde supõem-se que o centro da elipse seja o ponto de coordenadas (X_0, Y_0) , o eixo maior seja paralelo ao

eixo horizontal, sendo **a** e **b** os raios maiores e menores respectivamente. Assim, para a definição de uma elipse precisa-se conhecer (X_0, Y_0) , **a** e **b**. Se a elipse fizer um ângulo com o eixo horizontal, tem-se este ângulo como um parâmetro adicional. O centro da elipse pode ser definido como segue. Considere dois pontos *P* e *Q* (veja figura 44) em uma elipse e calcule as tangentes destes pontos. Onde *r* é o ponto onde estas tangentes cruzam-se e *M* o ponto central do segmento *PQ*. Para uma elipse, o centro ficará sobre a linha que se origina em *r* e passa por *M*. Linhas formadas por pares diferentes de pontos da elipse cruzar-se-ão no centro da mesma. Esses parâmetros da transformada de Hough são usados para acumular estas linhas, sendo que o máximo do acumulador corresponderá a interseção das mesmas. Como este algoritmo registra uma entrada para cada par de *pixel* na imagem, ele é computacionalmente muito custoso.

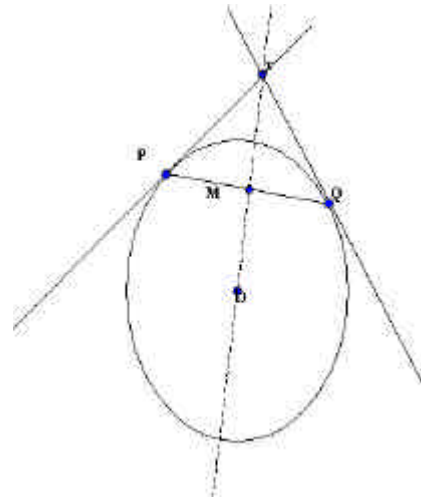


Figura 44: Representação geométrica de uma elipse não paralela aos eixos horizontal e vertical

Outras formas de Transformada de Hough

A transformada randômica de Hough apresentada em 1989 por Lei Xu, Erkki Oja e Pekka Kultanen, considera que na detecção de uma forma, um par de *pixel* é randomicamente escolhido, e os parâmetros da forma que passam através desses *pixels* são computados. Esta forma é registrada como uma entrada no acumulador. Este procedimento é repetido um número predeterminado de vezes, onde o número de vezes é bem menor que a quantidade de pares de *pixels* da imagem. Este algoritmo é repetido para cada segmento de linha a ser detectado. Assim o máximo do acumulador é achado, e a equação da forma é computada. Então, os *pixels* da imagem sob essa forma são removidos, sobrando uma imagem mais simples para ser analisada. O processo é repetido para encontrar a próxima forma; e somente pára quando nenhuma forma for detectada em um determinado número de iterações.

A transformada **Hierárquica de Hough** é baseada em uma estrutura de pirâmide, onde cada camada divide a imagem em sub-imagens. A pirâmide convencional é confeccionada com $P+1$ níveis. Em cada nível l , a imagem é dividida em $2^{P-l} \times 2^{P-l}$ sub-imagens, onde P é o nível superior da pirâmide, representando a imagem completa. No nível inferior da pirâmide, pequenos segmentos de linhas são detectados através da transformada de Hough em uma pequena sub-imagem, facilitando a diferenciação entre linhas 'reais' e pontos casualmente alinhados. O método é também relativamente eficiente, já que somente um pequeno vetor acumulador é necessário para representar todas possíveis linhas que passam através da sub-imagem. Tipicamente, cada sub-imagem contém duas linhas. Esses resultados são propagados através da pirâmide, para obter-se a representação global dos segmentos de linha, formados a partir do agrupamento de linhas existentes nas sub-imagens [23].

A Transformada **Combinatorial de Hough** foi apresentada por D. Ben Tzvi e M. B. Sandler em 1989, como a forma mais eficiente de calcular-se a Transformada de Hough. Cada 2 pontos na imagem indicam uma possível colinearidade. Se (r_0, θ_0) , são os parâmetros que descrevem a linha que une os dois pontos, então os parâmetros do segmento de linha podem ser derivados da equação paramétrica :

$$r = x \cdot \cos(\theta) + y \cdot \sin(\theta),$$

dados os pontos (x_1, y_1) e (x_2, y_2) .

$$1) r_0 = x_1 \cdot \cos(\theta_0) + y_1 \cdot \sin(\theta_0)$$

$$2) r_0 = x_2 \cdot \cos(\theta_0) + y_2 \cdot \sin(\theta_0)$$

subtraindo as duas equações acima temos:

$$\theta = \arctan(Dx / Dy)$$

Para determinar se qualquer parte de um conjunto m de pontos é colinear, todas as possíveis combinações de segmentos de linha entre dois pontos precisam ser calculadas.

$$C_m^2 = m! / (2!(m-2)!)$$

Cada um destes cálculos, cada par de ponto, contribui para o incremento do acumulador na posição (r_0, θ_0) .

Introduzida em 1990 por N. Kiryati, Y. Eldar e A. M. Bruckshtein., a **transformada probabilística** de Hough mostra que para descobrir objetos é suficiente computar a transformada de Hough só de uma porção de pixels na imagem. Estes *pixels* são aleatoriamente escolhidos de acordo com uma função de densidade de probabilidade uniforme, definida de acordo com a imagem. Isto é equivalente a computar a transformada padrão de uma amostra da imagem.

Um método **unificado para detectar por Hough qualquer curva genérica de segundo grau**

foi desenvolvido por M. M. G. Macedo em 2005 [18]. Esse método permite identificar qualquer uma das cinco formas cônicas possíveis (retas, arcos de círculos, círculos, elipses, parábolas ou hipérboles) que possam estar presentes na imagem. Permite também definir os parâmetros destas curvas mesmo que qualquer número delas esteja presente, em qualquer ordem, com qualquer combinação e ainda mesmo que com ruídos, falhas ou usando só desenhos de forma aproximada.

Finalmente a **transformada Generalizada de Hough** trata de um única maneira qualquer padrão (definido por um conjunto de pontos) presente na imagem, através da busca das possíveis transformações afins que identifiquem a forma que se procura. Sendo mais uma forma de reconhecimento de padrões flexíveis do que de busca de equações nas imagens [31].

Agradecimentos

O primeiro autor agradece ao CNPq pelo apoio financeiro.

Referências

- [1]Berssen, J., "Dynamic thresholding of grey level images", Proc. Eighth Int. Comf. Pattern Recognition, pp. 1251-1255, Paris, 1986.
- [2]Bovik , Al. Handbook of Image and Video Processing. Academic Press Limited. 2000.
- [3]Brigham, E.O.. The fast fourier transform and its applications. Prentice Hall. 1988.
- [4]Castleman, K.R.. Digital image processing. Prentice Hall. 1996.
- [5]Chow, C. K. , Kaneko, T. "Automatic boundary detection of left ventricle from cineangiograms", Computer in Biomedical Research, vol. 5, pp. 388-410, 1972.
- [6]Conci, A. "Introdução ao Processamento de Imagens ", curso de Análise de Imagens UFF: <http://www.ic.uff.br/~aconci/AI.html/intHomo.eps>, 2001.
- [7]Conci, A., Proença, C. B., "A system for real-time fabric inspection and industrial decision", Proceedings of ACM International Conference on Software Engineering & Knowledge Engineering, Ischia, Itália, Julho 15 - 18, 2002, KSI - Knowledge Systems Institute, pp. 707-714, ACM-SIGSOFT-ACM International Conference Proceedings Series, KSI ,Skokie, USA:
- [8]Conci, A., Proença, C. B., "A Comparison between Image-processing Approaches on Textile

- Inspection", Journal of the Textile Institute - ISSN 0400-5000, Manchester, Ref.: 76-98, URL: <http://www.texti.org.>, vol. 91 Part 1- Fiber Science and Textile Technology, No. 2, pp.317-323, 2000.
- [9]Dougherty, E.. Random processes for image and signal processing. SPIE Optical Engineering Press. 1998.
- [10]Duda, R.O.; Hart, P.E.. Pattern recognition and scene analysis. John Wiley & Sons. 1978.
- [11]Gabor, D., Theory of communication. Journal of the Institute of Electrical Engineers, v. 93, n. 26, p. 429-457, 1946.
- [12]Gonzalez, R.C., Woods, R.E., Digital Image Processing. 2.Ed. New Jersey: Prentice Hall, 2002.
- [13]Jain, R., Kasturi, K. Schunck, B.G. "Machine Vision", The Mit Press and Mc Graw Hill, 1995.
- [14]Jain , R., Pandit, S.M., "Image Decomposition and Automatic Multithresholding Using Histogram Modes, 1997.
- [15]Kasturi, R., Schinck, B.G., *Machine Vision*, The Mit Press and Mc Graw Hill, 1995.
- [16]Lemnie,P.;Trevathan,C.; Van Essen, D.; Wassle,H.. Parallel processing of visual information. Visual Perception: The Neurophysiological foundation. L. Spillman and J. Werner. Ed. Academic Press, New York , 1990.
- [17]Lim, J. S., "*Two-Dimensional Signal and Image Processing*", Prentice- Hall, 1990.
- [18]Macedo, M.M.G. , "Uso da Transformado de Hough na vetorização de moldes e outras aplicações", dissertação de mestrado, Dep. Ciência da Computação, UFF, 2005.
- [19]Marr, D.; Hildreth, E..Theory of edge detection. Proceedings of the Royal Society of London. Series B. Vol. 207. 187-217. 1980
- [20]Marshall, S.. Review of shape coding techniques. Image and Vision Computing, 7(3):281-294, November 1989.
- [21]Martins, O.R. "Segmentação de Histogramas Multimodais: Simulações, Implementações e Aplicação em Cheques Bancários Brasileiros", dissertação de mestrado, PPGIA - PUC-PN, 2000.-
- [22]Nowak, R. "Fast Convolution Using the FFT". <<http://cnx.rice.edu/content/m10963/latest/>>, 2003.
- [23]Oliveira, D.B. , Conci A. Segenreich, S.A. "Uma metodologia para vetorização controlada de moldes para a industria do vestuário", Anais do I Congresso de Dinamica, Controle e Aplicacoes, vol1. in Capítulo 26 Dinâmica, Controle e Aplicações- Proc. DINCON 2002, Rio Claro, SP, pp. 825-830. Editado por J. M. Balthazar, M. Boaventura, G. N. Silva e M. Tsuchida, Sociedade Brasileira de Matemática Aplicada e Computacional - SBMAC. Trab. P22, 2002.
- [24]Parker, J.R. *Algorithms for Image Processing*, John Wiley & Sons, Inc. Englewood Cliffs, NJ: 1996
- [25]Pavlidis, T. Algorithms for graphics and image processing. Computer Science Press. 1982.
- [26]Pratt, W.K.. Digital image processing. John Wiley & Sons. Third edition.1989.
- [27]Romberg, J. The Fast Fourier Transform (FFT).<<http://cnx.rice.edu/content/m10783/latest/>>, 2004.
- [28]Schalkoff, R. J. - Digital Image Processing and Computer Vision - John Wiley & Sons, Inc., 1989.
- [29]Sriganesh, M.S., Fast Fourier Transform (FFT). <<http://www.howtodothings.com>>, 2002.
- [30]Trier, O.D. e Jain, A.K., "Goal-directed evaluation of binarization methods", IEEE Trans. Pattern Anal. Machine Intell., vol. 17, Dec., pp.1191-1201, 1995.
- [31]Watt, A. and Policarpo, F. *The Computer Image*, Addison-Wesley, 1998.
- [32]Wayne, K., Fast Fourier Transform. <<http://www.cs.princeton.edu/~wayne/cs423/lectures/fft-4up.pdf>> 2000.
- [33]Weeks, Jr., A. R., *Fundamentals of Eletronic Image Processing*, SPIE/IEEE Press, (series on imaging science & engineering.) NJ: 1996
- [34]Yen, J.C. Chang F.J., Chang, S., "A new criterion for automatic multilevel thresholding", IEEE Trans. Image Process, vol. 4, No. 3, pp.370-378, 1995.
- [35]Yuen, H. K. Illingworth J. Kittler J. , "Detecting partially occluded ellipses using the Hough transform", *Image and Vision Computing*, vol. 7, no. 1, pp. 31-37, Feb 1989.
- [36]"*The FFT Demystified v2.1*", Engineering Productivity Tools Ltd., in: www.eptools.com/tn/ , 1999.