Computer Vision AI – Assignment 1
Homography and Image Stitching

Thursday 2nd April, 2015

The practical course consists of four assignments which will be combined under one final report at the end of the course (exact date of delivery will be specified later). The results, analysis and the source code must be included in the final delivery. Students should work on the assignments in groups of 2 people. Students are supposed to work on this (first) assignment for two weeks.

1 Homography

In this assignment you will write a function that takes two images as input and computes the homography (projective transformation) between them. You will work with supplied boat images. The overall scheme is as follows:

1. Detect interest points in each image.
2. Characterize the local appearance of the regions around interest points.
3. Get the set of supposed matches between region descriptors in each image.
4. Perform RANSAC to estimate the homography between images.

The first three steps can be performed using VLFeat functions (to download check http://www.vlfeat.org/download.html).

In the next stage, the homography matrix H can be estimated with RANSAC as follows:

- Repeat N times:
- Pick P matches at random from the total set of matches T.
  How many matches do we need to estimate a homography matrix formulated as shown in Figure 1? Please, write the answer in your report.
- Construct a matrix A and vector $\mathbf{b}$ using the P pairs of points and find transformation parameters $(h_{11}, h_{12}, h_{13}, h_{21}, h_{22}, h_{23}, h_{31}, h_{32})$ by solving the linear equation $A\mathbf{x} = \mathbf{b}$. Hint: Linear equation $A\mathbf{x} = \mathbf{b}$ can be solved using pseduo-inverse: $\mathbf{x} = (A^T A)^{-1} A^T \mathbf{b}$, or use `numpy.linalg.pinv(A)` to obtain the pseudo-inverse of A in Python.

Given a point (x, y) in one image and its corresponding point (x', y') in the other image, the homography matrix H holds

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix}}_{H} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{1}$$

It can be rewritten as

$$\begin{bmatrix} x & y & 1 & 0 & 0 & 0 & -x'x & -x'y \\ 0 & 0 & 0 & x & y & 1 & -y'x & -y'y \end{bmatrix} \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \end{bmatrix} = \begin{bmatrix} x' \\ y' \end{bmatrix} \tag{2}$$

or

$$A\mathbf{x} = \mathbf{b},\ A = \begin{bmatrix} x & y & 1 & 0 & 0 & 0 & -x'x & -x'y \\ 0 & 0 & 0 & x & y & 1 & -y'x & -y'y \end{bmatrix}, \mathbf{x} = \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \end{bmatrix}, \mathbf{b} = \begin{bmatrix} x' \\ y' \end{bmatrix} \tag{3}$$

Figure 1: Homograhpy estimation.

- Using the transformation parameters, transform the locations of all T points in image 1. If the transformation is correct, they should lie close to their counterparts in image 2. Plot the two images side by side with a line connecting the original T points in image 1 and transformed T points over image 2.

- Count the number of inliers, inliers being defined as the number of transformed points from image 1 that lie within a radius of 10 pixels of their pair in image 2.

- If this count exceeds the best total so far, save the transformation parameters and the set of inliers.

- End repeat.

- Finally, transform image 1 using this final set of transformation parameters. If you display this image you should find that the pose of the object in the scene should correspond to its pose in image 2.

How many iterations on average are needed to find a good estimation of homography? Please, write the answer in your report.

A new improved version of RANSAC, the locally optimized RANSAC (LO-RANSAC), was introduced in [1]. You can also implement it and compare with standard RANSAC.

2 Image Stitching

In this assignment you will write a function that takes two images as input and stitch them together. You will work with supplied images *bus_left.jpg* and *bus_right.jpg*. The overall scheme can be summarized as follows:

1. As in previous task you should first find the best transformation between input images.

2. Then you should estimate the size of the stitched image. Hint: calculate the transformed coordinate of corners of the *bus_right.jpg*.

3. Finally, combine the *bus_left.jpg* with the transformed *bus_right.jpg* in one image.

References

[1] Chum, O., Matas, J., Kittler, J.: Locally optimized ransac. In: Pattern Recognition (2003)