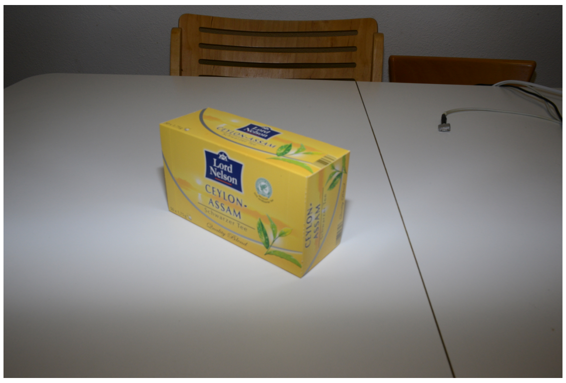
Technische Universität München
Fakultät für Informatik
PD Dr. Slobodan Ilic

WS 17/18
Exercise Sheet 3
November 2, 2017

Exercises in Tracking & Detection

In 3D Computer Vision object pose estimation is one of the key elements for model based tracking. Its applications are evident in many application areas like Augmented Reality, Robotics, Machine Vision and Medical Imaging. In this project you will develop a method that will estimate the pose of a camera in respect to the 3D model of the object exploiting the texture information of the object described by its visual features. You are given a calibrated camera with its known intrinsic parameters, 3D CAD model of the object described as a triangular mesh containing vertices and faces and a number of input RGB camera images. Initially the object doesn't contain associated texture, but you will be given couple of images of the object seen from different view points , which will serve to associate texture information to the object. The test images contain two sets. The first one are images depicting the object from different viewpoint and they are not consecutively taken, so have to be used separately for pose estimation. The second one is a video sequence and will be used for tracking the object. The project will contain several parts explained below and they should be implemented one after the other. The implementation has to be done in MATLAB. Please follow the requirements below in terms of the functions that can be used from MATLAB Toolboxes. If not specified that available functions from MATLAB can be used it means that the functions must be implemented by yourself.


(a) Teabox image for texturing.

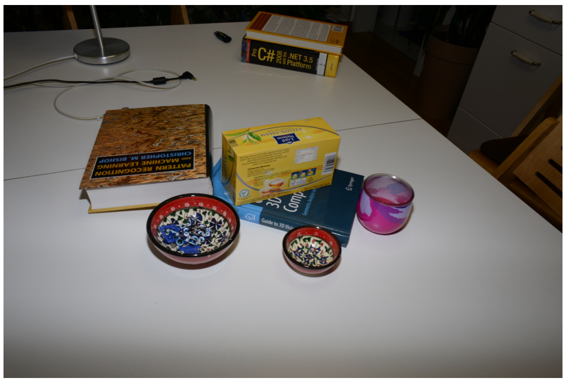

(b) Cluttered scene for detection.

Figure 1: Example images of the teabox for initial textring (a) and detection and pose refinement (b)

Task 1 Model preparation

As a first step, you will need to associate the texture information to the give 3D model from the couple of input images depicting the object from different viewpoints. The 3D model is called *teabox.ply* given as ASCII file in PLY format. The description of the format can be found `http://paulbourke.net/dataformats/ply/`. The model contains 8 vertices accompanied with per vertex normals and 12 triangles (faces). The model can be visualized using Meshlab program `http://www.meshlab.net/`. In MATLAB use function pcread available in Computer Vision System Toolbox of MATLAB 2017b.

In addition to the teabox.ply file describing the object's geometry you are also give the intrinsic camera parameters being: $f_x = f_y = 2960.37845$ $c_x = 1841.68855$ $c_y = 1235.23369$. The world coordinate system is attached to the centroid of the object and it is right handed

Your firs task is to align the object to the input images that can be found in the folder **images\init_texture**. An example image is shown in Fig. 1a. The alignment can be made by clicking on the object vertices in the images providing 2D correspondences to the available 3D vertices of the object. The manual 3D-2D correspondences should be used with the PnP algorithm to compute the initial camera poses allowing alignment of the 3D model to the input texture images. The next step is to associate keypoints detected in the input texture images to the actual 3D model, thus creating additional 2D-3D correspondences that will be used in the following steps of the method. For detection of the keypoints SIFT has to be extracted using VLFeat library `http://www.vlfeat.org/`.

Task 2 Pose estimation with PnP

Having the 3D model with associated 2D-3D correspondence and the camera intrinsics you have to estimate the camera pose in all subsequent test images given in the folder **images\detection** separately for each image. An example image is depicted in Fig. 1b. For that you need to to use PnP and RANSAC. For PnP you can use *estimateWorldCameraPose* function from MATLAB's Computer Vision System Toolbox and RANSAC has to be implemented by yourself. The result of this procedure has to be the best pose hypothesis with the maximal number of inliers. You have to report in which images you manged to get close to correct pose and explain why.

Task 3 Pose refinement with non-linear optimization

After getting the best pose hypothesis from the previous step you need to do pose refinement. This includes minimizing the re-projection error between 3D points on the model corresponding to the detected feature points in the input texture images and the feature points detected in the image. This will result in writing an objective function that should be minimized in terms of the camera pose (rotation and translation). Rotation should be parameterized using Lie Algebra. In order to optimize the objective function, Jacobian has to be computed both analytically and with finite differences and Levemberg-Marquardt method for optimization has to be implemented. The expected output is the precise pose indicated by correct alignment of the 3D model.

Task 4 Tracking

The final task is to preform tracking of the camera in respect to the given 3D model. The test sequence is given in **images\tracking** folder.