IMAGE MOSAICS

(Computer Vision)

Mohamed Murad 68
Youssif Mubark 88

Project Report

TABLE OF CONTENTS

1] GET CORRESPONDENCES

- getCorrespondence function is used to get pairs of points in the given source and destination images, number of pairs to find is pass as an argument beside the boolean manualSelection which is used internally to check if to get pair from the user user by clicking on the shown figure, or find pairs automatically.
- in the case of automatic we just pass set of pre-determined pairs.

2] HOMOGRAPHY CALCULATION

- Given n pairs of matched points, we can easily fit a model to represent them.

$$\begin{pmatrix} x'_1 \\ y'_1 \\ x'_2 \\ y'_2 \\ x'_3 \\ y'_3 \\ x'_4 \\ y'_4 \end{pmatrix} = \begin{pmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x_1x'_1 & -y_1x'_1 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -x_1y'_1 & -y_1y'_1 \\ x_2 & y_2 & 1 & 0 & 0 & 0 & -x_2x'_2 & -y_2x'_2 \\ 0 & 0 & 0 & x_2 & y_2 & 1 & -x_2y'_2 & -y_2y'_2 \\ x_3 & y_3 & 1 & 0 & 0 & 0 & -x_3x'_3 & -y_3x'_3 \\ 0 & 0 & 0 & x_3 & y_3 & 1 & -x_3y'_3 & -y_3y'_3 \\ x_4 & y_4 & 1 & 0 & 0 & 0 & -x_4x'_4 & -y_4x'_4 \\ 0 & 0 & 0 & x_4 & y_4 & 1 & -x_4y'_4 & -y_4y'_4 \end{pmatrix} \cdot \begin{pmatrix} a_{11} \\ a_{12} \\ a_{13} \\ a_{21} \\ a_{22} \\ a_{23} \\ a_{31} \\ a_{32} \end{pmatrix}$$

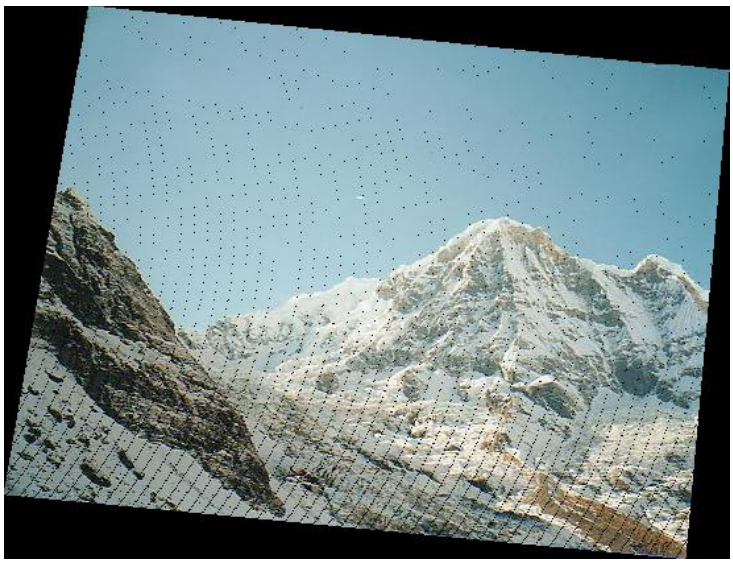
The above figure shown for only 4 pairs, our target is to find the 8 unknown a's

- in our code we do that by calcH() function. First we calculate matrix B (the first shown vector), then matrix A with the help of matrix getSubMatrixA() function which calculate 2 rows at a time to build this matrix.
- Our objective is to caculate H in AH=B, we do that with the built-in function linalg.lstsq which take A, and B as arguments and return H

3] WARPING, MERGE AND RESULTS

- To do this task we use warpAndMerge() function, which take the two images and H.
- first we get the corners of our destination image, transform them using H, calculate max and min width and height for the transformed image.

- We save the vertical and horizontal shifts because of out of bounds pixels during transformation.
- Now we initialize a new tensor (for the new warped image), initialized with zeros.
- To fill it we have two methods
- Method1: is to iterate for each pixels in the reference image and then transform it. But it’s bad as the result picture will have many black holes (no matched pair in the source image)



- Method2: overcome this problem but it’s more time costly. We iterate for each pixel in the destination image, and find it’s pair in the reference image buy using inverseTransform() function.
  We interpolate if the pair not localized in an absolute pixel

- finally we calculate the new width and height needed to merge the reference and warped image and save.

- Another Example