Exercises in Tracking & Detection

Exercise 1 Normalized Direct Linear Transformation (DLT)

For planar scenes each possible transformation can be expressed with a homography. An homography is a mapping of four points to the same four points in another image by the equation:

$$\mathbf{x}'_i \propto \mathbf{H}\mathbf{x}_i \tag{1}$$

where $\mathbf{x}_i$ is the point in the reference image, $\mathbf{x}'_i$ the corresponding point in the warped image and $\mathbf{H}$ the corresponding homography. In order to compute the homography, at least four points are necessary. However, in order to make it more robust to image and detection noise, the computation becomes more stable and reliable the more points are used.

One fast way to compute the homography in a least squares sense is to use the Normalized Direct Linear Transform (normalized DLT - for more details have a look to "Multiple View Geometry" by Richard Hartley and Andrew Zissermann, p.88 Section 4.1).

The Normalized Direct Linear Transformation Algorithm computes a homography for a projective transformation by using at least 4 point correspondences $\{\mathbf{x}'_i \leftrightarrow \mathbf{x}_i\}$ and minimizing the norm $\|\mathbf{A}\mathbf{h}\|$, where $\mathbf{A}$ contains the stacked-up constraints given by the homography equation $\mathbf{x}'_i \propto \mathbf{H}\mathbf{x}_i$ for all point correspondences and $\mathbf{h}$ is a (9×1) vector consisting of the entries of $\mathbf{H}$.

The steps of the normalized DLT algorithm are summarized as follows:

- Normalize the points of the one image with the transformation $\mathbf{U}$ such that their centroid is at the origin and that the average distance from the origin is equal to $\sqrt{2}$.
- Normalize the points of the warped image with the transformation $\mathbf{T}$ such that their centroid is at the origin and that the average distance from the origin is equal to $\sqrt{2}$.
- For each normalized point correspondence $\widetilde{\mathbf{x}}'_i \leftrightarrow \widetilde{\mathbf{x}}_i$ compute the matrix $\mathbf{A}_i$ given by the homography equation $\widetilde{\mathbf{x}}'_i \propto \mathbf{H}\widetilde{\mathbf{x}}_i$. The matrix $\widetilde{\mathbf{A}}_i$ has the explicit form:

$$\widetilde{\mathbf{A}}_i = \begin{pmatrix} 0 & -\widetilde{w}'_i\widetilde{\mathbf{x}}^\top & \widetilde{y}'\widetilde{\mathbf{x}}^\top \\ \widetilde{w}'\widetilde{\mathbf{x}}^\top & 0 & -\widetilde{x}'\widetilde{\mathbf{x}}^\top \end{pmatrix} \tag{2}$$

  where the normalized point in reference image is $\widetilde{\mathbf{x}}$ and in the warped image is $\widetilde{\mathbf{x}}'_i = (\widetilde{x}'_i\widetilde{y}'_i\widetilde{w}'_i)^\top$.

- Assemble the matrices $\mathbf{A}_i$ into a single $(2n \times 9)$ matrix $\mathbf{A}$. The final matrix $\widetilde{\mathbf{A}}$ has the form

$$\widetilde{\mathbf{A}} = \begin{pmatrix} \widetilde{\mathbf{A}}_1 \\ \widetilde{\mathbf{A}}_2 \\ \vdots \\ \widetilde{\mathbf{A}}_n \end{pmatrix} \tag{3}$$

- The solution $\widetilde{\mathbf{A}}\widetilde{\mathbf{h}} = 0$ is found via SVD ($\widetilde{\mathbf{h}}$ is the unit singular vector corresponding to the smallest singular value). Reshaping the (9×1) vector $\widetilde{\mathbf{h}}$ gives the homography $\widetilde{\mathbf{H}}$. After reassembling the matrix $\widetilde{\mathbf{H}}$ we have found a solution to $\widetilde{\mathbf{x}}'_i = \widetilde{\mathbf{H}}\widetilde{\mathbf{x}}_i$ where actually a solution to $\mathbf{x}'_i = \mathbf{H}\mathbf{x}_i$ is required. Denormalizing leads to the desired solution $\mathbf{H}$ which can be computed as

$$\begin{aligned} \widetilde{\mathbf{x}}'_i &= \widetilde{\mathbf{H}}\widetilde{\mathbf{x}}_i & (4) \\ \mathbf{T}\mathbf{x}'_i &= \widetilde{\mathbf{H}}\mathbf{U}\mathbf{x}_i & (5) \\ \mathbf{x}'_i &= \mathbf{T}^{-1}\widetilde{\mathbf{H}}\mathbf{U}\mathbf{x}_i & (6) \end{aligned}$$

Read the pages 87-91 in the "Multiple View Geometry" book and try to understand it. Implement a program in MatLab that computes the homography with the normalized DLT out of at least 4 point correspondences using the normalized DLT (the program should be designed such that any arbitrary number of point correspondences ≥ 4 can be used).

<u>Exercise 2</u> RANSAC

In this exercise we will implement a robust method for fitting a model to a data set S, which contains outliers. In case of feature point matching S consists of possible matchings between feature points. As model we consider a homography that maps feature points of one image into feature points of another image. For the estimation of the homography use the DLT algorithm implemented in the previous exercise. RANSAC (RANdom SAmple Consensus) removes outliers by randomly selecting a sample of minimum possible size from S and estimating the size of the consensus set in the remaining set. In order to remove outliers it always keeps the random sample with the highest consensus set in mind. In "Multiple View Geometry in Computer Vision" the RANSAC algorithm is specified as follows:

a) Randomly select a sample of s data points from S and instantiate the model from this subset.

b) Determine the set of data points S_i that are within a distance threshold t of the model. The set S_i is the consensus set of the sample and defines the inliers of S.

c) If the size of S_i (the number of inliers) is greater than some threshold T, re-estimate the model using all the points in S_i and terminate.

d) If the size of S_i is less than T, select a new subset and repeat the above.

e) After N trials the largest consensus set S_i is selected, and the model is re-estimated using all the points in the subset S_i.

Implement this algorithm such that all necessary parameters can be selected by the user. Implement also the adaptive version of RANSAC as mentioned in "Multiple View Geometry in Computer Vision", where the number of samples is determined adaptively:

- $N = \infty$, sample_count = 0.
- While $N >$ sample_count repeat
  - Choose a sample and count the number of inliers.
  - Set $\epsilon = 1 -$ (number_of_inliers)/(total_number_of_points).
  - Set $N = \log(1-p)/\log(1-(1-\epsilon)^s)$.
  - Increment sample_count by 1.
- Terminate.

Exercise 3 Image Stitching

In this exercise we use the techniques of the previous exercises in order to robustly match SIFT features for image stitching. For this, you first have to download a SIFT implementation from the internet. We recommend to use the matlab version from http://www.vlfeat.org/ vedaldi/code/sift.html . Now, apply SIFT to the two images supplied on the web-page and estimate the homography between the images by applying RANSAC from the previous homework (which uses the DLT from exercise 1). Finally, warp the images such such that the common regions overlap. Show the resulting image.