

## Visão por Computador

### 11 - Feature

Diogo Corte, Diogo Silva  
DETI, Universidade de Aveiro  
Aveiro, Portugal  
{diogo.corte, dbtds}@ua.pt

**Resumo** – Este relatório foi realizado para a unidade curricular de Visão por Computadores com o objetivo de demonstrar a perceção da aula (VC Exercises 11). Sendo que o objetivo é explorar algoritmos que permitam extrair features.

#### NOTA SOBRE AS IMAGENS

Apesar de as imagens terem aparência pequena, dispõem de boa resolução sendo que se for aplicado zoom sobre a imagem é possível ver com detalhe que se pretende.

#### I. DESCRIÇÃO DOS FICHEIROS

O conteúdo da pasta deste relatório contém os seguintes ficheiros:

1. hough.cpp corresponde ao ficheiro principal da representação da transformada de Hough;
2. hough\_representation.[cpp,h] corresponde aos ficheiros que guardam a representação da matriz acumulada de Hough.
3. siftExample.cpp corresponde ao ficheiro de correspondência de features com deteção SIFT (Scale-Invariant Feature Transform)
4. surfExample.cpp corresponde ao ficheiro de correspondência de features com deteção SURF (Speeded-Up Robust Features).

#### II. TRANSFORMADA DE HOUGH

##### A. Código desenvolvido

A transformada de Hough foi desenvolvida a partir de um código [1] já existente no GitHub do utilizador Saburo Okita (<https://github.com/subokita>) em que foram feitas algumas alterações para acrescentar funcionalidades.

O programa inclui as seguintes possibilidades:

- Representação da transformada de Hough completa ou com um determinado threshold (trackbar);
- Visualização do operador Canny e aplicar modificações no threshold (duas trackbars);
- Visualização de um ponto em específico da transformada de Hough na imagem real (linha verde).

##### B. Explicação Teórica

A transformada de Hough é uma representação de Rho por Theta. Sendo que Rho é a distância absoluta desde a

origem até ao ponto mais perto e o theta é o angulo que o eixo dos x faz com a recta que une a origem ao ponto em questão.

A transformada de Hough funciona como uma espécie de votação, sendo que quantos mais pontos tiverem numa determinada recta, maior será a intensidade num dado ponto na transformada de Hough, sendo fácil de visualizar rectas (pontos únicos de grande intensidade). De certa forma parecido com o RANSAC na sua forma de votação, recta com maior número de inliers é a considerada para fazer o fitting dos pontos em questão.

##### C. Explicação Prática

Nos seguintes exemplos é possível verificar sempre a transformada de Hough do lado direito e a imagem que está a ser visualizada do lado esquerdo.

##### C.1 Figura abstracta

Na imagem 1 é possível visualizar a verdade uma reta, sendo que foi marcado a amarelo o círculo a qual ela corresponde na transformada de Hough, se repararmos é dos pontos mais fortes do diagrama, sendo que é o que é dos que contém mais pontos. Os outros pontos de interesse foram marcados a vermelho, que são 3, mais o amarelo ficam 4. Estes 4 pontos representam as quatro retas do quadrado.

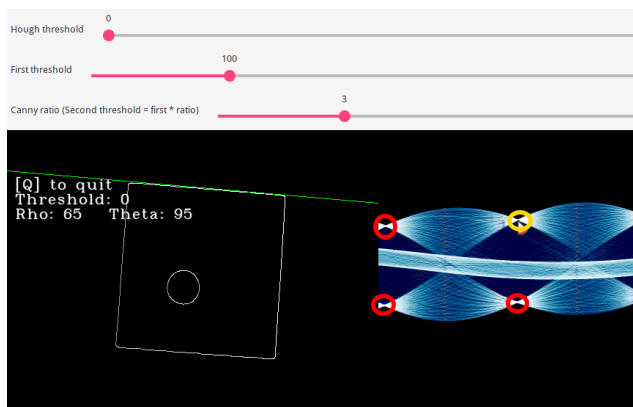


Figura 1: Representação de uma figura abstracta na transformada de Hough - Marcação de uma linha horizontal

Ao comparar a figura 1 com a figura 2 é possível visualizar

no texto que existe uma diferença de  $90^\circ$  no theta, como era de esperar. A diferença de ângulos entre uma recta vertical do quadrado e uma horizontal é de 90 graus.

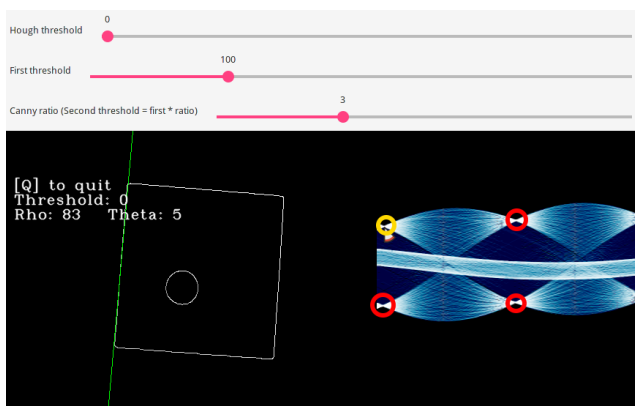


Figura 2: Representação de uma figura abstrata na transformada de Hough - Marcação de uma linha vertical

Para além das retas ainda existe um pequeno círculo no meio das figuras, este é representado por uma faixa enorme que atravessa toda a transformada de Hough. É possível reparar que a faixa contém uma linha superior mais branca e uma linha inferior mais branca também, uma corresponde ao semi-círculo da parte superior e a outra corresponde ao semi-círculo da parte inferior, respetivamente. Isto acontece porque o diagrama de Hough contém variações no theta desde 0 a 180 graus.

Para estimar o tamanho destes círculos é a diretamente proporcional a diferença entre a reta superior e a reta inferior da faixa.

### C.2 Tetromino de forma T

Para o seguinte exemplo (figura 3 talvez seja preciso zoom para visualizar os pontos da transformada de Hough) foi aplicado um threshold na representação da transformada de Hough para perceber melhor, deixando apenas pontos com mais influência.

Sendo que a vermelho são todas as rectas que foram incluídas para o diagrama de Hough, mais do que necessárias para representar o tetromino.

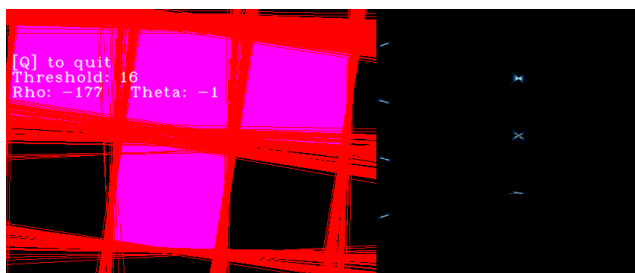


Figura 3: Retas de um Tetromino

Na transformada de Hough é possível verificar que existem apenas 7 pontos, mas um tetromino contém 8 retas. Isto

acontece porque a reta intermédia que é separada por um quadrado inferior corresponde ao mesmo ponto na transformada de Hough devido a terem exatamente a mesma equação (mesmo declive e mesma interseção com a origem).

## III. DETEÇÃO E CORRESPONDÊNCIA DE FEATURES

O código desenvolvido teve por base os tutoriais de OpenCV para o módulo features2D disponível em: [http://docs.opencv.org/3.0-beta/doc/tutorials/features2d/table\\_of\\_content\\_features2d/table\\_of\\_content\\_features2d.html](http://docs.opencv.org/3.0-beta/doc/tutorials/features2d/table_of_content_features2d/table_of_content_features2d.html)

### A. Detecção de Features

O OpenCV dispõe de vários algoritmos de deteção de features SIFT, SURF, FAST, entre outros. Neste exemplo [2] fornecido pelo OpenCV dão uso ao algoritmo SURF, que usamos e alteramos de forma a obter também os resultados do algoritmo SIFT.



Figura 4: Objecto e features do objecto

O algoritmo SURF tem tempos de execução mais rápidos no entanto as features encontradas caracterizam o objecto de uma forma menos precisa que o algoritmo SIFT.



Figura 5: Features SIFT na cena

Na figura 5 está a ser aplicado o algoritmo SIFT a uma cena com o objeto no mesmo plano 2D que o objecto original, em que grande parte das features detetadas tem uma

concentração espacial semelhante às features detetadas no objecto original.

### B. Correspondência de Features

Para correspondência de Features o OpenCV disponibiliza essencialmente 2 abordagens, Flann Based e Brute Force, esta ultima com várias implementações, estes algoritmos de correspondência são aplicados aos descritores das features, do objeto e da cena, visando encontrar o maior número de conjuntos binários de features com correlação espacial proporcional, sendo que neste exemplo [3] usa-se Flann Based.

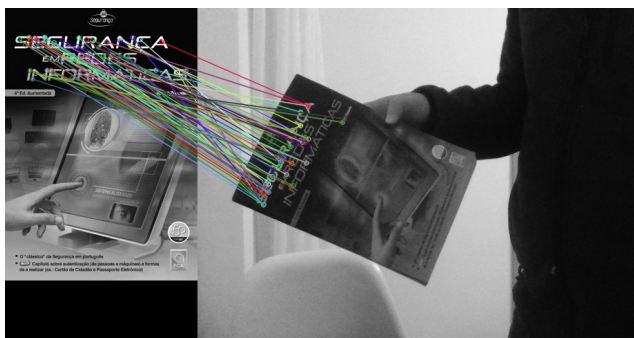


Figura 6: Forte correspondência de features SIFT

Na figura 6 está representado o conjunto de correspondências de features SIFT entre o objecto e a cena, tem um grande número de correspondências corretas o que se verifica também e com precisão aceitável, para deslocamentos de translação e rotação nos eixos dos xx e yy e translação no eixo zz.

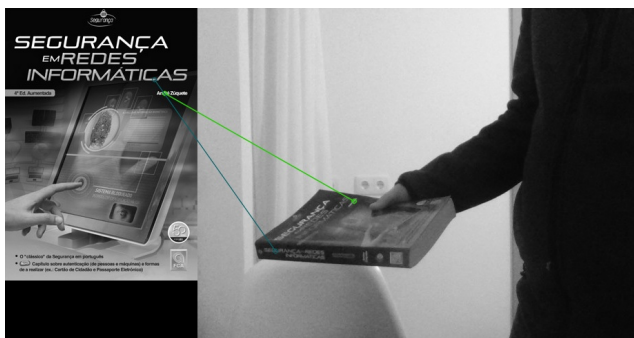


Figura 7: Fraca correspondência de features SIFT

Para movimentos no objecto de rotação no eixo zz ambos os algoritmos usados obtêm progressivamente piores resultados, tal como era esperado para algoritmos de detecção de features a duas dimensões.

### C. Homografia

Neste contexto homografia significa igualdade ou semelhança gráfica entre dois objectos. Dando uso às features e seus descritores das correspondências encontradas o OpenCV disponibiliza a função findHomography para localizar o objecto em cena, neste exemplo [4] para esse efeito usa-se o método RANSAC.

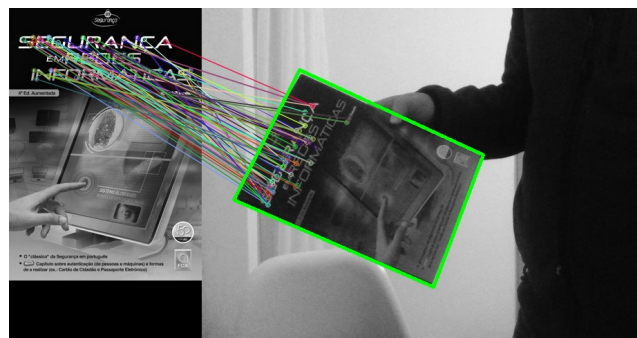


Figura 8: Detecção SIFT homográfica do objecto

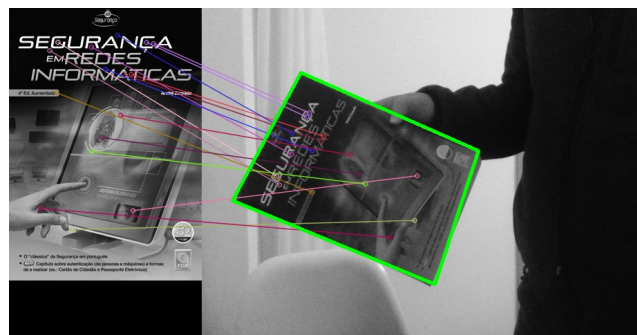


Figura 9: Detecção SURF homográfica do objecto

É possível verificar nas diferenças de caracterização do objecto entre a figura 8 e 9, o tradeoff entre velocidade de execução e precisão do resultado, sendo SURF o mais rápido e SIFT o mais preciso.

### REFERÊNCIAS

- [1] Transformada de Hough - Saburo Okita *GitHub*. Acedido a 30 de Novembro.
- [2] OpenCV tutorial: Detecção de features. Acedido a 5 de Dezembro.
- [3] OpenCV tutorial: Correspondência de features com Matcher FLANN. Acedido a 5 de Dezembro.
- [4] OpenCV tutorial: Detecção homográfica de um objecto conhecido. Acedido a 5 de Dezembro.