Projeto 3 de Princípios de Visão Computacional

Gabriel Martins de Miranda
130111350
Universidade de Brasília
Email:gabrielmirandat@hotmail.com

***Resumo*—A presente demonstração se baseia no algoritmo de oito pontos de Hartley. Foram realizadas duas abordagens. A primeira implementou o algoritmo não normalizado, enquanto que a segunda usou a função pronta do opencv que implementa o algoritmo de Hartley normalizado. O objetivo do projeto é visualizar as retas epipolares encontradas a partir da matriz fundamental que relaciona as imagens $mantle1$ e $mantle2$ dados oito pontos de correspondência entre elas.**

I. Introdução

Seguem alguns conceitos importantes para o entendimento do leitor:

$cameraMatrix$: conhecida como K, é uma matriz 3x3 que contém informações dos parâmetros intrínsecos da câmera. Quando multiplicada pela $poseMatrix$, matriz 3x4 formada da concatenação das matrizes de rotação e translação da cena, forma a $projectionMatrix$, matriz 3x4 que relaciona pontos numa imagem 2D com seus correspondentes pontos 3D em coordenadas do mundo.

$fundamentalMatrix$: conhecida por F, é uma matriz 3x3 que relaciona pontos correspondentes entre imagens estéreo. Pode ser estimada com no mínimo sete pontos de correspondência. Ela é obtida quando não são conhecidos os parâmetros intrínsecos da câmera, ou seja, a matriz K.

$essentialMatrix$: conhecida por E, assim como a matriz F, é uma 3x3 que relaciona pontos correspondentes em imagens estéreo assumindo câmera pinhole. Ao contrário da $fundamentalMatriz$, E é obtida quando os parâmetros intrínsecos da câmera são conhecidos e são necessários somente 5 parâmetros, sendo 3 de rotação e 2 de translação.

A equação que relaciona as matrizes fundamental e essencial é dada por: $F = K^{-T}EK'^{-1}$, onde K é a $cameraMatrix$ relacionada à primeira câmera e K' é a $cameraMatrix$ relacionada à segunda câmera.

$rank\ de\ uma\ matriz$: para definir o conceito de $rank$ de uma matriz, primeiro definiremos os conceitos de $row\ rank$ e $column\ rank$. Seja A uma matriz mxn qualquer. Assim, temos que o $row\ rank$ de A é o número máximo de linhas linearmente independentes de A, ou seja, linhas que não podem ser obtidas por combinação linear de umas com as outras. Intuitivamente, o $column\ rank$ de A é o número máximo de colunas linearmente independentes de A. Como A possui m linhas e n colunas, $row\ rank$ de $A <= m$ e $column\ rank$ de $A <= n$.Porém, temos que $row\ rank$ de $A = colums\ rank$ de A sempre!! Logo, falamos de rank da matriz. Então tiramos que rank(A_{mxn}) <= min(m, n).Atraves de operações entre as linhas, achamos a matriz minimizada. O número de linhas diferente de zero é o $rank\ da\ matriz$.

$Singular\ Value\ Decomposition\ (SVD)$: é a fatorização de uma matriz real ou complexa. Tomando por exemplo a mesma matriz A mxn usada anteriormente, o SVD de A é $A = U\Sigma V^T$, em que U é mxm e ortogonal (ou seja, sua transposta coincide com sua inversa), V é nxn e também ortogonal e Σ é uma matriz diagonal mxn com entradas diagonais não negativas dadas por $\sigma 1 >= \sigma_2 >= ... >= \sigma_p$, em que p = min(m, n). Estes valores são conhecidos como valores singulares de A. O SVM é uma decomposição extremamente útil que produz muitas informações sobre A, incluindo seu range, rank e espaço nulo. Além disto, podemos obter características úteis da álgebra linear no SVD. A saber, as m colunas de U e n colunas de V são os chamados vetores singulares esquerda e vetores singulares direta de A, respectivamente. Os vetores singulares esquerda de A são autovetores de AA*. Os vetores singulares direita de A são autovetores de A*A, e por fim, os valores singulares diferentes de zero de A são as raízes dos autovalores diferentes de zero de ambas A*A e AA*.

Diante do que foi explicitado, podemos finalmente entender como funciona o algoritmo de 8-pontos para cálculo da $matriz\ fundamental$.

II. Metodologia

De maneira geral, os passos para a realização do algoritmo de 8-pontos são:

- solução de mínimos quadrados usando SVD nas equações dos oito pares de correspondências. Daqui temos uma primeira estimativa para F, que chamaremos de $F \sim$.
- Otimizar $F \sim$ forçando a restrição de que a matriz fundamental F é rank 2, o que implica que det(F) = 0. Para isto, usamos o SVD na própria $F \sim$. O resultado é a matriz fundamental F que procuramos.

Nos aprofundando mais neste algoritmo, temos o seguinte:

1) Resolver um sistema de equações lineares.

- Escrever o sistema de equações

$$x^T F x' = 0$$

Abrindo esta equação, ficamos com

$$x'xf_{11} + x'yf_{12} + x'f_{13} + y'xf_{21} + y'yf_{22} + y'f_{23} + xf_{31} + yf_{32} + f_{33}$$

de onde tiramos a seguinte multiplicação matricial

$(lembrando\, que\, devemos\, fazer\, para\, todos\, os\, 8\, pontos)$

$$\begin{pmatrix} x'_1x_1 & x'_1y_1 & x'_1 & y'_1x_1 & y'_1y_1 & y'_1 & x_1 & y_1 & 1 \\ x'_2x_2 & x'_2y_2 & x'_2 & y'_2x_2 & y'_2y_2 & y'_2 & x_2 & y_2 & 1 \\ x'_3x_3 & x'_3y_3 & x'_3 & y'_3x_3 & y'_3y_3 & y'_3 & x_3 & y_3 & 1 \\ x'_4x_4 & x'_4y_4 & x'_4 & y'_4x_4 & y'_4y_4 & y'_4 & x_4 & y_4 & 1 \\ x'_5x_5 & x'_5y_5 & x'_5 & y'_5x_5 & y'_5y_5 & y'_5 & x_5 & y_5 & 1 \\ x'_6x_6 & x'_6y_6 & x'_6 & y'_6x_6 & y'_6y_6 & y'_6 & x_6 & y_6 & 1 \\ x'_7x_7 & x'_7y_7 & x'_7 & y'_7x_7 & y'_7y_7 & y'_7 & x_7 & y_7 & 1 \\ x'_8x_8 & x'_8y_8 & x'_8 & y'_8x_8 & y'_8y_8 & y'_8 & x_8 & y_8 & 1 \end{pmatrix} \begin{pmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{pmatrix} = 0$$

$ou \qquad Af = 0$

- Resolver f de Af=0 usando SVD. A primeira estimativa de F, a $F \sim$, é obtida da matriz V^T da decomposição SVD de A. A última linha de nove elementos de V^T corresponde aos nove elementos de $F \sim$ distribídos em 3x3.

```
//Computa SVD.
Mat w, u, vt;
SVD::compute(A, w, u, vt);

//Da matriz vt (right singular transposta), encontramos a primeira estimativa de F
Mat F1 = (Mat_<float>(3,3) << vt.at<float>(7,0),vt.at<float>(7,1),vt.at<float>(7,2),
      »       »       »       »       vt.at<float>(7,3),vt.at<float>(7,4),vt.at<float>(7,5),
      »       »       »       »       vt.at<float>(7,6),vt.at<float>(7,7),vt.at<float>(7,8));
```

Fig. 1: Estimativa de $F \sim$.

2) Resolver a restrição que det(F)=0, usando o SVD.

- Precisamos forçar a restrição de singularidade.
  A matriz fundamental tem rank 2: det(F)=0
  Fazer com que as linhas epipolares coincidem num mesmo ponto.

```
//Computa SVD em F1 - segunda estimativa - considerando rank 2.
Mat w2, u2, vt2;
SVD::compute(F1, w2, u2, vt2);

Mat W2 = Mat::zeros(3,3,CV_32F);
W2.at<float>(0,0) = w2.at<float>(0,0);
W2.at<float>(1,1) = w2.at<float>(1,0);

//Segunda estimativa de F, restrição de que det(F1)=0, rank 2.
Mat F2 = u2 * W2 * vt2;
```

Fig. 2: Estimativa de F.

3) Um adicional importante que se pode fazer é resolver o algoritmo em sua forma *normalizada*, já que não foram considerados fatores de escala. Para isto podemos apenas chamar a função pronta do opencv especificando a flag CV_FM_8POINT, que calcula o algoritmo de 8-pontos normalizado de Hartley.

- Normalizamos as coordenadas da imagem com $\tilde{x}$ = Tx e $\tilde{x'}$ = $T'x'$
- Desnormalizamos as coordenadas da imagem com F = $T'^{-1}\tilde{F}T$

```
//calcula estimativa da matriz fundamental. Aqui é usado o
//algoritmo 8-pontos normalizado de Hartley
Mat F = findFundamentalMat(pontos_1, pontos_2, CV_FM_8POINT);
```

Fig. 3: Estimativa de F com algoritmo de 8-pontos normalizado.

III. Resultados

1) Aqui são mostrados os resultado obtidos e a estimativa das retas epipolares:

Fig. 4: imagem mantle1.

Fig. 5: imagem mantle2.

Fig. 6: 8 pontos de correspondência de mantle1.

Fig. 7: 8 pontos de correspondência de mantle2.

Fig. 8: epipolares sem restrição rank2 não normalizado.

Fig. 9: epipolares com restrição rank2 não normalizado.

Fig. 10: epipolares com restrição e normalizado.

IV. Conclusão

Através do algoritmo aqui proposto, é possível obter uma boa estimativa para as retas epipolares através da matriz fundamental encontrada pelo método de Hartley. Foi possível perceber que há uma grande diferença em se tratando do algoritmo não-normalizado com o normalizado, que foi bem mais preciso.

V. Referências

[1] http://robotics.stanford.edu/ birch/projective/node20.html

[2] http://answers.opencv.org/question/27155/from-fundamental-matrix-to-rectified-images/

[3] http://answers.opencv.org/question/38340/estimate-camera-pose- extrinsic-parameters-from-homography-essential-matrix/

[4] https://www8.cs.umu.se/kurser/TDBD19/VT05/reconstruct-4.pdf

[5] http://web.mit.edu/be.400/www/SVD/Singular_Value_ Decomposition.htm

[6] http://www.hasper.info/opencv-draw-epipolar-lines/

[7] http://www.cs.utexas.edu/users/inderjit/public_papers/HLA_SVD.pdf
[8] http://docs.opencv.org/master/da/de9/tutorial_py_ epipolar_geometry.html
[9] http://www.cliffsnotes.com/math/algebra/linear-algebra/real-euclid ean-vector-spaces/the-rank-of-a-matrix
[10] http://answers.opencv.org/question/36751/kmeans-clustering-for-vec torpoint2f-data-structure/
[11] http://arxiv.org/pdf/1403.4806.pdf
[12] http://www.quora.com/Why-is-the-fundamental-matrix-in-computer-visi on-rank-2
[13] http://en.wikipedia.org/wiki/Fundamental_matrix_
[14] http://www.emgu.com/wiki/files/1.3.0.0/html/55d6f4d2-223d-8c55- 2770-2b6a9c6eefa2.htm
[15] http://stackoverflow.com/questions/12029486/matlab-svd-output- in-opencv
[16] http://en.wikipedia.org/wiki/Singular_value_ decomposition#Applications_of_the_SVD
[17] https://www.youtube.com/watch?v=JEYLfIVvR9I
[18] http://docs.opencv.org/master/df/df7/classcv_1_1SVD.html#details
[19] http://websites.uwlax.edu/twill/svd/svd/index.html
[20] http://web.stanford.edu/class/cme335/lecture6.pdf
[21] http://www.cs.carleton.edu/cs_comps/0607/recommend/recommender /svd.html
[22] http://en.wikipedia.org/wiki/Singular_value_decomposition
[23] http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.114.2762