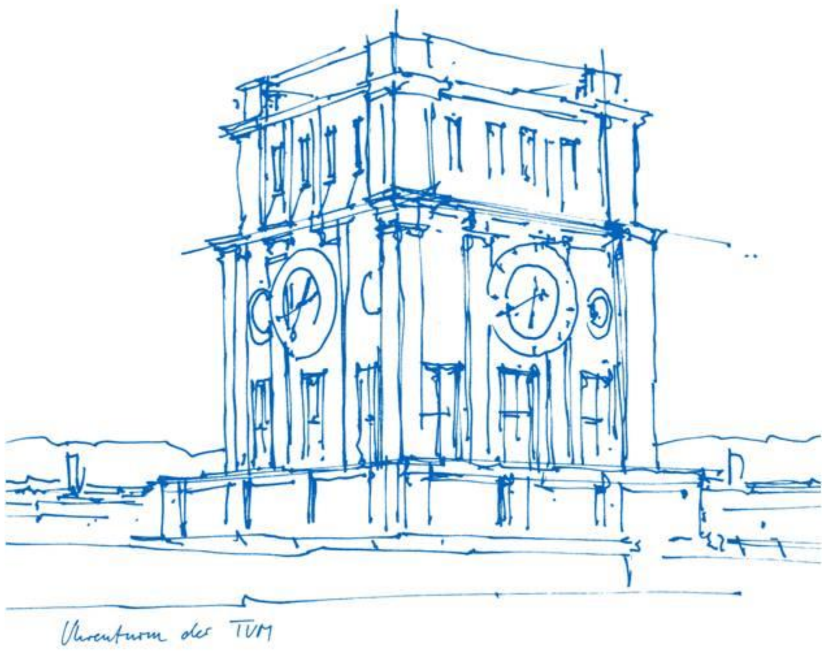
Robust Odometry Estimation for RGB-D Cameras

Kerl, Sturm, Cremers 2013

Florian Becker

Technical University of Munich



Goals and Purpose

[1]

Odometry: Tracking the pose of a robot (position and orientation)

Purpose

- Visual odometry important: no external reference system available (e.g GPS)
- Position control of quadrocopters (indoor)

Goal

Develop method that

- estimates motion of a RGB-D camera
- has high frame rates, low latency and is robust to outliers
- uses these estimates for local navigation and position control (of a quadrocopter)

State of the Art

Real-time visual odometry from dense RGB-D images [2]

- Minimize photo consistency error (energy formulation)
- Linearization of objective
- Assumes static scene

→ Real-time at 12.5 Hz (different hardware)

Kinect Fusion [3]

- Uses only depth map values
- Global model of the environment using ICP algorithm

→ Only applicable for small indoor rooms (memory issues)

→ GPU for real-time (update surface model + process new input)

Contributions

Probabilistic formulation for direct motion estimation (first time)

Robust sensor model from real world data

Integration of temporal prior

Runs in real-time (30 Hz) on a single CPU core (open-source implementation provided)

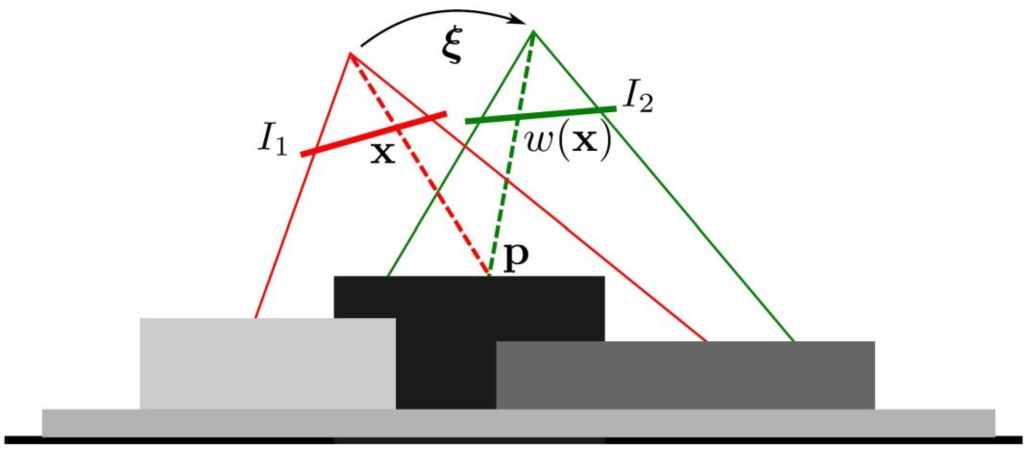
Demonstrating robustness of approach on several benchmarks

Direct Motion Estimation

Based on photo consistency-assumption:

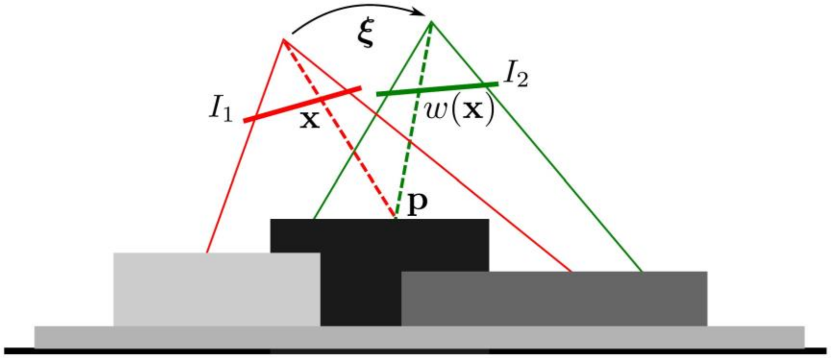
$$I_1(x) = I_2(\tau(\boldsymbol{\xi}, \boldsymbol{x}))$$

→ Goal: Find $\boldsymbol{\xi}$ (camera motion) that yields the best photo-consistency for all pixels



[4]

Warping Function



[4]

3D-point $\boldsymbol{p} = (x, y, z)^T \underset{\pi^{-1}}{\overset{\pi}{\rightleftarrows}}$ Pixel $\boldsymbol{x} = (u, v)^T$

Projection function π

$$\boldsymbol{x} = (u, v)^T = \pi(\boldsymbol{p}) = \left(\frac{f_x x}{z} + c_x, \frac{f_y y}{z} + c_y\right)^T$$

$$\boldsymbol{p} \; = \; \pi^{-1}\big(\boldsymbol{x}, Z_{\mathbf{1}}(\boldsymbol{x})\big) = Z_1(\boldsymbol{x}) \cdot \left(\frac{u - c_x}{f_x}, \frac{v - c_y}{f_y}, 1\right)^T$$

Point $\boldsymbol{p}$ viewed in second frame ($\boldsymbol{p}'$)

$$\boldsymbol{p}' = R\boldsymbol{p} + \boldsymbol{t}$$

Twist coordinates $\boldsymbol{\xi} \in \mathbb{R}^6$

$$\exp\big(\hat{\boldsymbol{\xi}}\big) = \begin{pmatrix} R & \boldsymbol{t} \\ 1 & 0 \end{pmatrix}$$

→

$$\tau(\boldsymbol{\xi}, \boldsymbol{x}) = \pi(R\boldsymbol{p} + \boldsymbol{t}) \; with \; \boldsymbol{p} = \pi^{-1}\big(\boldsymbol{x}, Z_{\mathbf{1}}(\boldsymbol{x})\big)$$

Likelihood Function

Residual of the i_{th} pixel

$$r_i(\boldsymbol{\xi}) = I_2\big(\tau(\boldsymbol{\xi}, \boldsymbol{x}_i)\big) - I_1(\boldsymbol{x}_i)$$

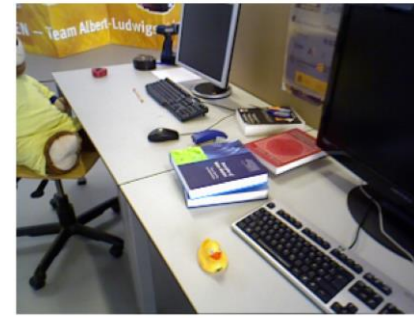
Residual image $\boldsymbol{r}$

$$p(\boldsymbol{r}|\boldsymbol{\xi}) = \prod_i p(r_i|\boldsymbol{\xi})$$

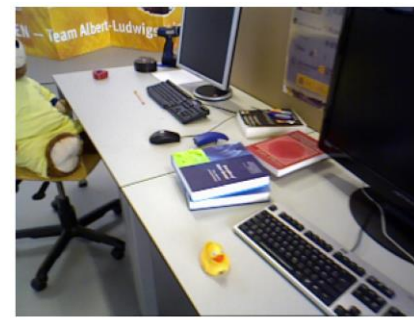
→ Bayes rule

$$p(\boldsymbol{\xi}|\boldsymbol{r}) = \frac{p(\boldsymbol{r}|\boldsymbol{\xi})p(\boldsymbol{\xi})}{p(\boldsymbol{r})}$$

Sensor model $p(\boldsymbol{r}|\boldsymbol{\xi})$
Motion prior $p(\boldsymbol{\xi})$



(a) reference image I_1



(b) current image I_2

(c) residuals

[4]

Maximum A Posteriori (MAP) Estimation

$$\boldsymbol{\xi}_{\boldsymbol{MAP}} = \arg\min_{\xi} -\log p(\boldsymbol{\xi}|\boldsymbol{r}) = \arg\min_{\xi} -\sum_i \log p(r_i|\boldsymbol{\xi}) - \log p(\boldsymbol{\xi})$$

Find minimizer (ignore prior for now):

$$\frac{\partial}{\partial \boldsymbol{\xi}} \log p(\boldsymbol{\xi}|\boldsymbol{r}) \overset{!}{=} 0 \rightarrow \sum_i \frac{\partial r_i}{\partial \boldsymbol{\xi}} w(r_i) r_i = 0 \quad with\, w(r_i) = \frac{1}{r_i} \cdot \frac{\partial}{\partial r_i} \log p(r_i|\boldsymbol{\xi})$$

→ Minimizer also minimizes the weighted least squares problem

$$\boldsymbol{\xi}_{\boldsymbol{MAP}} = \arg\min_{\xi} \sum_i w(r_i)(r_i(\boldsymbol{\xi}))^2$$

Solve minimization problem

→ iteratively reweighted least squares algorithm:

Alternate between computation of weights and estimates of $\boldsymbol{\xi}$ until convergence

Linearization

Need to solve:

$$\sum_i \frac{\partial r_i}{\partial \boldsymbol{\xi}} w(r_i) r_i = 0 \qquad (I)$$

Problem: r_i non-linear in $\boldsymbol{\xi}$

→ Linearize $\boldsymbol{\xi}$:

$$r_{lin}(\boldsymbol{\xi}, \boldsymbol{x}_i) = r(\boldsymbol{0}, \boldsymbol{x}_i) + \left. \frac{\partial r\big(\tau(\boldsymbol{\xi}, \boldsymbol{x}_i)\big)}{\partial \boldsymbol{\xi}} \right|_{\boldsymbol{\xi}=\boldsymbol{0}} \Delta\boldsymbol{\xi}$$

$$= r(\boldsymbol{0}, \boldsymbol{x}_i) + J_i \Delta\boldsymbol{\xi}$$

→ Into (I)

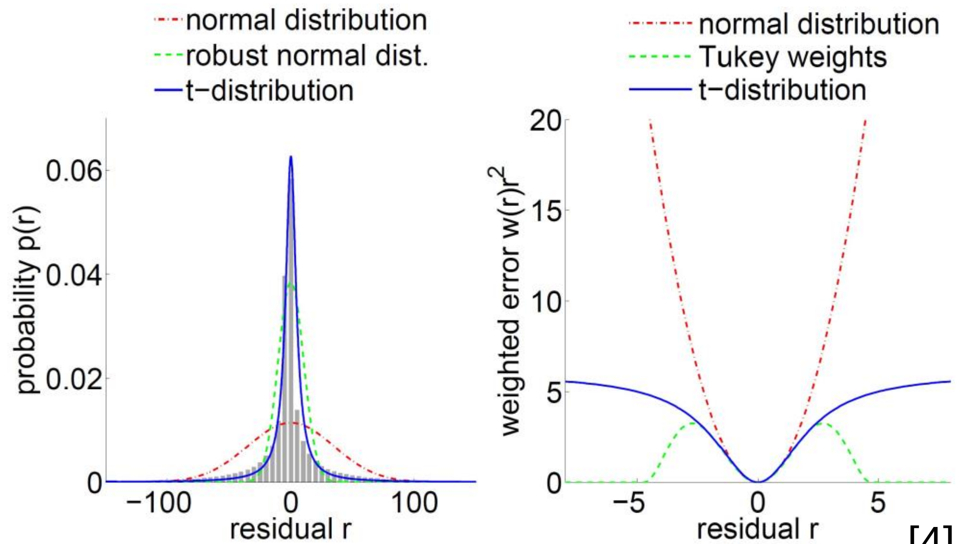
$$J^T W J \Delta\boldsymbol{\xi} = -J^T W \boldsymbol{r}(\boldsymbol{0})$$

Update parameters:

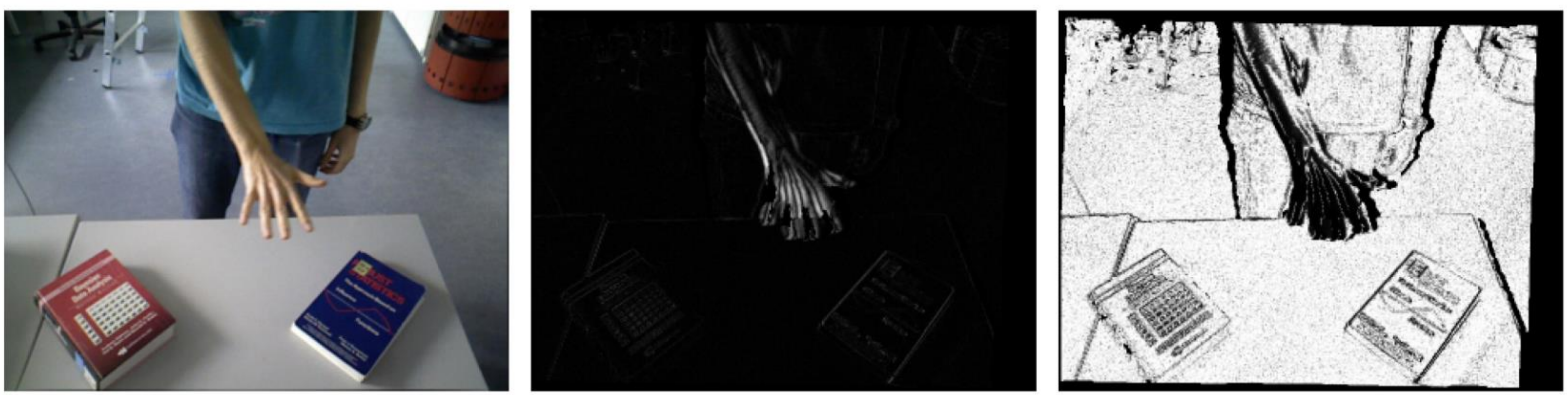
$$\boldsymbol{\xi}^{(k+1)} = \log\Big(\exp\big(\boldsymbol{\xi}^{(k)}\big) \exp\big(\Delta\boldsymbol{\xi}^{(k)}\big)\Big)$$

Robust Estimation

Sensor model $p(r|\xi)$
→ t-distribution fits the assumptions better



[4]



[4]

(a) scene (b) residuals (c) weights

Robust Estimation

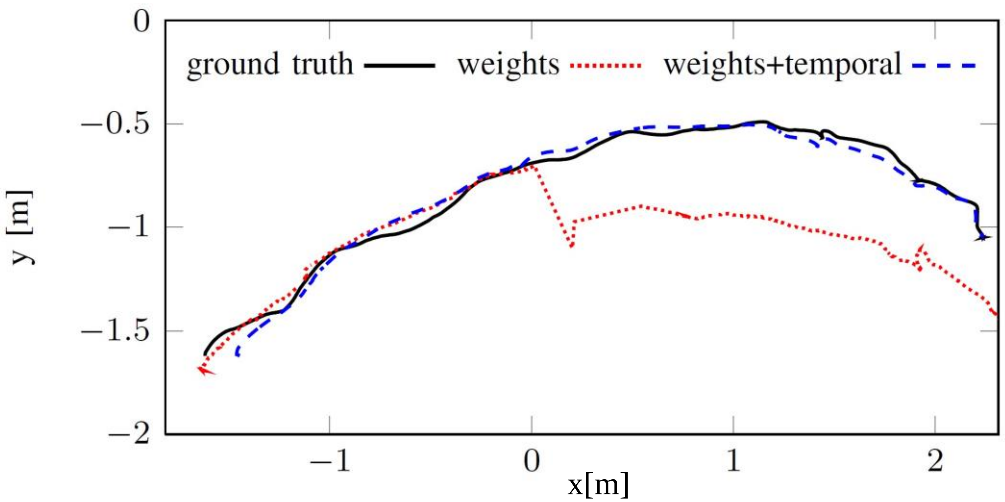
Motion prior $p(\xi)$

Temporal prior based on constant velocity model with a normal distribution:

$$p(\boldsymbol{\xi}_t) = \mathcal{N}(\boldsymbol{\xi}_{t-1}, \Sigma)$$

→Normal equation:

$$\left(J^T W J + \Sigma^{-1}\right)\Delta\boldsymbol{\xi} = -J^T W \boldsymbol{r}(\mathbf{0}) + \Sigma^{-1}\left(\boldsymbol{\xi}_{t-1} - \boldsymbol{\xi}_t^{(k)}\right)$$



[4]

Evaluation

Synthetic Sequences

- Generated perfect ground truth (two types)
  - Static scene
  - Small dynamic object
- Focus on real-time parameters

Conclusion

→ Robust weighting important for noise and outliers

→ t-distribution better suited than gaussian distribution (no weights)

static

Method	Parameter set	RMSE [m/s]	Improvement	ØRuntime [s]
reference		0.0751	0.0 %	0.038
no weights	realtime	0.0223	70.0 %	0.032
Tukey weights	realtime	0.0497	33.8 %	0.050
t-dist. weights	realtime	**0.0142**	81.1 %	0.043

dynamic

Method	Parameter set	RMSE [m/s]	Improvement	ØRuntime [s]
reference		0.1019	0.0 %	0.035
no weights	realtime	0.0650	36.2 %	0.030
Tukey weights	realtime	0.0382	62.5 %	0.056
t-dist. weights	realtime	**0.0296**	71.0 %	0.047

Evaluation

TUM RGB-D Benchmark Sequences

- Real world data tracked by highly accurate motion capture system
- Focus on real-time parameters

Conclusion

Temporal prior: Particularly useful when many outliers are present or high velocity of camera

static vs. dynamic

Method	ØDrift [m/s]	Improvement
reference	0.2425	0.0 %
no weights	0.0588	75.74 %
Tukey weights	0.1491	38.50 %
t-dist. weights	0.0432	82.18 %
t-dist. weights + temporal	**0.0428**	**82.35 %**

Method	ØDrift [m/s]	Improvement
reference	0.0800	0.0 %
no weights	0.0350	56.23 %
Tukey weights	0.1038	-29.66 %
t-dist. weights	0.0470	41.33 %
t-dist. weights + temporal	**0.0316**	**60.55 %**

Summary

Probabilistic formulation

- Directly estimate the camera motion from RGB-D images
- Use of custom probability distributions for the sensor and motion model possible

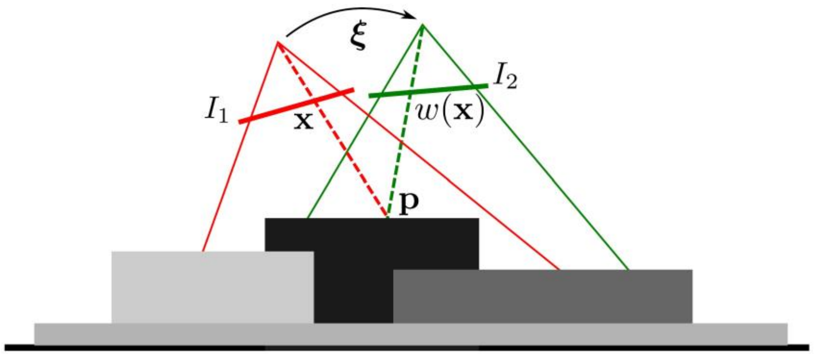
Increase **robustness**

- t-distribution (address outliers)
- Temporal prior (guides optimization)

Fast

Runs in real-time (30 Hz) on a single CPU core

Discussion



$$r_i(\boldsymbol{\xi}) = I_2\big(\tau(\boldsymbol{\xi}, \boldsymbol{x}_i)\big) - I_1(\boldsymbol{x}_i)$$

$$p(\boldsymbol{\xi}|\boldsymbol{r}) \propto p(\boldsymbol{r}|\boldsymbol{\xi})p(\boldsymbol{\xi})$$

$\boldsymbol{\xi}_{MAP}$

- Iteratively reweighted LS
- Linearization
- Coarse-to-fine image pyramid

Evaluation

- $p(\boldsymbol{\xi}|\boldsymbol{r})$: t-distribution
- $p(\boldsymbol{\xi})$: temporal prior

References

- [1] https://www.rcmoment.com/p-rm11238w.html
- [2] F. Steinbrücker, J. Sturm, and D. Cremers, “Real-time visual odometry from dense RGB-D images,” in Workshop on Live Dense Reconstruction with Moving Cameras at the Intl. Conf. on Computer Vision (ICCV), 2011
- [3] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon, “KinectFusion: Real-time dense surface mapping and tracking,” in IEEE Intl. Symposium on Mixed and Augmented Reality (ISMAR), 2011
- [4] and where no different source is explicitly stated: C. Kerl, J. Sturm, and D. Cremers, “Robust odometry estimation for RGB-D cameras,” in IEEE Intl. Conf. on Robotics and Automation (ICRA), 2013.

t-distribution

$$p(r) = \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\sqrt{\nu\pi}\,\Gamma\left(\frac{\nu}{2}\right)}\left(1+\frac{r^2}{\nu}\right)^{-\frac{\nu+1}{2}}$$

→

$$w(r) = \frac{1}{r}\cdot\frac{\partial}{\partial r}\log p(r)$$

$$= \frac{\nu+1}{\nu+\left(\frac{r}{\sigma}\right)^2}$$

With

$$\sigma^2 = \frac{1}{n}\sum_i r_i^2 \frac{\nu+1}{\nu+\left(\frac{r}{\sigma}\right)^2}$$

(has to be solved iteratively, but converges fast)

Experiments: $\nu = 5$ good

Twist coordinates

$$\boldsymbol{\xi} = \begin{pmatrix} \upsilon_1 \\ \upsilon_2 \\ \upsilon_3 \\ \omega_1 \\ \omega_2 \\ \omega_3 \end{pmatrix}$$

$$exp(\hat{\boldsymbol{\xi}}) = exp\begin{pmatrix} 0 & -\omega_3 & \omega_2 & \upsilon_1 \\ \omega_3 & 0 & -\omega_1 & \upsilon_2 \\ -\omega_2 & \omega_1 & 0 & \upsilon_3 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$exp(\hat{\boldsymbol{\xi}}) = \begin{pmatrix} R & \boldsymbol{t} \\ 1 & 0 \end{pmatrix}$$

Video

https://www.youtube.com/watch?v=TMqPwoCCmto

Synthetic Sequences

static

Method	Parameter set	RMSE [m/s]	Improvement	ØRuntime [s]
reference		0.0751	0.0 %	0.038
no weights	realtime	0.0223	70.0 %	0.032
Tukey weights	realtime	0.0497	33.8 %	0.050
t-dist. weights	realtime	**0.0142**	81.1 %	0.043
no weights	precision	0.0145	80.7 %	0.115
Tukey weights	precision	0.0279	62.8 %	0.230
t-dist. weights	precision	**0.0124**	83.5 %	0.405

dynamic

Method	Parameter set	RMSE [m/s]	Improvement	ØRuntime [s]
reference		0.1019	0.0 %	0.035
no weights	realtime	0.0650	36.2 %	0.030
Tukey weights	realtime	0.0382	62.5 %	0.056
t-dist. weights	realtime	**0.0296**	71.0 %	0.047
no weights	precision	0.0502	50.7 %	0.130
Tukey weights	precision	0.0270	73.5 %	0.279
t-dist. weights	precision	**0.0133**	87.0 %	0.508

Real-time visual odometry from dense RGB-D images

Energy formulation

$$E(\boldsymbol{\xi}) = \int I_2\big(\tau(\boldsymbol{\xi}, \boldsymbol{x})\big) - I_1(\boldsymbol{x})$$

→ Linearize $I_2\big(\tau(\boldsymbol{\xi}, \boldsymbol{x})\big)$ and $\tau(\boldsymbol{\xi}, \boldsymbol{x})$ around $\boldsymbol{\xi} = \mathbf{0}$

Rigid body motion g

$$E(\boldsymbol{\xi}) \sim \int \left(I_2(\boldsymbol{x}) - I_1(\boldsymbol{x}) + \nabla I_2^T \left(\frac{d\pi}{dg}\right) \left(\frac{dg}{d\boldsymbol{\xi}}\right) \boldsymbol{\xi} \right)^2$$

→ Normal equations