PIM

Prof. Gilmário

Trabalho Prático – aplicação do operador gradiente na detecção de bordas
Salvo recomendação alguma explícita, não utilize operadores já prontos do

Implemente as convoluções necessárias.

Veja os detalhes da entrega no link de upload no Moodle

Entrega acompanhada de relatório

I) Objetivo

Estudo da aplicação de filtro passa baixa e o filtro diferencial (Sobel, Prewitt e Scharr) na composição do operador gradiente na detecção de borda (pontos de intensidade máxima local).

Aplicação de Python-SSIM e OpenCV.

II) Descrição do problema

Uma imagem em tons de cinza é uma função bidimensional f(x,y) que pode ser interpretada como uma superfície contendo aclives e declives correspondentes às transições, mais ou menos acentuadas, de intensidades de brilho. Essas transições podem ser úteis, por exemplo, como fonte de informação para configurar um descritor da imagem ou para uma operação de realce de bordas de elementos da imagem.

A taxa de variação pela transição de regiões de brilho na imagem pode ser detectada, em cada coordenada de pixel, pela magnitude do vetor $|\nabla(f)|$ obtido a partir da aplicação do operador gradiente $\nabla(f)$. Como o gradiente é um vetor, a sua determinação depende dos componentes horizontal e vertical (Gx,Gy) que o compõem em cada coordenada da imagem,

$$G_x = \frac{\partial f}{\partial x}$$

$$G_y = \frac{\partial f}{\partial y}$$

$$\nabla(f)=\begin{pmatrix}\frac{\partial f}{\partial x}\\ \frac{\partial f}{\partial y}\end{pmatrix}$$

O cálculo dos componentes equivale à aplicação de derivadas direcionais no domínio discreto, as quais podem ser realizadas por filtros derivativos adequados tais como Sobel, Prewitt e Scharr.

Além da aplicação de filtros derivativos existem algumas operações adicionais que podem ser necessárias. Filtros derivativos são muito sensíveis a ruídos na imagem. Se imagem de entrada estiver “ruidosa”, será necessário aplicar um filtro adequado para atenuação desse ruído.

As etapas básicas da aplicação do operador gradiente constam abaixo, na Figura 1.

OBSERVAÇÃO: caso a imagem de teste esteja em RGB, antes de qualquer operação, a converta para tons de cinza.

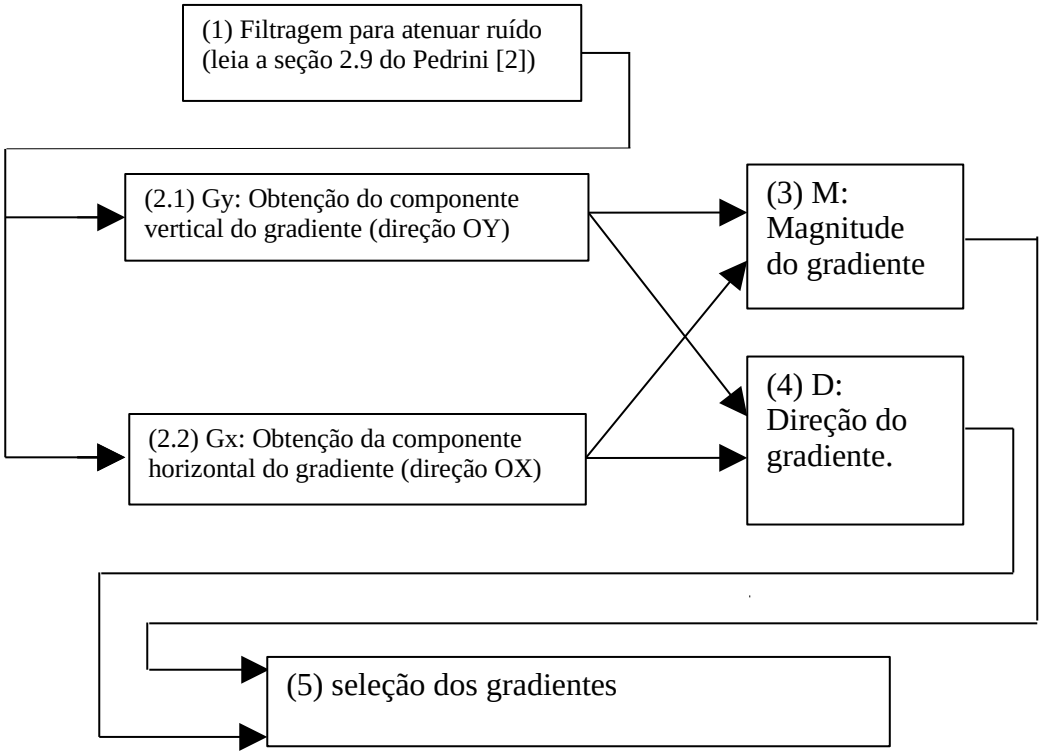

Figura 1: Fluxograma do cálculo do gradiente de uma imagem em tons de cinza.

1) Pré-filtragem

As etapas 2.1 e 2.2 são realizadas por filtros (*h*) baseados em derivadas, os quais são sensíveis às componentes de alta frequência. Por conta disso, a aplicação destes filtros normalmente é precedida por uma atenuação dos ruídos (leia a seção 2.9 do Pedrini [2]) na imagem a ser tratada.

2) Aplicação do filtro derivativo
2.1) Determinação das componentes horizontais (Gx) utilizando máscara adequada (Sobel, Prewitt e Scharr) para a direção OX (taxa de variação entre colunas da imagem);
2.2) Determinação das componentes verticais (Gy) utilizando máscara adequada (Sobel, Prewitt e Scharr) para a direção OY (taxa de variação entre linhas da imagem);

Atenção: você não pode misturar operadores, por exemplo: aplicar Gx por Sobel e Gy por Prewitt.

3) Determinação da magnitude do gradiente (M)
As matrizes/imagens Gx e Gy são utilizadas para a construção da matriz/imagem M, a qual representa as magnitudes dos pixels de *f*.

$$M(i,j) = \sqrt{(G_y(i,j))^2 + (G_x(i,j))^2}$$

4) Determinação da direção do gradiente (D)
As matrizes/imagens Gx e Gy são utilizadas para a construção da matriz/imagem D, a qual representa as direções dos gradientes cujas magnitudes foram calculadas em M(i,j):

$$D(i,j) = \arctan \frac{G_y(i,j)}{G_x(i,j) + \varepsilon}$$

Computacionalmente, o denominador precisa sem somado a um epsilon igual a um valor próximo de zero, por exemplo, $\varepsilon = 10^{-8}$).

Sugere-se a utilização da função *math.atan2* do Python (AQUI).

5) Seleção dos máximos locais (supressão dos não máximos)
5.1) Máximos locais simples:

A magnitude M(i,j) será máximo local (pertence a uma borda) se for maior do que K (K >= 1) vezes as magnitudes dos gradientes dos seus dois vizinhos colineares adjacentes na direção D(i,j) do gradiente.

Caso a intensidade de M(i,j) seja confirmada como um máximo local em relação às duas intensidades vizinhas na mesma direção, a intensidade M(i,j) identificará um pixel de borda fazendo imagemSaida(i,j)=1, caso contrário imagemSaida(i,j)=0.

Como implementar a seleção dos gradientes?

Leia Gradiente_complemento_V1.pdf disponível no Moodle.

III) Tarefa

Utilize as imagens moedas.png, Lua1_gray.jpg, chessboard_inv.png e img2.jpg disponíveis no site da disciplina no Moodle (você pode utilizar outras adicionalmente, mas não em substituição a essas imagens citadas).

Devem ser testadas duas máscaras dentre Sobel, Prewitt e Scharr. Implemente o operador gradiente de acordo com o que foi especificado anteriormente.

A seleção de gradientes deve ser produzida por você utilizando a seleção de máximos locais conforme descrito nos item 2 em *Gradiente_complemento_V1.pdf*.

Compare os resultados das soluções pelas duas máscara que você escolheu (Prewitt e Scharr por exemplo), analise quais resultados apresentaram bordas mais bem definidas e coerentes com as respectivas imagens originais. Analise o impacto da variação do parâmetro K.

$$Operadores\ \ de\ \ Prewitt:$$

$$Pw_x = \begin{pmatrix} -1 & 0 & +1 \\ -1 & 0 & +1 \\ -1 & 0 & +1 \end{pmatrix} Pw_y = \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ +1 & +1 & +1 \end{pmatrix}$$

$$Operadores\ \ de\ \ Sobel:$$

$$Sb_x = \begin{pmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{pmatrix} Sb_y = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{pmatrix}$$

$$Operadores\ \ de\ \ Scharr:$$

$$Sc_x = \begin{pmatrix} -3 & 0 & +3 \\ -10 & 0 & +10 \\ -3 & 0 & +3 \end{pmatrix} Sc_y = \begin{pmatrix} -3 & -10 & -3 \\ 0 & 0 & 0 \\ +3 & +10 & +3 \end{pmatrix}$$

Bibliografia

[1] Gonzalez, R. e Woods, R. "Processamento digital de Imagens", 3a ed. Ed. Pearson, 2010.

[2] Pedrini, Hélio. Livros Análise de Imagens Digitais - Princípios, Algoritmos e Aplicações. Editora Thomson Learning,2007.