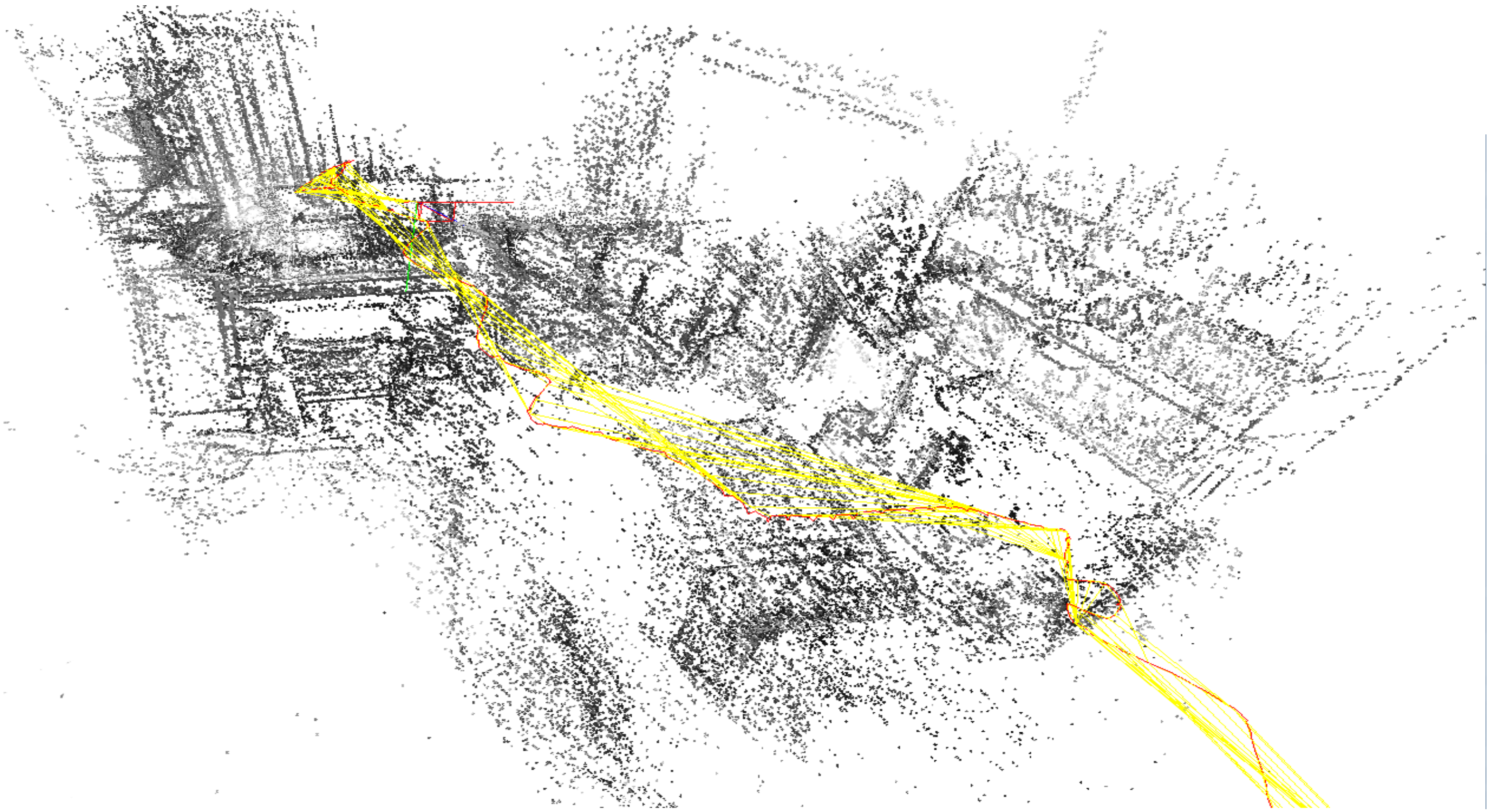
An Introduction to Direct Sparse Odometry with Loop Closure

Tong Ling



FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG
TECHNISCHE FAKULTÄT

LIKE

Agenda

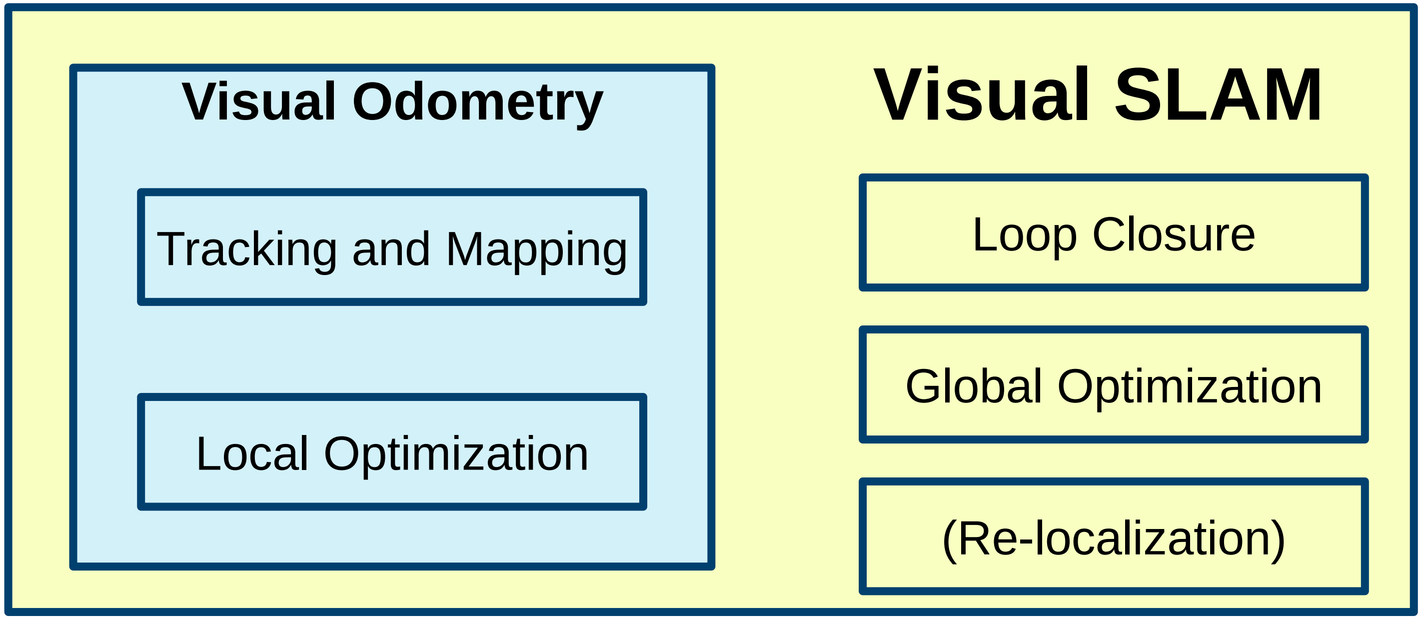
1. Introduction
2. Direct Sparse Odometry [Engel-18]
3. Direct Sparse Odometry with Loop Closure [Gao-18]

Agenda

1. Introduction
   - Visual Odometry vs. Visual SLAM
   - Notation
   - Indirect vs. Direct Method
   - Sparse vs. Dense
   - Why Direct + Sparse?
2. Direct Sparse Odometry
3. Direct Sparse Odometry with Loop Closure

Visual Odometry vs. Visual SLAM

- VO computes the camera path incrementally (pose after pose).
- VO is only concerned with the local consistency of the trajectory, whereas V-SLAM with the global consistency. [Scaramuzza-11]



Notation

- Notation in Computer Vision / Robotics [Hartley-03] [Barfoot-17]
  - Point (homogeneous) : $[x \quad y \quad w]^{\mathbf{T}}$, $[x \quad y \quad z \quad w]^{\mathbf{T}}$.
  - Camera intrinsics (3×3) : $\mathbf{K}$
  - Camera extrinsics (3×4): $\mathbf{T}_{\mathrm{cw}} = [\mathbf{R} \quad \mathbf{t}]$, where $\mathbf{R} \in \mathbb{R}^{3\times3}$, $\mathbf{t} \in \mathbb{R}^3$
  - Projection matrix (3×4) : $\mathbf{\Pi}_{\mathrm{pw}} = \mathbf{T}_{\mathrm{pc}}\mathbf{T}_{\mathrm{cw}} = \mathbf{K}[\mathbf{R} \quad \mathbf{t}]$
  - Rotation (Lie group) : $\mathbf{SO}(3) = \{\mathbf{R} \in \mathbb{R}^{3\times3} | \mathbf{R}\mathbf{R}^{\mathbf{T}} = 1, \det \mathbf{R} = 1\}$
  - Pose (Lie group) : $\mathbf{SE}(3) = \left\{\mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^{\mathrm{T}} & 1 \end{bmatrix} \in \mathbb{R}^{4\times4} \middle| \mathbf{R} \in \mathbf{SO}(3), \mathbf{t} \in \mathbb{R}^3\right\}$
  - Rotation (Lie algebra) : $\mathfrak{so}(3) = \{\mathbf{\Phi} = \phi^\wedge \in \mathbb{R}^{3\times3} | \phi \in \mathbb{R}^3\}$
  - Pose (Lie algebra) : $\mathfrak{se}(3) = \{\mathbf{\Xi} = \xi^\wedge \in \mathbb{R}^{4\times4} | \xi \in \mathbb{R}^6\}$
  - Relationship : $\mathbf{R} = \exp(\phi^\wedge), \mathbf{T} = \exp(\xi^\wedge)$

$\boxed{\frac{dF(\mathbf{T})}{d\mathbf{T}}???}$

LIKE

Notation

- Notation in slides
  - $\mathbf{X}$: big, bold → Matrix / Function with matrix output
  - X: big, italic → Function with scalar output
  - $\mathbf{x}$: small, bold → Vector / Function with vector output
  - x: small, italic → Scalar
  - x: small → Letter / Word

  - $a \coloneqq b$: Assign b's value to a
  - $a \propto b$: a is proportional to b

Indirect vs. Direct Method

- **Core**: Maximum Likelihood Estimation.

  $\leftrightarrow \mathbf{x}^* \coloneqq \arg\max_{\mathbf{x}} P(\mathbf{z}|\mathbf{x})$

  $P(\mathbf{z}|\mathbf{x})$: Probabilistic model (Likelihood)

  $\mathbf{x}$: Model parameters (Camera pose, 3D geometry, camera intrinsics)

  $\mathbf{z}$: Observations / Measurements.

- Indirect

  $\mathbf{z}$: Position of 2D points. $\leftrightarrow$ Geometric measurement.

- Direct

  $\mathbf{z}$: Pixel intensities. $\leftrightarrow$ Photometric measurement.

Indirect vs. Direct Method

- **Example**

$\mathbf{z} = \mathbf{h}(\mathbf{x}) + \mathbf{n}$, where $\mathbf{n} \sim N(\mathbf{0}, \boldsymbol{\Sigma})$.

  - $P(\mathbf{z}|\mathbf{x}) = N(\mathbf{h}(\mathbf{x}), \boldsymbol{\Sigma}) = \det(2\pi\boldsymbol{\Sigma})^{-\frac{1}{2}}\exp\left(-\frac{1}{2}\left(\mathbf{z}-\mathbf{h}(\mathbf{x})\right)^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\left(\mathbf{z}-\mathbf{h}(\mathbf{x})\right)\right)$

    $-\ln\left(P(\mathbf{z}|\mathbf{x})\right) = -\frac{1}{2}\ln(\det(2\pi\boldsymbol{\Sigma})) + \frac{1}{2}\left(\mathbf{z}-\mathbf{h}(\mathbf{x})\right)^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\left(\mathbf{z}-\mathbf{h}(\mathbf{x})\right)$

  - $\mathbf{x}^* \coloneqq \arg\max_{\mathbf{x}} P(\mathbf{z}|\mathbf{x}) = \arg\min_{\mathbf{x}}(\frac{1}{2}\left(\mathbf{z}-\mathbf{h}(\mathbf{x})\right)^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\left(\mathbf{z}-\mathbf{h}(\mathbf{x})\right))$

    $= \arg\min_{\mathbf{x}}(E(\mathbf{x}))$

    $= \arg\min_{\mathbf{x}}(\frac{1}{2}\mathbf{e}(\mathbf{x})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\mathbf{e}(\mathbf{x}))$, where $\mathbf{e}(\mathbf{x}) = [\mathrm{z}_1 - \mathrm{h}_1(\mathbf{x}) \quad \cdots \quad \mathrm{z}_i - \mathrm{h}_i(\mathbf{x})]^{\mathbf{T}}$

  - $E(\mathbf{x})$: cost/energy/error function.

    $\mathbf{e}(\mathbf{x})$: residual vector made of single residuals.

    $\boldsymbol{\Sigma}$: covariance matrix.

LIKE

Indirect vs. Direct Method

- **Linearization**
  - Linearization at $\mathbf{x}_k$: $\mathbf{e}(\mathbf{x}) \approx \mathbf{e}(\mathbf{x}_k) + \mathbf{J}(\mathbf{x}_k)(\mathbf{x} - \mathbf{x}_k)$
  - $E(\mathbf{x}) = \frac{1}{2}\mathbf{e}(\mathbf{x})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\mathbf{e}(\mathbf{x})$

$$\approx \frac{1}{2}(\mathbf{e}(\mathbf{x}_k) + \mathbf{J}(\mathbf{x}_k)(\mathbf{x} - \mathbf{x}_k)\ )^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{e}(\mathbf{x}_k) + \mathbf{J}(\mathbf{x}_k)(\mathbf{x} - \mathbf{x}_k)\ )$$

$$= \mathrm{const.} + \mathbf{e}(\mathbf{x}_k)^{\mathbf{T}}\boldsymbol{\Sigma}^{-1}\mathbf{J}(\mathbf{x}_k)\mathbf{x} + \frac{1}{2}\mathbf{x}^{\mathbf{T}}\mathbf{J}(\mathbf{x}_k)^{\mathbf{T}}\boldsymbol{\Sigma}^{-1}\mathbf{J}(\mathbf{x}_k)\mathbf{x}$$

$$= \mathrm{const.} + \mathbf{g}(\mathbf{x}_k)^{\mathbf{T}}\mathbf{x} + \frac{1}{2}\mathbf{x}^{\mathbf{T}}\mathbf{H}(\mathbf{x}_k)\mathbf{x}$$

  - $\mathbf{g}(\mathbf{x}_k) = -\boldsymbol{\xi}, \quad \mathbf{H}(\mathbf{x}_k) = \boldsymbol{\Omega}.$

$\mathbf{J}$: Jacobian matrix

$\mathbf{g}$: gradient vector

$\mathbf{H}$: Hessian matrix

Information form: $\mathbf{x} \backsim N^{-1}(\boldsymbol{\xi}, \boldsymbol{\Omega})$

- **Information parameterization ($\boldsymbol{\xi} = \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}$, $\boldsymbol{\Omega} = \boldsymbol{\Sigma}^{-1}$)** [Thrun-05]

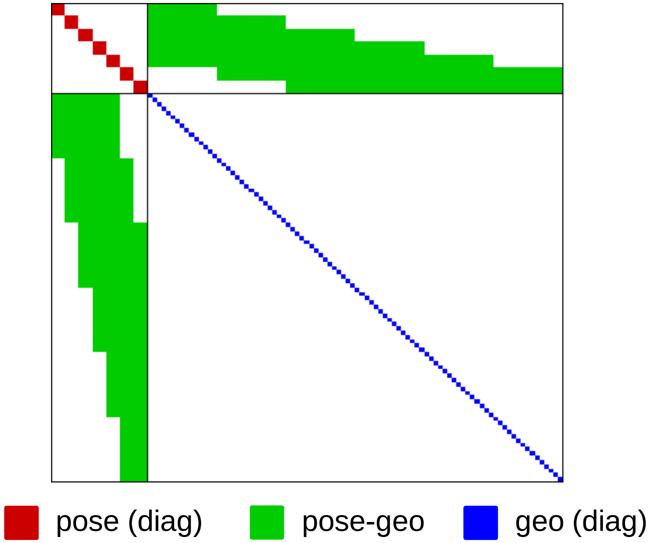
$$P(\mathbf{x}) = \det(2\pi\boldsymbol{\Sigma})^{-\frac{1}{2}}\exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu})\right) = \eta\, \exp(\boldsymbol{\xi}^{\mathbf{T}}\mathbf{x} - \frac{1}{2}\mathbf{x}^{\mathbf{T}}\boldsymbol{\Omega}\mathbf{x})$$

$$-\ln P(\mathbf{x}) = const. + (-\boldsymbol{\xi}^{\mathbf{T}}\mathbf{x} + \frac{1}{2}\mathbf{x}^{\mathbf{T}}\boldsymbol{\Omega}\mathbf{x})$$

Indirect vs. Direct Method

- **Hessian structure** [Frese-05]

$$\mathbf{H}(\mathbf{x}) = \mathbf{J}(\mathbf{x}_k)^{\mathbf{T}}\boldsymbol{\Sigma}^{-1}\mathbf{J}(\mathbf{x}_k),\ \mathbf{J}(\mathbf{x}) = \begin{bmatrix} \frac{\partial \mathbf{e}_1}{\partial \mathbf{x}_1} & \cdots & \frac{\partial \mathbf{e}_1}{\partial \mathbf{x}_n} \\ \vdots & \vdots & \vdots \\ \frac{\partial \mathbf{e}_n}{\partial \mathbf{x}_1} & \cdots & \frac{\partial \mathbf{e}_n}{\partial \mathbf{x}_n} \end{bmatrix}.$$

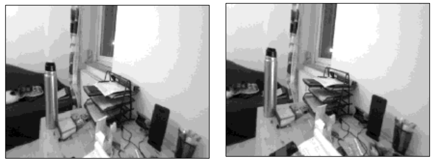


An example of Hessian structure [Engel-18]

Indirect vs. Direct Method

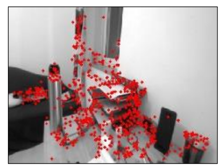
Indirect (Feature-Based)

Input Images



Extract & Match Features

(ORB / SIFT / …)

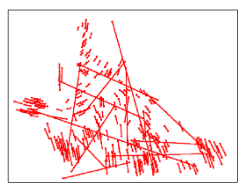


Abstract image to feature observations

Track:

min. **reprojection** error

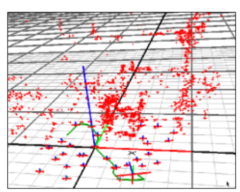
(point distances)



Map:

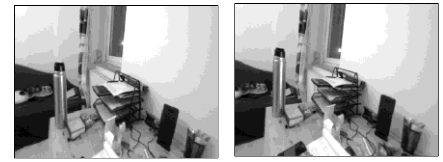
est. feature-parameters

(3D points / normals)



Direct

Input Images



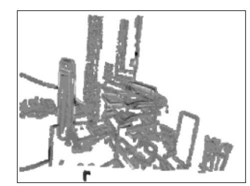
Source: [Schöps-14]

keep full images (no abstraction)

Track:

min. **photometric** error

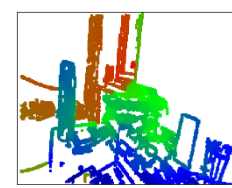
(intensity differences)



Map:

est. per-pixel depth

(semi-dense depth map)



Indirect vs. Direct Method

- $\mathbf{p}_i, \mathbf{p}_i'$: pixel coordinates.

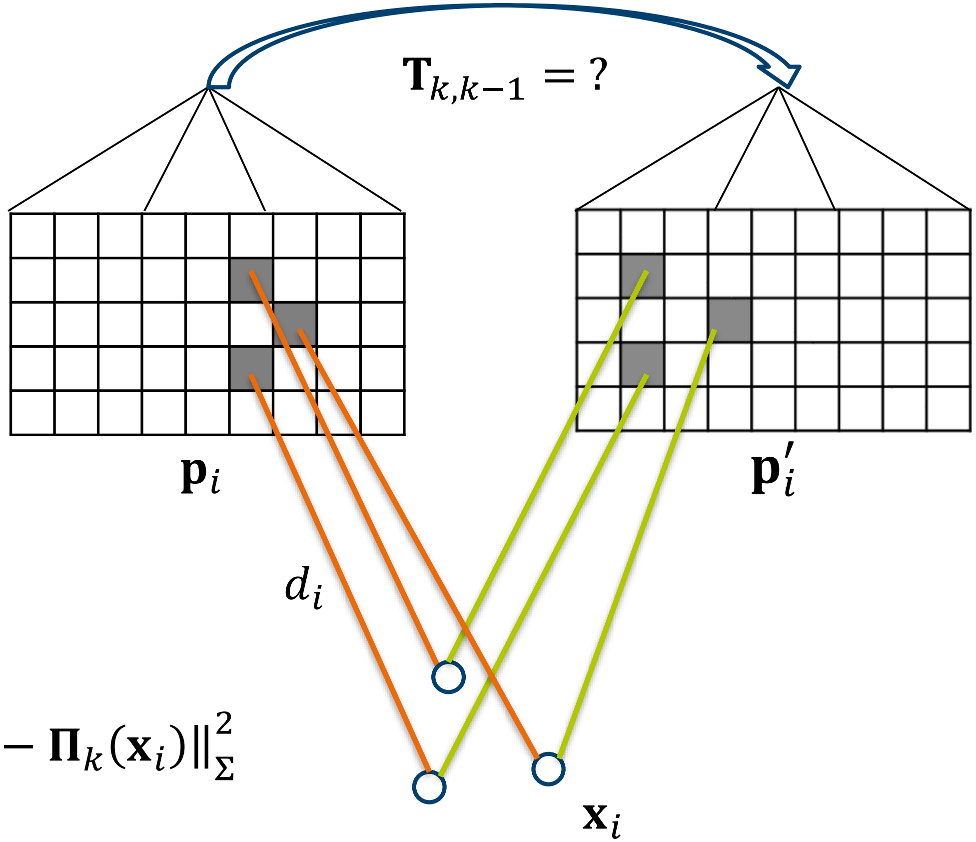
  $\mathbf{x}_i$: coordinates in world system.

  d_i: depth or inverse depth.

  $\mathbf{\Pi}_k(\mathbf{x}_i)$: projection.

  $I_k(\mathbf{p}_i)$: pixel intensity.

  $\mathbf{T}_{k,k-1}$: relative transformation.



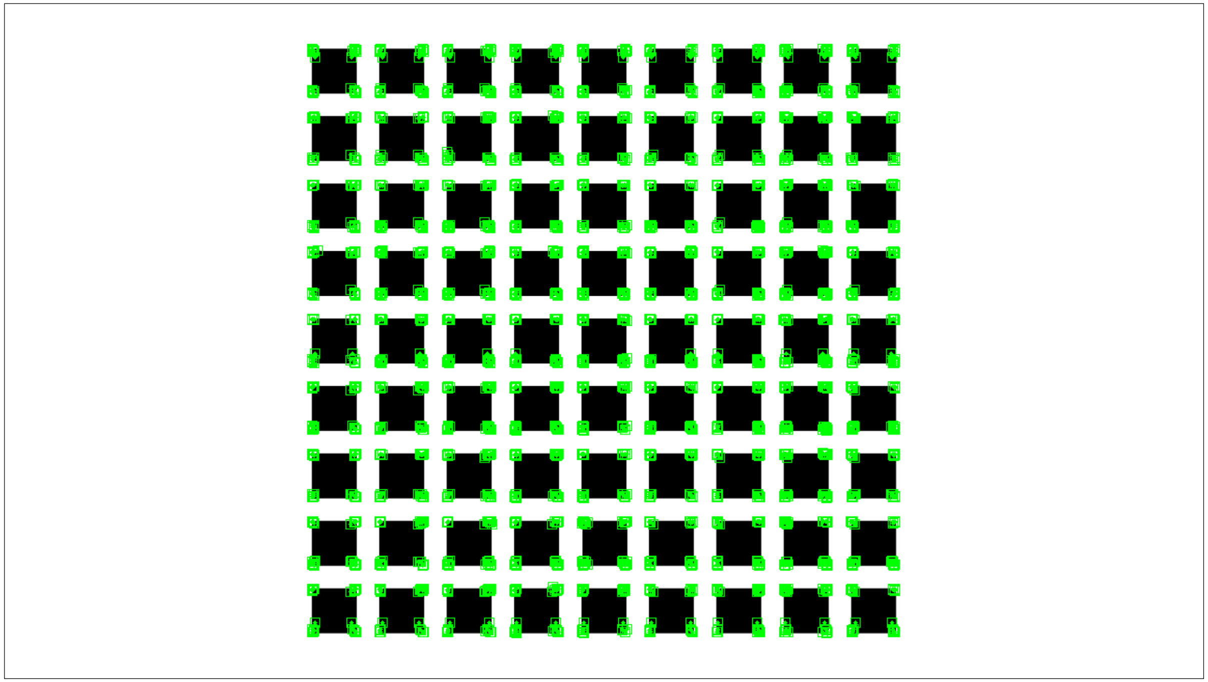
- **Indirect**: $\mathbf{T}_{k,k-1} = \arg\min_{\mathbf{T}} \sum_i \|\mathbf{p}_i' - \mathbf{\Pi}_k(\mathbf{x}_i)\|_\Sigma^2$

- **Direct**: $\mathbf{T}_{k,k-1} = \arg\min_{\mathbf{T}} \sum_i \|I_k(\mathbf{p}_i') - I_{k-1}(\mathbf{p}_i)\|_\sigma^2$

  where $\mathbf{p}_i' = \mathbf{\Pi}_{k-1}\left(\mathbf{T} \cdot (\mathbf{\Pi}_{k-1}^{-1}(\mathbf{p}_i, d)\right)$

Sparse vs. Dense

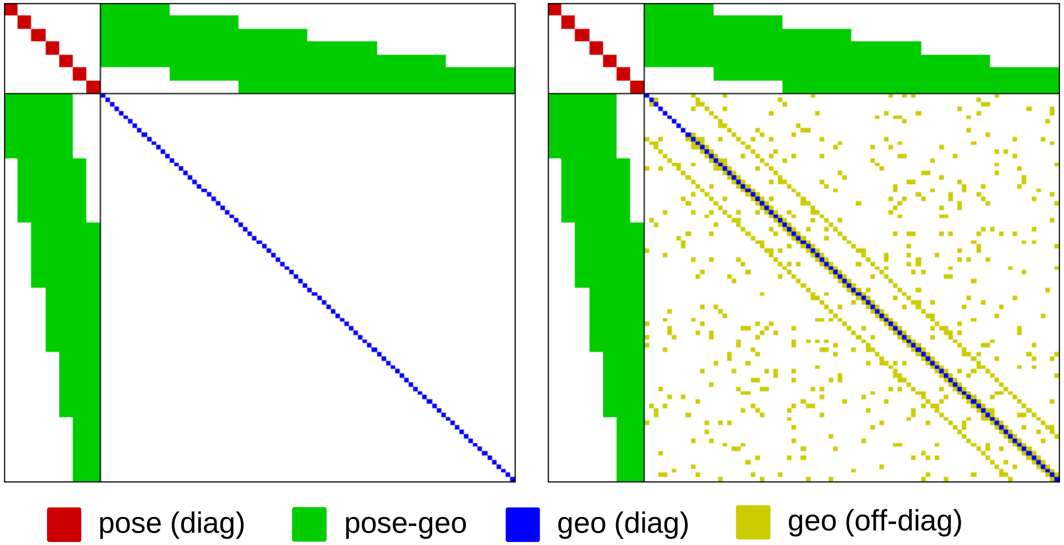
- **The extent of used image region**
  - Sparse: a set of independent points (traditionally corners).
  - Dense: all pixels in the 2D image domain.



Sparse: ORB-SLAM2 ~ 1980 pixels out of a 1280x720 image

Sparse vs. Dense

- **The addition of a geometry prior**
  - Sparse: No notion of neighbors → conditionally independent geometry parameters (keypoint positions).
  - Dense: The connectedness of the used image region. → "The world is smooth."



Sparse vs. Dense Hessian structure [Engel-18]

Why Direct + Sparse?

- **Direct**
  - Special cameras for computer vision algorithms

    → A complete sensor model available
  - No need to know the 3D point position

    → *Inverse depth* parametrization [Civera-08]
  - More useful data in images

    → edge, weak intensity variations
- **Sparse**
  - Feasible in real time.

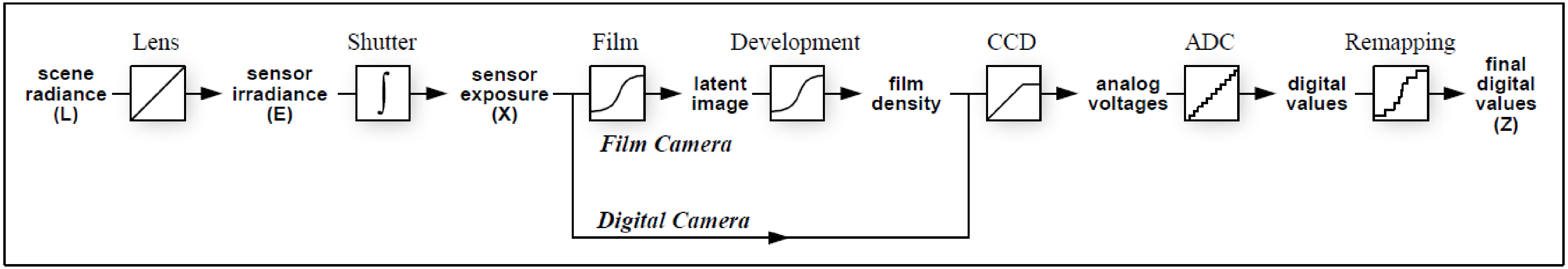
Agenda

1. Introduction
2. Direct Sparse Odometry
   - Photometric Calibration
   - Photometric Error
   - Windowed Optimization
3. Direct Sparse Odometry with Loop Closure

Photometric Calibration

- **Motivation**
  - Take full of the sensor’s capabilities
  - Incorporate knowledge about the sensor design
  - → More meaningful data.

- **Image Acquisition Pipeline** [Debevec-08]



Photometric Calibration

- **Image formation model** [Engel-16]

$$I_i(\mathbf{x}) = G\left(t_i V(\mathbf{x}) B_i(\mathbf{x})\right)$$

pixel value

exposure time

irradiance image

camera response function
(gamma function)

attenuation factor
(vignetting)

$$I_i^{\mathrm{new}}(\mathbf{x}) := t_i B_i(\mathbf{x}) = \frac{G^{-1}(I_i(\mathbf{x}))}{\mathrm{V}(\mathbf{x})}$$

Photometric Error
Known Exposure Time

- Photometric error of one point

$$E_{\mathbf{p}j} := \left\| I_j[\mathbf{p}'] - \frac{t_j}{t_i} I_i[\mathbf{p}] \right\|_\gamma$$

Point $\mathbf{p}$'s host frame is frame i

- $I_i(\mathbf{p}) = \begin{cases} t_i B_i(\mathbf{p}), & \text{with photometric calibration} \\ \text{original image}, & \text{without photometric calibration} \end{cases}$

$\mathbf{c}$: camera intrinsics.

$\mathbf{T}$: relative pose between camera pose $\mathbf{T_i}$ and $\mathbf{T_j}$.

$\mathbf{p}' = \Pi_{\mathbf{c}}(\mathbf{T}\mathbf{\Pi}_{\mathbf{c}}^{-1}(\mathbf{p}, d_{\mathbf{p}})),$

$d_{\mathbf{p}}$: the point's inverse depth.

$\|\cdot\|_\gamma$: Huber norm. $\|r\|_\gamma = \begin{cases} \frac{r^2}{2}, & \text{if } |r| \le \gamma \\ \gamma\left(|r| - \frac{\gamma}{2}\right), & \text{otherwise} \end{cases}$

Photometric Error
Unknown Exposure Time

- Photometric error of one point

$$E_{\mathbf{p}j} \coloneqq \left\| \left(I_j[\mathbf{p}'] - b_j\right) - \frac{t_j e^{a_j}}{t_i e^{a_i}} \left(I_i[\mathbf{p}] - b_i\right) \right\|_\gamma$$

- t_i, t_j: 1.

  $e^{-a_i}(I_i(\mathbf{p}) - b_i)$: affine brightness transfer function.

  $I_i(\mathbf{p}) = \begin{cases} t_i B_i(\mathbf{p}), & \text{with photometric calibration} \\ \text{original image}, & \text{without photometric calibration} \end{cases}$

  $\mathbf{c}$: camera intrinsics.

  $\mathbf{T}$: relative pose between camera pose $\mathbf{T_i}$ and $\mathbf{T_j}$.

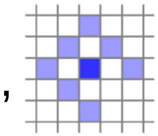
  $\mathbf{p}' = \Pi_{\mathbf{c}}\left(\mathbf{T}\boldsymbol{\Pi}_{\mathbf{c}}^{-1}\left(\mathbf{p}, d_{\mathbf{p}}\right)\right),$

  $d_{\mathbf{p}}$: the point's inverse depth.

  $\|\cdot\|_\gamma$: Huber norm.

LIKE

Photometric Error
General

- Photometric error of one point over a small neighborhood $\mathcal{N}_{\mathbf{p}}$ (8 pixels, )

$$E_{\mathbf{p}j} := \sum_{\mathbf{p}\in\mathcal{N}_{\mathbf{p}}} w_p \left\| \left(I_j[\mathbf{p}'] - b_j\right) - \frac{t_j e^{a_j}}{t_i e^{a_i}} \left(I_i[\mathbf{p}] - b_i\right) \right\|_\gamma$$

$$E_{\text{photo}} := \sum_{i\in\mathcal{F}} \sum_{\mathbf{p}\in\mathcal{P}_i} \sum_{j\in\text{obs}(\mathbf{p})} E_{\mathbf{p}j}$$

other frames observing $\mathbf{p}$

host frame i's points

all frames

- $w_p := \frac{c^2}{c^2 + \|\nabla I_i(\mathbf{p})\|_2^2}$

$e^{-a_i}(I_i - b_i)$: affine brightness transfer function.

$$I_i(\mathbf{p}) = \begin{cases} t_i B_i(\mathbf{p}) & \text{, with photometric calibration} \\ \text{original image} & \text{, without photometric calibration} \end{cases}$$

$\mathbf{c}$: camera intrinsics.

$\mathbf{T}$: relative pose between camera pose $\mathbf{T_i}$ and $\mathbf{T_j}$.

$\mathbf{p}' = \Pi_{\mathbf{c}}(\mathbf{T}\boldsymbol{\Pi}_{\mathbf{c}}^{-1}(\mathbf{p}, d_{\mathbf{p}})),$

$d_{\mathbf{p}}$: the point's inverse depth.

Photometric Error

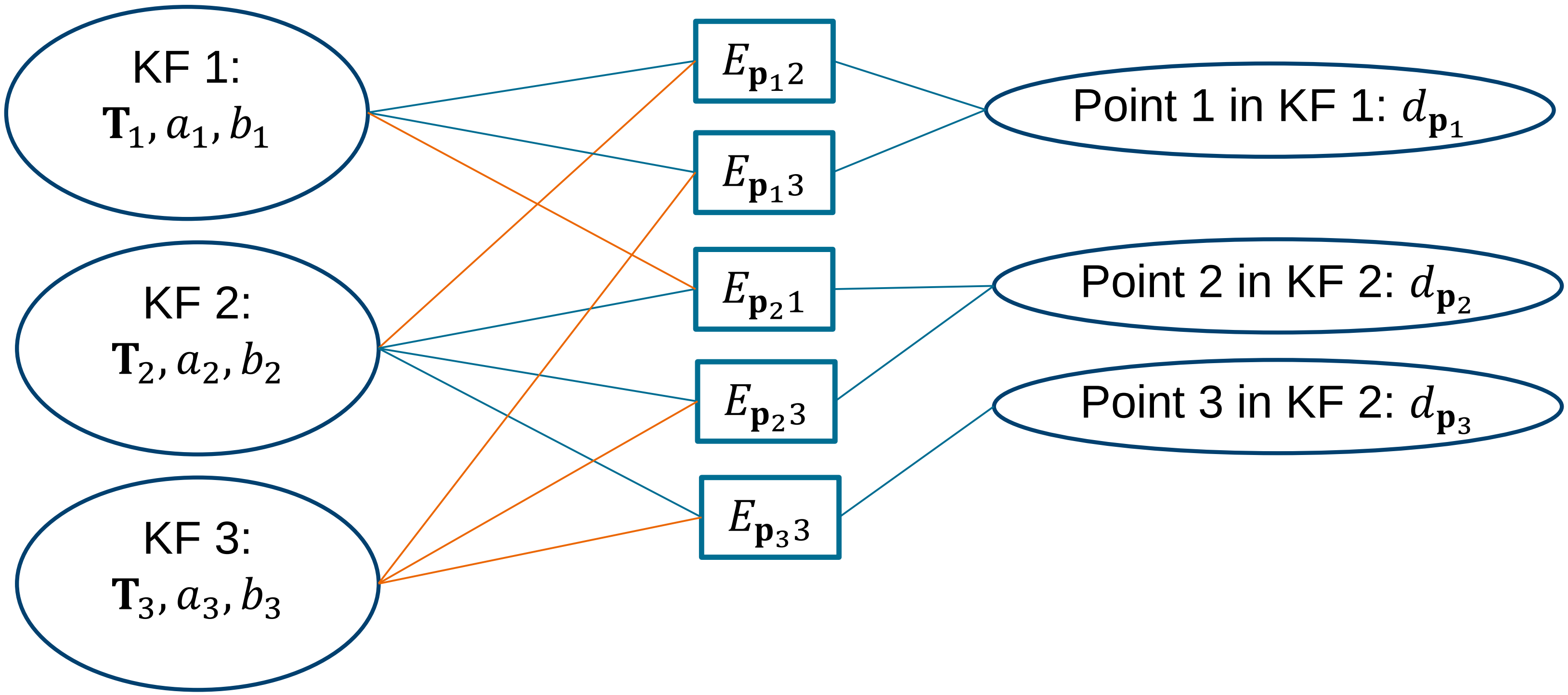
$$E_{\mathbf{p}j} := \sum_{\mathbf{p}\in\mathcal{N}_\mathbf{p}} w_p \left\| \left(I_j[\mathbf{p}'] - b_j\right) - \frac{t_j e^{a_j}}{t_i e^{a_i}} \left(I_i[\mathbf{p}] - b_i\right) \right\|_\gamma$$

$$\boxed{E_{\text{photo}} := \sum_{i\in\mathcal{F}} \sum_{\mathbf{p}\in\mathcal{P}_i} \sum_{j\in\text{obs}(\mathbf{p})} E_{\mathbf{p}j}}$$

all frames

host frame i's points

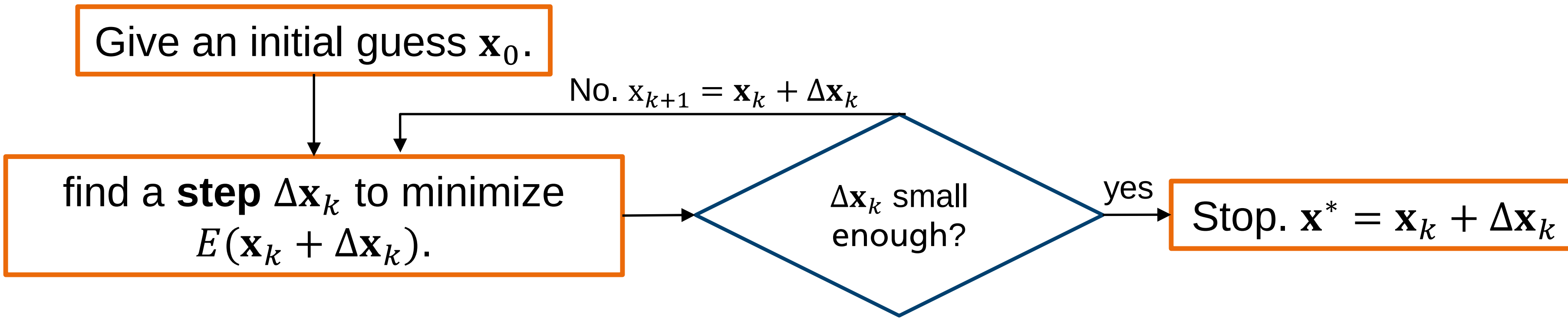
other frames observing $\mathbf{p}$



Windowed Optimization

Least-Squares Minimization [Triggs-99]

- **Example**: $\mathbf{x}^* \coloneqq \arg\min_{\mathbf{x}} (\frac{1}{2}(x_1 - 1)^2 + \frac{1}{2}(x_2 - 1)^2)$, where $\mathbf{x} = [x_1 \quad x_2]^{\mathbf{T}}$.
- **Task**: Find $\mathbf{x}^* \coloneqq \arg\min_{\mathbf{x}} E(\mathbf{x})$, where $E(\mathbf{x}) = \frac{1}{2}\mathbf{e}(\mathbf{x})^{\mathbf{T}}\mathbf{W}\mathbf{e}(\mathbf{x})$.
- **Intuition**: Solve $\frac{dE(\mathbf{x})}{d\mathbf{x}} = \mathbf{0}$. → (sometimes) very complex and complicated.
- **Iterative Estimation Method**



Windowed Optimization
Least-Squares Minimization

- **New Task**: Find $\Delta\mathbf{x}_k \coloneqq \arg\min_{\Delta\mathbf{x}} E(\mathbf{x}_k + \Delta\mathbf{x})$, where $E(\mathbf{x}) = \frac{1}{2}\mathbf{e}(\mathbf{x})^{\mathbf{T}}\mathbf{W}\mathbf{e}(\mathbf{x})$.

$$\boxed{\mathbf{g}(\mathbf{x}_k)^{\mathbf{T}} = \left.\frac{dE(\mathbf{x})}{d\mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}_k},\ \mathbf{H}(\mathbf{x}_k) = \left.\frac{d^2E(\mathbf{x})}{d\mathbf{x}^2}\right|_{\mathbf{x}=\mathbf{x}_k}}$$

- First order (***gradient descent***):

$$E(\mathbf{x}_k + \Delta\mathbf{x}) \approx E(\mathbf{x}_k) + \mathbf{g}(\mathbf{x}_k)^{\mathbf{T}}\Delta\mathbf{x},$$

→ The **direction** of $\Delta\mathbf{x}$: $-\mathbf{g}(\mathbf{x}_k)$.

- Second order (***Newton's method***):

$$E(\mathbf{x}_k + \Delta\mathbf{x}) \approx E(\mathbf{x}_k) + \mathbf{g}(\mathbf{x}_k)^{\mathbf{T}}\Delta\mathbf{x} + \frac{1}{2}\Delta\mathbf{x}^{\mathbf{T}}\mathbf{H}(\mathbf{x}_k)\Delta\mathbf{x},$$

→ $\frac{dE(\mathbf{x}_k+\Delta\mathbf{x})}{d\Delta\mathbf{x}} = 0$

$$\frac{dE(\mathbf{x}_k+\Delta\mathbf{x})}{d\Delta\mathbf{x}} = \mathbf{g}(\mathbf{x}_k)^{\mathbf{T}} + \Delta\mathbf{x}^{\mathbf{T}}\mathbf{H}(\mathbf{x}_k) = \mathbf{0} \Longrightarrow \Delta\mathbf{x} = -\mathbf{H}(\mathbf{x}_k)^{-\mathbf{1}}\mathbf{g}(\mathbf{x}_k)$$

Windowed Optimization
Least-Squares Minimization

- **Taylor expansion**:

$$E(\mathbf{x}_k + \Delta\mathbf{x}) \approx E(\mathbf{x}_k) + \mathbf{g}(\mathbf{x}_k)^{\mathbf{T}}\Delta\mathbf{x} + \frac{1}{2}\Delta\mathbf{x}^{\mathbf{T}}\mathbf{H}(\mathbf{x}_k)\Delta\mathbf{x}$$

$$\mathbf{J}(\mathbf{x}_k) = \left.\frac{d\mathbf{e}(\mathbf{x})}{d\mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}_k}$$

$$\mathbf{g}(\mathbf{x}_k)^{\mathbf{T}} = \left.\frac{dE(\mathbf{x})}{d\mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}_k} = \left.\frac{dE(\mathbf{x})}{d\mathbf{e}(\mathbf{x})}\frac{d\mathbf{e}(\mathbf{x})}{d\mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}_k} = \mathbf{e}(\mathbf{x}_k)^{\mathbf{T}}\mathbf{W}\mathbf{J}(\mathbf{x}_k),$$

$$\mathbf{H}(\mathbf{x}_k) = \left.\frac{d^2E(\mathbf{x})}{d\mathbf{x}^2}\right|_{\mathbf{x}=\mathbf{x}_k} = \frac{\mathrm{d}}{\mathrm{dx}}\left.\left(\mathbf{e}(\mathbf{x})^{\mathbf{T}}\mathbf{W}\mathbf{J}(\mathbf{x})\right)\right|_{\mathbf{x}=\mathbf{x}_k} = \mathbf{J}(\mathbf{x}_k)^{\mathbf{T}}\mathbf{W}\mathbf{J}(\mathbf{x}_k) + \mathbf{e}(\mathbf{x}_\mathbf{k})^{\mathbf{T}}\mathbf{W}\underbrace{\left.\frac{d^2\mathbf{e}(\mathbf{x})}{d\mathbf{x}^2}\right|_{\mathbf{x}=\mathbf{x}_k}}_{=\mathbf{0},\text{ when linear}}$$

- ***Gauss-Newton's method***:

$$\mathbf{H}(\mathbf{x}_k) \approx \mathbf{J}(\mathbf{x}_k)^{\mathbf{T}}\mathbf{W}\mathbf{J}(\mathbf{x}_k)$$

$$\Longrightarrow \mathbf{H}(\mathbf{x}_k)\Delta\mathbf{x} = -\mathbf{g}(\mathbf{x}_k)$$

Windowed Optimization
Least-Squares Minimization

- **Example**: $\mathbf{x}^* := \arg\min_{\mathbf{x}}(\frac{1}{2}(x_1 - 1)^2 + \frac{1}{2}(x_2 - 1)^2)$, where $\mathbf{x} = [x_1 \quad x_2]^{\mathbf{T}}$.
- $\mathbf{x}^* := \arg\min_{\mathbf{x}} \frac{1}{2}\mathbf{e}(\mathbf{x})^{\mathbf{T}}\mathbf{W}\mathbf{e}(\mathbf{x})$, where $\mathbf{e}(\mathbf{x}) = [x_1 - 1 \quad x_2 - 1]^{\mathbf{T}}$, $\mathbf{W} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$
- **Procedure**
  - Step 1: Initial guess $\mathbf{x}_0 = [2 \quad 2]^{\mathbf{T}}$.
  - Step 2: $\mathbf{J}(\mathbf{x}_0) = \frac{d\mathbf{e}(\mathbf{x})}{d\mathbf{x}}\Big|_{\mathbf{x}=\mathbf{x}_0} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, $\mathbf{g}(\mathbf{x}_0)^{\mathbf{T}} = \mathbf{e}(\mathbf{x}_0)^{\mathbf{T}}\mathbf{W}\mathbf{J}(\mathbf{x}_0) = [1 \quad 1]$,

    $\mathbf{H}(\mathbf{x}_0) \approx \mathbf{J}(\mathbf{x}_0)^{\mathbf{T}}\mathbf{W}\mathbf{J}(\mathbf{x}_0) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, $\Delta\mathbf{x}_0 = -\mathbf{H}(\mathbf{x}_0)^{-1}\mathbf{g}(\mathbf{x}_0) = \begin{bmatrix} -1 \\ -1 \end{bmatrix}$
  - Step 3: $E(\mathbf{x}_0 + \Delta\mathbf{x}_0) = 0 \implies$ Stop.

Windowed Optimization
Sliding Window

- Minimize E_{photo} in a sliding window ($N_{KF} = 7$) ~ Gauss-Newton (L-M).
- $E_{\text{photo}} \coloneqq \sum_{i \in \mathcal{F}} \sum_{\mathbf{p} \in \mathcal{P}_i} \sum_{j \in \text{obs}(\mathbf{p})} E_{\mathbf{p}j}$,

  where $E_{\mathbf{p}j} \coloneqq \sum_{\mathbf{p} \in \mathcal{N}_{\mathbf{p}}} w_p \left\| \left(I_j[\mathbf{p}'] - b_j\right) - \frac{t_j e^{a_j}}{t_i e^{a_i}} \left(I_i[\mathbf{p}] - b_i\right) \right\|_\gamma$
- Optimized variables $\boldsymbol{\zeta}$

  Geometric: camera poses, inverse depth values, camera intrinsics.

  Photometric: affine brightness parameters.
- **One residual** $r_m = \left(I_j\left[\mathbf{p}'(\mathbf{T}_\mathrm{i}, \mathbf{T}_\mathrm{j}, \mathrm{d}, \mathbf{c})\right] - b_j\right) - \frac{t_j e^{a_j}}{t_i e^{a_i}} \left(I_i[\mathbf{p}] - b_i\right)$

  → Related variables: $\underbrace{\mathbf{T}_\mathrm{i}, \mathbf{T}_\mathrm{j}, \mathrm{d}, \mathbf{c}}_{\text{geo.}}, \underbrace{a_i, a_j, b_i, b_j}_{\text{photo.}}$.

Windowed Optimization
Naive Formulation

Initial value	$\mathbf{x}_0$	$\boldsymbol{\zeta}_0$
current value	$\mathbf{x}_k$	$\boldsymbol{\zeta}_k$
step	$\Delta\mathbf{x}$	$\boldsymbol{\delta}$
new value	$\Delta\mathbf{x} + \mathbf{x}_k$	$\boldsymbol{\delta} + \boldsymbol{\zeta}_k$
single residual	$e_m(\mathbf{x}_k)$	$r_m(\boldsymbol{\zeta}_k)$
Jacobian	$\mathbf{J}_m(\mathbf{x}_k) = \left.\frac{de_m(\mathbf{x})}{d\mathbf{x}}\right\rvert_{\mathbf{x}=\mathbf{x}_k}$	$\mathbf{J}_m(\boldsymbol{\zeta}_k) = \left.\frac{\partial r_m(\boldsymbol{\zeta})}{\partial \boldsymbol{\zeta}}\right\rvert_{\boldsymbol{\zeta}=\boldsymbol{\zeta}_k}$

- $\mathbf{J}_m = \left.\frac{\partial r_m(\boldsymbol{\zeta})}{\partial \boldsymbol{\zeta}}\right\rvert_{\boldsymbol{\zeta}=\boldsymbol{\zeta}_k} = \left.\left[\frac{\partial r_m(\boldsymbol{\zeta})}{\partial \mathbf{p}'}\frac{\partial \mathbf{p}'}{\partial \boldsymbol{\zeta}_{\text{geo}}}, \frac{\partial r_m(\boldsymbol{\zeta})}{\partial \boldsymbol{\zeta}_{\text{photo}}}\right]\right\rvert_{\boldsymbol{\zeta}=\boldsymbol{\zeta}_k}$

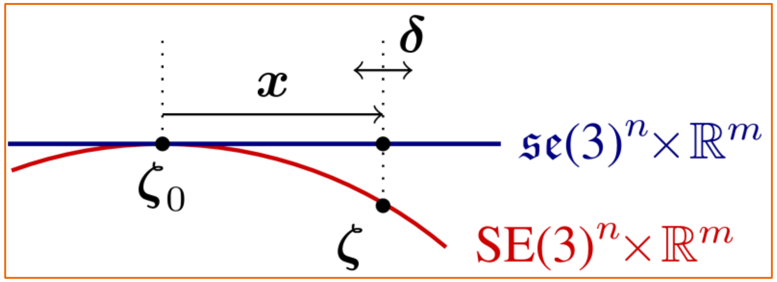
$$= \left[\left.\frac{\partial I_j}{\partial \mathbf{p}'}\right\rvert_{\boldsymbol{\zeta}=\boldsymbol{\zeta}_k} \left.\frac{\partial \mathbf{p}'}{\partial \boldsymbol{\zeta}_{\text{geo}}}\right\rvert_{\boldsymbol{\zeta}=\boldsymbol{\zeta}_k}, \left.\frac{\partial r_m(\boldsymbol{\zeta})}{\partial \boldsymbol{\zeta}_{\text{photo}}}\right\rvert_{\boldsymbol{\zeta}=\boldsymbol{\zeta}_k}\right]$$

$$\boxed{r_m = \left(I_j[\mathbf{p}'(\mathbf{T}_\mathrm{i}, \mathbf{T}_\mathrm{j}, \mathrm{d}, \mathbf{c})] - b_j\right) - \frac{t_j e^{a_j}}{t_i e^{a_i}}\left(I_i[\mathbf{p}] - b_i\right)}$$

Windowed Optimization
Better Formulation

Initial value	$\mathbf{x}_0$	$\boldsymbol{\zeta}_0$
current value	$\mathbf{x}_k$	$\boldsymbol{\zeta}_k = \mathbf{x} \boxplus \boldsymbol{\zeta}_0$
step	$\Delta\mathbf{x}$	$\boldsymbol{\delta}$
new value	$\Delta\mathbf{x} + \mathbf{x}_k$	$(\boldsymbol{\delta} + \mathbf{x}) \boxplus \boldsymbol{\zeta}_0$
single residual	$e_m(\mathbf{x}_k)$	$r_m(\mathbf{x} \boxplus \boldsymbol{\zeta}_0)$
Jacobian	$\mathrm{J}_m(\mathbf{x}_k) = \left.\frac{de_m(\mathbf{x})}{d\mathbf{x}}\right\rvert_{\mathbf{x}=\mathbf{x}_k}$	$\mathrm{J}_m = \frac{\partial r_m((\boldsymbol{\delta} + \mathbf{x}) \boxplus \boldsymbol{\zeta}_0)}{\partial \boldsymbol{\delta}}$

- $\mathbf{x}, \boldsymbol{\delta} \in \mathfrak{se}(3)^n \times \mathbb{R}^m$, $\boldsymbol{\zeta}_0, \boldsymbol{\zeta} \in \mathbf{SE}(3)^n \times \mathbb{R}^m$
- $\mathbf{x}$: accumulated updates $\sum \boldsymbol{\delta}$



$$r_m = \left(I_j[\mathbf{p}'(\mathrm{T_i}, \mathrm{T_j}, \mathrm{d}, \mathbf{c})] - b_j\right) - \frac{t_j e^{a_j}}{t_i e^{a_i}}\left(I_i[\mathbf{p}] - b_i\right)$$

Windowed Optimization

Better Formulation

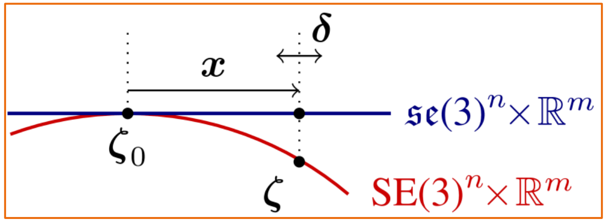
Initial value	$\mathbf{x}_0$	$\boldsymbol{\zeta}_0$
current value	$\mathbf{x}_k$	$\boldsymbol{\zeta}_k = \mathbf{x} \boxplus \boldsymbol{\zeta}_0$
step	$\Delta\mathbf{x}$	$\boldsymbol{\delta}$
new value	$\Delta\mathbf{x} + \mathbf{x}_k$	$(\boldsymbol{\delta} + \mathbf{x}) \boxplus \boldsymbol{\zeta}_0$
single residual	$e_m(\mathbf{x}_k)$	$r_m(\mathbf{x} \boxplus \boldsymbol{\zeta}_0)$
Jacobian	$\mathbf{J}_m(\mathbf{x}_k) = \left.\frac{de_m(\mathbf{x})}{d\mathbf{x}}\right\rvert_{\mathbf{x}=\mathbf{x}_k}$	$\mathbf{J}_m = \frac{\partial r_m((\boldsymbol{\delta} + \mathbf{x}) \boxplus \boldsymbol{\zeta}_0)}{\partial\boldsymbol{\delta}}$

- $$\mathbf{J}_m = \frac{\partial r_m((\boldsymbol{\delta}+\mathbf{x})\boxplus\boldsymbol{\zeta}_0)}{\partial\boldsymbol{\delta}}$$
$$= \left[\frac{\partial r_m((\boldsymbol{\delta}+\mathbf{x})\boxplus\boldsymbol{\zeta}_0)}{\partial\mathbf{p}'}\frac{\partial\mathbf{p}'}{\partial\boldsymbol{\delta}_{\text{geo}}}, \frac{\partial r_m((\boldsymbol{\delta}+\mathbf{x})\boxplus\boldsymbol{\zeta}_0)}{\partial\boldsymbol{\delta}_{\text{photo}}}\right]$$
$$= \left[\left.\frac{\partial I_j}{\partial\mathbf{p}'}\right\rvert_{\boldsymbol{\delta}=\mathbf{0}} \left.\frac{\partial\mathbf{p}'((\boldsymbol{\delta}+\mathbf{x})\boxplus\boldsymbol{\zeta}_0)}{\partial\boldsymbol{\delta}_{\text{geo}}}\right\rvert_{\boldsymbol{\delta}=\mathbf{0}}, \left.\frac{\partial r_m((\boldsymbol{\delta}+\mathbf{x})\boxplus\boldsymbol{\zeta}_0}{\partial\boldsymbol{\delta}_{\text{photo}}}\right\rvert_{\boldsymbol{\delta}=\mathbf{0}}\right]$$

$$r_m = \left(I_j\left[\mathbf{p}'(\mathbf{T}_{\mathrm{i}}, \mathbf{T}_{\mathrm{j}}, \mathrm{d}, \mathbf{c})\right] - b_j\right) - \frac{t_j e^{a_j}}{t_i e^{a_i}}\left(I_i[\mathbf{p}] - b_i\right)$$

Windowed Optimization

First-Estimates-Jacobian Formulation



- $\mathbf{x}, \boldsymbol{\delta} \in \mathfrak{se}(3)^n \times \mathbb{R}^m, \boldsymbol{\zeta}_0, \boldsymbol{\zeta} \in \mathbf{SE}(3) \times \mathbb{R}^m$
- $\mathbf{J}_m = \frac{\partial r_m((\boldsymbol{\delta}+\mathbf{x})\boxplus\boldsymbol{\zeta}_0)}{\partial \boldsymbol{\delta}}$

$$= \left[\frac{\partial r_m((\boldsymbol{\delta}+\mathbf{x})\boxplus\boldsymbol{\zeta}_0)}{\partial \mathbf{p}'} \frac{\partial \mathbf{p}'((\boldsymbol{\delta}+\mathbf{x})\boxplus\boldsymbol{\zeta}_0)}{\partial \boldsymbol{\delta}_{\mathrm{geo}}}, \frac{\partial r_m((\boldsymbol{\delta}+\mathbf{x})\boxplus\boldsymbol{\zeta}_0)}{\partial \boldsymbol{\delta}_{\mathrm{photo}}}\right]$$

$$= [\underbrace{\left.\frac{\partial I_j}{\partial \mathbf{p}'}\right|_{\boldsymbol{\delta}=\mathbf{0}}}_{\mathbf{J}_I} \underbrace{\left.\frac{\partial \mathbf{p}'((\boldsymbol{\delta}+\mathbf{x})\boxplus\boldsymbol{\zeta}_0)}{\partial \boldsymbol{\delta}_{\mathrm{geo}}}\right|_{\boldsymbol{\delta},\mathbf{x}=\mathbf{0}}}_{\mathbf{J}_{\mathrm{geo}}}, \underbrace{\left.\frac{\partial r_m((\boldsymbol{\delta}+\mathbf{x})\boxplus\boldsymbol{\zeta}_0}{\partial \boldsymbol{\delta}_{\mathrm{photo}}}\right|_{\boldsymbol{\delta},\mathbf{x}=\mathbf{0}}}_{\mathbf{J}_{\mathrm{photo}}}]$$

- Update $\mathbf{x}$ after every iteration: $\boldsymbol{\delta} = -\mathbf{H}^{-1}\mathbf{g} \Rightarrow \mathbf{x}^{\mathrm{new}} \coloneqq \boldsymbol{\delta} + \mathbf{x}$
- Update $\boldsymbol{\zeta}$ after optimization:

remaining variables of $\boldsymbol{\zeta}$

$$\boldsymbol{\zeta}^{\mathrm{r}} \coloneqq \mathbf{x}^{\mathrm{r}} \boxplus \boldsymbol{\zeta}_0^{\mathrm{r}}$$

$$\downarrow$$

$$\boldsymbol{\zeta}_0^{\mathrm{new}}$$

marginalized variables of $\boldsymbol{\zeta}$

Marginalization of $\boldsymbol{\zeta}^{\mathrm{m}}$

Windowed Optimization
Marginalization

- **Motivation**: Restrain the computational costs of optimization.
- **Goal**: optimized variables' number $\searrow$
- **Example**: $E(\mathbf{x}) = \frac{1}{2}(x_1 - x_2)^2 + \frac{1}{2}(x_2 - 2x_3)^2$, where $\mathbf{x} = [x_1 \quad x_2 \quad x_3]^{\mathbf{T}}$.

  Last guess $\mathbf{x}_k = [2 \quad 2 \quad 2]^{\mathbf{T}}$.
- **Goal:** Marginalize $x_1 \Leftrightarrow$ Replace $E(\mathbf{x})$ with a $E^{\text{new}}(x_2, x_3)$

Windowed Optimization

Marginalization – Probabilistic Model [Thrun-05]

- **Given**: $P(\mathbf{x}^\mathrm{r}, \mathbf{x}^\mathrm{m}) = N\left(\begin{bmatrix}\boldsymbol{\mu}^\mathrm{r} \\ \boldsymbol{\mu}^\mathrm{m}\end{bmatrix}, \begin{bmatrix}\boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22}\end{bmatrix}\right) = N^{-1}\left(\begin{bmatrix}\boldsymbol{\xi}^\mathrm{r} \\ \boldsymbol{\xi}^\mathrm{m}\end{bmatrix}, \begin{bmatrix}\mathbf{H}_{11} & \mathbf{H}_{12} \\ \mathbf{H}_{21} & \mathbf{H}_{22}\end{bmatrix}\right)$

$$\mathbf{H}^{-1} = \begin{bmatrix}\mathbf{H}_{11} & \mathbf{H}_{12} \\ \mathbf{H}_{21} & \mathbf{H}_{22}\end{bmatrix}^{-1} \xRightarrow{\text{Schur complement}} \begin{bmatrix}\left(\mathbf{H}_{11} - \mathbf{H}_{12}^\mathrm{T}\mathbf{H}_{22}^{-1}\mathbf{H}_{21}\right)^{-1} & \dots \\ \dots & \dots\end{bmatrix}.$$

- Marginalize $\mathbf{x}^\mathrm{m} \Longrightarrow P(\mathbf{x}^\mathrm{r}) = N(\boldsymbol{\mu}^r, \boldsymbol{\Sigma}_{11}) = N^{-1}(\boldsymbol{\xi}^\mathrm{new}, \mathbf{H}^\mathrm{new})$,

- Calculation based on the information form:

$$\begin{bmatrix}\boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22}\end{bmatrix} = \begin{bmatrix}\mathbf{H}_{11} & \mathbf{H}_{12} \\ \mathbf{H}_{21} & \mathbf{H}_{22}\end{bmatrix}^{-1} = \begin{bmatrix}\left(\mathbf{H}_{11} - \mathbf{H}_{12}^\mathrm{T}\mathbf{H}_{22}^{-1}\mathbf{H}_{21}\right)^{-1} & \dots \\ \dots & \dots\end{bmatrix}$$

$$\mathbf{H}^\mathrm{new} = \boldsymbol{\Sigma}_{11}^{-1} = \mathbf{H}_{11} - \mathbf{H}_{12}^\mathrm{T}\mathbf{H}_{22}^{-1}\mathbf{H}_{21}$$

$$\boldsymbol{\xi}^\mathrm{new} = \boldsymbol{\Sigma}_{11}^{-1}\boldsymbol{\mu}^r = \boldsymbol{\xi}^\mathrm{r} - \mathbf{H}_{12}^\mathrm{T}\mathbf{H}_{22}^{-1}\boldsymbol{\xi}^\mathrm{m}$$

Windowed Optimization
Marginalization – Cost Function

- **Task**: Marginalize $\mathbf{x}^{\mathrm{m}}$, where $E'(\mathbf{x}) = \frac{1}{2}\mathbf{e}'(\mathbf{x})^{\mathbf{T}}\mathbf{W}'\mathbf{e}'(\mathbf{x})$, $\mathbf{x} = [\mathbf{x}^{\mathrm{r}} \quad \mathbf{x}^{\mathrm{m}}]^{\mathbf{T}}$,

  → Replace $E'(\mathbf{x})$ with a $E'^{\mathrm{new}}(\mathbf{x}^{\mathrm{r}})$ based on last guess $\mathbf{x}_k$.

residuals which depend on $\mathbf{x}^{\mathrm{m}}$

- $E'(\mathbf{x}) = E'(\mathbf{x}_k + (\mathbf{x} - \mathbf{x}_k))$

$$\approx E'(\mathbf{x}_k) + \mathbf{g}(\mathbf{x}_k)^{\mathbf{T}}(\mathbf{x} - \mathbf{x}_k) + \frac{1}{2}(\mathbf{x} - \mathbf{x}_k)^{\mathbf{T}}\mathbf{H}(\mathbf{x}_k)(\mathbf{x} - \mathbf{x}_k)$$

$$= E'(\mathbf{x}_k) + \mathbf{g}(\mathbf{x}_k)^{\mathbf{T}}\mathbf{x} - \mathbf{g}(\mathbf{x}_k)^{\mathbf{T}}\mathbf{x}_k + \frac{1}{2}\mathbf{x}^{\mathbf{T}}\mathbf{H}(\mathbf{x}_k)\mathbf{x} + \frac{1}{2}\mathbf{x}_k^{\mathbf{T}}\mathbf{H}(\mathbf{x}_k)\mathbf{x}_k - \mathbf{x}_k^{\mathbf{T}}\mathbf{H}(\mathbf{x}_k)\mathbf{x}$$

$$= \underbrace{\left(E'(\mathbf{x}_k) - \mathbf{g}(\mathbf{x}_k)^{\mathbf{T}}\mathbf{x}_k + \frac{1}{2}\mathbf{x}_k^{\mathbf{T}}\mathbf{H}(\mathbf{x}_k)\mathbf{x}_k\right)}_{\text{const.}} + \underbrace{\left(\mathbf{g}(\mathbf{x}_k)^{\mathbf{T}}\mathbf{x} - \mathbf{x}_k^{\mathbf{T}}\mathbf{H}(\mathbf{x}_k)\mathbf{x}\right)}_{\underbrace{(\mathbf{g}(\mathbf{x}_k)+\mathbf{H}(\mathbf{x}_k)\mathbf{x}_k)^{\mathbf{T}}\mathbf{x}}_{\mathbf{g}'(\mathbf{x}_k)^{\mathbf{T}}\mathbf{x}}} + \frac{1}{2}\mathbf{x}^{\mathbf{T}}\mathbf{H}(\mathbf{x}_k)\mathbf{x}$$

$$= \text{const.} + \mathbf{g}'(\mathbf{x}_k)^{\mathbf{T}}\mathbf{x} + \frac{1}{2}\mathbf{x}^{\mathbf{T}}\mathbf{H}(\mathbf{x}_k)\mathbf{x}$$

Windowed Optimization
Marginalization – Cost Function

- **Task**: Marginalize $\mathbf{x}^{\mathrm{m}}$, where $E'(\mathbf{x}) = \frac{1}{2}\mathbf{e}'(\mathbf{x})^{\mathbf{T}}\mathbf{W}'\mathbf{e}'(\mathbf{x})$, $\mathbf{x} = [\mathbf{x}^{\mathrm{r}} \quad \mathbf{x}^{\mathrm{m}}]^{\mathbf{T}}$,

  → Replace $E'(\mathbf{x})$ with a $E'^{\mathrm{new}}(\mathbf{x}^{\mathrm{r}})$ based on last guess $\mathbf{x}_k$.

- $E'(\mathbf{x}) = \text{ const.} + \mathbf{g}'(\mathbf{x}_k)^{\mathbf{T}}\mathbf{x} + \frac{1}{2}\mathbf{x}^{\mathbf{T}}\mathbf{H}(\mathbf{x}_k)\mathbf{x}$

$$\begin{bmatrix}\boldsymbol{\xi}^{\mathrm{r}} \\ \boldsymbol{\xi}^{\mathrm{m}}\end{bmatrix} = -\mathbf{g}'(\mathbf{x}_k) = \begin{bmatrix}-\mathbf{g}'_1 \\ -\mathbf{g}'_2\end{bmatrix} \qquad \mathbf{H}(\mathbf{x}_k) = \begin{bmatrix}\mathbf{H}_{11} & \mathbf{H}_{12} \\ \mathbf{H}_{21} & \mathbf{H}_{22}\end{bmatrix}$$

$$\mathbf{H}^{\mathrm{new}} = \mathbf{H}_{11} - \mathbf{H}_{12}\mathbf{H}_{22}^{-1}\mathbf{H}_{21}, \quad \mathbf{g}'^{\mathrm{new}} = -\boldsymbol{\xi}^{\mathrm{new}} = -(\boldsymbol{\xi}^{\mathrm{r}} - \mathbf{H}_{12}^{\mathbf{T}}\mathbf{H}_{22}^{-1}\boldsymbol{\xi}^{\mathrm{m}})$$

$$E'^{\mathrm{new}}(\mathbf{x}^{\mathrm{r}}) = \mathbf{g}'^{\mathrm{new}^{\mathrm{T}}}\mathbf{x}^{\mathrm{r}} + \frac{1}{2}\mathbf{x}^{\mathrm{r}\mathbf{T}}\mathbf{H}^{\mathrm{new}}\mathbf{x}^{\mathrm{r}}$$

Jacobian $\mathbf{J}$ is evaluated at $\mathbf{x}_k^{\mathrm{r}}$, $\mathbf{x}_k^{\mathrm{m}}$!

Windowed Optimization

First Estimates Jacobian [Huang-08] [Dong-Si-12]

- **Optimization after marginalization**

$$\mathbf{H}(\mathbf{x}_0)\,\Delta\mathbf{x} = -\mathbf{g}(\mathbf{x}_0)$$

$$\mathbf{H}(\mathbf{x}_1)\,\Delta\mathbf{x} = -\mathbf{g}(\mathbf{x}_1)$$

$$\vdots$$

$$\mathbf{H}(\mathbf{x}_{k'})\,\Delta\mathbf{x} = -\mathbf{g}(\mathbf{x}_{k'})$$

Some of $\mathbf{J}$ at old states $\mathbf{x}_k^{\mathrm{r}}$, $\mathbf{x}_k^{\mathrm{m}}$,
some at new states $\mathbf{x}_{k'}^{\mathrm{r}}$, $\mathbf{x}_{k'}^{\mathrm{new}}$

- **Inconsistency**
  - Change of observability (Reduction of null spaces)

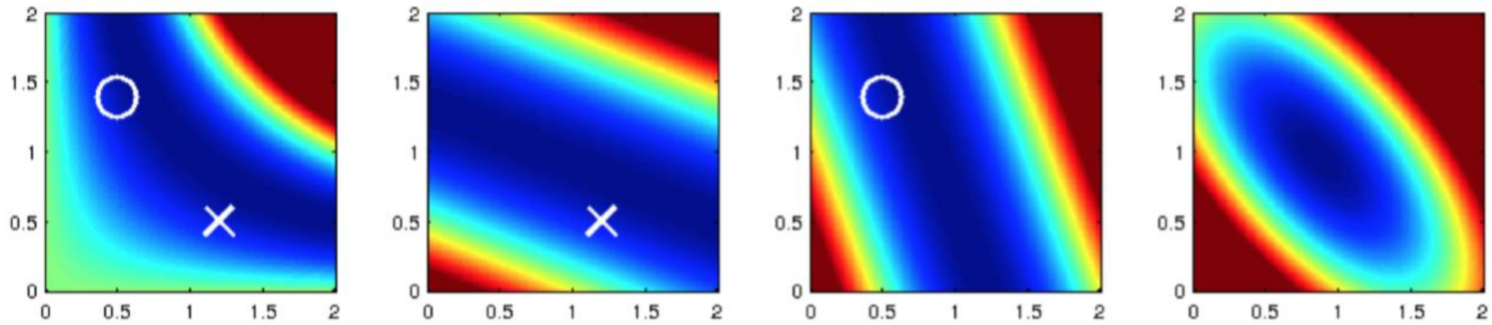
- **Null space $\mathbf{n}$**

$$\mathbf{H}\,\mathbf{n} = \mathbf{0}$$

$$\rightarrow \mathbf{H}(\mathbf{x}_k)\,(\Delta\mathbf{x} + \mathbf{n}) = -\mathbf{g}(\mathbf{x}_k)$$

Unobservable Degrees of Freedom:
- Absolute position
- Absolute orientation
- Scale

Windowed Optimization
First Estimates Jacobian



$$E = E_1 + E_2$$
$$= (xy-1)^2 + (xy-1)^2$$

$$E_1' = E_1$$
Lin. around (0.5, 1.4)

$$E_2' = E_2$$
Lin. around (1.2, 0.5)

$$E' = E_1' + E_2'$$

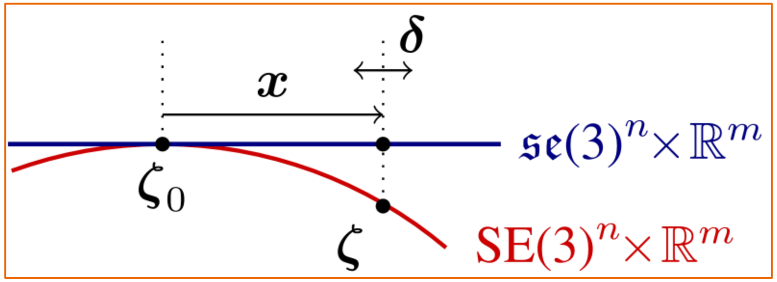
Nullspace disappears!

[Courtesy: J. Engel]

Never combine linearizations around different points!

Windowed Optimization
First Estimates Jacobian

- $\mathbf{x}, \boldsymbol{\delta} \in \mathfrak{se}(3)^n \times \mathbb{R}^m$, $\boldsymbol{\zeta}_0, \boldsymbol{\zeta} \in \mathbf{SE}(3) \times \mathbb{R}^m$



- $\mathbf{J}_m = \frac{\partial r_m((\boldsymbol{\delta}+\mathbf{x}) \boxplus \boldsymbol{\zeta}_0)}{\partial \boldsymbol{\delta}}$

$$= \left[\frac{\partial r_m((\boldsymbol{\delta}+\mathbf{x}) \boxplus \boldsymbol{\zeta}_0)}{\partial \mathbf{p}'} \frac{\partial \mathbf{p}'}{\partial \boldsymbol{\delta}_{\text{geo}}}, \frac{\partial r_m((\boldsymbol{\delta}+\mathbf{x}) \boxplus \boldsymbol{\zeta}_0)}{\partial \boldsymbol{\delta}_{\text{photo}}}\right]$$

$$= [\underbrace{\frac{\partial I_j}{\partial \mathbf{p}'}\bigg|_{\boldsymbol{\delta}=\mathbf{0}}}_{\mathbf{J}_I} \underbrace{\frac{\partial \mathbf{p}'}{\partial \boldsymbol{\delta}_{\text{geo}}}\bigg|_{\boldsymbol{\delta},\mathbf{x}=\mathbf{0}}}_{\mathbf{J}_{\text{geo}}}, \underbrace{\frac{\partial r_m((\boldsymbol{\delta}+\mathbf{x}) \boxplus \boldsymbol{\zeta}_0}{\partial \boldsymbol{\delta}_{\text{photo}}}\bigg|_{\boldsymbol{\delta},\mathbf{x}=\mathbf{0}}}_{\mathbf{J}_{\text{photo}}}]$$

- Advantages of FEJ
  - Consistency
  - Smaller computational costs

Agenda

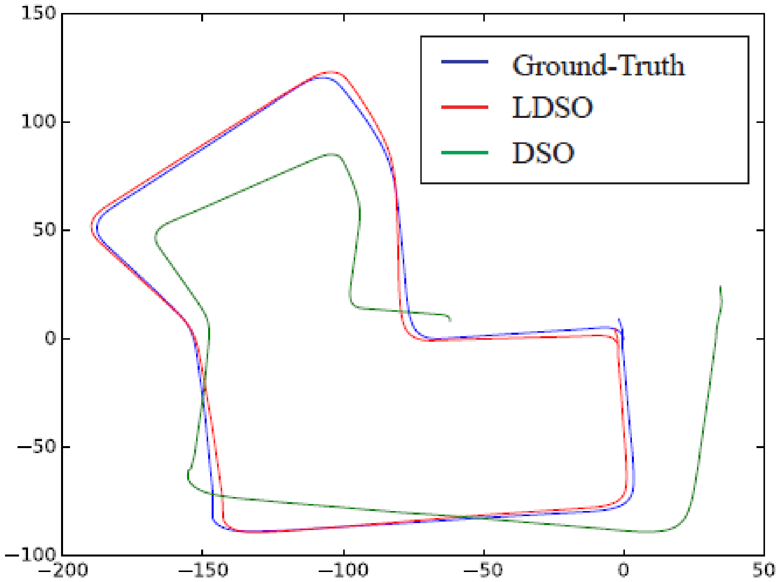
1. Introduction
2. Direct Sparse Odometry
3. Direct Sparse Odometry with Loop Closure
   - Motivation
   - Loop Detection
   - Pose Graph Optimization

Motivation

- Inevitable accumulated drift in DSO

  → Modifications in LDSO:

  - DSO: high-gradient pixels. → LDSO: high-gradient + corner pixels
  - Add loop closure



Aligned trajectories of KITTI sequence 07 [Gao-18]

Loop Detection
Bag of Words

- **Core**: Check the similarity of two images based on a score.
- **Technique**: Bag of Words (BoW) [Gálvez-López-12]

  → Describe an image using words in a dictionary.

- **Naive example**: Given a dictionary = $\{chair, table, book\}$

  Image 1 has a $chair$ and a $table$. Image 2 has a $chair$ and a $book$.

  $\mathbf{v}_1 = 1 \cdot chair + 1 \cdot table + 0 \cdot book$

  $\mathbf{v}_2 = 1 \cdot chair + 0 \cdot table + 1 \cdot book$

  $s(\mathbf{v}_1, \mathbf{v}_2) = \|\mathbf{v}_1 - \mathbf{v}_2\|_{\text{Hamming}}$

Loop Detection
Bag of Words

- **Building BoW models in LDSO**

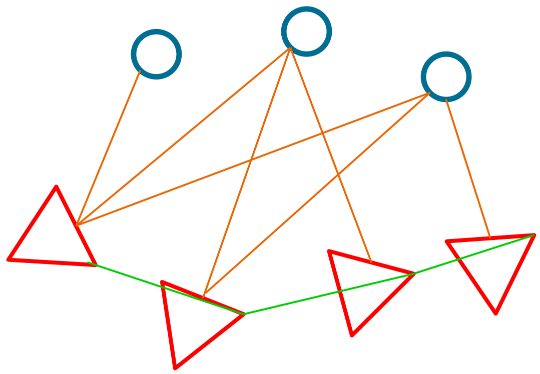
Load a dictionary created before → Pick Shi-Tomasi corner in image → Compute ORB descriptors → Build bag-of-words vectors

- **Measurement of similarity**

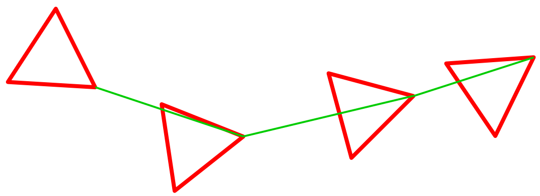
E.g. $L1$-score: $s(\mathbf{v}_1, \mathbf{v}_2) = 1 - \frac{1}{2}\left|\frac{\mathbf{v}_1}{|\mathbf{v}_1|} - \frac{\mathbf{v}_2}{|\mathbf{v}_2|}\right|$

Pose Graph Optimization

- Bundle adjustment vs. Pose graph optimization



Bundle adjustment



Pose graph

Vertex – Landmark location

Vertex – Camera pose

Edge – Landmark-based Measurement

Edge – Odometry-based Measurement

Summary

- Direct Sparse Odometry (DSO)
  - Photometric model and error
  - Optimization (Information filter)

- Direct Sparse Odometry with Loop Closure (LDSO)
  - High-gradient + Corner
  - BoW-based loop detection + Pose graph optimization

Bibliography

[Barfoot-17] Barfoot, T.D., 2017. *State estimation for robotics*. Cambridge University Press.

[Civera-08] Civera, J., Davison, A.J. and Montiel, J.M., 2008. Inverse depth parametrization for monocular SLAM. *IEEE transactions on robotics*, *24*(5), pp.932-945.

[Debevec-08] Debevec, P.E. and Malik, J., 2008, August. Recovering high dynamic range radiance maps from photographs. In *ACM SIGGRAPH 2008 classes* (p. 31). ACM.

[Dong-Si-12] Dong-Si, T.C. and Mourikis, A.I., 2012, May. Consistency analysis for sliding-window visual odometry. In *2012 IEEE International Conference on Robotics and Automation* (pp. 5202-5209). IEEE.

[Engel-16] Engel, J., Usenko, V. and Cremers, D., 2016. A photometrically calibrated benchmark for monocular visual odometry. *arXiv preprint arXiv:1607.02555*.

[Engel-18] Engel, J., Koltun, V. and Cremers, D., 2018. Direct sparse odometry. *IEEE transactions on pattern analysis and machine intelligence*, *40*(3), pp.611-625.

Bibliography

[Frese-05] Frese, U., 2005, April. A proof for the approximate sparsity of SLAM information matrices. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation* (pp. 329-335). IEEE.

[Gao-18] Gao, X., Wang, R., Demmel, N. and Cremers, D., 2018, October. LDSO: Direct sparse odometry with loop closure. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 2198-2204). IEEE.

[Gálvez-López-12] Gálvez-López, D. and Tardos, J.D., 2012. Bags of binary words for fast place recognition in image sequences. *IEEE Transactions on Robotics*, *28*(5), pp.1188-1197.

[Hartley-03] Hartley, R. and Zisserman, A., 2003. *Multiple view geometry in computer vision*. Cambridge university press.

[Huang-08] Huang, G.P., Mourikis, A.I. and Roumeliotis, S.I., 2008, May. Analysis and improvement of the consistency of extended Kalman filter based SLAM. In *2008 IEEE International Conference on Robotics and Automation* (pp. 473-479). IEEE.

Bibliography

[Scaramuzza-11] Scaramuzza, D. and Fraundorfer, F., 2011. Visual odometry [tutorial]. *IEEE robotics & automation magazine*, *18*(4), pp.80-92.

[Schöps-14] Schöps, T., Engel, J. and Cremers, D., 2014, September. Semi-dense visual odometry for AR on a smartphone. In *2014 IEEE international symposium on mixed and augmented reality (ISMAR)* (pp. 145-150). IEEE.

[Thrun-05] Thrun, S., Burgard, W. and Fox, D., 2005. *Probabilistic robotics*. MIT press.

[Triggs-99] Triggs, B., McLauchlan, P.F., Hartley, R.I. and Fitzgibbon, A.W., 1999, September. Bundle adjustment—a modern synthesis. In *International workshop on vision algorithms* (pp. 298-372). Springer, Berlin, Heidelberg.