TDCV Project 1:Keypoint based object detection, pose estimation and refinement

Task 3: Pose Refinement with non-linear optimization

Mai Bui, Ivan Shugurov and Roman Kaskman

Where are we?

- Extraction of SIFT features to establish 2D-3D correspondences between image points and the 3D model
- Detection of the object of interest
- Pose Estimation using PnP and RANSAC

Visualization of predicted poses as projection of 3D bounding box

Task 3: Tracking

- Given an initial detection and pose of the object,
- we want to track the object in the following frames

→ Non-Linear Optimization

Non-Linear Optimization

- Non-Linear least squares problem

$$E(\theta) = E(R,T) = \sum_i \|e_i\|^2$$

- where *e* is a residual function, in our case the re-projection error of correspondences *M* and *m,* and A describes the intrinsics of the camera

$$e_i(R,T) = A(RM_i + T) - m_i$$

- Objective: $\arg\min_{R,T} E(R,T)$

Gauss-Newton

- Function of interest evaluated with changed parameter values can be approximated by first order Taylor expansion

$$\mathrm{E}(\theta + \Delta) = E(\theta) + J\Delta$$

- minimization by using the normal equation leads to the solution

$$\Delta_k = -(J^T J)^{-1} J^T E_k$$

- and update of θ by

$$\theta_{k+1} = \theta_k + \Delta_k$$

Levenberg-Marquardt

- Additional parameter λ:

$$\Delta_k = -(J^T J \ + \lambda I)^{-1} J^T E_k$$

If E($\theta_k + \Delta_k$) > E(θ_k): λ = 10 λ

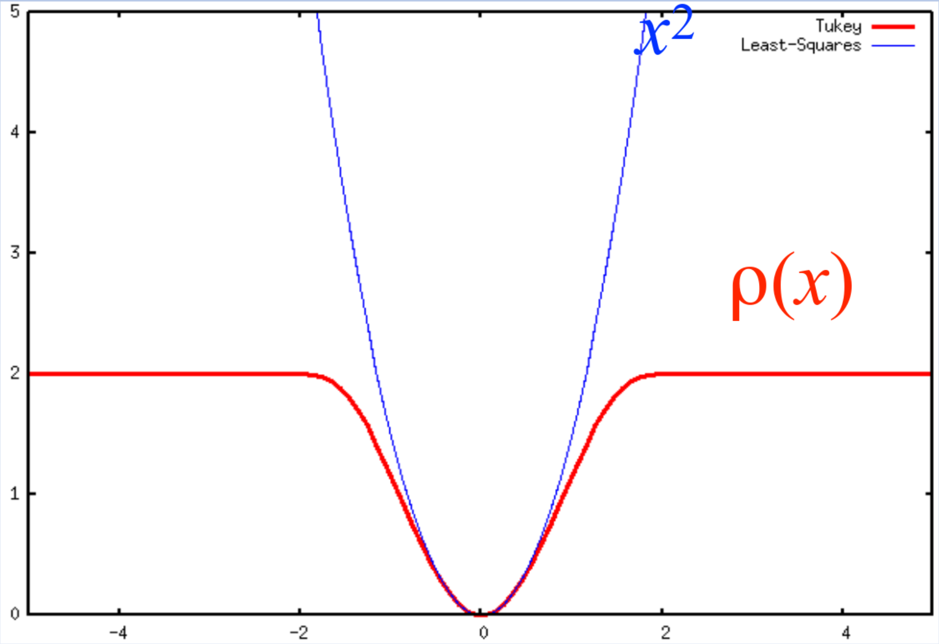
If E($\theta_k + \Delta_k$) < E(θ_k): λ = 0.1 λ and update θ

What about outliers?

Robust Estimator: Tukey's Loss Function

$$\rho(e) = \begin{cases} \frac{c^2}{6}(1-(1-\frac{e}{c})^2)^3, if\ e < c \\ \frac{c^2}{6}, otherwise \end{cases}$$

c is a statistically chosen tuning constant



What about outliers?

Weighted Iterative Least Squares

$$E(R,T) = \sum_i w_i d^2$$

$$w_i = \begin{cases} (1 - \frac{e_i^2}{c^2})^2, if\ e < c \\ 0, otherwise \end{cases}$$

More detailed information can be found here:
https://onlinecourses.science.psu.edu/stat501/print/book/export/html/351/

What about outliers?

Weighted Gauss-Newton:

$$\Delta_k = -(J^T W J)^{-1} J^T W E_k$$

Weighted Levenberg-Marquadt:

$$\Delta_k = -(J^T W J \ + \lambda I)^{-1} J^T W E_k$$

Task 3: Summary

- Compute an initial camera pose for the first frame using PnP and RANSAC from the previous exercise
- Refine the pose for each of the following frames using the robust form of the Levenberg-Marquardt algorithm
- Visualize the resulting camera trajectory for the given image sequence