

Project X Writeup

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.

Implementação

Esta seção apresenta alguns detalhes relevantes da implementação do algoritmo RANSAC. O Código 1 ilustra o algoritmo/passos utilizados para determinar a matriz de homografia:

- **Linha 1:** definição da função da distância.
- **Linha 4:** seleção de 8 *indexes* aleatórios.
- **Linha 5:** determinação da matriz de homografia *matrix* a partir dos 8 pontos selecionados.
- **Linha 6:** matriz que armazena as distâncias calculadas.
- **Linha 7:** aplicação da matriz de homografia nas coordenadas.
- **Linhas 9 a 13:** cálculo das distâncias por meio da fórmula da função *ssd* definida na *Linha 1*.
- **Linhas 15 e 16:** obtenção dos *indexes* que possuem distância inferior ao *threshold* definido, como também da quantidade de *inliers*.
- **Linhas 17 a 22:** atualização dos valores (se a quantidade de *inliers* for superior em determinada interação).
- **Linhas 27 a 29:** recálculo da matriz de homografia considerando todos os *inliers* obtidos, além da definição dos valores de retorno da função (*inlier_ind* e *H*).

Listing 1: Algoritmo RANSAC para determinação da matriz de homografia *matrix*.

```
1  ssd = @(x, y) sum((x-y).^2);  
2  
3  while i <= ransacIterations  
4      randSample = randsample(matches_size, numberOfPoints  
                               );
```

```
5     matrix = est_homography( x1(randSample), y1(
6         randSample), x2(randSample), y2(randSample) );
7     distance = zeros( matches_size, 1 );
8     [x, y] = apply_homography(matrix, x2, y2);
9
10    for k = 1:matches_size
11        a1 = [x(k); y(k)];
12        a2 = [x1(k); y1(k)];
13        distance(k) = ssd(a1, a2);
14    end
15
16    idxes = find( abs(distance) <= thresh );
17    futureInliers = size( idxes, 1 );
18    if ( futureInliers > presentInliers )
19        presentInliers = futureInliers;
20        MATRIX = matrix;
21        IDXES = idxes;
22        distances = distance;
23    end
24
25    i = i + 1;
26 end
27 MATRIX2 = est_homography( x1(IDXES), y1(IDXES), x2(IDXES)
28     , y2(IDXES) );
29 inlier_ind = IDXES;
30 H = MATRIX2;
```

Resultados

Esta seção apresenta os resultados obtidos.

1. Balcony: Figura 1 (imagem final).

- Threshold = 0.007.
- Iterações = 5000.



Figura 1: Resultado da imagem final.

2. Laboratory: Figura 2 (imagem final).

- Threshold = 0.05.
- Iterações = 5000.

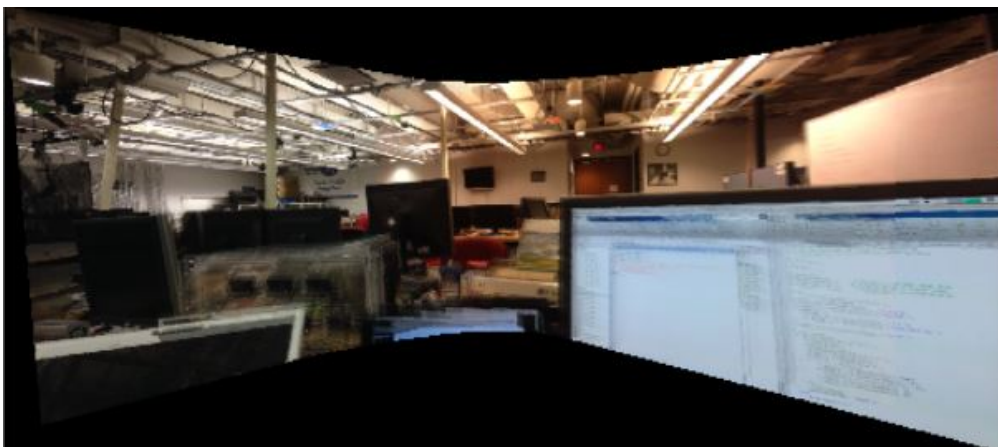


Figura 2: Resultado da imagem final.

3. Plants (minhas imagens): Figuras 3 (imagem parcial) e 4 (imagem final).

- Threshold = 5.
- Iterações = 5000.



Figura 3: Imagem parcial.



Figura 4: Resultado da imagem final.

4. Front (minhas imagens): Figuras 5 (imagem parcial 1), 6 (imagem parcial 2), 7 (imagem parcial 3) e 8 (imagem final).

- Threshold = 7.
- Iterações = 5000.

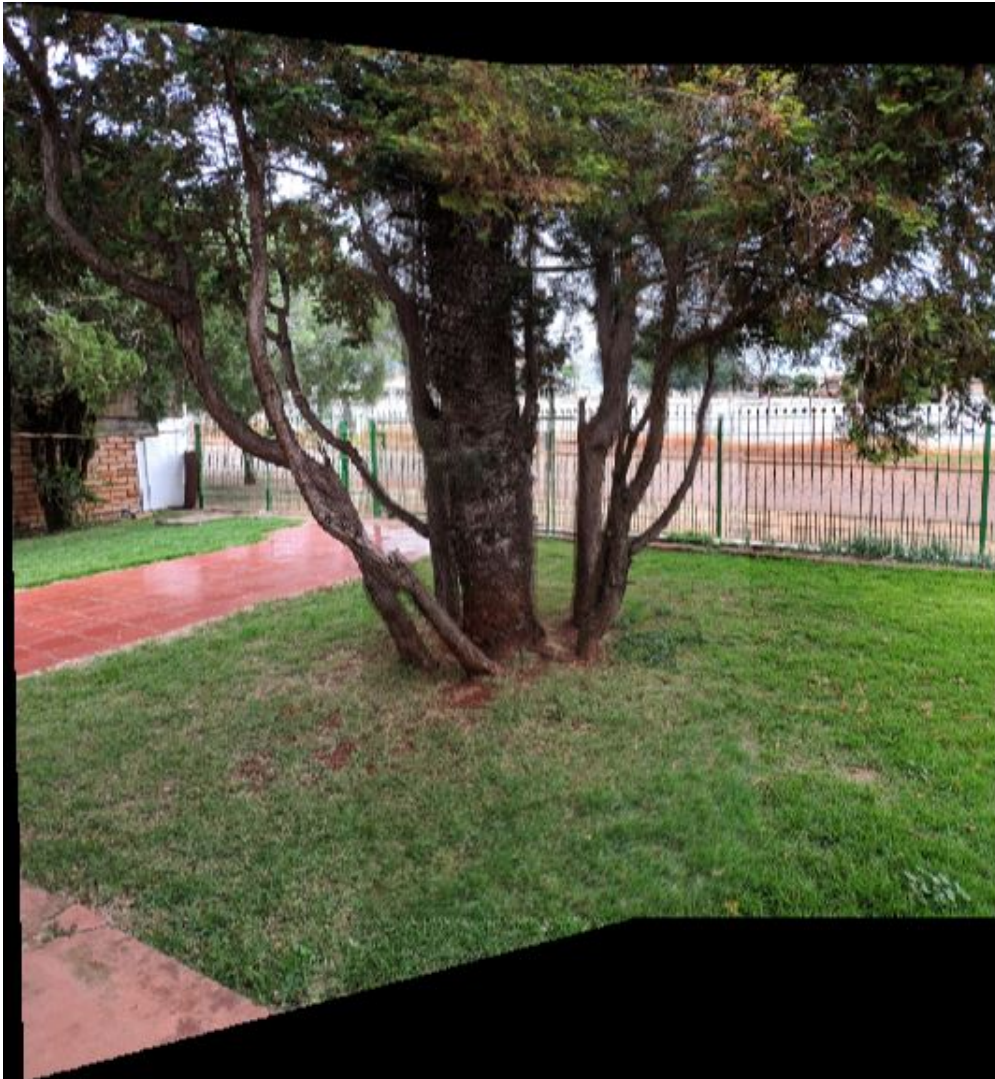


Figura 5: Imagem parcial 1.

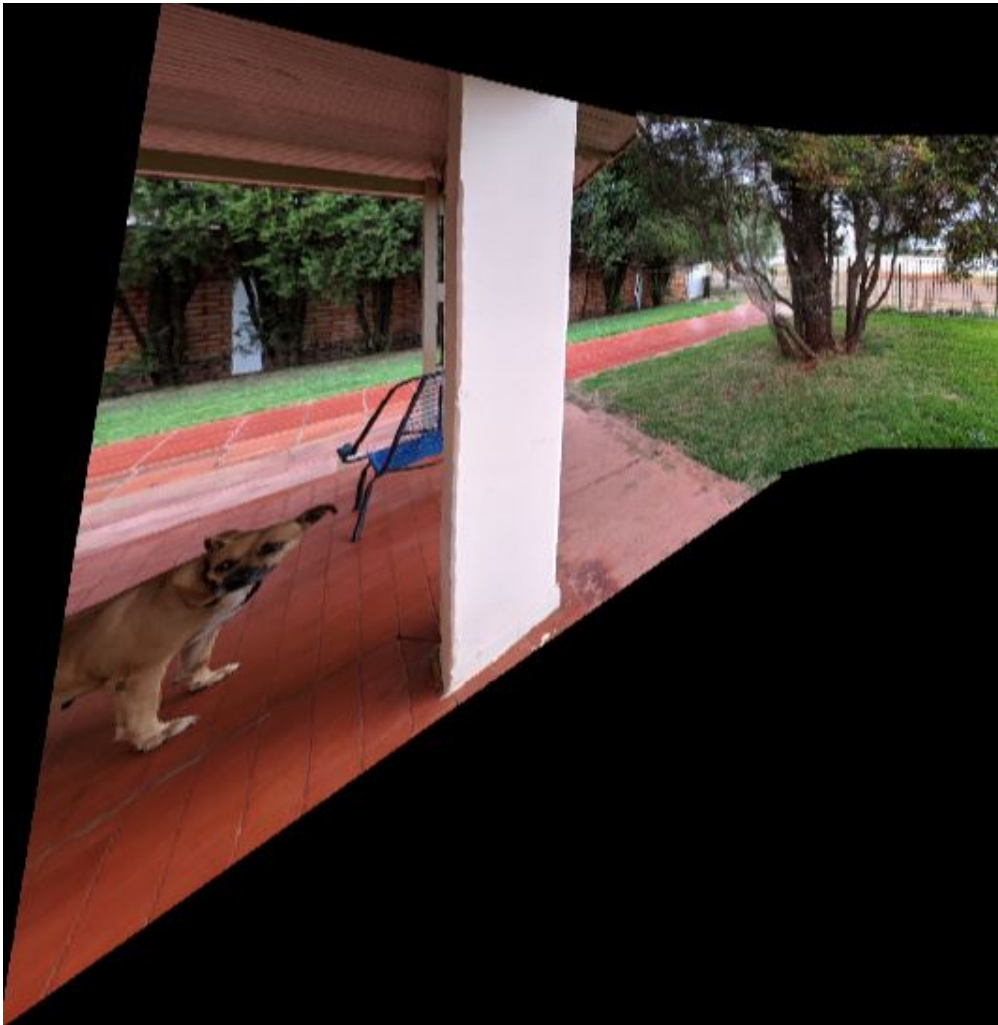


Figura 6: Imagem parcial 2.

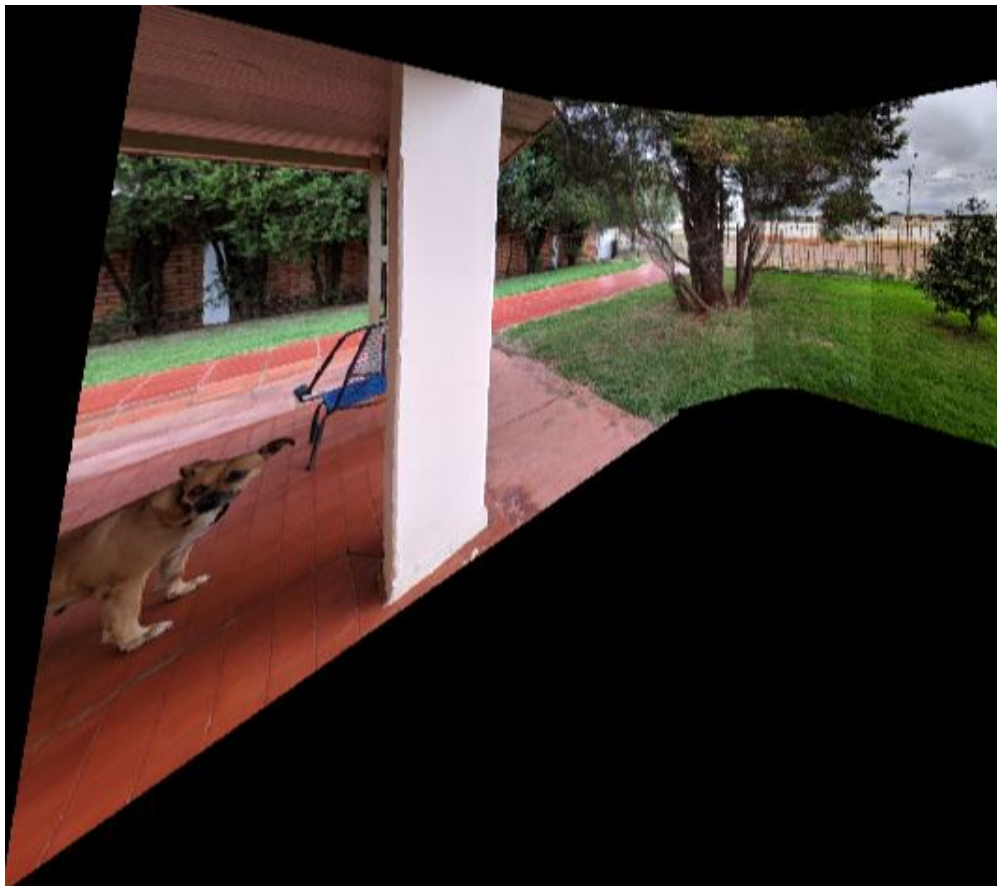


Figura 7: Imagem parcial 3.



Figura 8: Resultado da imagem final.