Visão por Computador - Guião 04

André Salgueiro, 50645 & Filipe Costa, 65092

***Resumo* - Resolução e conclusões sobre os exercícios do guião 04.**

***Abstract* - Resolution and conclusions on the exercise sheet 04.**

I. INTRODUÇÃO

Os temas para os exercícios deste guião incluem gradientes aplicados a imagens e, ainda, deteção de arestas e contornos.

II. RESOLUÇÃO E CONCLUSÕES

A. Exercício 1

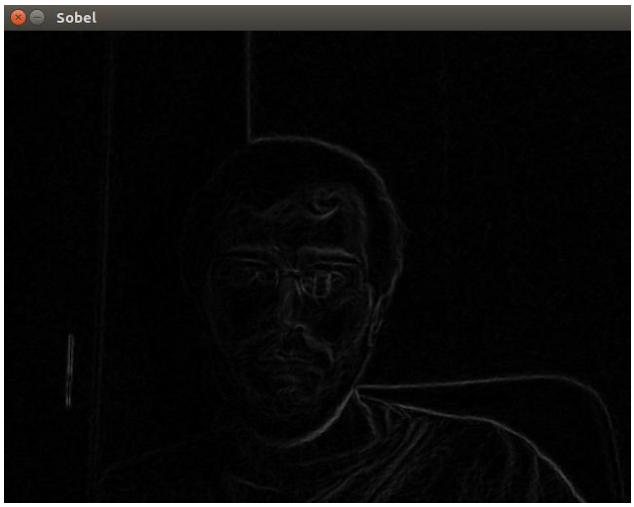

Fig. 1 - Imagem obtida a partir do algoritmo Sobel.

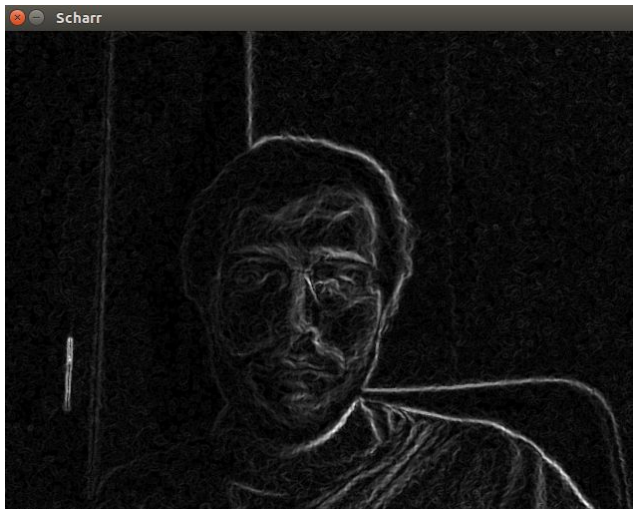

Fig. 2 - Imagem obtida a partir do algoritmo Scharr.

Para os valores padrão de scale = 1 e delta = 0 o algoritmo de scharr apresenta uma maior quantidade detalhes para uma imagem semelhante.

Alterando estes valores para ambos os algoritmos vamos obter maiores detalhes mas também mais ruído na imagem.

B. Exercício 2

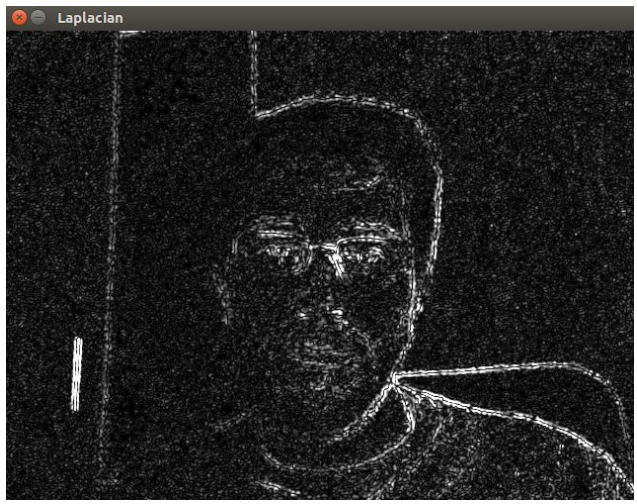

Fig. 3 - Imagem obtida a partir do algoritmo Laplacian com um kernel de tamanho 5.

Fig. 4 - Imagem obtida a partir do algoritmo Laplacian com um kernel de tamanho 7.

Para o algoritmo de Laplace começamos com um kernel de tamanho 3 mas como se notavam poucos detalhes decidimos utilizar kernel de tamanho 7. Mas para este tamanho a imagem apresenta uma grande quantidade de ruído.

Escolhemos então o kernel de tamanho 5 como sendo o que apresenta os melhores resultados.

C. Exercício 3

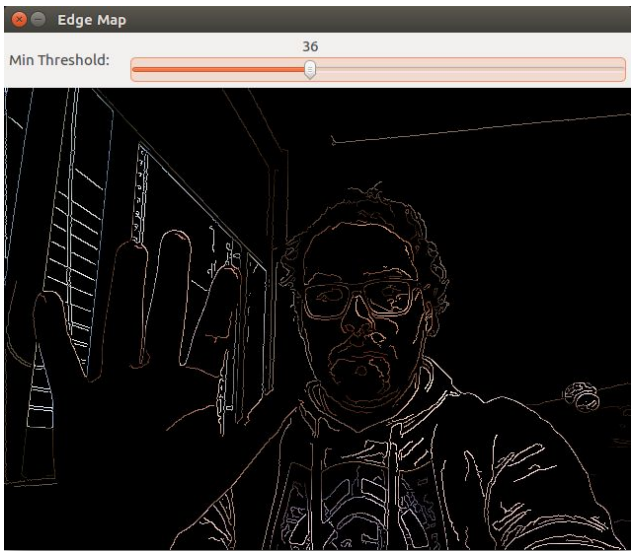

Fig. 5 - Imagem obtida a partir do algoritmo de Canny

Para a imagem utilizada, escolhemos manualmente o *threshold* de 36 apresentado uma grande quantidade de detalhes sem que surja ruído.

Alterando os valores do *threshold*, vemos que para um threshold mais próximo de 255 obtemos uma imagem com todo o detalhe mas com muito ruído. Já para valores mais próximos de zero perdem-se os detalhes.

D. Exercício 4

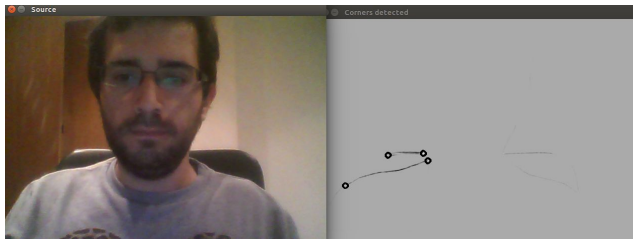

Fig. 6 - Imagem obtida a partir do algoritmo de Harris para deteção de cantos.

A imagem da esquerda é a imagem original e a imagem da direita apresenta os cantos, através de círculos, obtidos a partir da original.

Para aumentar a quantidade de cantos obtidos a partir da imagem poderíamos melhorar a iluminação ou ter uma transição entre cores para as quais houvesse uma maior diferença quando convertidas para a imagem em tons de cinzento.

E. Exercício 5

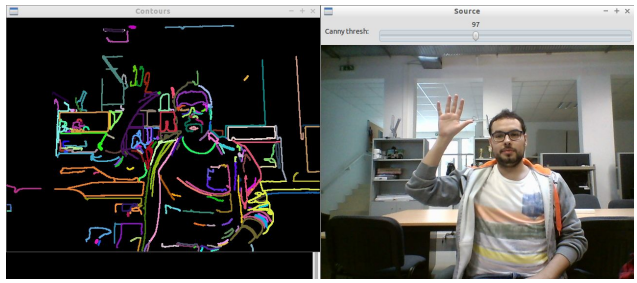

Fig. 7 - Imagem obtida a partir do método findingContours

Usando o método findingContours com um *threshold* 97 conseguimos uma imagem com contornos bem definidos em que cada cor representa um segmento. Com um valor de *threshold* inferior obtemos linhas maiores mas com bastante mais ruído.