

# Elementos de Imagem e Visão por Computador

V. Santos, 2006

## Algumas Referências

### ■ Livros

- Computer and Robot Vision - *R. Halarick & L. Shapiro, Addison Wesley, 1992*
- Robotics: Control, Sensing, Vision, and Intelligence – *K. S. Fu, R. C. Gonzalez, C. D. G. Lee, McGrawHill, 1987*
- Digital Image Processing – *R.C. Gonzalez, P. Wintz, Addison-Wesley, 1987*

### ■ Internet

- Image Processing Fundamentals (introdutório)
  - <http://www.ph.tn.tudelft.nl/Courses/FIP/frames/fip.html>
  - [http://www.cee.hw.ac.uk/hipr/html/hipr\\_top.html](http://www.cee.hw.ac.uk/hipr/html/hipr_top.html)
- Computer Vision Online (mais completo e avançado)
  - <http://homepages.inf.ed.ac.uk/rbf/CVonline/>

### ■ Software

- Matlab – *image processing toolbox*
- OpenCV Library

### ■ Entre muitas..., muitas outras fontes!

# Revisões e Generalidades

## Visão por Computador (VC) – que é?

### ■ Algumas definições:

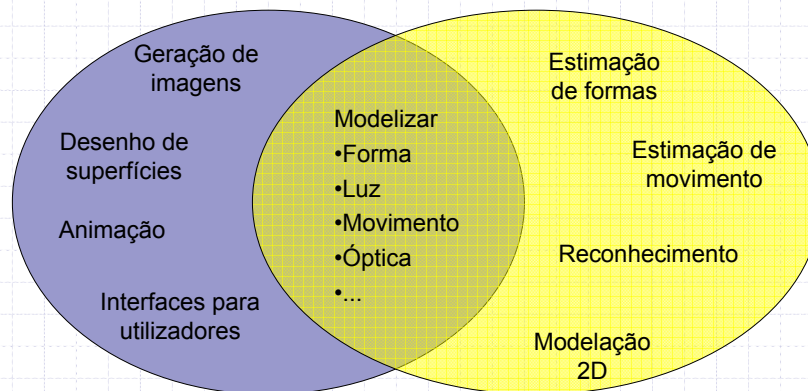
- Compreensão do conteúdo de imagens
  - Está ligada a: Inteligência Artificial, comportamento
- Um modo sensorial em robótica
- Emulação computadorizada da visão humana
- O inverso de computação gráfica (CG)
  - A CG parte das concepções (ou modelos do mundo) para as representações – imagens
  - A VC parte das imagens para extrair os modelos

### ■ Formalmente, também se diz que:

- Visão por computador é o processo de ...
- ... extracção, caracterização e interpretação
- ... de informação a partir de imagens do mundo tridimensional

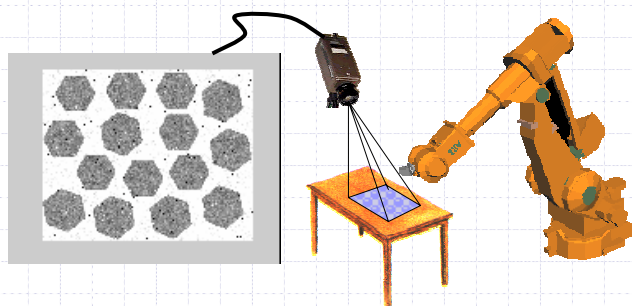
## Intersecção entre a Visão e a Computação Gráfica

### Computação Gráfica      Visão por computador



vsantos, 2006

## Um exemplo de problema a resolver com Visão por Computador



**Como fazer?**

- A partir de uma imagem com diversos elementos, deve o robot localizar correctamente cada um deles para depois o poder pegar
  - Passo 1 – Obtenção e representação (numérica) da imagem
  - Passo 2 – Identificação da posição de cada objecto de interesse
  - Passo 3 – Cálculo dos movimentos a impor ao robot para pegar nos objectos e manipulá-los

vsantos, 2006

## O velho “jogo” do fundo e do objecto!



É exactamente a mesma imagem... Rodada 180°!

vsantos, 2006

<http://www.wisdom.weizmann.ac.il/~hassner/cv0203/>

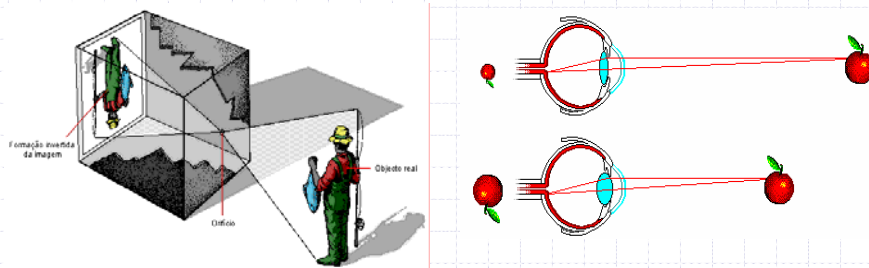
## Áreas e níveis da visão por computador

Baixo nível	<b>1-Percepção</b> Aquisição ou obtenção de imagem
	<b>2-Pré-processamento</b> Redução de ruído ou realce de detalhes
Médio nível	<b>3-Segmentação</b> Partição da imagem em zonas de interesse (objectos)
	<b>4-Descrição</b> Determinação de propriedades de objectos (dimensões, formas,...)
Alto nível	<b>5-Reconhecimento</b> Identificação de objectos descritos a partir de uma base de dados.
	<b>6-Interpretação</b> Atribuir significados a um conjunto de objectos reconhecidos

vsantos, 2006

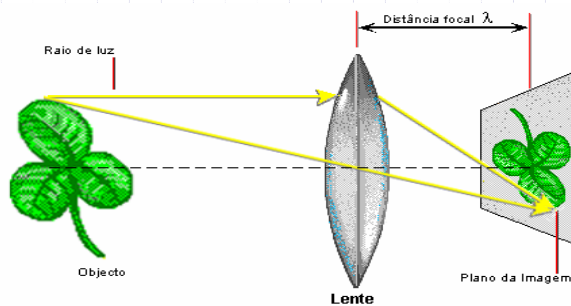
## Formação da imagem

- ...na câmara escura
  - As dimensões e definição do orifício definem a nitidez e o plano focal
- ...no olho humano
  - A imagem forma-se no plano da retina por projecção da luz vinda do mundo 3D através de uma lente (cristalino) que muda de dimensões para ajustar a distância focal



vsantos, 2006

## Imagem através de uma Lente



- Todos os raios de luz que saem de um dado ponto de um objecto na direcção de uma lente convergente vão confluir, do outro lado, num outro ponto a uma certa distância  $\lambda$  do centro dessa lente: trata-se da distância focal.
- O conjunto de todos os pontos para os quais se verifica esse facto constitui o plano focal.
- Assim, a imagem observada nesse plano diz-se focada.

vsantos, 2006

## Imagem digital

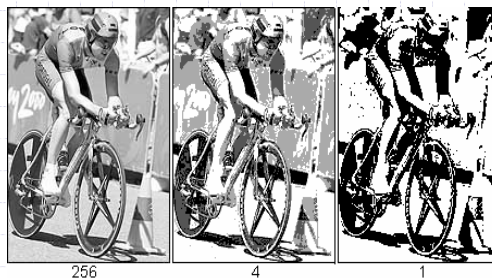
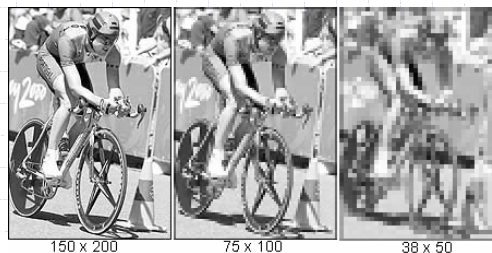
- Uma imagem como uma função  $z=f(x,y)$ 
  - $z$  - intensidade da luz no ponto  $(x,y)$
  - $x, y$  - linha e coluna na imagem respectivamente
- Uma imagem amostrada é representável por uma matriz de dimensão  $(N \times M)$

$$f(x, y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0, M-1) \\ f(1,0) & f(1,1) & \dots & f(1, M-1) \\ \vdots & \ddots & & \vdots \\ f(N-1,0) & f(N-1,1) & \dots & f(N-1, M-1) \end{bmatrix}$$

vsantos, 2006

## Resolução numa Imagem

- Resolução espacial
  - Número de linhas e colunas – *pixels* verticais e horizontais
  - Efeito mosaico
- Resolução de amplitude ou profundidade
  - Número de níveis de cinzento ou cores por *pixel* – nas imagens a cores há mais que um valor por *pixel* (é usual três)
  - O número de bits e efeito de “solarização”
  - Nas imagens binárias, os *pixels* são representados por um só bit

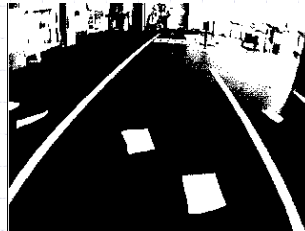


vsantos, 2006

## Questões de Iluminação

### ■ Importância:

- A iluminação pode simplificar os algoritmos a usar, eliminando sombras, reflexões especulares, etc.
- Ex. no caso abaixo, a iluminação inadequada da cena condiciona a correcta extracção de informação:



- É muito difícil (por ser muito variável o limite de decisão) estabelecer o que são linhas e o que é pista!

vsantos, 2006

## Técnicas de Iluminação

### ■ Definição

- Metodologias de aplicar a luz sobre uma cena de forma a simplificar a extracção de propriedades.

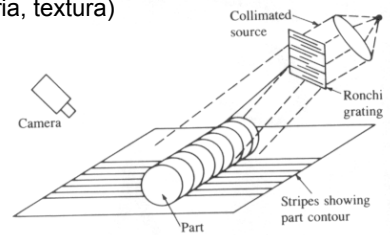
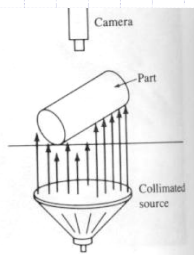
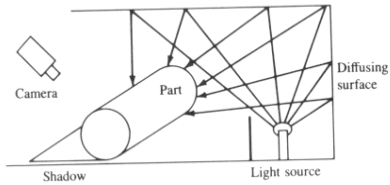
### ■ Técnicas principais:

- Iluminação difusa (não directa)
  - Para objectos de superfícies suaves e regulares
- Retro-iluminação (contraluz)
  - Para imagens a dois níveis onde só a geometria dos objectos tenha interesse.
  - Clara definição logo à partida de objecto e de fundo
- Luz estruturada
  - Projecção de pontos, linhas ou grelhas na superfície de trabalho (sobre a cena ou objectos)
  - Útil para a percepção tridimensional de objectos.

vsantos, 2006

## Técnicas de Iluminação-Ilustração

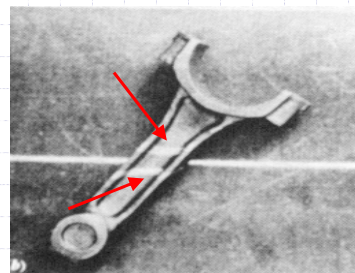
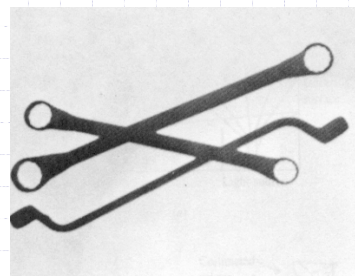
- Iluminação difusa
  - Eliminar sombras
- Luz estruturada
  - Análise geométrica
- Retro-iluminação
  - Alto contraste-imagens binárias
- Iluminação direccionada
  - Realce de propriedades (geometria, textura)



vsantos, 2006

## Exemplos de Retro-Iluminação e de Luz Estruturada

- Conjunto de chaves sobre superfície translúcida iluminada por “baixo”. Grande contraste na imagem obtida.
- Dois feixes de luz coincidentes num plano perdem essa coincidência quando projectados sobre uma superfície a uma cota diferente do plano de apoio.

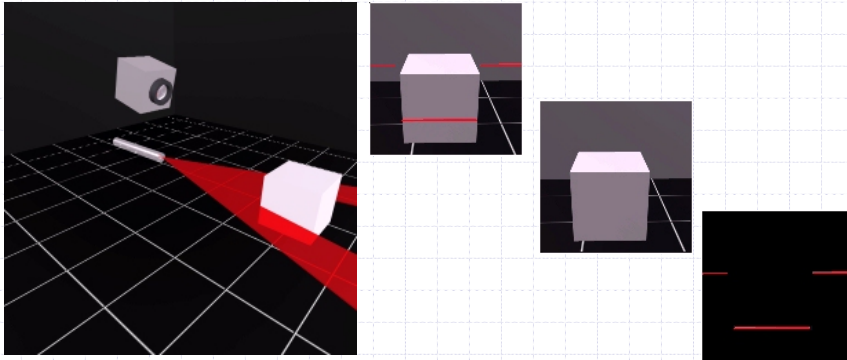


vsantos, 2006



## Luz Estruturada: Princípios - I

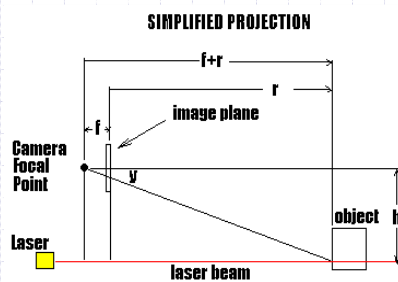
- Projecção de padrões visíveis numa superfície
  - Projecção com feixes laser ou outras fontes coerentes
  - Percepção dos padrões da imagem visível e extracção das propriedades



vsantos, 2006

## Luz Estruturada: Princípios - II

- Câmara posicionada à distância  $r$  (desconhecida) do objecto e à altura  $h$  (conhecida) do feixe laser
- $y$  – coordenada do ponto do padrão luminoso na imagem obtida
- Por simples relação de triângulos, sabendo a distância focal  $f$  da câmara, determina-se a distância  $r$  do objecto



$$\frac{y}{f} = \frac{h}{r+f} \quad r = \frac{f \cdot h}{y} - f$$

vsantos, 2006

# Geometria da Formação da Imagem

## Geometria da Formação de Imagem

- Ponto real e ponto na imagem

- Ponto real:  $\mathbf{w} \rightarrow (X, Y, Z)$
- Ponto na imagem:  $\mathbf{c} \rightarrow (x, y)$

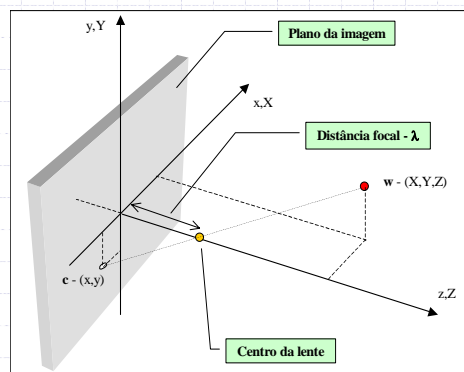
- Relações de distâncias:

- Se  $Z > \lambda$  virá:

$$\frac{x}{\lambda} = -\frac{X}{Z - \lambda} = \frac{X}{\lambda - Z}$$

$$\frac{y}{\lambda} = -\frac{Y}{Z - \lambda} = \frac{Y}{\lambda - Z}$$

$$x = \frac{\lambda X}{\lambda - Z} \quad y = \frac{\lambda Y}{\lambda - Z}$$

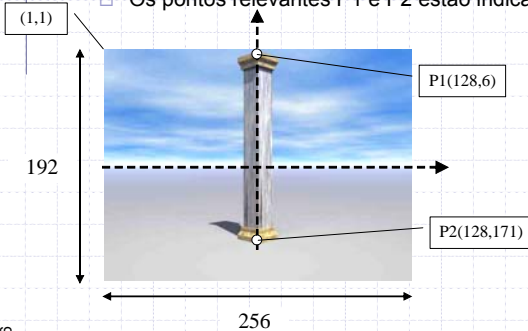


- Significado das expressões:

- Calcular as coordenadas na imagem a partir das reais e vice-versa, mas esta última com limitações...! Quais?

## Exemplo de cálculo geométrico da imagem

- Numa imagem de 256 X 192 pixels
  - CCD de 150 pixels/mm
  - $\lambda = 6$  mm
- Determinar a distância a que está uma coluna que se sabe estar paralela ao plano de imagem (da câmara) e ter 2 metros de altura.
  - Os pontos relevantes P1 e P2 estão indicados na figura



Passos:

- 1-Converter os pontos de *pixels* em coordenadas na câmara
- 2-Escrever as equações para os pontos conhecidos P1 e P2
- 3-Estabelecer a relação conhecida dos pontos correspondentes no ambiente real
- 4-Resolver as equações resultantes e obter a distância a que está a coluna

Solução: aprox. 10.9 m

$$x = \frac{\lambda X}{\lambda - Z} \quad y = \frac{\lambda Y}{\lambda - Z}$$

vsantos, 2006

vsantos, 2006

## Geometria da Imagem – Abordagem matricial

- Abordagem integrada das coordenadas
- As Coordenadas homogêneas de um ponto  $w$  do mundo físico ( $w_h$ )
  - Ponto real:  $w_h \rightarrow (X, Y, Z, 1)$ 
    - É o caso particular de  $(kX, kY, kZ, k)$ , onde  $k \neq 0$  é um factor de escala global
  - Nesta representação de coordenadas, qual o valor do ponto na imagem:  $c_h$  ?
    - As coordenadas têm de resultar como na transformação geométrica anterior
- Seja a Transformação de perspectiva, designada por  $P$ , tal que:
  - $c_h = P w_h$

$$\mathbf{P} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -\frac{1}{\lambda} & 1 \end{bmatrix} \mathbf{c}_h = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -\frac{1}{\lambda} & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \mathbf{c}_h = \begin{bmatrix} X \\ Y \\ Z \\ -\frac{Z}{\lambda} + 1 \end{bmatrix} = \begin{bmatrix} X \\ Y \\ Z \\ \frac{\lambda - Z}{\lambda} \end{bmatrix}$$

Factor de escala global no sistema da câmara

## Recuperação das coordenadas normalizadas

- Para recuperar as coordenadas “normais” a partir das homogêneas é necessário levar em conta o factor de escala global.
- Ou seja, das coordenadas em formato homogêneo...
- ... passa-se às coordenadas normalizadas para um factor de escala global de 1, portanto, condizente com a medida física no elemento sensível (sensor CCD)
  - ... e que tem geralmente uma relação directa com a numeração dos *pixels* na imagem
- A coordenada  $z$  na imagem não tem significado prático nem grande utilidade dado que na imagem só há duas dimensões!
  - É um subproduto do processo matricial
- E qual a vantagem do processo matricial?

$$\mathbf{c}_h = \begin{bmatrix} X \\ Y \\ Z \\ \frac{\lambda - Z}{\lambda} \end{bmatrix}$$

$$\mathbf{c} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \frac{\lambda X}{\lambda - Z} \\ \frac{\lambda Y}{\lambda - Z} \\ \frac{\lambda Z}{\lambda - Z} \end{bmatrix}$$

## Exemplo de cálculo matricial

- Numa câmara com CCD de 150 pixels/mm e  $\lambda = 6$  mm como aparecem na imagem os 4 pontos com as seguintes coordenadas no mundo real?
  - (0, 0, 2000), (0, 100, 2000), (100, 100, 2100), (100, 0, 2100)
- É aqui vantajoso usar a operação matricial e depois normalizar as coordenadas

$$\mathbf{c} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \frac{\lambda X}{\lambda - Z} \\ \frac{\lambda Y}{\lambda - Z} \\ \frac{\lambda Z}{\lambda - Z} \end{bmatrix}$$

$$[\mathbf{c}_h] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -\frac{1}{\lambda} & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0.1 & 0.1 \\ 0 & 0.1 & 0.1 & 0 \\ 2 & 2 & 2.1 & 2.1 \\ 1 & 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0.1 & 0.1 \\ 0 & 0.1 & 0.1 & 0 \\ 2.0 & 2.0 & 2.1 & 2.1 \\ -0.332 & -0.332 & -0.349 & -0.349 \end{bmatrix}$$

- Em coordenadas reais na imagem (x e y) virão os seguintes 4 pontos, após a divisão das coordenadas pelo termo de escala global respectivo:
  - -0.332 ou -0.349 respectivamente, neste exemplo

$$[\mathbf{c}] = \begin{bmatrix} 0 & 0 & -0.2865 & -0.2865 \\ 0 & -0.3009 & -0.2865 & 0 \end{bmatrix}$$

vsantos, 2006

## Calibração (parcial) da Câmara

- Quando os sistemas de coordenadas da câmara e do mundo real são diferentes
- Necessidade de adequada transformação geométrica da câmara antes da perspectiva
  - Simples relação de transformações...

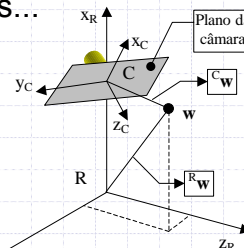
$${}^R \mathbf{w} = {}^R \mathbf{T}_C \cdot {}^C \mathbf{w}$$

$${}^C \mathbf{w} = ({}^R \mathbf{T}_C)^{-1} \cdot {}^R \mathbf{w} = {}^C \mathbf{T}_R \cdot {}^R \mathbf{w}$$

- ... e transformação perspectiva:

$$\mathbf{c} = \mathbf{P} \cdot {}^C \mathbf{w} = \mathbf{P} \cdot ({}^R \mathbf{T}_C)^{-1} \cdot {}^R \mathbf{w} = \mathbf{P} \cdot {}^C \mathbf{T}_R \cdot {}^R \mathbf{w}$$

Relação entre a câmara e o referencial



vsantos, 2006

# Processamento a Baixo Nível

## Relações entre *pixels*

### ■ Vizinhança de um *pixel* (nº de *pixels* vizinhos)

- $N_4$  (horizontal e vertical)



- $N_D$  (diagonal)



- $N_8 = N_4 \cup N_D$



### ■ Medidas de distância entre *pixels*

- Euclidiana

$$D_E(p, q) = \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2}$$

- D4 (ou *city-block*)

$$D_4(p, q) = |p_x - q_x| + |p_y - q_y|$$

- D8 (ou *chessboard*)

$$D_8(p, q) = \max(|p_x - q_x|, |p_y - q_y|)$$

## Princípio da utilização de um filtro

- Novo *pixel* é função do *pixel* original e do filtro:

$$g(x, y) = \sum_{i=1}^N w_i \cdot p_i$$

onde  $p_i$  pertence à vizinhança do ponto em causa

- Note-se que  $g(x,y)$  pode ser o valor directo do novo pixel, ou ser usado de forma indirecta e assim permitir calcular o seu novo valor.
- Os coeficientes de um filtro de 3x3:

w1	w2	w3
w4	w5	w6
w7	w8	w9

vsantos, 2006

## Filtro para detectar pontos isolados

- Numa imagem binária (*pixels* com valores 0 ou 1), o seguinte filtro pode ser usado para detectar se um *pixel* é um ponto isolado no meio de outros.

-1	-1	-1
-1	8	-1
-1	-1	-1

- Se  $g(x,y) = 8$  ou  $-8 \Rightarrow$  o ponto  $(x,y)$  é isolado!
- Note que os valores de  $-7, -6, -5, \dots, 6, 7$  correspondem a todas as outras possibilidades que não são pontos isolados

vsantos, 2006

## Técnicas de Suavização (*smoothing*)

- As técnicas de suavização dizem normalmente respeito à **redução** de ruído
  - Média de imagens (série de N imagens da mesma cena)

$$g(x, y) = \frac{1}{N} \sum_{i=1}^N f_i(x, y)$$

- Filtro de média
  - Um filtro de 3x3 com todos os pesos de valor 1/9.
- Filtro de mediana
  - Substituir cada *pixel* pela mediana da distribuição dos níveis de todos os *pixels* da sua vizinhança (incluindo o próprio). Filtros de 3x3 ou 5x5 são comuns. Vantagens em relação à média...?
  - Exemplo a 3x3:

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

12	10	15
11	12	9
10	11	15

9, 10, 10, 11, 11, 12, 12, 15, 15

12	10	15
11	11	9
10	11	15

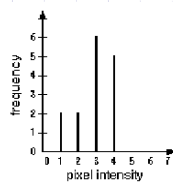
vsantos, 2006

## Técnicas de realce (*Enhancement*)

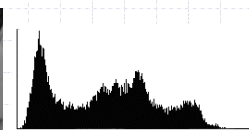
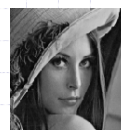
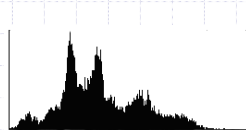
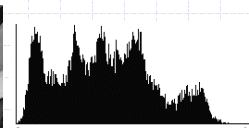
- Para adaptação ou compensação face a alterações de iluminação da cena
- O Histograma de uma imagem
  - Conceito:

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

image

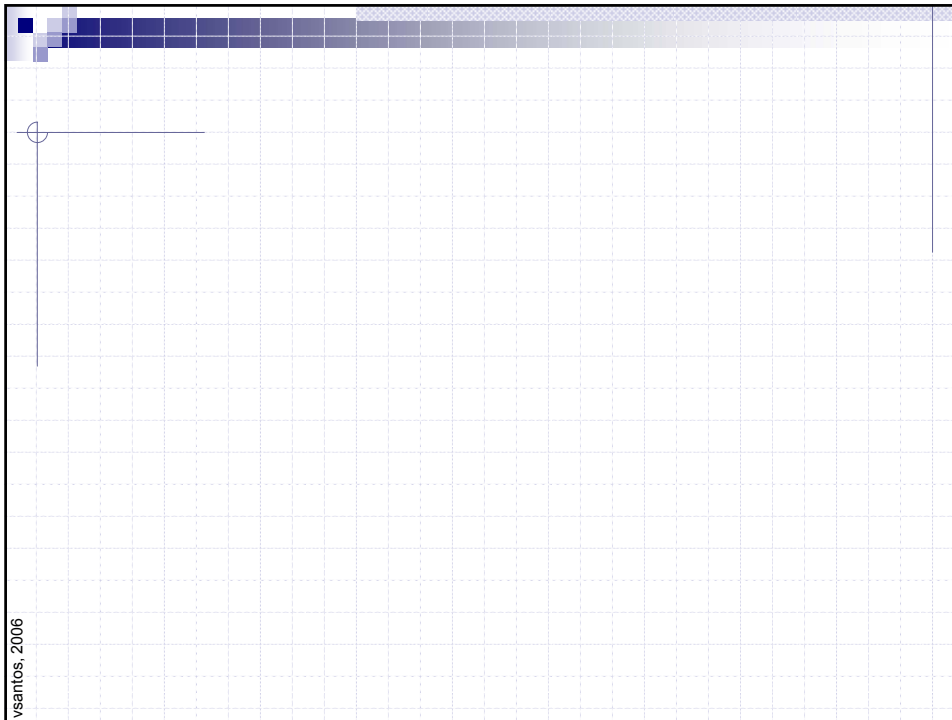


### Exemplos



vsantos, 2006





## Operações no histograma

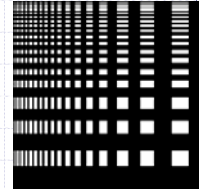
- Equalização
  - alteração do valor de cada *pixel* para uniformizar a distribuição de níveis de cinzento na imagem
- Histogramas locais em vez de histograma global
  - Para afectar a imagem parcialmente sem levar em conta todo o resto da imagem
- Aplicação de um histograma pré-definido
  - Por exemplo, para realçar as partes mais escuras tornando-as mais claras, e escurecendo as partes mais intensas
- Expansão do contraste
  - Alteração da imagem para varrer toda a gama dinâmica de cinzentos com B bits:

$$g(x, y) = (2^B - 1) \frac{f(x, y) - \min[f(x, y)]}{\max[f(x, y)] - \min[f(x, y)]}$$

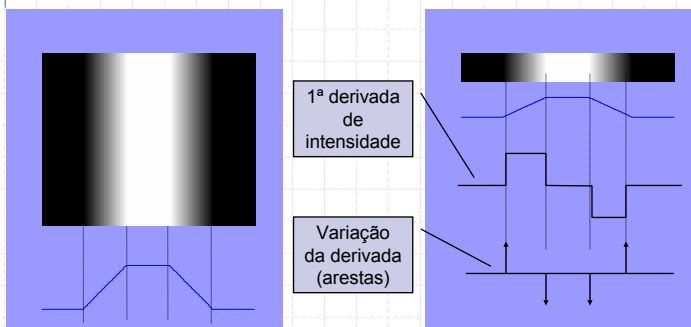
## Detecção de arestas (*Edge detection*)

- Arestas são regiões de grande contraste nas intensidades dos pixels.
- São importantes para a detecção de objectos.
- A técnica principal é a de detectar variações de intensidade usando um operador derivativo local

Exemplo simples (BW):



Exemplo de perfil de intensidades e suas derivadas:



vsantos, 2006

## Operador de gradiente para arestas

- Definição  $\vec{G} = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} \end{bmatrix}^T$
- Norma do gradiente e sua aproximação usual

$$\|\vec{G}\| = \sqrt{G_x^2 + G_y^2} \approx |G_x| + |G_y|$$

- Variante discreta...

$$G_x = f(x, y) - f(x-1, y) \quad G_y = f(x, y) - f(x, y-1)$$

- ... e as máscaras correspondentes

0	0	0
-1	1	0
0	0	0

0	-1	0
0	1	0
0	0	0

- Obtido o valor do gradiente em todos os *pixels*...

- ...procuram-se aqueles que têm um máximo local ou o que excedem um dado limiar;
- ...esses pontos pertencerão a arestas!

vsantos, 2006

## Outros operadores de gradiente

### Operadores de Sobel

- Alternativa usualmente mais eficaz para detectar arestas do que simples gradiente

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

### Laplaciano

- Assemelha-se a operação de 2ª derivada.
- Muito sensível ao ruído. Pouco usado para detecção de arestas.

$$L[f(x, y)] = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

$$L = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

vsantos, 2006

## Outros operadores para arestas (cont.)

### Operador de Prewitt

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

$$G_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

### Operador de Roberts (2X2)

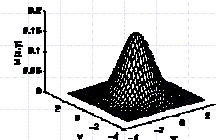
$$G_x = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

$$G_y = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

### Método de Canny (dos mais populares)

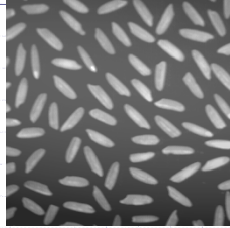
- Operação com duas fase principais:
  - Aplicação de uma Convolução Gaussiana
  - Aplicação de Filtro de Roberts

Ver mais detalhes em <http://www.cee.hw.ac.uk/hipr/html/canny.html>

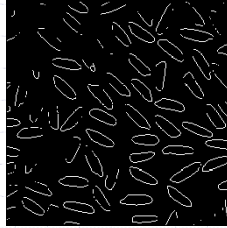


vsantos, 2006

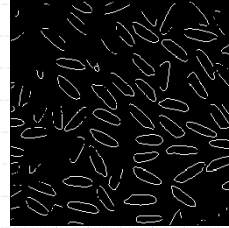
## Exemplos de detecção de arestas - 1



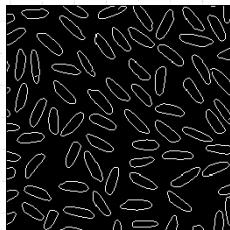
Raw



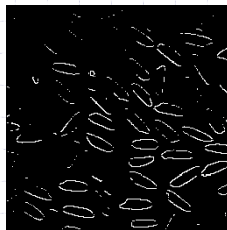
Prewitt



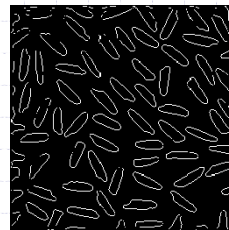
Sobel



Canny



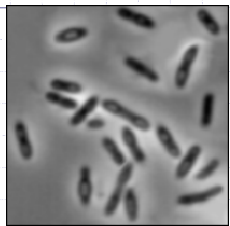
Roberts



Log

vsantos, 2006

## Exemplos de detecção de arestas - 2



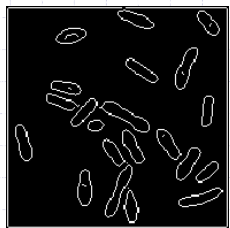
Raw



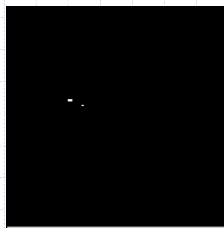
Prewitt



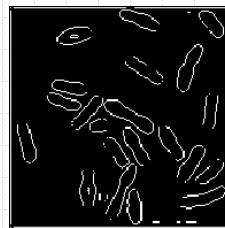
Sobel



Canny



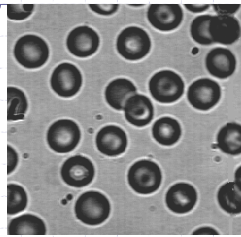
Roberts



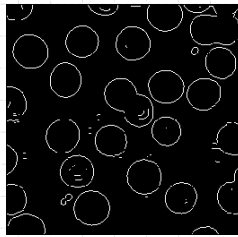
Log

vsantos, 2006

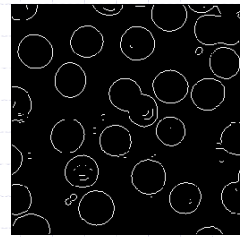
## Exemplos de detecção de arestas - 3



Raw



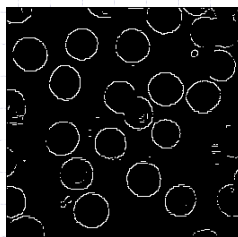
Prewitt



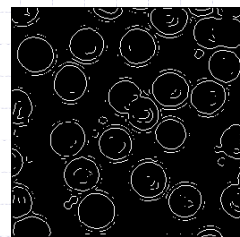
Sobel



Canny



Roberts



Log

vsantos, 2006

## Limitação (*Thresholding*)

- Processo de redução dos níveis de cinzento de uma imagem:
  - Em geral para 2 níveis, ou seja, imagem binária.

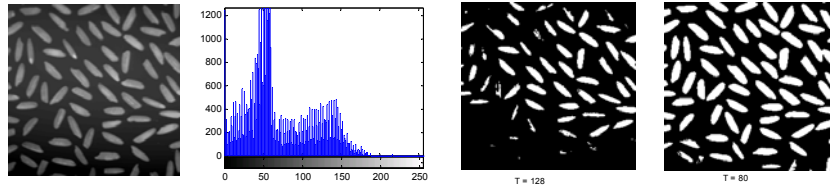
$$g(x, y) = \begin{cases} 1 & \Leftarrow f(x, y) > T \\ 0 & \Leftarrow f(x, y) \leq T \end{cases}$$

- A limitação é a técnica mais utilizada em visão e provavelmente a mais determinante nos processos actuais de percepção.
- É uma primeira operação para “separar” o objecto do fundo.

vsantos, 2006

## Limitação dinâmica

- Qual o *threshold* que mais informação preserva no processo de limitação? O valor central...? Nem sempre!
- Exemplo de limitação numa imagem com 256 níveis



- A melhor solução é obtida após análise do histograma
  - Procura das regiões de separação de picos (vales)
  - T=128 (pior) e T=80 (melhor)

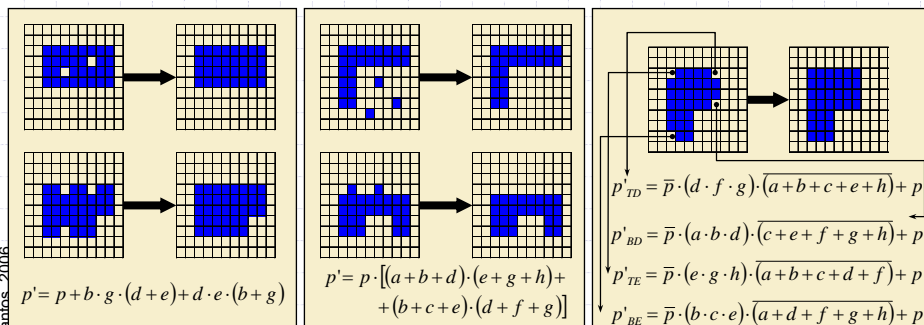
vsantos, 2006

## Suavização de imagens binárias

- Casos comuns de “suavização” em imagem binária
  - Pontos isolados ou “buracos” em imagens
  - Cavidades/protuberâncias em troços rectos
  - Vértices de objectos (cantos) ausentes
- Uso de operações lógicas (binárias) com os pixels!
  - Vizinhança de 3X3 onde os pixels são designados assim:

a	b	c
d	p	e
f	g	h




0 - fundo  
1 - objecto



vsantos, 2006

## Abordagem a Médio Nível e Morfologia

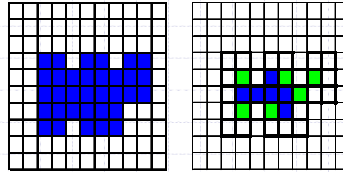
### Visão por computador – O médio nível

- Operadores morfológicos (binários)
  - Operadores para lidar com as partes da imagem divididas em “objectos” e “fundo” (*background*)
  - Um objecto é um conjunto de *pixels* que partilha uma mesma propriedade – como o estarem ligados entre si (conexão 4- ou 8-, indicando estar ligado a 4 ou a 8 vizinhos)
- O elemento estruturante de um operador morfológico
  - Vizinhança de efectividade do operador:
    - $N_4$    $N_8$  
    - Poderão ser definidos outros elementos estruturantes, ex.: 
- A segmentação
  - Separação de objectos “de interesse” do fundo da imagem
  - Referência a algumas técnicas (sequências de operações morfológicas) para questões de tratamento de “objectos”.

## Operadores Morfológicos - I

### ■ Erosão

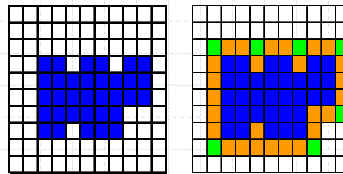
- Tomar cada *pixel* de objecto (valor 1) que esteja N-ligado ( $N_4$ ,  $N_8$ , ...) a um *pixel* de fundo (valor 0) e colocar esse *pixel* do objecto a 0.
  - Em geral, reduz as dimensões do objecto



Erosão com  $N_4$  e  $N_8$

### ■ Dilatação

- Tomar cada *pixel* de objecto (valor 1) e pôr todos os *pixels* de fundo (valor 0) que estão N-ligados ( $N_4$ ,  $N_8$ , ...) a esse *pixel* como objecto (para o valor 1).
  - Em geral, aumenta as dimensões do objecto



Dilatação com  $N_4$  e  $N_8$

vsantos, 2006

## Operadores Morfológicos - II

### ■ Fecho

- Operação de dilatação seguida de erosão
  - O resultado comum mais interessante é o de fundir objectos que inicialmente só estariam ligados por um único *pixel*.
  - “Suaviza” o objecto pelo exterior do seu contorno

### ■ Abertura

- Operação de erosão seguida de dilatação
  - O resultado comum mais interessante é o de separar objectos que inicialmente só estariam ligados por poucos *pixels*, nomeadamente, linhas de um só *pixel* de “largura”.
  - “Suaviza” o objecto pelo interior do seu contorno

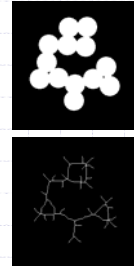
vsantos, 2006



## Operadores Morfológicos - III

### ■ Esqueletização

- Operação de determinação do esqueleto
- Definição de esqueleto:
  - Objecto filiforme (1 *pixel* de largura) ...
  - .. que passa pelo "meio" do objecto
  - ... e que preserva a topologia do objecto original



### ■ Uma variante ("thinning") baseada na erosão condicionada (menos exigente computacionalmente)

- Um *pixel* não é erodido nas seguintes condições:

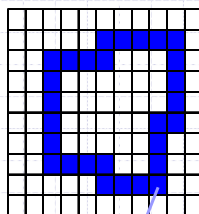
- Se for isolado - condição  $C_1$ :
- Se removê-lo afectar a conectividade - condição  $C_2$ :
- Se removê-lo afectar o comprimento da linha - Condição  $C_3$ :

vsantos, 2006

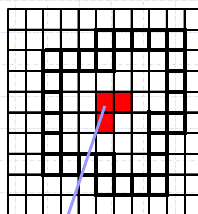
## Operadores Morfológicos – IV

### ■ Propagação (*propagation* ou *fill*)

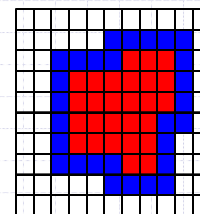
- Operação de dilatação sucessiva de um objecto "semente" (por exemplo um esqueleto) até aos limites dados por uma imagem "máscara", com base num dado elemento estruturante.



Máscara

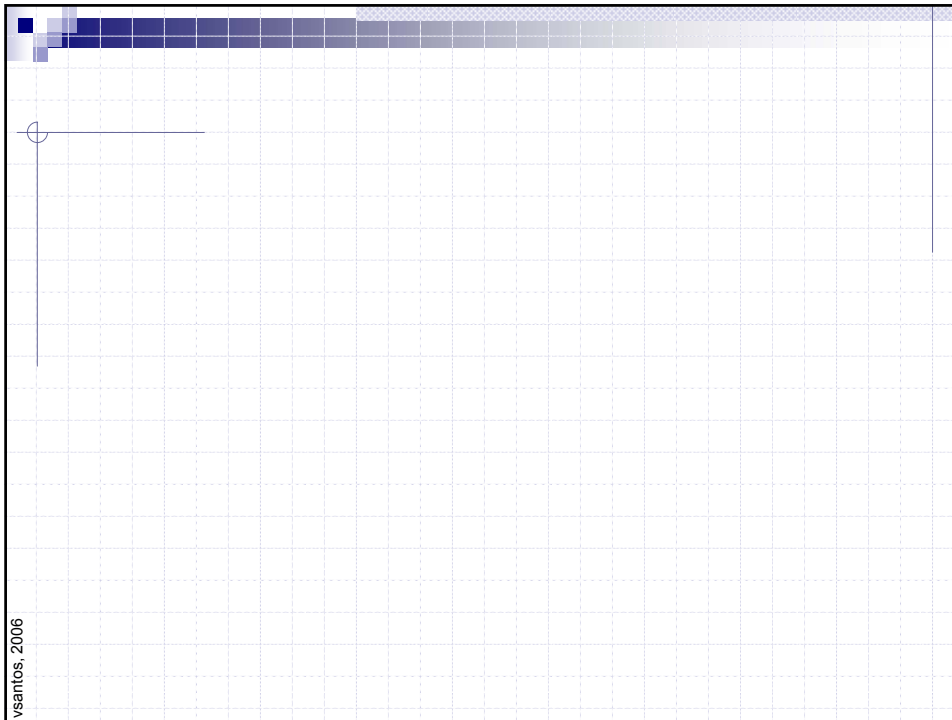


Semente



Em geral, o processo resulta numa máscara "cheia"

vsantos, 2006



vsantos, 2006

## Variantes de Esqueletização

- No algoritmo de “*thinning*” há diversas combinações das condições  $C_1$ ,  $C_2$  e  $C_3$ .
  - Só condição  $C_1$ 
    - Esqueleto reduzido a um *pixel*
  - Só condição  $C_2$ 
    - Esqueleto reduzido a linhas fechadas se houver objectos com buracos. Objectos sem buracos desaparecem.
  - $C_1 + C_2$ 
    - O esqueleto é constituído por *pixels* isolados para objectos sem “buracos” e por contornos fechados para objectos com “buracos”.
  - $C_1 + C_2 + C_3$ 
    - Ter-se-á o esqueleto completo tradicional

vsantos, 2006

## Segmentação – separação de objectos

- Algumas técnicas de limitação dinâmica
  - Limite baseado na média do valor de todos os pixels da imagem
    - É um método com limitações quando há grande desequilíbrio entre dimensão de objectos e fundo!
  - Limite baseado em médias locais
    - Define-se uma vizinhança de dimensão limitada.
  - Limite baseado no histograma
    - Algoritmo de “isodados”
      - Processo iterativo que procura o nível médio do que é fundo e do que é objecto.
    - Algoritmo do “triângulo”
      - Particularmente eficiente quando os objectos não se demarcam claramente no histograma
    - Algoritmo de “simetria do fundo”
      - Pico demarcado e com simetria no histograma
- O recurso à detecção de arestas
  - Obtenção dos contornos dos objectos (através de filtros de gradiente)

vsantos, 2006

## Binarização- Algoritmo de “isodados”

- Obtenção do limiar T de binarização de forma iterativa.
  - Imagem a B bits com um histograma H(i) onde  $i=0, 1, \dots, 2^B-1$
  - Iteração inicial: ponto central da gama dinâmica  $T_0=2^{B-1}$
  - Calcular as médias “ponderadas” dos *pixels* separados em “objectos” ( $m^f$ ) e “fundo” ( $m^b$ ):

$$m_k^b = \frac{\sum_{i=0}^{T_{k-1}-1} i \cdot H(i)}{\sum_{i=0}^{T_{k-1}-1} H(i)} \quad m_k^f = \frac{\sum_{i=T_{k-1}}^{2^B-1} i \cdot H(i)}{\sum_{i=T_{k-1}}^{2^B-1} H(i)}$$

- Nova iteração: o novo valor do limiar é obtido como a média aritmética dessas médias.

$$T_k = \frac{m_{k-1}^f + m_{k-1}^b}{2}$$

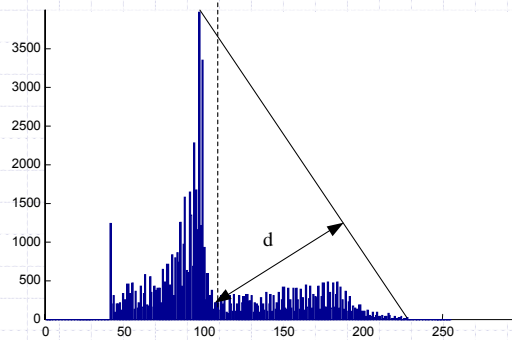
- O processo é repetido até que o limiar deixe de variar.

$$T_k = T_{k-1}$$

vsantos, 2006

## Binarização – algoritmo do “triângulo”

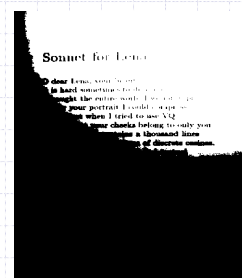
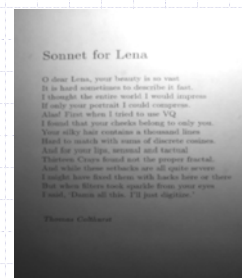
- Determinar no histograma o ponto de maior frequência
- Determinar no histograma o ponto de menor frequência
- Definir uma linha unindo os dois pontos perfazendo um triângulo
- O ponto do histograma entre os pontos definidos que mais se afastar dessa linha ao longo de uma perpendicular é o limiar...



vsantos, 2006

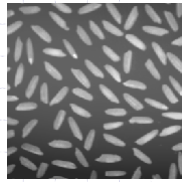
## Binarização – Médias locais

- Quando há gradientes de iluminação, é impossível definir um limiar fixo para toda a imagem
- Solução:
  - Definir uma vizinhança de determinada dimensão
  - fazer a binarização de cada pixel com base na média dessa vizinhança
- Nem sempre o processo é ideal, dependendo da natureza da imagem...
  - Qual a melhor dimensão da vizinhança?
  - Zonas vastas de fundo (ou de objecto) com flutuações, como se garantem de uma só “côr”?
    - Usar as chamadas *mean-C* (*média-C*): médias com valores limites de aceitação:
      - Se a média for “muito” baixa o pixel deve ser considerado “fundo” mesmo se o seu valor é mais elevado que essa média, e reciprocamente!
  - Em resumo, o processo tem dois parâmetros
    - Dimensão da vizinhança
    - Tolerância para a *média-C*

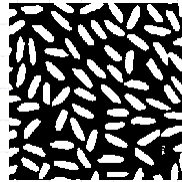


vsantos, 2006

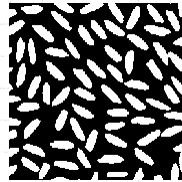
## Binarização dinâmica – Comparações - 1



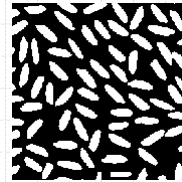
Original



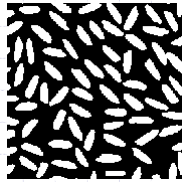
Media Local: 30 viz.



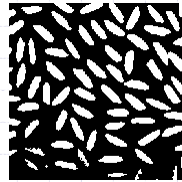
Media Local: 50 viz.



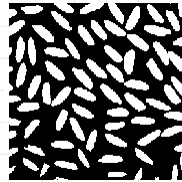
Media Local: 70 viz.



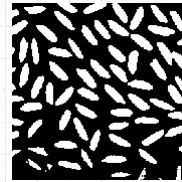
Media Local: 90 viz.



Isodados



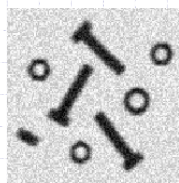
Media Global



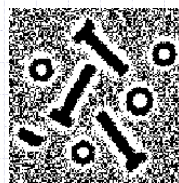
Limiar global

vsantos, 2006

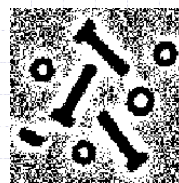
## Binarização dinâmica – Comparações - 2



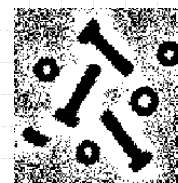
Original



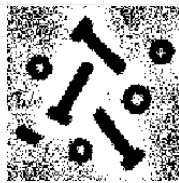
Media Local: 30 viz.



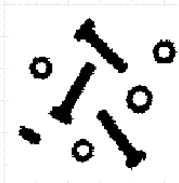
Media Local: 50 viz.



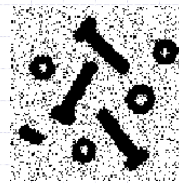
Media Local: 70 viz.



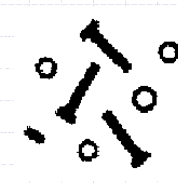
Media Local: 90 viz.



Isodados



Media Global



Limiar global

vsantos, 2006

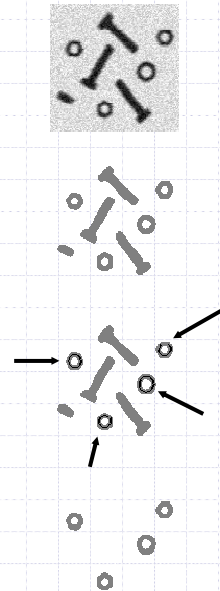
## Operações de morfologia binária

- Alguns exemplos comuns de operações morfológicas
  - Isolar objectos com buracos
  - Preencher os buracos em objectos
  - Remover objectos nos bordos da imagem
  - Cálculo do Exo-esqueleto
  - Separar objectos que se tocam
  - Etc.
- Observações
  - Nos procedimentos seguintes, quando se menciona “**binarizar**”, assume-se também um processamento básico para eliminar pontos isolados ou outros elementos de ruído com operações na imagem antes e/ou depois do próprio acto de **binarizar**! (médias, medianas, pontos isolados, suavização, realce, etc.)

vsantos, 2006

## Operações Morfológicas – I

- Isolar objectos com buracos
  1. Binarizar imagem e guardar para usar como máscara
  2. Determinar o “esqueleto” não preservando os *pixels* terminais
  3. Eliminar pontos isolados do esqueleto (ficam os anéis)
  4. Propagar o restante esqueleto (os anéis) até à máscara obtida no ponto 1.



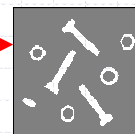
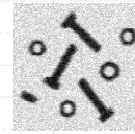
vsantos, 2006

## Operações Morfológicas - II

### ■ Preencher buracos em objectos

1. Binarizar
2. Complementar a imagem (negar)
3. Definir o bordo (caixilho) da imagem como semente (com a “cor” de objecto depois do ponto 1, ou seja, preto, neste exemplo)
4. Propagar do bordo (semente) até aos limites da imagem negada
5. Tomar a imagem propagada e invertê-la

Obs. Algumas aplicações de *software* têm já funções para fazer esta operação num passo único (como o caso do Matlab e que aliás não tem exactamente o operador de propagação como formulado aqui, i.e. , independente da cor a propagar!)

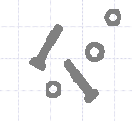
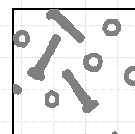
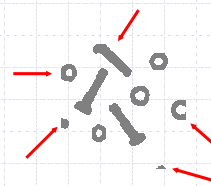


vsantos, 2006

## Operações Morfológicas - III

### ■ Remove objectos que tocam no bordo da imagem

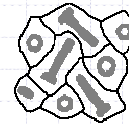
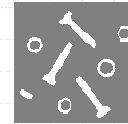
1. Binarizar (obter máscara)
2. Gerar um objecto semente com o bordo (caixilho) da imagem
3. Propagar a semente até à máscara (vai preencher os objectos ligados ao bordo com a “cor” oposta)
4. O que não foi propagado não está no bordo – é essa a imagem final, ou seja a imagem após a propagação!



vsantos, 2006

## Operações Morfológicas - IV

- Obtenção do Exo-esqueleto
  - Definição de exo-esqueleto: esqueleto do “fundo” que contém os objectos, “criando regiões”
  - 1. Binarização
  - 2. Complemento da imagem (negação)
  - 3. Cálculo do esqueleto pela metodologia que elimina as linhas terminais (condições  $C_1$  e  $C_2$  na definição da operação de *thinning*)



vsantos, 2006

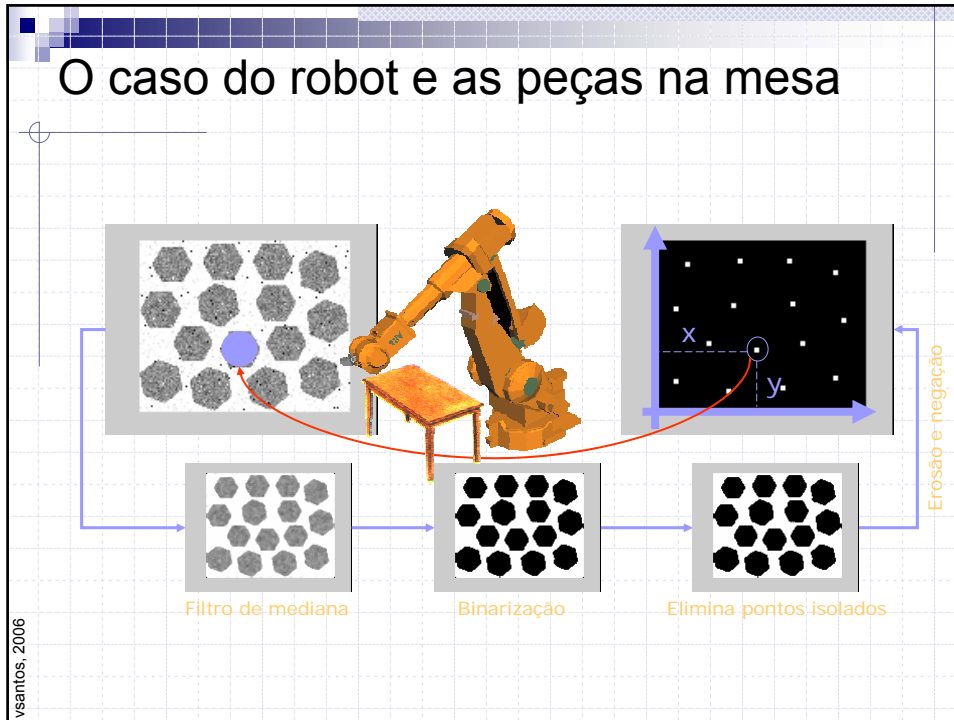
## Funções em MatLab (v5.3+)

- $BW2 = \text{BWMORPH}(BW1, \text{OPERATION}, N)$  aplica a operação  $N$  vezes.  $N$  can be Inf, in which case the operation is repeated until the image no longer changes.
  - OPERATION is a string that can have one of these values:
    - 'bothat' Subtract the input image from its closing
    - 'bridge' Bridge previously unconnected pixels
    - 'clean' Remove isolated pixels (1's surrounded by 0's)
    - 'close' Perform binary closure (dilation followed by erosion)
    - 'diag' Diagonal fill to eliminate 8-connectivity of background
    - 'dilate' Perform dilation using the structuring element ones(3)
    - 'erode' Perform erosion using the structuring element ones(3)
    - 'fill' Fill isolated interior pixels (0's surrounded by 1's)
    - 'hbreak' Remove H-connected pixels
    - 'majority' Set a pixel to 1 if five or more pixels in its 3-by-3 neighborhood are 1's
    - 'open' Perform binary opening (erosion followed by dilation)
    - 'remove' Set a pixel to 0 if its 4-connected neighbors are all 1's, thus leaving only boundary pixels
    - 'shrink' With  $N = \text{Inf}$ , shrink objects to points; shrink objects with holes to connected rings
    - 'skel' With  $N = \text{Inf}$ , remove pixels on the boundaries of objects without allowing objects to break apart
    - 'spur' Remove end points of lines without removing small objects completely.
    - 'thicken' With  $N = \text{Inf}$ , thicken objects by adding pixels to the exterior of objects without connected previously unconnected objects
    - 'thin' With  $N = \text{Inf}$ , remove pixels so that an object without holes shrinks to a minimally connected stroke, and an object with holes shrinks to a ring halfway between the holdand outer boundary
    - 'tophat' Subtract the opening from the input image
- **imfill()** ou **bwfill()** – preenchem pixels de fundo (valor "0" partindo de um ou mais pontos iniciais)
- **bwlabel()** – devolve uma divisão em regiões numeradas da imagem de objectos não ligados entre si.

vsantos, 2006



## O caso do robot e as peças na mesa



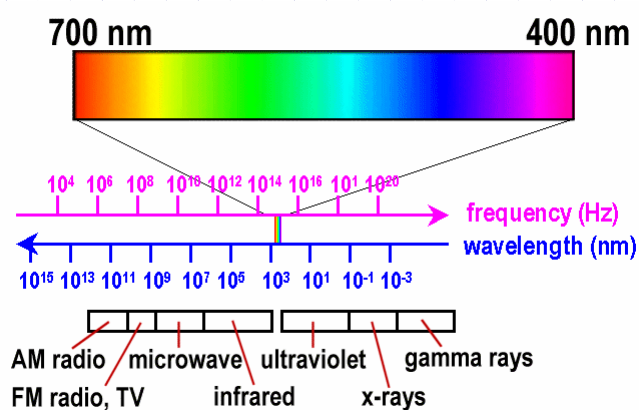
Uso da Cor em Imagens

## Origem da Cor

- Associada ao comprimento de onda da radiação electromagnética
- O olho humano possui sensores (cones) de três tipos:
  - Esses tipos apresentam sensibilidades diferentes ao comprimento de onda da radiação incidente:
    - R(ed), G(reen), B(lue)
    - Estas são as chamadas cores primárias
      - As suas combinações permitem obter outras cores

vsantos, 2006

## As cores no espectro visível



vsantos, 2006

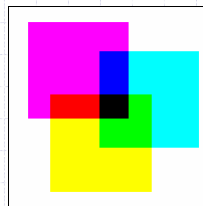
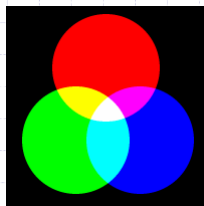
## Tarefas de visão industrial usando cor

- Industrialmente, a visão com cor aplica-se para resolver três categorias de aplicações:
  - *Verificação de cor* - Verificar se a cor de um determinado componente coincide com a que o sistema de visão está programado para encontrar
  - *Ordenação por cor* – Ordenar componentes baseadas na sua cor.
  - *Inspecção de cor* – Inspecionar componentes coloridos procurando defeitos que as técnicas baseadas em níveis de cinzento não conseguem detectar.

vsantos, 2006

## Formação das cores do espectro

- Há dois sistemas comuns para a formação de cores por “mistura” de cores primárias
  - O sistema “RGB” começa com preto e depois adiciona proporções de R, G e B. Proporções máximas destas cores resulta em branco.
  - O sistema “CMYK” começa com branco e depois adiciona proporções de Cyan, Magenta e Amarelo. Em teoria, as contribuições máximas de CMY resultam em preto. Nas impressoras reais isso não é feito por imperfeição das tintas; por isso, usa-se uma quarta cor de tinta para lidar com o preto (K).



vsantos, 2006

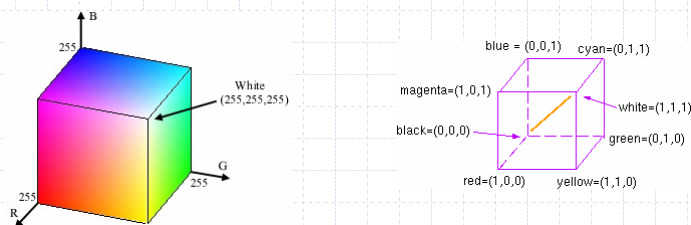
## Espaços de cor

- RGB (Red, Green, Blue)
  - sistema base do olho humano – usado em monitores e televisores – sistemas emissores de luz (ou aditivos)
- CMYK (Cyan, Magenta, Yellow, Black)
  - para impressão e impressoras - sistemas absorsores de luz (ou subtractivos)
- HSV (Hue (tom), Saturation (saturação), Value)
  - Adequado para segmentação de cor
- NTSC (Luminância, tom (Hue), Saturação)
  - sistema USA para vídeo digital
- YUV Luminância, (blue-yellow), (red-green)
  - PAL – vídeo
- YCbCr (Em vídeo Digital)
  - Luminância, Crominância (2 sinais):
    - Cb = (componente azul – valor de referência)
    - Cr = (componente vermelho – valor de referência)
- CIE (Commission Internationale d'Éclairage)
  - Intensidade (O "Value" do HSV) + 2 coordenadas ortogonais para cor

vsantos, 2006

## Representação dos espaços de cor

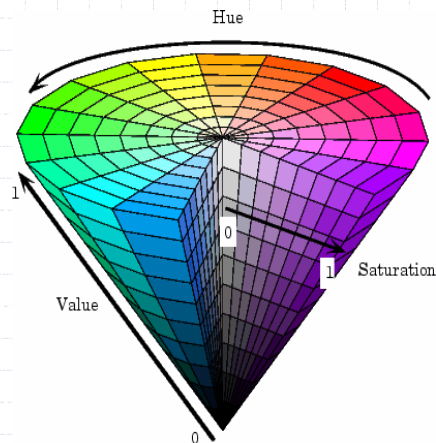
- Os espaços de cor são representados normalmente por entidades tridimensionais já que são 3 as componentes em qualquer um deles.
- Exemplo RGB – cubo de cores



vsantos, 2006

## O modelo HSV

- H(ue)
  - Tom ou tez
  - Exprime-se num valor angular (0-360°)
- S(aturação)
  - “pureza” da cor
  - Qualquer cor pouco “pura” aproxima-se de cinza
  - Varia de 0 a 1
- Valor (intensidade)
  - Varia de 0 a 1
  - Traduz “uma intensidade associada ao brilho”



vsantos, 2006

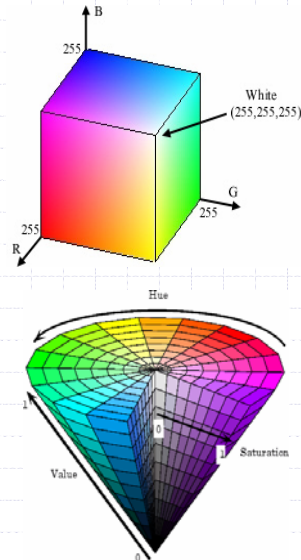
## Conversão de RGB para cinzentos

- Na conversão de RGB para níveis de cinzento...
  - ... poder-se-ia pensar numa média das três componentes, mas o olho humano é mais sensível ao verde. A proporção aproximada é:
    - $I = 0.30 R + 0.59 G + 0.11 B$
  - Em Matlab pode usar a função:
    - `RGB2GRAY()`

vsantos, 2006

## Conversão entre RGB e HSV

- Os valores RGB aparecem em duas variantes principais:
  - [0, 255] ou [0, 1]
- Os valores de HSV aparecem nas seguintes gamas:
  - H em graus [0, 360]
    - 0-60 vermelhos
    - 60-120 amarelos
    - 120-180 verdes
    - 180-240 cians
    - 240-300 azuis
    - 300-360 magentas
  - S e V na gama [0, 1]



vsantos, 2006

## Conversão de RGB para HSV

- V – o maior dos valores de RGB
  - $V = \max(R, G, B)$
- S – Gama de valores RGB em relação a V
  - $S = [\max(R, G, B) - \min(R, G, B)] / \max(R, G, B)$
  - $S = [V - \min(R, G, B)] / V$
- H – depende de RGB do seguinte modo:
  - Se  $R=G=B$ , H é indefinido (não há cor)
  - Caso contrário o processo depende de  $\max(R, G, B)$
- A literatura dá mais detalhes
- O Matlab dispõe da função `RGB2HSV()`

vsantos, 2006

## Conversão de HSV para RGB

- Processo em geral menos imediato que RGB→HSV
  - Se S=0 então R=G=B=V
  - Caso contrário:
    - Obtém-se a cor geral:  $C=H/60$
    - Obtém-se o termo  $F = (H-60xC)/60$
    - Obter  $P=V(1-S)$ ;  $Q=V(1-SF)$ ;  $R=V(1-S(1-F))$
    - Finalmente
      - $R= V+ P$
      - $G= V+ Q$
      - $B= V+ R$
  - No Matlab há a função:
    - `HSV2RGB()`

vsantos, 2006

## Conversão RGB-YUV

- No modelo YUV (sistema PAL de video) há a luminância e duas componentes de cor resultantes da diferença de componentes primárias
- O Matlab (v6.5) não tem funções específicas para a conversão, mas a seguinte relação é comum:

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.30 & 0.59 & 0.11 \\ -0.17 & -0.33 & 0.50 \\ 0.50 & -0.42 & -0.08 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

vsantos, 2006

## Conversão RGB-YIQ (NTSC)

- Também neste modelo há a luminância e duas componentes de cor resultantes da diferença de componentes
- Em Matlab há as funções:
  - NTSC2RGB() e RGB2NTSC()

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1.000 & 0.956 & 0.621 \\ 1.000 & -0.272 & -0.647 \\ 1.000 & -1.106 & 1.703 \end{bmatrix} \begin{bmatrix} Y \\ I \\ Q \end{bmatrix} \quad \begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.274 & -0.322 \\ 0.211 & -0.523 & 0.312 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

vsantos, 2006

## Conversão de YCbCr para RGB

- Formato comum em muitas webcams
- O Matlab dispõe de duas funções para as conversões:
  - YCBCR2RGB()
  - RGB2YCBCR()

vsantos, 2006



## Links

- <http://www.cs.nott.ac.uk/~tpp/G5AIV1/pdf/lecture08.pdf>
- <http://www.cs.nott.ac.uk/~tpp/G5AIV1/lectures.html>
- [http://dx.sheridan.com/advisor/cmyk\\_color.html](http://dx.sheridan.com/advisor/cmyk_color.html)
- <http://escience.anu.edu.au/lecture/cg/Color/index.en.html>
- <http://escience.anu.edu.au/lecture/cg/Color/printCG.en.html>