Robotic 3D Vision

Lecture 8: Visual Odometry 3 – Direct Methods

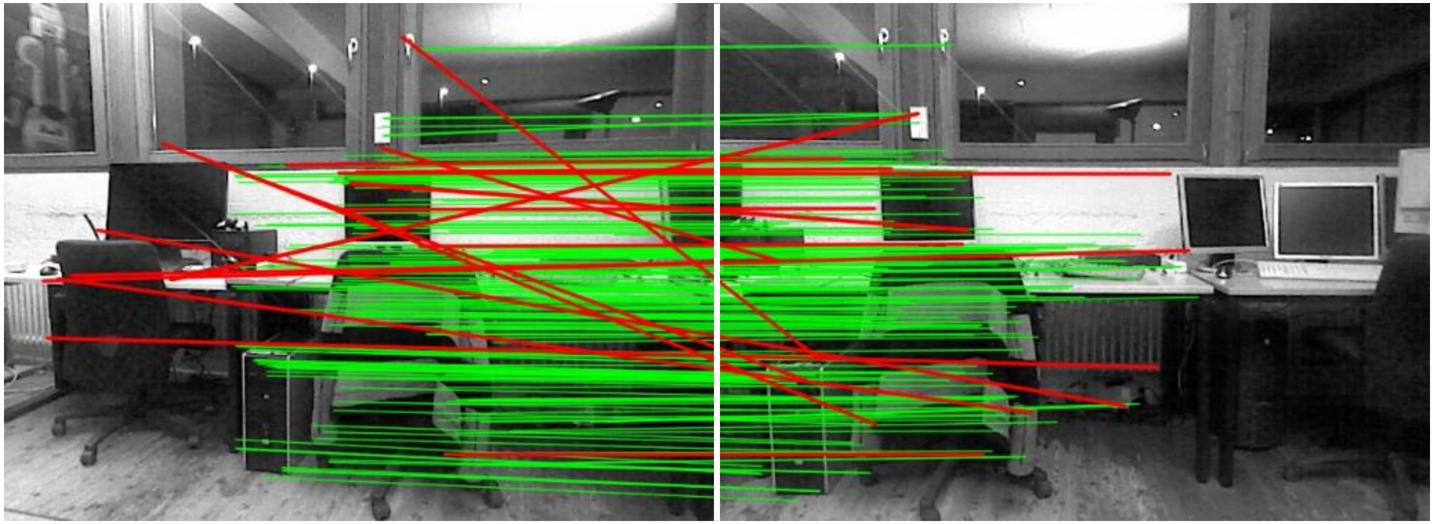
WS 2020/21

Dr. Niclas Zeller
Artisense GmbH

What We Will Cover Today

- RANSAC (leftover from last lecture)
- Direct visual odometry methods
  - Principles of direct image alignment
  - Photometric alignment
  - Geometric alignment
- Direct visual odometry for RGB-D cameras
- Direct visual odometry for monocular cameras
  - Semi-dense monocular odometry
- Photometric calibration
-
- Stereo extensions

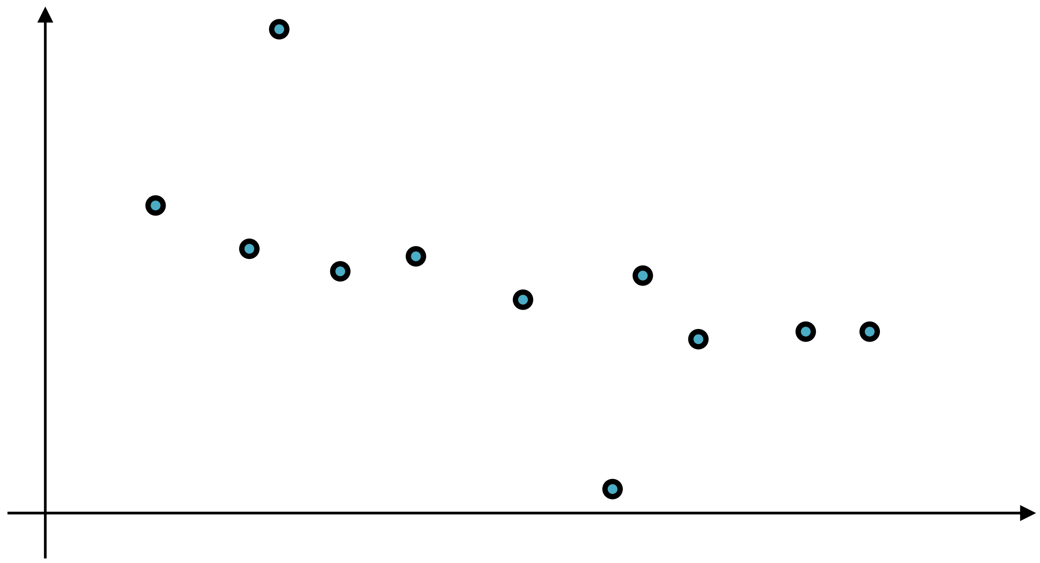
Recap: Keypoint Matching



- Only accept matches with distance smaller a threshold
- What else can we do?

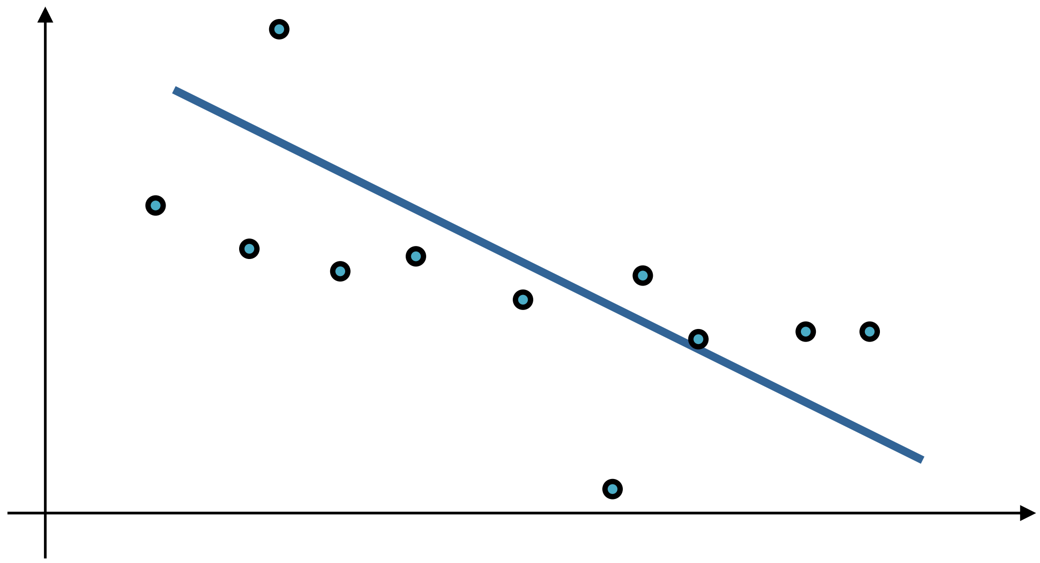
Random Sample Consensus (RANSAC)

- Model fitting in presence of noise and outliers
- Example: fitting a line through 2D points



RANSAC

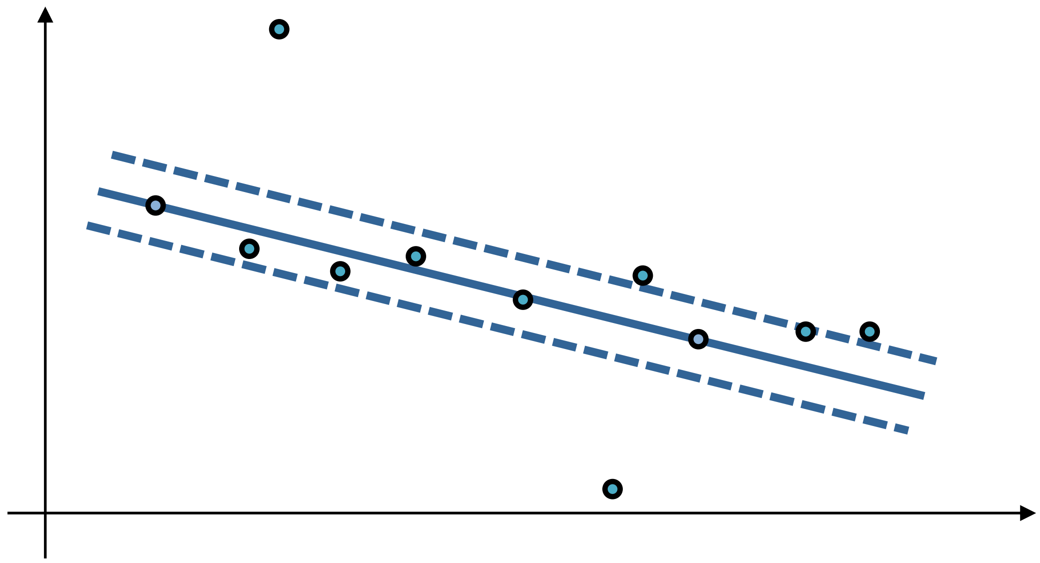
- Least-squares solution, assuming constant noise for all points



Bad!

RANSAC

- We only need 2 points to fit a line. Let's try 2 random points

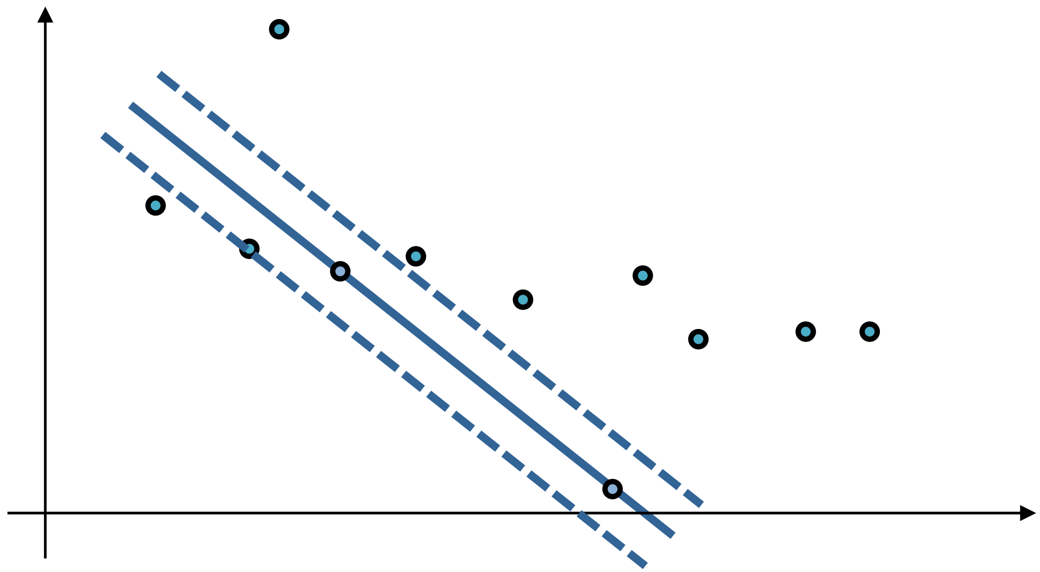


Quite ok

7 inliers
4 outliers

RANSAC

- Let's try 2 other random points

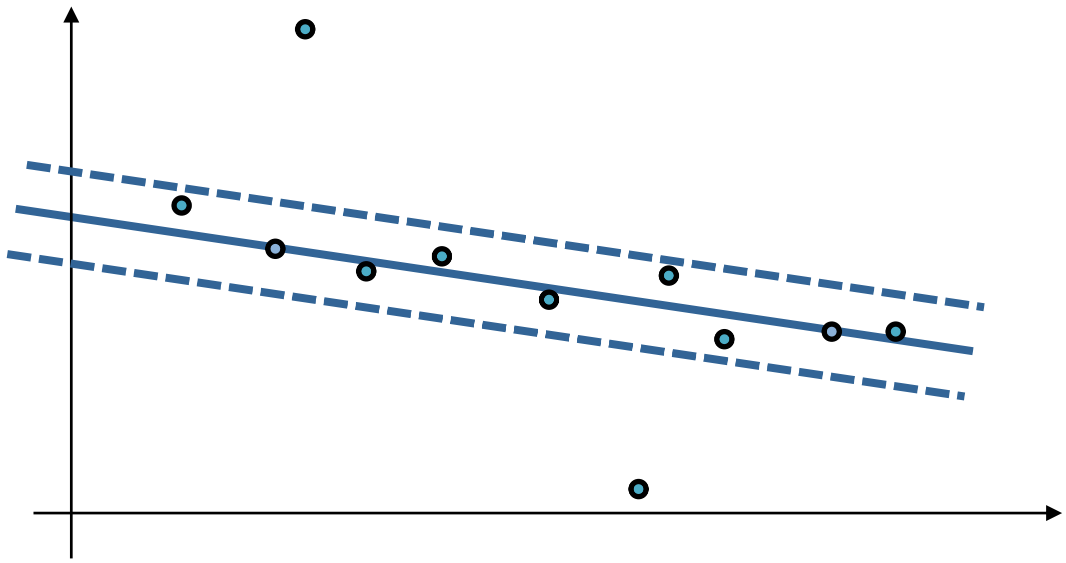


Quite bad

3 inliers
8 outliers

RANSAC

- Let's try yet another 2 random points

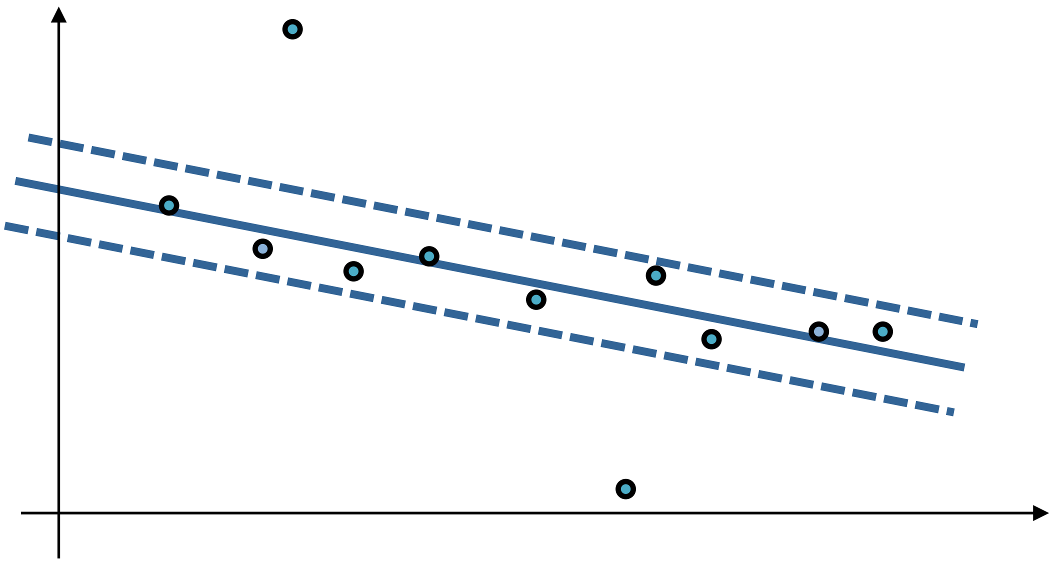


Quite good!

9 inliers
2 outliers

RANSAC

- Let’s use the inliers of the best trial so far to perform least squares fitting



Even better!

RANSAC

- How many iteration do we need to find the optimal solution
  - p - probability of finding the correct solution
  - ϵ - outlier ration → $w = 1 - \epsilon$ (inlier ratio)
  - s - number of data points required to calculate solution
  - N - number of iterations

$$\underbrace{1 - p}_{\text{Probability of not a single correct solution}} = (\overbrace{1 - w^s}^{\text{Probability of picking at least one outlier}})^N = (1 - (1 - \epsilon)^s)^N$$

Probability of picking s inliers

$$N \geq \frac{\log(1 - p)}{\log(1 - (1 - \epsilon)^s)}$$

RANSAC Algorithm

- RANdom SAmple Consensus algorithm formalizes this idea
- Algorithm:

Input: data D, s required #data points for fitting, success probability p, outlier ratio ϵ

Output: inlier set

1. Compute required number of iterations $N \geq \frac{\log(1-p)}{\log(1-(1-\epsilon)^s)}$

2. For N iterations do:
   1. Randomly select a subset of s data points
   2. Fit model on the subset
   3. Count inliers and keep model/subset with largest number of inliers

3. Refit model using found inlier set

RANSAC

N for $p = 0.99$

	Required points s	Outlier ratio ϵ 10%	20%	30%	40%	50%	60%	70%
Line	2	3	5	7	11	17	27	49
Plane	3	4	7	11	19	35	70	169
Essential matrix	8	9	26	78	272	1177	7025	70188

Direct Visual Odometry Pipeline

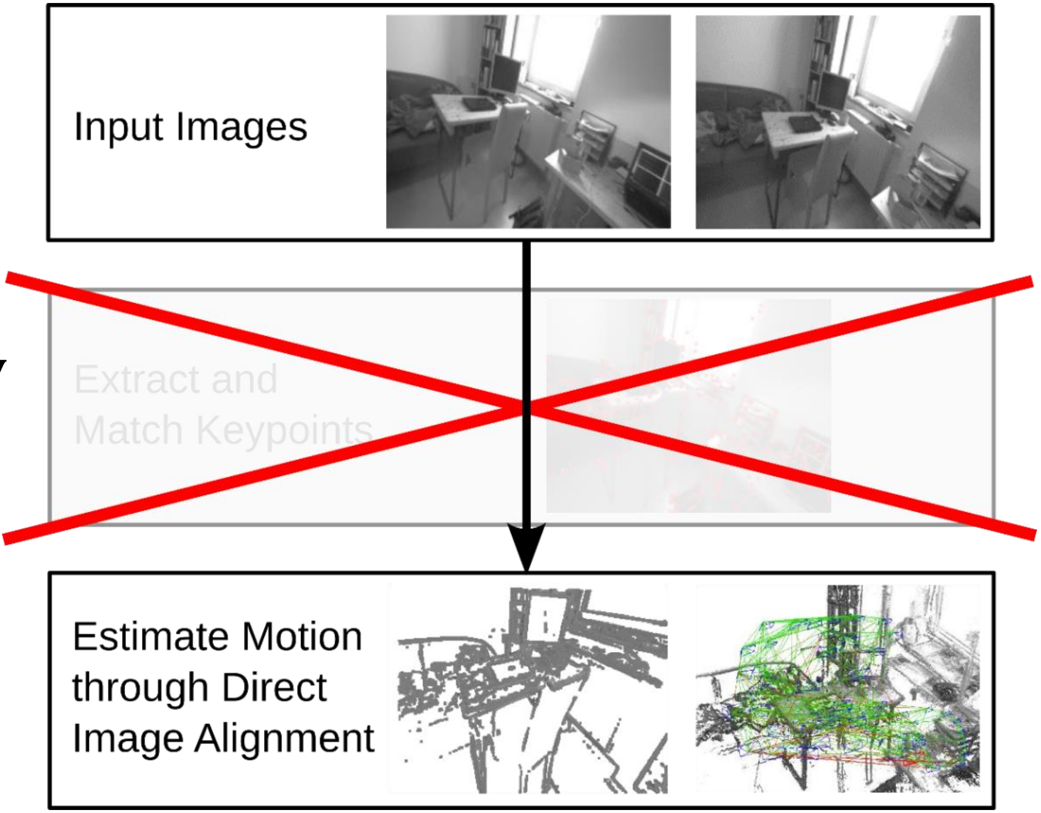
- Avoid manually designed keypoint detection and matching

- Instead: direct image alignment

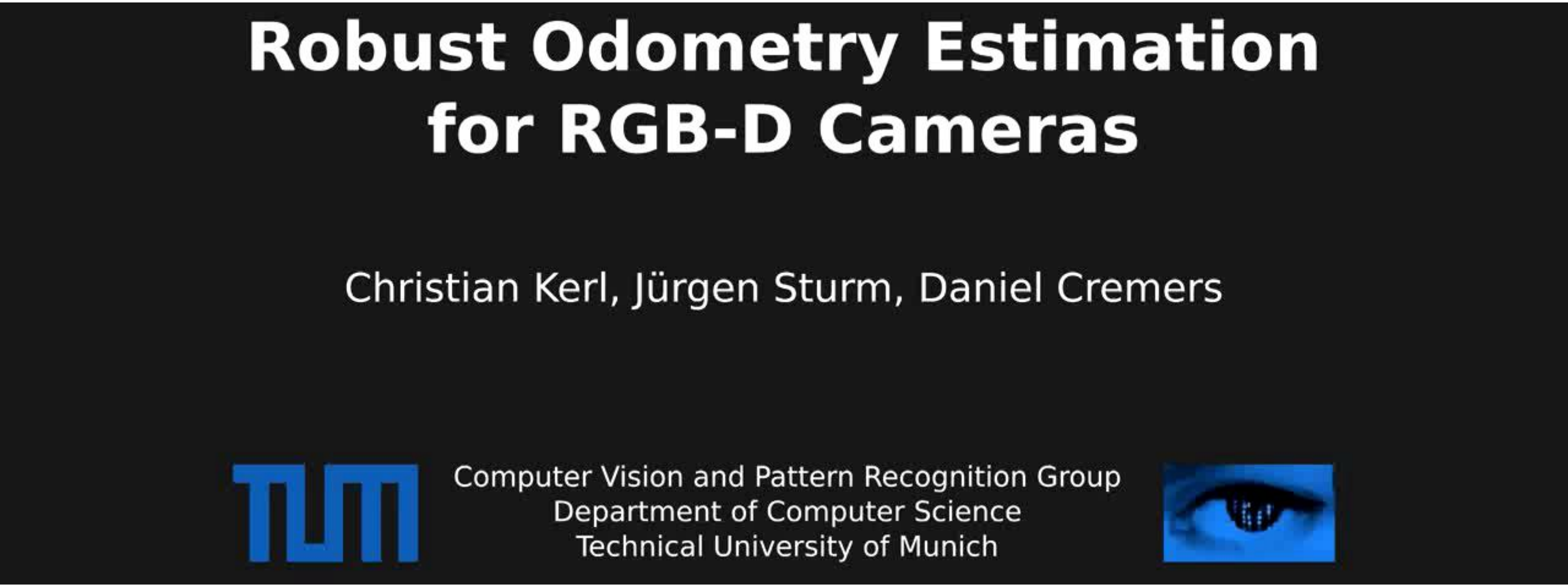
$$E(\boldsymbol{\xi}) = \int_{\mathbf{y}\in\Omega} \left| I_1(\mathbf{y}) - I_2\big(\omega(\mathbf{y},\boldsymbol{\xi})\big) \right| d\mathbf{y}$$

$$E(\boldsymbol{\xi}) = \sum_i \left| I_1(\mathbf{y}_i) - I_2\big(\omega(\mathbf{y}_i,\boldsymbol{\xi})\big) \right|$$

- Warping requires depth
  - RGB-D
  - Fixed-baseline stereo
  - Temporal stereo, tracking and (local) mapping



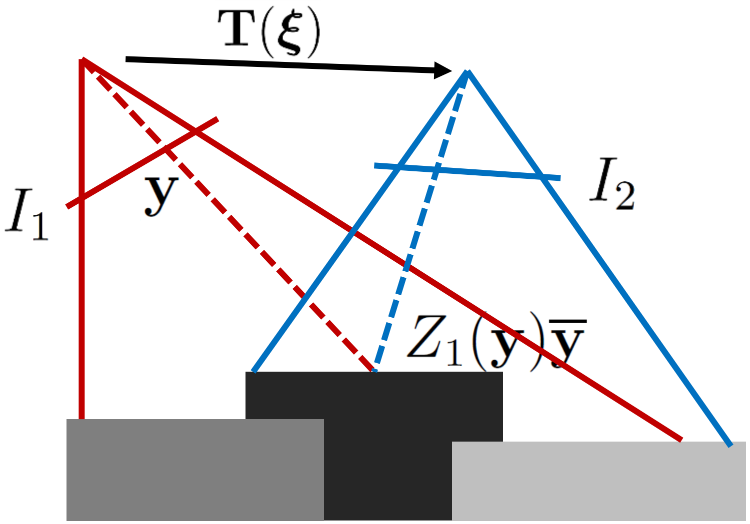
Direct Visual Odometry Example (RGB-D)



(Kerl, Sturm, Cremers, ICRA 2013)

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

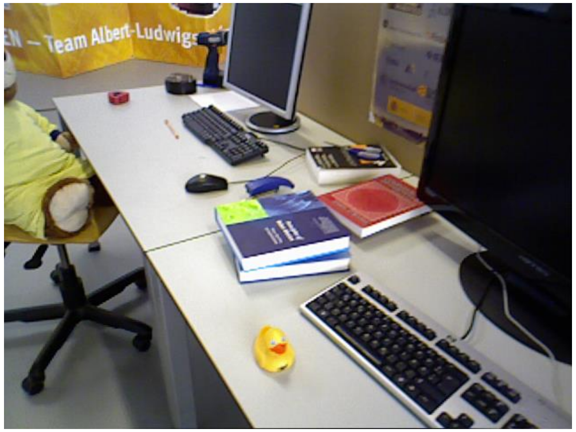
Direct Image Alignment Principle



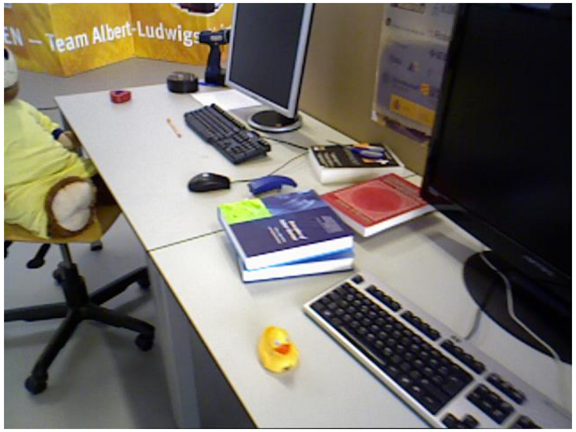
- If we know pixel depth, we can synthesize an image from a different view point
- Idealy, the intensities of the synthesized warped image are the same as from the real one

$$I_1(\mathbf{y}) = I_2\left(\pi\left(\mathbf{T}(\boldsymbol{\xi}) Z_1(\mathbf{y})\overline{\mathbf{y}}\right)\right)$$

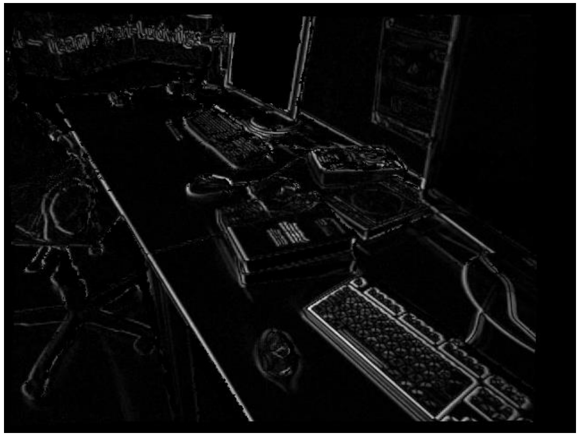
Derivative of Image Warp



I_1

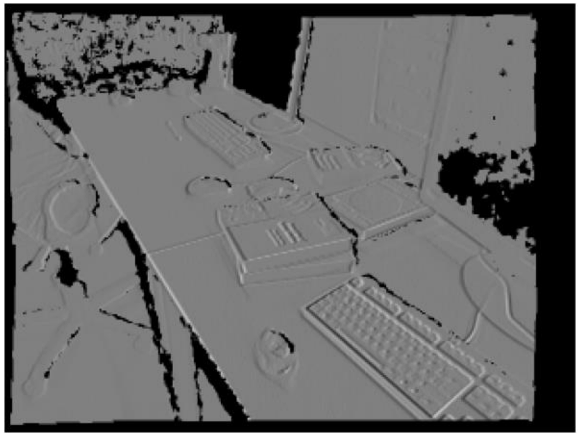


I_2



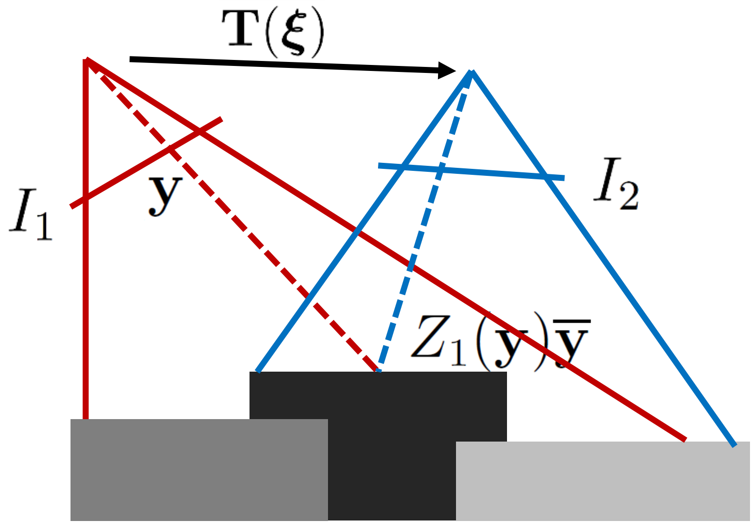
$|I_1 - I_2|$

Images from Kerl et al., ICRA 2013



$\frac{\partial I_2}{\partial v_x}$ (derivative of image intensity with respect to linear motion in x)

Direct RGB-D Image Alignment



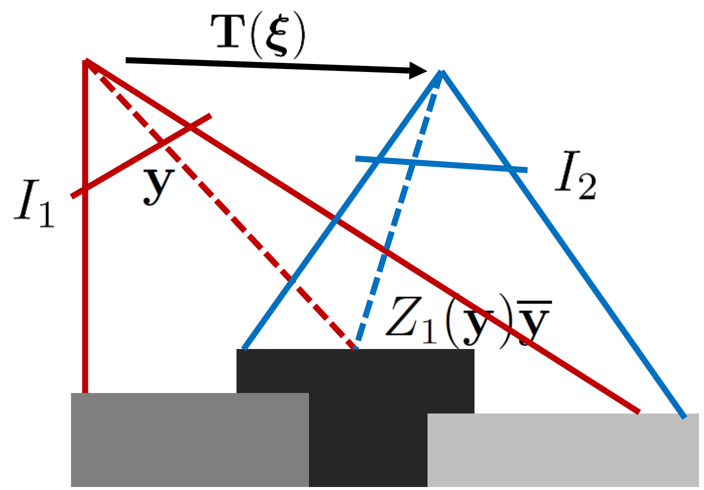
- RGB-D cameras measure depth, we only need to estimate camera motion!
- In addition to the photometric error

$$I_1(\mathbf{y}) = I_2\left(\pi\left(\mathbf{T}(\boldsymbol{\xi})Z_1(\mathbf{y})\overline{\mathbf{y}}\right)\right)$$

we can measure geometric error directly

$$\left[\mathbf{T}(\boldsymbol{\xi})Z_1(\mathbf{y})\overline{\mathbf{y}}\right]_z = Z_2\left(\pi\left(\mathbf{T}(\boldsymbol{\xi})Z_1(\mathbf{y})\overline{\mathbf{y}}\right)\right)$$

Probabilistic Direct Image Alignment



- Measurements are affected by noise

$$I_1(\mathbf{y}) = I_2\left(\pi\left(\mathbf{T}(\boldsymbol{\xi})Z_1(\mathbf{y})\overline{\mathbf{y}}\right)\right) + \epsilon$$

- A convenient assumption is Gaussian noise

$$\epsilon \sim \mathcal{N}(0, \sigma_I^2)$$

- If we further assume that noise of pixel intensities is stochastically independent accross the image, we can formulate the a-posteriori probability

$$\begin{aligned} p(\boldsymbol{\xi} \mid I_1, I_2) &\propto p(I_1 \mid \boldsymbol{\xi}, I_2) p(\boldsymbol{\xi}) \\ &\propto p(\boldsymbol{\xi}) \prod_{\mathbf{y}\in\Omega} \mathcal{N}\left(I_1(\mathbf{y}) - I_2\left(\pi\left(\mathbf{T}(\boldsymbol{\xi})Z_1(\mathbf{y})\overline{\mathbf{y}}\right)\right); 0, \sigma_I^2\right) \end{aligned}$$

Optimization Approach

- Optimize negative log-likelihood
  - Product of exponentials becomes a summation over quadratic terms
  - Normalizers are independent of the pose
  - We ignore the pose prior $p(\boldsymbol{\xi})$

$$E(\boldsymbol{\xi}) = \sum_{\mathbf{y} \in \Omega} \frac{r(\mathbf{y}, \boldsymbol{\xi})^2}{\sigma_I^2}$$

, stacked residuals: $$E(\boldsymbol{\xi}) = \mathbf{r}(\boldsymbol{\xi})^\top \mathbf{W} \mathbf{r}(\boldsymbol{\xi})$$

$$r(\mathbf{y}, \boldsymbol{\xi}) = I_1(\mathbf{y}) - I_2\left(\pi\left(\mathbf{T}(\boldsymbol{\xi}) Z_1(\mathbf{y}) \overline{\mathbf{y}}\right)\right)$$

- Non-linear least squares problem can be efficiently optimized using standard optimization tools (Gauss-Newton, Levenberg-Marquardt)

Recap: Gauss-Newton Method

- Approximate Newton's method to minimize E(x)
  - Approximate E(x) through linearization of residuals

$$\widetilde{E}(\mathbf{x}) = \frac{1}{2}\widetilde{\mathbf{r}}(\mathbf{x})^\top \mathbf{W}\widetilde{\mathbf{r}}(\mathbf{x})$$

$$= \frac{1}{2}\left(\mathbf{r}(\mathbf{x}_k) + \mathbf{J}_k\left(\mathbf{x} - \mathbf{x}_k\right)\right)^\top \mathbf{W}\left(\mathbf{r}(\mathbf{x}_k) + \mathbf{J}_k\left(\mathbf{x} - \mathbf{x}_k\right)\right) \qquad \mathbf{J}_k := \nabla_\mathbf{x}\mathbf{r}(\mathbf{x})|_{\mathbf{x}=\mathbf{x}_k}$$

$$= \frac{1}{2}\mathbf{r}(\mathbf{x}_k)^\top \mathbf{W}\mathbf{r}(\mathbf{x}_k) + \underbrace{\mathbf{r}(\mathbf{x}_k)^\top \mathbf{W}\mathbf{J}_k}_{=:\mathbf{b}_k^\top}\left(\mathbf{x} - \mathbf{x}_k\right) + \frac{1}{2}\left(\mathbf{x} - \mathbf{x}_k\right)^\top \underbrace{\mathbf{J}_k^\top \mathbf{W}\mathbf{J}_k}_{=:\mathbf{H}_k}\left(\mathbf{x} - \mathbf{x}_k\right)$$

  - Find root of $\nabla_\mathbf{x}\widetilde{E}(\mathbf{x}) = \mathbf{b}_k^\top + \left(\mathbf{x} - \mathbf{x}_k\right)^\top \mathbf{H}_k$ using Newton's method, i.e.

$$\nabla_\mathbf{x}\widetilde{E}(\mathbf{x}) = \mathbf{0} \text{ iff } \mathbf{x} = \mathbf{x}_k - \mathbf{H}_k^{-1}\mathbf{b}_k$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \mathbf{H}_k^{-1}\mathbf{b}_k$$

Recap: Levenberg-Marquardt Method

- Gradually transition between gradient descent and Gauss-Newton
  - Augment Hessian approximation of Gauss-Newton (damping)

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \left(\mathbf{H}_k + \lambda \mathbf{I}\right)^{-1} \mathbf{b}_k$$

  - Adaptive weighting: $\mathbf{x}_{k+1} = \mathbf{x}_k - \left(\mathbf{H}_k + \lambda \, \mathrm{diag}(\mathbf{H}_k)\right)^{-1} \mathbf{b}_k$

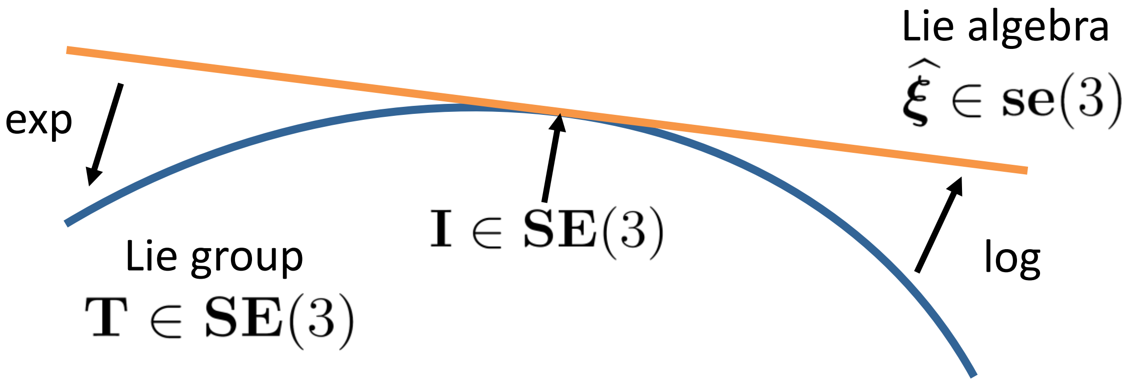
  - Start with $\lambda = 0.1$
  - Accept step and decrease lambda $\lambda \leftarrow \lambda / 2$ if error function decreases, otherwise discard step and increase lambda $\lambda \leftarrow 2\lambda$ (akin line search)

Pose Parametrization for Optimization

- Requirements on pose parametrization
  - No singularities
  - Minimal to avoid constraints

- Various pose parametrizations available
  - Direct matrix representation => not minimal
  - Quaternion / translation => not minimal
  - Euler angles / translation => singularities (gimbal lock)
  - Twist coordinates of elements in Lie Algebra se(3) of SE(3) (axis-angle / translation)

Recap: Representing Motion using Lie Algebra se(3)



$$\boldsymbol{\xi} := \begin{pmatrix} \boldsymbol{\omega} \\ \mathbf{v} \end{pmatrix} \in \mathbb{R}^6$$

$$\boldsymbol{\omega} \in \mathbb{R}^3$$

$$\mathbf{v} \in \mathbb{R}^3$$

$$\widehat{\boldsymbol{\xi}} := \begin{pmatrix} \widehat{\boldsymbol{\omega}} & \mathbf{v} \\ \mathbf{0} & 0 \end{pmatrix} \in \mathbb{R}^{4\times 4}$$

- $\mathbf{SE}(3)$ is a smooth manifold, i.e. a Lie group
- Its Lie algebra $\mathbf{se}(3)$ provides an elegant way to parametrize poses for optimization
- Its elements $\widehat{\boldsymbol{\xi}} \in \mathbf{se}(3)$ form the tangent space of $\mathbf{SE}(3)$ at identity
- The $\mathbf{se}(3)$ elements can be interpreted as rotational and translational velocities (twists)

Optimization with Twist Coordinates

- Twists provide a minimal local representation without singularities
- We can decompose transformations in each optimization step into the transformation itself and an infinitesimal increment

$$\mathbf{T}(\boldsymbol{\xi}) = \exp(\widehat{\boldsymbol{\delta\xi}})\,\mathbf{T}(\boldsymbol{\xi}) = \mathbf{T}(\boldsymbol{\delta\xi} \oplus \boldsymbol{\xi})$$

But!

$$\mathbf{T}(\boldsymbol{\delta\xi} + \boldsymbol{\xi}) \neq \mathbf{T}(\boldsymbol{\delta\xi})\mathbf{T}(\boldsymbol{\xi})$$

- We perform optimization with respect to auxiliary variable $\delta\boldsymbol{\xi}$
  - Example: Gradient descent on the auxiliary variable

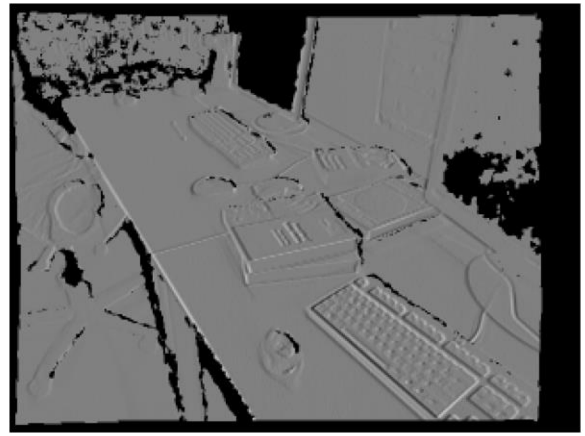
$$\boldsymbol{\delta\xi}^* = -\eta \nabla_{\boldsymbol{\delta\xi}} E(\boldsymbol{\xi}_k, \boldsymbol{\delta\xi})$$

$$\mathbf{T}(\boldsymbol{\xi}_{k+1}) = \exp\left(\widehat{\boldsymbol{\delta\xi}^*}\right)\mathbf{T}(\boldsymbol{\xi}_k)$$

- Similar for Gauss-Newton: calculate Jacobian of $\mathbf{r}(\boldsymbol{\xi}_k, \boldsymbol{\delta\xi})$ with respect to $\boldsymbol{\delta\xi}$
- Make sure the increment is applied from the correct side

Properties of Residual Linearization

$|I_1 - I_2|$



$\frac{\partial I_2}{\partial v_x}$

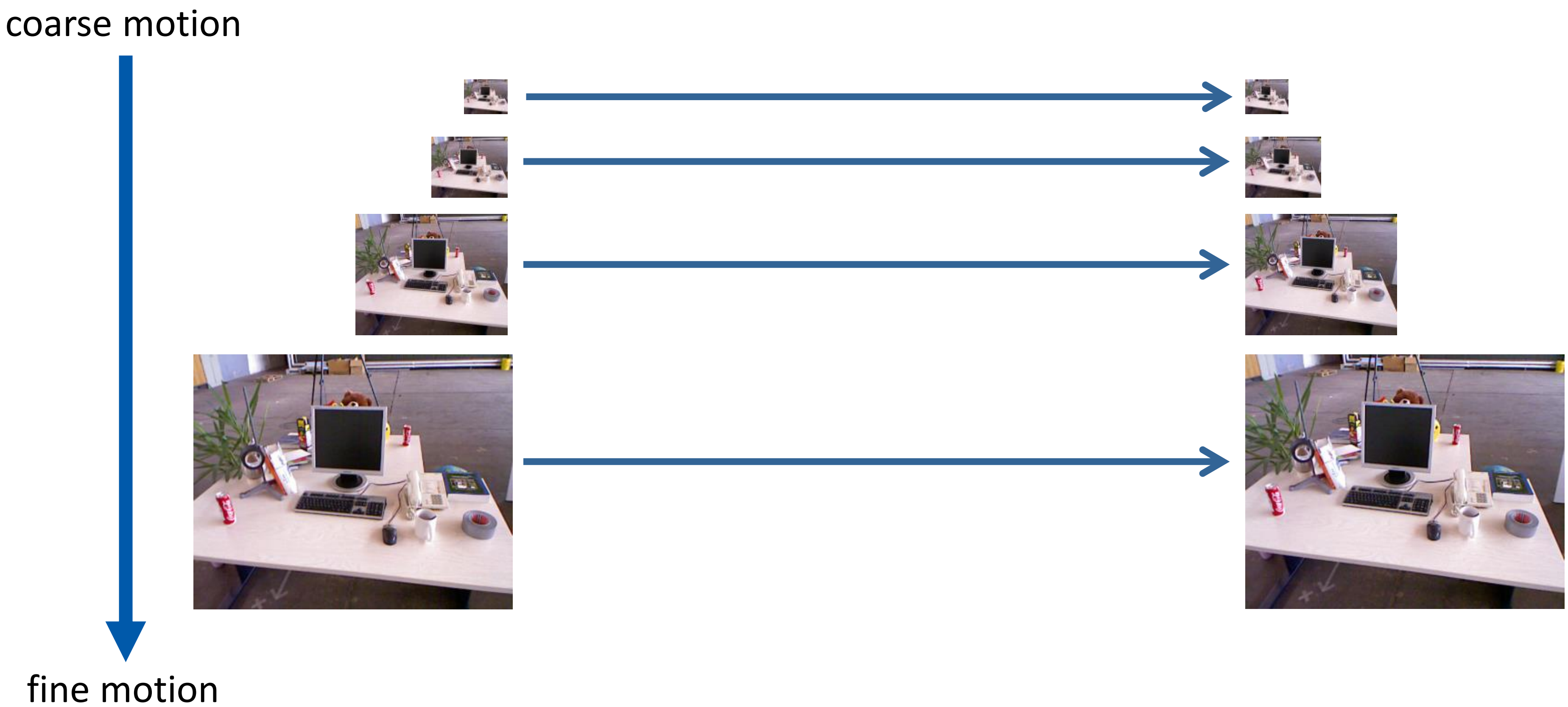
$$r(\mathbf{y}, \boldsymbol{\xi}) = I_1(y) - I_2\big(\omega(\mathbf{y}, \boldsymbol{\xi})\big) \qquad \text{with} \qquad \omega(\mathbf{y}, \boldsymbol{\xi}) \coloneqq \pi(\mathbf{T}(\boldsymbol{\xi}) Z_1(\mathbf{y}) \bar{\mathbf{y}})$$

- Gradient of residuals w.r.t. pose

$$\nabla_{\xi} r(\mathbf{y}, \boldsymbol{\xi}) = -\nabla_{\omega} I_2\big(\omega(\mathbf{y}, \boldsymbol{\xi})\big)\, \nabla_{\xi} \omega(\mathbf{y}, \boldsymbol{\xi})$$

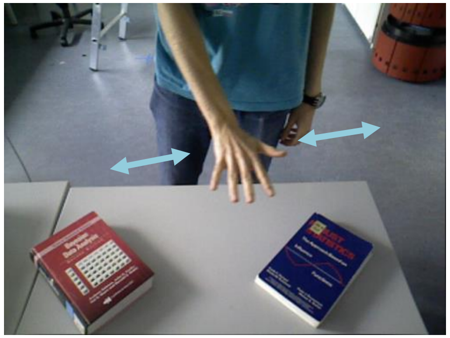
- Linearization is only valid for motions that change the projection in a small image neighborhood that is captured by the local gradient
- $E(\boldsymbol{\xi})$ is far from being a convex function (many local minima)

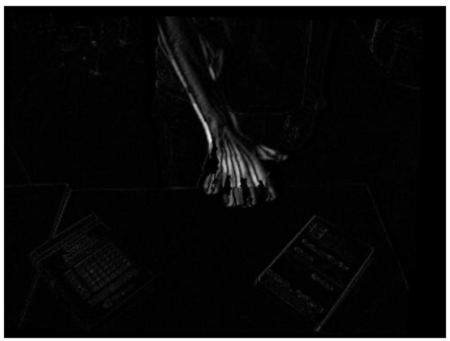
Coarse-To-Fine Optimization

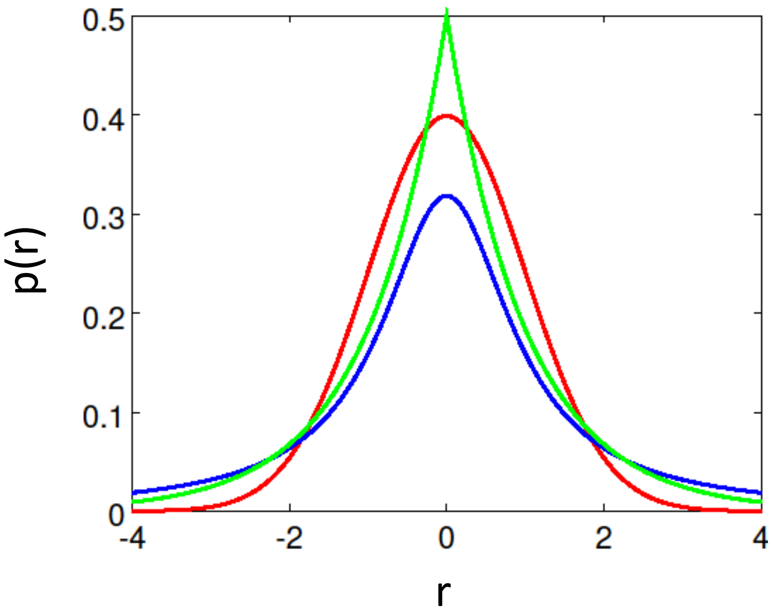


- Important: smooth image during downscaling
  - E.g. average over four neighboring pixels

Residual Distributions





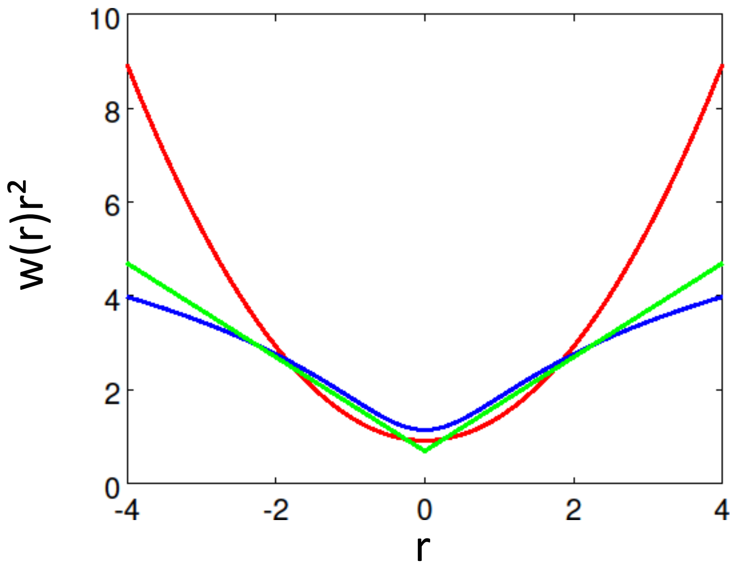


- Normal distribution
- Laplace distribution
- Student-t distribution

- Gaussian noise assumption on photometric residuals oversimplifies
- Outliers (occlusions, motion, etc.):
  - Residuals are distributed with more mass on the larger values

Images from Kerl et al., ICRA 2013

Optimizing Non-Gaussian Measurement Noise



- Normal distribution
- Laplace distribution
- Student-t distribution

- Can we change the residual distribution in least squares optimization?
- For specific types of distributions: yes!
- Iteratively reweighted least squares: Reweight residuals in each iteration

$$E(\boldsymbol{\xi}) = \sum_{\mathbf{y}\in\Omega} w\big(r(\mathbf{y},\boldsymbol{\xi})\big)\frac{r(\mathbf{y},\boldsymbol{\xi})^2}{\sigma_I^2}$$

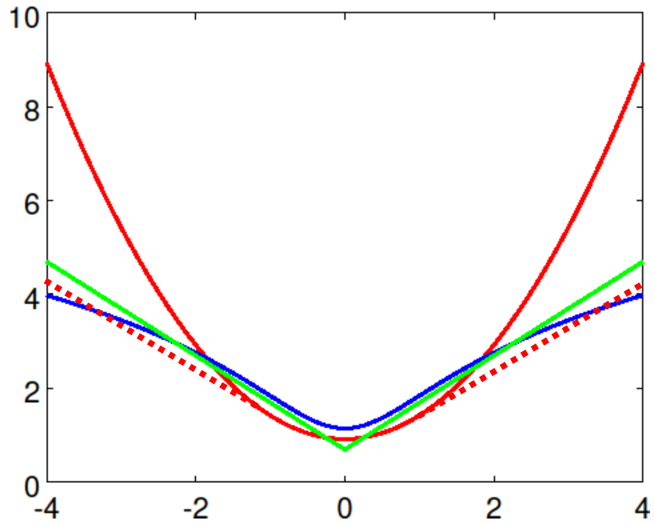
Laplace distribution:

$$w\big(r(\mathbf{y},\boldsymbol{\xi})\big) = |r(\mathbf{y},\boldsymbol{\xi})|^{-1}$$

Huber Loss

- Huber-loss „switches“ between Gaussian (locally at mean) and Laplace distribution

$$\|r\|_\delta = \begin{cases} 0.5r^2 & \text{for } |r| \leq \delta \\ \delta(|r| - 0.5\delta) & \text{otherwise} \end{cases}$$



- Normal distribution
- Laplace distribution
- Student-t distribution

Huber-loss for δ = 1

Efficient Non-Linear Least Squares

- Gauss-Newton / Levenberg-Marquardt can be applied very efficiently to direct image alignment:
  - $\mathbf{H}_k$ is only a 6x6 matrix
  - $\mathbf{b}_k = \mathbf{J}_k^{\mathrm{T}} \mathbf{W} \mathbf{r}(\xi_i)$ is a 6x1 vector
  - Since we treat each pixel stochastically independent from neighboring pixels, $\mathbf{H}_k$ and $\mathbf{b}_k$ are summed over individual pixels

$$\mathbf{H}_k = \sum_{\mathbf{y} \in \Omega} \frac{w(\mathbf{y}, \xi_k)}{\sigma_I^2} \mathbf{J}_{k,\mathbf{y}}^{\mathrm{T}} \mathbf{J}_{k,\mathbf{y}} \qquad \mathbf{b}_k = \sum_{\mathbf{y} \in \Omega} \frac{w(\mathbf{y}, \xi_k)}{\sigma_I^2} \mathbf{J}_{k,\mathbf{y}}^{\mathrm{T}} r(\mathbf{y}, \xi_k)$$

$$\mathbf{J}_{k,\mathbf{y}} \coloneqq \nabla_{\delta\xi} r(\mathbf{y}, \delta\xi \oplus \xi_k)$$

Algorithm: Direct RGB-D Visual Odometry

Input: RGB-D image sequence $I_{0:t}, Z_{0:t}$

Output: aggregated camera poses $\mathbf{T}_{0:t}$

Algorithm:

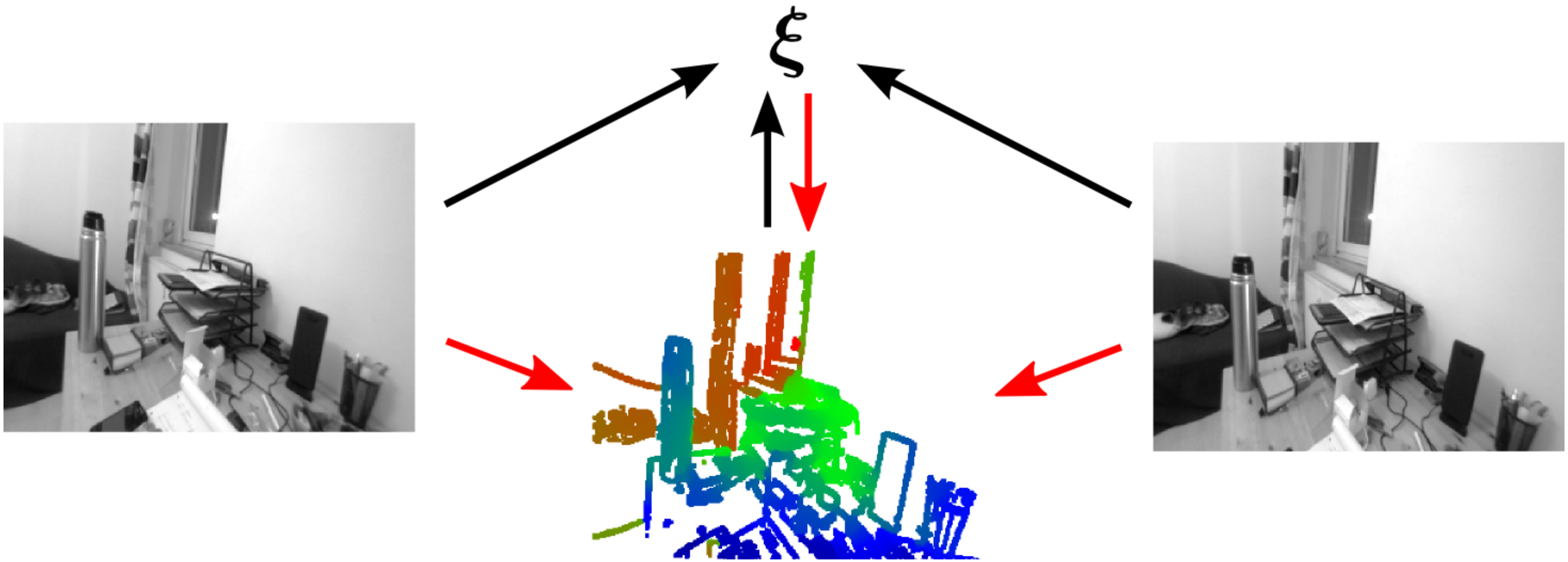
For each current RGB-D image I_k, Z_k :

1. Estimate relative camera motion $\mathbf{T}_k^{k-1}$ towards the previous RGB-D frame using direct image alignment
2. Concatenate estimated camera motion with previous frame camera pose to obtain current camera pose estimate

$$\mathbf{T}_k = \mathbf{T}_{k-1}\mathbf{T}_k^{k-1}$$

Monocular Direct Visual Odometry

- Estimate motion and depth concurrently



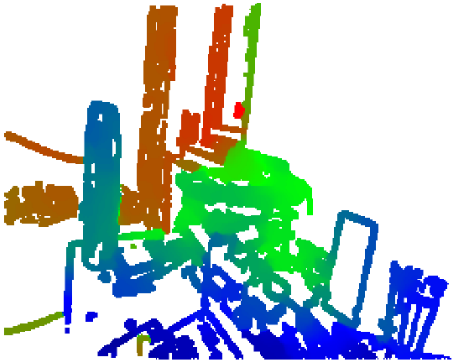
- Alternating optimization: **Tracking** and **Mapping**

Images from: Engel et al., ICCV 2013

Semi-Dense Mapping

- Estimate inverse depth and variance at high gradient pixels
- Correspondence search along epipolar line (5-pixel intensity SSD)



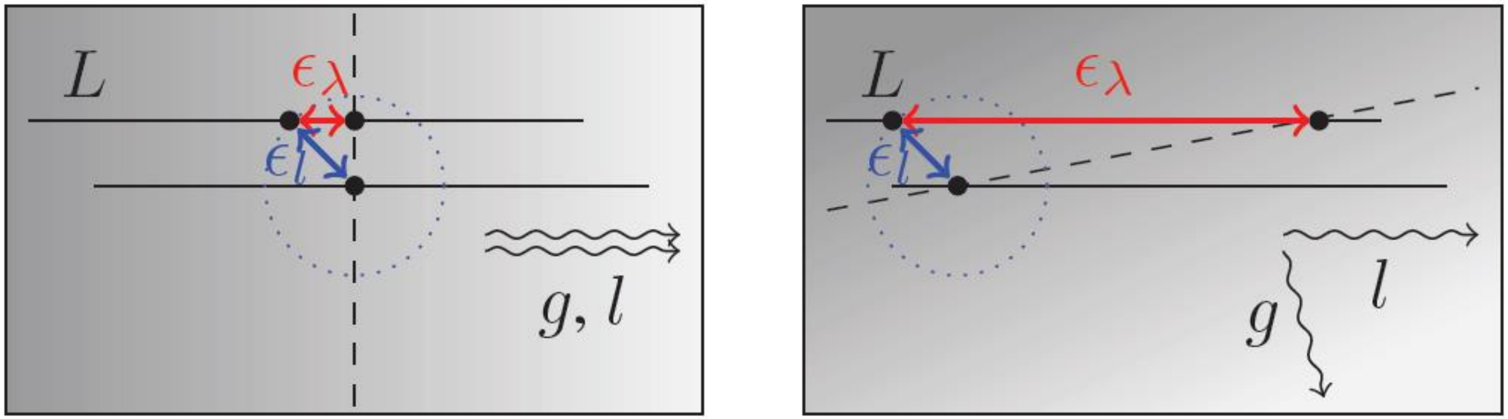


- Kalman-filtering of depth map:
  - Propagate depth map & variance from previous frame
  - Update depth map & variance with new depth observations

Images from: Engel et al., ICCV 2013

Semi-Dense Mapping

- Estimate for inverse depth uncertainty from geometric and intensity noise
  - Very simplified model, but works quite well in reality



Geometric noise

$$\sigma^2_{\lambda(\xi,\pi)} = \frac{\sigma_l^2}{\langle g, l \rangle^2}$$

pos. variance of epipolar line

gradient direction

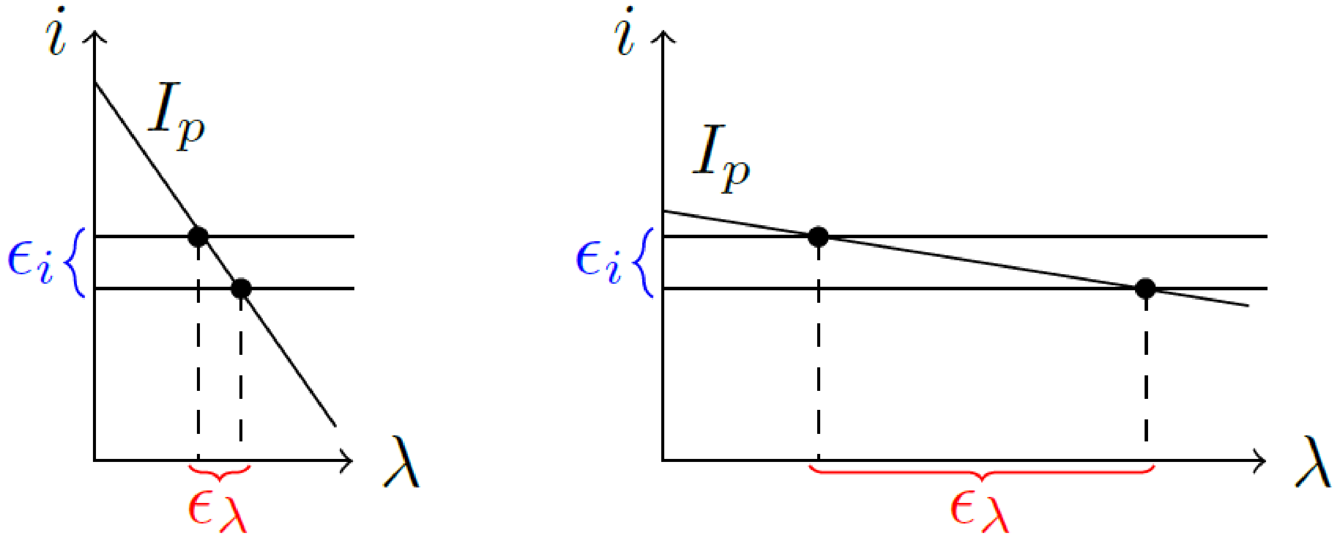
epipolar line direction

λ is the estimated disparity approx. proportionaly to inverse depth

Images from: Engel et al., ICCV 2013

Semi-Dense Mapping

- Estimate for inverse depth uncertainty from geometric and intensity noise



Intensity noise

$$\sigma^2_{\lambda(I)} = \frac{2\sigma_i^2}{g_p^2}$$

intensity noise variance

image gradient magnitude at epipolar line

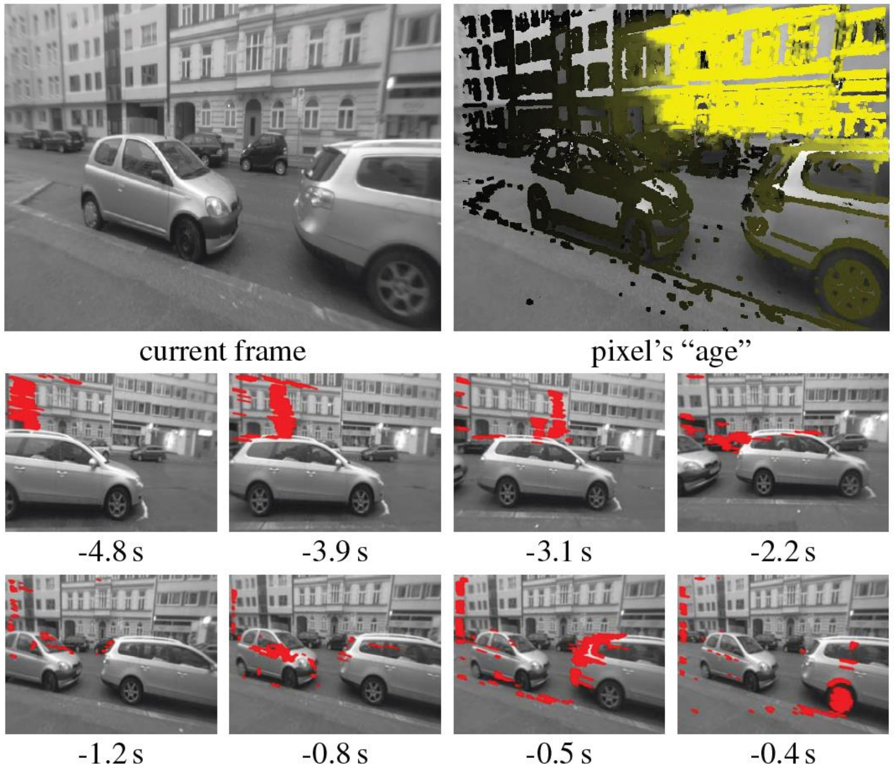
Paper:
https://openaccess.thecvf.com/content_iccv_2013/papers/Engel_Semi-dense_Visual_Odometry_2013_ICCV_paper.pdf

Images from: Engel et al., ICCV 2013

Choosing the Stereo Reference Frame

- Naive: use one specific reference frame (f.e. the previous frame or a keyframe)

- We can also select the reference frame for stereo comparisons for each pixel individually in order to achieve a trade-off between accuracy and computation time



Heuristics from Engel et al., ICCV 2013:
Use oldest frame in which pixel still visible but disparity search range and observation angle below threshold

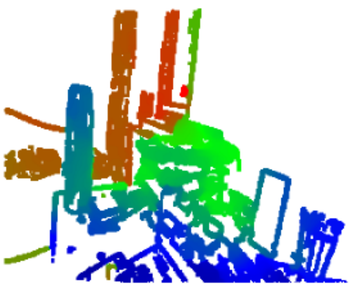
Images from: Engel et al., ICCV 2013

Semi-Dense Direct Image Alignment

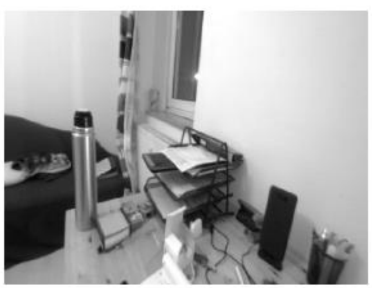
$$E(\boldsymbol{\xi}) = \sum_{\mathbf{y} \in \Omega^Z} w\left(r(\mathbf{y}, \boldsymbol{\xi})\right) \frac{r(\mathbf{y}, \boldsymbol{\xi})^2}{\sigma^2_{Z(\mathbf{y})}}$$

$$r(\mathbf{y}, \boldsymbol{\xi}) = I_1\left(\mathbf{y}\right) - I_2\left(\pi\left(\mathbf{T}(\boldsymbol{\xi}) Z_1(\mathbf{y}) \overline{\mathbf{y}}\right)\right)$$

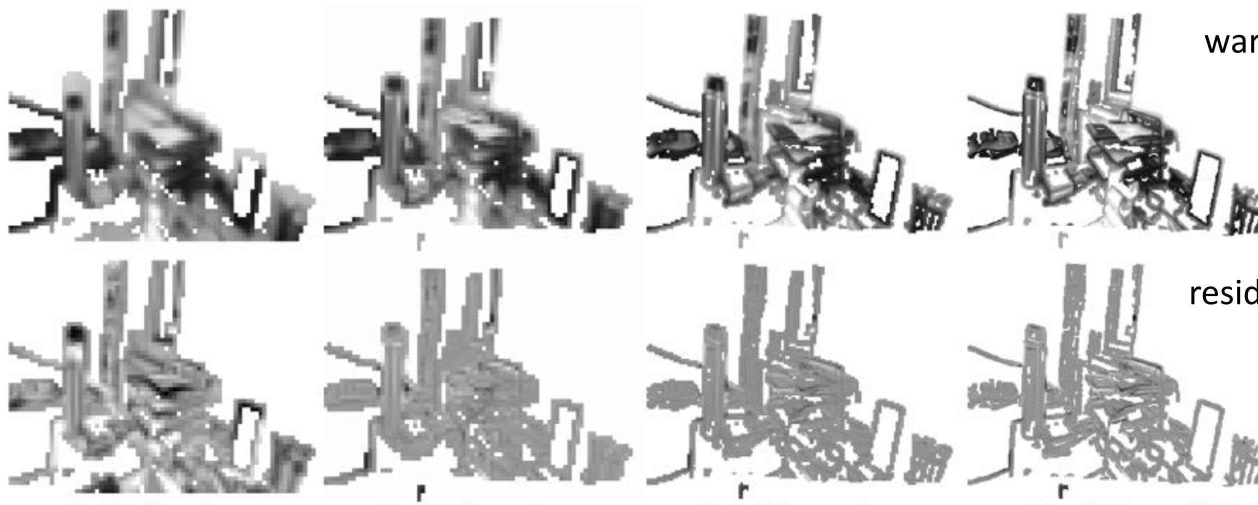
I_1



Z_1



I_2



warped I_2

residuals

initialization on lvl 3 (80 × 60)	after 8 iterations on lvl 3 (80 × 60)	after 3 iterations on lvl 2 (160 × 120)	after 3 iterations on lvl 1 (320 × 240)

Images from: Engel et al., ICCV 2013

Algorithm: Direct Monocular Visual Odometry

Input: Monocular image sequence $I_{0:t}$
Output: aggregated camera poses $\mathbf{T}_{0:t}$

Algorithm:
Initialize depth map Z_0

- E.g. from first two frames with a point-based method

For each current image I_k:

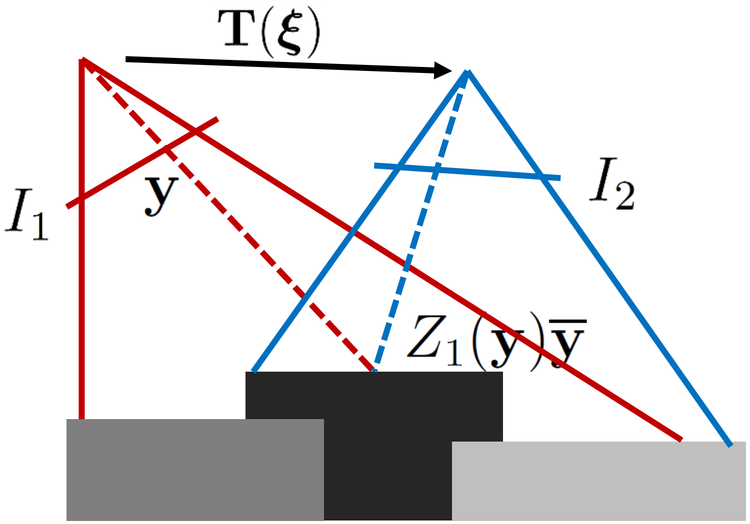
1. Estimate relative camera motion $\mathbf{T}_k^{k-1}$ towards the previous image with estimated semi-dense depth map Z_{k-1} using direct image alignment
2. Concatenate estimated camera motion with previous frame camera pose to obtain current camera pose estimate $\mathbf{T}_k = \mathbf{T}_{k-1}\mathbf{T}_k^{k-1}$
3. Propagate semi-dense depth map Z_{k-1} from previous frame to current frame to obtain $\widetilde{Z}_k$
4. Update propagated semi-dense depth map $\widetilde{Z}_k$ with temporal stereo depth measurements to obtain Z_k

Direct Visual Odometry Example (Mono)

(Engel, Sturm, Cremers, ICCV 2013)

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

Direct Image Alignment Revisited



- If we know pixel depth, we can „simulate“ an image from a different view point
- Ideally, the warped image is the same as the image taken from that pose:

$$I_1(\mathbf{y}) = I_2\left(\pi\left(\mathbf{T}(\boldsymbol{\xi})Z_1(\mathbf{y})\overline{\mathbf{y}}\right)\right)$$

- What do we mean with „ideally“ ?

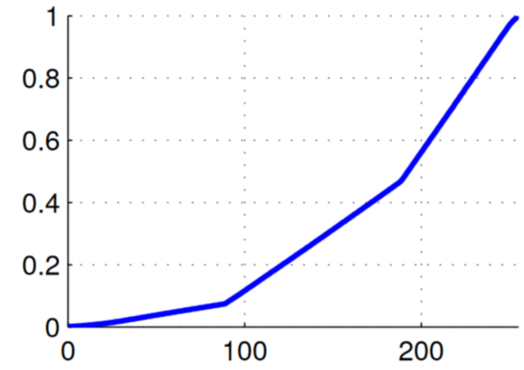
Recap: Camera Response Function

- The objects in the scene radiate light which is focused by the lens onto the image sensor

- The pixels of the sensor observe an irradiance $B : \Omega \rightarrow \mathbb{R}$ for an exposure time t

- The camera electronics translates the accumulated irradiance into intensity values according to a non-linear camera response function $G : \mathbb{R} \rightarrow [0, 255]$

example inv. G



- The measured intensity is $I(\mathbf{x}) = G(tB(\mathbf{x}))$

Recap: Vignetting

uncorrected

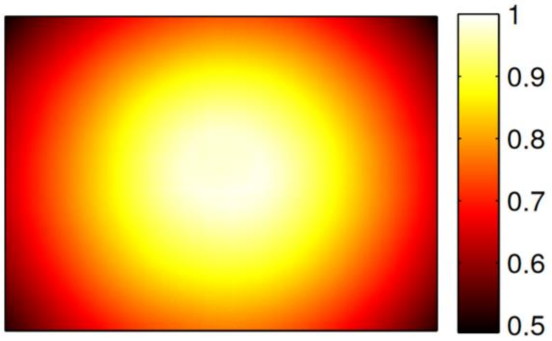
corrected

- Lenses gradually focus more light at the center of the image than at the image borders
- The image appears darker towards the borders
- Also called "lens attenuation"
- Lens vignetting can be modelled as a map $V : \Omega \rightarrow [0, 1]$

- Intensity measurement model

$$I(\mathbf{x}) = G(tV(\mathbf{x})B(\mathbf{x}))$$

$V(\mathbf{x})$

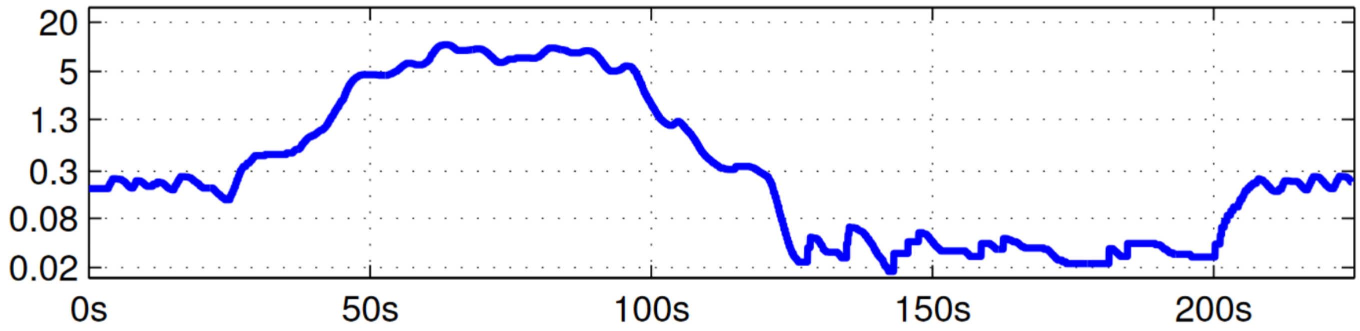


Brightness Constancy Assumption Revisited

- Camera images include vignetting effects and non-linear camera response function
- Idea: invert vignetting and camera response function using a known calibration
- Perform direct image alignment on irradiance images:

$$I'(\mathbf{y}) = tB(\mathbf{y}) = \frac{G^{-1}(I(\mathbf{y}))}{V(\mathbf{y})}$$

Brightness Constancy Assumption Revisited

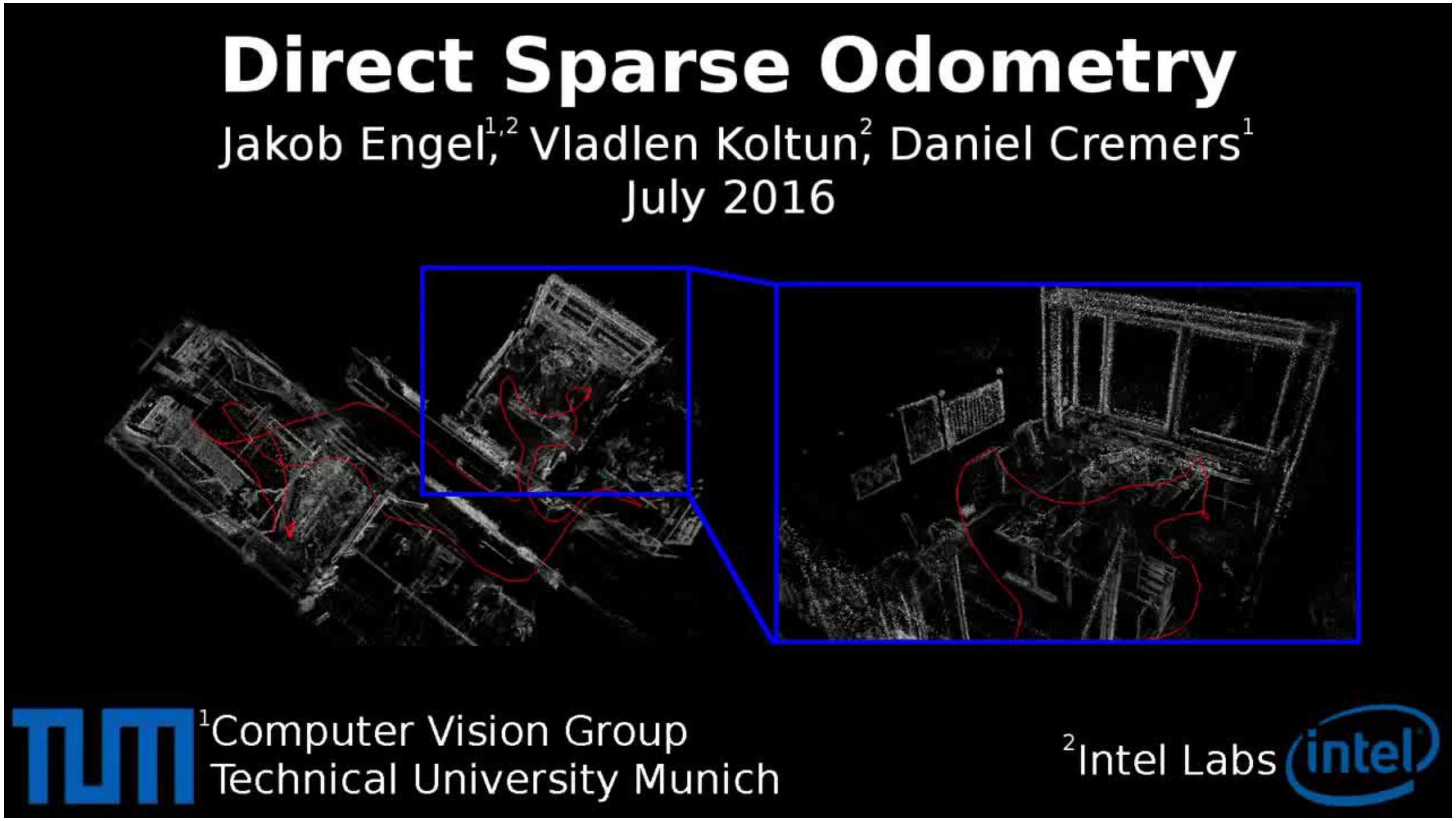


- Automatic exposure adjustment needed in realistic environments
- Add exposure parameters explicitly to objective function:

$$\left(I_2(\omega(\mathbf{y},\boldsymbol{\xi},Z_1(\mathbf{y}))) - b_2\right) - \frac{t_2 \exp(a_2)}{t_1 \exp(a_1)}\left(I_1(\mathbf{y}) - b_1\right)$$

Image: Engel et al. PAMI 2018

Direct Sparse Visual Odometry (Mono)



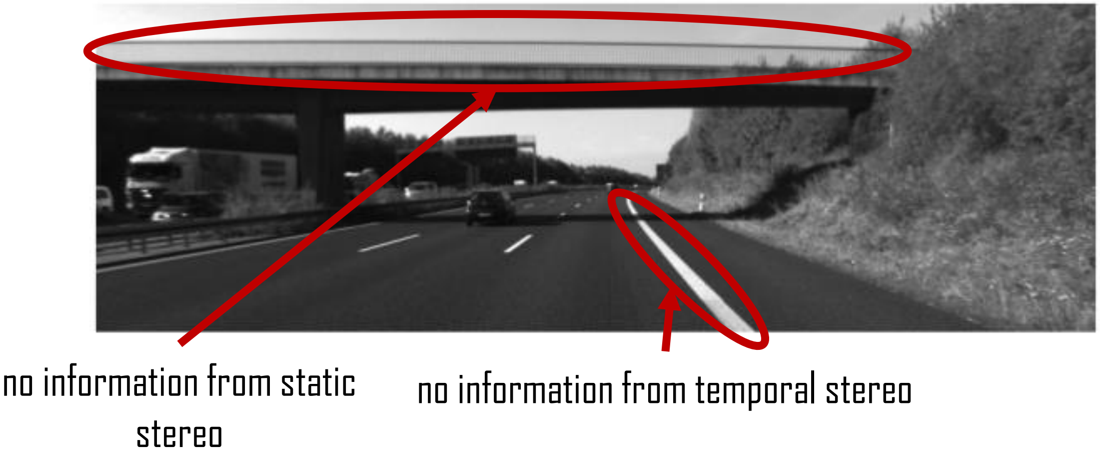
(Engel, Koltun, Cremers, T-PAMI 2018)

How does the robot move?

https://www.youtube.com/watch?v=C6-xwSOOdqQ

Direct Mapping with Stereo Cameras

- For stereo cameras, we can exploit the known camera extrinsics to estimate depth from static stereo (left-right images) in addition to temporal stereo (successive left or right images)



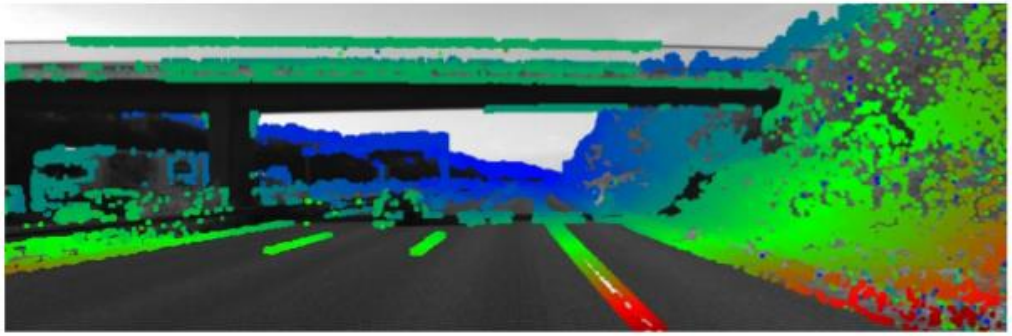
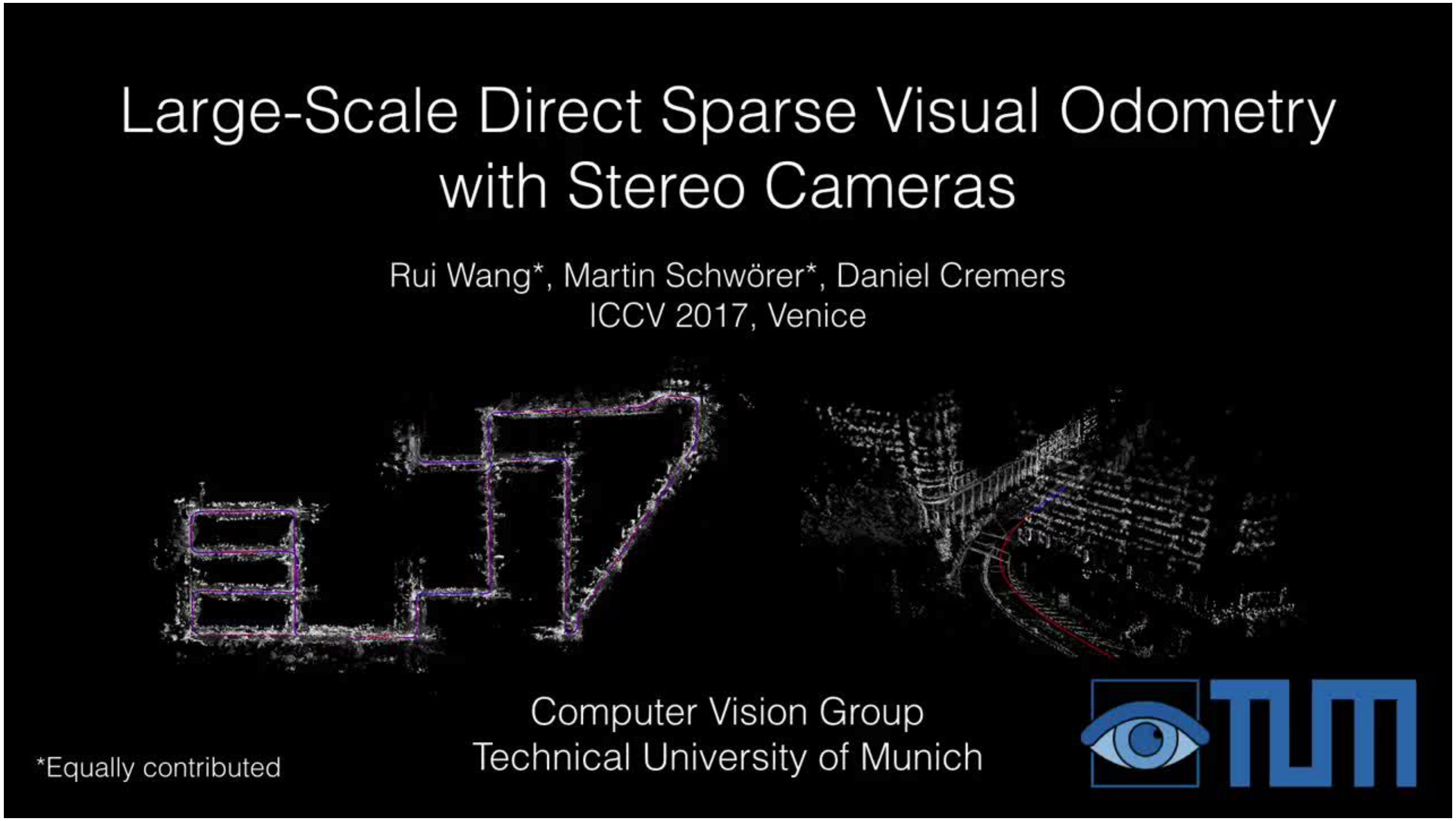


Image from: Engel et al. IROS 2015

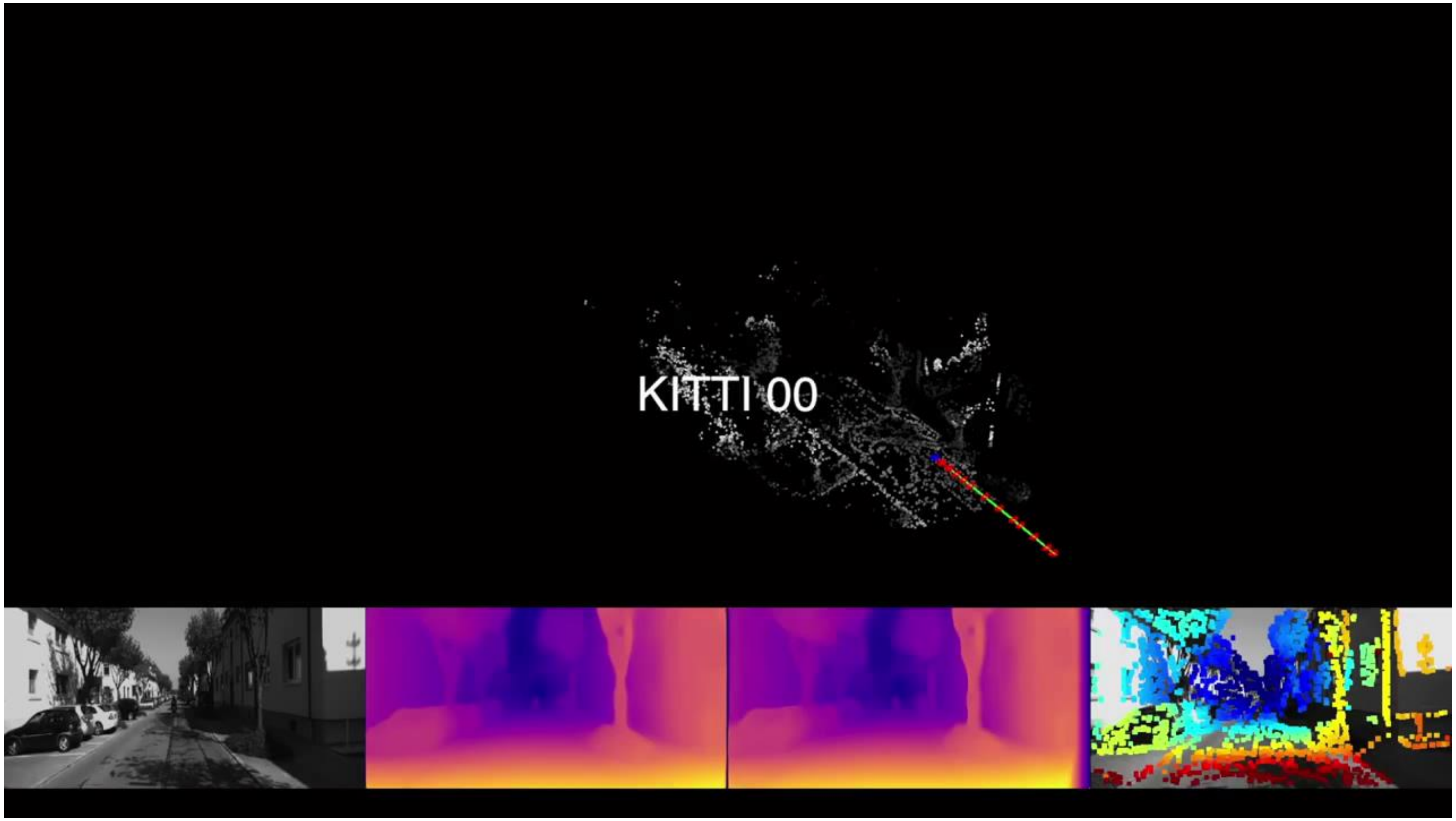
Direct Sparse Visual Odometry (Stereo)



(Wang, Schwörer, Cremers, ICCV 2017)

https://www.youtube.com/watch?v=A53vJO8eygw

Deep Direct Sparse VO (Mono)



(Yang, Wang, Stückler, Cremers, ECCV 2018)

https://www.youtube.com/watch?v=sLZOeC9z_tw&t=7s

Lessons Learned Today

- Direct image alignment avoids manually designed keypoints and can use all available image information
- Direct visual odometry
  - Dense RGB-D odometry by direct image alignment with measured depth
  - Direct image alignment for monocular cameras requires depth estimation from temporal stereo
  - Stereo cameras: Direct depth estimation using static and temporal stereo
- Direct image alignment as non-linear least squares problem
  - Linearization of the residuals requires a coarse-to-fine optimization scheme
  - SE(3) Lie algebra provides an elegant way of motion representation for gradient-based optimization
  - Iteratively reweighted least squares allows for wider set of residual distributions than Gaussians
- Photometric calibration and exposure parameter estimation

Thanks for your attention!

Slides Information

- These slides have been initially created by Jörg Stückler as part of the lecture “Robotic 3D Vision” in winter term 2017/18 at Technical University of Munich.
- The slides have been revised by myself (Niclas Zeller) for the same lecture held in winter term 2020/21
- Acknowledgement of all people that contributed images or video material has been tried (please kindly inform me if such an acknowledgement is missing so it can be added).