

DETECÇÃO DE OBJETOS ATRAVÉS DE HARRIS E SURF

MANOEL VIEIRA COELHO NETO

14/0152512*

*SQS 203 Bloco J
Brasília, DF, Brasil

Email: vieiranetoc@gmail.com

1 Objetivos

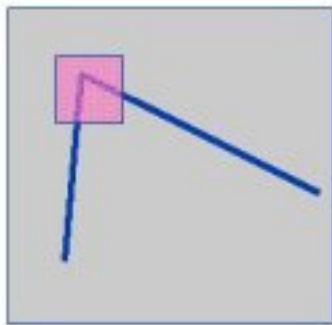
O presente relatório tem como objetivo implementar os métodos de Harris e SURF para extração de pontos de interesse de um objeto e detectá-lo em qualquer contexto em que estiver incluso.

2 Introdução

Os pontos de interesse de um objeto são extremamente importantes para a visão computacional em inúmeras aplicações, neste projeto demonstrativo são extraídos para que possamos detectar o mesmo objeto em diferentes contextos a partir desses pontos, comumente chamados *keypoints*.

Uma forma interessante de extrair os *keypoints* é através das quinas de um objeto, que nos fornece informações localizadas e extremamente úteis pois, onde há quina há o encontro de bordas. Uma boa forma de achar as quinas de um objeto é usando um detector de Harris.

A ideia básica consiste em reconhecer facilmente um ponto olhando os valores de intensidade em um *template*, após isso, ao avançar esse *template* em qualquer direção deverá existir uma variação muito grande em qualquer direção.



A partir disso caímos em três casos:

- ✓ Não há mudança em nenhuma direção.
- ✓ No caso de uma borda, não há mudança em uma direção.
- ✓ Em uma quina há mudanças significativas em todas as direções.

A função de Harris é dada por

$$E(u, v) = \sum w(x, y) [I(x + u, y + v) - I(x, y)]^2$$

Onde a função de pode ser dada por:

$$w(x, y) = 1 \text{ se pertencer ao } template \text{ e} \\ w(x, y) = 0 \text{ caso contrário}$$

Considerando a variação de $1px$, podemos dizer que Harris toma a variância do gradiente da imagem em uma direção, e os pontos de interesse é onde esse valor é alto.

Baseado no detector de Harris, há o SURF (*Speed Up Robust Detector*), que implementa o algoritmo de Harris, mas usando uma matriz Hessiana definida por

$$Hessian = \begin{vmatrix} L_{xx}(p, \theta) & L_{xy}(p, \theta) \\ L_{xy}(p, \theta) & L_{yy}(p, \theta) \end{vmatrix}$$

Onde L são as segundas derivadas de um ponto em uma direção em uma imagem em escala de cinza. O ponto de interesse é definido então e é aplicado um filtro de vizinhança ao redor do mesmo para que possamos definir a região de interesse desse objeto. Feito isso, basta procurar os pontos semelhantes na imagem do objeto.

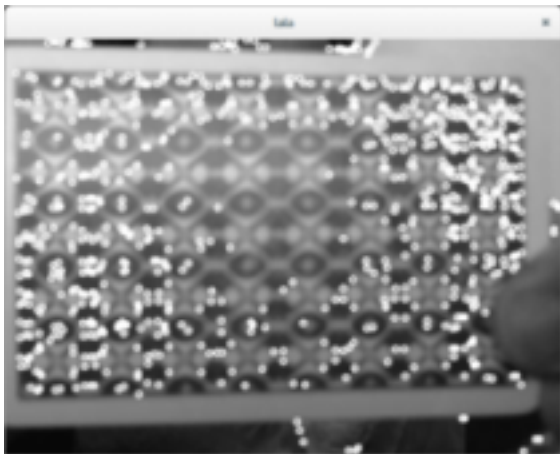
3 Metodologia

O código desenvolvido em C++ usando Opencv tem a parte do algoritmo de Harris (Resultado apresentado abaixo) e a parte de correspondência de pontos em seguida a qual foi feita usando SURF e um *matcher* baseado em Flann. Como sugerido na documentação da biblioteca. O código não está otimizado e apresenta-se extremamente lento. Primeiro, deve-se preencher o frame da câmera com o objeto desejado e clicar na janela, para que se possa capturar a imagem do objeto, logo após essa etapa o programa detecta os pontos de interesse e mapeia o objeto no vídeo baseado nos melhores pontos e desenhando uma linha ao redor dele. É importante frisar que o SURF é invariante à rotação do objeto. Para avançar da detecção de Harris para a comparação por SURF basta dar 'esc' na primeira janela.

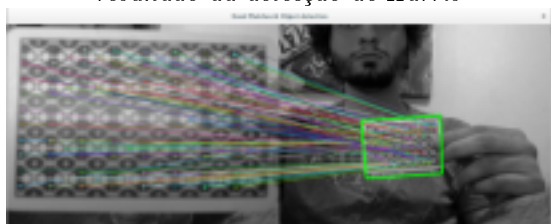
4 Resultados

Os resultados obtidos para cada etapa do objeto são apresentados a seguir.

Para o algoritmo de Harris, conseguimos o seguinte resultado:



resultado da detecção de Harris



resultado de SURF



Objeto rotacionado

5 Conclusão

Pode-se observar a grande quantidade de pontos de interesse extraídos de uma imagem bem detalhada pelo método de Harris, mas por haver o valor da distancia dos pontos de interesse e da região ao redor, é mais interessante um algoritmo mais robusto como SURF para comparação entre o objeto e a imagem, pode-se ver também que não há problema em ocluir parte do objeto ou rotacioná-lo nem mesmo mudar a escala altera a detecção, assim, temos um ótimo método para identificação de um objeto conhecido em um contexto qualquer

6 Bibliografia

LAGANIÈRE, Robert, 2011. OpenCV 2 Computer Vision Application Programming Cookbook
 COLLINS, Robert. Lecture 06: Harris Corner Detector. Disponível em:
<http://www.cse.psu.edu/~rtc12/CSE486/lecture06.pdf>