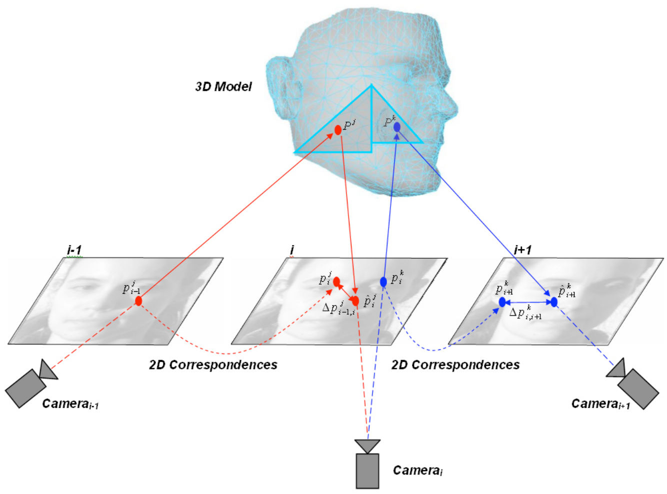
Technische Universität München
Fakultät für Informatik
PD Dr. Slobodan Ilic

WS 19/20
Exercise Sheet 3
November 21, 2019

Exercises in Tracking & Detection

Task 1 Minimization of the Reprojection Error



a) Just looking at the first two frames $Camera_{i-1}, Camera_i$ and point P^j, which of the solid red arrows corresponds to:

- Direct projection
- Back-projection
- Objective function

b) How to get from the reprojection error $\min_{R,T} \sum_i \|A(RM_i + T) - m_i\|^2$ to the objective function $\min_{R,T} \sum_k \|\Delta p_{i-1,i}\|^2$?

c) What is the difference between Algebraic vs Geometric error minimization and when to use which one? What are the advantages and drawbacks of each method?

Task 2 Non-linear optimization and robust estimation

a) Why do we need to use Non-linear optimization?

b) What is the general form of Non-linear least squares problems (NLS)?

c) How does the Gradient descent method differ from Gauss-Newton?

d) Levenberg-Marquardt (LM): Explain the main idea of the method (Algorithm 1), especially lines 4, 5 and how the update step $\Delta x = -(J^T J + \lambda I)^{-1} J^T \mathbf{r}$ in line 6 is derived.

e) Levenberg-Marquardt (LM): Explain the details of the method (Algorithm 1), in lines 7-11. When is the update accepted and λ increased? Similar to which other methods does LM behave if λ is large/small?

Algorithm 1: Levenberg-Marquardt (LM)

Input: x_0 initial parameters, f function to mimimize
Output: x optimized parameters

1 $\lambda \leftarrow \lambda_0$
2 $x \leftarrow x_0$
3 **while** *not converged* **do**
4 $\quad \mathbf{r} \leftarrow f(x)$ `// residual vector`
5 $\quad J \leftarrow \nabla \mathbf{r}(x)$ `// Jacobian`
6 $\quad \Delta x \leftarrow -(J^T J + \lambda I)^{-1} J^T \mathbf{r}$ `// Levenberg step`
7 $\quad$ **if** $E(x + \Delta x) < E(x)$ **then** `// error reduced`
8 $\qquad x \leftarrow x + \Delta x$
9 $\qquad$ increase λ
10 $\quad$ **else**
11 $\qquad$ decrease λ

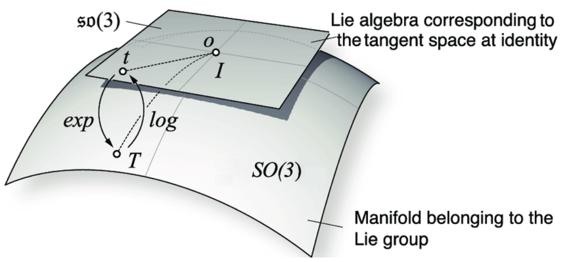
f) Iteratively re-weighted least-squares (IRLS): As an alternative to RANSAC, robust loss functions can be used in NLS. What is the general goal of them? What choices do we have and how to estimate their parameter and weighting factor?

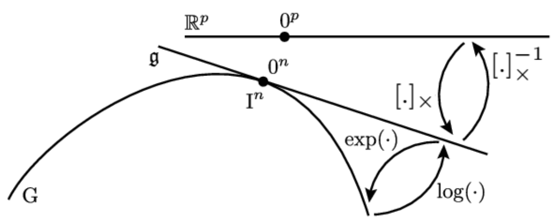
Task 3 Rotation parameterization, Lie Algebra and derivatives

a) What properties does SO(3) have? Why don't we directly optimize in this space?

b) Briefly discuss Euler Angles vs. Unit Quaternion vs Axis-Angle as alternative parameterizations of this space. What is the problem with gimbal lock, singularities and double-covering? Which representation is best suited to represent the parameter update Δx in the context of NLS methods like LM/IRLS from the last task?

c) Explain the pictures by reading section 2.3.1 'Canonical exponential coordinates' of *Y. Ma, S. Soatto, J. Kosecka, and S.S. Sastry (2001): An Invitation to 3-D Vision: From Images to Geometric Models.* `https://www.eecis.udel.edu/~cer/arv/readings/old_mkss.pdf` What are $\mathfrak{so}(3)$, skew-symmetric matrices and the Rodrigues formula $R = \mathrm{I} + [\bar{\mathbf{v}}]_\times \sin\theta + [\bar{\mathbf{v}}]_\times^2 (1 - \cos\theta)$ used for?





d) How to approximate / linearize infinitesimal Rotations?

e) In the LM Algorithm, line 8, how do you apply the update $x \leftarrow x + \Delta x$ if you were to choose to represent $x \in \mathrm{SO}(3)$, but $\Delta x \in \mathfrak{so}(3)$? What is an easier alternative?

f) What are the two options given for the derivative of a rotation $\mathrm{R}(\mathbf{v}) = \exp([\mathbf{v}]_\times)$ with respect to its exponential coordinates $\mathbf{v}$ in *Guillermo Gallego and Anthony Yezzi (2014): A compact formula for the derivative of a 3-D rotation in exponential coordinates.* `https://arxiv.org/pdf/1312.0788.pdf` and which formula do you prefer in a numerical optimization method?