

# TITULO A DEFINIR

Danilo Henrique Costa Souza

Universidade Federal do Pará  
Instituto de Tecnologia

Faculdade de Engenharia da Computação e Telecomunicações  
Orientador: Professor Ronaldo de Freitas Zampolo

25 de novembro de 2015

## 1 Introdução

## 2 A técnica estudada

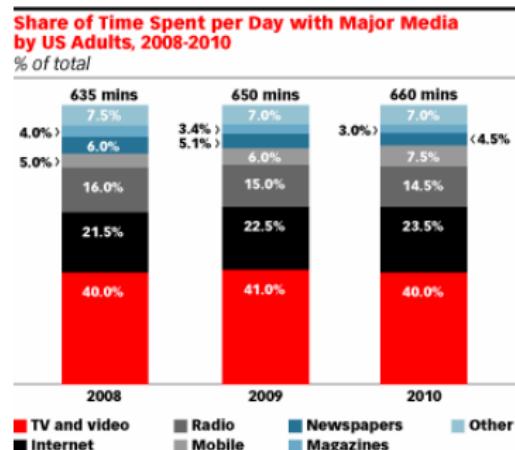
## 3 Metodologia utilizada

## 4 Resultados

## 5 Considerações finais

# Processamento Digital de Imagens (Crescimento)

- Dispositivos mais acessíveis.
- Interação entre pessoas mais digital.
- Quantidade de conteúdo multimídia.
- O que fazer com esse conteúdo?



Note: time spent with each medium includes all time spent with that medium, regardless of multitasking; for example, 1 hour of multitasking on the internet and watching TV was counted as 1 hour for TV and 1 hour for internet  
 Source: eMarketer, Dec 2010

122845

www.eMarketer.com

**Figura:** Consumo de conteúdo multimídia por adultos nos EUA no período 2008-2010

## Processamento Digital de Imagens (Importância)

- Por que estudar?
- Definições
  - *Pixels.*
  - *Texels.*
  - Textura.
- 3 níveis de processos (Gonzalez et. al):
  - Baixo - operações básicas.
  - Médio - segmentação e classificação.
  - Alto - semântica, extração de informação.

## Aplicações

- Presença em diversas áreas
  - Segurança: fiscalização de velocidade, detecção de movimentos).
  - Automação industrial: classificação de produtos, verificação de falhas).
  - Entretenimento: em redes sociais, reconhecimento facial e edição de fotos).
- Facilitar classificação de imagens

## Segmentação de Imagens - Definição

### Definição (Gonzales et. al)

Consiste em dividir uma imagem em sub-imagens formadas pelos objetos/regiões da imagem original. A divisão não deve ser realizada além do necessário para detectar os objetos e/ou regiões relevantes para a aplicação.

- ➊  $\bigcup_{i=1}^n R_i = R$ .
- ➋  $R_i$  é um conjunto conectado,  $i = 1, 2, \dots, n$ .
- ➌  $R_i \cap R_j = \emptyset$  para todo  $i$  e  $j$ , onde  $i \neq j$ .
- ➍  $Q(R_i) = VERDADEIRO$  para  $i = 1, 2, \dots, n$ .
- ➎  $Q(R_i \cup R_j) = FALSO$ , onde  $i \neq j$ , onde  $Q(R_k)$  representa a função e/ou propriedade (cor, textura, geometria) que deve ser satisfeita pelos *pixels* pertencentes à  $R_k$ .

## Segmentação de Imagens - Técnicas

- Limiar (*threshold*)
  - Forma mais simples de segmentação

$$g(x, y) = \begin{cases} 0, & \text{se } p(x, y) < T \\ 1, & \text{se } p(x, y) > T \end{cases} \quad (1)$$

- Dividir e mesclar
  - 1 Dividir a imagem em regiões de tamanhos iguais
  - 2 Verifica se as novas regiões obedecem o critério estabelecido
  - 3 Mescla as regiões em comum
  - 4 Repete os passo (1) e (2) até que não haja novas imagens
- Detecção de bordas
  - Calcula o gradiente de intensidade
  - Bordas possuem mudanças abruptas de intensidade

## Segmentação de Imagens - Limiar



Imagen original  
(<http://vejas.p.abril.com.br/>).

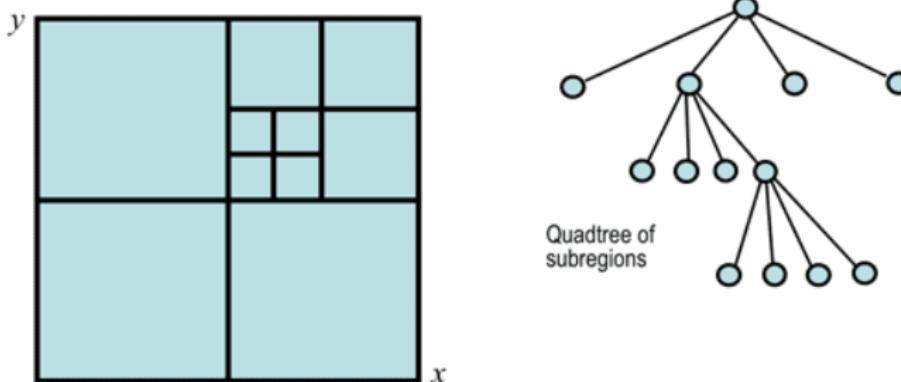


Limiar de 0.2.



Limiar de 0.5.

## Segmentação de Imagens - Dividir e mesclar



**Figura:** Dividir e Mesclar.

**Definição (Gonzales et. al)**

Não funciona bem para regiões não-uniformes.

## Segmentação de Imagens - Detecção de bordas

- Filtro de Sobel:

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



Imagen original.



Sobel horizontal.

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



$$G = (G_x * I) + (G_y * I)$$

Sobel vertical.



Sobel final.

## A técnica estudada (Protiere e Sapiro)

### Princípio

Classificar um *pixel* com base na distância e na intensidade, utilizando marcações do usuário (heurística).

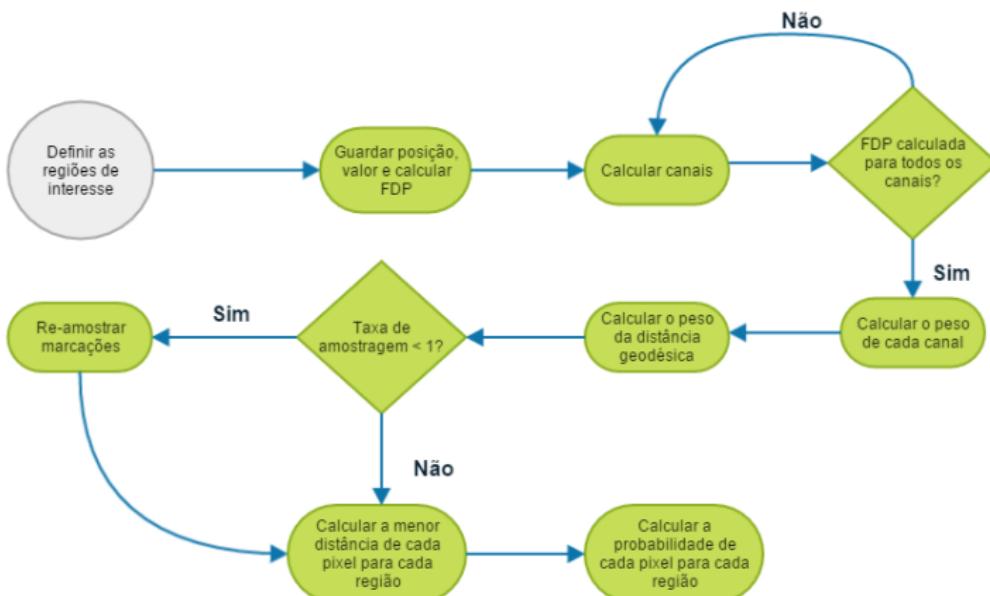
### Robustez

O algoritmo suporta a marcação de regiões não-uniformes, sendo capaz de realizar segmentações precisas em imagens complexas.

### O Problema

O tempo de execução do algoritmo é a sua principal desvantagem, sendo portanto o objeto de estudo deste trabalho.

## O algoritmo



## O algoritmo

- 1 Marcar as regiões de interesse.
- 2 Guardar valores e posição destes pontos e calcular as FDP's de cada região.
- 3 Calcular os canais.
- 4 Repetir o passo 2 para cada canal.
- 5 Calcular o peso de cada canal (adaptativo).
- 6 Calcular o peso da distância Geodésica.
- 7 Se  $T_{re-amostragem} < 1$ , então 8. Se  $T_{re-amostragem} = 1$ , então 9.
- 8 Re-amostrar os *pixels* das regiões de interesse.
- 9 Calcular a menor distância de cada *pixel* para todas as sub-regiões.
- 10 Calcular a probabilidade de cada *pixel* para todas as sub-regiões.

## Método de avaliação

Foram escolhidas duas métricas principais para avaliação:

- Tempo
  - Tempo relativo (Equação 2)
  - Funções *tic/toc* e *etime* do *MatLab*®
- Erro de classificação relativo
  - % de *pixels* classificados incorretamente (Equação 3).

$$T_r^i = 100 - \left( 100 \frac{T_{\text{total}}^i}{T_{\text{Full Set}}^{\text{total}}} \right), \text{ onde } i = 1, 10, 50. \quad (2)$$

$$\text{Erro} = \frac{N_{\text{pixels errados}}}{N_{\text{total de pixels}}}, \quad N_{\text{pixels errados}} = I_{100\%} - I_i \forall i = 50\%, 10\% \text{ e } 1\% \quad (3)$$

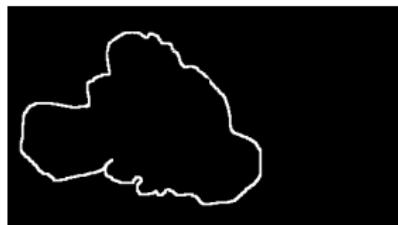
## Modificação realizada

### Redução do espaço de busca

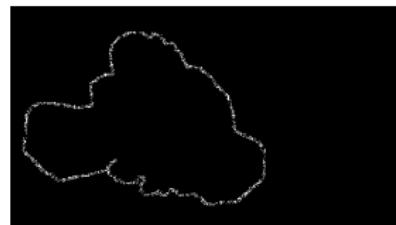
Reducir o número de cálculos para encontrar a menor distância sem perder informação.

Re-amostragem uniforme dos *pixels* das regiões de interesse usando taxas de re-amostragem de 50%, 10%, e 1%.

## Exemplo de re-amostragem



Sem re-amostragem.



Re-amostragem a 50%.

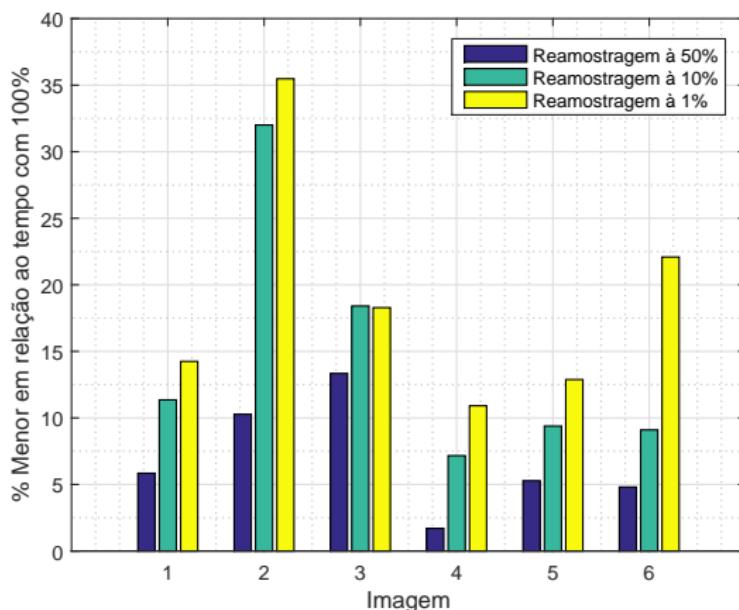


Re-amostragem a 10%.



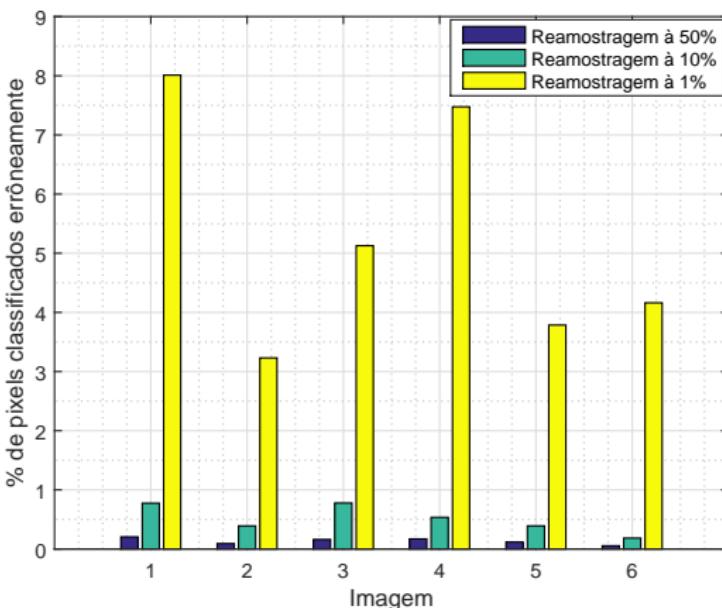
Re-amostragem a 1%.

## Resultados de tempo



**Figura:** Resultados de tempo relativo.

## Resultados de erro



**Figura:** Erro de classificação relativo.

## Imagens escolhidas



Imagen 1



Imagen 2



Imagen 3



Imagen 4



Imagen 5



Imagen 6

## Imagens resultantes (Imagen 1 - marcações)



Fundo da Imagem



Objeto 1



Objeto 2

## Imagens resultantes (Imagen 1 - objeto 1)



100%



50%

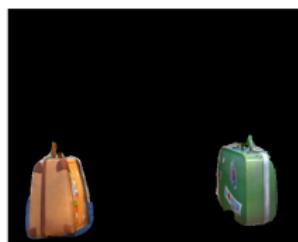


10%



1%

## Imagens resultantes (Imagen 1 - objeto 2)



100%



50%

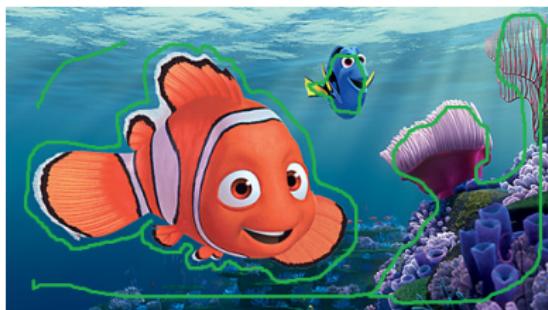


10%



1%

## Imagens resultantes (Imagen 2 - marcações)

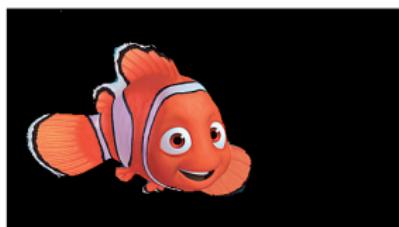


Fundo da Imagem 2

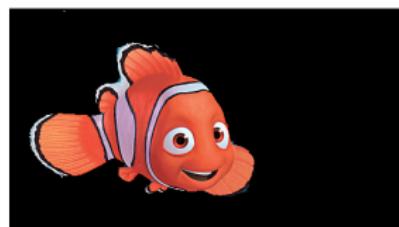


Região de interesse

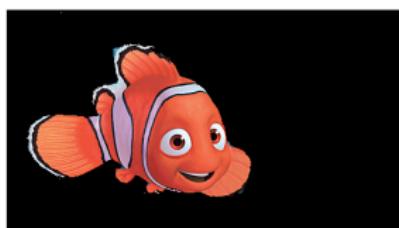
## Imagens resultantes (Imagen 2)



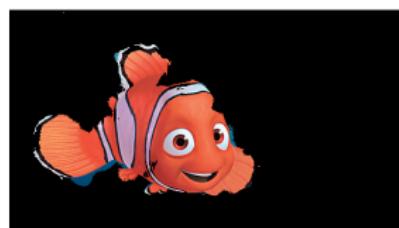
100%



50%



10%



1%

## Imagens resultantes (Imagen 3 - marcações)



Fundo da Imagem 3



Região de interesse

## Imagens resultantes (Imagen 3)



100%



50%



10%

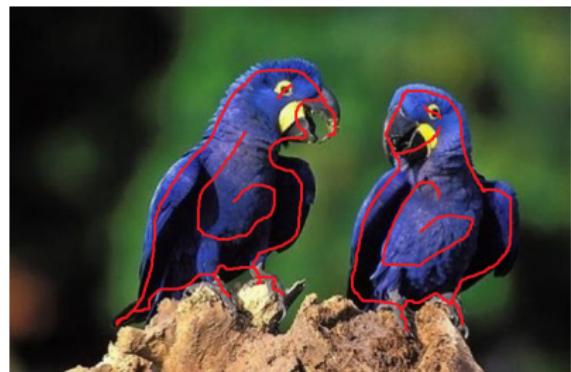


1%

## Imagens resultantes (Imagen 4 - marcações)



Fundo da Imagem 4



Região de interesse

## Imagens resultantes (Imagen 4)



100%



50%



10%

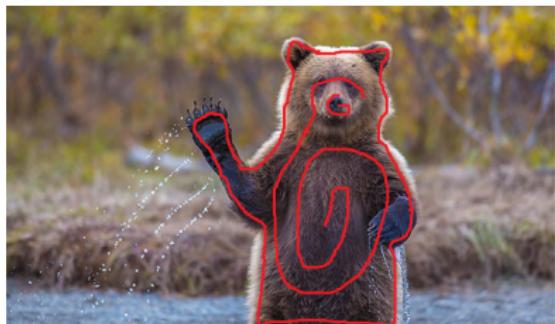


1%

## Imagens resultantes (Imagen 5 - marcações)



Fundo da Imagem 5



Região de interesse

## Imagens resultantes (Imagen 5)



100%



50%

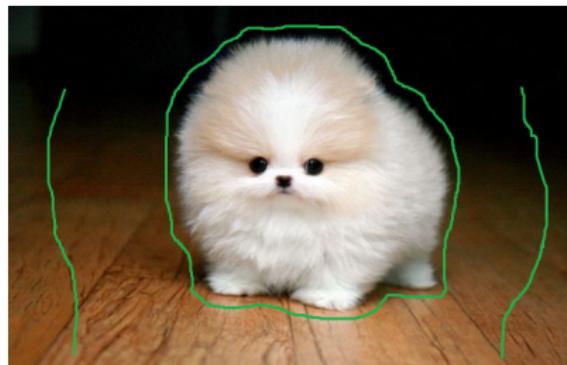


10%



1%

## Imagens resultantes (Imagen 6 - marcações)



Fundo da Imagem 6



Região de interesse

## Imagens resultantes (Imagen 6)



100%



50%



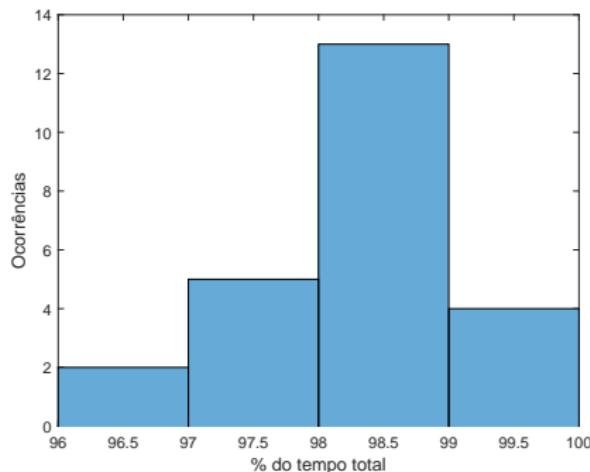
10%



1%

## Avaliação dos resultados

- A redução de tempo ficou abaixo de 35%.
- O erro de classificação ficou abaixo dos 10%.
- O cálculo das distâncias é a etapa mais demorada.



**Figura:** Histograma da relação  $\frac{T_{distancia}}{T_{total}}$  para as 24 imagens finais

## Conclusão

### Fator determinante

O total de *pixels* da imagem e das marcações influencia no tempo de execução.

### Desempenho

O algoritmo se mostrou capaz de segmentar imagens complexas.

### Contribuições

A re-amostragem apresentou resultados positivos.

### Avaliações futuras

É necessária uma análise com mais imagens (qualitativo e quantitativo).

# Obrigado!

Danilo Henrique Costa Souza  
e-mail: dhcsouza@gmail.com