Computer Vision, Assignment 5
Local Optimization and Structure from Motion

1 Instructions

In this assignment you will study model fitting using optimization. In particular you will compute the maximal likelihood solution for a couple of structure and motion problems. The data for the assignments is available from the course page `https://canvas.education.lu.se/courses/3379`

The assignment is due at the end of study week 7. The deadline is firm and reports are to be handed in on time through the canvas page. Not everything has to be correct the first time you hand in however you have to present solutions/attempts for each mandatory exercise. It is not ok to hand in blank solutions. (If you have problems solving something you should come to the Q & A-sessions or contact the lecturer by email in good time before the deadline.) In exceptional cases extensions can be offered (due to unforeseen circumstances). In such cases contact the lecturer by email before the deadline (or as soon as possible) for instructions on what to do.

Make sure you answer all questions and provide complete solutions to the exercises. Your solutions should be be submitted as a single pdf-file. Note that it is fine to submit handwritten solutions (by including scans in the pdf-file) as long as they are well structured and readable. After each exercise there is a gray box with instructions on what should be included in the report. In addition, all the code should be submitted as m- and mat-files in a zip-archive. Make sure that your matlab scripts are well commented and can be executed directly, that is, without loading any data, setting parameters etc. Such things should be done in the script.

If you run into problems with any of the exercises you can go to the Q & A-sessions to get help (see course schedule) or send an email to the lecturer (`carl.olsson@math.lth.se`).

The report should be written individually, however you are encouraged to work together. Keep in mind that everyone is responsible for their own report and should be able to explain all the solutions.

Some exercises are marked as OPTIONAL. You do not have to do these to pass the assignment. However if you submit good solutions to these you will be awarded at most 0.2 bonus points for the home-exam.

2 Maximum Likelihood Estimation for Structure from Motion Problems

Exercise 1. Suppose the 2D-point $x_{ij} = (x_{ij}^1, x_{ij}^2)$ is an observation of the 3D-point $\mathbf{X}_j$ in camera P_i. Also we assume that the observations are corrupted by Gaussian noise, that is,

$$(x_{ij}^1, x_{ij}^2) = \left(\frac{P_i^1 \mathbf{X}_j}{P_i^3 \mathbf{X}_j}, \frac{P_i^2 \mathbf{X}_j}{P_i^3 \mathbf{X}_j} \right) + \epsilon_{ij}, \tag{1}$$

where P_i^1, P_i^2, P_i^3 are the rows of the camera matrix P_i and ϵ_{ij} is normally distributed with covariance σI. The probability density function is then

$$p(\epsilon_{ij}) = \frac{1}{2\pi\sigma} e^{-\frac{1}{2\sigma^2}||\epsilon_{ij}||^2}. \tag{2}$$

Assuming that the ϵ_{ij} are independent, that is

$$p(\epsilon) = \prod_{i,j} p(\epsilon_{ij}), \tag{3}$$

show that the model configuration (points and cameras) that maximizes the likelihood of the obtaining the observations $x_{ij} = (x_{ij}^1, x_{ij}^2)$ is obtained by solving

$$\min \sum_{i=1}^{n} \sum_{j=1}^{m} || \left(x_{ij}^1 - \frac{P_i^1 \mathbf{X}_j}{P_i^3 \mathbf{X}_j},\ x_{ij}^2 - \frac{P_i^2 \mathbf{X}_j}{P_i^3 \mathbf{X}_j} \right) ||^2. \tag{4}$$

Hint: Maximize the log-likelihood $\log p$.

For the report: Complete solution.

3 Calibrated Structure from Motion and Local Optimization

Exercise 2. (**OPTIONAL.**) Unfortunately there is no formula for computing the ML estimation when we use general pinhole cameras. The only way to find the ML estimate is to try to improve a starting solution using local optimization. Suppose that we want to minimize

$$\sum_i r_i(v)^2 = ||r(v)||^2, \tag{5}$$

where $r_i(v)$ are the error residuals and $r(v)$ is a vector containing all the $r_i(v)$. The first order Taylor expansion at a point v_0 is

$$r(v) \approx r(v_0) + J(v_0)\delta v, \tag{6}$$

where $\delta v = (v - v_0)$ and $J(v_0)$ is a matrix whose rows are the gradients of $r_i(v)$ at v_0. Show that the steepest descent direction of the approximation

$$||r(v_0) + J(v_0)\delta v||^2 \tag{7}$$

at the point x_0 is

$$d = -2J(v_0)^T r(v_0). \tag{8}$$

For the report: Complete solution if you want the bonus.

Exercise 3. (**OPTIONAL.**) A direction d is called a descent direction (of f at v_0) if

$$\nabla f(v_0)^T d < 0, \tag{9}$$

since this means that the directional derivative in the direction d is negative (see your multidimensional calculus book).

Show that the steepest descent direction in (8) is a descent direction of the function in (5).

A matrix M is called positive definite if $v^T M v > 0$ for any v such that $||v|| \neq 0$. Show that the direction

$$d = -M\nabla f(x) \tag{10}$$

is a descent direction. In the Levenberg-Marquardt method we chose the update step

$$\delta v = -(J(v_0)^T J(v_0) + \lambda I)^{-1} J(v_0)^T r(v_0) \tag{11}$$

Is this a step in a descent direction?

For the report: Complete solution if you want the bonus.

Computer Exercise 1. In Computer Exercise 3 and 4 of Assignment 3, you computed a solution to the two-view structure form motion problem for the two images of Figure 1 using the 8-point algorithm. In this exercise the goal is to use the solution from Assignment 3 as a starting solution and locally improve it using the Levenberg-Marquardt method.

Figure 1: kronan1.jpg and kronan2.jpg.

The file `LinearizeReprojErr.m` contains a function that for a given set of cameras, 3D points and imagepoints, computes the linearization (6). The file `update_solution.m` contains a function that computes a new set of cameras and 3D points from an update δv computed by any method. The file `ComputeReprojectionError.m` computes the reprojection error for a given set of cameras, 3D points and image points. It also returns the values of all the individual residuals as a second output.

In the Levenberg-Maquardt method the update is given by

$$\delta v = -(J(v_k)^T J(v_k) + \lambda I)^{-1} J(v_k)^T r(v_k). \tag{12}$$

Using this scheme and starting from the solution that you got in Assignment 3, plot the reprojection error versus the iteration number for $\lambda = 1$. Also plot histograms of all the residual values before and after running the Levenberg-Maquardt method.

Try varying λ. What happens if λ is very large/small?

```
Useful matlab commands:
%Takes two camera matrices and puts them in a cell.
P = {P1,P2}

%Computes the reprejection error and the values of all the residuals
%for the current solution P,U,u.
[err,res] = ComputeReprojectionError(P,U,u);

%Computes the r and J matrices for the appoximate linear least squares problem.
[r,J] = LinearizeReprojErr(P,U,u)

% Computes the LM update.
C = J'*J+lambda*speye(size(J,2));
c = J'*r;
deltav = -C\c;
```

```
%Updates the variabels
[Pnew,Unew] = update_solution(deltav,P,U);
```

For the report: Submit plots of the function value versus the number of iterations and the two histograms.

Computer Exercise 2. (**OPTIONAL**) In this exercise the goal is to compute a dense textured model of kronan, see Figure 2, using the two images from the previous exercise. The file `compex2data.mat` contains foreground

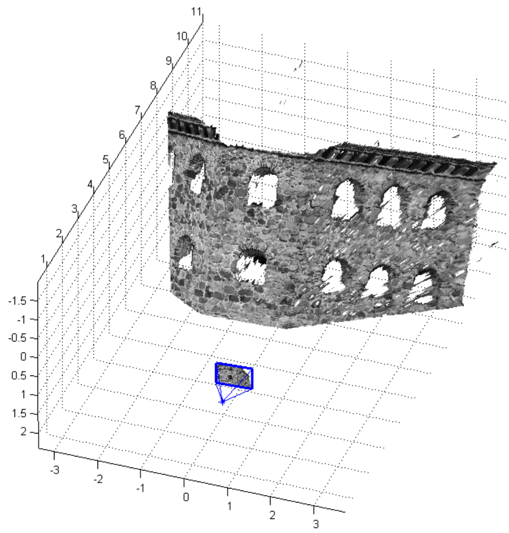

Figure 2: Dense reconstruction of kronan.

segmentations of the two images. The file `compute_ncc.m` contains a function that computes the normalized cross correlation for each pixel and depth using the two images and the segmentations. The file `disp_result.m` plots the resulting model. Use the commands listed below to compute the textured model.

```
Useful matlab commands:
%Selects suitable depths for the planesweep algorithm
d = linspace(5,11,200);

%rescales the images to incease speed.
sc = 0.25;
%Compute normalized corss correlations for all the depths
[ncc,outside_image] = compute_ncc(d,im2,P{2},im1,segm_kronan1,P{1},3,sc);

%Select the best depth for each pixel
[maxval,maxpos] = max(ncc,[],3);

%Print the result
disp_result(im2,P{2},segm_kronan2,d(maxpos),0.25,sc)
```

For the report: Show the 3D reconstruction.