Technische Universität München
Fakultät für Informatik
Dr. Slobodan Ilic

WS 13/14
Exercise Sheet 6
December 9, 2013

Exercises in Tracking & Detection

Exercise 1 Initial Pose Estimation

In this exercise we will implement a camera tracking based on SIFT and non-linear optimization. For this, we first have to initialize the tracking. That is, we have to extract SIFT features within the first image and assign corresponding 3D points to them. We set $\mathbf{R}_0 = \mathbf{I}_{3\times3}$ and $\mathbf{T}_0 = \mathbf{0}$ as initial pose for the camera, where $\mathbf{I}_{3\times3}$ is the 3×3 identity matrix. The intrinsic matrix $\mathbf{A}$ of the camera is

$$\mathbf{A} = \begin{bmatrix} 472.3 & 0.64 & 329.0 \\ 0 & 471.0 & 268.3 \\ 0 & 0 & 1 \end{bmatrix} \tag{1}$$

and is fixed for the whole sequence. The corresponding image sequence is offered on the webpage. The first image of the sequence shows a textured object parallel to the image plane of the camera. The first step is then to extract SIFT features within the area of the object. In order to compute the corresponding 3D points you simply project the obtained 2D points $\mathbf{m}_i$ (in homogeneous coordinates) to the virtual image plane (at $z = 1$) using the inverse of the intrinsic matrix:

$$\mathbf{M}_i = \mathbf{A}^{-1}\mathbf{m}_i. \tag{2}$$

Exercise 2 Tracking SIFT Points

In this exercise we want to find correctly matched SIFT points (inliers) between the image taken at time $t = 0$ and the image taken at time t. Hereby, you should proceed as follows:

a) Compute all SIFT points in your image taken at time t at 2D-locations $\mathbf{m}_{i,t}$.

b) Match these SIFT points with the SIFT points at locations $\mathbf{m}_{i,t=0}$ computed in the image taken at time $t = 0$. Use only the SIFT points for which you have already computed their 3D coordinates $\mathbf{M}_{i,t=0}$ in the previous exercise.

c) Compute all inliers using RANSAC, normalized DLT and Homography computation (as you have done in exercise sheet 4).

In order to show us the results we ask you to successively visualize the pairs of images taken at time $t = 0$ and at time $t = 1...n$ and to visualize the correct point correspondences by drawing lines between them.

For SIFT, RANSAC, DLT and Homography computation you can use your own implementation or any implementation you can find in the web. Again, we recommend to use http://www.vlfeat.org/~vedaldi/code/sift.html for SIFT but as said above you are free to choose any SIFT implementation you want.

For time reasons, please save the image pairs with the visualized correspondences such that we do not have to compute everything from the scratch during correction time.

Exercise 3 Non Linear Optimization and Pose Computation

In this exercise we want to compute the current pose of the object by using the results of the previous exercises. So far, you have an initial pose $[\mathbf{R}_0, \mathbf{T}_0]$, the internal calibration matrix $\mathbf{A}$ and correct point correspondences from the image taken at time $t = 0$ and to all other images. Additionally, you have the 3D coordinates of the 2D feature points in the image taken at time $t = 0$. In order to compute the correct current pose we have to use non-linear optimization tools.

For this reason we need an energy function. One possible energy function is shown in Equ. 3:

$$\mathbf{f}(\mathbf{A}, \mathbf{r}_\alpha, \mathbf{r}_\beta, \mathbf{r}_\gamma, \mathbf{t}_1, \mathbf{t}_2, \mathbf{t}_3, \mathbf{M}_i, \mathbf{m}_i) = \sum_i \|\mathbf{A}\,[\mathbf{R}(\alpha, \beta, \gamma), \mathbf{T}]\,\mathbf{M}_i - \mathbf{m}_i\|^2 \tag{3}$$

a) Implement the energy function $\mathbf{f}$ in Matlab which is dependent on the rotation parameters $\mathbf{r}_\alpha, \mathbf{r}_\beta, \mathbf{r}_\gamma$, on the translation parameters $\mathbf{t}_1, \mathbf{t}_2, \mathbf{t}_3$, on the internal calibration matrix $\mathbf{A}$ and on the 2D-3D correspondences $\mathbf{M}_i, \mathbf{m}_i$. You are free to parametrize your rotation matrix as you want to (e.g. you can use the Euler representation or the Rodriguez formula).

b) Now compute the current pose $[\mathbf{R}_t, \mathbf{T}_t]$ at time t by using the 2D-3D point correspondences $\mathbf{M}_i, \mathbf{m}_{i,t}$ and the 6 pose parameter at time $(t-1)$ $[\mathbf{r}_{\alpha,t-1}, \mathbf{r}_{\beta,t-1}, \mathbf{r}_{\gamma,t-1}, \mathbf{t}_{1,t-1}, \mathbf{t}_{2,t-1}, \mathbf{t}_{3,t-1}]$ as initial value. The current pose parameters are obtained by using the non linear optimization in Matlab. Use the *fminsearch* function in matlab.

c) Show the trajectory of the camera in a 3D graph by plotting the camera coordinates in the world coordinate frame. The camera coordinates in the world coordinate frame at time t are obtained by computing $-\mathbf{R}_t^\top \mathbf{T}_t$.

Exercise 4 Analytical Computation of the Jacobian

In exercise 3, we use *fminsearch* to do energy minimization. Notice that Equ. 3 is actually a non-linear least square. Thus we can approach the minimization in a different way:

a) First, find the Jacobian matrix of Equ. 3 in an analytic form. For this purpose, we use the symbolic toolbox of Matlab. Use *syms* to construct the symbolic expression of Equ. 3, and use *diff* function to differentiate. Return the Jacobian as the 2^{nd} output argument of your Matlab implementation of the energy function.

b) Secondly, instead of using *fminsearch*, use *lsqnonlin* to find the minima. Please take a close look at the instruction of *lsqnonlin* in Matlab Help.