DEEC
Faculdade de Ciências e Tecnologia
UNIVERSIDADE de COIMBRA

VISÃO POR COMPUTADOR

2º Semestre **Ano lectivo 2015/2016**

Trabalho Prático Nº 3

Detecção de pontos característicos na imagem ("Cantos, Rectas e Circunferências")

- ***Detecção de cantos***

Este método deve ser aplicado à imagem "chess_2.png".

Considere um ponto genérico da imagem ***P***, uma vizinhança ***Q*** e uma matriz ***C*** definida por

$$C = \begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix}$$

onde os somatórios são calculados na vizinhança ***Q***. I_x e I_y representam respectivamente as componentes horizontais e verticais do gradiente calculado no ponto ***P*** (relembrar trabalho prático nº 1 sobre diferenciação da imagem). A matriz ***C*** caracteriza a estrutura dos níveis de cinzento da imagem.

Sendo ***C*** uma matriz simétrica, ela pode ser diagonalizada através da rotação dos eixos de coordenadas, obtendo-se uma matriz diagonal

$$\boldsymbol{C}= \begin{vmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{vmatrix}$$

onde λ_1 e λ_2 correspondem aos valores próprios da matriz ***C***.

Os dois valores próprios λ_1 e λ_2 são ambos positivos. Para entender o significado associado aos valores próprios λ_1 e λ_2 e aos correspondentes vectores próprios considere-se o seguinte: Para uma vizinhança ***Q*** em que o padrão dos seus níveis de cinzento é uniforme as componentes de gradiante tendem para zero e ***C*** transforma-se numa matriz nula, resultando $\lambda_1=0$ e $\lambda_2=0$. Para um padrão binário que corresponde a uma transição em degrau entre o nível de cinzento branco e negro, obtem-se $\lambda_1>0$ e $\lambda_2=0$, e o vector próprio associado a λ_1 é paralelo ao gradiente da imagem. Se a vizinhança ***Q*** contiver o canto de um quadrado negro sobre fundo branco existem duas orientações principais de gradiente na vizinhança ***Q***, obtendo-se $\lambda_1>0$ e

$\lambda_2>0$, correspondendo aos valores mais elevados de valores próprios as transições com maior gradiante.

Um canto ("corner") é identificado através da intersecção de dois contornos com distintas orientações. Como $\lambda_1>=\lambda_2$, um canto é definido pela localização do ponto ***P*** onde o valor próprio mais pequeno λ_2 é suficientemente elevado.

Trabalho a realizar :

1 - Implemente um programa em MATLAB que permita detectar cantos na imagem. Para tal analise o algoritmo que a seguir se apresenta:

Algoritmo de detecção de cantos ("corners")

Para uma imagem ***I*** considere uma vizinhança ***Q*** de dimensão *2N+1* x *2N+1* pixeis. Definir um valor de limiar σ para λ_2 acima do qual se considera a existência de um canto.

1 - Calcular as componentes X e Y do gradiente em toda a imagem.

2 - Para cada ponto da imagem ***P***:

a) Obter a matrix ***C*** correspondente à vizinhança ***Q*** ;

b) Calcular os valores próprios de ***C***, e seleccionar o valor próprio mais baixo λ_2.

Utilize a função *eig* disponível no Matlab.

c) Se $\lambda_2>=\sigma$, guardar as coordenadas do ponto ***P*** numa lista ***L***.

3 - Ordenar a lista ***L*** por ordem decrescente dos valores de λ_2.

Utilize a função *sort* disponível no Matlab.

4 - Percorrer a lista ordenada no sentido descendente e para cada ponto ***P*** da lista, eliminar da lista todos os pontos que pertencem à área da sua vizinhança.

Esta eliminação assegura que não são detectados vértices com pontos de vizinhança comuns.

Este algoritmo fornece uma lista de pontos com $\lambda_2>=\sigma$ e cujas vizinhanças não se sobrepõem.

2 - Apresente o histograma de λ_2 ao longo da imagem ***I***. O histograma pode ajudar a definir um valor mais correcto para o limiar σ.

3 - Simule o algoritmo numa imagem sintética com cantos bem definidos. Por exemplo, uma imagem com quadrados negros em fundo branco. Use a função *checkerboard* (use o comando de modo a criar uma imagem em que há quadrados negros com fundo branco em metade da imagem, sendo que, na outra metade, o fundo tem um nível de cinzento médio (ex. 128). Teste o algoritmo com vizinhanças 3*3 e 5*5. Para os cantos que detectou represente também os vectores próprios da matrix C (tensor de estrutura). Que conclusões tira quanto à orientação dos vectores próprios? Qual é o efeito do nível de cinzento do fundo?

4 – Aplique o algoritmo numa imagem real. Teste o algoritmo com vizinhanças 3*3 e 5*5. Compare com os resultados obtidos com as seguintes funções Matlab: `detectCheckerboardPoints`, *corner* (as duas opções), *detectHarrisFeatures*, e *detectMinEigFeatures*.

- ## *Detecção de rectas e circunferências usando a transformada de Hough*

A detecção de rectas e de circunferências numa imagem pode ser um problema complexo. Vários métodos e processos existem para fazer essa estimação. Um dos

algoritmos, descrito nas aulas teóricas, baseia-se na transformada de Hough. Escreva um programa que, usando a transformada de Hough, permita a detecção das rectas e circunferências na imagem “lines_circles_2.jpg”.

O programa deve fazer a detecção das rectas e das circunferências com discretizações diferentes dos parâmetros (no caso das rectas deve ser usada a equação polar das rectas). O programa deve fazer o “display” das matrizes usadas pela transformada de Hough para determinar os parâmetros, com os níveis de cinzento quantificando o valor dos votos. Como saída do programa deve também fornecer os parâmetros das rectas (declive e ordenada na origem) e circunferências estimadas (coordenadas do centro e raio) e fazer o seu traçado numa imagem de saída que deve ser comparada com a original. Se o programa for muito lento pode reduzir a dimensão das imagens.

O Matlab tem disponíveis as funções *hough*, *houghppeaks*, e *houghlines*. Use estas funções para detectar rectas na imagem “chess_1.png”.