



# Reconhecimento de Imagens Semelhantes

Gustavo Ciotto Pinton

Joseph Lorenzo

Pedro Mariano de Sousa Bezerra

# Introdução

- Objetivo do projeto foi de fazer um sistema de busca de imagens por conteúdo(CBIR – Content-Based Image Retrieval)
- Semelhante ao que pode ser usado no Google Images, onde você usa uma imagem em vez de usar texto para pesquisar outras imagens parecidas
- Para esse sistema, utilizamos um algoritmo de Data Mining de clusterização chamado K-Means, que reúne grupos de imagens parecidas segundo "features" dadas a ele como parâmetro
- No nosso projeto, utilizamos as chamadas "features" de Tamura, que são consideradas por todas as literaturas como o básico para o reconhecimento de imagens, além de filtros de Gabor.

# Features de Tamura

- No projeto, utilizamos os principais features descritos no livro "Textural Features Corresponding to Visual Perception" H. Tamura, S. Mori, T. Yamawaki de 1978.
- São esses features:
- - Contraste
- - Grau de direcionalidade
- - Granulação

# Granulação- I

- Relaciona-se com as distâncias entre variações espaciais notáveis dos níveis de cinza da imagem, isto é, implicitamente, ao tamanho dos elementos primitivos (texels) que formam a textura. O cálculo é feito a partir das diferenças entre as imagens obtidas após a aplicação de filtros da média com diferentes tamanhos de janela.



## Granulação - II

- Em cada pixel  $(x,y)$ , computamos seis imagens resultantes da convolução da imagem original com o filtro média de tamanho  $2^k \times 2^k$ ,  $k = 1, 2, \dots, 6$ .
- Em cada pixel  $(X_k, Y_k)$  de cada matriz  $k$  computamos a maior diferença entre os pixels adjacentes, tanto na horizontal quanto na vertical e vemos em qual matriz  $k$  está a maior diferença absoluta. Colocamos então em uma matriz, na posição  $(x,y)$  o valor  $2^k$ .
- Fazemos então a média de todos os pixels dessa matriz, que nos dá o valor final de granulação da imagem.

[http://mstrzel.elel.p.lodz.pl/mstrzel/pattern\\_rec/texture\\_analysis.pdf](http://mstrzel.elel.p.lodz.pl/mstrzel/pattern_rec/texture_analysis.pdf)



coarse

Coarseness calculada: 3.306



fine

Coarseness calculada: 2.158

# Contraste - I

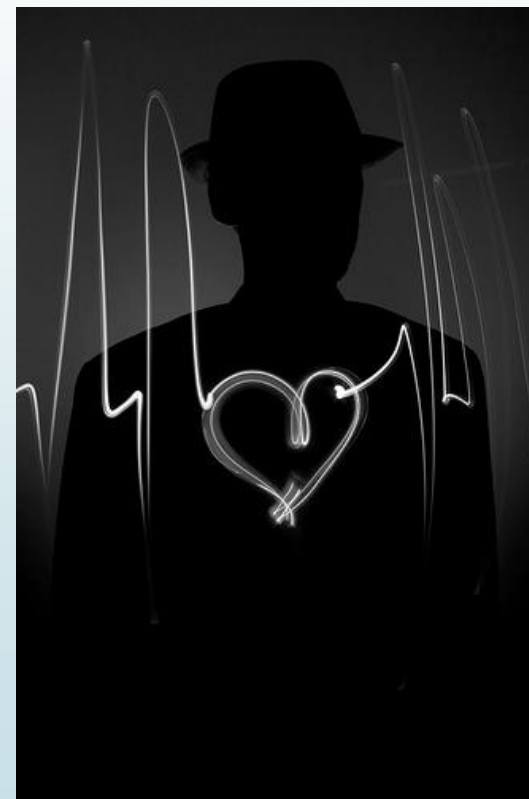
- Variação dos níveis de cinza em uma imagem
- Concentração da distribuição em preto ou branco

$$F_{con} = \frac{\sigma}{\alpha_4^n}, \text{ onde : } \alpha_4 = \frac{\mu_4}{\sigma^4}; \sigma^2 = \sum_{q=0}^{q_{max}} (q - m)^2 \frac{h(q)}{N}; \mu_4 = \sum_{q=0}^{q_{max}} (q - m)^4 \frac{h(q)}{N}$$

## Contraste - II



Contraste = 97.135



Contraste = 15.244

# Grau de direcionalidade - I

- Distribuição da frequência de bordas locais orientadas de acordo com seus ângulos direcionais
- Filtro de Sobel para detectar bordas
- Cálculo do Histograma Direcional

$$e(x, y) = 0.5(|\Delta_x(x, y)| + |\Delta_y(x, y)|)$$

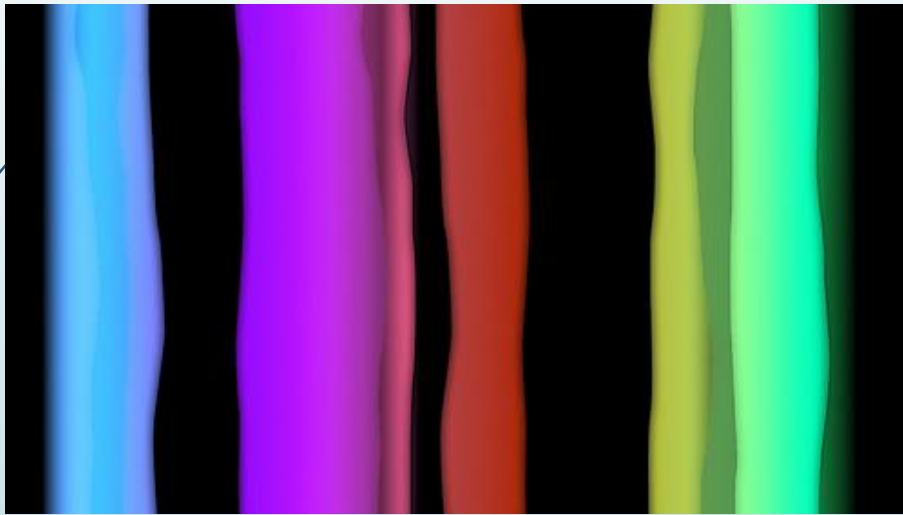
$$\alpha(x, y) = \tan^{-1} \left( \frac{\Delta_y(x, y)}{\Delta_x(x, y)} \right)$$

$$F_{dir} = 1 - r n_{peaks} \sum_{p=1}^{n_{peaks}} \sum_{a \in w_p} (a - a_p)^2 H_{dir}(a)$$

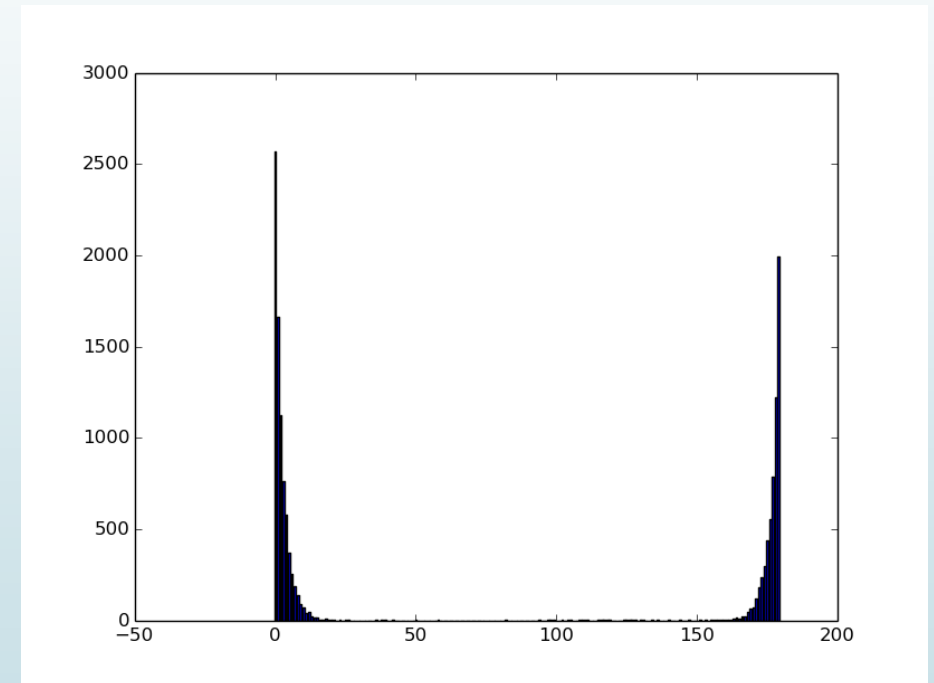


## Grau de direcionalidade - II

- Histograma apresenta picos para imagens altamente direcionais



Grau = 0.98643

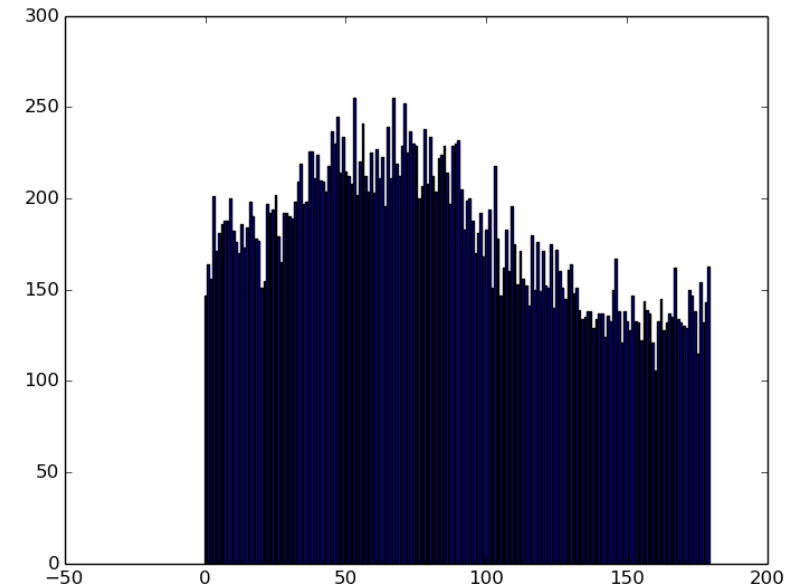


## Grau de direcionalidade - III

- Histograma relativamente uniforme para imagens sem forte orientação



Grau = 0.00076



# Filtros de Gabor - I

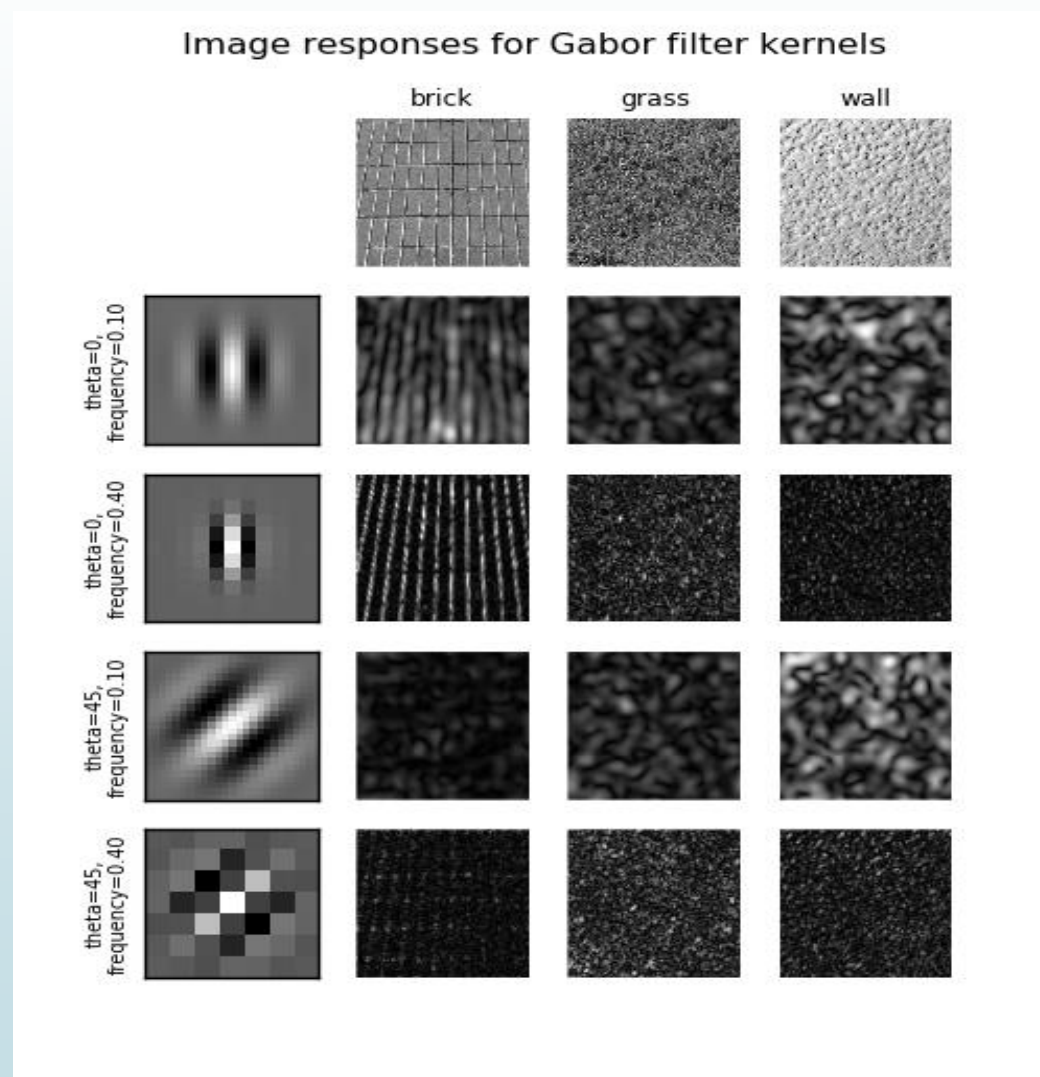
- Também podem ser utilizados para a detecção de bordas;
- Orientações e frequências em tais filtros correspondem às mesmas características encontradas na visão humana;
- Features: média e variação das imagens filtradas;
- Variação de theta, sigma e frequência;
  - Theta =  $[0, \pi/4, \pi/2, 3\pi/4]$ , sigma =  $[1, 2]$  e frequência =  $[0.05, 0.25]$
  - 16 filtros no total, 32 features por imagem
- Função de transferência do filtro:

$$g(x, y; \lambda, \theta, \psi, \sigma, \gamma) = \exp \left( -\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2} \right) \exp \left( i \left( 2\pi \frac{x'}{\lambda} + \psi \right) \right)$$

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

$$y' = -x \sin \theta + y \cos \theta$$

# Filtros de Gabor - II



# Cálculo do K-means

## Pré-processamento - I

- Base de imagens: 25000 extraídas do flicker. Tamanhos e objetos variáveis, coloridas ou não;
- Módulo `scipy.cluster.vq`;
- 2 níveis de K-means:
  - Primeiro nível calculado com features explicadas anteriormente;
  - Segundo nível calculado com histogramas de nível de cinza;
- Primeiro nível 200 centróides; segundo, 20 centróides:
- Estimativa de  $25000 / (200 \cdot 20) = 6$  imagens por centróide de 2o nível
- Alta densidade de imagens:
  - Tempo de cálculo: em torno de 10 segundos por imagem (coarseness ~9s)
  - 250000 segundos = 69.4 horas = 2.90 dias!
  - Arquivos salvos a cada estágio!

# Cálculo do K-means

## Pré-processamento - II

- Obter matriz de 25000 linhas, em que cada linha contém os features das imagens;
- `Scipy.cluster.vq.whitten`: colunas com variância unitária;
- Cálculo dos 200 centróides de 1o nível - `scipy.cluster.vq.kmeans`.
- Para cada centróide:
  - Obter histogramas em nível de cinza das imagens contidas;
  - Calcular `kmeans` com 20 centróides com estas imagens;
- Salvar arquivos! - Complexidade do `kmeans`:  $O(n^{dk+1} \log n)$ 
  - K centróides, n amostras e d dimensões

# Encontrar resultados

- Calcular features da imagem pesquisada
- Encontrar o seu centróide de 1o nível, baseando-se nos arquivos gerados;
- Carregar 20 centróides de 2o nível referentes a ele (também a partir de arquivos gerados no pré-processamento);
- Encontrar seu centróide de 2o nível;
- Algoritmo de NearestNeighbors: módulo `sklearn.neighbors`.
  - Recuperação dos 3 vizinhos mais próximos DENTRO do centróide de 2o nível.
- Mostrar resultados!



# Resultados - I



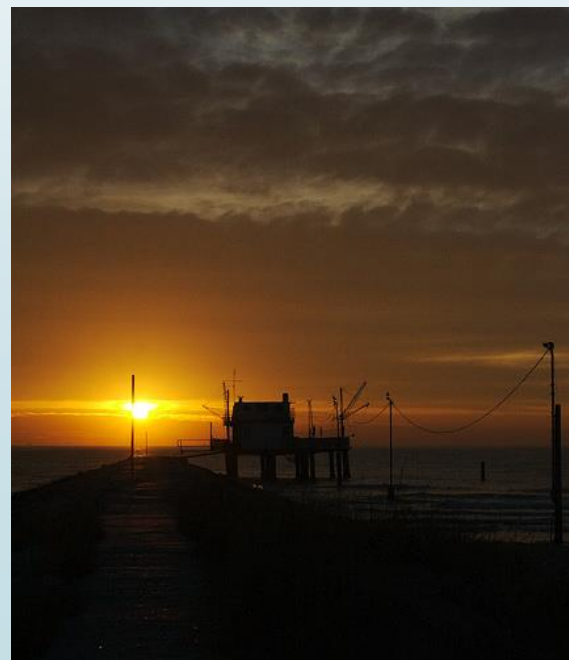
Pesquisada



Vizinho 1



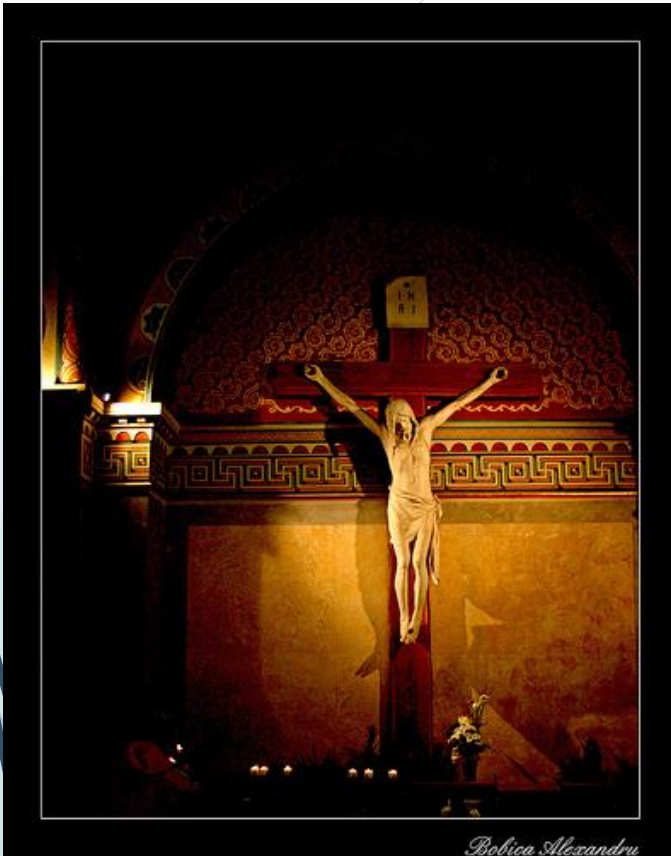
Vizinho 2



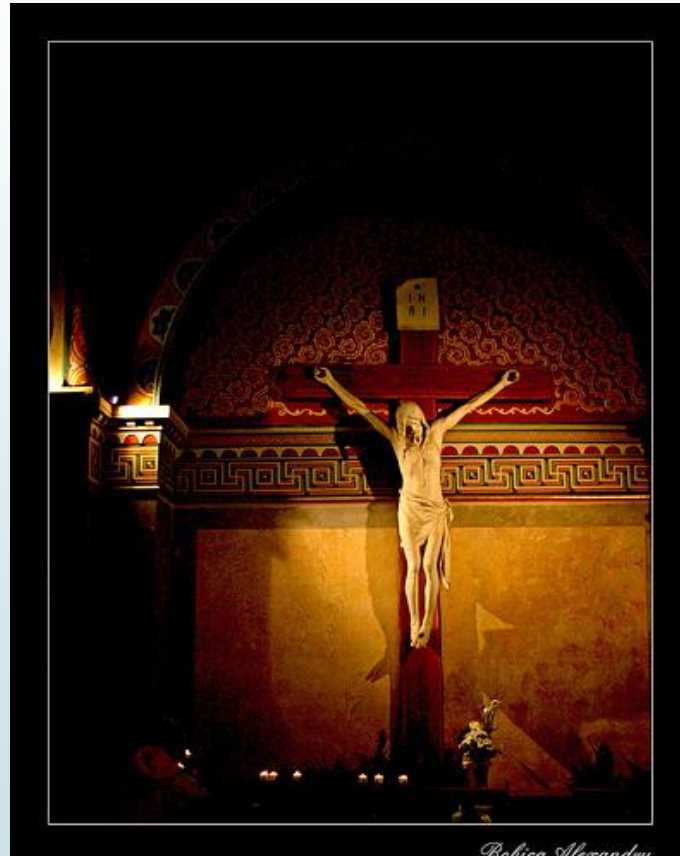
Vizinho 3



## Resultados - II



Pesquisada



Vizinho 1



Vizinho 2



Vizinho 3

# Resultados - III



Pesquisada



Vizinho 1



Vizinho 3

Vizinho 2



## Resultados - IV



Pesquisada



Vizinho 2



Vizinho 1



Vizinho 3

# Referências

[1] Prof. Georgy Gimel'farb. Lecture Notes from the course "Hypermedia and Multimedia Systems" at the University of Auckland. "CBIR: Texture Features". Disponível em:

<https://www.cs.auckland.ac.nz/courses/compsci708s1c/lectures/Glect.html/topic4c708FSC.htm#tamura>

[2] Wikipédia - Gabor Filter. Disponível em:

[http://en.wikipedia.org/wiki/Gabor\\_filter](http://en.wikipedia.org/wiki/Gabor_filter)

[3] <http://mplab.ucsd.edu/tutorials/gabor.pdf>