

# Computer Vision, Assignment 4

## Model Fitting and Local Optimization

### 1 Instructions

In this assignment, you will first study model fitting, in particular, estimating essential matrices. You will use random sampling consensus RANSAC to robustly estimate essential matrices. Secondly, you will study local optimization. You will use Levenberg-Marquardt to refine the 3D points obtained from an initial solution to the SfM problem. After completing this assignment, you will have a simple functional SfM pipeline.

Please see Canvas for detailed instructions on what is expected for a passing/higher grade. All exercises not marked **OPTIONAL** are “mandatory” in the sense described on Canvas.

**The maximum amount of points for the theoretical exercises in Assignment 4 is 16. 50% of 16 is 8.**

## 2 Robust Epipolar Geometry and Two-View Reconstruction

In this first section of Assignment 4, you will work on robust estimation using RANSAC.

### *Theoretical Exercise 1 (3 points).*

Suppose that two calibrated cameras are represented by the camera matrices  $P_1 = [R_1 \ t_1]$  and  $P_2 = [R_2 \ t_2]$ . What is the essential matrix for this pair of cameras?

For the report: Complete solution.

### *Theoretical Exercise 2 (5 points).*

Suppose that we want to find an essential matrix that relates two sets of 2D points. How many degrees of freedom does an essential matrix have?

What is the minimal number of point correspondences (matches) that you need to determine E?

How many point correspondences are required by the eight point solver (eight point algorithm) to determine E?

If the proportion of incorrect correspondences is 15%, how many iterations of RANSAC with an *eight point solver* do you need to find an outlier free sample set with 99% probability?

For the report: Answers are enough.

*Computer Exercise 1.* (See Jupyter Notebook)

*Computer Exercise 2.* (See Jupyter Notebook)

## 3 Levenberg-Marquardt for Structure from Motion Problems

In this second section of Assignment 4, you will study Levenberg-Marquardt for optimizing non-linear least squares objectives, in particular minimizing the reprojection error of an SfM solution.

Suppose that the 2D point  $x_{ij} = (x_{ij,1}, x_{ij,2})$  is an observation of the 3D point  $\mathbf{X}_j$  in camera  $P_i$ . The model configuration (points and cameras) that maximizes the likelihood of obtaining the observations  $x_{ij} = (x_{ij,1}, x_{ij,2})$  under the assumption of i.i.d. Gaussian noise on the observations is then found by minimizing the reprojection error, i.e. solving

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

Here,  $P_i^\ell$  denotes the  $\ell$ 'th row of  $P_i$ , and we write  $\mathbf{X}_j$  in homogeneous coordinates. The solution to (1) is called the Maximum Likelihood (ML) estimate. Unfortunately, there is no closed-form solution for computing the ML estimate 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.

### *Theoretical Exercise 3 (8 points).*

In this exercise you will derive the Jacobian and residual expressions needed for the refinement of the 3D points using the Levenberg-Marquardt method. You will then in the next exercise implement these formulas and refine the 3D points you obtained in Computer Exercise 2.

We write the reprojection error for a particular 3D point  $\mathbf{X}_j$  as

$$\sum_{i=1}^m \|\mathbf{r}_i(\mathbf{X}_j)\|^2 \quad (2)$$

where

$$\mathbf{r}_i(\mathbf{X}_j) = (r_{i,1}(\mathbf{X}_j), r_{i,2}(\mathbf{X}_j))^T = \left( x_{ij,1} - \frac{P_i^1 \mathbf{X}_j}{P_i^3 \mathbf{X}_j}, x_{ij,2} - \frac{P_i^2 \mathbf{X}_j}{P_i^3 \mathbf{X}_j} \right)^T \quad (3)$$

is the residual vector of  $\mathbf{X}_j$  in image  $i$  with individual components  $r_{i,1}, r_{i,2}$ . In total we get two residuals per image and 3D point, so  $2 \cdot m$  residuals for every 3D point.

When using Levenberg-Marquardt (and related methods) for minimizing (2) we approximate the residual vector at an updated 3D point  $\mathbf{X}_j + \delta \mathbf{X}_j$  using a first order Taylor approximation at  $\mathbf{X}_j$ , i.e.,

$$\mathbf{r}_i(\mathbf{X}_j + \delta \mathbf{X}_j) \approx \mathbf{r}_i(\mathbf{X}_j) + J_i(\mathbf{X}_j) \delta \mathbf{X}_j, \quad (4)$$

where  $J_i(\mathbf{X}_j)$  is the Jacobian of  $\mathbf{r}_i(\mathbf{X}_j)$ .

**Part I.** Show that the Jacobian matrix of  $\mathbf{r}_i(\mathbf{X}_j)$  relative to  $\mathbf{X}_j$  for camera  $i$  is a  $2 \times 4$  matrix given by

$$J_i(\mathbf{X}_j) = \begin{bmatrix} \frac{(P_i^1 \mathbf{X}_j)}{(P_i^3 \mathbf{X}_j)^2} P_i^3 - \frac{1}{P_i^3 \mathbf{X}_j} P_i^1 \\ \frac{(P_i^2 \mathbf{X}_j)}{(P_i^3 \mathbf{X}_j)^2} P_i^3 - \frac{1}{P_i^3 \mathbf{X}_j} P_i^2 \end{bmatrix}. \quad (5)$$

*HINT:* To simplify the derivations you can use an intermediate representation  $Z_{ij} = P_i \mathbf{X}_j$ . This will allow you to decompose the calculation into two (simpler) parts by using the chain rule for derivatives. For instance, for the first row of  $J_i$ , i.e. the derivative of  $r_{i,1}$  w.r.t.  $\mathbf{X}_j$ , using  $Z_{ij} = P_i \mathbf{X}_j$  and the chain rule we get

$$\frac{\partial r_{i,1}}{\partial \mathbf{X}_j} = \frac{\partial r_{i,1}}{\partial Z_{ij}} \frac{\partial Z_{ij}}{\partial \mathbf{X}_j}, \quad (6)$$

where  $\frac{\partial r_{i,1}}{\partial Z_{ij}}$  ( $1 \times 3$  vector) and  $\frac{\partial Z_{ij}}{\partial \mathbf{X}_j}$  ( $3 \times 4$  matrix) are much easier to find than  $\frac{\partial r_{i,1}}{\partial \mathbf{X}_j}$ . A first step would be to write (3) replacing  $Z_{ij} = P_i \mathbf{X}_j$  and find  $\frac{\partial r_{i,1}}{\partial Z_{ij}}$  and  $\frac{\partial r_{i,2}}{\partial Z_{ij}}$ . For computing  $\frac{\partial Z_{ij}}{\partial \mathbf{X}_j}$  you might find the relation  $\frac{\partial(Ax)}{\partial x} = A$  useful, for a matrix  $A$  and a vector  $x$ . Other (possibly) useful matrix relations can be found in the Matrix Cookbook.

**Part II.** Show how to build  $\mathbf{r}(\mathbf{X}_j)$  and  $J(\mathbf{X}_j)$  from  $\mathbf{r}_i(\mathbf{X}_j)$  and  $J_i(\mathbf{X}_j)$  such that

$$\sum_{i=1}^m \|\mathbf{r}_i(\mathbf{X}_j) + J_i(\mathbf{X}_j) \delta \mathbf{X}_j\|^2 = \|\mathbf{r}(\mathbf{X}_j) + J(\mathbf{X}_j) \delta \mathbf{X}_j\|^2. \quad (7)$$

In particular, what are the dimensions of  $\mathbf{r}(\mathbf{X}_j)$  and  $J(\mathbf{X}_j)$ ?

For the report: Complete derivations.

*Computer Exercise 3.* (See Jupyter Notebook)

*Computer Exercise 4.* (**OPTIONAL, 10 optional points.** See Jupyter Notebook)

*Theoretical Exercise 4.* (**OPTIONAL, 15 optional points.**)

**Background.** We begin by some short preliminaries on Levenberg-Marquardt (LM) that can also be found in Computer Exercise 3. LM is an iterative method for minimizing a non-linear least squares objective

$$F(v) = \|\mathbf{r}(v)\|^2, \quad (8)$$

with respect to  $v$ . In LM, the update is given by

$$\delta v = -(J(v)^T J(v) + \mu I)^{-1} J(v)^T \mathbf{r}(v), \quad (9)$$

where  $J(v)$  is the Jacobian of  $\mathbf{r}(v)$  at  $v$ . Here  $\mu > 0$  is a damping factor that is adjusted adaptively (see Lecture Slides 9) and  $v$  is the previous solution. The new solution is  $v + \delta v$  and we iterate until convergence (or for a fixed number of iterations). In our setting, we have  $v = \mathbf{X}_j$  and  $\mathbf{r}(v) = \mathbf{r}(\mathbf{X}_j)$  with the  $\mathbf{r}$  from (7).

**Exercise.** A direction  $d$  is called a descent direction of a function  $F : \mathbb{R}^N \rightarrow \mathbb{R}$  at the point  $v$  if

$$\nabla F(v)^T d < 0, \quad (10)$$

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

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

$$d = -M \nabla F(v) \quad (11)$$

is a descent direction for  $F$  at  $v$  (provided  $v$  is not a stationary point).

Is the update chosen in Levenberg-Marquardt (9) in a descent direction for the  $F(v)$  in (8)?

For the report: Complete solution.