Project 4 Writeup

Dupla

Renan Viana Hoshi

Felipe Torres Minorelli

Instructions

- Describe any interesting decisions you made to write your algorithm.
- Show and discuss the results of your algorithm.
- Feel free to include code snippets, images, and equations.
- Use as many pages as you need, but err on the short side If you feel you only need to write a short amount to meet the brief, th
- **Please make this document anonymous.**

Introdução

Neste trabalho temos como objetivo a estimação de uma matrix fundamental para um par de imagens reais, e usá-las para a rejeitar as falsas correspondências entre os pontos semelhantes das mesmas. Para alcançar tal finalidade, o algoritmo RANSAC foi utilizado para a melhoria de equivalência dos pontos.

Implementação

O objetivo deste projeto é apresentar a câmera e a geometria da cena. Especificamente, estimaremos a matriz fundamental, que relaciona pontos em uma cena a linhas epipolares em outra. A matriz fundamental pode ser estimada usando correspondências pontuais. Para estimar a matriz fundamental, a entrada corresponde a 2d pontos em duas imagens. Iremos estimar a matriz fundamental usando as correspondências pontuais de SIFT e RANSAC.

Código utilizado para realização da função RANSAC:

```
% RANSACFUNCTION

function [Best_Fmatrix, inliers_a, inliers_b] =
   ransac_fundamental_matrix(matches_a, matches_b)


%%%%%%%%%%%%%%%%
% Your code here
%%%%%%%%%%%%%%%%

%num iteracoes
numIter = 5000;

%auxiliar para carregar o melhor inlier obtido a cada
   iteracao
aux = 0;

%limite de distancia especificado pelo usuario
limit = 0.05;

sizeA = size(matches_a, 1);
sizeB = size(matches_b, 1);

%melhor matrix, a que obter a maior quantidade de inliers
Best_Fmatrix = [];
inliers_a = [];
inliers_b = [];

    for i = 1 : numIter
              %coleta as amostras, sizeB ou sizeA podem
                 ser utilizados, 10 amostras estao sendo
                 coletadas
        n = randsample(sizeB, 10);
        pointA = matches_a(n, :);
        pointB = matches_b(n, :);

        %estima a matrix fundamental das amostras
           coletadas
        F = estimate_fundamental_matrix(pointA, pointB);

        newInA = [];
        newInb = [];

        for j = 1 : sizeB
            pointA = [matches_a(j, :) 1];
```

```
41              pointB = [matches_b(j, :) 1];
42
43              %calculando a metrica de distancia
44              err = pointB * F * pointA';
45
46              %se o valor obtido for menor que o limite
                   especificado, os inliers sao atualizados
47              if (abs(err) < limit)
48                  newInA = [newInA; pointA(:, 1:2)];
49                  newInb = [newInb; pointB(:, 1:2)];
50              end
51          end
52
53          %seleciona os melhores inliers e a matrix
               fundamental
54          if (size(newInA, 1) > aux)
55              inliers_a = newInA;
56              inliers_b = newInb;
57              Best_Fmatrix = F;
58              aux = size(newInA, 1);
59          end
60
61      end
62
63  end
```

Resultados

Os resultados utilizando a imagem do Mount Rushmore podem ser vizualidos nas figuras 1 e 2. Esses resultados foram obtidos utilizando 5000 iterações na função RANSAC e limite de 0.05.

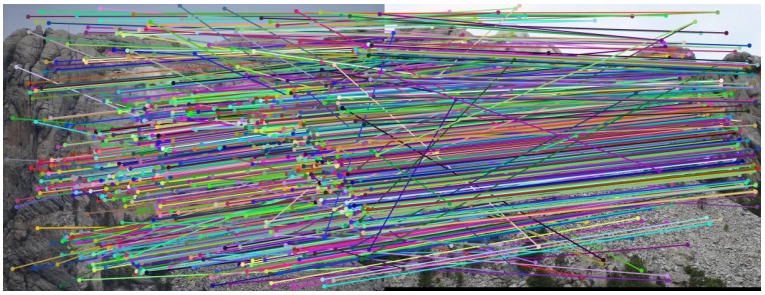

Figure 1: Sem a utilização do RANSAC

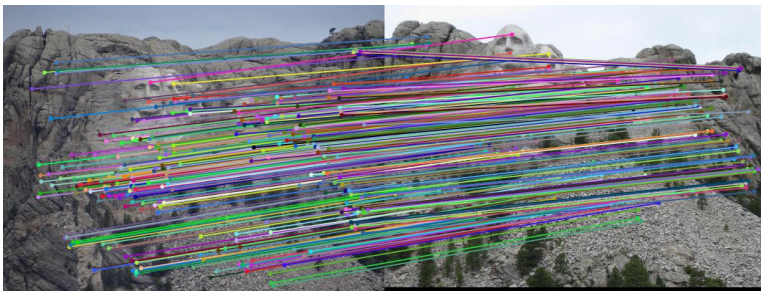

Figure 2: Utilizando a função do RANSAC

Dificuldades

Uma das maiores dificuldades deste projeto foi a instalação do VLFEAT, assim como a instalação do pacote *statistics* do *Octave*, já que a ferramenta, o pacote, e o software se utilizão de versões diferentes das bibliotecas durante a instalação.

Outra dificuldade foi encontrar a solução para a questão Q2 no livro de Szeliski, o artigo *Epipolar Geometry*[1] foi utilizado para a resolução desta questão.

[1] https://docs.opencv.org/3.2.0/da/de9/tutorial$_p y_e pipolar_g eometry.html$