Computer Vision 2
WS 2018/19

Part 14 – Visual Odometry III
19.12.2018

Prof. Dr. Bastian Leibe

RWTH Aachen University, Computer Vision Group
http://www.vision.rwth-aachen.de

Course Outline

- Single-Object Tracking
- Bayesian Filtering
- Multi-Object Tracking
  - Introduction
  - MHT, (JPDAF)
  - Network Flow Optimization
- Visual Odometry
  - Sparse interest-point based methods
  - Dense direct methods
- Visual SLAM & 3D Reconstruction
- Deep Learning for Video Analysis

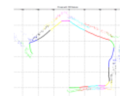


image source: [Zhang, Li, Nevatia, CVPR'08]

Topics of This Lecture

- Recap: Point-based Visual Odometry
  - Further Considerations
- Direct Methods
  - Direct image alignment
  - Pose parametrization
  - Lie group se(3) and the exponential map
  - Residual linearization
  - Optimization considerations

Recap: Direct vs. Indirect Methods

- Direct methods
  - formulate alignment objective in terms of photometric error (e.g., intensities)

$$p(\mathbf{I}_2 \mid \mathbf{I}_1, \boldsymbol{\xi}) \quad \longrightarrow \quad E(\boldsymbol{\xi}) = \int_{\mathbf{u}\in\Omega} |\mathbf{I}_1(\mathbf{u}) - \mathbf{I}_2(\omega(\mathbf{u}, \boldsymbol{\xi}))| \, d\mathbf{u}$$

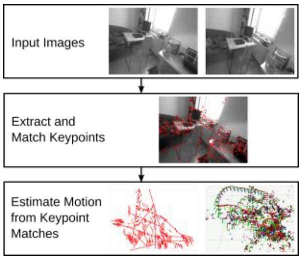
- Indirect methods
  - formulate alignment objective in terms of reprojection error of geometric primitives (e.g., points, lines)

$$p(\mathbf{Y}_2 \mid \mathbf{Y}_1, \boldsymbol{\xi}) \quad \longrightarrow \quad E(\boldsymbol{\xi}) = \sum_i |\mathbf{y}_{1,i} - \omega(\mathbf{y}_{2,i}, \boldsymbol{\xi})|$$

Slide credit: Jörg Stückler

Recap: Point-based Visual Odometry Pipeline

- Keypoint detection and local description (CV I)
- Robust keypoint matching (CV I)
- Motion estimation
  - 2D-to-2D: motion from 2D point correspondences
  - 2D-to-3D: motion from 2D points to local 3D map
  - 3D-to-3D: motion from 3D point correspondences (e.g., stereo, RGB-D)



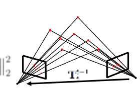
Slide credit: Jörg Stückler
Images from Jakob Engel

Recap: Motion Estimation from Point Correspondences

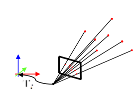
- 2D-to-2D
  - Reproj. error:

$$E\left(\mathbf{T}_t^{t-1}, X\right) = \sum_{i=1}^{N} \|\bar{\mathbf{y}}_{t,i} - \pi(\bar{\mathbf{x}}_i)\|_2^2 + \|\bar{\mathbf{y}}_{t-1,i} - \pi\left(\mathbf{T}_t^{t-1}\bar{\mathbf{x}}_i\right)\|_2^2$$

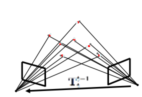
  - Introduced linear algorithm: 8-point



- 2D-to-3D
  - Reprojection error: $E(\mathbf{T}_t) = \sum_{i=1}^{N} \|\mathbf{y}_{t,i} - \pi(\mathbf{T}_t\bar{\mathbf{x}}_i)\|_2^2$
  - Introduced linear algorithm: DLT PnP



- 3D-to-3D
  - Reprojection error: $E\left(\mathbf{T}_t^{t-1}\right) = \sum_{i=1}^{N} \|\overline{\mathbf{x}}_{t-1,i} - \mathbf{T}_t^{t-1}\overline{\mathbf{x}}_{t,i}\|_2^2$
  - Introduced linear algorithm: Arun's method



Slide credit: Jörg Stückler

Recap: Keypoint Detectors

- Corners
  - Image locations with locally prominent intensity variation
  - Intersections of edges
- Examples: Harris, FAST
- Scale-selection: Harris-Laplace

- Blobs
  - Image regions that stick out from their surrounding in intensity/texture
  - Circular high-contrast regions
- E.g.: LoG, DoG (SIFT), SURF
- Scale-space extrema in LoG/DoG


Harris Corners


DoG (SIFT) Blobs

Recap: RANSAC

- RANdom SAmple Consensus algorithm for robust estimation

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

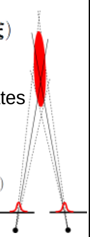
Probabilistic Modelling

- Model image point observation likelihood $p(\mathbf{y}_i \mid \mathbf{x}_i, \boldsymbol{\xi})$
  - E.g., Gaussian: $p(\mathbf{y}_i \mid \mathbf{x}_i, \boldsymbol{\xi}) \sim \mathcal{N}\left(\mathbf{y}_i; \pi\left(\mathbf{T}(\boldsymbol{\xi})\mathbf{x}_i\right), \boldsymbol{\Sigma}_{\mathbf{y}_i}\right)$
- Optimize maximum a-posteriori likelihood of estimates

$$p(X, \boldsymbol{\xi} \mid Y) \propto p(Y \mid X, \boldsymbol{\xi})\, p(X, \boldsymbol{\xi}) = p(X, \boldsymbol{\xi}) \prod_{i=1}^{N} p(\mathbf{y}_i \mid \mathbf{x}_i, \boldsymbol{\xi})$$

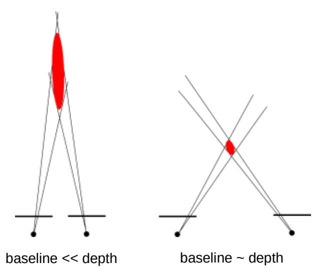
  - Neg. log-likelihood: $E(X, \boldsymbol{\xi}) = -\log(p(X, \boldsymbol{\xi})) \sum_{i=1}^{N} \log(p(\mathbf{y}_i \mid \mathbf{x}_i, \boldsymbol{\xi}))$
  - Gaussian prior and observation likelihood:

$$E(X, \boldsymbol{\xi}) = \text{const.} + \left(\boldsymbol{\xi} - \boldsymbol{\mu}_{\boldsymbol{\xi},0}\right)^{\top} \boldsymbol{\Sigma}_{\boldsymbol{\xi},0}^{-1} \left(\boldsymbol{\xi} - \boldsymbol{\mu}_{\boldsymbol{\xi},0}\right) + \sum_{i=1}^{N} \left(\mathbf{x}_i - \boldsymbol{\mu}_{\mathbf{x}_i,0}\right)^{\top} \boldsymbol{\Sigma}_{\mathbf{x}_i,0}^{-1} \left(\mathbf{x}_i - \boldsymbol{\mu}_{\mathbf{x}_i,0}\right) + \left(\mathbf{y}_i - \pi\left(\mathbf{T}(\boldsymbol{\xi})\mathbf{x}_i\right)\right)^{\top} \boldsymbol{\Sigma}_{\mathbf{y}_i}^{-1} \left(\mathbf{y}_i - \pi\left(\mathbf{T}(\boldsymbol{\xi})\mathbf{x}_i\right)\right)$$



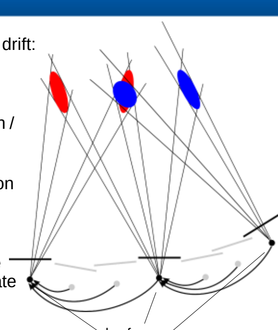
Drift in Motion Estimates

- Estimation errors accumulate: Drift
- Noisy observations in 2D image point location
- Motion estimation and triangulation accuracy depend on ratio of baseline to depth
- 3D-to-3D vs. 2D-to-3D:
  - Low 3D triangulation accuracy for small baseline
  - 3D-to-3D: 2x triangulation, typically less accurate than 2D-to-3D



Keyframes

- Popular approach to reduce drift: Keyframes
- Carefully select reference images for motion estimation / triangulation
- Incrementally estimate motion towards keyframe
- If baseline sufficient (and/or image overlap small), create next keyframe [and triangulate 3D positions of keypoints]



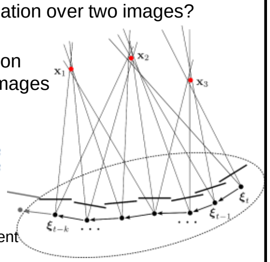
Motion Estimation for Input Type

Correspondences	Monocular	Stereo	RGB-D
2D-to-2D	X	X	X
2D-to-3D	X	X	X
3D-to-3D		X	X

Local Optimization Windows

- Can we do better than optimization over two images?
- Optimize motion / reconstruction on a local current window of images

$$E(X_{t-k:t}, \boldsymbol{\xi}_{t-k:t}) = \sum_{j=0}^{k} \sum_{i=1}^{N_{t-j}} \left\| \mathbf{y}_{t-j,i} - \pi\left(\mathbf{T}(\boldsymbol{\xi}_{t-j})\mathbf{x}_{t-j,i}\right) \right\|_2^2$$



- Local bundle adjustment
- Local motion-only bundle adjustment (3D keypoint positions held fixed)
- Initialize with algebraic approaches

Summary

- Visual odometry estimates relative camera motion from image sequences
- Indirect point-based methods
  - Minimize geometric reprojection error
  - 2D-to-2D, 2D-to-3D, 3D-to-3D motion estimation
  - RANSAC for robust keypoint matching
  - Keyframes can reduce drift
  - Local optimization window can further increase accuracy
- *Next: direct methods*

Topics of This Lecture

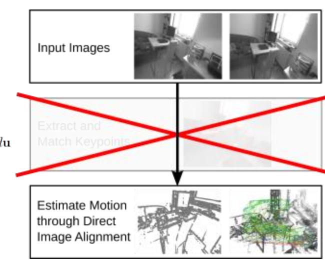
- Recap: Point-based Visual Odometry
  - Further Considerations
- Direct Methods
  - Direct image alignment
  - Pose parametrization
  - Lie group se(3) and the exponential map
  - Residual linearization
  - Practical considerations

Direct Visual Odometry Pipeline

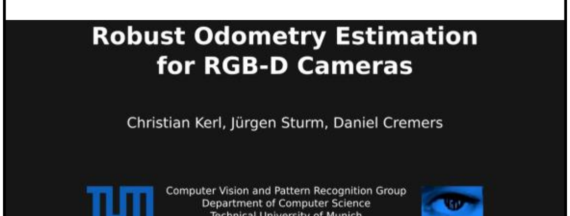
- Avoid manually designed keypoint detection and matching
- Instead: direct image alignment

$$E(\boldsymbol{\xi}) = \int_{\mathbf{u}\in\Omega} \left| \mathbf{I}_1(\mathbf{u}) - \mathbf{I}_2(\omega(\mathbf{u}, \boldsymbol{\xi})) \right| d\mathbf{u}$$

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

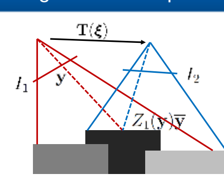


Direct Visual Odometry Example (RGB-D)



C. Kerl, J. Sturm, D. Cremers. Robust Odometry Estimation for RGB-D Cameras. ICRA 2013

Direct Image Alignment Principle



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

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

Recap: General Lukas-Kanade Alignment

- Goal
  - Find the warping parameters $\mathbf{p}$ that minimize the sum-of-squares intensity difference b/w the template image and the warped input image
- LK formulation
  - Formulate this as an optimization problem

$$\arg\min_{\mathbf{p}} \sum_{\mathbf{x}} \left[I(\mathbf{W}(\mathbf{x};\mathbf{p})) - T(\mathbf{x}) \right]^2$$

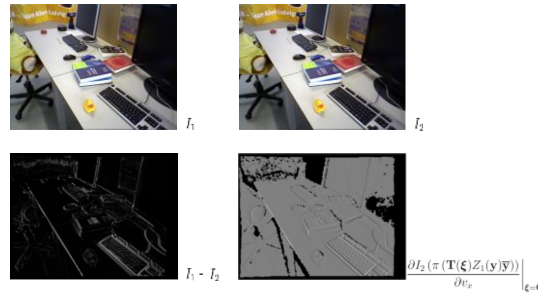
$$I_2\left(\pi\left(\mathbf{T}(\boldsymbol{\xi})Z_1(\mathbf{y})\overline{\mathbf{y}}\right)\right) \qquad I_1(\mathbf{y})$$

  - We assume that an initial estimate of $\mathbf{p}$ is known and iteratively solve for increments to the parameters $\Delta\mathbf{p}$:

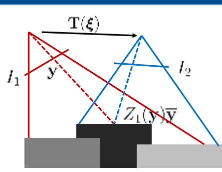
$$\arg\min_{\Delta\mathbf{p}} \sum_{\mathbf{x}} \left[I(\mathbf{W}(\mathbf{x};\mathbf{p}+\Delta\mathbf{p})) - T(\mathbf{x}) \right]^2$$

$$\delta\boldsymbol{\xi} \oplus \boldsymbol{\xi}$$

Derivative of Image Warp



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

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

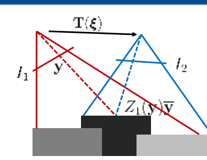
Probabilistic Direct Image Alignment

- 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 pixel measurements are stochastically independent, we can formulate the a-posteriori probability

$$p(\boldsymbol{\xi} \mid I_1, I_2) \propto p(I_1 \mid \boldsymbol{\xi}, I_2)p(\boldsymbol{\xi})$$

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

Optimization Approach

- Optimize negative log-likelihood
  - Product of exponentials becomes a summation over quadratic terms
  - Normalizers are independent of the pose

$$E(\boldsymbol{\xi}) = \sum_{\mathbf{y}\in\Omega} \frac{r(\mathbf{y},\boldsymbol{\xi})^2}{\sigma_I^2}$$, 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 second-order tools (Gauss-Newton, Levenberg-Marquardt)

Gauss-Newton for Non-Linear Least Squares

- Gauss-Newton method, iterate:
  - Linearize residuals:

$$\widetilde{\mathbf{r}}(\boldsymbol{\xi}) = \mathbf{r}(\boldsymbol{\xi}_i) + \nabla_{\boldsymbol{\xi}}\mathbf{r}(\boldsymbol{\xi}_i)(\boldsymbol{\xi} - \boldsymbol{\xi}_i) \qquad \mathbf{J}_i := \nabla_{\boldsymbol{\xi}}\mathbf{r}(\boldsymbol{\xi}_i) \in \mathbb{R}^{\dim(\mathbf{r})\times\dim(\boldsymbol{\xi})}$$

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

$$\nabla_{\boldsymbol{\xi}}\widetilde{E}(\boldsymbol{\xi}) = \mathbf{J}_i^\top \mathbf{W}\widetilde{\mathbf{r}}(\boldsymbol{\xi})$$

$$\nabla_{\boldsymbol{\xi}}^2\widetilde{E}(\boldsymbol{\xi}) = \mathbf{J}_i^\top \mathbf{W}\mathbf{J}_i =: \mathbf{H}_i \in \mathbb{R}^{\dim(\boldsymbol{\xi})\times\dim(\boldsymbol{\xi})}$$

  - Find minimum of linearized system, linearize and set $\nabla_{\boldsymbol{\xi}}\widetilde{E}(\boldsymbol{\xi}) = \mathbf{0}$:

$$\nabla_{\boldsymbol{\xi}}\widetilde{E}(\boldsymbol{\xi}) \approx \nabla_{\boldsymbol{\xi}}\widetilde{E}(\boldsymbol{\xi}_i) + \nabla_{\boldsymbol{\xi}}^2\widetilde{E}(\boldsymbol{\xi}_i)(\boldsymbol{\xi} - \boldsymbol{\xi}_i)$$

$$\boldsymbol{\xi}_{i+1} = \boldsymbol{\xi}_i - \left(\nabla_{\boldsymbol{\xi}}^2\widetilde{E}(\boldsymbol{\xi}_i)\right)^{-1}\nabla_{\boldsymbol{\xi}}\widetilde{E}(\boldsymbol{\xi}_i) = \boldsymbol{\xi}_i - \mathbf{H}_i^{-1}\mathbf{J}_i^\top \mathbf{W}\mathbf{r}(\boldsymbol{\xi}_i)$$

Levenberg-Marquardt Method

- Due to linearization, $\mathbf{H}_i$ may not be a good approximation of the Hessian far from the optimum (could even be degenerate)
- Idea: „damping“ of step-length trades-off between Gauss-Newton and gradient descent

$$\boldsymbol{\xi}_{i+1} = \boldsymbol{\xi}_i - (\mathbf{H}_i + \lambda \mathbf{I})^{-1} \mathbf{J}_i^\top \mathbf{W} \mathbf{r}(\boldsymbol{\xi}_i)$$

  - If error decreases, decrease λ to shift towards Gauss-Newton
  - If error increases, reject update and increase λ to rather perform gradient descent
  - Can converge from worse starting conditions than Gauss-Newton, but requires more iterations

Efficient Non-Linear Least Squares

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

$$\mathbf{H}_i = \sum_{\mathbf{y} \in \Omega} \frac{w(\mathbf{y}, \boldsymbol{\xi}_i)}{\sigma_I^2} \mathbf{J}_{i,\mathbf{y}}^\top \mathbf{J}_{i,\mathbf{y}} \qquad \mathbf{b}_i = \sum_{\mathbf{y} \in \Omega} \mathbf{J}_{i,\mathbf{y}}^\top \frac{w(\mathbf{y}, \boldsymbol{\xi}_i)}{\sigma_I^2} r(\mathbf{y}, \boldsymbol{\xi}_i)$$

$$\mathbf{J}_{i,\mathbf{y}} := \nabla_{\delta\boldsymbol{\xi}} r(\mathbf{y}, \delta\boldsymbol{\xi} \oplus \boldsymbol{\xi}_i)$$

  - This allows for highly efficient parallel processing, e.g., using a GPU

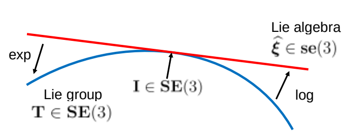
Pose Parametrization for Optimization

- 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
  - Twist coordinates of elements in Lie Algebra se(3) of SE(3) (axis-angle / translation)

Topics of This Lecture

- Recap: Point-based Visual Odometry
  - Further Considerations
- Direct Methods
  - Direct image alignment
  - Pose parametrization
  - Lie group se(3) and the exponential map
  - Residual linearization
  - Practical considerations

Representing Motion using Lie Algebra se(3)



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

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

$$\hat{\boldsymbol{\xi}} := \begin{pmatrix} \hat{\boldsymbol{\omega}} & \mathbf{v} \\ \mathbf{0} & 0 \end{pmatrix} \in \mathbb{R}^{4\times4}$$

- $\mathbf{SE}(3)$ 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 $\hat{\boldsymbol{\xi}} \in \mathbf{se}(3)$ form the tangent space of $\mathbf{SE}(3)$ at identity
- The $\mathbf{se}(3)$ elements can be interpreted as rotational and translational velocities (twists)

Insights into se(3)

- Let‘s look at rotations first and assume time-continuous motion
  - We know that $\mathbf{R}(t)\mathbf{R}^\top(t) = \mathbf{I}$
  - Taking the derivative for time yields $\dot{\mathbf{R}}(t)\mathbf{R}^\top(t) = -\mathbf{R}(t)\dot{\mathbf{R}}^\top(t)$
  - This means there exists a skew-symmetric matrix $\hat{\boldsymbol{\omega}}(t) = -\hat{\boldsymbol{\omega}}^\top(t)$ such that $\dot{\mathbf{R}}(t) = \hat{\boldsymbol{\omega}}(t)\mathbf{R}(t)$
  - Assume constant $\hat{\boldsymbol{\omega}}(t)$ and solve linear ordinary differential equation (ODE):

$$\mathbf{R}(t) = \exp(\hat{\boldsymbol{\omega}} t)\mathbf{R}(0)$$

  - Further assuming $\mathbf{R}(0) = \mathbf{I}$, we obtain $\mathbf{R}(t) = \exp(\hat{\boldsymbol{\omega}} t)$
  - Matrix exponential has a closed-form solution; $\hat{\boldsymbol{\omega}} t$ corresponds to minimal axis-angle representation

Further Insights into se(3)

– For continuous rigid-body motion we can write

$$\dot{\mathbf{T}}(t) = \left(\dot{\mathbf{T}}(t)\mathbf{T}^{-1}(t)\right)\mathbf{T}(t) = \widehat{\xi}(t)\mathbf{T}(t) \qquad \widehat{\xi}(t) := \begin{pmatrix} \widehat{\omega}(t) & \mathbf{v}(t) \\ \mathbf{0} & 0 \end{pmatrix}$$

– Interpretation: tangent vector along curve of $\mathbf{T}(t)$

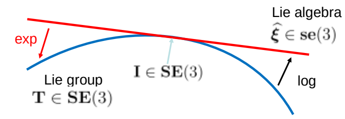
– Again, for constant $\widehat{\xi}(t)$ this linear ODE has a unique solution:

$$\mathbf{T}(t) = \exp\left(\widehat{\xi}t\right)\mathbf{T}(0)$$

– For initial condition $\mathbf{T}(0) = \mathbf{I}$, we have $\mathbf{T}(t) = \exp\left(\widehat{\xi}t\right)$

– To reduce clutter in notation, we will absorb t into $\widehat{\omega}$ and $\widehat{\xi}$

Exponential Map of SE(3)

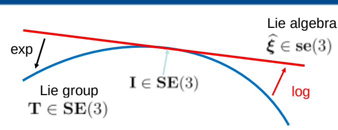


• The exponential map finds the transformation matrix for a twist:

$$\exp\left(\widehat{\xi}\right) = \begin{pmatrix} \exp(\widehat{\omega}) & \mathbf{Av} \\ \mathbf{0} & 1 \end{pmatrix}$$

$$\exp(\widehat{\omega}) = \mathbf{I} + \frac{\sin|\omega|}{|\omega|}\widehat{\omega} + \frac{1-\cos|\omega|}{|\omega|^2}\widehat{\omega}^2 \qquad \mathbf{A} = \mathbf{I} + \frac{1-\cos|\omega|}{|\omega|^2}\widehat{\omega} + \frac{|\omega| - \sin|\omega|}{|\omega|^3}\widehat{\omega}^2$$

Logarithm Map of SE(3)



• The logarithm maps twists to transformation matrices:

$$\log(\mathbf{T}) = \begin{pmatrix} \log(\mathbf{R}) & \mathbf{A}^{-1}\mathbf{t} \\ \mathbf{0} & 0 \end{pmatrix}$$

$$\log(\mathbf{R}) = \frac{|\omega|}{2\sin|\omega|}\left(\mathbf{R} - \mathbf{R}^T\right) \qquad |\omega| = \cos^{-1}\left(\frac{\mathrm{tr}(\mathbf{R}) - 1}{2}\right)$$

Some Notation for Twist Coordinates

• Let's define the following notation:

– Inversion of hat operator: $\begin{pmatrix} 0 & -\omega_3 & \omega_2 & v_1 \\ \omega_3 & 0 & -\omega_1 & v_2 \\ -\omega_2 & \omega_1 & 0 & v_3 \\ 0 & 0 & 0 & 0 \end{pmatrix}^{\vee} = (\omega_1\ \omega_2\ \omega_3\ v_1\ v_2\ v_3)^{\top}$

– Conversion: $\xi(\mathbf{T}) = (\log(\mathbf{T}))^{\vee}, \quad \mathbf{T}(\xi) = \exp(\widehat{\xi})$

– Pose inversion: $\xi^{-1} = \log(\mathbf{T}(\xi)^{-1}) = -\xi$

– Pose concatenation: $\xi_1 \oplus \xi_2 = (\log(\mathbf{T}(\xi_2)\mathbf{T}(\xi_1)))^{\vee}$

– Pose difference: $\xi_1 \ominus \xi_2 = \left(\log\left(\mathbf{T}(\xi_2)^{-1}\mathbf{T}(\xi_1)\right)\right)^{\vee}$

Optimization with Twist Coordinates

• Twists provide a minimal local representation without singularities

• Since $\mathbf{SE}(3)$ is a smooth manifold, we can decompose transformations in each optimization step into the transformation itself and an infinitesimal increment

But!

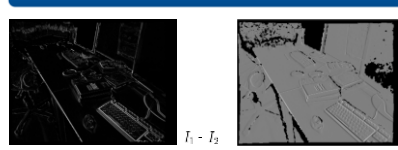
$$\mathbf{T}(\xi) = \mathbf{T}(\xi)\exp\left(\widehat{\delta\xi}\right) = \mathbf{T}(\delta\xi \oplus \xi) \qquad \mathbf{T}(\xi + \delta\xi) \neq \mathbf{T}(\xi)\,\mathbf{T}(\delta\xi)$$

• Example: Gradient descent on the auxiliary variable

$$\delta\xi^* = \mathbf{0} - \eta\nabla_{\delta\xi}E(\xi_i, \delta\xi)$$

$$\mathbf{T}(\xi_{i+1}) = \mathbf{T}(\xi_i)\exp\left(\widehat{\delta\xi^*}\right)$$

Properties of Residual Linearization



• Linearizing residuals yields

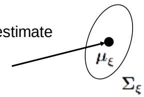
$$\nabla_{\xi} r(\mathbf{y}, \xi) = -\nabla_{\pi} I_2(\omega(\mathbf{y}, \xi))\ \nabla_{\xi}\omega(\mathbf{y}, \xi)$$

with $\omega(\mathbf{y}, \xi) := \pi(\mathbf{T}(\xi)Z_1(\mathbf{y})\bar{\mathbf{y}})$

– Linearization is only valid for motions that change the projection in a small image neighborhood that is captured by the local gradient

Distribution of the Pose Estimate

- Remark
  - Non-linear least squares determines a Gaussian estimate



$$p(\boldsymbol{\xi} \mid I_1, I_2) = \mathcal{N}\left(\boldsymbol{\mu}_\xi, \Sigma_\xi\right)$$

$$\Sigma_\xi = \left(\nabla_\xi \mathbf{r}(\boldsymbol{\xi})^\top \mathbf{W} \nabla_\xi \mathbf{r}(\boldsymbol{\xi})\right)^{-1}$$

  - Due to right-multiplication of pose increment $\boldsymbol{\delta\xi}$, the covariance from the Hessian is expressed in camera frame I_1
  - Pose covariance in frame I_2 can be obtained using the adjoint in $\mathbf{SE}(3)$

$$p(\boldsymbol{\xi} \mid I_1, I_2) = \mathcal{N}\left(\boldsymbol{\mu}_\xi, \mathrm{ad}_{\mathbf{T}(\boldsymbol{\xi})} \Sigma_{\delta\xi} \mathrm{ad}_{\mathbf{T}(\boldsymbol{\xi})}^\top\right)$$

$$\Sigma_{\delta\xi} = \left(\nabla_{\delta\xi} \mathbf{r}(\boldsymbol{\delta\xi}, \boldsymbol{\xi})^\top \mathbf{W} \nabla_{\delta\xi} \mathbf{r}(\boldsymbol{\delta\xi}, \boldsymbol{\xi})\right)^{-1}$$

$$\mathrm{ad}_{\mathbf{T}(\boldsymbol{\xi})} = \begin{pmatrix} \mathbf{R}(\boldsymbol{\xi}) & \mathbf{0} \\ \hat{\mathbf{t}}\mathbf{R}(\boldsymbol{\xi}) & \mathbf{R}(\boldsymbol{\xi}) \end{pmatrix}$$

Algorithm: Direct RGB-D Visual Odometry

Input: RGB-D image sequence $I_{0:t}, Z_{0:t}$
Output: aggregated camera poses $\mathbf{T}_{0:t}$

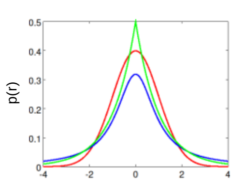
Algorithm:
For each current RGB-D image I_k, Z_k :
1. Estimate relative camera motion $\mathbf{T}_k^{k-1}$ towards the previous RGB-D frame using direct image alignment
2. Concatenate estimated camera motion with previous frame camera pose to obtain current camera pose estimate $\mathbf{T}_k = \mathbf{T}_{k-1}\mathbf{T}_k^{k-1}$

Topics of This Lecture

- Recap: Point-based Visual Odometry
  - Further Considerations
- Direct Methods
  - Direct image alignment
  - Pose parametrization
  - Lie group se(3) and the exponential map
  - Residual linearization
  - Practical considerations

Coarse-To-Fine Optimization

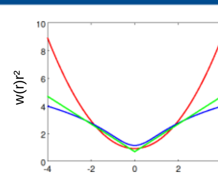
Residual Distributions



- Normal distribution
- Laplace distribution
- Student-t distribution

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

Optimizing Non-Gaussian Measurement Noise



- Normal distribution
- Laplace distribution
- Student-t distribution

- Accommodating different noise distributions
  - 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

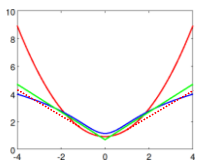
$$E(\boldsymbol{\xi}) = \sum_{\mathbf{y} \in \Omega} w\left(r(\mathbf{y}, \boldsymbol{\xi})\right) \frac{r(\mathbf{y}, \boldsymbol{\xi})^2}{\sigma_I^2}$$

Laplace distribution:
$$w\left(r(\mathbf{y}, \boldsymbol{\xi})\right) = |r(\mathbf{y}, \boldsymbol{\xi})|^{-1}$$

Huber Loss

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

$$\|r\|_\delta = \begin{cases} \frac{1}{2}\|r\|_2^2 & \text{if } \|r\|_2 \leq \delta \\ \delta\left(\|r\|_1 - \frac{1}{2}\delta\right) & \text{otherwise} \end{cases}$$

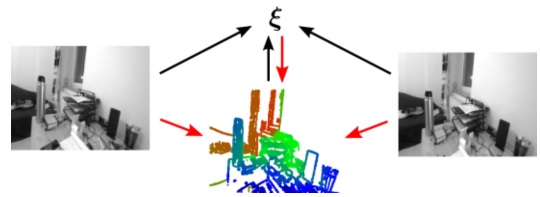


- Normal distribution
- Laplace distribution
- Student-t distribution

Huber-loss for δ = 1

Monocular Direct Visual Odometry

- Estimate motion and depth concurrently



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

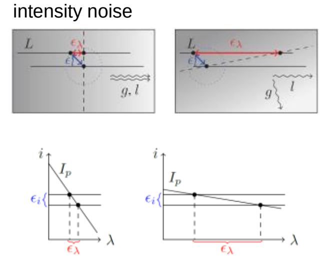
Semi-Dense Mapping

- Idea
  - 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

Semi-Dense Mapping

- Inverse depth uncertainty estimate from geometric and intensity noise



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

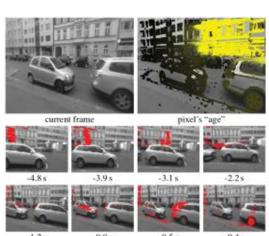
Intensity noise

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

intensity noise variance; image gradient magnitude at epipolar line

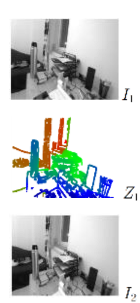
Choosing the Stereo Reference Frame

- Naive:
  - Use one specific reference frame (e.g., the previous frame or a keyframe)
- Better alternative:
  - Select the reference frame for stereo comparisons for each pixel individually in order to achieve a trade-off between accuracy and computation time



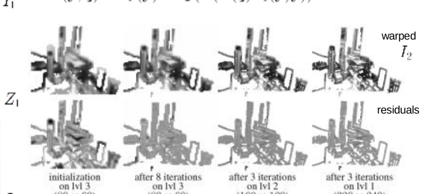
- Heuristics from Engel et al., ICCV 2013:
  - Use oldest frame in which pixel is still visible but disparity search range and observation angle are below threshold

Semi-Dense Direct Image Alignment



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

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

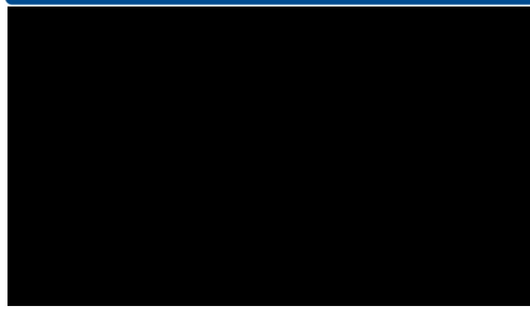


Algorithm: Direct Monocular Visual Odometry

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

Algorithm:
Initialize depth map Z_0 f.e. 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 $\tilde{Z}_k$
4. Update propagated semi-dense depth map $\tilde{Z}_k$ with temporal stereo depth measurements to obtain Z_k

Direct Visual Odometry Example (Monocular)



Engel et al., Semi-Dense Visual Odometry for a Monocular Camera, ICCV 2013

Summary

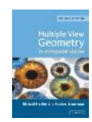
- Direct image alignment avoids manually designed keypoints, can use all available image information
- Direct visual odometry
  - Dense RGB-D odometry by direct image alignment with measured depth
  - Direct image alignment for monocular cameras requires depth estimation from temporal stereo
- Direct image alignment as non-linear least squares problem
  - Linearization of the residuals requires a coarse-to-fine optimization scheme
  - Gaussian distribution on pose can be obtained
  - 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

References and Further Reading

- MASKS and MVG textbooks

An Invitation to 3D Vision, Y. Ma, S. Soatto, J. Kosecka, and S. S. Sastry, Springer, 2004

MASKS



Multiple View Geometry in Computer Vision, R. Hartley and A. Zisserman, Cambridge University Press, 2004

MVG

- Publications:
  - C. Kerl, J. Sturm, D. Cremers. Robust Odometry Estimation for RGB-D Cameras. ICRA 2013.
  - J. Engel, J. Sturm, D. Cremers. Semi-Dense Visual Odometry for a Monocular Camera. ICCV 2013.