

## Trabalho 3

Implemente sua atividade sozinho sem compartilhar, olhar código de seus colegas, ou buscar na Internet. Procure usar apenas os conceitos já vistos nas aulas.

### Extração de Características Locais e Multiescala

Nesse trabalho você deverá implementar um sistema que permita extrair atributos de imagens em diversas escalas. A idéia é capturar características presentes em diversas resoluções, obtendo informações sobre detalhes finos e de baixo contraste e também de objetos maiores e de alto contraste.

O sistema deverá abrir imagens para extrair as características. Para cada imagem:

1. Calcular uma pirâmide de imagens com 3 níveis.
2. Detectar pontos chave em cada nível utilizando o método de Harris.
3. Utilizar os pontos chave para obter o histograma ao redor desses pontos e criar um vetor de características composto da concatenação dos histogramas em cada escala.

### Pirâmide

A pirâmide será montada da seguinte forma, recursivamente, a partir de uma imagem de entrada de nível  $j$ :

1. Filtre a imagem utilizando filtro Gaussiano  $k \times k$  (usando  $\sigma = 1.44$ ),
2. Realize uma subamostragem por um fator de 2,
3. Posicione a imagem resultante no nível  $j - 1$  da pirâmide.

A subamostragem pode ser feita de diversas formas. Considerando a imagem original  $f$  indexada por  $(x, y)$  e a imagem reduzida  $g$  indexada por  $(m, n)$ , dois métodos possíveis são:

1. Calcular a nova imagem usando a média entre os pixels de regiões  $2 \times 2$ . Exemplos:  
 $g(0, 0) = \frac{1}{4}(f(0, 0) + f(0, 1) + f(1, 0) + f(1, 1))$ ,  $g(0, 1) = \frac{1}{4}(f(0, 1) + f(0, 2) + f(1, 1) + f(1, 2))$ .
2. Utilizar o valor de um pixel na posição  $(x, y)$  e deslocar 2 pixels à frente. Exemplos:  
 $g(0, 0) = f(0, 0)$ ,  $g(0, 1) = f(0, 2)$ ,  $g(1, 0) = f(2, 0)$ .

Você poderá ainda utilizar alguma função pronta de alguma biblioteca para realizar a redução da imagem.

### Pontos chave

1. Obtenha uma versão em níveis de cinza da imagem.
2. Compute as respostas passa-altas  $X = I_x^2(\mathbf{x})$ ,  $Y = I_y^2(\mathbf{x})$  e  $Z = I_x I_y(\mathbf{x})$
3. Gere um filtro gaussiano  $5 \times 5$  com  $\sigma = 1.44$  e obtenha as versões filtradas de  $X$ ,  $Y$  e  $Z$ , chamadas  $A$ ,  $B$  e  $C$ .

4. Calcule  $Harris = [AB - C^2] - \alpha(A + B)^2$
5. Encontre os  $N$  máximos locais, para isso:
  - a) Crie uma matriz  $PC$  para armazenar os pontos chave (máximos locais), com valor inicial de 0 para todos os pixels.
  - b) As respostas nos primeiros 4 pixels próximos da borda da imagem não serão consideradas.
  - c) Centre uma janela  $7 \times 7$  (a partir do pixel (5,5)) e verifique se esse pixel é máximo naquela vizinhança. Se for máximo, atribua 1 ao pixel relativo na matriz de pontos chave  $PC$ . Desloque a janela em um pixel e repita a operação.
  - d) Mantenha apenas os  $N$  pontos chave de maior resposta na matriz  $Harris$ .

### Extraindo as características: Local Multiscale Histograms (LMH)

Para cada nível da pirâmide:

1. Compute os  $N$  pontos chave a partir do método de Harris.
2. Compute o histograma da imagem acumulando a cor dos pixels numa janela  $w \times w$  centrada em cada ponto chave (veja um exemplo na Figura 1), a ordem de concatenação deve ser do ponto chave de maior resposta para o menor.

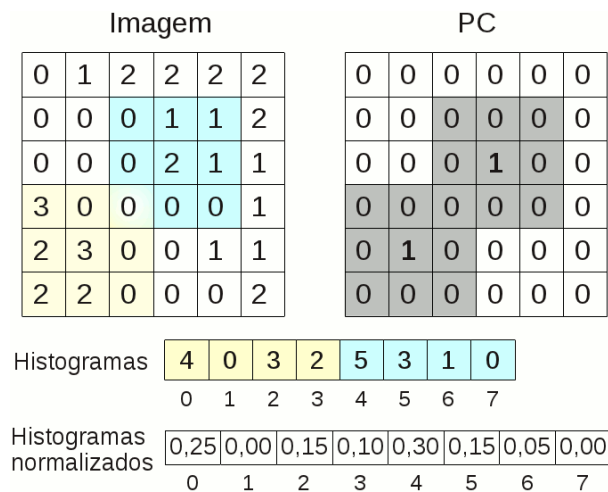


Figura 1: Exemplo usando janelas com  $w = 3$  considerando 4 cores e  $N = 2$

### Requisitos

Funções a serem implementadas:

- `void ImagePyramid(IplImage *I, double k, int J, IplImage **pyramid)` que retorne na variável `pyramid` uma pirâmide de  $J$  níveis filtrada com filtro de aproximação de tamanho  $k \times k$ , utilizando como base da pirâmide a imagem  $I$ .

- `void LMH(IplImage *I, double k, int J, int w, double *features)` que retorne no vetor `features` os atributos extraídos das imagens em  $J$  escalas utilizando filtro gaussiano de tamanho  $k$  e janelas para calcular os histogramas ao redor dos pontos chave de tamanho  $w$ .

Observação: a assinatura das funções acima usam a sintaxe da linguagem C/OpenCV, o que pode ser modificado se você usar linguagem de outra sintaxe.

## Experimento

Serão fornecidas 4 imagens (em arquivo separado): `0.jpg`, `1.jpg`, `2.jpg` e `3.jpg` para que sejam extraídas as características. Para esse experimento, você deverá utilizar  $J = 3$  níveis de escala, filtros de tamanho  $k = 5$  e janelas para extração de características  $w = 9$ .

O número de pontos-chave deverá ser calculado a partir do tamanho da imagem, utilizando  $N = \lfloor \max(n, m) \cdot 0,015625 \rfloor$ , sendo a imagem de tamanho  $n \times m$ . Por exemplo, numa imagem de  $800 \times 600$  o número de pontos chave será  $N = \lfloor \max(800, 600) \cdot 0,015625 \rfloor = \lfloor 800 \cdot 0,015625 \rfloor = 12$ .

Repare que, dessa forma, em imagens de menor resolução da pirâmide será considerado um menor número de pontos-chave.

Você deverá calcular os descritores multiescala para cada imagem, e imprimir na tela uma matriz com as distâncias (usar distância  $L_0$  — Manhattan) entre as 4 imagens.

## Instruções

- Trabalho em duplas
- Data de entrega: 12/12/2011
- **Incluir no arquivo compactado um Makefile e/ou instruções sobre como compilar e executar seu programa.**
- **Não incluir imagens no arquivo.**

O projeto será avaliado levando em consideração:

1. Construção da pirâmide (20%).
2. Tratamento dos pontos-chave (30%).
3. Extração dos histogramas, montagem do vetor de características (30%).
4. Experimento de comparação entre as imagens (20%).