

INSTITUTO NACIONAL DE MATEMÁTICA PURA E APLICADA

HALLISON OLIVEIRA DA PAZ

CURSO DE PROCESSAMENTO DE IMAGENS

Assignment 3: Image Features

RIO DE JANEIRO

2015

1 Introdução

Este relatório descreve a implementação do *assignment Image Features* do curso de Processamento de Imagens. Foram utilizados a biblioteca OpenCV (3.0) para implementação das funções de visão computacional, em particular para a detecção de cantos com o algoritmo de *Harris*, e o ambiente de desenvolvimento integrado QT Creator para a confecção das interfaces gráficas. O código fonte encontra-se no repositório [1].

2 Explicação do Algoritmo

O Algoritmo de Harris para detecção de cantos e arestas é um dos algoritmos mais antigos para a detecção de *features* e baseia-se em uma ideia bastante simples e intuitiva. A ideia é perceber que em uma imagem, as regiões que representam cantos são regiões onde o gradiente apresenta uma alta variação em qualquer direção. Isto porque, em regiões de transição (variações bruscas) no domínio da imagem observamos altas frequências. Como um canto tipicamente está na interseção de arestas (delimitadores de transição) com direções diferentes, pensar nos cantos como zonas onde o gradiente tem alta variação é bastante natural.

Assim, para criar um modelo matemático que possa lidar com esta ideia, podemos definir janelas que representem vizinhanças de pixels na imagem e, então, calculamos uma medida de variação de energia obtida ao movermos esta janela em várias direções. Pode-se atribuir pesos diferentes para pixels mais afastados do centro da janela, caso se deseje. A equação 1 traduz esta ideia.

$$E(u, v) = \sum_{x, y} w(x, y) [I(x + u, y + v) - I(x, y)]^2 \quad (1)$$

Na equação, temos:

- $w(x, y)$ é a função de janela referente à posição (x, y) da imagem (pesos dos pixels).
- $I(x, y)$ é a intensidade do pixel na posição (x, y) da imagem.
- $I(x + u, y + v)$ é a intensidade na janela movida para a posição $(x + u, y + v)$ (transladada de (u, v)).

A variação medida pela equação 1 nos permite caracterizar a região em análise de acordo com os 3 casos seguintes:

1. Se a variação for pequena em toda as direções, estamos em uma região lisa, de poucos detalhes na imagem.
2. Se a variação for grande em apenas uma direção e pequena nas demais, estamos sobre uma aresta. A direção de maior variação é a direção normal à aresta (direção do gradiente).
3. Se a variação for grande em várias direções, então temos um bom candidato a canto.

A figura 1 ilustra esta classificação de regiões.

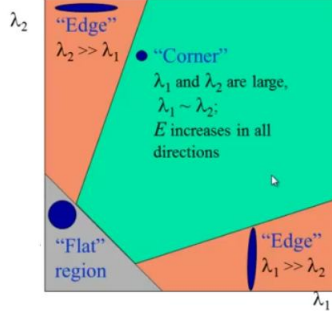


Figure 1: Classificação das regiões[2]

Portanto, a ideia do algoritmo de Harris é procurar por janelas onde há alta variação, o que se traduz em maximizar a função de energia da equação (1).

Usando uma aproximação por expansão de Taylor, temos:

$$E(u, v) \approx \sum_{x,y} w(x, y) [I(x, y) + uI_x + vI_y - I(x, y)]^2 \approx \sum_{x,y} w(x, y) (u^2 I_x^2 + 2uv I_x I_y + v^2 I_y^2)$$

Que pode ser expressa em forma matricial como:

$$E(u, v) \approx \begin{bmatrix} u & v \end{bmatrix} \left(\sum_{x,y} w(x, y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \right) \begin{bmatrix} u \\ v \end{bmatrix}$$

Seja:

$$M = \sum_{x,y} w(x, y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

Substituindo, chegamos à expressão:

$$E(u, v) \approx \begin{bmatrix} u & v \end{bmatrix} M \begin{bmatrix} u \\ v \end{bmatrix} \quad (2)$$

Com esta formulação (equação 2), podemos analisar em que caso um determinado pixel se encontra a partir de uma análise dos autovalores da matriz M . Se ambos os autovalores têm uma magnitude pequena, trata-se do caso 1 (região lisa). Caso um dos autovalores tenha uma magnitude significativamente superior à do outro, temos o 2º caso (aresta). Finalmente, se ambos os autovalores têm alta magnitude, temos um possível canto.

Uma grande contribuição do trabalho de Harris está na definição de uma “função de resposta do canto” (equação 3), que permite uma análise direta destes casos a partir de funções simples sobre a matriz M . Sejam λ_1 e λ_2 os autovalores de M e considerando que $\det(M) = \lambda_1 \lambda_2$ e $\text{traço}(M) = \lambda_1 + \lambda_2$. A função de resposta do canto foi definida como:

$$R = \det(M) - k(\text{traço}(M))^2 \quad (3)$$

Em que k é uma constante determinada empiricamente como um valor entre 0.04 e 0.06 para bons resultados.

Assim, estabelece-se um critério bastante objetivo para classificação das arestas. De acordo com a implementação do algoritmo de Harris, se o valor absoluto de R for pequeno, tem-se uma região lisa, se R for negativo, tem-se uma aresta e se for positivo temos um candidato a canto. Assim, basta apenas estipular um limiar positivo acima do qual consideraremos que há um canto dentro da janela em questão.

Por conta de sua simplicidade, algumas propriedades importantes não podem ser captadas pelos algoritmo de Harris, imprimindo-lhe algumas deficiências. Em particular, a sensibilidade à variação de escala, devido à utilização de janelas de tamanho fixo, restringe uma quantidade considerável de possíveis aplicações.

3 Implementação

O programa implementado permite que o usuário selecione uma imagem em seu computador ou utilize a câmera principal do sistema para realizar um processamento contínuo.

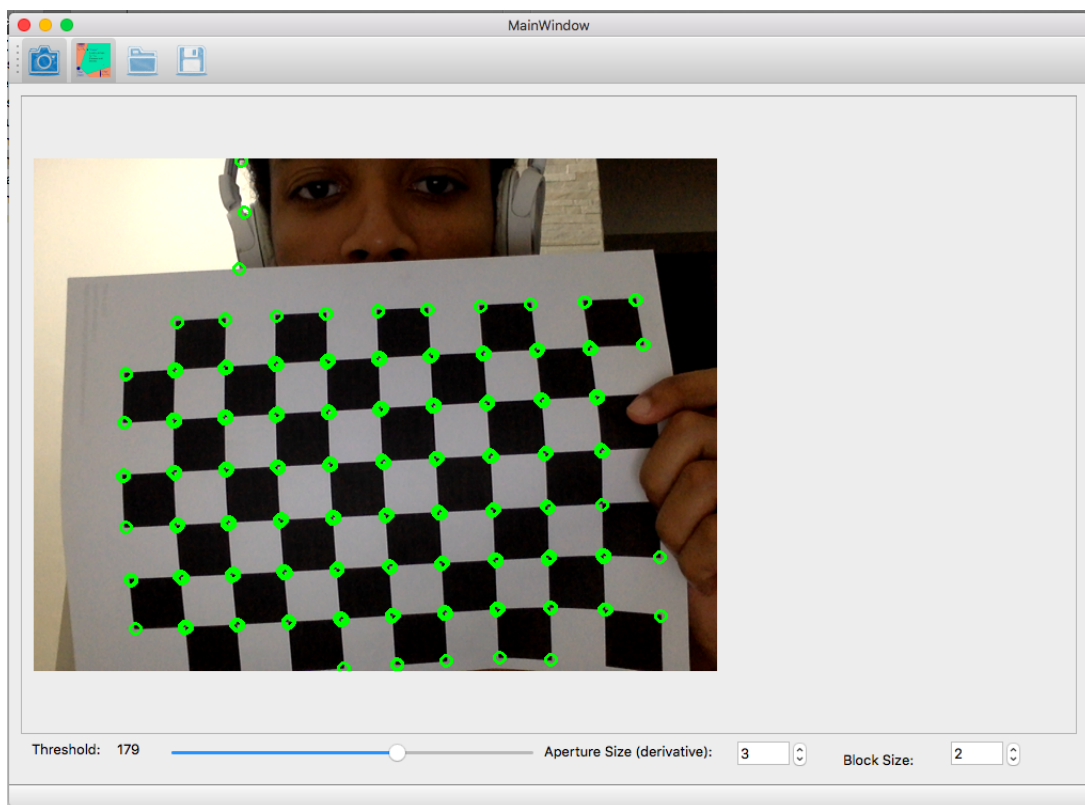


Figure 2: Interface do programa

A imagem carregada é exibida na janela principal do programa. Caso seja escolhida a opção de captura contínua, a leitura da câmera fará atualização constante dessa imagem.

Ao selecionar a opção “Apply Harris Detector”, realiza-se uma cópia da imagem exibida, convertendo-a para um espaço de cor em escala de cinza. A imagem monocromática é, então, submetida ao algoritmo de Harris para detecção de cantos. Os parâmetros do algoritmo (tamanho da janela, tamanho da máscara de derivada, valor limiar para a função de resposta de canto e constante k empírica) são expostos para manipulação do usuário, que pode verificar o efeito de alterar cada um deles.

References

- [1] Hallison's github: [internet] Disponível em: <<https://github.com/hallpaz/Image-Processing-IMPAC-2015>>
- [2] Harris regions. [internet] Acessado em 21 de outubro de 2015. Disponível em: <http://opencv-python-tutroals.readthedocs.org/en/latest/py_tutorials/py_feature2d/py_features_harris/py_features_harris.html>