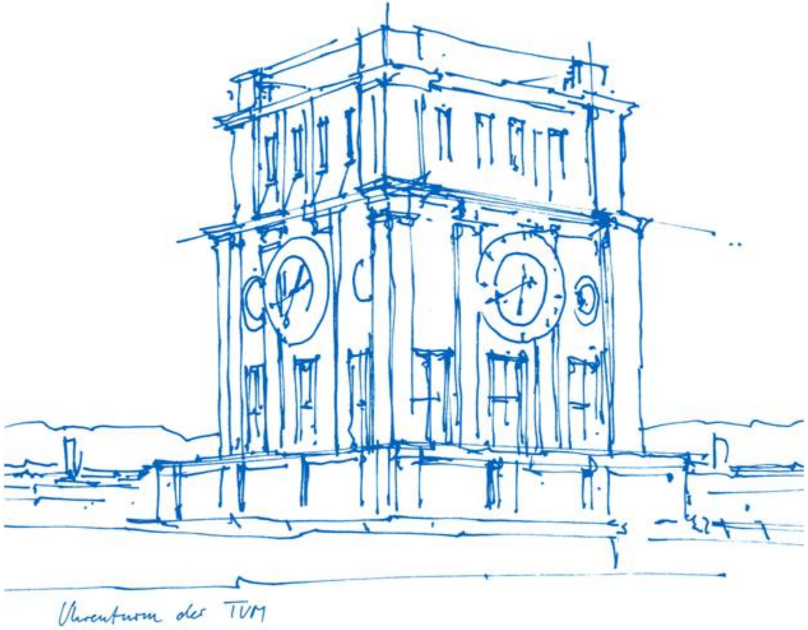
Master-Seminar - The Evolution of Motion Estimation and Real-time 3D Reconstruction



Markus Feurstein

Garching, 23. October 2019

Paper Presented

Dense Visual SLAM for RGB-D Cameras

Christian Kerl, Jurgen Sturm, and Daniel Cremers, 2013 IEEE/RSJ

Charts without source indicated stem from this paper

Depth Camera Images

Generates 2 data sets:

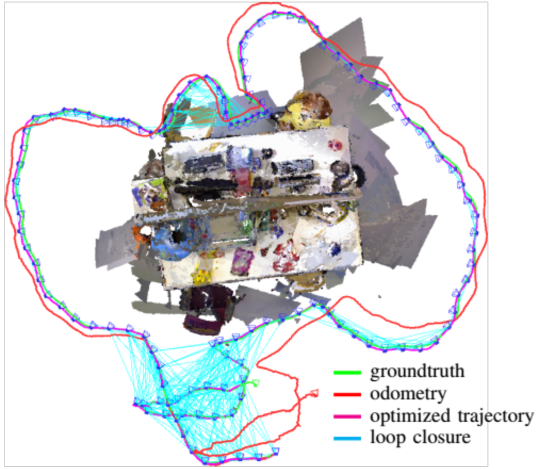
- Intensity Image
- Depth image

Rate: 10-30 Hz

Source: Peter Henry et.al. RGB-D Mapping: Using Depth Cameras for Dense 3D Modeling of Indoor Environments, 2014

Motivation

- Want to know the pose (Location & Orientation) of the camera relative to world frame in real time
- System should be stable: Going around a closed loop should close the trajectory



Solution:
Photometric and geometric errors are used **simultaneously** using a **Bayesian approach**:

- Photometric model works well for scenes with texture
- Geometric model is works well for scenes with structure

(a) texture (b) structure (c) structure + texture

Keyframe approach to define global map
Entropy based measures to select keyframes and loop closure
Graph optimization

Defining the Pose of the Camera

Estimate the frame to frame transformation
T between image frames k and k+1

$$T_k^{k+1} = \begin{bmatrix} R_k^{k+1} & t_k^{k+1} \\ 0 & 1 \end{bmatrix}$$

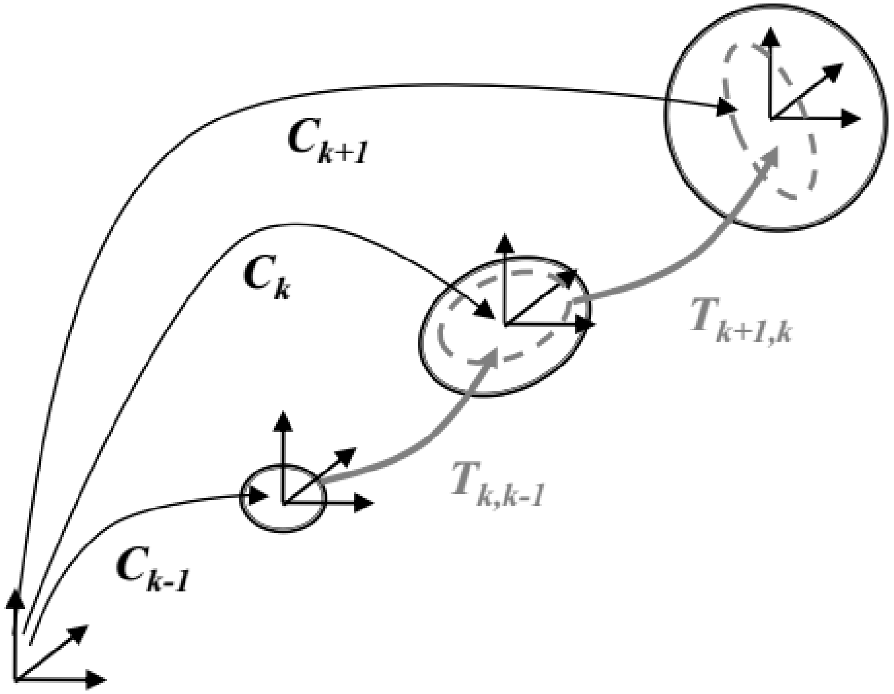
T has 6 DOF. Can be parameterized with 6 twist coordinates under Lie algebra *se(3)*

$$T(\xi) = e^{\hat{\xi}}$$

The pose C_k of the camera is given by concatenation

$$C_k = T_{k-1}^{k} * T_{k-2}^{k-1} * .... * T_0^1$$

Problem: Lots of errors are accumulated



Source:Davide Scaramuzza , VO Tutorial 2011

Pinhole Camera Model

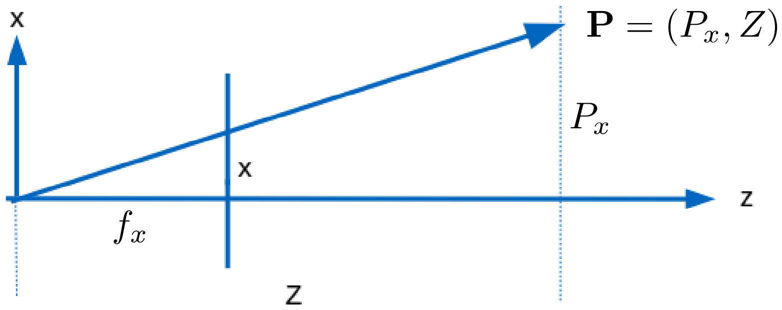
Back Projection of pixel to 3D point $\mathbf{P} = (P_x, P_y, Z)$

$$\mathrm{P} = \Pi^{-1}(\mathbf{x}, Z) = Z * (\frac{x+c_x}{f_x}, \frac{y+c_y}{f_y}, 1)$$

Known from depth image

Projection from 3D point **P** to pixel coordinates **x**

$$\mathbf{x} = \Pi(\mathbf{P}) = (\frac{P_x * f_x}{Z} - c_x, \frac{P_y * f_y}{Z} - c_y)$$



$$\frac{P_x}{Z} = \frac{x}{f_x}$$

The Warping Function

Need relation between projections x and x' of point P

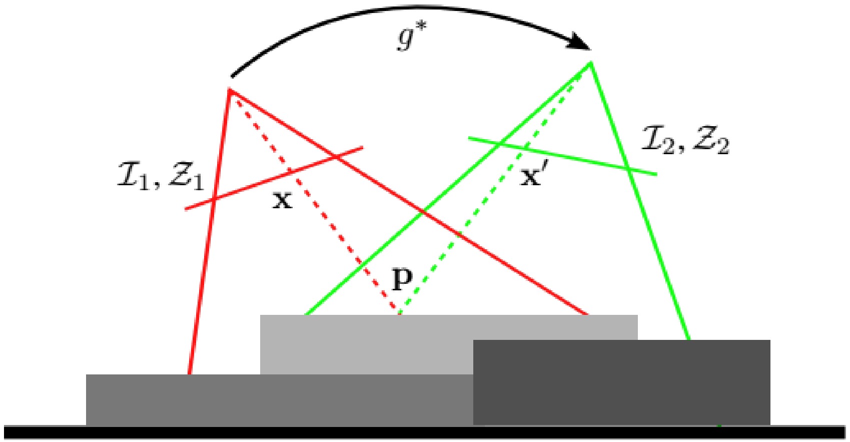
$$\mathbf{P} = \Pi^{-1}(\mathbf{x}, \mathcal{Z}_1)$$

$$\mathbf{P}' = T(\xi, \mathbf{P}) = \mathbf{R} * \mathbf{P} + \mathbf{t}$$

$$\mathbf{x}' = \Pi(\mathbf{P}')$$

Plugging all in gives warping function

$$\mathbf{x}' = \tau(\mathbf{x}, \xi) = \Pi(T(\xi, \Pi^{-1}(\mathbf{x}, \mathcal{Z}_1)))$$



Dense Estimation of Frame to Frame Transforms T

Photometric error:

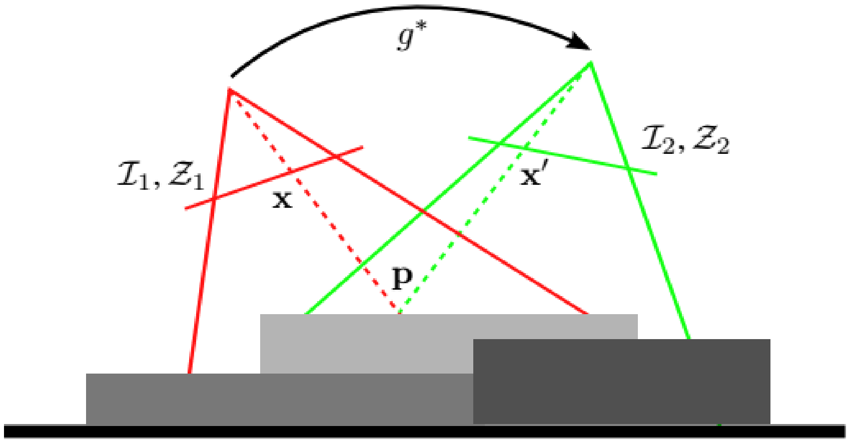
The intensity of point P seen in subsequent Images should be the same

$$r_i^I(\xi) = I_2(\tau(\mathbf{x_i}, \xi)) - I_1(\mathbf{x_i})$$

Geometric error:

The depths of x’ in image 2 should be the same as the depths of P transformed from x.

$$r_i^Z(\xi) = \mathcal{Z}_2(\tau(\mathbf{x_i}, \xi)) - [T\Pi^{-1}(\mathbf{x_i}, \mathcal{Z}_1(\mathbf{x_i}))]_z$$

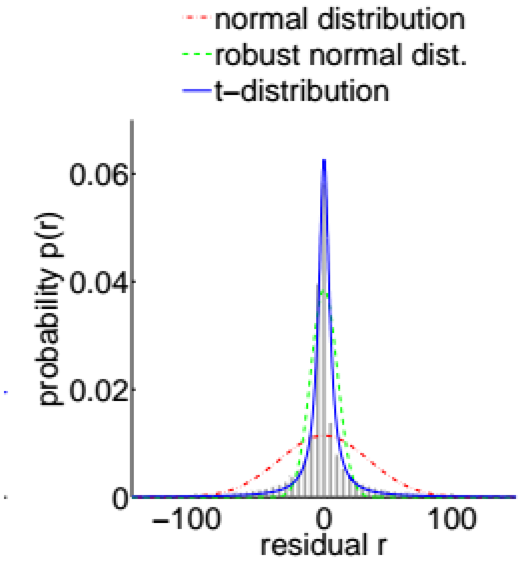


Simple Parameter Estimation:

Could use Ordinary Least Square to estimate parameters:

$$\xi^*_{OLS} = \underset{\xi}{\operatorname{argmin}} \sum_{i=1}^{N} (r_i(\xi))^2$$

Residuals not Gaussian!



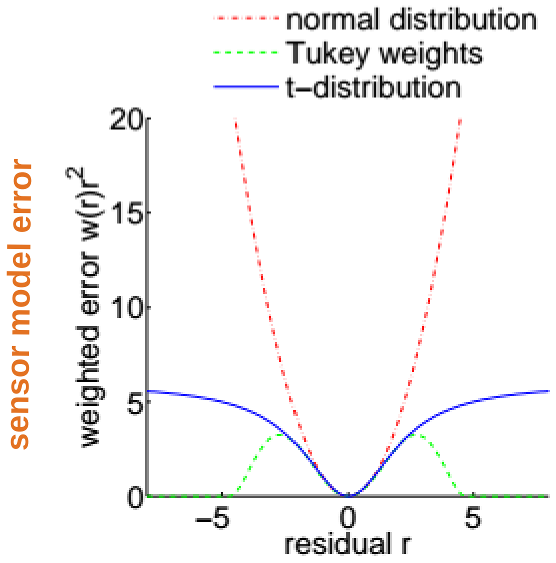
Bayesian Parameter Estimation

Assuming iid. maximize posterior likelihood

$$\xi^*_{MAP} = \underset{\xi}{\operatorname{argmax}} \sum_{i=1}^{N} \log(p(\xi|r_i)) \qquad p(\xi|r_i) = \frac{p(r_i|\xi)p(\xi)}{p(r_i)}$$

$$\xi^*_{MAP} = \underset{\xi}{\operatorname{argmin}} \sum_{i=1}^{N} -\log(p(r_i|\xi)) - \log(p(\xi))$$

Sensor Model **Motion Prior**



Advantages

- Arbitrary distribution for sensor noise
- Can include prior knowledge on motion (not used)

Sensor Model

Contains both, photometric and geometric error

$$\mathbf{r} = (\mathbf{r}^I, \mathbf{r}^Z)$$

Modeled as bivariate t-distribution with unknown scale matrix $\mathbf{\Sigma}$

Can be formulated as iteratively weighted least square:

$$\xi^* = \underset{\xi}{\operatorname{argmin}} \sum_{i=1}^{N} w_i \mathbf{r_i}^{\mathbf{T}} \Sigma^{-1} \mathbf{r_i} \qquad \text{with} \qquad w_i = \frac{\nu + 1}{\nu + \mathbf{r_i}^{\mathbf{T}} \mathbf{\Sigma}^{-1} \mathbf{r_i}}$$

Advantages:

- Weighting between photometric and geometric error automatically optimized
- Outliers are down weighted if either error component is large

Transformation Parameter Estimation

Use Gauss Newton algorithm:
1) Linearize **r** by taylor series expansion

2) Plug linearized **r** into sensor model and
set derivative wrt. $\Delta\xi$ to 0
$\rightarrow$ normal equations:

Iteratively solve by EM algorithm for t-distribution:

1) E-Step:At iteration step s, we know $\xi_\mathbf{s}$
- Update $\mathbf{r_i}(\xi_\mathbf{s}, \mathbf{x_i}),\ \mathbf{\Sigma}(\mathbf{r}, \mathbf{x}),\ w(\mathbf{r_i}, \mathbf{\Sigma}),\ \mathbf{J_i}(\xi_\mathbf{s}, \mathbf{x_i})$

2) M-Step: We know normal equation
- Solve for $\Delta\xi$ $\rightarrow$ $\xi_\mathbf{s+1} = \xi_\mathbf{s} + \Delta\xi$

3) s = s+1
- go to 1)

$$\mathbf{r}(\xi, \mathbf{x_i}) \approx \mathbf{r}(\mathbf{0}, \mathbf{x_i}) + \mathbf{J_i}\mathbf{\Delta}\xi$$

$$\mathbf{J_i} = \frac{\partial \mathbf{r}(\tau(\mathbf{x_i}, \xi))}{\xi}\Delta\xi$$

$$\sum_{i=1}^{N} w_i \mathbf{J_i^T} \mathbf{\Sigma^{-1}} \mathbf{\Delta}\xi = -\sum_{i=1}^{N} w_i \mathbf{J_i^T} \mathbf{\Sigma^{-1}} \mathbf{r_i}$$

Nice:
We get an estimate for parameter uncertainty for free (information theory)

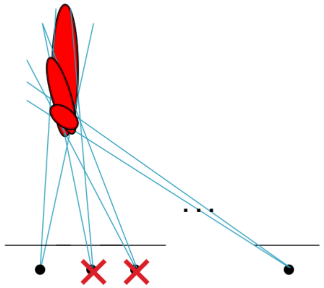
$$\Sigma_\xi^{-1} \approx \sum_{i=1}^{N} w_i \mathbf{J_i^T} \mathbf{\Sigma^{-1}}$$

(needed for graph optimization later)

Keyfame-based SLAM

Problem: frame-to-frame transformation accumulate lots of errors

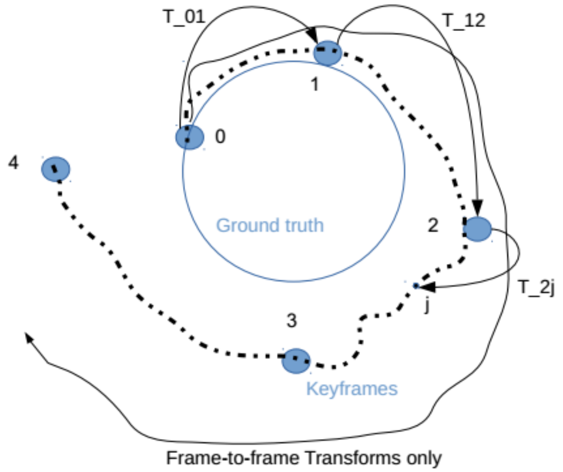
- Short baseline
- No relation between frame-to-frame transformations



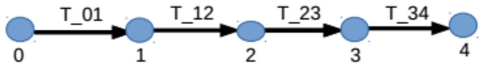
Source:Davide Scaramuzza
VO Tutorial 2011

Solution:

1) Lengthen baseline
- Certain frames are selected as keyframes (How?)
- Pose of frame j is based on keyframes



2) Define keyframe map
- Keyframes are the nodes of a graph
- Edges are the frame transformations between nodes

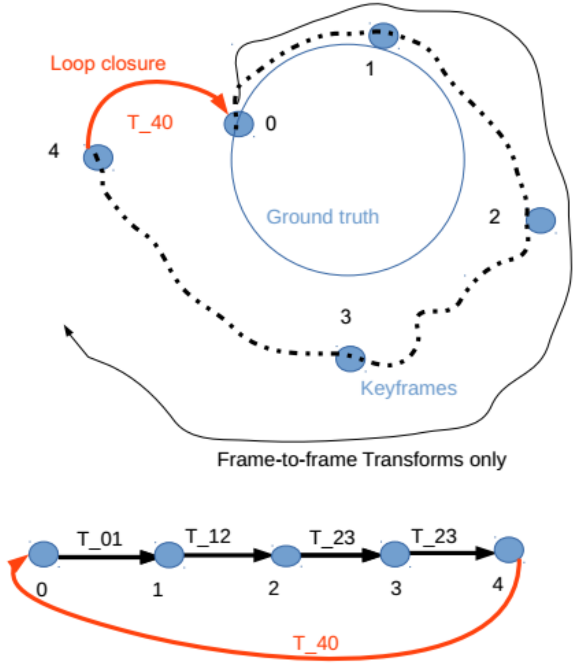


→ Linear graph

Keyfame-based SLAM

2) Detect loops:
New keyframe (4) is compared with other nodes (0) to detect loops, If test is successful:
→ add new edge (orange) to graph
→ Transformation along loop not consistent

3) Optimize graph along loop:
- g2o framework
- weights proportional to parameter uncertainty Σ_ξ



Key Frame Selection

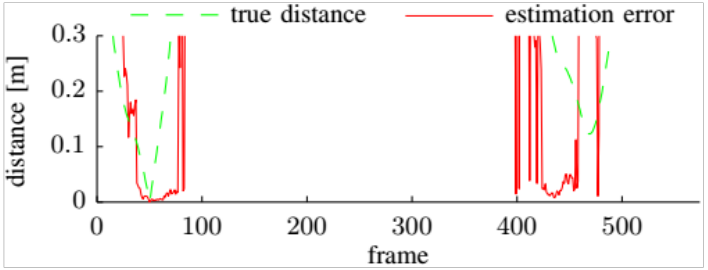
Is the current frame a keyframe?

- Need a measure of uncertainty between last keyframe k and current frame j

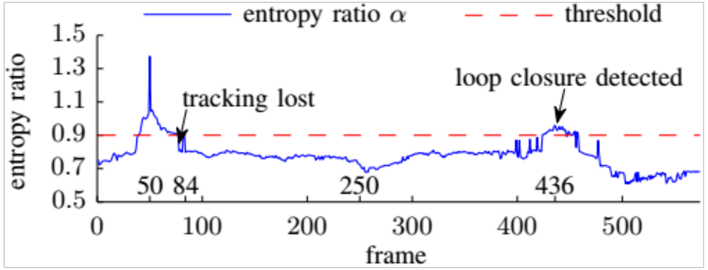
- Differential Entropy $H(k,j) \propto log(|\Sigma_{k,j}|)$

Scene invariant measure: $$\alpha(k,j) = \frac{H(k,j)}{H(k,k+1)}$$

If $\alpha > 0.9$
→ current frame j becomes the new keyframe k+1



(a) estimate error w.r.t. frame 50



Closure Detection

Is the new keyframe k+1 closing a loop?

1) Selection of candidates:

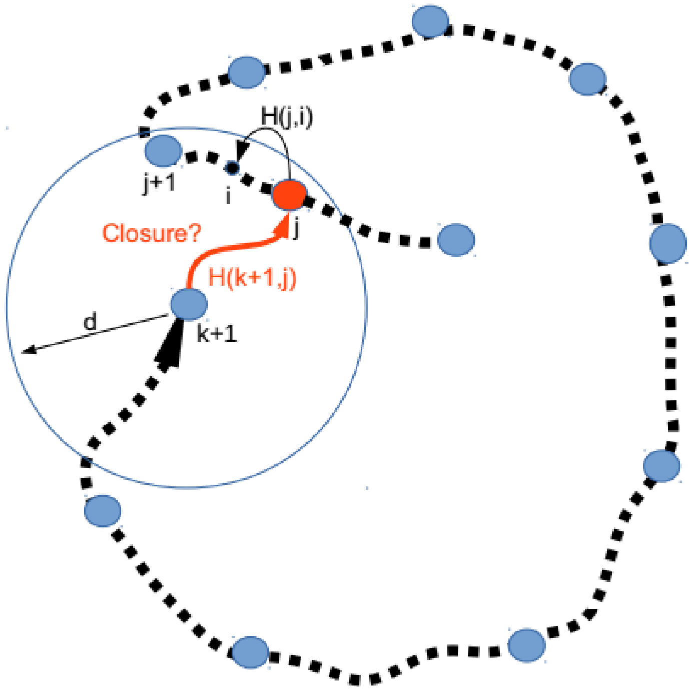
- Take keyframes within metric distance d

2) For each candidate j, evaluate ratio:

$$\beta(k+1,j) = \frac{H(k+1,j)}{\underset{i}{\text{mean}}(H(j,i))}, \quad j < i < j+1$$

3) Decide:

- If $\beta > 0.9$ → loop closed → insert edge



Results

1) Comparison between bivariate and univariate RMSE [m/s]:

- Outperforms photo and geo method on data with only structure or texture.
- Better generalization over different scenes

2) Influence of keyframe and map optimization RMSE [m/s]:

- Keyframe map brings 16% improvement
- Map optimization gives additional 4% (But big improvement in absolute trajectory error)

3) Relation of state of the art Visual Slam RMSE [m]:

- Absolute trajectory error best in class for most data sets

structure	texture	distance	RGB	Depth	RGB+Depth
-	x	near	0.0591	0.2438	**0.0275**
-	x	far	0.1620	0.2870	**0.0730**
x	-	near	0.1962	0.0481	**0.0207**
x	-	far	0.1021	0.0840	**0.0388**
x	x	near	**0.0176**	0.0677	0.0407
x	x	far	**0.0170**	0.0855	0.0390

Dataset	RGB+D	RGB+D+KF	RGB+D+KF+Opt
fr1/desk	0.036	0.030	**0.024**
fr1/desk (v)	0.035	0.037	**0.035**
fr1/desk2	**0.049**	0.055	0.050
fr1/desk2 (v)	0.020	0.020	**0.017**
fr1/room	0.058	0.048	**0.043**
fr1/room (v)	0.076	**0.042**	0.094
fr1/360	0.119	0.119	**0.092**
fr1/360 (v)	0.097	0.125	**0.096**
fr1/teddy	0.060	0.067	**0.043**
fr1/floor	fail	**0.090**	0.232
fr1/xyz	0.026	0.024	**0.018**
fr1/xyz (v)	**0.047**	0.051	0.058
fr1/rpy	0.040	0.043	**0.032**
fr1/rpy (v)	0.103	0.082	**0.044**
fr1/plant	0.036	0.036	**0.025**
fr1/plant (v)	0.063	**0.062**	0.191
avg. improvement	0%	16%	20%

Dataset	# KF	Ours	RGB-D SLAM	MRSMap	KinFu
fr1/xyz	68	**0.011**	0.014	0.013	0.026
fr1/rpy	73	**0.020**	0.026	0.027	0.133
fr1/desk	67	**0.021**	0.023	0.043	0.057
fr1/desk2	93	0.046	**0.043**	0.049	0.420
fr1/room	186	**0.053**	0.084	0.069	0.313
fr1/360	126	0.083	0.079	**0.069**	0.913
fr1/teddy	181	**0.034**	0.076	0.039	0.154
fr1/plant	156	0.028	0.091	**0.026**	0.598
fr2/desk	181	**0.017**	-	0.052	-
fr3/office	168	**0.035**	-	-	0.064
average		**0.034**	0.054	0.043	0.297

Summary

- Probabilistic formulation for visual SLAM based on dense RGB-D images.
- Big advantage: ability to run in real time.

- The approach uses the photometric and the geometric error simultaneously to esimate frame to frame transformation.

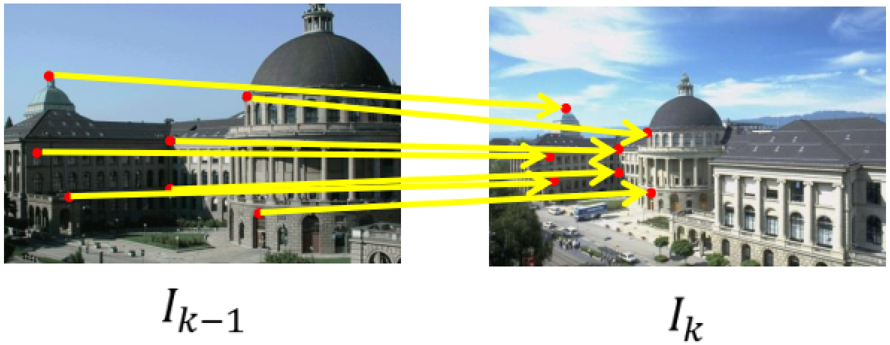
→ Big improvement in generalization over other methods that set weights manually

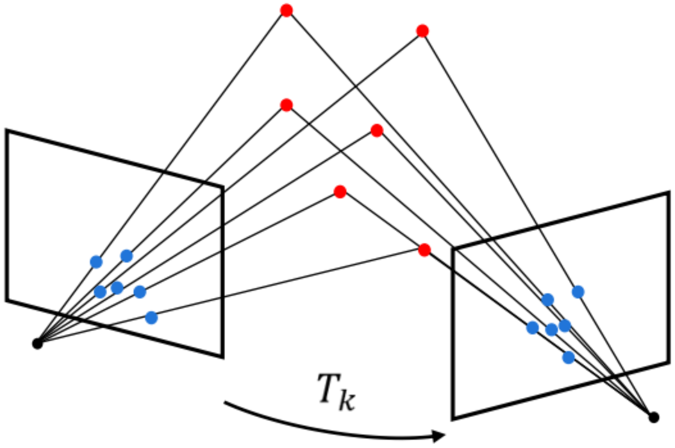
- The performance of the methodology proposed outperforms other state-of-the-art algorithms on most benchmarks.

- The combination of the bivariate approach and the optimized keyframe map shines in terms of absolute trajectory error for long and complex datasets.

Backup Slides

Feature based SLAM





Source:Davide Scaramuzza , VO Tutorial 2011