Exercise session using MATLAB:
— Structure and Motion in Flatland

Overview

In this laboratory session you are going to use matlab to study structure and motion estimation using the L_∞-norm with 1D cameras. In this so-called flatland problem, one assumes that the world consists of a 2D plane and that the cameras capture 1D image lines of the 2D scene structure. Important references are "An L_∞ Approach to Structure and Motion Problems in 1D-Vision" by Åström et al. as well as the lectures notes on "Quasiconvexity: Optimization over SO(3)".

First, unless you have already done so, download and unpack `OCV2008.zip` from the course homepage.

`http://www2.imm.dtu.dk/projects/OCVschool/Materials.html`

To start the correct version of MatLab on DTU's unix terminals you should type `matlab72` in the terminal window. Remember to run `startup.m` to get the correct paths. The files you are going to use can be found in the subdirectory `Flatland`. Some of the instructions here are matlab commands that you are supposed to try.

Structure and Motion in Flatland

In this session we will study three specific instances of Structure and Motion with 1D cameras. Each example setup consists of 10 points in the 2D plane which are viewed by 3 calibrated 1D cameras.

Recall that the (perspective) camera equation can be written

$$\lambda \mathbf{u} = \mathbf{P}\mathbf{U}$$

where $\mathbf{u} = [cos(\alpha), sin(\alpha)]^T$ denotes a point on the image line (or rather circle) and the elements of $\mathbf{U}$ are homogeneous coordinates for a point in the world plane. Also, $\mathbf{P}$ denotes a 2×3 camera matrix. Since the camera is calibrated one can assume that the matrix has the form:

$$\mathbf{P} = \begin{bmatrix} a & b & t_1 \\ -b & a & t_2 \end{bmatrix}$$

where $a = cos(\theta)$, $b = sin(\theta)$ and θ is the rotational angle of the camera. Without loss of generality one can assume that the first camera is $\mathbf{P}_1 = [\mathbf{I} \,|\, \mathbf{0}]$ by choice of coordinate system. One may think of the camera device as measuring angles to points in the scene, and hence it makes sense to minimize residuals between measured and reprojected angles. A convenient way of expressing this residual between a measured angle $\tilde{\alpha}$ and a reprojected angle α - which depends on $\mathbf{P}$ and $\mathbf{U}$ - is through

$$\frac{\mathbf{u} \times \mathbf{PU}}{\mathbf{u} \cdot \mathbf{PU}} = \frac{|\mathbf{u}||(\mathbf{PU})|\sin(\alpha - \tilde{\alpha})}{|\mathbf{u}||(\mathbf{PU})|\cos(\alpha - \tilde{\alpha})} = \tan(\alpha - \tilde{\alpha}). \tag{1}$$

Here $\mathbf{a} \times \mathbf{b}$ denotes the scalar $a_1 b_2 - a_2 b_1$. Since $\mathbf{u} \cdot \mathbf{PU} > 0$, checking whether $|\alpha - \tilde{\alpha}| \leq \Delta$ is equivalent to

$$|\mathbf{u} \times \mathbf{PU}| \leq \tan(\Delta)(\mathbf{u} \cdot \mathbf{PU}). \tag{2}$$

Example 1. This first example is just for illustration purposes in order to get acquinted with the problem. Not only the 3 images $\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3$ are provided, but also the ground truth camera matrices $\mathbf{P}_1$, $\mathbf{P}_2$ and $\mathbf{P}_3$ as well as the true 2D structure $\mathbf{U}$ are given. Run the script `OCV_demo.m` to learn more.

Example 2. In this second example, the rotational part of the camera matrices, that is, the first 2×2 blocks of the camera matrices, $\mathbf{R}_1$, $\mathbf{R}_2$ and $\mathbf{R}_3$ are given, and your task is to solve for 2D structure $\mathbf{U}$ and camera translations (the third row of the camera matrix). The images are encoded as previously in $\mathbf{u}_1$, $\mathbf{u}_2$ and $\mathbf{u}_3$. Type `load OCV_example2` to load the data.

Recall that for a given angle Δ (in radians), the problem of testing whether there exists a solution with error equal or less than Δ (a so-called feasibility problem) can be solved with Linear Programming, cf. (2). This feasibility test has already been implemented in the function `lp_knownrotation.m`.

Task: Solve for the optimal Δ^* and plot your solution (both 2D points and camera centres) in a figure.

Example 3. In the last example, `OCV_example3`, only the images $\mathbf{u}_1$, $\mathbf{u}_2$ and $\mathbf{u}_3$ are given, and the task is to solve for both 2D structure and camera motion. This can be done by branch and bound over the camera rotation angles θ_2 and θ_3 for the domain $[0, 2\pi] \times [0, 2\pi]$. (Note that $\theta_1 = 0$ according to the chosen coordinate system).

Suppose that there exists a solution with maximum error d_{max} radians. Now consider the square $[\theta_2 - \Delta, \theta_2 + \Delta] \times [\theta_3 - \Delta, \theta_3 + \Delta]$ in the domain of rotation angles. Then, we know (Lemma 2) that there are no feasible solutions in this square provided that the following feasibility problem is *infeasible*:

$$|\mathbf{u}_1 \times \mathbf{P}_1\mathbf{U}| \leq \tan(d_{max})(\mathbf{u}_1 \cdot \mathbf{P}_1\mathbf{U}), \tag{3}$$
$$|\mathbf{u}_2 \times \mathbf{P}_2\mathbf{U}| \leq \tan(d_{max} + \Delta)(\mathbf{u}_2 \cdot \mathbf{P}_2\mathbf{U}), \tag{4}$$
$$|\mathbf{u}_3 \times \mathbf{P}_3\mathbf{U}| \leq \tan(d_{max} + \Delta)(\mathbf{u}_3 \cdot \mathbf{P}_3\mathbf{U}), \tag{5}$$

where $\mathbf{P}_2 = \mathbf{P}_2(\theta_2)$ and $\mathbf{P}_3 = \mathbf{P}_3(\theta_3)$. (Note that $d_{max} + \Delta$ should be less than $\pi/2$ for the statement to be valid). This can be used to discard squares in the domain of possible poses. If the above problem turns out to be feasible, then there may exist a better solution in the square, and one has to subdivide the square into smaller squares.

Task: Use branch and bound to find a solution with a min-max error better than $d_{max} \leq 0.03$ radians. Also, locate the possible regions of the (θ_2, θ_3)-plane where such solutions may exist. What is the best solution you are able to compute?