

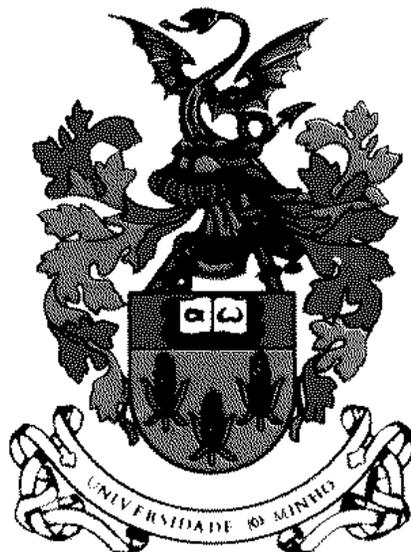
Universidade do Minho

Escola de Engenharia

Departamento de Informática

Largo do Paço - 4709 BRAGA Codex- PORTUGAL

Tel +351-53-604470- Fax +351-53-612954



Visão por Computador

Luís Paulo Peixoto dos Santos

Texto de Apoio
UM-DI-TA-95-02

03/Mar/1995

Visão por Computador

Luís Paulo Peixoto dos Santos

psantos@di.uminho.pt

Departamento de Informática
Universidade do Minho
Largo do Paço
4709 BRAGA Codex
PORTUGAL

Tel +351-53-604470

Fax +351-53-612954

Texto de Apoio

UM-DI-TA-95-02

03/Mar/1995

Índice

1	Processamento de Imagem versus Visão por Computador	1
2	Características de uma imagem digital	2
2.1	Problemas do processo de digitalização	3
3	Arquitectura para visão por computador	4
4	Brilho e contraste	4
5	Histograma	5
5.1	Equalização do histograma	7
6	Convolução	8
6.1	Filtros passa-baixo (<i>smoothing filters</i>)	10
6.2	Filtros passa-alto (<i>edge detection</i>)	12
6.3	Filtros não-lineares	14
6.4	Outros métodos	15
7	Segmentação	16
7.1	Binarização	16
7.2	Binarização adaptativa	18
7.3	Operadores morfológicos	18
7.4	Textura	20
7.5	Segmentação baseada em regiões	23
7.6	Movimento	24
8	Detecção de formas	27
8.1	<i>Template matching</i>	28
8.2	A transformada de Hough	29

9	Descrição de formas	31
9.1	<i>Chain code</i>	31
9.2	Assinaturas	32
9.3	Aproximação poligonal	32
9.4	Esqueleto	34
	Referências bibliográficas	35

Lista de Figuras

1	Níveis de processamento	2
2	Arquitetura para visão	4
3	Histogramas típicos	6
4	Equalização do histograma	7
5	Imagem $n \times n$	9
6	Detecção de contornos	10
7	Aplicação da média	11
8	Magnitude do gradiente (Sobel)	13
9	Cálculo do Laplaciano	13
10	Aplicação da mediana	15
11	Histograma bimodal	17
12	Histograma n-modal	17
13	Erosão	19
14	Operador $P(d, \theta)$	21
15	Partições do espectro	23
16	Region splitting & merging	25
17	Objecto em movimento	25
18	Diferenças acumuladas	26
19	AADI, PADI e NADI	27
20	Template matching	28
21	Pirâmide de resolução	29
22	Coordenadas polares	30
23	Transformada de Hough	30
24	Identificadores para chain codes	31
25	Chain codes	32

26	Assinatura em função da distância	33
27	Assinatura em função do ângulo	33
28	Aproximação poligonal	33
29	Aproximação poligonal com grande precisão	34
30	Esqueletos de várias regiões	35

1 Processamento de Imagem versus Visão por Computador

O interesse no processamento de imagens digitais surge no âmbito de duas áreas de aplicação: melhoria do aspecto da imagem para observação humana e processamento da imagem para percepção autónoma da máquina.

Estas duas áreas não se excluem mutuamente, aliás interceptam-se em muitos pontos. Para extrair informação sobre a localização de um objecto numa imagem, por exemplo, é muitas vezes necessário melhorar o aspecto dessa imagem, retirando o ruído e melhorando o contraste.

Por processamento de imagem designam-se as técnicas que permitem melhorar o aspecto da imagem. Estas técnicas têm como característica receberem como *input* uma imagem, e gerarem como *output* uma outra imagem, que resulta da primeira mas que sofreu algumas alterações.

Visão por computador designa a ciência que extrai, de forma automática, informação da imagem para percepção autónoma da máquina. Exemplos de aplicações neste campo são o reconhecimento de caracteres, controlo de *robots* industriais, localização de objectos em movimento, etc. As técnicas de visão por computador recebem como *input* uma imagem e/ou estruturas de dados e geram outras estruturas de dados que representam alguma(s) característica(s) da imagem.

Uma imagem digitalizada contém uma grande quantidade de informação:

“Uma imagem vale por mil palavras”

Os algoritmos de processamento de imagem lidam com toda esta informação, sendo portanto bastante pesados. Os algoritmos de visão por computador lidam com estruturas de dados abstractas que representam características da imagem (localização de objectos, contornos de objectos, texturas, etc.). A quantidade de informação a processar é bastante menor.

Os métodos que lidam com toda a imagem e geram outra imagem (como a remoção de ruídos, por exemplo) são classificados de **baixo nível**. Os métodos que recebem como *input* uma imagem e geram uma estrutura de dados que representa alguma característica dessa imagem (como a descrição do contorno de um objecto) são classificados de **nível intermédio**. Os métodos que lidam apenas com estruturas de dados abstractas e geram outras estruturas de dados ou tomam decisões baseados no seu *input* são classificados de **alto nível**. À medida que aumenta o nível, diminui a quantidade de dados a processar mas aumenta a complexidade do algoritmo (fig. 1).

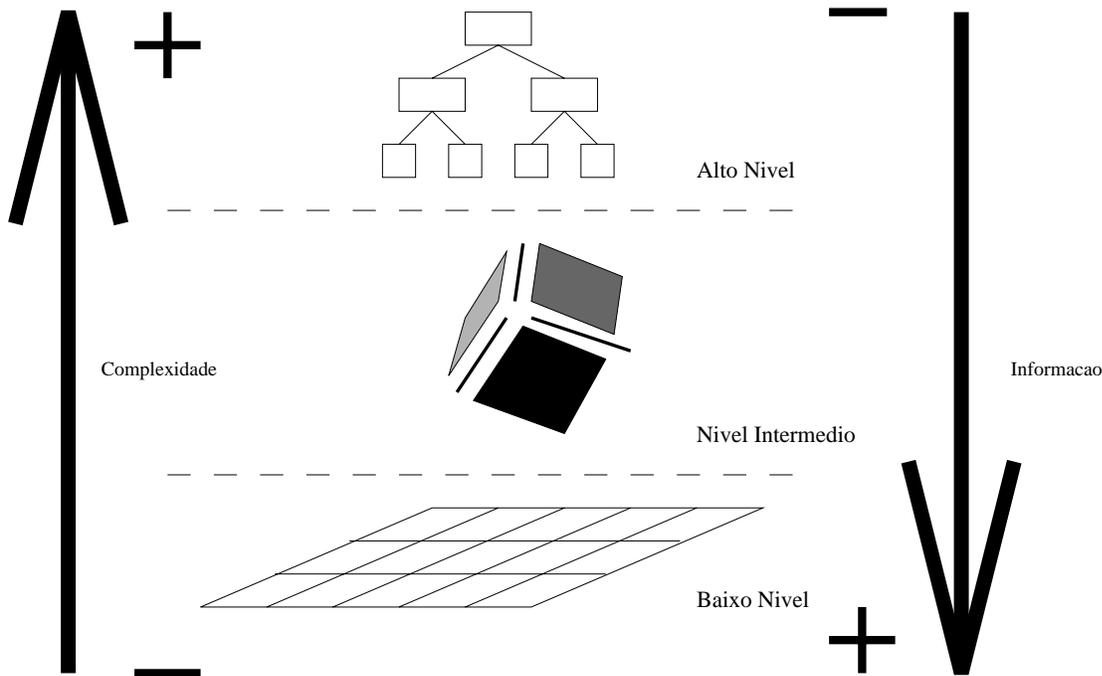


Figura 1: Níveis de processamento

2 Características de uma imagem digital

Uma imagem é uma função bidimensional $f(x, y)$ da intensidade da luz, onde x e y representam as coordenadas espaciais (localização) de um ponto na imagem e $f(x, y)$ representa o brilho, ou a cor, da imagem nesse ponto. Uma imagem pode ser vista como tridimensional, com o terceiro eixo representando o brilho da imagem. Assim os pontos mais brilhantes são visualizados como sendo mais altos, enquanto os pontos mais escuros aparecem como sendo mais baixos. Zonas da imagem com grandes variações no brilho (contornos dos objectos, por exemplo) são visualizadas como planos inclinados. Zonas com um brilho constante apresentam-se como planas, lembrando planícies ou planaltos, dependendo do seu brilho. Zonas com muito ruído apresentam picos de altura aleatória e distribuídos aleatoriamente.

Uma imagem digitalizada consiste numa matriz bidimensional de inteiros que representam o brilho, ou a cor, de cada ponto da cena. Cada ponto desta matriz é normalmente designado por 'pixel', o que corresponde à abreviatura de *picture element*.

O processo de digitalização exige que sejam tomadas decisões relativamente ao número de pontos em que a imagem é amostrada (dimensão da imagem) e o número de diferentes tons que cada ponto pode tomar (tamanho de cada elemento da matriz). Como os computadores são máquinas binárias é vantajoso que estas quantidades sejam potências de dois. Valores típicos para a dimensão da imagem são 256x256, 512x512, 1024x1024.

A dimensão de cada elemento da matriz, isto é, cada ponto da imagem, determina o número de diferentes brilhos, ou cores, que cada ponto pode apresentar. As imagens digitalizadas em *true color* têm 24 bits para cada ponto. Estes 24 bits estão subdivididos em 3 bytes, cada um representando a quantidade de vermelho, verde e azul que devem ser combinadas para obter a cor desejada. Este sistema é designado por RGB.

As imagens monocromáticas não têm cores, mas sim tons de cinzento. O valor 0 representa preto (ausência de brilho), o maior valor possível representa branco e os valores intermédios representam os vários tons de cinzento. É vulgar as imagens monocromáticas apresentarem 256 tons de cinzento, sendo necessário 1 byte para representar cada pixel.

Como a imagem digital é uma aproximação a uma imagem contínua é natural interrogarmo-nos sobre qual a dimensão que a matriz digital deve ter e quantas cores ou tons de cinzento são necessários para obter uma boa aproximação. Por resolução entende-se o grau de discernimento de detalhes que a imagem apresenta. A resolução depende da dimensão da imagem (n^o de linhas e colunas) e do número de tons que esta apresenta. Quanto maiores forem estes parâmetros, maior é a qualidade da imagem digital. Infelizmente aumenta também a quantidade de informação a armazenar e, conseqüentemente, o poder de computação necessário para a processar. Quando a imagem se destina a visualização humana a resolução não deve ir além daquilo que o olho humano é capaz de ver. Este é limitado no número de cores e no tamanho dos pormenores que consegue distinguir. É difícil definir o que é uma boa imagem, pois os requerimentos de qualidade variam de aplicação para aplicação.

2.1 Problemas do processo de digitalização

Numa câmara, a luz proveniente de uma cena tridimensional é focada, através das lentes, numa superfície bidimensional. A luz é então amostrada numa grelha de sensores distribuídos uniformemente nesta superfície. A imagem digital não é mais do que uma matriz bidimensional de valores inteiros que representam a intensidade da luz em cada um destes pontos.

A imagem digital não é perfeita, sendo vários os factores que podem contribuir para a sua imperfeição:

- Intervalo de amplitudes suportado pelos sensores;
- Intervalo espectral suportado pelos sensores;
- Tempo de amostragem;
- Desfasamento temporal entre os sensores;
- Erros de quantificação

O sinal digital não pode conter tanta informação como o sinal analógico, devido à sua natureza discreta. Os erros de quantificação são portanto inevitáveis. As altas frequências do sinal analógico não podem ser representadas se estão acima do limite de Nyquist, que é metade da frequência de amostragem. Teoricamente, a frequência de amostragem tem que ser pelo menos o dobro da mais alta frequência que existir no sinal analógico. Na prática, isto aumenta a resolução do sinal digital, pelo que a solução mais comum é filtrar as frequências acima do limite de Nyquist. Quando estas frequências não são removidas (filtradas) surge o efeito designado como *aliasing*.

3 Arquitectura para visão por computador

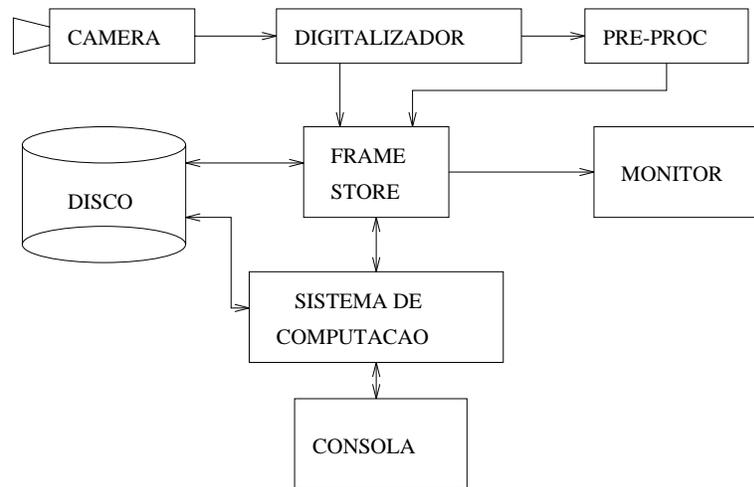


Figura 2: Arquitectura para visão

A figura 2 apresenta um sistema de processamento de imagem genérico.

A placa digitalizadora (*frame grabber*) é responsável por converter a imagem analógica para digital. A *frame store* é uma zona de memória onde a imagem é mantida e pode ser acedida pelos diversos módulos. O pré-processador é responsável pelas operações de baixo nível, como alteração do brilho e contraste, convolução, cálculo de transformadas, etc. Este pode ser um processador digital de sinal (DSP), uma máquina tipo SIMD ou um processador vectorial.

O sistema de computação é responsável por operações de alto nível. Para que seja possível obter resultados em tempo real esta máquina deve ter um grande poder de processamento. O exemplo mais comum são máquinas de memória distribuída com um grande número de processadores. A máquina existente na Universidade do Minho é baseada em Transputers. No entanto, estes processadores estão bastante ultrapassados como nodos de computação.

4 Brilho e contraste

Muitas técnicas de processamento de imagem são baseadas na manipulação de propriedades estatísticas das imagens. O brilho e o contraste são exemplos destas propriedades. O brilho pode ser calculado como o valor médio de todos os pixels de uma imagem. Se $v_{(i,j)}$ é o brilho do pixel da linha i , coluna j então o brilho da imagem \bar{v} é:

$$\bar{v} = \frac{1}{n} \sum_{i,j} v_{(i,j)}$$

onde n representa o número de pixels.

O desvio padrão, σ , dá uma medida do contraste:

$$\sigma = \sqrt{\frac{1}{n} \sum_{i,j} (v_{(i,j)} - \bar{v})^2}$$

Esta expressão tem o inconveniente de exigir 2 passagens pela imagem, pois a média tem que ser calculada primeiro. No entanto pode ser reescrita da seguinte forma:

$$\sigma = \sqrt{\frac{1}{n} \sum_{i,j} v_{(i,j)}^2 - \frac{1}{n^2} (\sum_{i,j} v_{(i,j)})^2}$$

É possível alterar os valores dos pixels de forma a alterar o brilho e contraste da imagem para valores desejados. Para tal é necessário calcular 2 grandezas: g (ganho) e b (bias). Estas são baseadas no brilho pretendido ($\overline{v_{new}}$), contraste pretendido (σ_{new}), brilho anterior (\overline{v}) e contraste anterior (σ).

$$g = \frac{\sigma_{new}}{\sigma} \quad , \quad b = \overline{v_{new}} - g\overline{v}$$

Os novos valores dos pixels são dados por

$$v_{(i,j)out} = gv_{(i,j)} + b$$

O método apresentado tem o inconveniente de calcular o ganho e o bias em toda a imagem, não tomando em consideração as características locais de cada região da imagem. Uma alternativa com melhores resultados consiste em calcular ganhos e bias diferentes para cada ponto da imagem, baseados na informação existente numa vizinhança $m * m$ de cada pixel.

Para cada vizinhança $m * m$ do ponto (i, j) são calculadas a média e desvio padrão, $\overline{v_{(i,j)}}$ e $\sigma_{(i,j)}$, respectivamente. O ganho e bias para cada ponto são dados por

$$g_{(i,j)} = \frac{\sigma_{new}}{\sigma_{(i,j)}} \quad , \quad b_{(i,j)} = \overline{v_{new}} - g_{(i,j)}\overline{v_{(i,j)}}$$

O novo valor do pixel (i, j) é dado por

$$v_{(i,j)out} = g_{(i,j)}v_{(i,j)} + b_{(i,j)}$$

Este método permite obter melhores resultados mas exige mais tempo de processamento pois para cada pixel têm que ser calculados várias grandezas. Quanto maior é a vizinhança considerada, mais pesado se torna o problema. É, no entanto, um ótimo algoritmo para paralelizar, usando máquinas MIMD ou SIMD, podendo facilmente ser decomposto no domínio dos dados.

O conceito de vizinhança levanta alguns problemas nas margens da imagem. Para estes pontos não é possível calcular \overline{v} e σ pois a vizinhança $m * m$ não existe.

5 Histograma

O histograma de uma imagem indica o número de pontos que existem para cada tom de cinzento, ou cor, possíveis nessa imagem. Por exemplo, se numa imagem são possíveis 4 tons de cinzento, variando de 0 a 3, e a imagem tem 16 pontos

4	2	8	2
---	---	---	---

é um histograma $h(v)$ dessa imagem indicando que 4 pontos são do tom 0, 2 do tom 1 e assim sucessivamente.

O histograma normalizado $H(v)$ é dado por

$$H(v) = \frac{h(v)}{n}$$

em que n é o número de pontos na imagem. Este histograma é obtido dividindo cada elemento de $h(v)$ por n .

0.25	0.125	0.5	0.125
------	-------	-----	-------

Cada elemento indica a probabilidade de um pixel da imagem ter um determinado tom. Neste exemplo, a probabilidade de um pixel ter o tom 2 é 0.5. O histograma é uma função de distribuição.

O histograma dá uma boa representação gráfica do brilho e contraste associados a uma imagem (fig. 3).

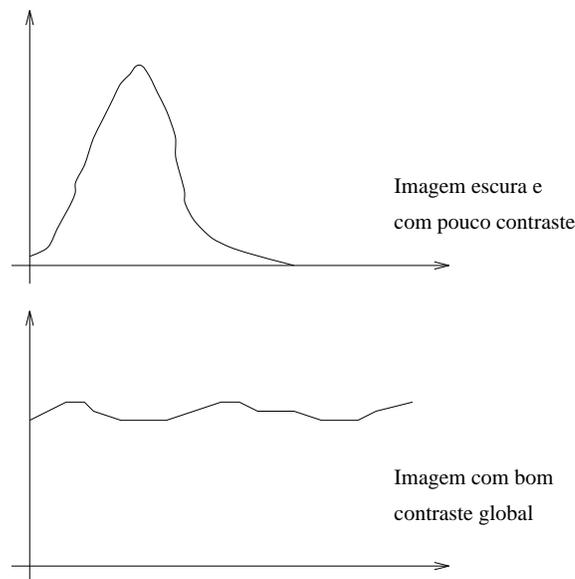


Figura 3: Histogramas típicos

O histograma acumulado $H_{acc}(v)$ é dado por

$$H_{acc}(v) = \sum_{i=0}^v H(v_i)$$

Cada elemento deste histograma indica a probabilidade de um pixel ter um valor menor ou igual a um determinado tom. No nosso exemplo temos H_{acc} igual a

0.25	0.375	0.875	1.0
------	-------	-------	-----

A probabilidade de um pixel ter um tom menor ou igual a 2 é 0.875.

5.1 Equalização do histograma

Esta técnica permite aumentar, de uma forma automática, o contraste duma imagem.

Uma imagem apresenta um bom contraste global quando os pontos se encontram distribuídos uniformemente pelos vários tons possíveis. Se todos os pontos têm valores concentrados numa pequena faixa do intervalo de tons o contraste é baixo. A equalização do histograma consiste em distribuir os valores dos pixels pelo intervalo de tons disponível (fig. 4).

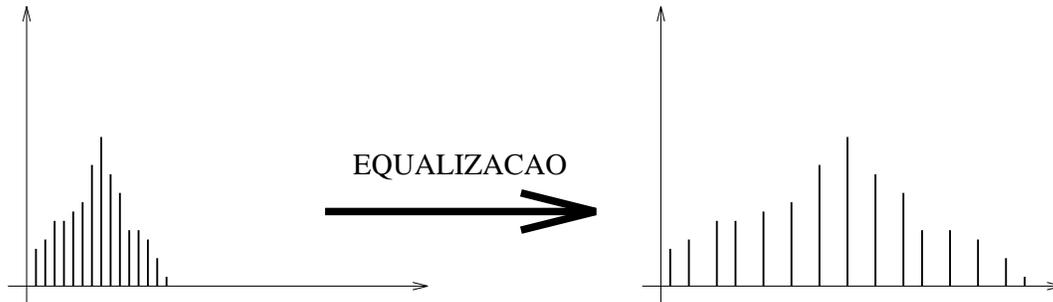


Figura 4: Equalização do histograma

ALGORITMO

1. Calcular histograma
2. Calcular histograma normalizado
3. Calcular histograma acumulado
4. Multiplicar histograma acumulado pelo maior tom possível
5. Usar a tabela anterior como função de transferência para os valores dos pixels

EXEMPLO

Imagem de 4x4 com 8 tons a variar entre 0 e 7:

Imagem original

0	0	1	2
0	0	1	2
1	1	1	2
2	1	2	2

Histograma

4	6	6	0	0	0	0	0
---	---	---	---	---	---	---	---

Histograma normalizado

0.25	0.375	0.375	0	0	0	0	0
------	-------	-------	---	---	---	---	---

Histograma acumulado

0.25	0.625	1.0	1.0	1.0	1.0	1.0	1.0
------	-------	-----	-----	-----	-----	-----	-----

Função de transferência

2	4	7	7	7	7	7	7
---	---	---	---	---	---	---	---

**Imagem
final**

2	2	4	7
2	2	4	7
4	4	4	7
7	4	7	7

O método descrito sofre dos inconvenientes de todos os métodos que apenas tomam em conta informação global, não considerando as características locais de cada região da imagem.

A equalização, tal como foi descrita, calcula a função de transferência baseada no histograma de toda a imagem e usa esta função para todos os pixels. No entanto, diferentes regiões da imagem podem apresentar brilhos e contrastes muito diferentes. Para obter melhores resultados a equalização deve ser local. Para cada pixel (i, j) calcula-se o histograma da sua vizinhança $m * m$. Baseado nesta histograma é calculada uma função de transferência, que é usada apenas para o ponto (i, j) . Esta operação repete-se para todos os pontos da imagem, excepto para as margens pois para estes pontos a vizinhança $m * m$ não existe.

Esta técnica é muito mais pesada mas permite obter melhores resultados. O algoritmo pode sofrer algumas alterações para diminuir o poder de processamento necessário:

1. Em vez de calcular o histograma para cada vizinhança centrada no ponto (i, j) é possível dividir a imagem em k regiões não sobrepostas, calcular o histograma e função de transferência para cada região e aplicar esta função a todos os pontos daquela região. Este método produz, frequentemente, um efeito indesejado de tabuleiro de xadrez.
2. De cada vez que o algoritmo avança do ponto (i, j) para o ponto $(i, j + 1)$ apenas uma coluna é acrescentada à nova vizinhança e uma coluna é removida da antiga vizinhança. É possível alterar o histograma já calculado de modo a corresponder à nova vizinhança, sem ter que o recalculá-lo inteiramente.

Este algoritmo é um bom candidato à paralelização com decomposição no domínio dos dados, devido à grande localidade exibida.

6 Convolução

Em processamento de sinal, um filtro linear é caracterizado pela sua resposta a um impulso. A resposta $g(x)$ de um filtro linear $h(x)$ a um sinal contínuo unidimensional $f(x)$ é dada pelo integral da convolução :

$$g(x) = \int f(t)h(x - t)dt$$

As imagens digitais são bidimensionais e discretas, logo a convolução é dada por um so-

matório duplo

$$g(i,j) = \sum_{k=i-w}^{i+w} \sum_{l=j-v}^{j+v} f(k,l)h(i-k,j-l)$$

onde $\pm w, \pm v$ são os limites do filtro em cada dimensão.

Um filtro é aproximado por uma matriz que é aplicada a cada ponto da imagem.

Filtro 3x3

$$\begin{bmatrix} h_{1,1} & h_{1,2} & h_{1,3} \\ h_{2,1} & h_{2,2} & h_{2,3} \\ h_{3,1} & h_{3,2} & h_{3,3} \end{bmatrix}$$

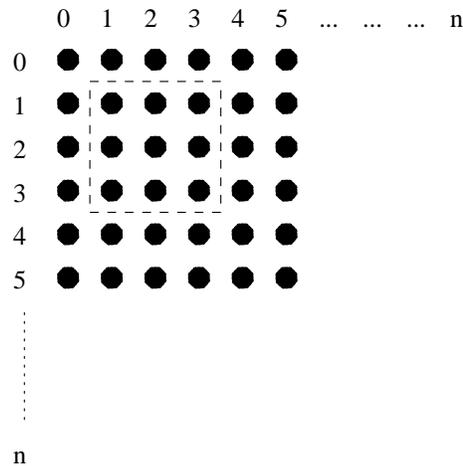


Figura 5: Imagem nxn

O resultado da aplicação deste filtro ao ponto (2,2) da imagem representada na figura 5 é dado por

$$\begin{aligned} r_{2,2} = & h_{1,1} * i_{(1,1)} + h_{1,2} * i_{(1,2)} + h_{1,3} * i_{(1,3)} + h_{2,1} * i_{(2,1)} + h_{2,2} * i_{(2,2)} + \\ & h_{2,3} * i_{(2,3)} + h_{3,1} * i_{(3,1)} + h_{3,2} * i_{(3,2)} + h_{3,3} * i_{(3,3)} \end{aligned}$$

Para obter a imagem final o filtro deve ser aplicado a todos os pontos da imagem (excepto aos da fronteira, pois para estes não existe vizinhança). Quanto maior for a dimensão do filtro, maior o tempo de processamento e mais pontos de fronteira são perdidos.

A convolução permite filtrar imagens, retirando-lhes algumas frequências espaciais, dependendo do filtro que é aplicado.

Por frequência espacial entende-se o modo como as alterações nos valores dos pixels estão distribuídas ao longo da imagem. Zonas que apresentam uma cor constante têm baixas frequências (não há variações), zonas com grandes mudanças nos tons dos pixels têm grandes frequências. Portanto uma zona da imagem que represente um objecto com uma só cor e sem ruído terá frequência zero, enquanto as zonas com ruído ou mudanças bruscas de tons (como os contornos dos objectos) têm altas frequências.

Um filtro passa-baixo caracteriza-se por remover as altas frequências, logo remove o ruído e suaviza os contornos. A suavização dos contornos é, normalmente, um efeito prejudicial, pois dificulta a detecção dos mesmos em passos posteriores da aplicação. Estes filtros

são normalmente designados por *smoothing filters* devido precisamente a este efeito de suavização da imagem.

Um filtro passa-alto remove as baixas frequências logo acentua o ruído e os contornos, removendo informação sobre as regiões constantes da imagem. Estes filtros, regra geral, são extremamente sensíveis ao ruído. É prática comum, quando da detecção de contornos, usar, num primeiro passo, um filtro passa-baixo para remover o ruído, seguido de um filtro passa-alto para detectar os contornos.

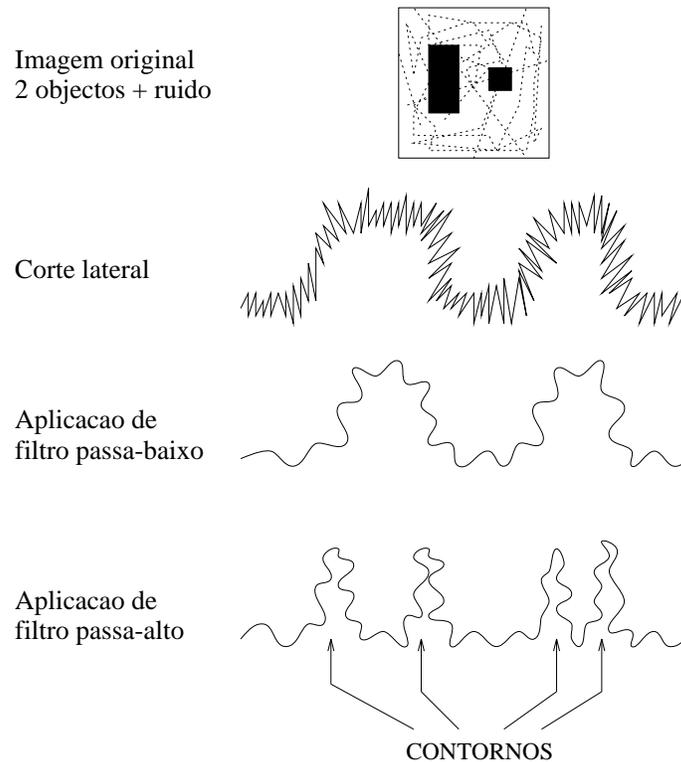


Figura 6: Detecção de contornos

Considere a figura 6. O filtro passa-baixo remove o ruído e suaviza os contornos. O filtro passa-alto aplicado a seguir remove a informação sobre as zonas de tom constante da imagem, realçando aquelas onde se verificam variações, que correspondem precisamente aos contornos. Note que, como estes filtros são aproximações discretas ao filtro ideal contínuo, não são perfeitos. Este efeito nota-se essencialmente pela quantidade de ruído (altas frequências) que o filtro passa-baixo deixa passar. Note também que o efeito de suavização não é suficiente para que o filtro passa-alto deixe de detectar os contornos.

6.1 Filtros passa-baixo (*smoothing filters*)

Os filtros passa-baixo removem ruído e atenuam os contornos. Esta atenuação, ou suavização, dos contornos faz com que as imagens filtradas pareçam esbatidas, daí designarem-se como *smoothing filters*.

Um filtro passa-baixo extremamente simples é a média. Esta consiste em atribuir a um ponto o valor da média de todos os pontos na sua vizinhança $m * m$. Se o valor deste ponto é muito diferente dos seus vizinhos essa diferença é atenuada.

$$\text{Média } 3 \times 3 \begin{bmatrix} \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \end{bmatrix}$$

Este filtro exige 9 multiplicações e 8 adições para cada ponto, no entanto apresenta a característica de ser separável, obtendo-se o mesmo efeito com 2 convoluções:

$$h_x = \left\{ \begin{array}{c} \frac{1}{3} \\ \frac{1}{3} \\ \frac{1}{3} \end{array} \right\} \quad h_y = \left\{ \frac{1}{3} \quad \frac{1}{3} \quad \frac{1}{3} \right\}$$

com apenas 6 multiplicações e 5 adições.

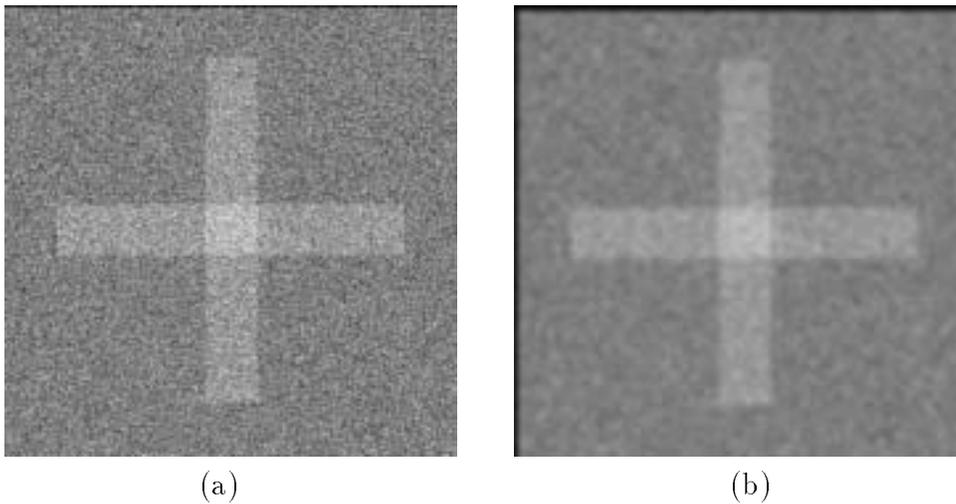


Figura 7: Média 5x5. a) Imagem original. b) Imagem filtrada

O equivalente digital do filtro Gaussiano é também muito usado. Uma superfície Gaussiana é dada por

$$\mathcal{G}_{(x,y)} = \frac{1}{\sqrt{2\pi\sigma}} \exp \left[-\frac{x^2 + y^2}{2\sigma^2} \right]$$

O valor de σ controla o grau de *smoothing* e a dimensão do filtro. Com $\sigma = 0.3$ obtem-se

$$\begin{bmatrix} 0.25 & 0.43 & 0.25 \\ 0.43 & 0.73 & 0.43 \\ 0.25 & 0.43 & 0.25 \end{bmatrix}$$

Este filtro também é separável, tendo-se

$$\mathcal{G}_{(x,y)} = \mathcal{G}_x * \mathcal{G}_y * \sqrt{2\pi\sigma}$$

6.2 Filtros passa-alto (*edge detection*)

Estes filtros removem as baixas frequências, realçando o ruído e os contornos dos objectos. São normalmente usados para detectar ou realçar contornos de objectos.

Os contornos correspondem a variações nos tons dos pixels, logo é intuitivo usar gradientes para medir estas variações. O gradiente ∇f de uma região f , é um vector com a magnitude e direcção da maior variação em f . O gradiente deve ser calculado em duas direcções perpendiculares.

As componentes x e y do gradiente podem ser obtidas com 2 convoluções:

$$g_x = h_x * f$$

$$g_y = h_y * f$$

A magnitude do gradiente é

$$G = \sqrt{g_x^2 + g_y^2}$$

e a direcção

$$\theta = \arctan\left(\frac{g_y}{g_x}\right)$$

Existem várias aproximações digitais ao gradiente

$$g_y = \left\{ \begin{array}{c} -1 \\ 0 \\ 1 \end{array} \right\} \quad g_x = \left\{ \begin{array}{cc} -1 & 1 \end{array} \right\}$$

Os vectores apresentados têm o inconveniente de o gradiente ficar centrado num ponto inexistente da imagem. Daí adoptar-se normalmente:

Geral	$g_y = \left\{ \begin{array}{c} -1 \\ 0 \\ 1 \end{array} \right\}$	$g_x = \left\{ \begin{array}{ccc} -1 & 0 & 1 \end{array} \right\}$
Sobel	$g_y = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}$	$g_x = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}$
Prewitt	$g_y = \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$	$g_x = \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix}$
Roberts	$g_y = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$	$g_x = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$

Os filtros de Roberts têm também a desvantagem de o gradiente ficar centrado num ponto inexistente.

Uma imagem $f(x,y)$ que contem apenas os contornos mais fortes de uma outra imagem original pode ser obtida com

$$f(x,y) = \begin{cases} 255 & \Leftarrow G(x,y) \leq T \\ 0 & \Leftarrow G(x,y) > T \end{cases}$$

onde $G(x,y)$ é a magnitude do gradiente centrado em (x,y) e T é um limite escolhido a partir do qual o contorno é considerado forte.



Figura 8: Gradiente: Sobel. a) Imagem original. b) Imagem filtrada

Aproximação digital ao Laplaciano

O Laplaciano é a 2ª derivada de uma função $f(x,y)$. Como 2ª derivada não dá informação sobre a direcção dos contornos, mas indica se um determinado ponto está no lado claro ou no lado escuro de um contorno, podendo ser usado para determinar se um determinado ponto pertence ou não a um objecto (figura 9). Tem ainda a vantagem de ser calculável numa única passagem.

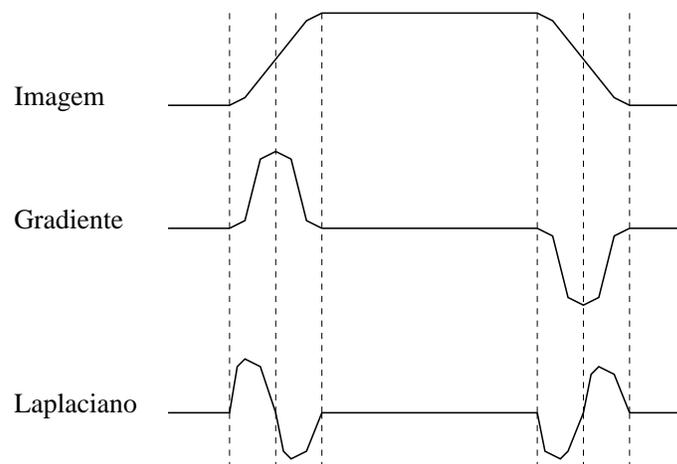


Figura 9: Cálculo do Laplaciano

Conclusões sobre o Laplaciano:

- Em zonas planas é nulo

- Cruza o zero, mudando de sinal, nos pontos em que o contorno é mais forte
- É positivo nas zonas do contorno que estão na parte mais clara da imagem
- É negativo nas zonas do contorno que estão na parte mais escura da imagem.

O Laplaciano tem o inconveniente de ser extremamente sensível ao ruído.

As aproximações digitais ao Laplaciano são

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \text{ para uma vizinhança } 4$$

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \text{ para uma vizinhança } 8$$

É frequentemente usada uma combinação do Laplaciano com o Gaussiano para controlar o tamanho dos objectos cujos contornos se pretende detectar. Os contornos dos objectos mais pequenos são suavizados pelo Gaussiano e não são detectados pelo Laplaciano. A combinação dos dois é dada por

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

Quanto mais pequeno for σ mais sensível é o filtro a pequenos objectos e a ruído.

O Laplaciano, tal como apresentado, é usado para detectar contornos. Esta técnica é também usada frequentemente para realçar contornos (*edge enhancement*). A ideia é somar o resultado do Laplaciano à imagem original. O filtro digital usado é

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix} \text{ para uma vizinhança } 4$$

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix} \text{ para uma vizinhança } 8$$

6.3 Filtros não-lineares

Os filtros discutidos dizem-se lineares porque é aplicável o teorema da convolução. Existem alguns filtros passa-baixo não-lineares que realizam um bom trabalho na remoção de ruído.

A moda consiste em escolher para o pixel central o valor que ocorre mais vezes na sua vizinhança $m * m$.

Imaginemos uma janela 3x3 centrada num ponto

$$\begin{vmatrix} 10 & 10 & 11 \\ 10 & 12 & 11 \\ 10 & 12 & 12 \end{vmatrix}$$

Aplicando a moda o ponto central toma o valor 10 pois este é o que ocorre mais vezes naquela vizinhança.

A mediana consiste em ordenar, por ordem crescente, todos os pontos da vizinhança $m * m$ do ponto central, e escolher para novo valor deste ponto o valor que fica no meio desta sequência ordenada. No exemplo anterior teríamos

Sequência ordenada {10, 10, 10, 10, 11, 11, 12, 12, 12}

logo o resultado da mediana é 11.

Este método elimina eficazmente o ruído, quando este consiste em picos fortes e isolados, e não suaviza os contornos, sendo portanto mais eficaz do que a média (figura 10). Tem o inconveniente de exigir a ordenação dos valores dos tons exigindo maior poder de computação. Isto agrava-se com o aumento do tamanho da vizinhança.

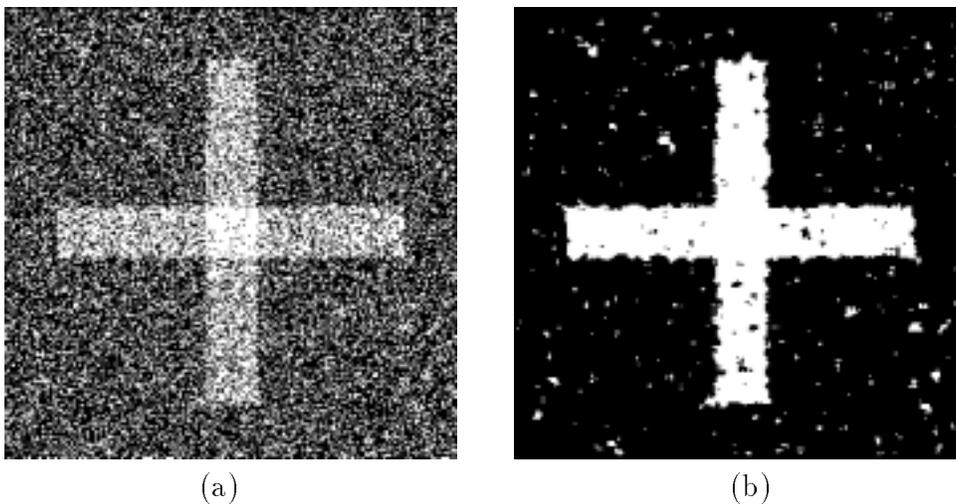


Figura 10: Filtragem com mediana. a) Imagem binarizada. b) Imagem filtrada

6.4 Outros métodos

- Um método vulgarmente utilizado para eliminar ruído aleatório numa sequência de imagens é utilizar, não uma imagem, mas a média das últimas k imagens. Como o ruído é aleatório este ficará diluído pela média de várias imagens iguais no seu conteúdo mas com ruído em pontos diferentes. Este método exige que se disponha de uma sequência de imagens da mesma cena, não serve para processamento em tempo real (em que nenhuma *frame* pode ser despedaçada) e não tolera variações bruscas na imagem.
- Com a detecção de contornos pretende-se identificar todos os pontos que pertencem ao contorno de um determinado objecto. Na prática, os contornos detectados quase

nunca são contínuos pois o ruído e a iluminação não-uniforme das cenas causam irregularidades no brilho da imagem. Os algoritmos de detecção de contornos são, por isso, normalmente seguidos de técnicas de ligação dos contornos. Um exemplo é a transformada de Hough (a estudar mais à frente) que permite detectar se vários pontos formam figuras geométricas como rectas, círculos, elipses, etc. Uma outra possibilidade é fazer uma análise local das características dos pixels identificados como pertencendo a contornos. Todos os pontos que numa determinada vizinhança apresentem uma magnitude e direcção do gradiente similares são unidos e considerados como fazendo parte do mesmo contorno. Para verificar a similaridade da magnitude e direcção dos gradientes utilizam-se as expressões

$$\left|G_{(x,y)} - G_{(x',y')}\right| \leq T$$

$$\left|\alpha_{(x,y)} - \alpha_{(x',y')}\right| \leq A$$

onde A e T são limites de tolerância previamente escolhidos.

7 Segmentação

A segmentação é o processo de divisão da imagem nas suas partes constituintes (objectos). Este é o primeiro passo no processo de encontrar, reconhecer ou contar objectos numa cena. Esta fase não inclui a detecção ou reconhecimento de formas particulares mas apenas a divisão da imagem em regiões distintas. A segmentação é um dos elementos mais importantes na análise automática de de imagens porque consiste na extracção de objectos, ou outras entidades, para processamento posterior, tal como reconhecimento ou descrição.

Os algoritmos de segmentação baseiam-se geralmente em uma de duas propriedades básicas das imagens: descontinuidade e similaridade. Na primeira categoria a imagem é segmentada em pontos onde o nível de cinzento sofre alterações bruscas. As principais áreas de aplicação são a detecção de pontos isolados, linhas e contornos. As principais abordagens na segunda categoria (similaridade) são a binarização (*thresholding*), *region growing* e *region splitting & merging*. O nosso estudo vai basear-se essencialmente nesta categoria. A similaridade entre dois pontos (ou duas regiões) numa imagem é baseada em propriedades como o tom ou a textura.

A segmentação pode ser aplicada a uma única imagem ou a uma sequência de imagens. Neste último caso o movimento pode ser utilizado como um dado que permite aumentar o desempenho dos algoritmos.

7.1 Binarização

A binarização de uma imagem (*thresholding*) é uma das técnicas mais importantes em segmentação. Basicamente pretende-se converter uma imagem com vários tons numa outra apenas com preto e branco. A imagem binarizada contém menos informação (sendo portanto mais fácil de processar) e contém objectos já segmentados.

Imagine uma imagem com o histograma apresentado na figura 11.

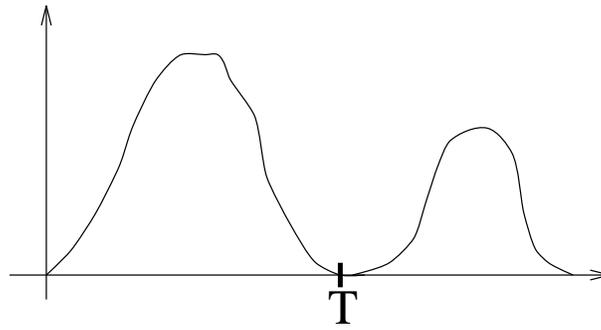


Figura 11: Histograma bimodal

Este corresponde a uma imagem composta por objectos claros num fundo escuro. Um método óbvio para extrair os objectos consiste em atribuir a todos os pixels com um valor superior a um limite T o valor branco e a todos os pixels com valor inferior a T o valor preto.

$$b_{(x,y)} = \begin{cases} 255 & \Leftarrow f_{(x,y)} > T \\ 0 & \Leftarrow f_{(x,y)} \leq T \end{cases}$$

Esta técnica pode ser generalizada e englobar várias classes de objectos classificados de acordo com o seu brilho. O histograma da figura 12 corresponde a uma imagem com fundo escuro, alguns objectos mais brilhantes e objectos extremamente brilhantes.

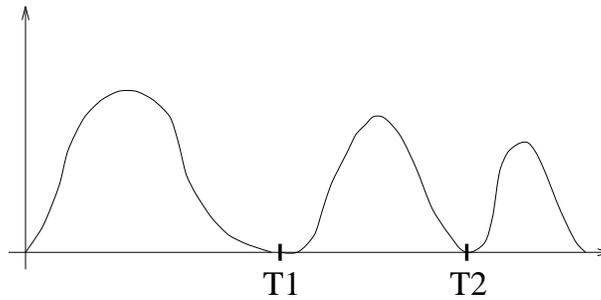


Figura 12: Histograma n-modal

Podemos escrever uma função de *thresholding* multinível que considere as várias classes de objectos.

$$b_{(x,y)} = \begin{cases} 255 & \Leftarrow f_{(x,y)} \geq T_2 \\ 128 & \Leftarrow f_{(x,y)} \geq T_1 \\ 0 & \Leftarrow f_{(x,y)} < T_1 \end{cases}$$

Claro que esta função pode ser estendida para considerar um número arbitrário de níveis. Este *threshold* multinível é geralmente menos fiável do que a binarização devido à dificuldade em determinar os vários T_i que efectivamente isolam regiões de interesse. Este género de problemas, quando resolvidos com *thresholding*, são normalmente processados usando técnicas de binarização adaptativa, que estudaremos posteriormente.

A selecção do valor de T que proporciona a melhor segmentação é o principal problema da binarização. O erro neste processo é dado pela adição do número de pontos pertencentes

ao fundo que foram classificados como pertencentes ao objecto com o número de pontos pertencentes ao objecto que foram classificados como pertencendo ao fundo. O processo de escolha do valor de T passa normalmente por várias tentativas. No entanto, quando este processo não pode ser interactivo, tem que ser encontrado um método automático. O histograma de imagens reais raramente apresenta 2 picos distintos e disjuntos associados ao fundo e aos objectos. Os valores possíveis para estas entidades sobrepõem-se devido a:

- Iluminação irregular ao longo da cena (sombra, por exemplo)
- Não uniformidade nos tons do fundo e objectos
- Presença de vários objectos diferentes

O valor de T pode ser calculado em função do brilho \bar{v} e contraste σ da imagem

$$T = k\sigma + \bar{v}$$

Uma possibilidade é o estudo analítico do histograma para identificar mínimos ou pontos onde a sua característica muda significativamente. Para um estudo mais aprofundado consultar [Gonzalez, p. 354...368].

7.2 Binarização adaptativa

A binarização global não considera variações locais ao longo da imagem. A binarização adaptativa é uma alternativa que calcula o valor de T para cada ponto individualmente, de acordo com a média e desvio padrão da sua vizinhança $m * m$.

$$t_{(i,j)} = k\sigma_{(i,j)} + \overline{v_{(i,j)}}$$

$$b_{(x,y)} = \begin{cases} 225 & \Leftarrow f_{(x,y)} \geq t_{(i,j)} \\ 0 & \Leftarrow f_{(x,y)} < t_{(i,j)} \end{cases}$$

Esta técnica é, obviamente, mais pesada que a binarização global.

7.3 Operadores morfológicos

Após a binarização os objectos encontrados podem necessitar de processamento adicional para retirar, ou reduzir, irregularidades nos contornos, eliminar buracos no interior e remover ruído. Isto é conseguido por operações conhecidas como erosão (*thinning*) e dilatação (*growing*). Estes são 2 operadores conhecidos como morfológicos porque alteram a forma das regiões.

Para erodir uma região todos os pontos pertencentes ao seu contorno são alterados para passarem a pertencer ao fundo. Remove-se assim uma camada de pontos ao objecto. Para dilatar uma região procede-se ao contrário, isto é, acrescenta-se uma camada de pontos ao contorno do objecto.

A erosão tem a vantagem de remover saliências e remover pequenas regiões isoladas que poderão ser ruído. Tem no entanto o inconveniente de aumentar buracos no interior dos objectos, reentrâncias nos seus contornos e de poder separar suas partes do mesmo objecto que se encontrem unidas por linhas finas.

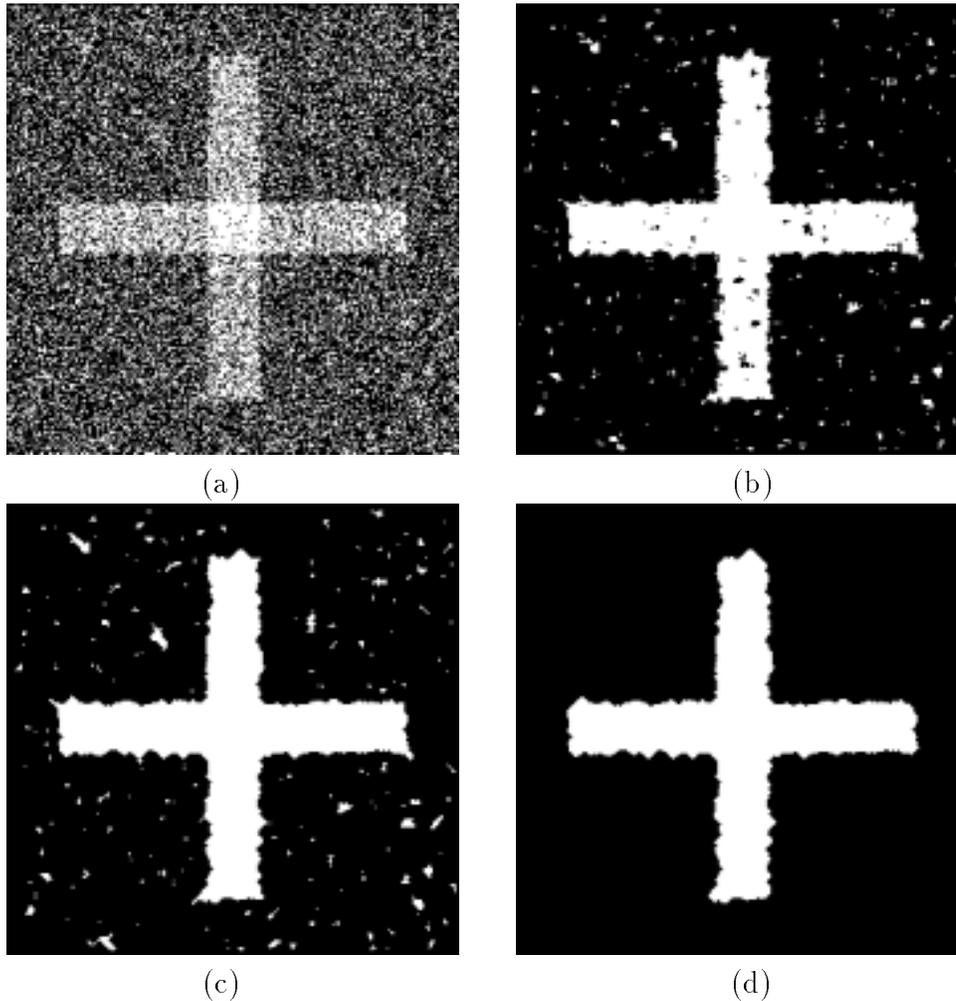


Figura 13: Erosão. a) Imagem binarizada. b) Imagem filtrada (mediana) c) Imagem dilatada d) Imagem erodida

A dilatação tem a vantagem de eliminar reentrâncias nos contornos e buracos no interior dos objectos. Tem as desvantagens de realçar saliências, aumentar pequenas regiões que poderão ser ruído e ligar dois objectos distintos que estejam muito próximos.

Para reduzir estas desvantagens usa-se normalmente uma qualquer combinação destas 2 técnicas. Se o objectivo é erodir um objecto é possível dilatá-lo primeiro para eliminar pequenas reentrâncias ou buracos e, seguidamente, erodi-lo duas vezes seguidas (figura 13).

Existem vários algoritmos para estas técnicas. Uma possibilidade, para a erosão, é remover qualquer ponto que tenha como vizinho um ponto do fundo. Da mesma forma, na dilatação, pode-se acrescentar ao objecto qualquer ponto do fundo que tenha como vizinho um ponto do objecto.

A erosão parece dar melhores resultados eliminando apenas os pontos com menos de 6

vizinhos pertencentes ao objecto. Em [Gonzalez p.398] é proposto um outro algoritmo para a erosão com o qual se pretende determinar o esqueleto de um objecto. O algoritmo a usar em cada situação depende do fim a atingir com as operações morfológicas e deve ser escolhido com base em experiências.

7.4 Textura

Todas as técnicas até agora estudadas usavam como descritor de uma região o seu brilho ou contraste. Uma outra abordagem para a descrição de uma região é quantificar a sua textura. Embora não exista nenhuma definição formal para a textura, esta é uma importante fonte de informação para a segmentação. Intuitivamente podemos ver a textura como uma medida de propriedades como suavidade, aspereza e regularidade de uma superfície.

Uma textura pode ser classificada como sendo de granularidade fina (mármore) ou granularidade grossa (parede de tijolos) e como sendo regular (rede de pesca) ou aleatória (estrada de alcatrão).

Com o estudo da textura levanta-se o problema da escolha da resolução a considerar. A imagem de uma estrada, por exemplo, apresenta uma textura com variações quase ao nível do pixel enquanto uma parede de tijolos tem repetições de elementos com várias centenas de pontos de tamanho. Da mesma forma, se o objectivo é distinguir 2 paredes de tijolos feitos de materiais diferentes mas com o mesmo tamanho, é preciso considerar a textura de cada tijolo individualmente, e não da parede como um todo.

As 3 principais abordagens para descrever texturas são:

- modelo estrutural
- modelo estatístico
- modelo espectral

O modelo estrutural lida com elementos básicos da textura (*texel*). Idealmente estes elementos repetir-se-ão ao longo da região. O modelo estatístico quantifica características como a suavidade, aspereza, granularidade, etc. O modelo espectral lida com as regiões no domínio das frequências espaciais.

Modelo estrutural

Este modelo lida com o conceito de uma primitiva da textura (*texel*). Esta primitiva é um elemento fixo que se repete numa área da imagem. É o caso de uma imagem com grãos de café ou moedas, uma parede de tijolos ou uma rede de pesca. Para caracterizar o elemento básico destas texturas pode-se usar uma função de densidade ou uma imagem com o elemento que se repete. A função de densidade é particularmente útil quando o *texel* pode sofrer rotações ou ampliações /reduções.

O *texel* pode ser difícil de identificar quando a textura a analisar é muito complexa. Isto ocorre, por exemplo, em problemas de cristalografia onde se pretende distinguir e/ou identificar vários cristais.

Modelo estatístico

Quando não é possível definir um elemento básico que identifique a textura, é vulgar recorrer-se ao modelo estatístico para quantificar algumas propriedades daquela superfície. Isto acontece quando a resolução da textura é muito alta e/ou tem um arranjo pseudo-aleatório. Com o modelo estatístico tenta-se quantificar propriedades como a suavidade, aspereza e regularidade da superfície.

Uma das abordagens mais simples consiste em usar os momentos do histograma de uma região para caracterizar essa região. Os dois momentos mais comuns são a média (brilho) e o desvio-padrão (contraste). A variância (σ^2) é uma medida directa da aspereza da região e pode ser usada para construir uma medida de suavidade relativa

$$R = 1 - \frac{1}{1 + \sigma^2}$$

que é 0 para regiões de intensidade constante e aproxima-se de 1 em regiões muito ásperas.

As limitações destas medidas é que não contêm informação sobre a posição relativa dos pixels. Para analisar esta informação devem-se considerar não apenas as intensidades, mas também a posição dos pontos uns relativamente aos outros. Para tal constrói-se a matriz de coocorrência A .

Para construir esta matriz define-se um operador $P(d, \theta)$ que estabelece um deslocamento d na direcção θ .

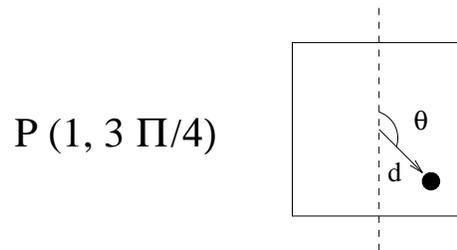


Figura 14: Operador $P(d, \theta)$

A é uma matriz $k * k$ onde k é o número de tons possíveis na imagem. Cada elemento $a_{(i,j)}$ é o número de vezes que um ponto com o tom i aparece na posição especificada por $P(d, \theta)$ relativamente a um ponto com o tom j .

Considere a seguinte imagem 6x5 com 2 tons 0, 1 e o operador $P(1, 3\pi/4)$. A matriz de coocorrência correspondente é a apresentada

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

$$A = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}$$

O elemento a_{00} (canto superior esquerdo), por exemplo, representa o número de vezes que um elemento com o tom 0 aparece à distância 1 e à direita e abaixo de outro ponto com o tom 0. É importante notar que o tamanho da matriz de coocorrência depende do número de tons. Para manter este tamanho dentro de limites manuseáveis é usual diminuir o número de tons possíveis em cada imagem.

A matriz de coocorrência A não caracteriza univocamente as texturas, mas a comparação de algumas medidas extraídas destas matrizes ajuda a averiguar a similaridade de 2 texturas. Por exemplo, a presença de valores elevados na diagonal principal de A indica a presença de faixas na imagem com direcção igual a θ e espessura inferior a d .

A matriz de coocorrência normalizada C pode ser obtida dividindo cada elemento de A pelo número de pares de pontos na imagem que satisfazem $P(d, \theta)$

$$C_{(i,j)} = \frac{a_{i,j}}{n}$$

$$n = \sum_{i,j} a_{i,j}$$

São várias as medidas que podem ser extraídas da matriz C

Probabilidade máxima $\max(c_{i,j})$

Inércia $\sum_i \sum_j (i-j)^k c_{i,j}$

Homogeneidade $\sum_i \sum_j \frac{c_{i,j}}{(i-j)^k}, i \neq j$

Entropia $\sum_i \sum_j c_{i,j} \log(c_{i,j})$

Energia $\sum_i \sum_j c_{i,j}^2$

A ideia é caracterizar a textura usando estas medidas. Um aspecto interessante é que estas não têm nenhuma correspondência psicológica, isto é, o observador humano não reconhece as variações na textura que estas quantidades medem. Os valores típicos destas medidas para um conjunto de texturas podem ser fornecidos a um sistema, que posteriormente identificará texturas de outras imagens por comparação dos seus descritores com aqueles armazenados em memória.

Modelo espectral

Este modelo baseia-se no uso de técnicas no domínio das frequências espaciais. Se uma textura é periódica no espaço então o espectro apresenta picos nas altas frequências. A posição destes picos pode ser identificada e usada como um descritor de texturas. Este tipo de texturas periódicas é, normalmente, difícil de detectar usando técnicas espaciais devido à localidade destes métodos. É frequente, para isolar estes picos, efectuar partições radiais ou angulares do espectro. Com estas partições pretende-se isolar as frequências que descrevem a textura. As partições radiais são utilizadas quando não existe uma direcção associada à textura (figura 15).

O modelo espectral tem 2 características úteis para a descrição de texturas:

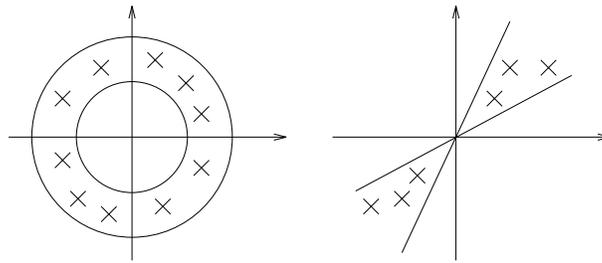


Figura 15: Partições do espectro

- Os picos mais proeminentes no espectro representam a direcção principal dos padrões das texturas e a sua localização no plano das frequências indica o período espacial destes padrões.
- A remoção das componentes periódicas (por filtragem) resulta numa imagem sem elementos periódicos que pode depois ser descrita usando técnicas estatísticas.

7.5 Segmentação baseada em regiões

O objectivo da segmentação é subdividir a imagem em regiões. Foi sugerido que uma imagem pode ser dividida em várias regiões aplicando um *thresholding* multinível. Esta técnica pode resultar num grande número de regiões irregulares, que não correspondem a nenhuma característica importante da imagem. Nesta secção vamos procurar regiões baseados nas propriedades locais dos pixels. A ideia é que todos os pontos de uma região devem ter uma (ou mais) propriedade(s) comum(s), seja ela brilho, textura, gradiente, etc.

Seja R a imagem. A segmentação pode ser vista como o processo de subdividir a imagem em subregiões $R_1, R_2 \dots R_n$ tais que

1. $\bigcup_{i=1}^n R_i = R$
2. R_i é uma região conectada
3. $R_i \cap R_j = \emptyset$
4. $P(R_i) = \text{TRUE}$
5. $P(R_i) \cup P(R_j) = \text{FALSE}$ se R_i e R_j são conectadas.

A condição 1 indica que a segmentação deve ser completa, isto é, qualquer pixel da imagem original pertence a uma região. Juntamente com a condição 3 garantimos que cada pixel pertence a uma e uma só região e que todos os pixels pertencem a uma região. A condição 2 garante que todos os pontos numa região estão conectados. A condição 4 assegura as propriedades que devem ser satisfeitas pelos pontos para pertencerem à mesma região. Um exemplo é dizer que $P(R_i) = \text{TRUE}$ se todos os pontos naquela região têm a mesma intensidade. A condição 5 indica que, se 2 regiões são contíguas, então têm que ser diferentes relativamente à propriedade P .

Region growing by pixel aggregation

Esta técnica agrupa pixels ou subregiões em regiões maiores. Começando por um conjunto de pontos semente *seed* a região cresce, agregando pontos vizinhos que tenham propriedades similares (brilho, textura, gradiente, etc.). Este processo repete-se até que não seja possível agregar mais pontos à região.

As propriedades a considerar para a condição de agregação podem ser várias. Um exemplo é a diferença absoluta entre o valor do pixel a agregar e o valor do ponto semente não ultrapassar um limite T . Outras características, como o gradiente, podem também ser consideradas.

$$\left| P_{semente} - P_{(x,y)} \right| \leq T$$

Os dois problemas mais imediatos com esta técnica são a selecção de pontos semente adequados e a escolha das propriedades a considerar para a condição de agregação.

A escolha dos pontos iniciais é dependente da natureza do problema. Se se sabe *a priori* que o objecto é claro e o fundo escuro, então podem ser escolhidos como sementes o ponto mais escuro e o ponto mais claro. Se não existe qualquer ideia sobre como distinguir claramente os objectos é possível calcular, para todos os pontos, as mesmas propriedades que estão associadas à condição de agregação. Se os resultados deste cálculo apresentarem blocos de valores, então os pontos no centro destes blocos podem ser usados como pontos semente. Note que não é desperdiçado tempo de computação pois no processo de agregação as propriedades não precisam de ser recalculadas, já estão disponíveis.

A condição de agregação pode ser estática ou evoluir com a região. É possível, por exemplo, utilizar sempre as propriedades dos últimos pontos agregados em vez das propriedades do ponto-semente. Outro factor importante é a condição de paragem. Basicamente o processo de agregação termina quando mais nenhum ponto satisfaz a condição de agregação. No entanto, se existir algum conhecimento sobre os resultados esperados, o critério de paragem pode ser enriquecido com informação sobre o tamanho da região, forma da região, brilho da região, etc.

Region splitting & merging

O método anterior começa com um ponto e constroi a região a partir daí. Uma alternativa é começar com toda a imagem e subdividi-la em subregiões regulares de acordo com um conjunto de regras. Estas subregiões podem ser ainda subdivididas (se não são homogéneas relativamente a um conjunto de propriedades) ou reunidas de novo se são similares.

O método mais vulgar consiste em dividir a imagem em 4 quadrantes. Para cada um destes quadrantes é estudada a sua homogeneidade relativamente à condição P . Os quadrantes não homogéneos são subdivididos de novo, e assim sucessivamente até que não sejam necessárias mais subdivisões. Nesta fase unem-se regiões contíguas que sejam similares relativamente à mesma condição P (figura 16).

7.6 Movimento

O movimento é frequentemente utilizado pelos humanos e outros animais para extrair objectos de interesse de um fundo com pormenores irrelevantes. Numa sequência de ima-

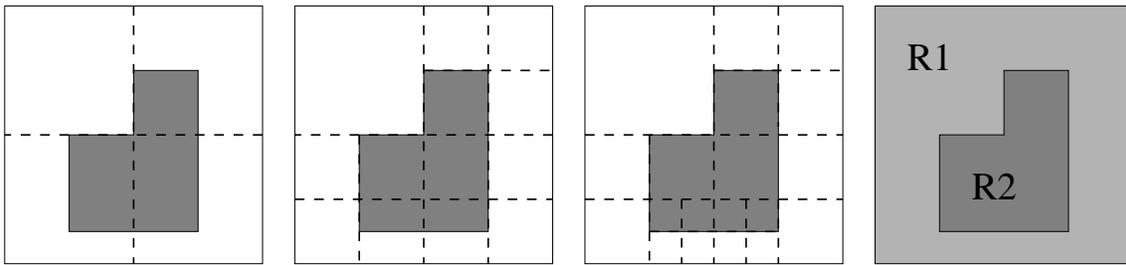


Figura 16: Region splitting & merging

gens o movimento deriva de um deslocamento relativo entre os sensores (câmera) e a cena visualizada.

Uma das aproximações mais simples para detectar variações entre duas imagens adquiridas em instantes de tempo diferentes, t_i e t_j , é subtraí-las ponto-a-ponto, obtendo a imagem diferença. Supondo que temos uma imagem referência que contem apenas elementos estacionários, podemos detectar objectos em movimento subtraindo uma imagem posterior a esta imagem referência. Na imagem diferença resultante todos os objectos estacionários foram anulados, ficando apenas as regiões que apresentam características diferentes, supostamente devido a movimento. A imagem diferença pode ser obtida calculando

$$d_{(x,y)} = \begin{cases} 1 & \Leftarrow |r_{(x,y)} - f_{(x,y)}| \geq T \\ 0 & \Leftarrow |r_{(x,y)} - f_{(x,y)}| < T \end{cases}$$

onde T é um limite de tolerância previamente escolhido.

Esta técnica pode ser aplicada usando uma imagem referência ou a imagens sucessivas. A figura 17 ilustra o resultado da aplicação desta técnica a duas imagens sucessivas com um objecto em movimento. A imagem diferença apresenta 2 regiões: uma correspondente à zona para onde o objecto se deslocou, outra correspondente à zona que o objecto abandonou.

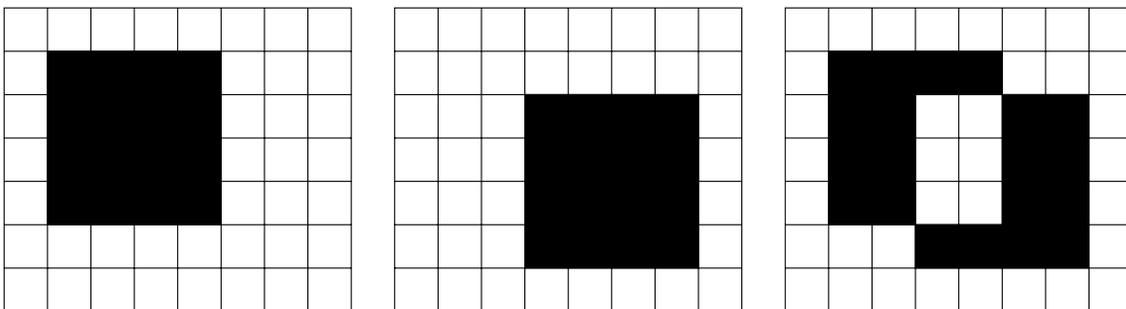


Figura 17: Objecto em movimento

O uso de uma imagem diferença levanta vários problemas. Primeiro a diferença entre 2 imagens nunca é exacta devido ao ruído. Assim a imagem diferença apresentará pontos diferentes de zero que não correspondem a movimento. Como, provavelmente, estes serão pontos isolados, ou pequenos blocos de pontos, podem ser posteriormente removidos por filtragem (mediana?) mas isto pode resultar na remoção de informação relativa a pequenos objectos ou objectos em movimento lento. A imagem diferença entre 2 imagens sucessivas contem, por objecto, 2 zonas de pontos, como já foi visto. Quando existem muitos objectos

em movimento é difícil agrupar estas zonas em pares que realmente correspondem ao mesmo objecto. Se um objecto se desloca muito lentamente, então cada um destes blocos pode ser extremamente fino (e por isso sensível ao processo de filtragem) ou o movimento pode não ser detectado entre alguns pares de imagens, levando ao desaparecimento virtual (mas temporário) daquele objecto.

O uso de uma imagem referência, que contem apenas objectos estacionários, levanta outros problemas. Não é fácil conseguir uma imagem sem movimento numa cena com muito tráfego (por exemplo uma rua muito concorrida), portanto a própria imagem de referência pode ser impossível de obter. Uma vez obtida, coloca-se o problema de decidir quanto tempo é que pode ser usada. Se pensarmos numa avenida da qual conseguimos uma imagem sem qualquer objecto em movimento (obtida de madrugada, por exemplo), será que a podemos usar durante todo o dia? É necessário pensar que as condições de iluminação vão mudar, vão existir veículos estacionados que não estão em movimento mas também não estão contidos na referência, etc. Daqui se conclui que a utilização de imagens consecutivas ou imagens de referência depende da natureza da própria aplicação.

Diferenças acumuladas

O uso de uma imagem diferença sofre de alguns inconvenientes como o facto de ser muito sensível ao ruído. Este problema é ultrapassado usando imagens de diferenças acumuladas. Cada imagem da sequência é subtraída à imagem referência e um contador por cada pixel é incrementado na imagem de diferenças acumuladas sempre que existe uma diferença significativa entre aquele pixel na imagem e o pixel correspondente na referência. Assim quando a k -ésima imagem da sequência está a ser processada, a imagem acumulada indica quantas vezes cada pixel foi diferente da imagem referência. Consegue-se assim ignorar variações esporádicas na imagem.

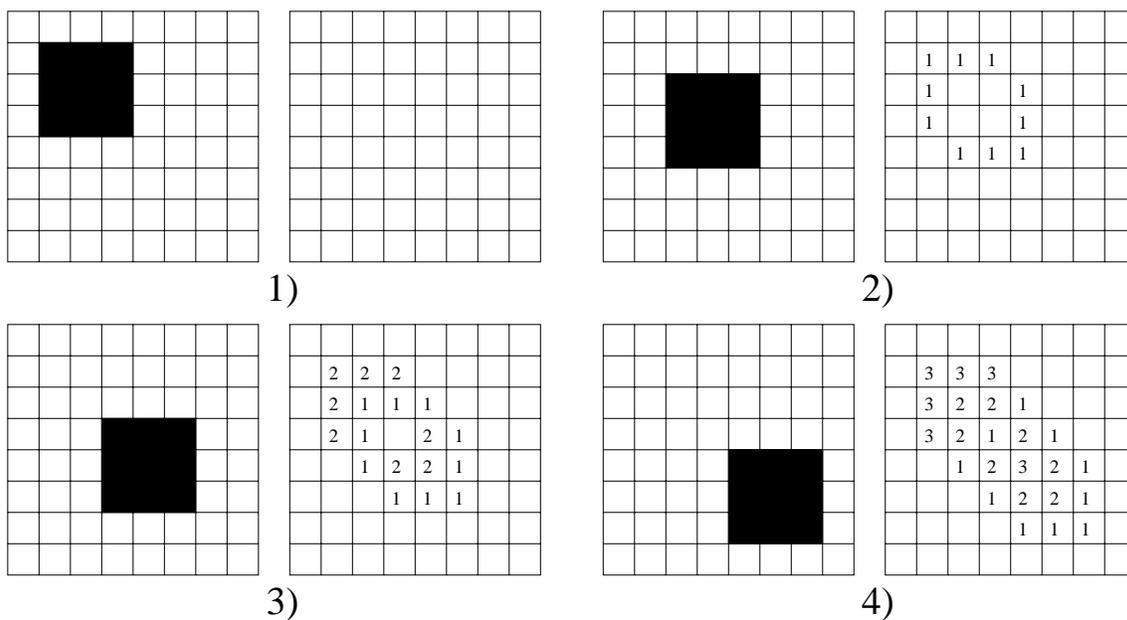


Figura 18: Diferenças acumuladas

Em certas aplicações é útil considerar 3 tipos de diferenças acumuladas. O método estudado até agora lida com o valor absoluto da diferença entre a referência e a imagem actual, sendo por isso designado por AADI (*Absolute Accumulative Differences Image*). Quando

os objectos têm um valor mais elevado que o fundo define-se PADI (*Positive ADI*) como sendo o método que apenas incrementa os contadores quando a diferença entre os pontos é positiva. NADI (*Negative ADI*) é o método oposto, os contadores só são incrementados quando a diferença entre os pontos é negativa. Quando o fundo é mais claro que os objectos estas definições invertem-se.

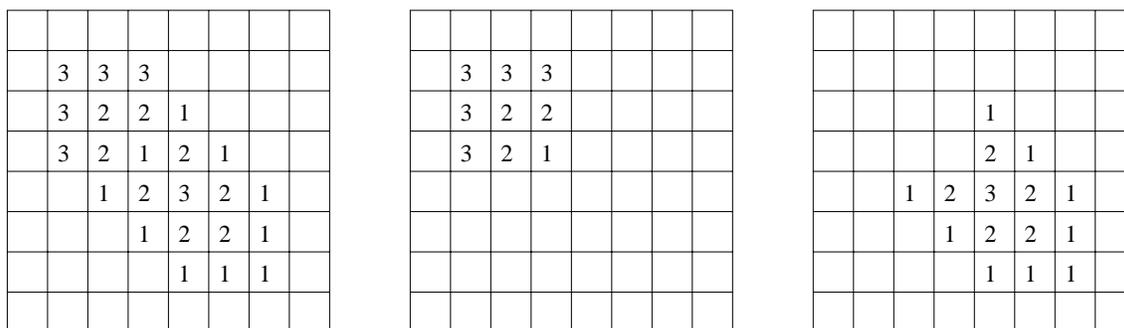


Figura 19: AADI, PADI e NADI

O PADI tem a característica de apenas crescer enquanto o objecto está parcial ou totalmente sobreposto com a sua posição na imagem referência. Quando o objecto se desloca completamente da sua posição inicial o PADI não regista mais variações. O NADI comporta-se exactamente ao contrário, apenas registando alterações nas zonas não ocupadas inicialmente pelo objecto (figura 19).

Imagem referência

A chave para o sucesso das técnicas descritas dependem da existência de uma imagem de referência com a qual possam ser feitas comparações. Na prática nem sempre é possível conseguir uma imagem de referência apenas com elementos estacionários e esta tem que ser construída a partir de uma sequência de imagens que contêm objectos em movimento. Isto é particularmente verdadeiro quando se processam cenas com muito movimento ou que precisam de actualizações frequentes.

Descreve-se a seguir uma aproximação simplificada para construir uma imagem de referência a partir da primeira imagem de uma sequência. Quando um objecto em movimento se desloca completamente para fora da sua posição nesta primeira imagem, o fundo correspondente a essa posição inicial pode ser obtido da imagem actual e substituído na imagem referência. Quando todos os objectos se tiverem deslocado da sua posição inicial a imagem de referência deve conter apenas elementos estacionários. O deslocamento dos objectos pode ser estabelecido monitorizando o crescimento do PADI. Esta é uma aproximação simplificada porque quando um objecto em movimento se desloca da sua posição inicial, outro objecto qualquer pode estar a entrar nessa zona dificultando a obtenção dessa região do fundo.

8 Detecção de formas

As técnicas discutidas nesta secção destinam-se a detectar numa imagem formas previamente conhecidas. Vamos distinguir duas técnicas diferentes: numa está disponível um

template da forma que se pretende encontrar, na outra dispõe-se de uma descrição matemática da forma a encontrar (caso de rectas, por exemplo).

8.1 *Template matching*

Este é um método de filtragem muito popular com o qual se pretende detectar um objecto em particular numa imagem. O filtro é uma sub-imagem que contém o objecto a detectar (ou parte dele) e é designado por *template*.

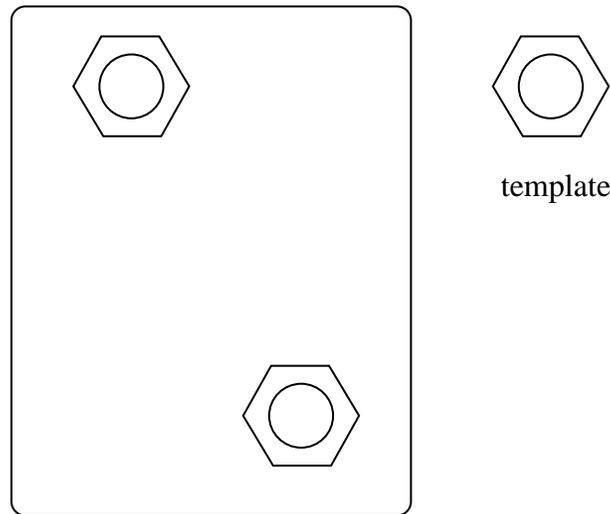


Figura 20: Template matching

O método consiste em subtrair o filtro a uma subregião da imagem centrando-o num determinado pixel. Para esta subtracção usa-se habitualmente a distância Euclidiana d dada por

$$d_{(i,j)}^2 = \sum_x \sum_y [I_{(x,y)} - t_{(x-i,y-j)}]^2$$

onde x e y são as dimensões do *template*. Esta operação é repetida centrando o *template* em todos os pontos da imagem. O resultado da subtracção é zero quando a região da imagem a que o filtro está a ser aplicado é exactamente igual ao filtro. Na prática o resultado nunca será exactamente 0 devido ao ruído, diferentes condições de iluminação e diferenças entre o objecto a detectar e o *template*. A abordagem a seguir consiste em procurar mínimos na matriz bidimensional resultante do cálculo da distância Euclidiana (designada por superfície de correlação). Se estes mínimos estiverem dentro de um limite de tolerância previamente estabelecido, então o objecto está encontrado.

Este método é computacionalmente muito pesado e extremamente sensível a rotações e diferenças de escala entre o objecto a detectar e o *template*.

Pirâmide de resolução

A resolução da imagem original pode não ser a mais indicada para procurar objectos ou contornos. Quando a resolução da imagem é demasiado grande a aplicação de técnicas como o *template matching* torna-se demasiado pesada. É mais eficiente localizar primeiro o objecto numa imagem com menor resolução e, posteriormente, refinar o resultado obtido

passando para a imagem de maior resolução. Este processo introduz a noção de pirâmide de resolução.

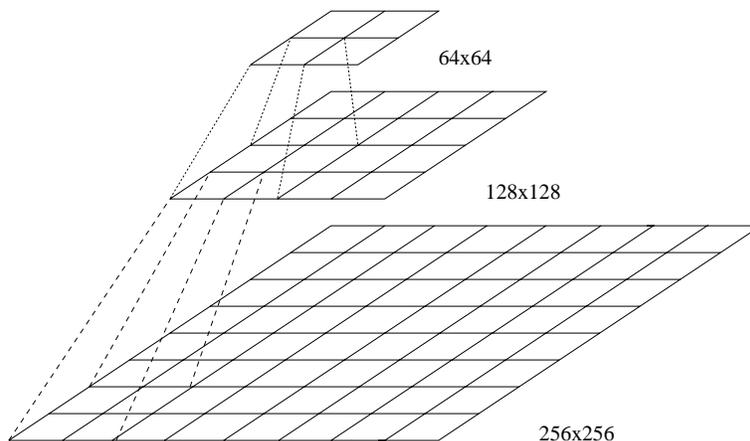


Figura 21: Pirâmide de resolução

Esta imagem pode ser obtida achando a média de cada 4 pixels e gerando, a partir daí, um novo ponto na imagem com menor resolução. Este processo pode ser repetido várias vezes dando origem a uma pirâmide com vários níveis. O algoritmo de detecção de formas pode ser aplicado à imagem de menor resolução e o valor obtido pode ser refinado progredindo na direcção da base da pirâmide.

8.2 A transformada de Hough

Esta transformada permite encontrar linhas numa imagem. As linhas podem estar em qualquer posição e com qualquer orientação. Inclusivamente podem ter descontinuidades devidas a ruído, ou estar parcialmente invisíveis devido à sobreposição de outros objectos. A ideia é procurar pontos colineares e, dependendo do seu número, decidir se pertencem ou não a uma linha.

A estratégia a seguir consiste em para cada ponto, não pertencente ao fundo, calcular *todas* as rectas que podem passar por ele. A equação da recta é $y = mx + b$ onde m é o declive e b a ordenada na origem. Se dispusermos de uma matriz bidimensional H cujos eixos sejam m e b , inicializada a 0, podemos para cada ponto (x, y) da imagem calcular os pares (m, b) de *todas* as rectas que passam por esse ponto, e incrementar na matriz H todas as células cujos índices sejam os pares calculados. Depois de todos os pontos processados cada célula da matriz H indica o número de pontos da imagem que pertencem à recta com os parâmetros (m, b) . Escolhendo apenas as células da matriz H com valores mais elevados temos determinadas as rectas mais fortes da imagem, cujos parâmetros (m, b) são os índices dessas células.

O problema com o processo descrito é que tanto m como b podem variar até ao infinito, não sendo possível conceber uma matriz que possa representar este espaço. Mas usando coordenadas polares temos a equação da recta

$$\rho = x \cos \theta + y \sin \theta$$

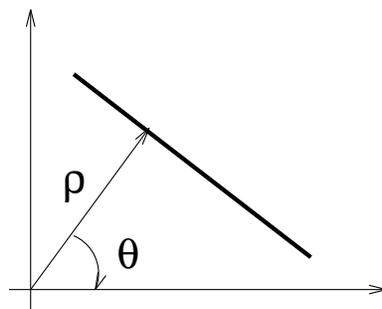


Figura 22: Coordenadas polares

Os valores entre os quais ρ e θ podem variar são limitados, logo a técnica atrás descrita pode ser implementada usando uma matriz bidimensional H cujos eixos são ρ e θ . Não é possível, por motivos óbvios, calcular todas as rectas que passam num ponto (x, y) , mas variando θ com um passo adequado é possível calcular um grande número de rectas que passam em cada ponto e, portanto, obter vários pares (ρ, θ) .

A matriz H tem o aspecto apresentado na figura 23. Cada linha corresponde a todas as rectas que passam num ponto presente na imagem. Os pontos com as coordenadas (ρ, θ) onde duas ou mais linhas se cruzam correspondem a rectas para as quais existem vários pontos colineares na imagem original. Quanto maior o número de linhas que se cruzam nas coordenadas (ρ, θ) mais pontos colineares existem pertencentes a essa recta.

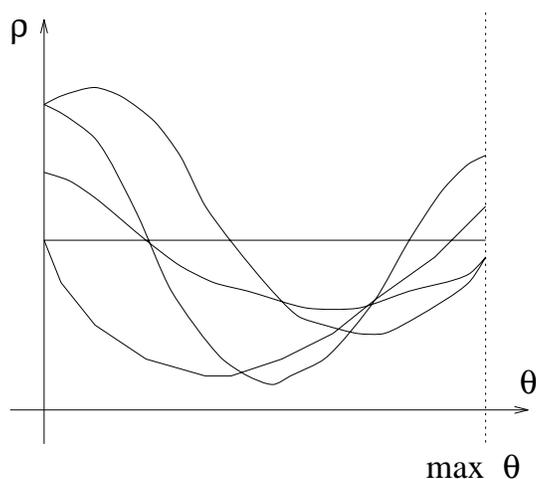


Figura 23: Transformada de Hough

Normalmente a transformada de Hough é aplicada a imagens que foram pré-processadas de modo a que na imagem só estejam presentes pontos que possam corresponder a características relevantes. Isto porque esta transformada é um algoritmo bastante pesado. O caso mais comum consiste em aplicar um algoritmo de detecção de contornos (ex. gradiente) e aplicar posteriormente a Transformada de Hough, quer para eliminar descontinuidades destes contornos, quer para reconhecer a forma dos mesmos.

Esta transformada pode ser aplicada a uma variedade de outras formas geométricas, como por exemplo círculos, com a equação

$$(x - a)^2 + (y - b)^2 = r^2$$

seguindo a abordagem descrita. A principal diferença é que agora temos 3 parâmetros em vez de dois, logo a matriz H deve ser tridimensional. Outras figuras possíveis são:

Parábola: $(y - y_r) = 4S_x(x - x_r)$

Elipse: $\frac{(y-y_r)^2}{S_y^2} + \frac{(x-x_r)^2}{S_x^2} = 1$

9 Descrição de formas

Depois de uma imagem ter sido segmentada em regiões com significado é usual representar e descrever estas regiões de uma forma adequada para futuro processamento. Este processamento consiste, normalmente, em identificação ou comparação de formas, pelo que não é necessário armazenar toda a imagem, mas apenas os objectos de interesse, conseguindo-se assim a vantagem adicional de poupar espaço de armazenamento. Basicamente existem duas possibilidades para descrever uma região: descrever o seu contorno, ou descrever o seu conteúdo. Vamos limitar o nosso estudo a algumas técnicas de descrição baseadas em contornos. Uma característica importante para estas técnicas de descrição é que devem ser o mais insensíveis que fôr possível a variações de tamanho e rotações.

9.1 Chain code

Esta técnica baseia-se no uso de identificadores para caracterizar a direcção dos contornos. O número de identificadores usados depende da conectividade considerada (ver figura 24).

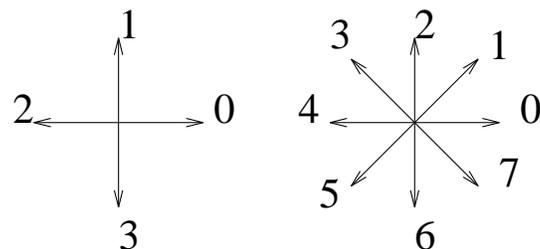


Figura 24: Identificadores para chain codes

O *chain code* é gerado começando por um ponto do contorno do objecto (normalmente o ponto mais acima e à esquerda) e avançando ao longo do contorno. A direcção do movimento de um ponto para outro é armazenada, construindo-se assim a sequência designada por *chain code*. O processo repete-se até que se regresse ao ponto de partida (figura 25). O algoritmo usado para caminhar ao longo do contorno deve ser cuidadosamente escolhido. O ideal é procurar o próximo ponto directamente à esquerda da direcção de entrada no ponto actual (quando o passeio ao longo do objecto é feito na direcção dos ponteiros do relógio) e depois procurar sucessivamente nas outras direcções (mais uma vez no sentido dos ponteiros do relógio) até ser encontrado um ponto que pertença ao contorno.

Na prática é inaceitável considerar todos os pontos do contorno por duas razões: a sequência resultante é demasiado grande e demasiado sensível a pequenas irregularidades existentes

no contorno (ruído?). A alternativa é gerar o *chain code* avançando de n em n pontos de cada vez.

O *chain code* gerado é dependente do ponto inicial e é sensível a rotações do objecto. Para obter uma sequência insensível a rotações pode-se normalizar o *chain code*, usando a diferença entre cada duas direcções sucessivas em vez do próprio código das direcções. Esta diferença é calculada contando o número de direcções que separam dois elementos sucessivos da sequência (no sentido contrário aos ponteiros do relógio). Para o primeiro elemento calcular a diferença entre o último e o primeiro elementos da sequência não normalizada. O ponto de início da sequência pode ser redefinido rodando a sequência normalizada até que esta forme o menor inteiro possível.

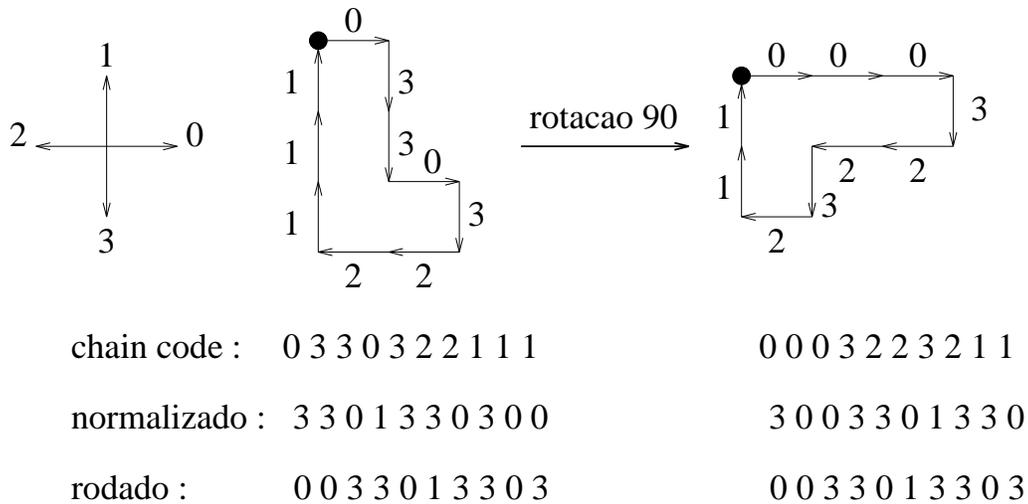


Figura 25: Chain codes

9.2 Assinaturas

Uma assinatura é uma representação unidimensional de um contorno. Existem vários métodos para gerar assinaturas. Independentemente do método usado a ideia é obter uma representação unidimensional que é mais simples que o contorno bidimensional original.

Uma das assinaturas mais simples consiste em representar a distância dos pontos do contorno a um ponto central do objecto em função de um ângulo θ (figura 26). Esta representação é invariante com a rotação, a menos de uma rotação horizontal do gráfico.

Um outro método para gerar uma assinatura é considerar o ângulo formado por 3 pontos sucessivos do contorno (figura 27). Este ângulo deve ser medido sempre na mesma direcção.

9.3 Aproximação poligonal

O contorno de qualquer objecto pode ser representado, com a precisão desejada, por um polígono. Uma descrição exacta é conseguida quando o número de lados do polígono é igual ao número de pontos do contorno, de tal forma que cada par de pontos adjacentes

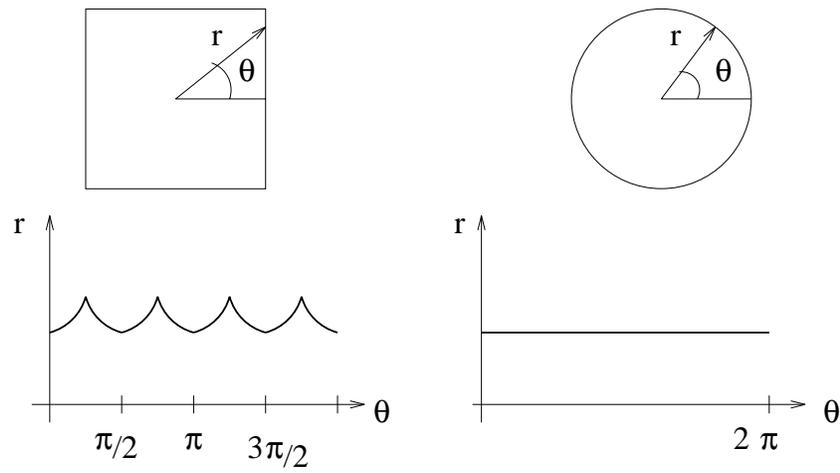


Figura 26: Assinatura em função da distância

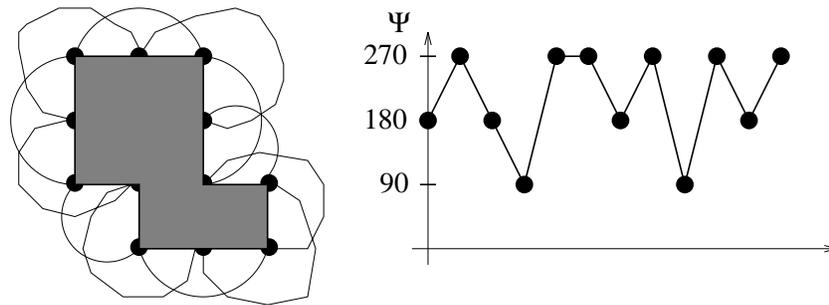


Figura 27: Assinatura em função do ângulo

define um segmento de recta. O que se pretende, no entanto, é determinar um polígono que dê uma aproximação ao contorno do objecto.

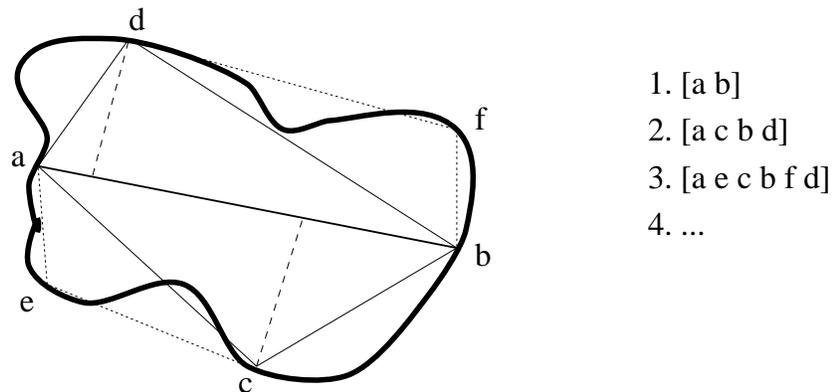


Figura 28: Aproximação poligonal

Descreve-se a seguir um método iterativo para obter uma aproximação poligonal (consultar figura 28):

1. Calcular uma linha que une dois extremos (a,b)
2. Determinar os dois pontos (c e d) que estão mais afastados do segmento [ab], um de cada lado deste segmento. Formar o polígono [acbd].

3. Se nenhum ponto do contorno está mais afastado do polígono até agora encontrado do que um determinado limite T , então terminar.
4. Para cada segmento do polígono determinar se existe um ponto do contorno que esteja mais afastado dele que o limite T , mas não está mais perto de nenhum outro segmento do polígono. Caso estes pontos existam então o polígono é estendido para os incluir.
5. Regressar ao passo 3

Cada iteração deste algoritmo resulta num polígono cada vez mais próximo do contorno do objecto em questão. A precisão do algoritmo depende do parâmetro T , que é a máxima distância que qualquer ponto do contorno pode ter relativamente ao segmento do polígono que está mais próximo dele. O nosso exemplo poderia resultar num polígono como o da figura 29.

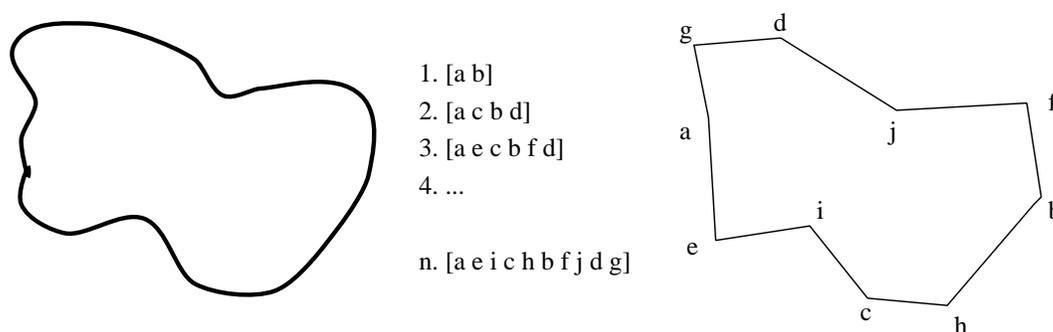


Figura 29: Aproximação poligonal com grande precisão

9.4 Esqueleto

Para representar a estrutura de uma região podemos reduzi-la a um grafo. Isto é conseguido usando técnicas de erosão. Em Gonzalez é proposto um algoritmo de erosão vocacionado para a esqueletização de objectos. Uma outra abordagem para determinar o esqueleto consiste em calcular a *Medial Axis Transform* (MAT). Para cada ponto p da região é calculado o conjunto de pontos do contorno que estão mais perto de p . Se este conjunto tiver mais do que um elemento, então p pertence ao contorno. Este algoritmo é muito agradável de um ponto de vista conceptual, mas é computacionalmente proibitivo, porque implica o cálculo da distância de todos os pontos da região a todos os pontos do contorno.

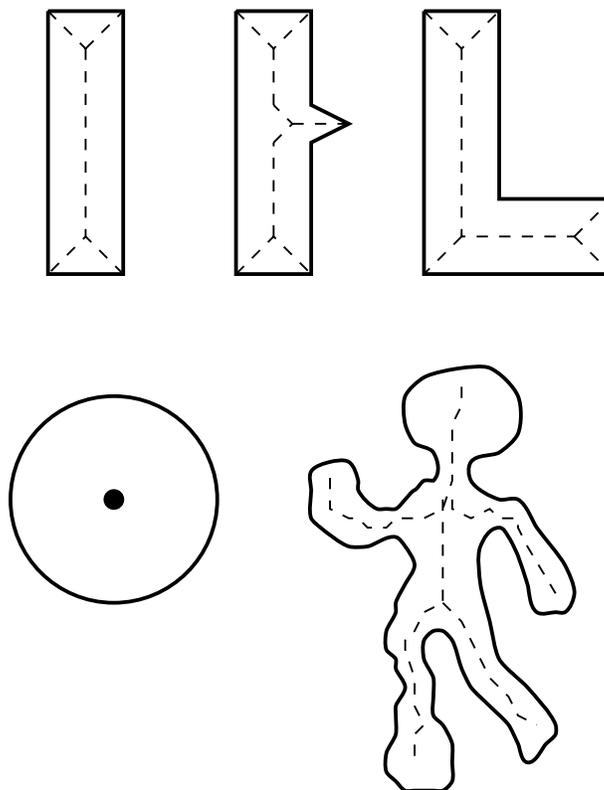


Figura 30: Esqueletos de várias regiões

Referências bibliográficas

- Gonzalez, Rafael and Wintz, Paul. Digital Image Processing. Addison-Wesley Publishing. 2ª edição. 1987
- Storer, Richard and Morgan, Andy. Introduction to Image Processing & Computer Vision. Universidade do Minho. 1991