

Computação Gráfica

Técnicas de Segmentação -Limiarização (Thresholding)
(Aula 7)

Leonardo Medeiros

Instituto Federal de Alagoas

14 de Maio de 2017

Roteiro

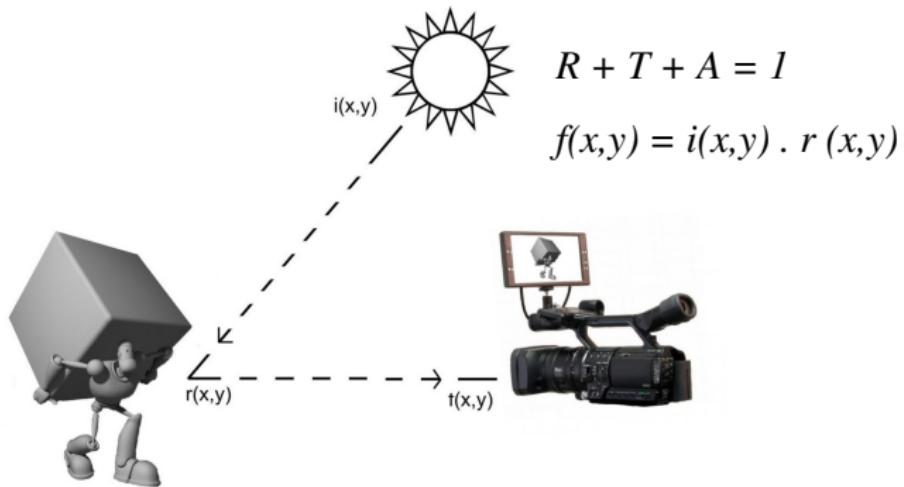
1 Segmentação de Imagem

2 Práticas

3 Limiarização Adaptativa

4 Bibliografia

Aquisição de Imagens



Aquisição da Imagem.

Amostragem e Quantização

- **Amostragem (*sampling*)** – refere-se ao número de pontos adquiridos (pixels) que são usados na imagem digitalizada (resolução espacial).
- **Quantização** - quantidade de tons que pode ser atribuído a cada ponto digitalizado (no. bits que representam cada pixel).

Imagens reais possuem um número **ilimitado** de cores ou tons. No processamento de **imagens computacionais** é necessário **limitar** os níveis de **cores ou tons** possíveis de serem atribuídos a cada *pixel* da imagem (tem uma **gradação tonal finita**).

Amostragem e Quantização

$$f(x,y) = z$$

Amostragem (*sampling*) -> número de pixels = $x \times y$.

Quantização -> (no. bits que representam z) .

Segmentação

Objetivo



Isolar **regiões da imagem** (conjunto de pontos) de interesse por serem pertencentes a objetos ou partes importantes para posterior análise (como extração de atributos e cálculo de parâmetros descritivos) .

Extração de Atributos ou Características

Objetivo



A partir de imagens já segmentadas (em objeto e fundo) ou binárias busca obter **dados relevantes** ou atributos, das regiões ou objetos destacados.

Os tipos de atributos ou características mais comuns são: número total de objetos; **dimensões; geometria; propriedades de cor, luminosidade e textura.**

Classificação e Reconhecimento

Distinguir objetos na imagem agrupando-os, classificando-os ou reconhecendo-os segundo parâmetros dependentes da aplicação

Os objetos são:

reconhecidos como pertencentes a um mesmo grupo e então sejam classificados de acordo com uma **base de dados ou imagens.**

apresentados para o sistema, que compara suas **características** com aquelas de classes previamente estabelecidas.

Decisão

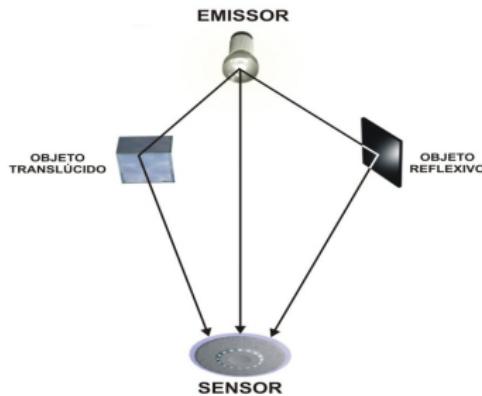
O objetivo de um sistema de VC é tomar decisões a partir da **extração de informações** do mundo real através de imagens.

A tomada de decisão pode ser feita a partir de **análises simples** (*menores distâncias, funções discriminantes, técnicas de clusterização, etc.*) ou de algoritmos mais complexos de **Inteligência Artificial** – IA (*SVM, redes neurais, fuzzy, etc.*) .

Imagen Digital

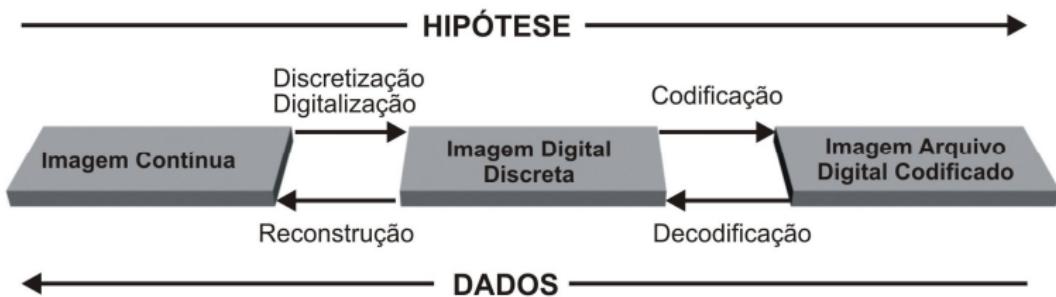
- Do latim *imago* - representação visual de um objeto.
- Em grego é chamada de *eidos* (raiz etimológica do termo *idea* ou *eide*) – considera a *idea* da coisa a sua imagem, seria uma projeção na mente.
- Imagen pode ser **adquirida** ou **gerada** pelo ser humano.
- Em sentido amplo uma imagen consiste em qualquer **forma visual de expressão de uma idéia**.

Formas de captação de uma imagem



Formas de Captação da Imagem por Radiação

Etapas do processamento de imagem digital



Discretização - conversão da imagem na forma contínua em um uma representação discreta.

Reconstrução - processo inverso da discretização.

Codificação - a partir da representação da imagem, gera-se um conjunto de dados representativos dela, estes que podem ser transformados em **arquivos**.

Decodificação - processo oposto à codificação no qual acessam-se informações codificadas para mostrá-la novamente como imagem.

Discretização e Reconstrução

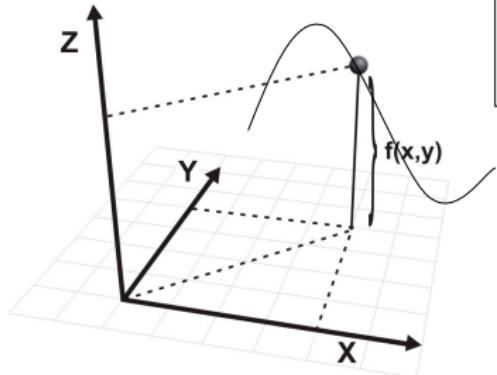
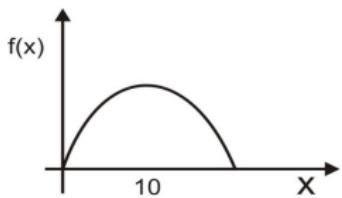


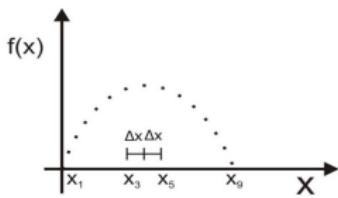
Gráfico de uma imagem contínua.

A forma de representar o mundo contínuo ou uma função contínua no computador é **discretizando-a**.

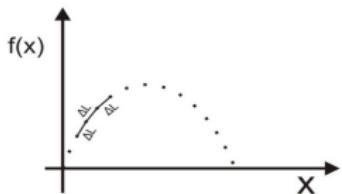
A operação que a partir dos valores discretos retorna uma aproximação da função contínua inicial é chamada de **reconstrução**.



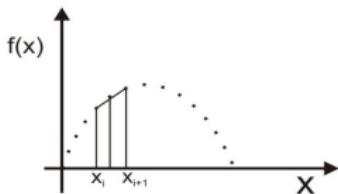
Representação Contínua



Representação Discreta a Intervalos Constantes em x



Amostragem a Intervalos Constantes em Relação a Distância entre os pontos $(x, f(x))$

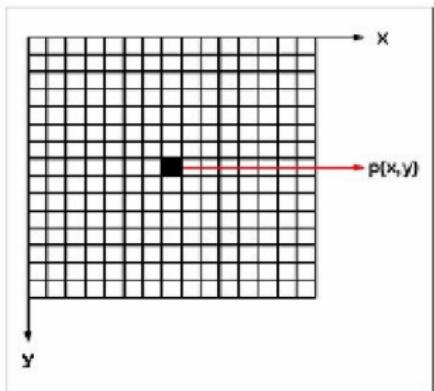


Reamostragem Criando Valores Intermediários por Interpolação Linear por Pares de Valores Anteriores

Formas de **Amostragem** e de **Re-amostragem** de uma função

$$y = f(x).$$

Amostragem e Quantificação



Reticulado uniforme da representação matricial da imagem.

Uma imagem digital é descrita por uma matriz NxM de valores de *pixel* ($p(x, y)$) inteiros positivos, que indica a intensidade de cor z em cada posição (x, y) da imagem.



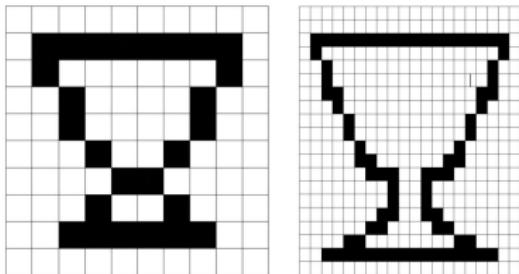
Um *pixel* é caracterizado pelo valor de **intensidade** de cor e pela sua **localização** na imagem.

47	52	64	132	153
51	58	121	149	142
49	99	143	144	164
94	135	161	170	199
138	165	180	212	213

Representação matricial de uma região da imagem.

Resolução Espacial

Ao ser digitalizada a imagem assume um **tamanho adimensional**, em *pixels*.



Mesma imagem em duas resoluções,
mas exibido no seu tamanho original.

Pode-se conhecer o tamanho da amostragem, conhecendo-se a razão entre o número de *pixels* obtido e um comprimento da imagem real.

A isso chama-se de resolução espacial, que em geral é medida **em pontos por polegada ou dpi (dots per inch)**.

Imagens reais - > Digitais

Para que sejam representadas no meio digital, seu formato **analógico** (contínuo) tem que ser convertido numa série de valores **discretos** (descontínuos).

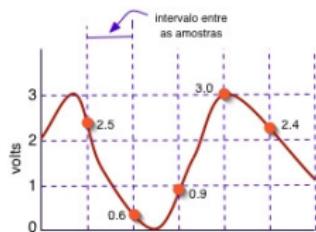
Esses valores são números (**dígitos**) que representam amostras (**samples**)

Amostragem

A conversão do **sinal analógico** para o digital é realizada por uma sequência de amostras da **variação de intensidade** do sinal original.

Cada amostra é arredondada para o número mais próximo da escala usada e depois convertida em um **número digital binário** (formado por "uns" e "zeros") para ser armazenado.

As amostras são medidas em **intervalos fixos**.

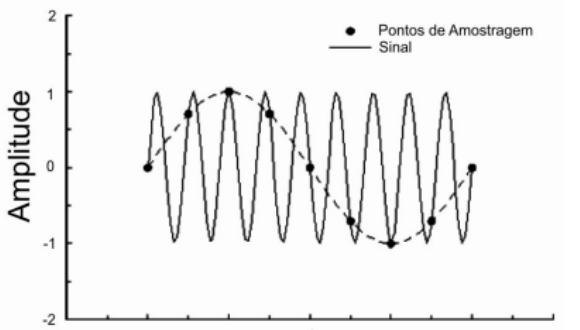


valores das amostras				
2.5	0.6	0.9	3.0	2.4
valores quantizados				
2	0	1	3	2
valores convertidos em dígitos binários				
1 0	0 0	0 1	1 1	1 0

O números de vezes em que se realiza a amostragem em uma unidade de tempo é a **taxa de amostragem**

Aliasing

Ocorre quando a **frequência de amostragem é inferior** à frequência de Nyquist.



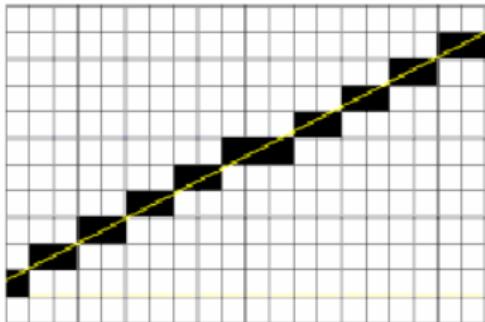
O sinal digitalizado fica completamente diferente do sinal original devido a sua **baixa frequência de amostragem**.

Teorema de Nyquist

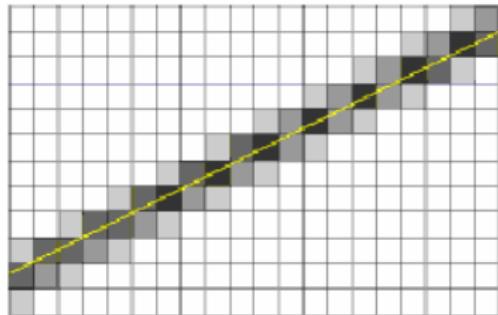
A **taxa de amostragem** dever ser pelo menos **duas vezes a maior** que a frequência que se **deseja registrar**.

Esse valor é conhecido como **frequência de Nyquist**.

Ao se tentar reproduzir uma frequência menor do que a frequência de Nyquist ocorre o fenômeno de **aliasing** (ou *foldover*)



“Dentes” na representação
de retas.



Uma forma de amenizar o
problema.

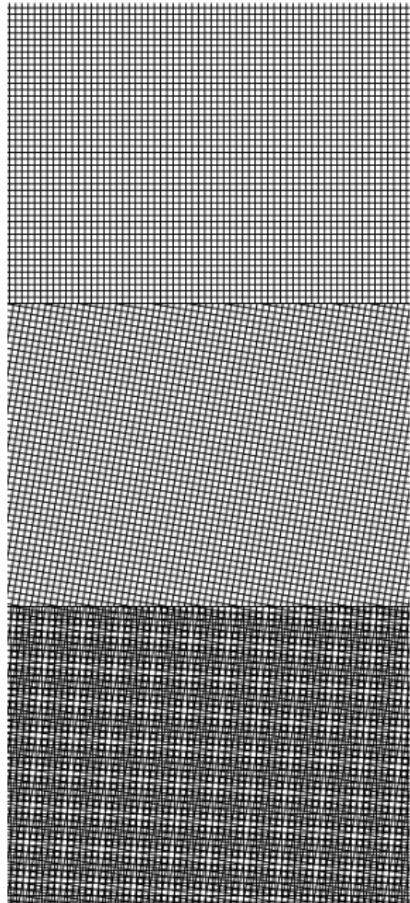
Problema do **aliasing** em descrição de baixa resolução.

Importante não confunda *aliasing* com *Moiré*

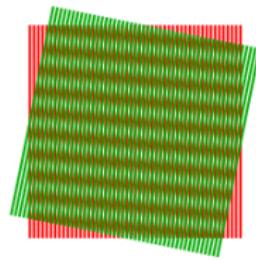
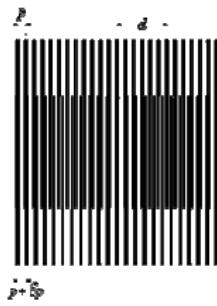
Moiré é um efeito de segunda ordem:

o corre quando se sobrepõem **duas resoluções diferentes** (escaneado em 200dpi e impresso em 300 dpi, por exemplo); ou há **grids não alinhados** (rotações ou aspect ratio)

(site: <http://www.ic.uff.br/~aconci/moire.html>)



Exemplificando a causa dos padrões texturais: *Moirés*

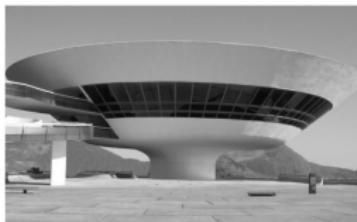


Moiré

Def.: é um padrão de **interferência** criado quando duas grades são **sobrepostas** em um determinado ângulo, ou quando elas têm pequenas diferenças nos tamanhos da malhas da grade.

Imagens Monocromáticas

Imagens monocromáticas são imagens digitais onde cada *pixel* possui apenas uma **banda espectral**.



Exemplos de imagens monocromáticas

O número de tons entre os valores limites, **branco** e **preto**, que se pode representar em tons, depende de quantos **bits** são alocados na matriz de imagem para armazenar o tom de cada *pixel*.

Número de elementos na Escala de cinza	Tons de cinza limites	Números de Bits necessários para representação do <i>pixels</i>
2^1 2 valores	0,1	1
2^3 8 valores	0 a 7	3
2^4 16 valores	0 a 15	4
2^8 256 valores	0 a 255	8

Uma imagem **monocromática** pode ser representada geometricamente também por **valores reais** quanto à posição dos *pixels* como no gráfico $G(f)$ da função f :

$$G(f) = \{ (x,y,z); (x,y) \in R^2 ; z = f(x,y) \} \quad (3.4)$$

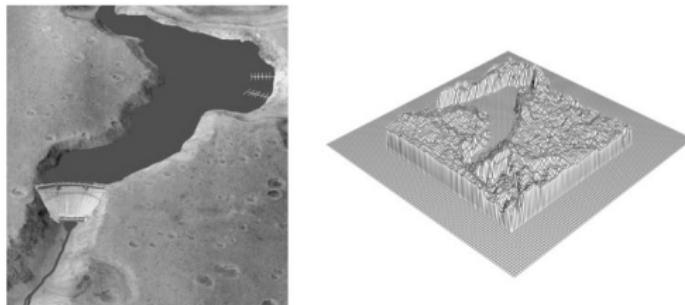


Imagen Monocromática de uma represa e seu gráfico 3D na forma de sua função $G(f)$ no R^3 .

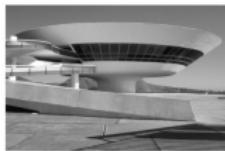
Imagens Coloridas

Imagens multibandas são imagens digitais onde cada *pixel* possui *n* bandas espectrais (*n*>1).

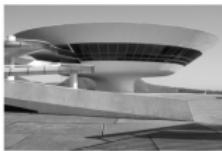
Quando uma imagem é representada pela composição de **três bandas visíveis (como as RGB)** tem-se uma **imagem colorida aos olhos humanos**.



(a) Imagem Colorida



(b) Banda Vermelha (Red)



(c) Banda Verde (Green)

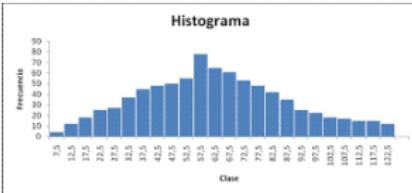


(d) Banda Azul (Blue)

Imagen colorida e cada uma de suas bandas RGB.

histograma,

também conhecido como distribuição de frequências ou diagrama das frequências, é a representação gráfica, em colunas (retângulos), de um conjunto de dados previamente tabulado e dividido em classes uniformes.



Momentos do histograma

Seja l o brilho de uma imagem (ou objeto) e $h(l)$, $l = 0, 1, \dots, L - 1$, o seu histograma normalizado. O n -ésimo momento central é dado por:

$$\mu_n = \sum_{l=0}^{L-1} (l - \mu)^n h(l)$$

$$\mu = \sum_{l=0}^{L-1} lh(l)$$

onde μ é o brilho médio.

Momentos do histograma

- Cont.

Quando $n = 2$, nós temos a variância (contraste) $\mu_2 = \sigma^2$.

Os momentos μ_3 e μ_4 medem *skewness* (assimetria em relação ao centro) e *kurtosis* (concentração em torno da média). Podemos usar, por exemplo, $R = 1 - \frac{1}{1+\sigma^2}$ como medida de contraste. Momentos de mais alta ordem não estão relacionados com a forma do histograma.

Textura: Momentos do Histograma

- Uma das abordagens mais simples para a descrição da textura é através dos momentos do histograma de níveis de cinza de uma região.
- Seja Z uma variável aleatória denotando a intensidade discreta de uma imagem
- Seja $p(z_i)$, $i = 1, 2, \dots, L$, a distribuição de probabilidade associada a esta variável, na qual L é o número de níveis de cinza.

Textura: Momentos do Histograma

- O n-ésimo momento de z em torno da média é dado por

$$\mu_n(z) = \sum_{i=1}^L (z_i - m)^n p(z_i)$$

- na qual m é valor médio de z

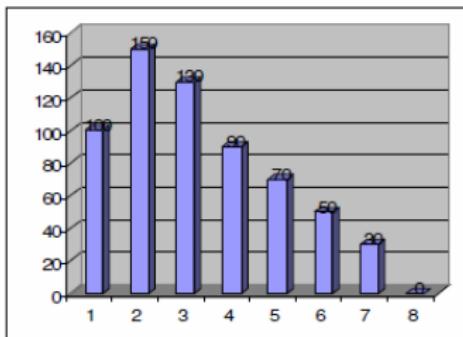
$$m = \sum_{i=1}^L z_i \times p(z_i)$$

Textura: Momentos do Histograma

- O segundo momento possui uma importância particular para a descrição da textura
 - Medida de contraste do nível de cinza
 - Pode ser usada no estabelecimento de descritores de suavidade relativa.
- O terceiro momento é uma medida de anti-simetria do histograma.
- Quatro momento fornece uma medida de achatamento.

Textura: Momentos do Histograma

- Exercício: Considere o seguinte histograma em 8 níveis de cinza.



Z	P(z)
1	0.16
2	0.24
3	0.20
4	0.14
5	0.11
6	0.08
7	0.05
8	0

Calcule o segundo e o terceiro momentos.

$$M=102, M^2 = 1274$$

Histograma de imagem digital

O histograma de uma imagem indica o **número ou o percentual** de *pixels* que a imagem tem em determinado nível de cinza ou cor.

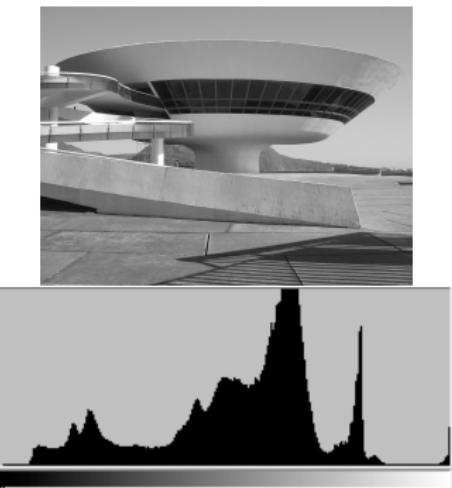
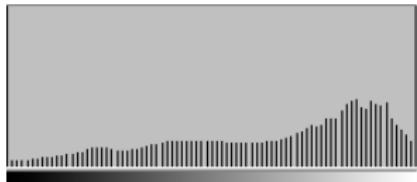
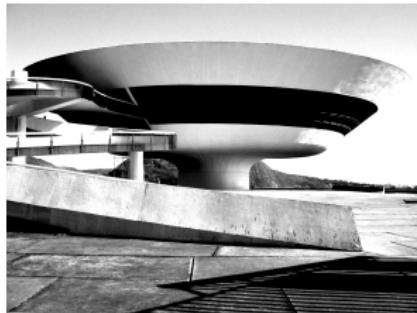
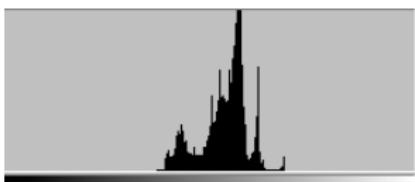
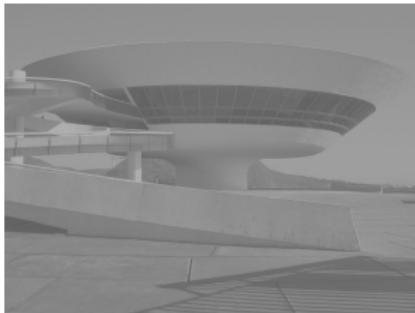
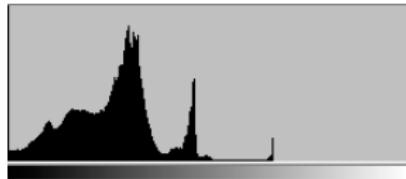
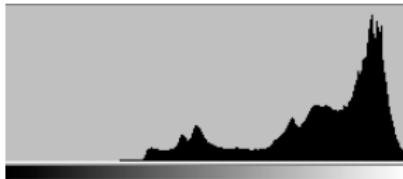
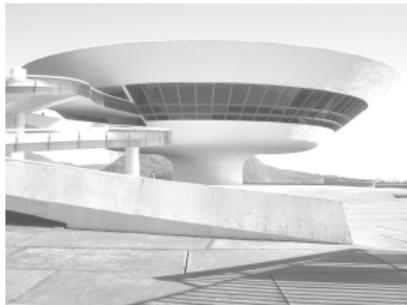


Imagen em tons de cinza e o seu histograma.

O histograma fornece uma indicação da **qualidade da imagem** quanto ao contraste e intensidade luminosa.

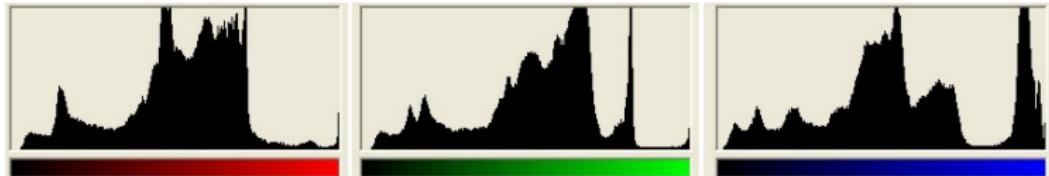
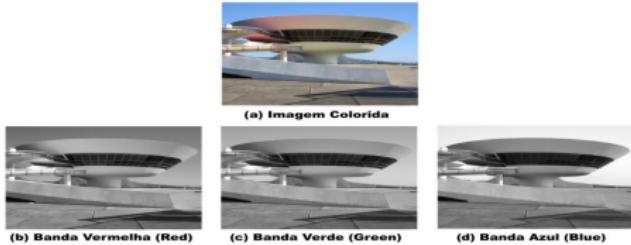


Imagens em tons de cinza e seus respectivos histogramas: com pouco (direita) e bom **contraste** (esquerda).



Imagens em tons de cinza e seus respectivos histogramas: com **alta luminosidade** (clara) e com **baixa luminosidade** (escura).

Histogramas das **bandas** de uma imagem colorida



Histogramas da Imagem colorida separado por canal R,G B.

Sistemas de visão binária

Limiarização (Thresholding):

Manual (baseada no histograma)

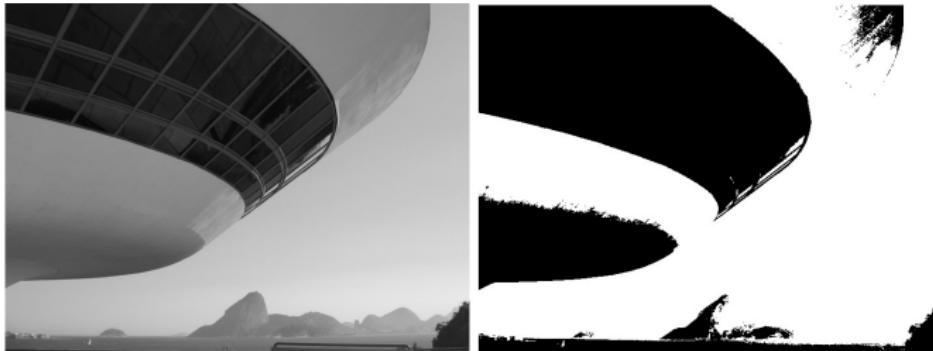
Automática:

Método de Otsu (*Thresholding Global*)

Método Iterativo

outros (ver site do curso)

Agrupamento por limiar (Thresholding):



Exemplo de binarização:

(a) Imagem em tons de cinza,

(b) Imagem binária

Agrupamento por limiar (limiarização)

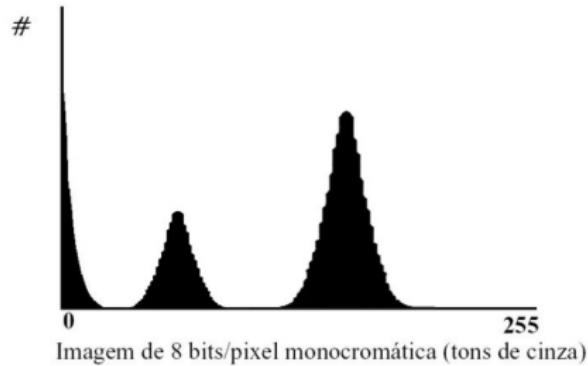
Uso: segmentação em imagens em que o objeto a ser segmentado apresenta uma tonalidade **bem diferente do fundo da imagem.**

Em um **histograma bimodal**, é possível estabelecer um limiar entre as duas tonalidades.



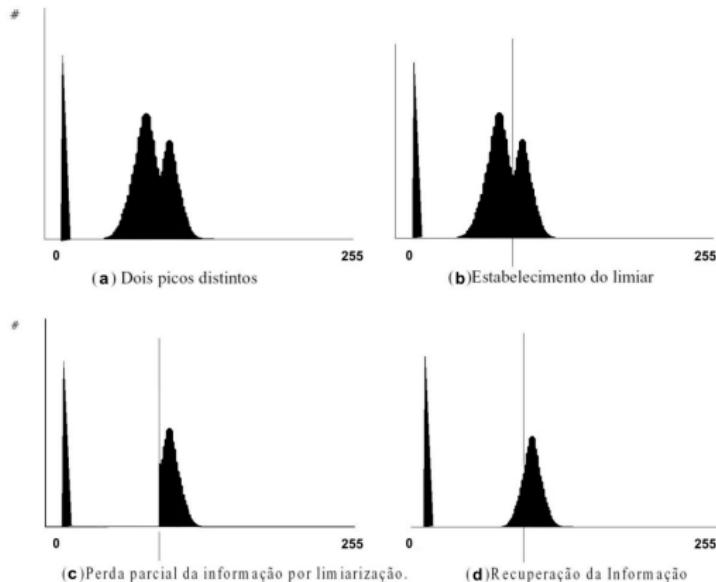
Exemplo de histograma bem-divisível.

Quando a imagem tem mais de dois objetos com cinzas diferentes em um fundo mais escuro, pode ser usada a técnica de **limiarização multinível** (*multilevel thresholding*).



Exemplo de um histograma trimodal.

Há casos em que torna-se necessário o uso de técnicas mais complexas



Limiarização por interpolação (Chow e Kaneko, 1972).

A limiarização **converte** uma imagem de entrada:

$f(x, y)$ de N níveis de cinza

em uma imagem $g(x,y)$, chamada de imagem limiarizada (ou **posterizada**), com número de níveis de cinza menor do que N .

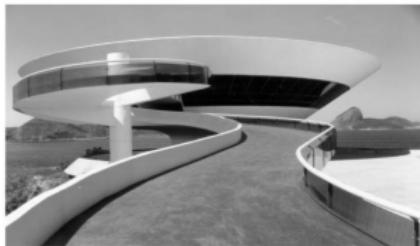
No limite, $g(x, y)$, terá só dois níveis de cinza, como na equação:

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

De uma maneira mais genérica, a definição de T é a função de **várias variáveis** na forma:

$$T = T [x, y, p(x,y), f(x,y)]$$

onde $p(x,y)$ é uma propriedade local , isto é função de x,y .



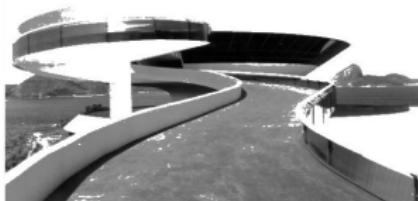
(a) Imagem com 256 tons de cinza



(b) Histograma



(c) Limiarização com valor 80



(d) Limiarização com valor 150

Influência do valor do limiar sobre a qualidade da limiarização.

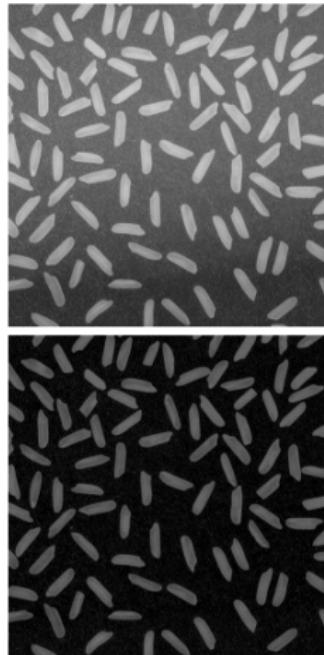
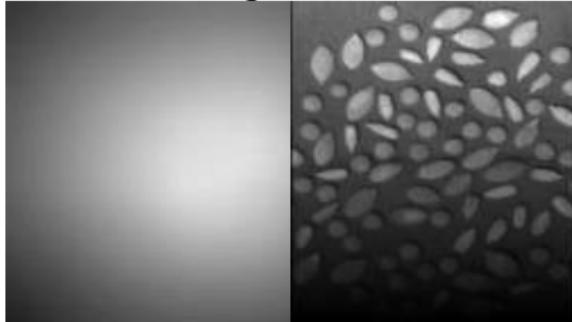
(A imagem (d) está posterizada nesta representação).

É uma técnica muito boa para separar elementos de uma imagem

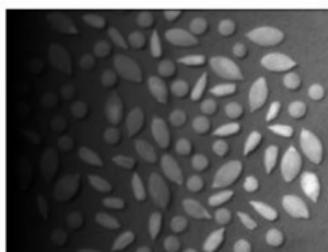
- Desde que a iluminação seja adequada
- E
- Os elementos sejam diferenciados por tons

Uneven illumination

- Subtração do fundo

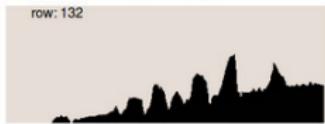


Subtração fundo



Input image.

row: 132



Profile of the row 132.

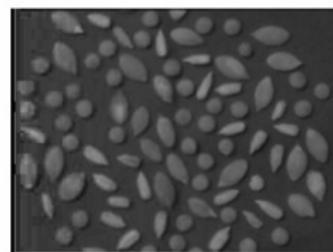


Background image.

row: 132

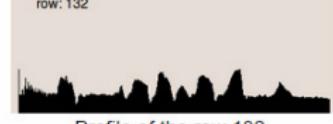


Profile of the row 132.



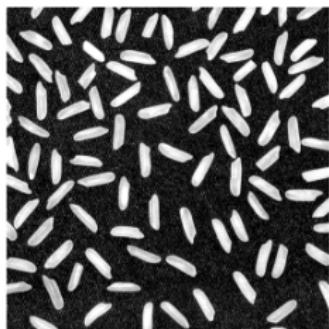
Output image.

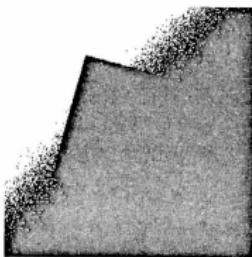
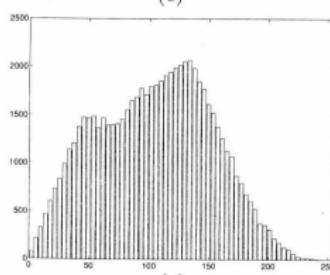
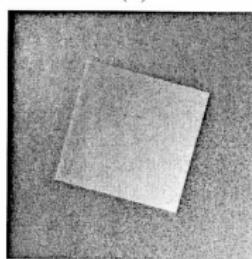
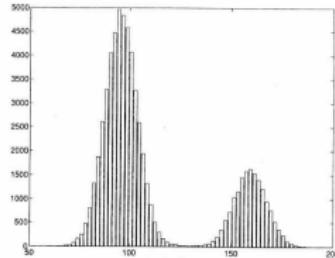
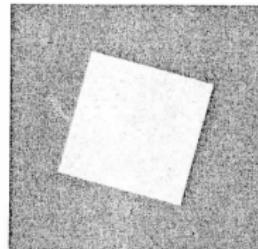
row: 132

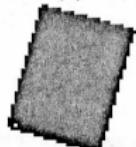
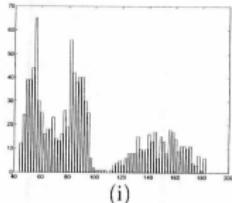
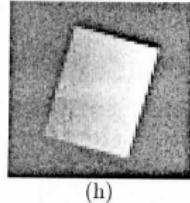
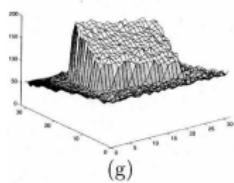
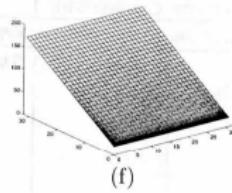
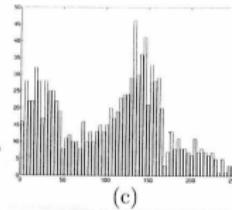
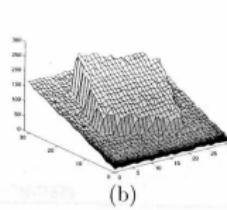
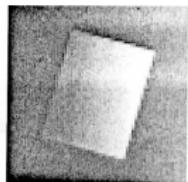


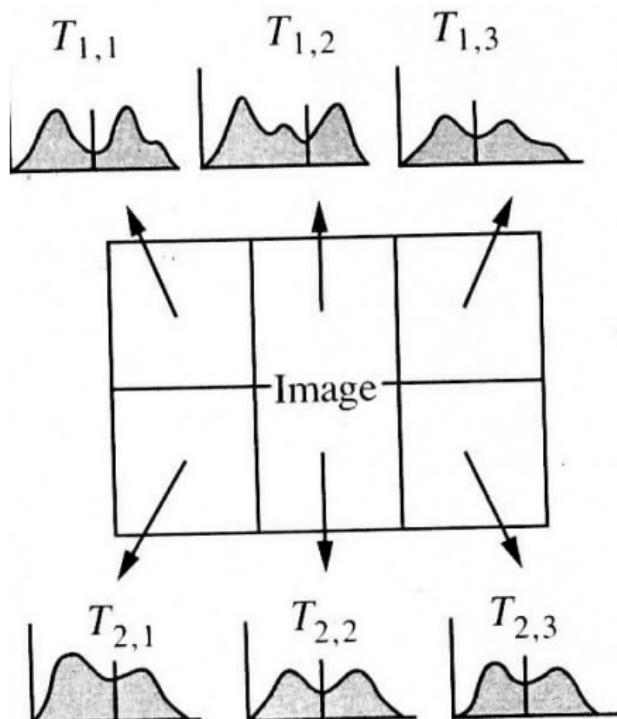
Profile of the row 132.

Aumento do contraste x limiarização







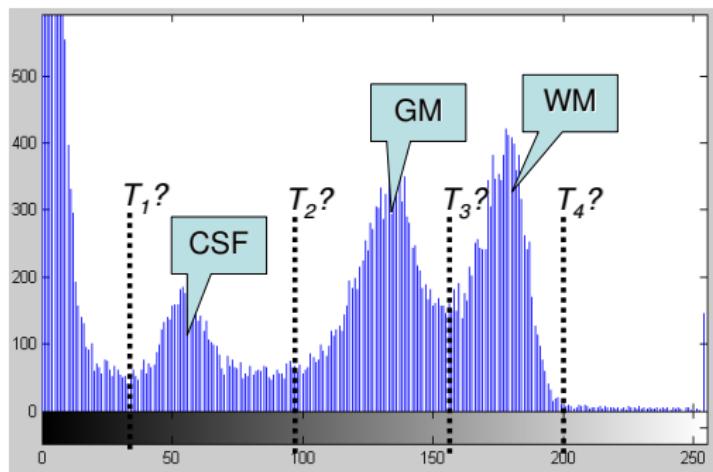


Experimentação

Seção RM do Crâneo



Histograma



Estruturas do encéfalo:

CSF: líquor

GM: subst. cinzenta

WM: subst. branca

$T_n?$ → Nível teórico de
limiarização de cada
estrutura

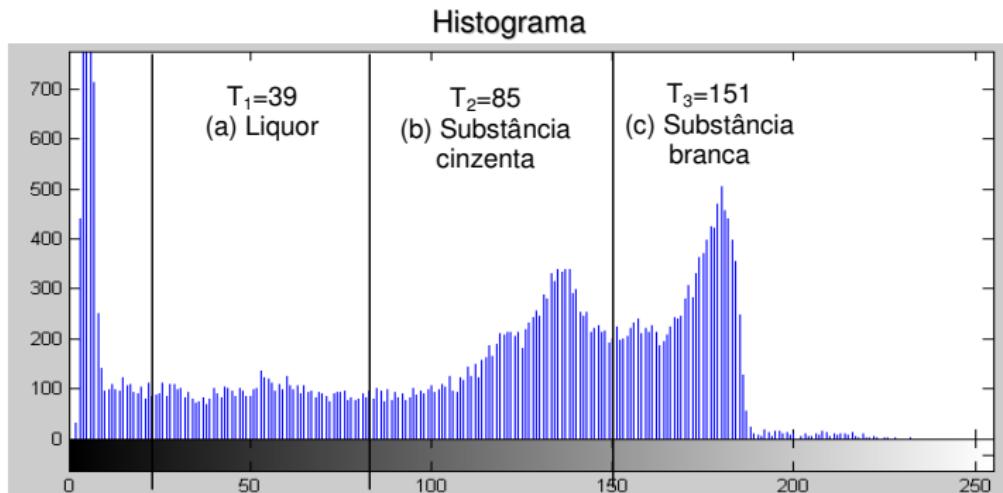
Fonte: Projeto *Brainweb*

<http://brainweb.bic.mni.mcgill.ca/brainweb/>

Segmentação utilizando método de Otsu:

Considerações:

Três níveis de limiarização T_1 , T_2 e T_3



comparação dos Resultados

Imagens binárias geradas pela segmentação por limiarização, utilizando os níveis encontrados pelo método de Otsu:



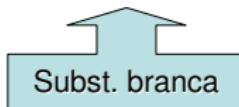
(a)



(b)

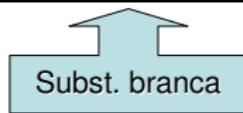
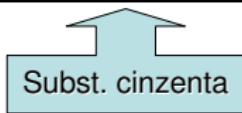


(c)



Unir as áreas para calcular volumes

Considerando a conectividade das regiões:



Imagens binárias geradas pela segmentação por limiarização, utilizando os níveis encontrados pelo método de Limiarização em Multiníveis de Niblack:

Os resultados encontrados através da aplicação do algoritmo de Niblack para seleção dos limites de cada região (segmentação) foram: $T_1 = 39$, $T_2 = 77$, $T_3 = 149$.



Líquor



Subst. cinzenta



Subst. branca

Limiarização Automática (*Thresholding*) baseada no histograma

O método mais simples (e mais caro computacionalmente) de calcular automaticamente o ***threshold*** é o **método iterativo.**

Este método é descrito por:

1. **Selecione** um tom T_i . inicial (entre o valor da intensidade mínima e máxima da imagem.)
2. **Segmenta** a imagem usando T_i . Isto é, divida-a em **dois grupos**: (G1) formado com os pixels cujos valores são $> T$ e (G2) com os pixels com tons $< T$.
3. Calcule a **intensidade média** dos grupos: μ_1 e μ_2
4. Calcule o **novo threshold**: $T_{i+1} = \frac{1}{2} (\mu_1 + \mu_2)$
5. Repita os passos **2 até 4** até que $T_{i+1} - T_i < (\text{parâmetro pré-definido})$

Vamos Brincar um Pouco...

Limiarização Simples

Primeiramente, vamos aprender como usar os mecanismos de Limiarização Simples (*Simple Threshold*).

Limiarização Simples

Aqui, vamos nos preocupar com o mecanismo de Limiarização simples. Iremos abrir uma imagem contendo um gradiente na escala de cinza. Caso os valores dos pixels em cinza mais próximo ao preto iremos atribuir o valor 0 (preto) e 1 (branco) em caso contrário.

Tipos de Limiarização

O OpenCV provê diferentes tipos de limiarização como:

- `cv2.THRESH_BINARY`
- `cv2.THRESH_BINARY_INV`
- `cv2.THRESH_TRUNC`
- `cv2.THRESH_TOZERO`
- `cv2.THRESH_TOZERO_INV`

Limiarização Binária

```
import cv2
import numpy as np

# Carrega a imagem na escala cinza
image = cv2.imread('../img/gradient.jpg',0)
cv2.imshow('Original', image)

# Valores abaixo de 127 irão para 0 (preto) o resto 1 (branco)
ret,thresh1 = cv2.threshold(image, 127, 255, cv2.THRESH_BINARY)
cv2.imshow('1-Limiarizacao-Binaria', thresh1)

cv2.imwrite('../img/limiarizacaobinaria.jpg', thresh1)

cv2.waitKey(0)
cv2.destroyAllWindows()
```

Limiarização Binária



Limiarização Binária Invertida

```
import cv2
import numpy as np

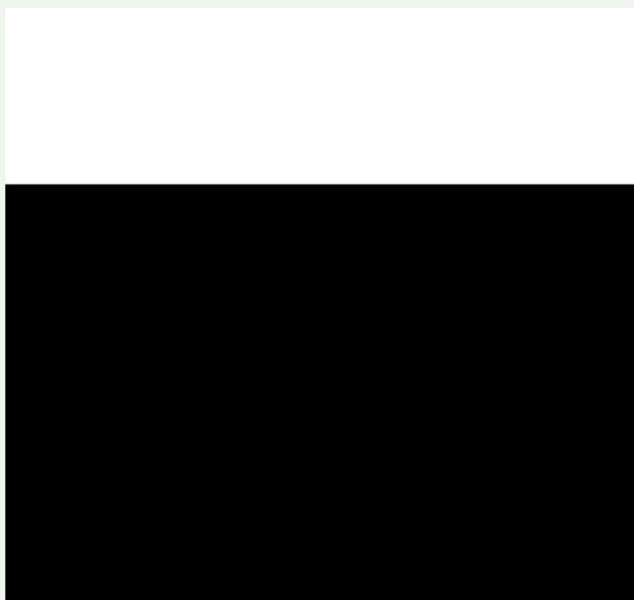
# Carrega a imagem na escala cinza
image = cv2.imread('../img/gradient.jpg',0)
cv2.imshow('Original', image)

# Valores abaixo de 127 irão para 1 (branco) o resto 0
# (preto)
ret,thresh2 = cv2.threshold(image, 127, 255, cv2.
    THRESH_BINARY_INV)
cv2.imshow('1-Limiarizacao_Binaria_Invertida', thresh2)

cv2.imwrite('../img/limiarizacaobinariainv.jpg',
    thresh2)

cv2.waitKey(0)
cv2.destroyAllWindows()
```

Limiarização Binária Invertida



Limiarização Binária Truncada

```
import cv2
import numpy as np

# Carrega a imagem na escala cinza
image = cv2.imread('../img/gradient.jpg',0)
cv2.imshow('Original', image)

# Valores acima de 127 são truncados e abaixo ficam intactos
ret,thresh3 = cv2.threshold(image, 127, 255, cv2.
    THRESH_TRUNC)
cv2.imshow('3-THRESH-TRUNC', thresh3)

cv2.imwrite('../img/limiarizacaobinariatrunc.jpg',
    thresh3)

cv2.waitKey(0)
cv2.destroyAllWindows()
```

Limiarização Binária Truncada



Limiarização Binária TreshtoZero

```
import cv2
import numpy as np

# Carrega a imagem na escala cinza
image = cv2.imread('../img/gradient.jpg',0)
cv2.imshow('Original', image)

# Valores abaixo de 127 vao a 0, acima de 127 ficam intactos
ret,thresh4 = cv2.threshold(image, 127, 255, cv2.THRESH_TOZERO)
cv2.imshow('4-THRESH_TOZERO', thresh4)

cv2.imwrite('../img/limiarizacaobinariatreshtozero.jpg', thresh4)

cv2.waitKey(0)
cv2.destroyAllWindows()
```

Limiarização Binária TreshToZero



Limiarização Binária TreshtoZero Invertida

```
import cv2
import numpy as np

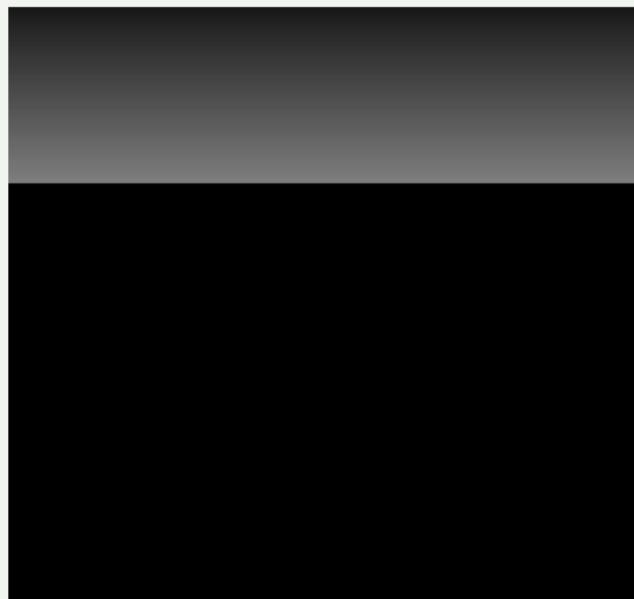
# Carrega a imagem na escala cinza
image = cv2.imread('../img/gradient.jpg',0)
cv2.imshow('Original', image)

# Valores abaixo de 127 ficam intactos, acima de 127
# vao a 0
ret,thresh5 = cv2.threshold(image, 127, 255, cv2.
    THRESH_TOZERO_INV)
cv2.imshow('5 - THRESH_TOZERO_INVERTED', thresh5)

cv2.imwrite('../img/limiarizacaobinariatreshtozeroinv.
    jpg', thresh5)

cv2.waitKey(0)
cv2.destroyAllWindows()
```

Limiarização Binária TreshToZero Invertida



Limiarização Adaptativa

Como vimos na Limiarização Simples, podemos utilizar um valor global para limiarizar minhas imagens. No entanto, esta prática pode não ser eficiente em diferentes condições de luzes e em diferentes regiões. Nestes casos, é interessante usarmos uma Limiarização Adaptativa para calcularmos os limiares para regiões pequenas de uma imagem ou com diferentes condições de luz.

Limiarização Adaptativa no OpenCV

Para usarmos a Limiarização Adaptativa no OpenCV precisamos de três parâmetros:

- Escolha do Método de Limiarização Adaptativa utilizado:
`cv2.ADAPTIVE_THRESH_MEAN_C` : valor do limiar é a média das regiões vizinhas.
`cv2.ADAPTIVE_THRESH_GAUSSIAN_C` : valor do limiar é o peso da soma dos valores vizinhos onde o peso é uma janela de gaussiana (isto será explicado no futuro quando falarmos de filtros de imagem).
- O parâmetro *Block Size* define qual a área das regiões vizinhas.
- O parâmetro *C* é a constante a ser subtraída da média das regiões vizinhas.

Limiarização Simples Binária

```
import cv2
import numpy as np

# Carregar nossa imagem
image = cv2.imread('../img/Origin_of_Species.jpg', 0)

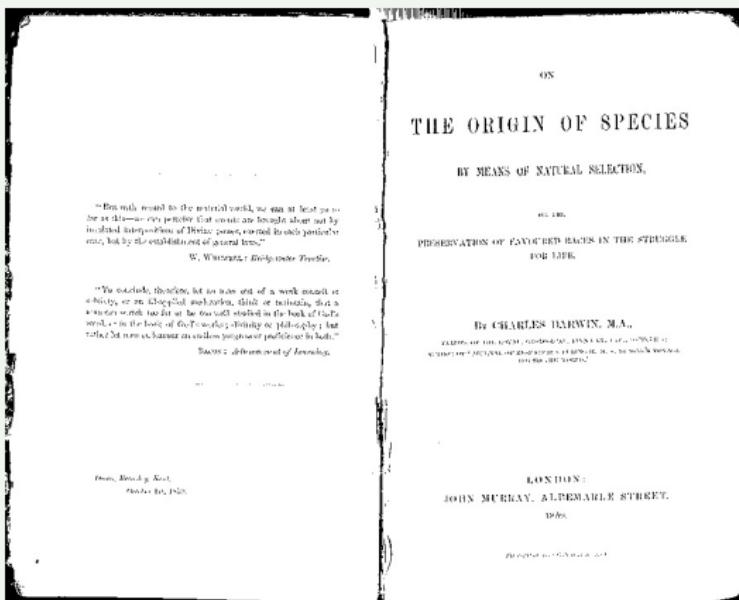
cv2.imshow('Original', image)
cv2.waitKey(0)

ret, thresh1 = cv2.threshold(image, 127, 255, cv2.
    THRESH_BINARY)
cv2.imshow('Threshold_Binary', thresh1)
cv2.imwrite('../img/limiarizacaobinariaorigin.jpg',
    thresh1)
cv2.waitKey(0)

cv2.destroyAllWindows()
```

Limiarização Simples Binária - Teste

Limiarizacao Simples Binaria - Resultado



Limiarização Adaptativa

```
import cv2
import numpy as np

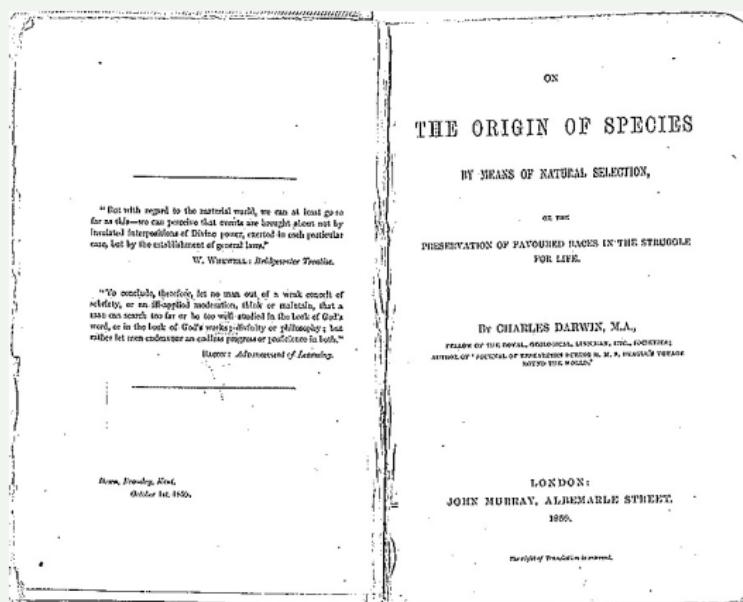
# Carregar nossa imagem
image = cv2.imread('../img/Origin_of_Species.jpg', 0)
cv2.imshow('Original', image)
cv2.waitKey(0)

# Filtro de blur para remover o ruido da imagem
image = cv2.GaussianBlur(image, (3, 3), 0)
thresh = cv2.adaptiveThreshold(image, 255,
    cv2.ADAPTIVE_THRESH_MEAN_C, cv2.THRESH_BINARY, 3, 5)
cv2.imshow("Adaptive_Mean_Thresholding", thresh)
cv2.imwrite('../img/limiarizacaoadaptativa.jpg', thresh)
cv2.waitKey(0)

cv2.destroyAllWindows()
```

Limiarização Adaptativa

Limiarizacao Adaptativa Média - Resultado



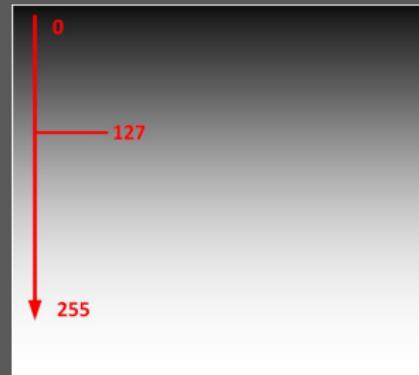
Thresholding, Binarization & Adaptive Thresholding

Thresholding is act of converting an image to a binary form.

```
cv2.threshold(image, Threshold Value, Max Value, Threshold Type)
```

Threshold Types:

- cv2.THRESH_BINARY – Most common
- cv2.THRESH_BINARY_INV – Most common
- cv2.THRESH_TRUNC
- cv2.THRESH_TOZERO
- cv2.THRESH_TOZERO_INV



NOTE: Image need to be converted to greyscale before thresholding.

Adaptive Thresholding

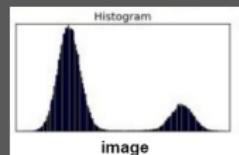
- Simple threshold methods require us to provide the threshold value.
- Adaptive threshold methods take that uncertainty away

```
cv2.adaptiveThreshold(image, Max Value, Adaptive type, Threshold Type, Block size, Constant that is subtracted from mean)
```

NOTE: Block sizes need to be odd numbers!

Adaptive Threshold Types:

- **ADAPTIVE_THRESH_MEAN_C** – based on mean of the neighborhood of pixels
- **ADAPTIVE_THRESH_GAUSSIAN_C** – weighted sum of neighborhood pixels under the Gaussian window
- **THRESH_OTSU** ([uses cv2.threshold function](#)) – Clever algorithm assumes there are two peaks in the gray scale histogram of the image and then tries to find an optimal value to separate these two peaks to find T.



Método de Otsu

A operação de limiarização é considerada como sendo o particionamento da imagem com L níveis de cinza em duas classes C_0 e C_1 , que representam o **objeto** e o fundo, ou vice-versa, sendo que esta partição se dará no **nível de cinza t** , que maximiza a razão da variância entre as classes e a variância total

C_0 = pixels da imagem com tons de 0 a t

C_1 = pixels da imagem com tons de t a l

Método de Otsu para escolher o melhor *threshold*: T .

Se baseia no **histograma normalizado** (da imagem), que pode ser visto como uma função de **densidade de probabilidade discreta** (**histograma normalizado**)

$$p_r(r_q) = \frac{n_q}{n}, q = 0, 1, 2, \dots, L - 1$$

onde n é o número total de píxeis na imagem, n_q é o número de píxeis que tem intensidade r_q

O **método de Otsu** escolhe o threshold de valor T que maximiza:

$$\sigma_B^2 = \omega_0(\mu_0 - \mu_T)^2 + \omega_1(\mu_1 - \mu_T)^2$$

$$\sigma_B^2 = \omega_0(\mu_0 - \mu_T)^2 + \omega_1(\mu_1 - \mu_T)^2$$

$$\text{onde,} \quad \mathcal{W}_0 = \sum_{q=0}^{k-1} p_q(r_q)$$

$$\mathcal{W}_0 = \sum_{q=k}^{L-1} p_q(r_q)$$

$$m_I = \sum_{q=k}^{L-1} qp_q(r_q)/\omega_1 \quad m_0 = \sum_{q=k}^{L-1} p_q(r_q)$$

$$m_T = \sum_{q=0}^{L-1} qp_q(r_q)$$

Limiarização Adaptativa Otsu

```
import cv2
import numpy as np

# Load our new image
image = cv2.imread('../img/Origin_of_Species.jpg', 0)

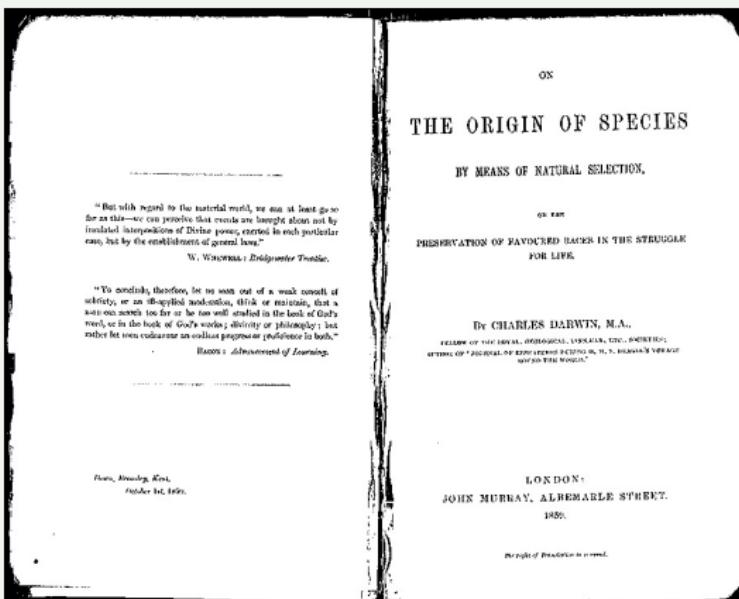
cv2.imshow('Original', image)
cv2.waitKey(0)

_, th2 = cv2.threshold(image, 0, 255, cv2.THRESH_BINARY
+ cv2.THRESH_OTSU)
cv2.imshow("Otsu's Thresholding", th2)
cv2.imwrite('../img/limiarizacaoadaptativaotsu.jpg',
th2)
cv2.waitKey(0)

cv2.destroyAllWindows()
```

Limiarização Adaptativa Otsu

Limiarizacao Adaptativa Otsu - Resultado



Bibliografia

- Solomon C., Breckon T. **Fundamentals of Digital Image Processing.. A Practical Approach with Examples in Matlab** (Wiley-Blackwell, 2011)(ISBN 9780470844724)(en)(355s)
- A. Conci, E. Azevedo e F.R. Leta - **Computação Gráfica: volume 2 (Processamento e Análise de Imagens Digitais)**, Campus/Elsevier. 2008 - ISBN 85-352-1253-3.
- Open CV Tutorials
- Tutorial de ColorSpace
http://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_thresholding/py_thresholding.html#simple-thresholding

DÚVIDAS ?