



## Multiple View Geometry: Exercise Sheet 9

Prof. Dr. Daniel Cremers, David Schubert, Mohammed Brahimi  
Computer Vision Group, TU Munich

<http://vision.in.tum.de/teaching/ss2019/mvg2019>

Exercise: July 10th, 2019

---

### Part I: Theory

The following exercises should be **solved at home**. Writing down your answer will help you present and discuss the answer during the tutorial.

This exercise partly builds upon the theory part of last week's exercise. Check the solutions of Sheet 8 in case there is something you do not understand.

#### 1. Robust Least Squares

In order to make the solution of the Direct Image Alignment from Sheet 8 more robust to outliers, one can replace the square in the energy

$$E(\xi) = \sum_i r_i(\xi)^2$$

by a robust loss function  $\rho$ :

$$E_\rho(\xi) = \sum_i \rho(r_i(\xi)).$$

(a) What situations can you think of where a robust loss function might be needed?

The minimizer of  $E_\rho$  also minimizes the weighted least squares problem

$$E_w(\xi) = \sum_i w(r_i) r_i(\xi)^2$$

with weights defined by  $w(t) := \rho'(t)/t$ .

(b) One example for a robust loss function is the Huber loss function  $\rho_\delta$ :

$$\rho_\delta(t) = \begin{cases} \frac{t^2}{2} & |t| \leq \delta \\ \delta|t| - \frac{\delta^2}{2} & \text{else} \end{cases}$$

Write down the weight function for the Huber loss.

## 2. Optimization Techniques

Define  $\mathbf{r}$  as the vector containing the residuals and  $J$  as the matrix containing gradients of all residuals at  $\xi = \mathbf{0}$ :

$$\mathbf{r}_i = r_i(\mathbf{0}), \quad J^{(i)} = \left. \frac{\partial r_i(\xi)}{\partial \xi} \right|_{\xi=\mathbf{0}}.$$

Furthermore, let  $W$  be the diagonal matrix with weights  $w(r_i(\mathbf{0}))$  on the diagonal. Write down the update step  $\Delta\xi$  for each of the following minimization methods:

- (a) Gradient descent, normal least squares,
- (b) Gradient descent, weighted least squares,
- (c) Gauss-Newton, normal least squares,
- (d) Gauss-Newton, weighted least squares,
- (e) Levenberg-Marquardt, normal least squares, and
- (f) Levenberg-Marquardt, weighted least squares.

## Part II: Practical Exercises

This exercise is to be solved **during the tutorial**.

In this exercise you will continue with the implementation of direct image alignment on SE(3) from Sheet 8. As a basis, you can either use your own implementation or download the package `ex9.zip` provided on the website that contains the solutions of Sheet 8 together with a Script `ex09` that needs to be completed.

1. Implement Huber weighting in order to make your solution robust to outliers (ll. 80). Use GIMP to add some "outliers" to the images, and test your implementation. You will have to implement iteratively re-weighted least-squares, i.e. recompute the matrix  $W$  in each iteration. Use the Huber norm with  $\delta = 4$  as robust loss function and perform **Gauss-Newton** minimization on the corresponding weighted least squares cost function.
2. Adapt your solution to use **Gradient Descent** instead of Gauss-Newton to minimize the weighted least squares error function (ll. 99). What is a good stepsize?
3. Adapt your solution to use the **Levenberg-Marquardt** algorithm for minimization (ll. 105). Test your implementation using the corrupted test images `rgb/*_broken.png` provided in `ex9.zip`. Compare results and convergence behavior with Gauss-Newton and gradient descent.