Direct Sparse Odometry

Jakob Engel[1], Vladlen Koltun[2] and Daniel Cremers[1]

[1]Technical University Munich

[2]Intel Labs

TPAMI 2018

Overview

Summary

- A **direct** and **sparse** formulation for the SLAM problem.
- **Direct** - Photometric error minimization.
- **Sparse** - Pixels sampled evenly throughout the images
- Can utilize information from edges/ white walls.
- Integrates full photometric Calibration.
- Outperformed SOTA direct and indirect methods (in 2016)
- Runs real time on a laptop computer.

Summary

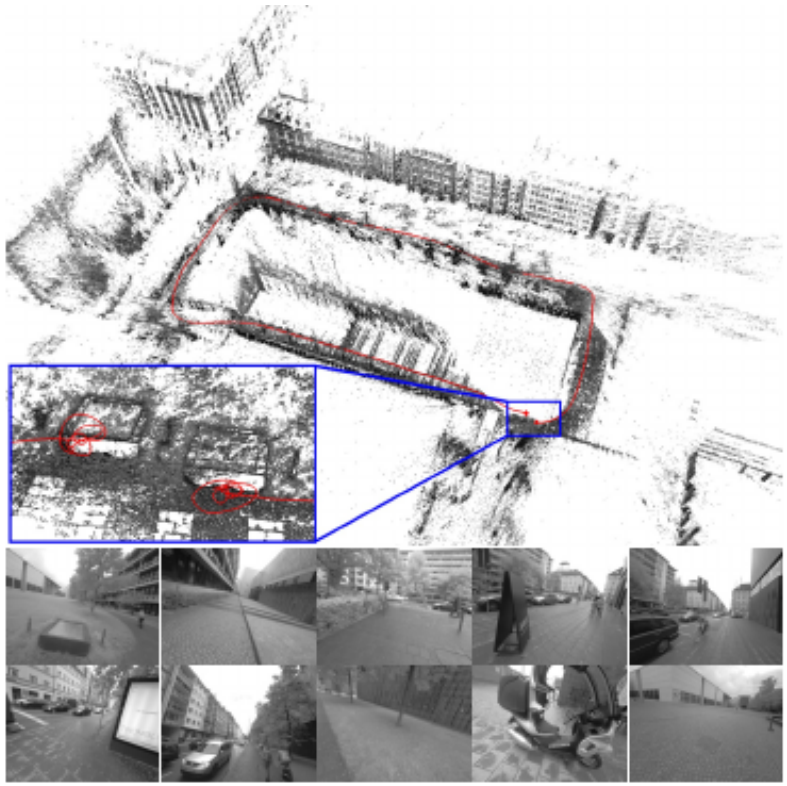


Figure 1: 3D reconstruction and camera trajectory tracked around a building. Bottom left inset shows the drift accumulated

Overview

- Introduction
  - SLAM Formulations
  - Motivation
  - Preliminaries
- Direct Sparse Model
  - Calibration
  - Model formulation
  - Windowed Optimization
- Visual Odometry Front End
  - Frame Management
  - Point Management
- Experiments and Results

Direct *vs* Indirect

The general objective: Given noisy measurements $\mathbf{Y}$ we aim to obtain true values $\mathbf{X}$ using a Maximum Likelihood approach:

$$\mathbf{X}^* = \text{argmax}_{\mathbf{X}} \; P(\mathbf{Y}|\mathbf{X})$$

Direct *vs* Indirect

The general objective: Given noisy measurements $\mathbf{Y}$ we aim to obtain true values $\mathbf{X}$ using a Maximum Likelihood approach:

$$\mathbf{X}^* = \text{argmax}_{\mathbf{X}} \; P(\mathbf{Y}|\mathbf{X})$$

Indirect Methods

Direct *vs* Indirect

Indirect Methods

- Minimize *reprojection error*.
- Geometry is estimated (usually) only for the small set of feature points.

Direct Methods

- Circumvent initial feature extraction step.
- Minimize a *photometric error* - directly operate on the pixel intensities.

Direct *vs* Indirect

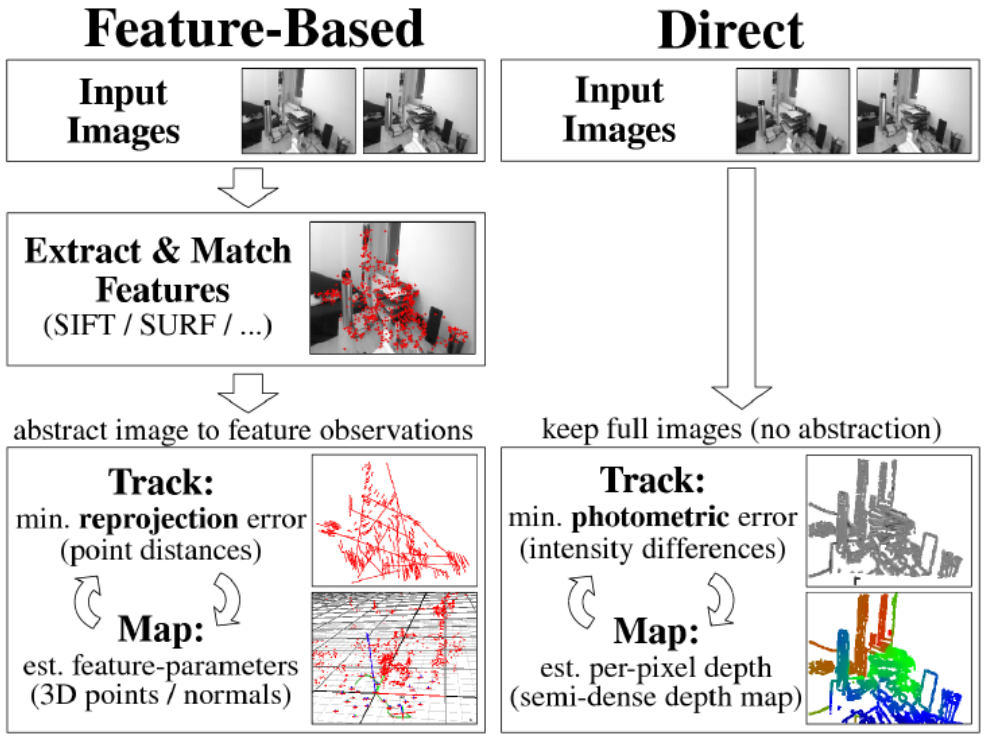


Figure 2: A comparison of direct and indirect methods in LSD SLAM

Dense *vs* Sparse

Sparse

- Employ and reconstruct only a selected set of independent points.
- Circumvents need for any geometric prior: positions of keypoints are independent given the camera parameters.

Dense *vs* Sparse

Sparse

- Employ and reconstruct only a selected set of independent points.
- Circumvents need for any geometric prior: positions of keypoints are independent given the camera parameters.

Dense

- Utilize all the available information in the 2D domain.
- Use a geometric prior, exploiting smoothness in the image region.

SLAM Formulations

- Sparse + Indirect
  - 3D geometry estimated from a set of keypoint matches, employing a geometric error without any prior.
  - Widely used! Ex: ORB-SLAM, PTAM.

- Dense + Indirect
  - 3D geometry estimated using a dense, regularized optical flow field.
  - Use a geometric prior - smoothness in the flow field.
  - Geometric error - ordering and smoothness in the depth maps.

SLAM Formulations

- Dense + Direct
  - 3D geometry estimated by minimizing a photometric error, employing a geometric prior.
  - Ex: DTAM, LSD-SLAM.

- Sparse + Direct
  - Formulation in DSO SLAM.
  - Optimize a photometric error, without any prior, for a selected set of points in the image.

Overview

- Introduction
  - SLAM Formulations
  - Motivation
  - Preliminaries
- Direct Sparse Model
  - Calibration
  - Model formulation
  - Windowed Optimization
- Visual Odometry Front End
  - Frame Management
  - Point Management
- Experiments and Results

Motivation

Why Direct?

- Keypoints are widely used since they are robust to geometric and photometric distortions such as exposure changes, gamma correction, etc.
- However, many of the current cameras provide the complete sensor model

Motivation

Why Direct?

- Keypoints are widely used since they are robust to geometric and photometric distortions such as exposure changes, gamma correction, etc.
- However, many of the current cameras provide the complete sensor model
- Auto-exposure and gamma correction aren't unknowns, they are features!
- Allows us to sample across all image data - including edges and white walls.

Motivation

Why Sparse?

- Geometric priors introduce correlations between geometry parameters.
- Leads to the addition of non-diagonal elements in the Hessian matrix.
- Non-sparse Hessian is harder for joint optimization.
- Current priors are limited in their expressive complexity.

Overview

Lie groups and Lie algebra

- Lie groups and Lie algebra:
  - A Lie group is a group whose elements are organised *continuously* and *smoothly*. Group elements are geometric objects.
  - The Lie algebra of a Lie group is the corresponding *linearized* version.

Ex : SO(3) - Special Orthogonal Group of size 3x3 (Rotation matrices) with corresponding Lie algebra $\mathfrak{so}(3)$

Lie groups and Lie algebra

- Lie groups and Lie algebra:
  - A Lie group is a group whose elements are organised *continuously* and *smoothly*. Group elements are geometric objects.
  - The Lie algebra of a Lie group is the corresponding *linearized* version.

  Ex : SO(3) - Special Orthogonal Group of size 3x3 (Rotation matrices) with corresponding Lie algebra $\mathfrak{so}(3)$
- 3D Rigid Body Transformations: A 3D rigid body transform $\mathbf{T} \in \mathrm{SE}(3)$ denotes rotation and translation in 3D i.e. is defined by

$$\mathbf{T} = \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{pmatrix}$$

Here, $\mathbf{R} \in \mathrm{SO}(3)$ and $\mathbf{t} \in \mathbb{R}^3$

Lie algebra in optimization

- Each element in the Lie group has a corresponding element in the Lie algebra.
- Lie algebra provides a natural way for optimisation.
- The camera pose $\mathbf{T}$ has a corresponding element $\boldsymbol{\xi} \in \mathfrak{se}(3)$ related as :

$$\mathbf{T} = \exp_{\mathfrak{se}(3)} \boldsymbol{\xi}$$

- With a slight abuse in notation, $\boldsymbol{\xi} \in \mathbb{R}^6$ is used to represent the poses in $\mathfrak{se}(3)$

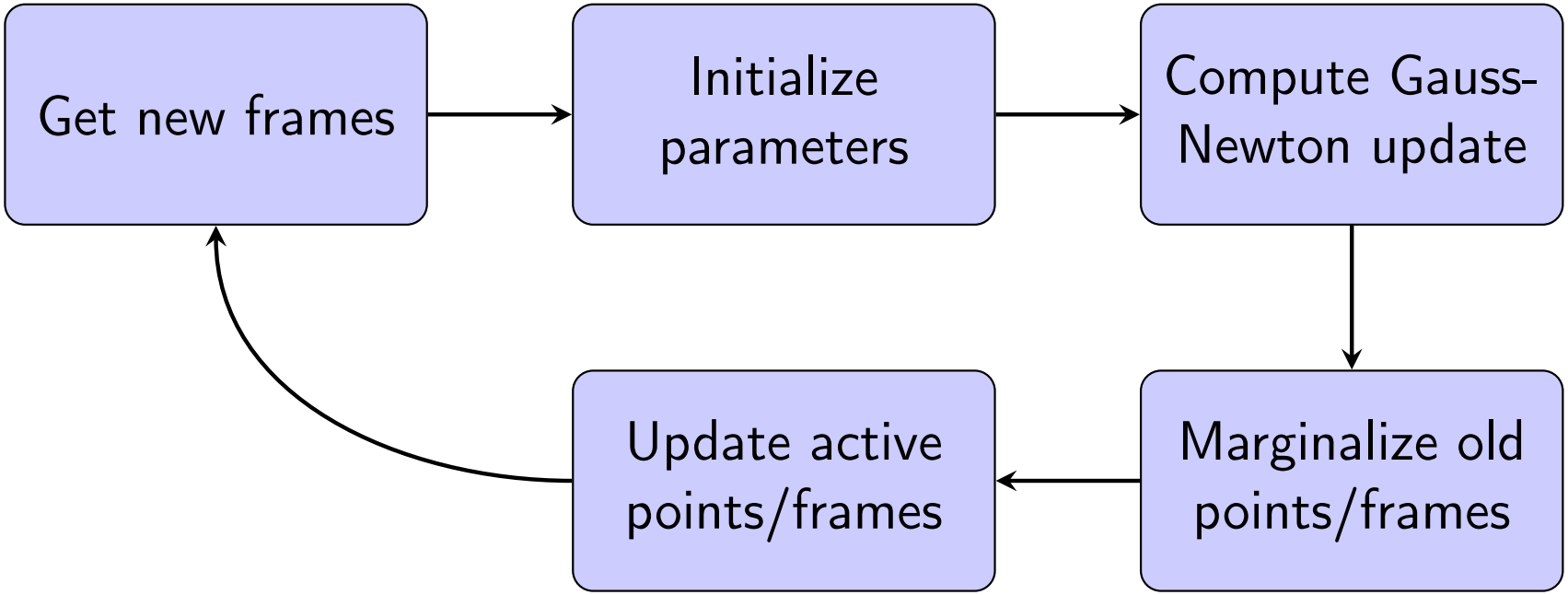
Notations

- Bold lower case letters ($\mathbf{x}$) represent vectors and bold upper case letters ($\mathbf{T}$) represent matrices.
- $\mathbf{T}_i \in \mathrm{SE}(3)$ represents the camera pose of the i-th frame.
- The operator $\boxplus : \mathfrak{se}(3) \times SE(3) \rightarrow SE(3)$ is defined as :

$$\mathbf{x}_i \boxplus \mathbf{T}_i = e^{\widehat{\mathbf{x}_i}} \cdot \mathbf{T}_i$$

- This notation, $\boxplus$ is extended to all optimized parameters, and for parameters other than SE(3), it denotes addition
- $\boldsymbol{\zeta} \in SE(3)^n \times \mathbb{R}^m$ is used to denote all optimized variables

Windowed optimisation outlined



Overview

- Introduction
  - SLAM Formulations
  - Motivation
  - Preliminaries
- Direct Sparse Model
  - Calibration
  - Model formulation
  - Windowed Optimization
- Visual Odometry Front End
  - Frame Management
  - Point Management
- Experiments and Results

The Direct Sparse Model

- The model is based on continuous optimization of photometric error over recent frames, using a photometrically calibrated model
- Joint optimization over all parameters makes the model equivalent to windowed sparse bundle adjustment
- Each 3d point corresponding to a frame has 1 degree of freedom(Inverse depth)

Overview

- Introduction
  - SLAM Formulations
  - Motivation
  - Preliminaries
- Direct Sparse Model
  - Calibration
  - Model formulation
  - Windowed Optimization
- Visual Odometry Front End
  - Frame Management
  - Point Management
- Experiments and Results

Calibration

- In direct approaches, along with a geometric camera model, it is also beneficial to use a photometric camera model.
- Unnecessary in indirect models - feature extractors are invariant to photometric variations.

Calibration

Geometric Calibration

- For simplicity, the pinhole model is used. DSO can be extended to other models, at the cost of computational complexity.
- The projection and inverse projection functions, throughout this discussion will be represented by $\Pi_c : \mathbb{R}^3 \rightarrow \Omega$ and $\Pi_c^{-1} : \Omega \times \mathbb{R} \rightarrow \mathbb{R}^3$, where c represents the internal camera parameters

Calibration

Photometric Calibration

- The combined model accounts for nonlinear response function of the sensor, $G : \mathbb{R} \rightarrow [0, 255]$, Lens attenuation/Vignetting, $V : \Omega \rightarrow [0, 1]$, Irradiance of the point B_i and exposure time t_i. The pixel values recorded are modelled as :

$$I_i(x) = G(t_i V(x) B_i(x))$$

- These readings are calibrated, by photometric correction as follows:

$$I_i'(x) = t_i B_i(x) = \frac{G^{-1}(I_i(x))}{V(x)}$$

Overview

- Introduction
  - SLAM Formulations
  - Motivation
  - Preliminaries
- Direct Sparse Model
  - Calibration
  - Model formulation
  - Windowed Optimization
- Visual Odometry Front End
  - Frame Management
  - Point Management
- Experiments and Results

Model Formulation

The photometric error for a point $\mathbf{p} \in \Omega_i$ frame I_i observed in a target frame I_j is defined as a weighted SSD over a neighbourhood. Let :

$$E_{pj} = \sum_{\boldsymbol{p} \in \mathcal{N}_{\boldsymbol{p}}} \omega_{\boldsymbol{p}} ||(I_j[\boldsymbol{p'}] - b_j) - \frac{t_j e^{a_j}}{t_i e^{a_i}}(I_i[\boldsymbol{p}] - b_i)||_\gamma$$

Here, $\boldsymbol{p'}$ is the projection of $\boldsymbol{p}$ in I_j, and is given by :

$$\boldsymbol{p'} = \Pi_c(\boldsymbol{T}_{ji}\Pi_c^{-1}(\boldsymbol{p}, d_{\boldsymbol{p}}))$$

where $d_{\mathbf{p}}$ is the inverse depth

Model Formulation

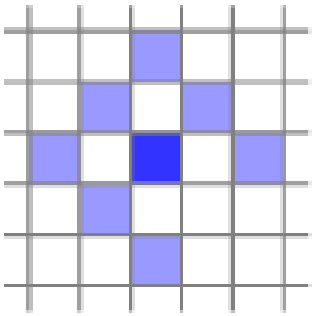


Figure 3: For a pixel p, the set $\mathcal{N}_p$ is chosen using this pattern. This pattern enables SSE Optimized processing

Model Formulation

- The constants a_i, b_i, a_j, b_j allow the same formulation to apply when exposure times t_i, t_j are unknown.
- $||.||_\gamma$ represents the huber norm, and ω_p is a weighting introduced to compensate for high gradients, given by:

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

for some constant c. This can be interpreted as the addition of small geometric noise.

Model Formulation

Finally, the photometric error is given by:

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

Model Formulation

Some comments on the final formulation

- Each term depends on 2 frames. While this adds some off diagonal terms in the hessian, the sparsity pattern is preserved after the application of the schur complement to marginalize point parameters
- If the exposure times are known, to pull the affine transfer function to 0, we further add:

$$E_{prior} := \sum_{i \in \mathcal{F}} (\lambda_a a_i^2 + \lambda_b b_i^2)$$

- If t_i, t_j are unknown,we set $\lambda_a = \lambda_b = 0$

Overview

- Introduction
  - SLAM Formulations
  - Motivation
  - Preliminaries
- Direct Sparse Model
  - Calibration
  - Model formulation
  - Windowed Optimization
- Visual Odometry Front End
  - Frame Management
  - Point Management
- Experiments and Results

Windowed Optimization

The photometric error is optimized over a sliding window, using Gauss Newton Optimization

Windowed Optimization

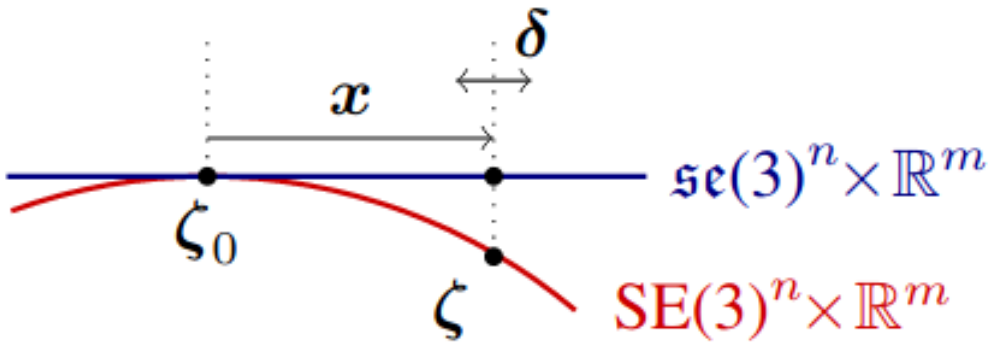


Figure 4: Marginalizing a residual that depends on a parameter in ζ fixes the tangent space in which future updates on the image are accumulated

Windowed Optimization

Gauss Newton Optimization

The optimization system is represented as :

$$\mathbf{H} = \mathbf{J}^{\mathbf{T}}\mathbf{W}\mathbf{J} \quad \text{and} \quad \mathbf{b} = -\mathbf{J}^{\mathbf{T}}\mathbf{W}\mathbf{r}$$

Where $\mathbf{W}$ is an nxn matrix containing the weights, $\mathbf{r}$ is a n dimensional stacked residual vector, and $\mathbf{J}$ is the Jacobian or $\mathbf{r}$

Windowed Optimization

- For simplicity, we will consider 1 residual r_k, and the corresponding row in the Jacobian $\mathbf{J}_k$
- The residuals are always evaluated at the current Parameter values.

$$r_k = r_k(\mathbf{x} \boxplus \zeta_0)$$

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

Windowed Optimization

The jacobian, is given by :

$$\mathbf{J}_k = \frac{\partial r_k((\delta + \mathbf{x}) \boxplus \zeta_0)}{\partial \boldsymbol{\delta}}$$

Further, this can be written as

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

Where, $\delta_{geo} := (T_i, T_j, d, c)$ and $\delta_{photo} := (a_i, a_j, b_i, b_j)$

Windowed Optimization

The Following approximations are made:

- $\boldsymbol{J}_{geo}$ and $\boldsymbol{J}_{photo}$ are calculated at $\mathbf{x} = 0$("First Estimate Jacobian"). $\boldsymbol{J}_{I}$ is calculated at each value of x
- $\boldsymbol{J}_{geo}$ is assumed to be same for all residuals belonging to a point, and only calculated for the centre pixel.

Windowed Optimization

Finally, the update step is given by :

$$\mathbf{x}^{\text{new}} \leftarrow \boldsymbol{\delta} + \mathbf{x} \quad \text{where} \quad \boldsymbol{\delta} = \mathbf{H}^{-1}\mathbf{b}$$

$$\zeta_0^{\text{new}} \leftarrow \mathbf{x} \boxplus \zeta_0 \quad \text{and} \quad \mathbf{x} \leftarrow \mathbf{0}$$

Windowed Optimization

Marginalization

- This is done to eliminate older variables, and to limit the set of variables to be optimized over
- This reduces the computational burden, and facilitates results in real time.

Windowed Optimization

Marginalization

- Let E' denote the energy in the residuals that depend on variables to be marginalized. An approximation on E' around the estimate $\zeta = \mathbf{x} \boxplus \zeta_0$ gives us

$$E'(\mathbf{x} + \zeta_0) \approx 2(\mathbf{x} - \mathbf{x}_0)^T \mathbf{b} + (\mathbf{x} - \mathbf{x}_0)^T H(\mathbf{x} - \mathbf{x}_0) + c$$

$$\begin{aligned} E'(\mathbf{x} + \zeta_0) &\approx 2\mathbf{x}^T(\mathbf{b} - \mathbf{H}\mathbf{x}_0) + \mathbf{x}^T\mathbf{H}\mathbf{x} + c + \mathbf{x}_0^T\mathbf{H}\mathbf{x}_0 - \mathbf{x}_0^T\mathbf{b} \\ &= 2\mathbf{x}^T b' + \mathbf{x}^T\mathbf{H}\mathbf{x} + c' \end{aligned}$$

  Where $\mathbf{x}_0$ is the current value of $\mathbf{x}$.

- Since this is a quadratic, the Schur complement trick can be applied to marginalize variables

Windowed Optimization

Marginalization : The Schur Complement trick :
The system can be written as

$$\begin{bmatrix} \boldsymbol{H}_{\alpha\alpha} & \boldsymbol{H}_{\alpha\beta} \\ \boldsymbol{H}_{\beta\alpha} & \boldsymbol{H}_{\beta\beta} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_{\alpha} \\ \boldsymbol{x}_{\beta} \end{bmatrix} = \begin{bmatrix} \boldsymbol{b}'_{\alpha} \\ \boldsymbol{b}'_{\beta} \end{bmatrix}$$

This reduces to $\widehat{H_{\alpha\alpha}x_{\alpha}} = \widehat{b'_{\alpha}}$ Where

$$\widehat{H}_{\alpha\alpha} = H_{\alpha\alpha} - H_{\alpha\beta}H_{\beta\beta}^{-1}H_{\beta\alpha}$$

$$\widehat{b}'_{\alpha} = b'_{\alpha} - H_{\alpha\beta}H_{\beta\beta}^{-1}b'_{\beta}$$

Windowed Optimization

This means that

$$E'(\boldsymbol{x}_\alpha \boxplus (\zeta_0)_\alpha) = 2\boldsymbol{x}_\alpha^T \widehat{\boldsymbol{b}'_\alpha} + \boldsymbol{x}_\alpha^T \widehat{H}_{\alpha\alpha} \boldsymbol{x}_\alpha$$

As this is a quadratic function on $\boldsymbol{x}$, It can be trivially added to the full photometric error during all subsequent optimization and marginalization operations

Overview

- Introduction
  - SLAM Formulations
  - Motivation
  - Preliminaries
- Direct Sparse Model
  - Calibration
  - Model formulation
  - Windowed Optimization
- Visual Odometry Front End
  - Frame Management
  - Point Management
- Experiments and Results

Visual Odometry Front End

The front-end needs to replace many operations that in the indirect setting are accomplished by keyframe detectors (determining visibility, point selection) and initialization procedures such as RANSAC. In particular,

- It determines the sets $\mathcal{F}, \mathcal{P}_i$ and obs($\mathbf{p}$) that make up the error terms of E_{photo} i.e. it decides which points and frames are used, and in which frames a point is visible.

Visual Odometry Front End

The front-end needs to replace many operations that in the indirect setting are accomplished by keyframe detectors (determining visibility, point selection) and initialization procedures such as RANSAC. In particular,

- It determines the sets $\mathcal{F}, \mathcal{P}_i$ and obs($\mathbf{p}$) that make up the error terms of E_{photo} i.e. it decides which points and frames are used, and in which frames a point is visible.
- It provides initialization for new parameters, required for optimizing E_{photo}.

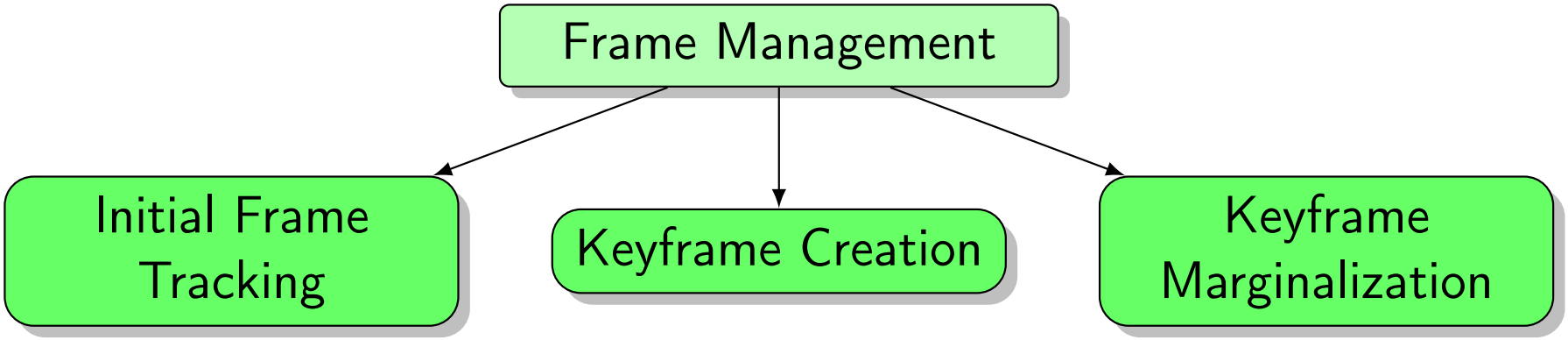
Visual Odometry Front End

The front-end needs to replace many operations that in the indirect setting are accomplished by keyframe detectors (determining visibility, point selection) and initialization procedures such as RANSAC. In particular,

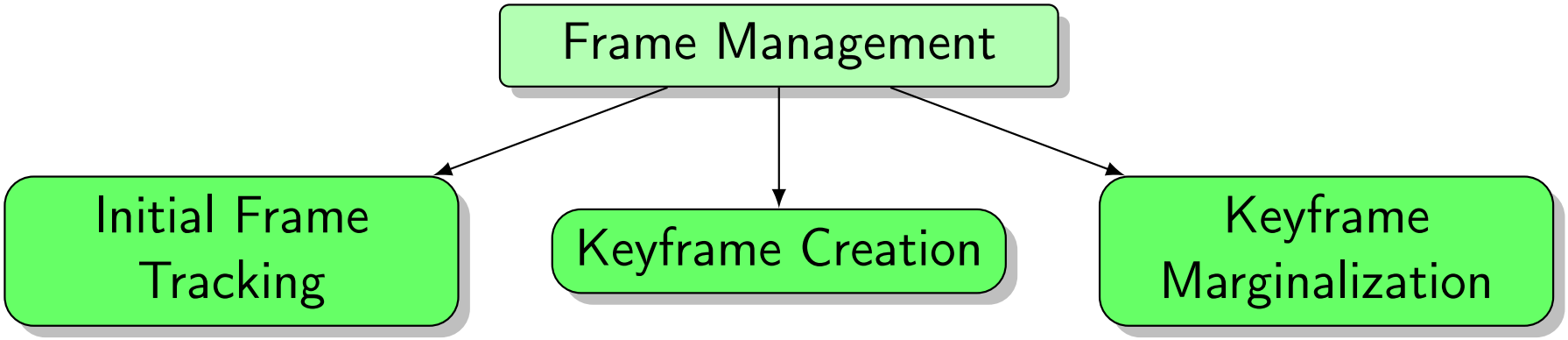
- It determines the sets $\mathcal{F}, \mathcal{P}_i$ and obs($\mathbf{p}$) that make up the error terms of E_{photo} i.e. it decides which points and frames are used, and in which frames a point is visible.
- It provides initialization for new parameters, required for optimizing E_{photo}.
- It decides when a point/frame needs to be marginalized.

Overview

Frame Management



Frame Management



Note

- This method always keeps a window of up to N_f active keyframes. In this case, $N_f = 7$.

Initial Frame Tracking

- When a new keyframe is created, all active points are projected into it, creating a semi-dense depth map.

Initial Frame Tracking

- When a new keyframe is created, all active points are projected into it, creating a semi-dense depth map.
- New frames are tracked w.r.t. **only** this frame using conventional two-frame direct image alignment, a multi-scale image pyramid and a constant motion model to initialize.

Initial Frame Tracking

- When a new keyframe is created, all active points are projected into it, creating a semi-dense depth map.
- New frames are tracked w.r.t. **only** this frame using conventional two-frame direct image alignment, a multi-scale image pyramid and a constant motion model to initialize.
- If the final RMSE for a frame is more than twice that of the frame before, we assume that direct image alignment failed and attempt to recover by initializing with up to 27 different small rotations in different directions.

Keyframe Creation

Three criteria are considered to determine if a new keyframe is required:

Keyframe Creation

Three criteria are considered to determine if a new keyframe is required:

- There is a change in field of view. This is measured by mean square optical flow i.e. $f := \sqrt{\frac{1}{n}\sum_{i=1}^{n} ||\mathbf{p} - \mathbf{p}'||^2}$

Keyframe Creation

Three criteria are considered to determine if a new keyframe is required:

- There is a change in field of view. This is measured by mean square optical flow i.e. $f := \sqrt{\frac{1}{n}\sum_{i=1}^{n} ||\mathbf{p} - \mathbf{p}'||^2}$
- There is a camera translation which causes occlusions and disocclusions. This is measured by the mean flow without rotation, i.e. $f_t := \sqrt{\frac{1}{n}\sum_{i=1}^{n} ||\mathbf{p} - \mathbf{p}'_t||^2}$, where $\mathbf{p}_t$ is the warped point position with $\mathrm{R} = \mathrm{I}_{3\times 3}$.

Keyframe Creation

Three criteria are considered to determine if a new keyframe is required:

- There is a change in field of view. This is measured by mean square optical flow i.e. $f := \sqrt{\frac{1}{n}\sum_{i=1}^{n} ||\mathbf{p} - \mathbf{p}'||^2}$
- There is a camera translation which causes occlusions and disocclusions. This is measured by the mean flow without rotation, i.e. $f_t := \sqrt{\frac{1}{n}\sum_{i=1}^{n} ||\mathbf{p} - \mathbf{p}'_t||^2}$, where $\mathbf{p}_t$ is the warped point position with $\mathrm{R} = \mathrm{I}_{3\times 3}$.
- There is a significant change in exposure time. This is measured by the rel. brightness factor between two frames $a := |log(e^{a_j - a_i} t_j t_i^{-1})|$.

Keyframe Creation

Three criteria are considered to determine if a new keyframe is required:

- There is a change in field of view. This is measured by mean square optical flow i.e. $f := \sqrt{\frac{1}{n}\sum_{i=1}^{n} ||\mathbf{p} - \mathbf{p}'||^2}$
- There is a camera translation which causes occlusions and disocclusions. This is measured by the mean flow without rotation, i.e. $f_t := \sqrt{\frac{1}{n}\sum_{i=1}^{n} ||\mathbf{p} - \mathbf{p}'_t||^2}$, where $\mathbf{p}_t$ is the warped point position with $\mathrm{R} = \mathrm{I}_{3\times 3}$.
- There is a significant change in exposure time. This is measured by the rel. brightness factor between two frames $a := |log(e^{a_j - a_i} t_j t_i^{-1})|$.

A new keyframe is taken if $w_f f + w_{f_t} f_t + w_a a > T_{kf}$, where w_f, w_{f_t}, w_a are relative weights and $T_{kf} = 1$ by default.

Keyframe Marginalization

Let $I_1 \dots I_n$ be the set of active keyframes, with I_1 being the newest and I_n being the oldest:

Keyframe Marginalization

Let $I_1 \dots I_n$ be the set of active keyframes, with I_1 being the newest and I_n being the oldest:

- The latest two keyframes (I_1 and I_2) are kept.

Keyframe Marginalization

Let $I_1 \dots I_n$ be the set of active keyframes, with I_1 being the newest and I_n being the oldest:

- The latest two keyframes (I_1 and I_2) are kept.
- Frames with less than 5% of their points visible in I_1 are marginalized.

Keyframe Marginalization

Let $I_1 \dots I_n$ be the set of active keyframes, with I_1 being the newest and I_n being the oldest:

- The latest two keyframes (I_1 and I_2) are kept.
- Frames with less than 5% of their points visible in I_1 are marginalized.
- If more than N_f frames are active, we marginalize the one (excluding I_1 and I_2) which maximizes a "distance score" $s(I_i)$, computed as:

$$s(I_i) = \sqrt{d(i,1)} \sum_{j \in [3,n] \setminus i} \frac{1}{d(i,j) + \epsilon}$$

where $d(i,j)$ is the Euclidean distance between keyframes I_i and I_j, and ϵ is a small constant.

Keyframe Management

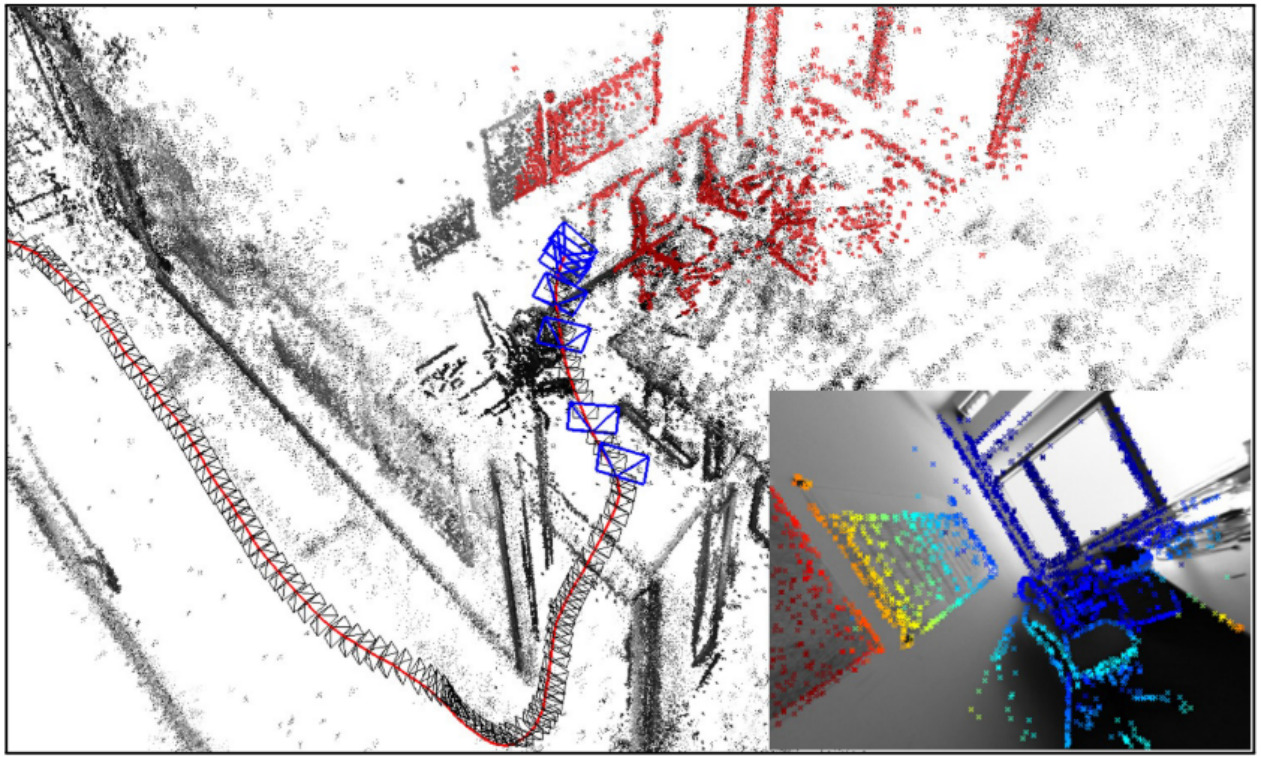
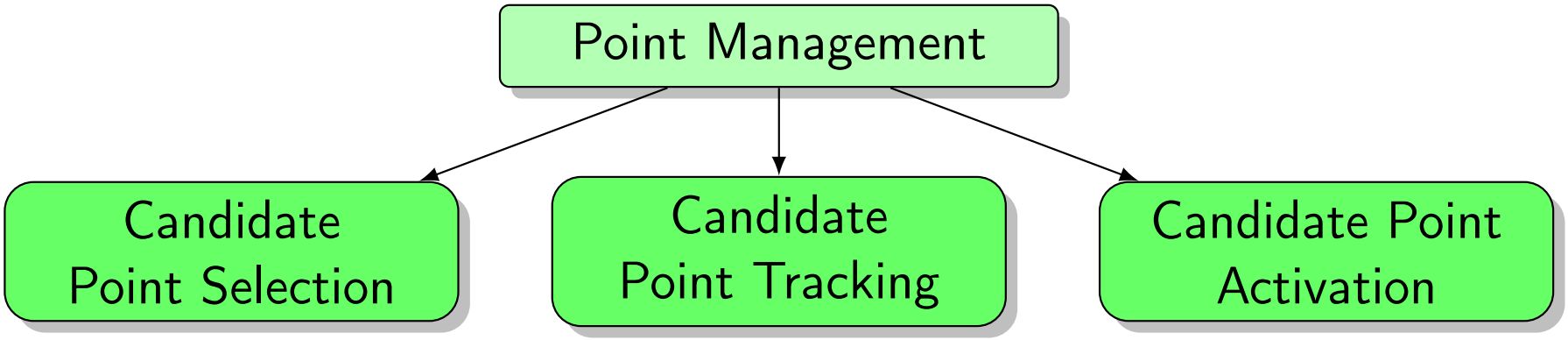


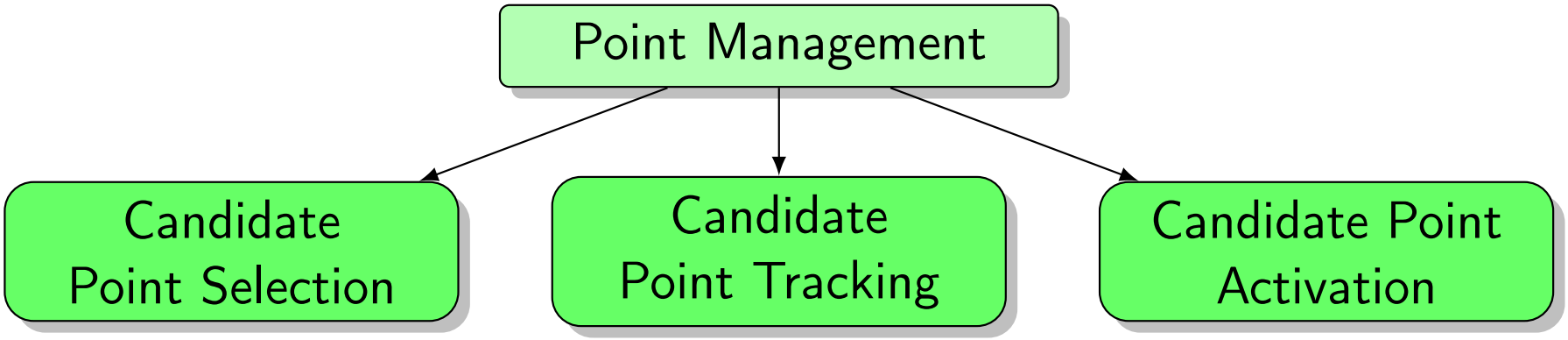
Figure 5: The image shows the full point cloud, as well as the positions of all keyframes (black camera frustums) – active points and keyframes are shown in red and blue respectively. The inlay shows the newly added keyframe.

Overview

Point Management



Point Management



Note

- This method always keeps a fixed number N_p of active points, equally distributed across space and active frames. In this case, $N_p = 2000$.
- Although there are N_p candidates *in each frame*, however only N_p active points are present *across all active frames combined*.

Candidate Point Selection

- A region-adaptive gradient threshold (RGT) is obtained by splitting the image into 32×32 blocks. For each block, the threshold is $\bar{g} + g_{th}$, where $\bar{g}$ is the median absolute gradient over all pixels in that block, and g_{th} a global constant (here, $g_{th} = 7$).

Candidate Point Selection

- A region-adaptive gradient threshold (RGT) is obtained by splitting the image into 32×32 blocks. For each block, the threshold is $\bar{g} + g_{th}$, where $\bar{g}$ is the median absolute gradient over all pixels in that block, and g_{th} a global constant (here, $g_{th} = 7$).
- To obtain an equal distribution of points throughout the image, the image split into $d \times d$ blocks, and from each block select the pixel with largest gradient **only** if it surpasses the RGT.

Candidate Point Selection

- A region-adaptive gradient threshold (RGT) is obtained by splitting the image into 32×32 blocks. For each block, the threshold is $\bar{g} + g_{th}$, where $\bar{g}$ is the median absolute gradient over all pixels in that block, and g_{th} a global constant (here, $g_{th} = 7$).
- To obtain an equal distribution of points throughout the image, the image split into $d \times d$ blocks, and from each block select the pixel with largest gradient **only** if it surpasses the RGT.
- It is found that it is often beneficial to also include some points with weaker gradient from regions where no high-gradient points are present.

Candidate Point Selection

- A region-adaptive gradient threshold (RGT) is obtained by splitting the image into 32×32 blocks. For each block, the threshold is $\bar{g} + g_{th}$, where $\bar{g}$ is the median absolute gradient over all pixels in that block, and g_{th} a global constant (here, $g_{th} = 7$).
- To obtain an equal distribution of points throughout the image, the image split into $d \times d$ blocks, and from each block select the pixel with largest gradient **only** if it surpasses the RGT.
- It is found that it is often beneficial to also include some points with weaker gradient from regions where no high-gradient points are present.
- To achieve this, we repeat this procedure twice more, with decreased gradient threshold and block-size $2d$ and $4d$.

Candidate Point Selection

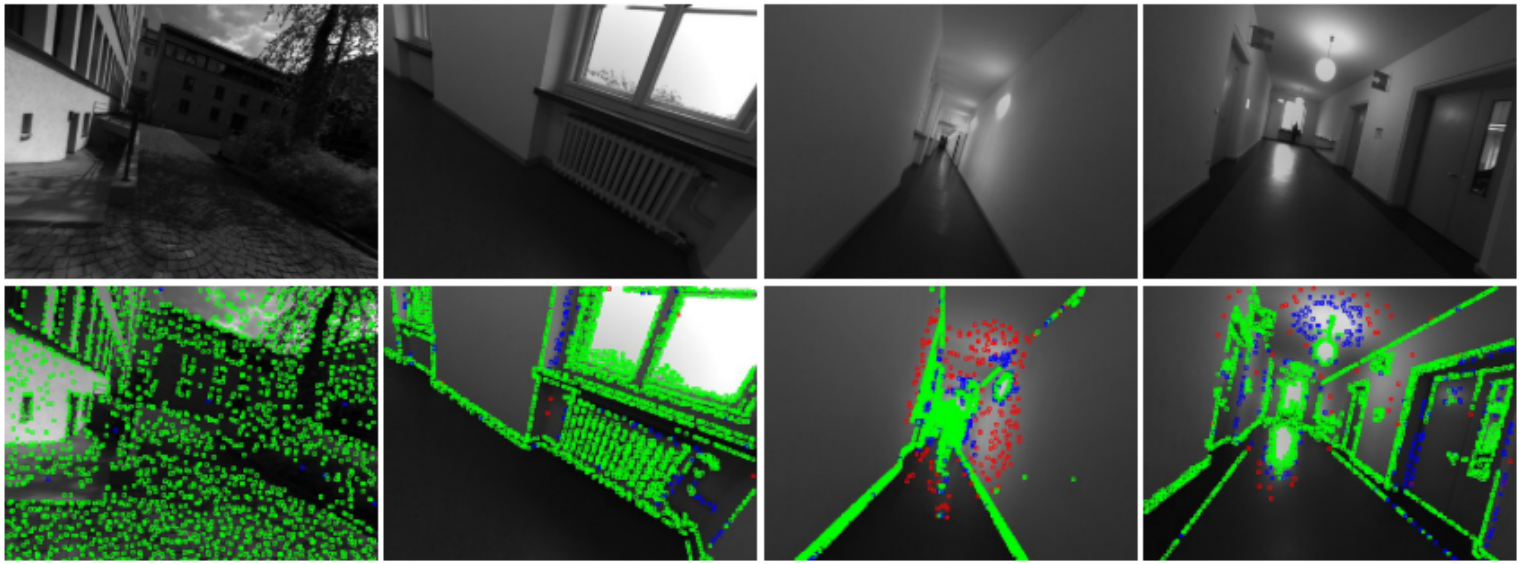


Figure 6: The top row shows the original images, the bottom row shows the points chosen as candidates to be added to the map (2000 in each frame). Points selected on the first pass are shown in green, those selected on the second and third pass in blue and red respectively

Candidate Point Tracking

- Point candidates are tracked in subsequent frames using a discrete search along the epipolar line, minimizing the photometric error as discussed earlier.

Candidate Point Tracking

- Point candidates are tracked in subsequent frames using a discrete search along the epipolar line, minimizing the photometric error as discussed earlier.
- From the best match the depth and associated variance is computed, which is used to constrain the search interval for the subsequent frame.

Candidate Point Activation

After a set of old points is marginalized, new point candidates are activated to replace them.

Candidate Point Activation

After a set of old points is marginalized, new point candidates are activated to replace them.

- At first, all active points are projected onto the most recent keyframe.

Candidate Point Activation

After a set of old points is marginalized, new point candidates are activated to replace them.

- At first, all active points are projected onto the most recent keyframe.
- Then, candidate points which – also projected into this keyframe – maximize the distance to any existing point are activated.

Outlier and Occlusion Detection

Since the available image data generally contains much more information than can be used in real time, it is important to identify and remove potential outliers as early as possible.

- First, when searching along the epipolar line during candidate tracking, points for which the minimum is not sufficiently distinct are permanently discarded
- Second, point observations for which the photometric error surpasses a threshold are removed. The threshold is continuously adapted with respect to the median residual in the respective frame.

Overview

Datasets

The performance of the model is evaluated on two datasets:

- **TUM monoVO dataset**: 50 photometrically calibrated sequences, comprising 105 minutes of video recorded in both indoors and outdoors. The alignment error (e_{align}) as defined in the above paper is calucated.
- **EuRoC MAV dataset**: 11 stereo-inertial sequences containing 19 minutes of video, recorded in 3 different indoor environments. The absolute trajectory error (e_{ate}) is calculated.

Methodology

- All sequences both forwards and backwards, 5 times each i.e. total 10 times. For the EuRoC MAV dataset we further run both the left and the right video separately.
- In total, this gives 500 runs for the TUM-monoVO dataset and 220 runs for the EuRoC MAV dataset.
- For ORB-SLAM, the video is played at 20% speed, whereas DSO is run in a sequentialized, single-threaded implementation that runs approximately four times slower than real time unless stated otherwise.

Results on TUM-monoVO dataset

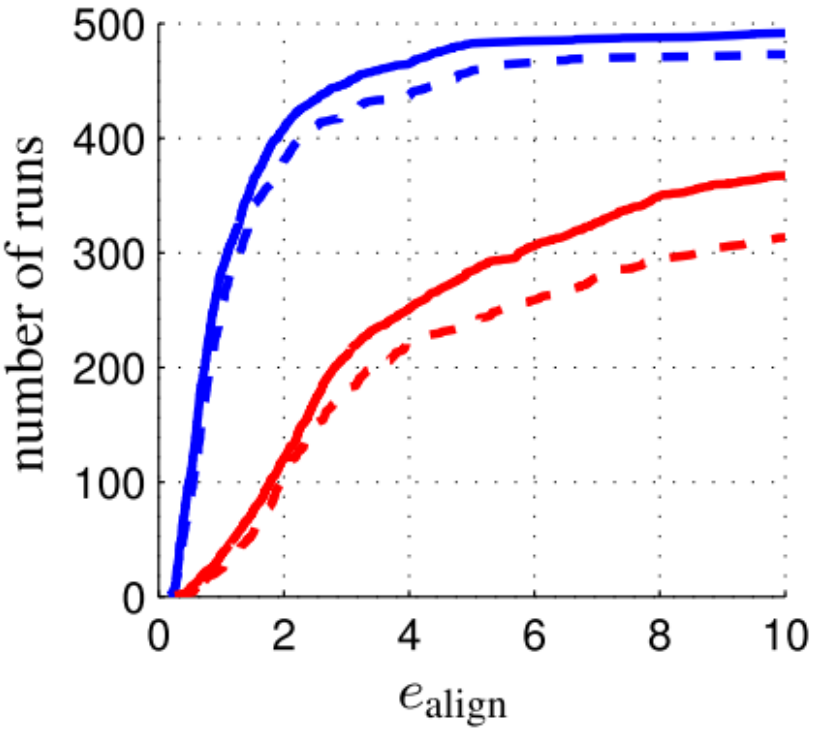


Figure 7: Cummulative Error Plot. The solid line corresponds to sequentialized, non-real-time execution, the dashed line to hard enforced real-time processing. The blue line represents DSO and the red line represents ORB-SLAM.

Results on EuRoC MAV dataset

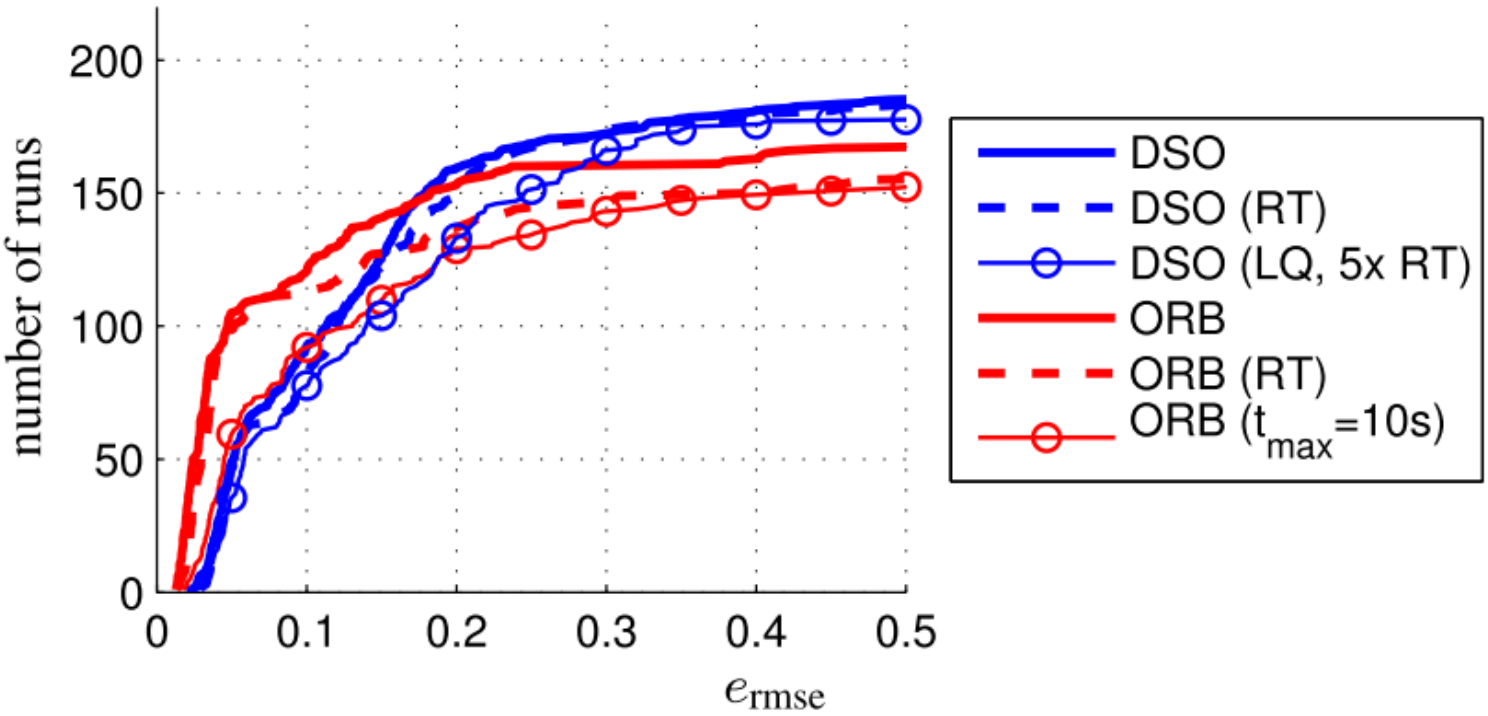


Figure 8: Cummulative Error Plot. Absolute Trajectory Error (e_{ate}) is calculated. RT (dashed) denotes hard-enforced real-time execution.

Predicted DSO Depth Maps

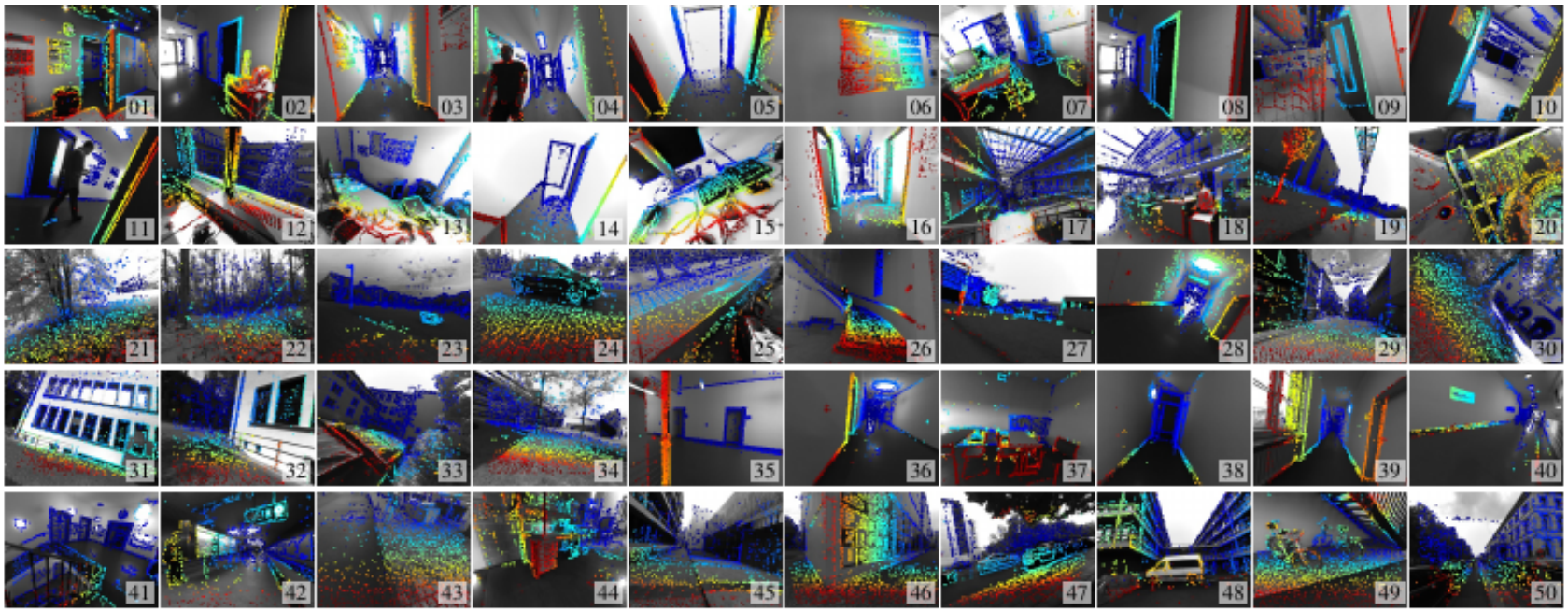


Figure 9: Results on the *TUM mono-VO* dataset. Depth maps have been overlayed on the original images.

3D Reconstruction

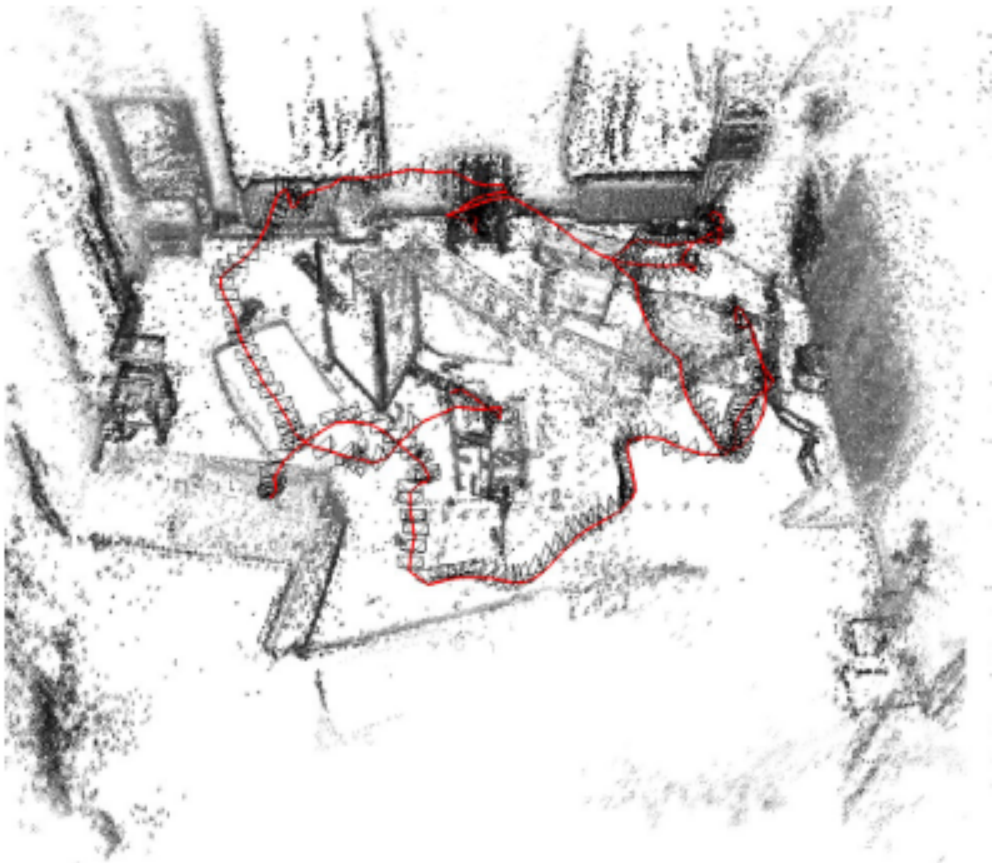


Figure 10: 3D Reconstruction of the environment along with the trajectory of the camera (red line).

3D Reconstruction

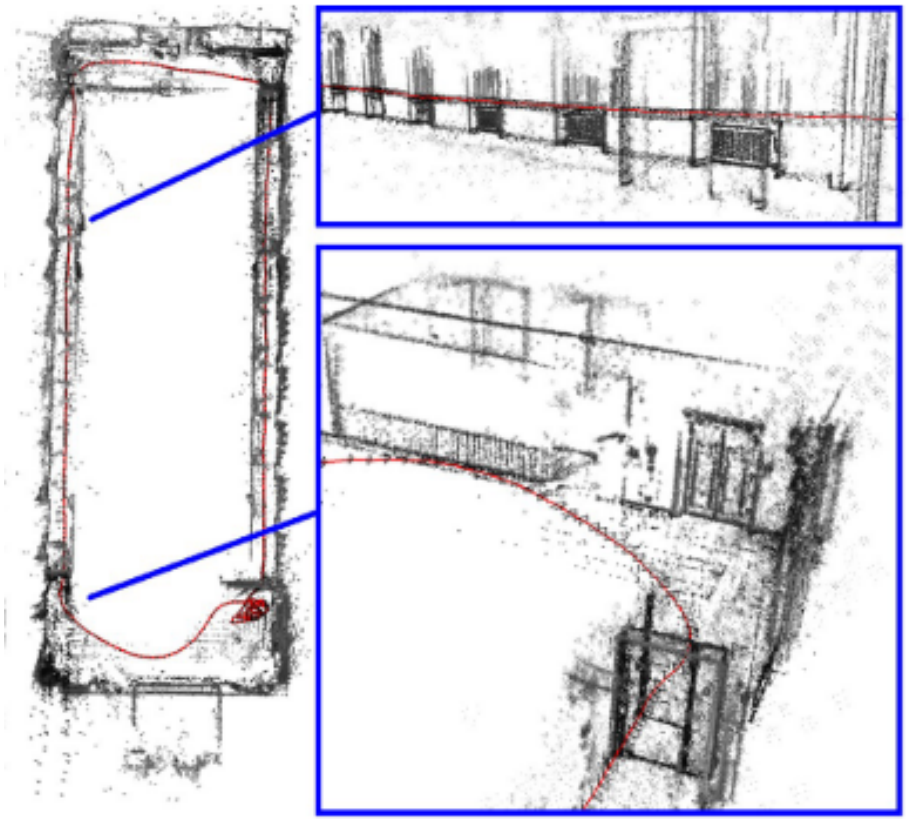


Figure 11: 3D Reconstruction of the environment along with the trajectory of the camera (red line).

3D Reconstruction

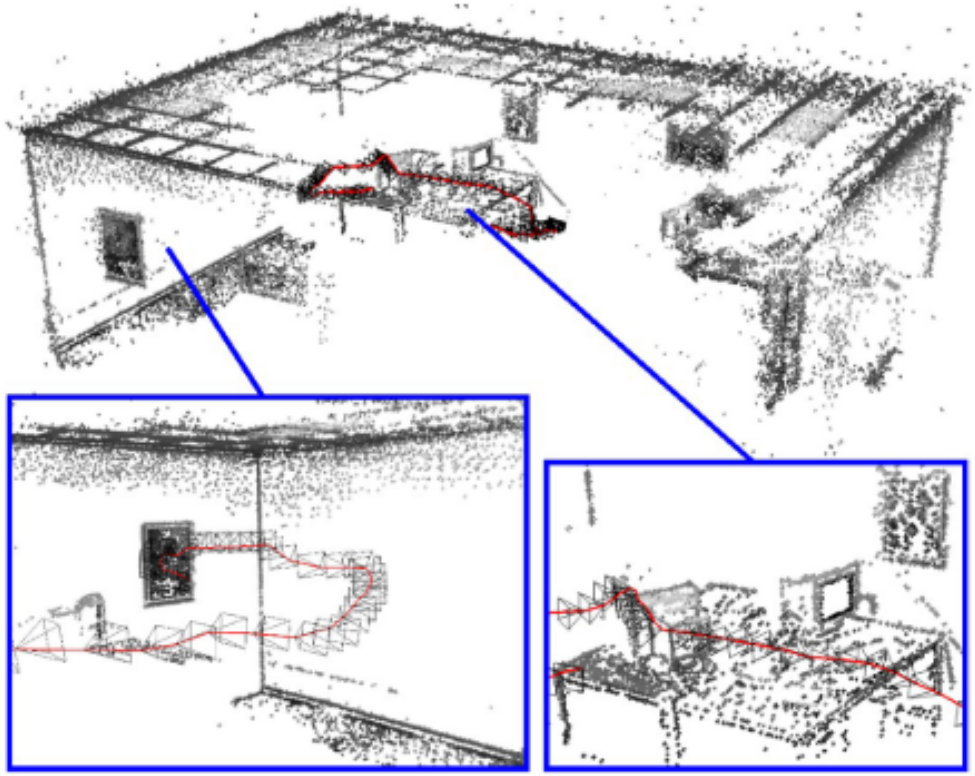


Figure 12: 3D Reconstruction of the environment along with the trajectory of the camera. Here, it is observed that some scenes contain very little texture, making them very challenging for indirect approaches.

Conclusion

- The paper presents a novel direct and sparse formulation for SLAM by combining the benefits of direct methods with the flexibility of sparse approaches.
- A comprehensive evaluation on several hours of video shows the superiority of the presented formulation relative to state-of-the-art direct and indirect methods.

Questions?