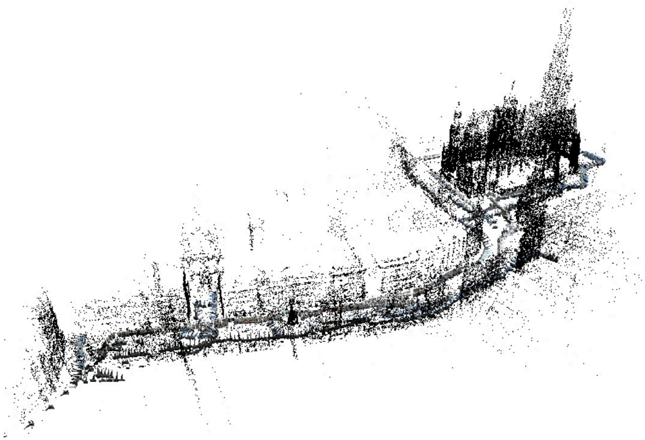
Computer Vision: Lecture 11

2023-12-04

Planned contents

Week 1	Intro, camera model	Projective geometry
Week 2	Camera calibration, DLT I	DLT II, feature matching
Week 3	Two-view geometry I	Two-view geometry II
Week 4	Robust estimation	Minimal solvers & degeneracies
Week 5	MLE & Non-linear opt.	Non-seq. SfM & project pres.
Week 6	Bundle adjustment & uncertainty	Factorization methods
Week 7	Non-rigid SfM (guest)	Dense reconstruction

Today's Lecture: Bundle Adjustment



Minimizing the Reprojection Error

Main goal

Choose $\psi(\cdot) = \|\cdot\|^2$. For given $\{\mathbf{x}_{ij}\}$ and $\{m_{ij}\}$ find a minimizer

$$\sum\nolimits_{i,j} m_{ij} \left\|\mathbf{x}_{ij} - \pi(\mathtt{P}_i \mathbf{X}_j)\right\|^2 \to \min_{\{\mathtt{P}_i\},\{\mathbf{X}_j\}}$$

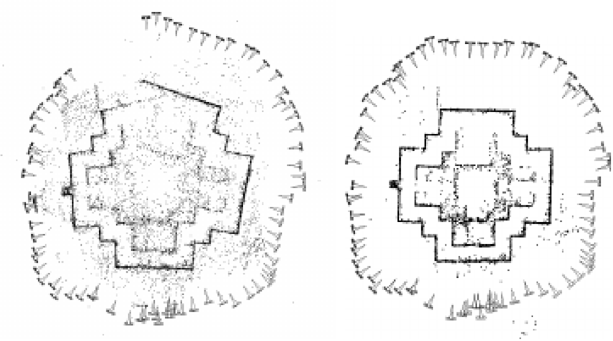
Local optimization needs good starting point

Why is this objective hard to minimize?

- Perspective division $\pi(\mathbf{X}) = \frac{1}{X_3}(X_1, X_2)^\top$
- Bilinear, non-convex terms $\mathtt{P}_i \mathbf{X}_j$
- Calibrated SfM: constraints $\mathtt{R}_i \in SO(3)$

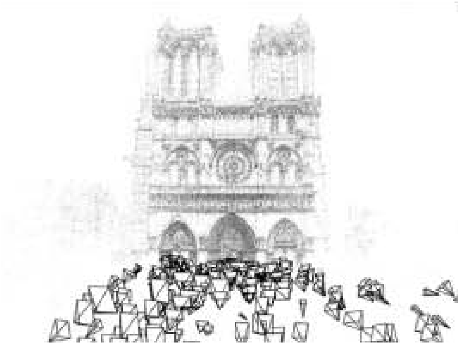
Recap from Previous Lecture(s)

Why non-sequential SfM: drift



Recap from Previous Lecture(s)

Why non-sequential SfM: more efficient for unstructured image collections



Recap from Previous Lecture(s)

Non-sequential SfM

Leverage pairwise relative orientations $(\mathtt{R}_{ij} \mid \mathbf{T}_{ij})$.
Example: rotation averaging followed by translation registration.

Rotation averaging

For given relative rotation matrices $\{\mathtt{R}_{ij}\}$ determine $\{\mathtt{R}_i\}$ such that

$$\mathtt{R}_{ij} \approx \mathtt{R}_j \mathtt{R}_i^\top \qquad \text{or} \qquad \mathtt{R}_{ij}\mathtt{R}_i - \mathtt{R}_j \approx 0$$

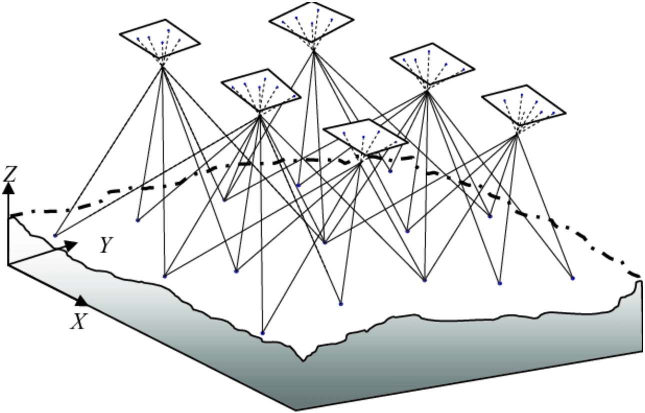
Translation registration

For fixed rotation matrices $\{\mathtt{R}_i\}$ determine translations $\{\mathbf{T}_i\}$ and scene points $\{\mathbf{X}_j\}$ such that

$$\mathbf{x}_{ij} \approx \pi(\mathtt{R}_i \mathbf{X}_j + \mathbf{T}_i)$$

Bundle Adjustment

- Term originates in photogrammetry
- Sometimes called “bundle block adjustment”



Bundle Adjustment

Bundle adjustment

For given $\{\mathbf{x}_{ij}\}$ and $\{m_{ij}\}$ use local optimization to find a solution of

$$\sum\nolimits_{i,j} m_{ij} \left\| \mathbf{x}_{ij} - \pi(\mathtt{P}_i \mathbf{X}_j) \right\|^2 \to \min_{\{\mathtt{P}_i\},\{\mathbf{X}_j\}} .$$

Maximum likelihood estimate for $\{\mathtt{P}_i\}$ and $\{\mathbf{X}_j\}$.

- Assumes good initial solution for $\{\mathtt{P}_i\}$ and $\{\mathbf{X}_j\}$ given
- What is a good (efficient) algorithm?
  - 1000 cameras/images $+\ 100\,000$ scene points $\implies \approx 300\,000$ unknowns
  - Special problem structure needs to be leveraged

Bundle Adjustment

Local optimization

Find $x^* = \arg\min_x F(x)$.

- Gradient descent

$$x^{(t+1)} \leftarrow x^{(t)} - \alpha^{(t)} \nabla_x F(x^{(t)})$$

  Very slow in practice for BA

- Newton method: Taylor expansion at $x^{(t)}$

$$F(x^{(t)} + \delta) \approx F(x^{(t)}) + \nabla_x F(x^{(t)})^\top \delta + \frac{1}{2}\delta^\top \mathtt{H}_F(x^{(t)})\delta$$

  Optimize quadratic model:

$$x^{(t+1)} \leftarrow x^{(t)} - \mathtt{H}_F(x^{(t)})^{-1} \nabla_x F(x^{(t)})$$

  Needs $\mathtt{H}_F(x^{(t)})$; unstable (may diverge); indefinite $\mathtt{H}_F(x^{(t)})$

- Quasi-Newton methods (BFGS, L-BFGS)
  Too slow in practice for BA (similar to GD)

Bundle Adjustment

Local optimization

Find $x^* = \arg\min_x F(x)$.

- Gradient descent

$$x^{(t+1)} \leftarrow x^{(t)} - \alpha^{(t)} \nabla_x F(x^{(t)})$$

  Very slow in practice for BA

- Newton method: Taylor expansion at $x^{(t)}$

$$F(x^{(t)} + \delta) \approx F(x^{(t)}) + \nabla_x F(x^{(t)})^\top \delta + \tfrac{1}{2}\delta^\top \mathtt{H}_F(x^{(t)})\delta$$

  Optimize quadratic model:

$$x^{(t+1)} \leftarrow x^{(t)} - \mathtt{H}_F(x^{(t)})^{-1} \nabla_x F(x^{(t)})$$

  Needs $\mathtt{H}_F(x^{(t)})$; unstable (may diverge); indefinite $\mathtt{H}_F(x^{(t)})$

- Quasi-Newton methods (BFGS, L-BFGS)
  Too slow in practice for BA (similar to GD)

Bundle Adjustment

Local optimization

Find $x^* = \arg\min_x F(x)$.

- Gradient descent

$$x^{(t+1)} \leftarrow x^{(t)} - \alpha^{(t)} \nabla_x F(x^{(t)})$$

  Very slow in practice for BA

- Newton method: Taylor expansion at $x^{(t)}$

$$F(x^{(t)} + \delta) \approx F(x^{(t)}) + \nabla_x F(x^{(t)})^\top \delta + \tfrac{1}{2} \delta^\top \mathtt{H}_F(x^{(t)}) \delta$$

  Optimize quadratic model:

$$x^{(t+1)} \leftarrow x^{(t)} - \mathtt{H}_F(x^{(t)})^{-1} \nabla_x F(x^{(t)})$$

  Needs $\mathtt{H}_F(x^{(t)})$; unstable (may diverge); indefinite $\mathtt{H}_F(x^{(t)})$

- Quasi-Newton methods (BFGS, L-BFGS)
  Too slow in practice for BA (similar to GD)

Bundle Adjustment

Non-linear least-squares optimization

Find $x^* = \arg\min_x F(x) = \arg\min_x \frac{1}{2}\sum_k f_k(x)^2 = \arg\min_x \frac{1}{2}\|\mathbf{f}(x)\|^2$.

- Gradient and Hessian

$$\frac{d}{dx}F(x) = \mathbf{f}(x)^\top \overbrace{\frac{d}{dx}\mathbf{f}(x)}^{=\mathtt{J}} = \mathbf{f}(x)^\top \mathtt{J} \qquad \mathtt{H}_F(x) = \mathtt{J}^\top\mathtt{J} + \sum_k f_k(x)^\top \mathtt{H}_{f_k}(x)$$

- Gauss-Newton approximation: drop 2nd order terms in $\mathtt{H}_F$

$$\mathtt{H}_F(x) \approx \mathtt{J}^\top\mathtt{J} \qquad F(x+\delta) \approx F(x) + \mathbf{f}(x)^\top\mathtt{J}\delta + \tfrac{1}{2}\delta^\top\mathtt{J}^\top\mathtt{J}\delta$$

  Note: $\mathtt{J}^\top\mathtt{J}$ is always p.s.d.
- Alternative derivation: linearize $\mathbf{f}(x)$

$$\|\mathbf{f}(x+\delta)\|^2 \approx \|\mathbf{f}(x) + \mathtt{J}\delta\|^2$$

- Optimal update δ

$$\delta = -(\mathtt{J}^\top\mathtt{J})^{-1}\mathtt{J}^\top\mathbf{f}(x)$$

- Large steps for δ

Bundle Adjustment

Non-linear least-squares optimization

Find $x^* = \arg\min_x F(x) = \arg\min_x \frac{1}{2} \sum_k f_k(x)^2 = \arg\min_x \frac{1}{2} \|\mathbf{f}(x)\|^2$.

- Gradient and Hessian

$$\frac{d}{dx}F(x) = \mathbf{f}(x)^\top \overbrace{\frac{d}{dx}\mathbf{f}(x)}^{=\mathtt{J}} = \mathbf{f}(x)^\top \mathtt{J} \quad \mathtt{H}_F(x) = \mathtt{J}^\top \mathtt{J} + \sum_k f_k(x)^\top \mathtt{H}_{f_k}(x)$$

- Gauss-Newton approximation: drop 2nd order terms in $\mathtt{H}_F$

$$\mathtt{H}_F(x) \approx \mathtt{J}^\top \mathtt{J} \qquad F(x+\delta) \approx F(x) + \mathbf{f}(x)^\top \mathtt{J}\delta + \frac{1}{2}\delta^\top \mathtt{J}^\top \mathtt{J}\delta$$

  Note: $\mathtt{J}^\top \mathtt{J}$ is always p.s.d.

- Alternative derivation: linearize $\mathbf{f}(x)$

$$\|\mathbf{f}(x+\delta)\|^2 \approx \|\mathbf{f}(x) + \mathtt{J}\delta\|^2$$

- Optimal update δ

$$\delta = -(\mathtt{J}^\top \mathtt{J})^{-1} \mathtt{J}^\top \mathbf{f}(x)$$

- Large steps for δ

Bundle Adjustment

Non-linear least-squares optimization

Find $x^* = \arg\min_x F(x) = \arg\min_x \frac{1}{2}\sum_k f_k(x)^2 = \arg\min_x \frac{1}{2}\|\mathbf{f}(x)\|^2$.

- Gradient and Hessian

$$\frac{d}{dx}F(x) = \mathbf{f}(x)^\top \overbrace{\frac{d}{dx}\mathbf{f}(x)}^{=\mathtt{J}} = \mathbf{f}(x)^\top \mathtt{J} \quad \mathtt{H}_F(x) = \mathtt{J}^\top\mathtt{J} + \sum\nolimits_k f_k(x)^\top \mathtt{H}_{f_k}(x)$$

- Gauss-Newton approximation: drop 2nd order terms in $\mathtt{H}_F$

$$\mathtt{H}_F(x) \approx \mathtt{J}^\top\mathtt{J} \qquad\qquad F(x+\delta) \approx F(x) + \mathbf{f}(x)^\top\mathtt{J}\delta + \frac{1}{2}\delta^\top\mathtt{J}^\top\mathtt{J}\delta$$

  Note: $\mathtt{J}^\top\mathtt{J}$ is always p.s.d.

- Alternative derivation: linearize $\mathbf{f}(x)$

$$\|\mathbf{f}(x+\delta)\|^2 \approx \|\mathbf{f}(x) + \mathtt{J}\delta\|^2$$

- Optimal update δ

$$\delta = -(\mathtt{J}^\top\mathtt{J})^{-1}\mathtt{J}^\top\mathbf{f}(x)$$

- Large steps for δ

Bundle Adjustment

Non-linear least-squares optimization

Find $x^* = \arg\min_x F(x) = \arg\min_x \frac{1}{2} \sum_k f_k(x)^2 = \arg\min_x \frac{1}{2} \|\mathbf{f}(x)\|^2$.

- Gradient and Hessian

$$\frac{d}{dx} F(x) = \mathbf{f}(x)^\top \overbrace{\frac{d}{dx}\mathbf{f}(x)}^{=\mathtt{J}} = \mathbf{f}(x)^\top \mathtt{J} \qquad \mathtt{H}_F(x) = \mathtt{J}^\top \mathtt{J} + \sum\nolimits_k f_k(x)^\top \mathtt{H}_{f_k}(x)$$

- Gauss-Newton approximation: drop 2nd order terms in $\mathtt{H}_F$

$$\mathtt{H}_F(x) \approx \mathtt{J}^\top \mathtt{J} \qquad\qquad F(x+\delta) \approx F(x) + \mathbf{f}(x)^\top \mathtt{J}\delta + \tfrac{1}{2}\delta^\top \mathtt{J}^\top \mathtt{J}\delta$$

  Note: $\mathtt{J}^\top \mathtt{J}$ is always p.s.d.

- Alternative derivation: linearize $\mathbf{f}(x)$

$$\|\mathbf{f}(x+\delta)\|^2 \approx \|\mathbf{f}(x) + \mathtt{J}\delta\|^2$$

- Optimal update δ

$$\delta = -(\mathtt{J}^\top \mathtt{J})^{-1} \mathtt{J}^\top \mathbf{f}(x)$$

- Large steps for δ

Bundle Adjustment

Gauss-Newton method

Task: determine

$$x^* = \arg\min_x F(x) = \arg\min_x \tfrac{1}{2}\sum_k f_k(x)^2 = \arg\min_x \tfrac{1}{2}\left\|\mathbf{f}(x)\right\|^2$$

Iterate

$$x^{(t+1)} \leftarrow x^{(t)} - (\mathsf{J}^\top \mathsf{J})^{-1}\mathsf{J}^\top \mathbf{r} \qquad \mathsf{J} = \frac{d\mathbf{f}(x^{(t)})}{dx} \qquad \mathbf{r} = \mathbf{f}(x^{(t)})$$

- Problem 1: $\mathsf{J}^\top \mathsf{J}$ may be singular
- Problem 2: $x^{(t)}$ is not guaranteed to converge
- Problem 3: $\mathsf{J}^\top \mathsf{J} \in \mathbb{R}^{300\,000 \times 300\,000}$!

Bundle Adjustment

Levenberg-Marquardt method

Choose $\mu^{(0)} > 0$ and iterate

1. Solve *augmented normal equation*

$$\delta \leftarrow -\left(\mathtt{J}^\top \mathtt{J} + \mu^{(t)} \mathtt{I}\right)^{-1} \mathtt{J}^\top \mathbf{r} \qquad \mathtt{J} = \frac{d\mathbf{f}(x^{(t)})}{dx} \qquad \mathbf{r} = \mathbf{f}(x^{(t)})$$

2. If $F(x^{(t)} + \delta) < F(x^{(t)})$ then $x^{(t+1)} \leftarrow x^{(t)} + \delta$, $\mu^{(t+1)} \leftarrow \mu^{(t)}/10$
3. Otherwise $x^{(t+1)} \leftarrow x^{(t)}$, $\mu^{(t+1)} \leftarrow 10\,\mu^{(t)}$

- Solves problems 1+2
  - $\mathtt{J}^\top \mathtt{J} + \mu^{(t)} \mathtt{I}$ is always invertible
  - $(F(x^{(t)}))_{t=1}^{\infty}$ is monotonically decreasing sequence
  - Under some assumptions quadratic convergence rate
- $\mu \approx 0$: Gauss-Newton steps
- $\mu \gg 0$: gradient descent with step size $1/\mu$
- Problem 3 remains

Bundle Adjustment

Levenberg-Marquardt method

Choose $\mu^{(0)} > 0$ and iterate

1. Solve *augmented normal equation*

$$\delta \leftarrow -\left(\mathtt{J}^\top \mathtt{J} + \mu^{(t)} \mathtt{I}\right)^{-1} \mathtt{J}^\top \mathbf{r} \qquad \mathtt{J} = \frac{d\mathbf{f}(x^{(t)})}{dx} \qquad \mathbf{r} = \mathbf{f}(x^{(t)})$$

2. If $F(x^{(t)} + \delta) < F(x^{(t)})$ then $x^{(t+1)} \leftarrow x^{(t)} + \delta$, $\mu^{(t+1)} \leftarrow \mu^{(t)}/10$
3. Otherwise $x^{(t+1)} \leftarrow x^{(t)}$, $\mu^{(t+1)} \leftarrow 10\,\mu^{(t)}$

- Solves problems 1+2
  - $\mathtt{J}^\top \mathtt{J} + \mu^{(t)} \mathtt{I}$ is always invertible
  - $(F(x^{(t)}))_{t=1}^{\infty}$ is monotonically decreasing sequence
  - Under some assumptions quadratic convergence rate
- $\mu \approx 0$: Gauss-Newton steps
- $\mu \gg 0$: gradient descent with step size $1/\mu$
- Problem 3 remains

Bundle Adjustment

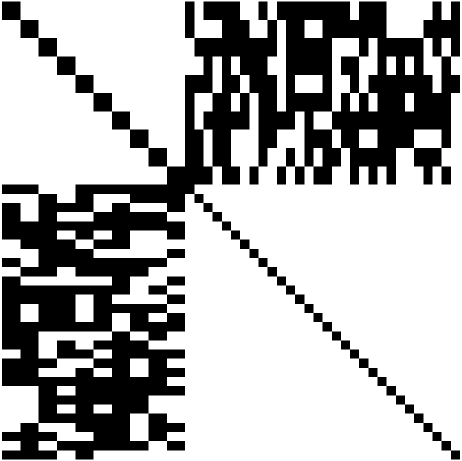
Bundle adjustment objective

$$\sum_{i,j} \underbrace{m_{ij} \left\| \mathbf{x}_{ij} - \pi(\mathtt{P}_i \mathbf{X}_j) \right\|^2}_{=f_{ij}^2} \to \min_{\{\mathtt{P}_i\},\{\mathbf{X}_j\}} .$$

- Each term f_{ij}^2 depends on one camera matrix $\mathtt{P}_i$ and one scene point $\mathbf{X}_j$
  - Bipartite dependency graph between unknowns
- Non-zero structure of $\mathtt{J}^\top$ (rows are unknowns, columns are image points)

Bundle Adjustment

- $\mathtt{J}^\top\mathtt{J}$ has very special structure



Bundle Adjustment

Solving the (augmented) normal equation

$$\delta \leftarrow -\left(\mathtt{J}^\top \mathtt{J} + \mu^{(t)} \mathtt{I}\right)^{-1} \mathtt{J}^\top \mathbf{r} \qquad \mathtt{J} = \frac{d\mathbf{f}(x^{(t)})}{dx} \qquad \mathbf{r} = \mathbf{f}(x^{(t)})$$

Sparse linear algebra routines

- Sparsity pattern of $\mathtt{J}^\top \mathtt{J}$ does not change
- Direct method
  1. Reorder columns and apply "symbolic" sparse Cholesky decomposition (once)
  2. Apply sparse Cholesky with current non-zero values
- Iterative methods
  1. Apply (preconditioned) conjugate gradient method
- Both methods can benefit from using the *Schur complement*

Bundle Adjustment

Solving the (augmented) normal equation

$$\delta \leftarrow -\left(\mathtt{J}^\top \mathtt{J} + \mu^{(t)} \mathtt{I}\right)^{-1} \mathtt{J}^\top \mathbf{r} \qquad \mathtt{J} = \frac{d\mathbf{f}(x^{(t)})}{dx} \qquad \mathbf{r} = \mathbf{f}(x^{(t)})$$

Sparse linear algebra routines

- Sparsity pattern of $\mathtt{J}^\top \mathtt{J}$ does not change
- Direct method
  1. Reorder columns and apply "symbolic" sparse Cholesky decomposition (once)
  2. Apply sparse Cholesky with current non-zero values
- Iterative methods
  1. Apply (preconditioned) conjugate gradient method
- Both methods can benefit from using the *Schur complement*

Bundle Adjustment

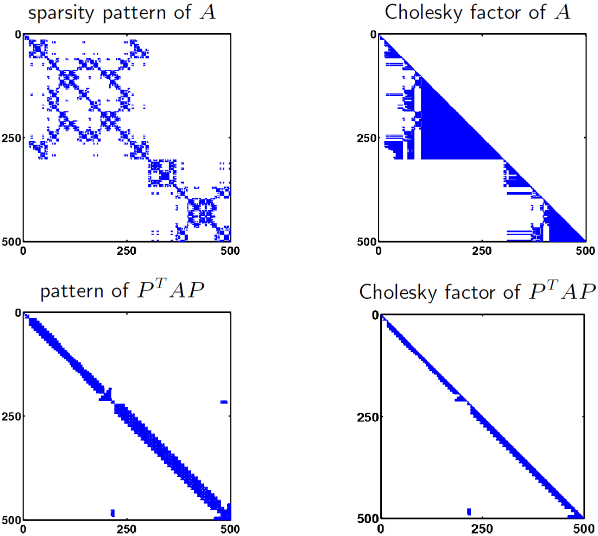
Solving the (augmented) normal equation

$$\delta \leftarrow -\left(\mathtt{J}^\top \mathtt{J} + \mu^{(t)} \mathtt{I}\right)^{-1} \mathtt{J}^\top \mathbf{r} \qquad \mathtt{J} = \frac{d\mathbf{f}(x^{(t)})}{dx} \qquad \mathbf{r} = \mathbf{f}(x^{(t)})$$

Sparse linear algebra routines

- Sparsity pattern of $\mathtt{J}^\top \mathtt{J}$ does not change
- Direct method
  1. Reorder columns and apply "symbolic" sparse Cholesky decomposition (once)
  2. Apply sparse Cholesky with current non-zero values
- Iterative methods
  1. Apply (preconditioned) conjugate gradient method
- Both methods can benefit from using the *Schur complement*

Bundle Adjustment



Cholesky decomposition of a sparse matrix without and with column reordering

Bundle Adjustment

Schur complement

Goal: solve

$$\begin{pmatrix} \mathtt{A} & \mathtt{B} \\ \mathtt{B}^\top & \mathtt{C} \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} = \begin{pmatrix} \mathbf{a} \\ \mathbf{b} \end{pmatrix} \iff \begin{pmatrix} \mathtt{A}\mathbf{x} + \mathtt{B}\mathbf{y} \\ \mathtt{B}^\top\mathbf{x} + \mathtt{C}\mathbf{y} \end{pmatrix} = \begin{pmatrix} \mathbf{a} \\ \mathbf{b} \end{pmatrix}$$

Reduce size of the problem by solving

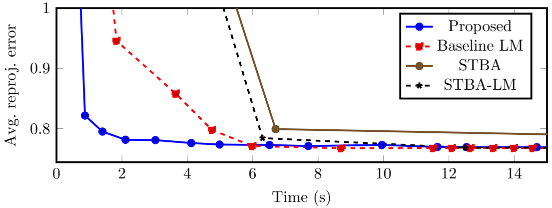
$$\begin{aligned} \left(\mathtt{A} - \mathtt{B}\mathtt{C}^{-1}\mathtt{B}^\top\right)\mathbf{x} &= \mathbf{a} - \mathtt{B}\mathtt{C}^{-1}\mathbf{b} \\ \mathtt{C}\mathbf{y} &= \mathbf{b} - \mathtt{B}^\top\mathbf{x} \end{aligned}$$

$$\mathtt{J}^\top\mathtt{J} = \begin{pmatrix} \mathtt{A} & \mathtt{B} \\ \mathtt{B}^\top & \mathtt{C} \end{pmatrix}$$

Bundle Adjustment

Use of Schur complement in bundle adjustment

1. Partition all unknowns into cameras and 3D points
2. C is block-diagonal
   - 3×3 blocks on the diagonal
   - Easy to compute $\mathtt{C}^{-1}$
3. Form the Schur complement and solve for camera matrix updates
   - 1000 cameras, $100\,000$ 3D points: 6000×6000 system matrix
   - Preconditioned CG or sparse Cholesky
4. Back-substitute to solve for 3D point updates



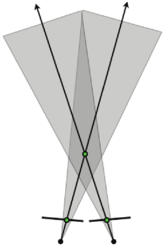
427 cameras, $\approx 310,000$ points

Bundle Adjustment and Uncertainty

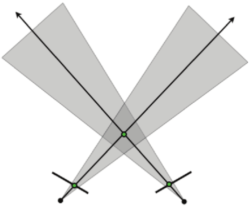
Question

How sensitive is the solution $\{\mathtt{P}_i\}$, $\{\mathbf{X}_j\}$ to different realizations of $\{\mathbf{x}_{ij}\}$?

- Image points are corrupted by (Gaussian) noise
- Running feature extraction again should lead to different noise realizations
  - At least according to the assumed noise model
- Feature uncertainty implies uncertainty in our maximum likelihood estimates
- Intuition for 3D points (analogous for camera matrices)
  - If X_j is seen in many images, different noise realizations will have little impact
  - Almost parallel rays will lead to large uncertainty in the depth direction



(a) Small baseline



(b) Large baseline

Bundle Adjustment and Uncertainty

Quadratic model

We collect $\mathcal{P} = (\mathtt{P}_i)_{i=1}^N$ and $\mathcal{X} = (\mathbf{X}_j)_{j=1}^M$. Consider the quadratic model at a local minimum $(\mathcal{P}^*, \mathcal{X}^*)$:

$$\frac{1}{2}\begin{pmatrix}\mathcal{P}-\mathcal{P}^* \\ \mathcal{X}-\mathcal{X}^*\end{pmatrix}^\top \begin{pmatrix}\mathtt{A} & \mathtt{B} \\ \mathtt{B}^\top & \mathtt{C}\end{pmatrix}\begin{pmatrix}\mathcal{P}-\mathcal{P}^* \\ \mathcal{X}-\mathcal{X}^*\end{pmatrix} + F(\mathcal{P}^*, \mathcal{X}^*)$$

- Note: $(\mathcal{P}^*, \mathcal{X}^*)$ is a local minimum $\implies$ gradient vanishes

$$\mathtt{J}^\top \mathbf{r} = \mathbf{0}$$

- Quadratic model can be interpreted as a multivariate Gaussian
  - Mean/mode: $(\mathcal{P}^*, \mathcal{X}^*)$
  - Precision matrix: $\mathtt{J}^\top\mathtt{J} = \begin{pmatrix}\mathtt{A} & \mathtt{B} \\ \mathtt{B}^\top & \mathtt{C}\end{pmatrix}$
  - Covariance matrix: $\Sigma = (\mathtt{J}^\top\mathtt{J})^{-1}$

Bundle Adjustment and Uncertainty

Quadratic model

We collect $\mathcal{P} = (\mathtt{P}_i)_{i=1}^N$ and $\mathcal{X} = (\mathbf{X}_j)_{j=1}^M$. Consider the quadratic model at a local minimum $(\mathcal{P}^*, \mathcal{X}^*)$:

$$\frac{1}{2}\begin{pmatrix}\mathcal{P}-\mathcal{P}^* \\ \mathcal{X}-\mathcal{X}^*\end{pmatrix}^\top \begin{pmatrix}\mathtt{A} & \mathtt{B} \\ \mathtt{B}^\top & \mathtt{C}\end{pmatrix}\begin{pmatrix}\mathcal{P}-\mathcal{P}^* \\ \mathcal{X}-\mathcal{X}^*\end{pmatrix} + F(\mathcal{P}^*, \mathcal{X}^*)$$

- Note: $(\mathcal{P}^*, \mathcal{X}^*)$ is a local minimum $\implies$ gradient vanishes

$$\mathtt{J}^\top \mathbf{r} = \mathbf{0}$$

- Quadratic model can be interpreted as a multivariate Gaussian
  - Mean/mode: $(\mathcal{P}^*, \mathcal{X}^*)$
  - Precision matrix: $\mathtt{J}^\top\mathtt{J} = \begin{pmatrix}\mathtt{A} & \mathtt{B} \\ \mathtt{B}^\top & \mathtt{C}\end{pmatrix}$
  - Covariance matrix: $\Sigma = (\mathtt{J}^\top\mathtt{J})^{-1}$

Bundle Adjustment and Uncertainty

Some facts about the multivariate Gaussian distribution

- $\mathbf{x} \sim \mathcal{N}(\bar{\mathbf{x}}, \Sigma)$. $\mathtt{\Lambda} = \Sigma^{-1}$ is called *precision matrix*.
- Affine transformation: let $\mathbf{x} \sim \mathcal{N}(\bar{\mathbf{x}}, \Sigma)$. Then

$$\mathtt{A}\mathbf{x} + \mathbf{b} \sim \mathcal{N}(\mathtt{A}\bar{\mathbf{x}} + \mathbf{b}, \mathtt{A}\Sigma\mathtt{A}^\top).$$

- Marginal distribution: let

$$\begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \bar{\mathbf{x}}_1 \\ \bar{\mathbf{x}}_2 \end{pmatrix}, \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{12}^\top & \Sigma_{22} \end{pmatrix} \right).$$

  Then

$$p(\mathbf{x}_1) = \int p(\mathbf{x}_1, \mathbf{x}_2)\, d\mathbf{x}_2 = \mathcal{N}(\bar{\mathbf{x}}_1, \Sigma_{11}).$$

- Conditional distribution:

$$p(\mathbf{x}_1 | \mathbf{x}_2 = \mathbf{b}) = \mathcal{N}\left(\bar{\mathbf{x}}_1 - \Sigma_{12}\Sigma_{22}^{-1}(\mathbf{b} - \bar{\mathbf{x}}_2), \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{12}^\top \right)$$

Bundle Adjustment and Uncertainty

Schur complement and the multivariate Gaussian distribution

Marginal distribution: let

$$\begin{pmatrix}\mathbf{x}_1 \\ \mathbf{x}_2\end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix}\bar{\mathbf{x}}_1 \\ \bar{\mathbf{x}}_2\end{pmatrix}, \begin{pmatrix}\Sigma_{11} & \Sigma_{12} \\ \Sigma_{12}^\top & \Sigma_{22}\end{pmatrix}\right).$$

Then

$$p(\mathbf{x}_1) = \int p(\mathbf{x}_1, \mathbf{x}_2)\, d\mathbf{x}_2 = \mathcal{N}(\bar{\mathbf{x}}_1, \Sigma_{11}).$$

- What is Σ_{11} in terms of the precision matrix Λ? Block matrix inversion:

$$\begin{pmatrix}\Sigma_{11} & \Sigma_{12} \\ \Sigma_{12}^\top & \Sigma_{22}\end{pmatrix} = \begin{pmatrix}\Lambda_{11} & \Lambda_{12} \\ \Lambda_{12}^\top & \Lambda_{22}\end{pmatrix}^{-1} = \begin{pmatrix}(\Lambda_{11} - \Lambda_{12}\Lambda_{22}^{-1}\Lambda_{12}^\top)^{-1} & \Sigma_{12} \\ \Sigma_{12}^\top & \Sigma_{22}\end{pmatrix}$$

- Σ_{11} is the inverse of the Schur complement $\Lambda_{11} - \Lambda_{12}\Lambda_{22}^{-1}\Lambda_{12}^\top$
- Schur complement $\Lambda_{11} - \Lambda_{12}\Lambda_{22}^{-1}\Lambda_{12}^\top$ is the precision matrix of the marginal distribution

Bundle Adjustment and Uncertainty

Application to bundle adjustment setting:

- $(\mathcal{P}^*, \mathcal{X}^*)$ is stationary point $\implies \delta^* = -(\mathtt{J}^\top \mathtt{J})^{-1}\mathtt{J}^\top \mathbf{r} = \mathbf{0}$ (since $\mathtt{J}^\top \mathbf{r} = \mathbf{0}$)
- Add noise to residuals $\mathbf{r}$: $\varepsilon \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathtt{I})$

$$\mathbf{r} = \begin{pmatrix} \vdots \\ \mathbf{x}_{ij} - \pi(\mathtt{P}_i^* \mathbf{X}_j^*) \\ \vdots \end{pmatrix} \qquad \mathbf{r} + \varepsilon = \begin{pmatrix} \vdots \\ \mathbf{x}_{ij} + \varepsilon_{ij} - \pi(\mathtt{P}_i^* \mathbf{X}_j^*) \\ \vdots \end{pmatrix}$$

- How does $(\mathcal{P}, \mathcal{X})$ change for perturbed residuals $\mathbf{r} + \varepsilon$?

  $\delta = \binom{\mathcal{P}}{\mathcal{X}} - \binom{\mathcal{P}^*}{\mathcal{X}^*}$ is linear transformation of ε:

$$\delta = -(\mathtt{J}^\top \mathtt{J})^{-1}\mathtt{J}^\top(\mathbf{r} + \varepsilon) = \overbrace{-(\mathtt{J}^\top \mathtt{J})^{-1}\mathtt{J}^\top}^{=\mathtt{A}} \varepsilon$$

$$\delta \sim \mathcal{N}\left(\mathbf{0}, \sigma^2 (\mathtt{J}^\top \mathtt{J})^{-1}\mathtt{J}^\top \mathtt{J}(\mathtt{J}^\top \mathtt{J})^{-1}\right) = \mathcal{N}\left(\mathbf{0}, \sigma^2 (\mathtt{J}^\top \mathtt{J})^{-1}\right)$$

- $(\mathcal{P}, \mathcal{X})$ has mean $(\mathcal{P}^*, \mathcal{X}^*)$ and the precision matrix

$$\Lambda = \frac{1}{\sigma^2}\mathtt{J}^\top \mathtt{J} \propto \mathtt{J}^\top \mathtt{J}$$

Bundle Adjustment and Uncertainty

Application to bundle adjustment setting:

- $(\mathcal{P}^*, \mathcal{X}^*)$ is stationary point $\implies \delta^* = -(\mathtt{J}^\top\mathtt{J})^{-1}\mathtt{J}^\top\mathbf{r} = \mathbf{0}$ (since $\mathtt{J}^\top\mathbf{r} = \mathbf{0}$)
- Add noise to residuals $\mathbf{r}$: $\varepsilon \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathtt{I})$

$$\mathbf{r} = \begin{pmatrix} \vdots \\ \mathbf{x}_{ij} - \pi(\mathtt{P}_i^*\mathbf{X}_j^*) \\ \vdots \end{pmatrix} \qquad \mathbf{r} + \varepsilon = \begin{pmatrix} \vdots \\ \mathbf{x}_{ij} + \varepsilon_{ij} - \pi(\mathtt{P}_i^*\mathbf{X}_j^*) \\ \vdots \end{pmatrix}$$

- How does $(\mathcal{P}, \mathcal{X})$ change for perturbed residuals $\mathbf{r} + \varepsilon$?

  $\delta = \binom{\mathcal{P}}{\mathcal{X}} - \binom{\mathcal{P}^*}{\mathcal{X}^*}$ is linear transformation of ε:

$$\delta = -(\mathtt{J}^\top\mathtt{J})^{-1}\mathtt{J}^\top(\mathbf{r} + \varepsilon) = \overbrace{-(\mathtt{J}^\top\mathtt{J})^{-1}\mathtt{J}^\top}^{=\mathtt{A}}\varepsilon$$

$$\delta \sim \mathcal{N}\left(\mathbf{0}, \sigma^2(\mathtt{J}^\top\mathtt{J})^{-1}\mathtt{J}^\top\mathtt{J}(\mathtt{J}^\top\mathtt{J})^{-1}\right) = \mathcal{N}\left(\mathbf{0}, \sigma^2(\mathtt{J}^\top\mathtt{J})^{-1}\right)$$

- $(\mathcal{P}, \mathcal{X})$ has mean $(\mathcal{P}^*, \mathcal{X}^*)$ and the precision matrix

$$\Lambda = \frac{1}{\sigma^2}\mathtt{J}^\top\mathtt{J} \propto \mathtt{J}^\top\mathtt{J}$$

Bundle Adjustment and Uncertainty

Application to bundle adjustment setting:

- $(\mathcal{P}^*, \mathcal{X}^*)$ is stationary point $\implies \delta^* = -(\mathtt{J}^\top \mathtt{J})^{-1}\mathtt{J}^\top \mathbf{r} = \mathbf{0}$ (since $\mathtt{J}^\top \mathbf{r} = \mathbf{0}$)
- Add noise to residuals $\mathbf{r}$: $\varepsilon \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathtt{I})$

$$\mathbf{r} = \begin{pmatrix} \vdots \\ \mathbf{x}_{ij} - \pi(\mathtt{P}_i^* \mathbf{X}_j^*) \\ \vdots \end{pmatrix} \qquad \mathbf{r} + \varepsilon = \begin{pmatrix} \vdots \\ \mathbf{x}_{ij} + \varepsilon_{ij} - \pi(\mathtt{P}_i^* \mathbf{X}_j^*) \\ \vdots \end{pmatrix}$$

- How does $(\mathcal{P}, \mathcal{X})$ change for perturbed residuals $\mathbf{r} + \varepsilon$?

  $\delta = \binom{\mathcal{P}}{\mathcal{X}} - \binom{\mathcal{P}^*}{\mathcal{X}^*}$ is linear transformation of ε:

$$\delta = -(\mathtt{J}^\top \mathtt{J})^{-1}\mathtt{J}^\top(\mathbf{r} + \varepsilon) = \overbrace{-(\mathtt{J}^\top \mathtt{J})^{-1}\mathtt{J}^\top}^{=\mathtt{A}} \varepsilon$$

$$\delta \sim \mathcal{N}\left(\mathbf{0}, \sigma^2(\mathtt{J}^\top \mathtt{J})^{-1}\mathtt{J}^\top \mathtt{J}(\mathtt{J}^\top \mathtt{J})^{-1}\right) = \mathcal{N}\left(\mathbf{0}, \sigma^2(\mathtt{J}^\top \mathtt{J})^{-1}\right)$$

- $(\mathcal{P}, \mathcal{X})$ has mean $(\mathcal{P}^*, \mathcal{X}^*)$ and the precision matrix

$$\Lambda = \frac{1}{\sigma^2}\mathtt{J}^\top \mathtt{J} \propto \mathtt{J}^\top \mathtt{J}$$

Bundle Adjustment and Uncertainty

Application to bundle adjustment setting:

- $(\mathcal{P}^*, \mathcal{X}^*)$ is stationary point $\implies \delta^* = -(\mathtt{J}^\top \mathtt{J})^{-1}\mathtt{J}^\top \mathbf{r} = \mathbf{0}$ (since $\mathtt{J}^\top \mathbf{r} = \mathbf{0}$)
- Add noise to residuals $\mathbf{r}$: $\varepsilon \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathtt{I})$

$$\mathbf{r} = \begin{pmatrix} \vdots \\ \mathbf{x}_{ij} - \pi(\mathtt{P}_i^* \mathbf{X}_j^*) \\ \vdots \end{pmatrix} \qquad \mathbf{r} + \varepsilon = \begin{pmatrix} \vdots \\ \mathbf{x}_{ij} + \varepsilon_{ij} - \pi(\mathtt{P}_i^* \mathbf{X}_j^*) \\ \vdots \end{pmatrix}$$

- How does $(\mathcal{P}, \mathcal{X})$ change for perturbed residuals $\mathbf{r} + \varepsilon$?

  $\delta = \binom{\mathcal{P}}{\mathcal{X}} - \binom{\mathcal{P}^*}{\mathcal{X}^*}$ is linear transformation of ε:

$$\delta = -(\mathtt{J}^\top \mathtt{J})^{-1}\mathtt{J}^\top(\mathbf{r} + \varepsilon) = \overbrace{-(\mathtt{J}^\top \mathtt{J})^{-1}\mathtt{J}^\top}^{=\mathtt{A}} \varepsilon$$

$$\delta \sim \mathcal{N}\left(\mathbf{0}, \sigma^2 (\mathtt{J}^\top \mathtt{J})^{-1}\mathtt{J}^\top \mathtt{J}(\mathtt{J}^\top \mathtt{J})^{-1}\right) = \mathcal{N}\left(\mathbf{0}, \sigma^2 (\mathtt{J}^\top \mathtt{J})^{-1}\right)$$

- $(\mathcal{P}, \mathcal{X})$ has mean $(\mathcal{P}^*, \mathcal{X}^*)$ and the precision matrix

$$\Lambda = \frac{1}{\sigma^2}\mathtt{J}^\top \mathtt{J} \propto \mathtt{J}^\top \mathtt{J}$$

Bundle Adjustment and Uncertainty

Application to bundle adjustment setting:

- $(\mathcal{P}^*, \mathcal{X}^*)$ is stationary point $\implies \delta^* = -(\mathtt{J}^\top \mathtt{J})^{-1}\mathtt{J}^\top \mathbf{r} = \mathbf{0}$ (since $\mathtt{J}^\top \mathbf{r} = \mathbf{0}$)
- Add noise to residuals $\mathbf{r}$: $\varepsilon \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathtt{I})$

$$\mathbf{r} = \begin{pmatrix} \vdots \\ \mathbf{x}_{ij} - \pi(\mathtt{P}_i^* \mathbf{X}_j^*) \\ \vdots \end{pmatrix} \qquad \mathbf{r} + \varepsilon = \begin{pmatrix} \vdots \\ \mathbf{x}_{ij} + \varepsilon_{ij} - \pi(\mathtt{P}_i^* \mathbf{X}_j^*) \\ \vdots \end{pmatrix}$$

- How does $(\mathcal{P}, \mathcal{X})$ change for perturbed residuals $\mathbf{r} + \varepsilon$?

  $\delta = \binom{\mathcal{P}}{\mathcal{X}} - \binom{\mathcal{P}^*}{\mathcal{X}^*}$ is linear transformation of ε:

$$\delta = -(\mathtt{J}^\top \mathtt{J})^{-1}\mathtt{J}^\top(\mathbf{r} + \varepsilon) = \overbrace{-(\mathtt{J}^\top \mathtt{J})^{-1}\mathtt{J}^\top}^{=\mathtt{A}} \varepsilon$$

$$\delta \sim \mathcal{N}\left(\mathbf{0}, \sigma^2 (\mathtt{J}^\top \mathtt{J})^{-1}\mathtt{J}^\top \mathtt{J}(\mathtt{J}^\top \mathtt{J})^{-1}\right) = \mathcal{N}\left(\mathbf{0}, \sigma^2 (\mathtt{J}^\top \mathtt{J})^{-1}\right)$$

- $(\mathcal{P}, \mathcal{X})$ has mean $(\mathcal{P}^*, \mathcal{X}^*)$ and the precision matrix

$$\Lambda = \frac{1}{\sigma^2}\mathtt{J}^\top \mathtt{J} \propto \mathtt{J}^\top \mathtt{J}$$

Bundle Adjustment and Uncertainty

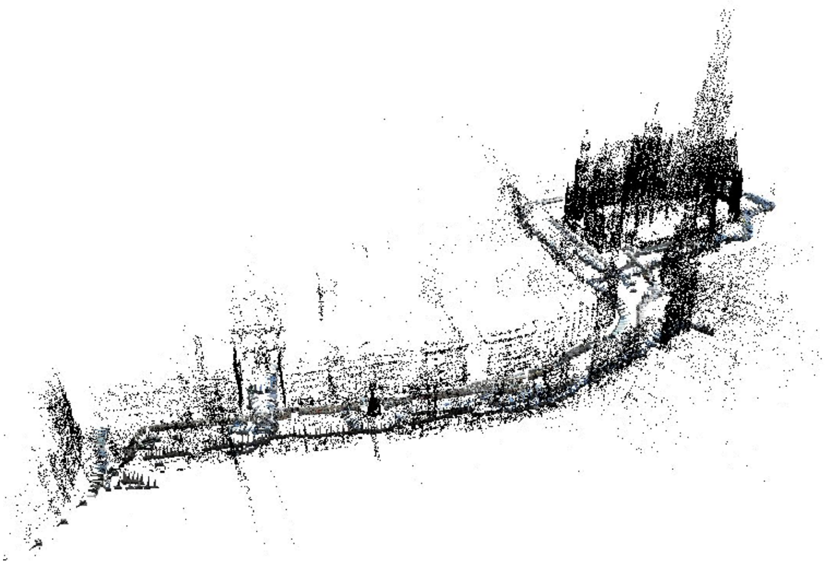
Summary

- Minimum $(\mathcal{P}, \mathcal{X})$ is approximately normally distributed:

$$\begin{pmatrix} \mathcal{P} \\ \mathcal{X} \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \mathcal{P}^* \\ \mathcal{X}^* \end{pmatrix}; \Sigma \right) \qquad \Sigma = \sigma^2 (\mathtt{J}^\top \mathtt{J})^{-1} = \sigma^2 \begin{pmatrix} \mathtt{A} & \mathtt{B} \\ \mathtt{B}^\top & \mathtt{C} \end{pmatrix}^{-1}$$

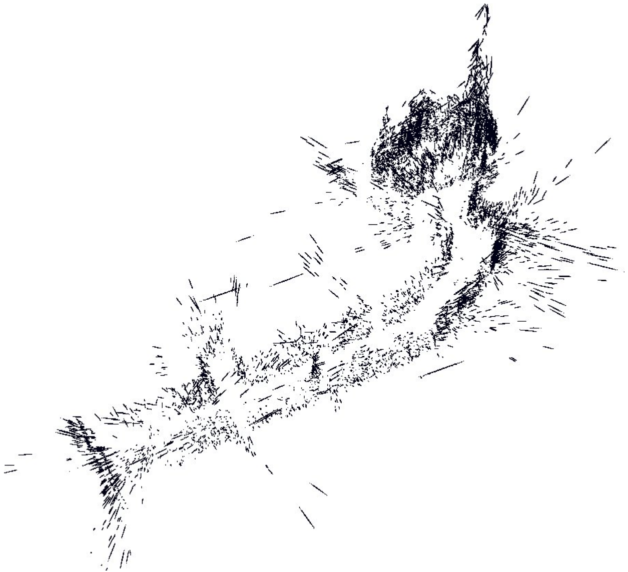
- $\mathcal{P}^*$ has precision (inverse covariance) matrix $\mathtt{A} - \mathtt{B}\mathtt{C}^{-1}\mathtt{B}^\top$
- $\mathcal{X}^*$ has precision (inverse covariance) matrix $\mathtt{C} - \mathtt{B}^\top\mathtt{A}^{-1}\mathtt{B}$
- $\mathcal{P}^*$ for fixed $\mathcal{X}^*$ has covariance matrix $\mathtt{A}^{-1}$ (block-diagonal)
  - Cameras are conditionally independent (when fixing $\mathcal{X}^*$)
- $\mathcal{X}^*$ for fixed $\mathcal{P}^*$ has covariance matrix $\mathtt{C}^{-1}$ (block-diagonal)
  - Points are conditionally independent (when fixing $\mathcal{P}^*$)

Bundle Adjustment and Uncertainty



Point cloud

Bundle Adjustment and Uncertainty



3D ellipsoids of point uncertainties

Bundle Adjustment and Uncertainty

Gauge freedom

In the context of bundle adjustment, gauge freedom is the projective ambiguity (projective BA) or ambiguity up to a similarity transformation (metric BA).

- $\mathtt{J}^\top \mathtt{J}$ is not full rank
  - 16/7 zero singular values
- $\{\mathtt{P}_i\}$ and $\{\mathbf{X}_j\}$ are completely uncertain in several directions
  - Restrict uncertainty analysis to row space of $\mathtt{J}$
- Fixing the gauge
  - Using additional data, e.g. GPS coordinates (with a noise model)
  - Fixing $\mathtt{P}_i = (\mathtt{I} \mid \mathbf{0})$ for some i etc.
    - Not a good idea: uncertainties will depend on i
    - Often hurts performance of BA
  - Usually not needed (BA with free gauge)

Modeling the Camera Parameters

- Projective bundle adjustment
  - No constraint on $\mathtt{P}_i \in \mathbb{R}^{3\times 4}$

$$\sum\nolimits_{i,j} m_{ij} \left\| \mathbf{x}_{ij} - \pi(\mathtt{P}_i \mathbf{X}_j) \right\|^2$$

  - Projective ambiguity, requires auto-calibration
- Metric bundle adjustment
  - $\mathtt{P}_i$ is structured

$$\sum\nolimits_{i,j} m_{ij} \left\| \mathbf{x}_{ij} - \mathtt{K}_i \pi(\mathtt{R}_i \mathbf{X}_j + \mathbf{T}_i) \right\|^2 \quad \text{or} \quad \sum\nolimits_{i,j} m_{ij} \left\| \tilde{\mathbf{x}}_{ij} - \pi(\mathtt{R}_i \mathbf{X}_j + \mathbf{T}_i) \right\|^2$$

  - Calibration matrix $\mathtt{K}_i$ known and fixed
  - Nonlinear constraints on $\mathtt{R}_i \in SO(3)$
  - Metric 3D model
- "Extended" metric bundle adjustment
  - Also optimze $\mathtt{K}_i = \begin{pmatrix} f_i & 0 & x_i \\ 0 & f_i & y_i \\ 0 & 0 & 1 \end{pmatrix}$
  - Maybe also include radial distortion coefficients
  - Metric 3D model

Modeling the Camera Parameters

- Projective bundle adjustment
  - No constraint on $\mathtt{P}_i \in \mathbb{R}^{3\times 4}$

$$\sum_{i,j} m_{ij} \left\| \mathbf{x}_{ij} - \pi(\mathtt{P}_i \mathbf{X}_j) \right\|^2$$

  - Projective ambiguity, requires auto-calibration
- Metric bundle adjustment
  - $\mathtt{P}_i$ is structured

$$\sum_{i,j} m_{ij} \left\| \mathbf{x}_{ij} - \mathtt{K}_i \pi(\mathtt{R}_i \mathbf{X}_j + \mathbf{T}_i) \right\|^2 \quad \text{or} \quad \sum_{i,j} m_{ij} \left\| \tilde{\mathbf{x}}_{ij} - \pi(\mathtt{R}_i \mathbf{X}_j + \mathbf{T}_i) \right\|^2$$

  - Calibration matrix $\mathtt{K}_i$ known and fixed
  - Nonlinear constraints on $\mathtt{R}_i \in SO(3)$
  - Metric 3D model
- "Extended" metric bundle adjustment
  - Also optimze $\mathtt{K}_i = \begin{pmatrix} f_i & 0 & x_i \\ 0 & f_i & y_i \\ 0 & 0 & 1 \end{pmatrix}$
  - Maybe also include radial distortion coefficients
  - Metric 3D model

Modeling the Camera Parameters

- Projective bundle adjustment
  - No constraint on $\mathtt{P}_i \in \mathbb{R}^{3\times 4}$

$$\sum\nolimits_{i,j} m_{ij} \left\| \mathbf{x}_{ij} - \pi(\mathtt{P}_i \mathbf{X}_j) \right\|^2$$

  - Projective ambiguity, requires auto-calibration
- Metric bundle adjustment
  - $\mathtt{P}_i$ is structured

$$\sum\nolimits_{i,j} m_{ij} \left\| \mathbf{x}_{ij} - \mathtt{K}_i \pi(\mathtt{R}_i \mathbf{X}_j + \mathbf{T}_i) \right\|^2 \quad \text{or} \quad \sum\nolimits_{i,j} m_{ij} \left\| \tilde{\mathbf{x}}_{ij} - \pi(\mathtt{R}_i \mathbf{X}_j + \mathbf{T}_i) \right\|^2$$

  - Calibration matrix $\mathtt{K}_i$ known and fixed
  - Nonlinear constraints on $\mathtt{R}_i \in SO(3)$
  - Metric 3D model
- "Extended" metric bundle adjustment
  - Also optimze $\mathtt{K}_i = \begin{pmatrix} f_i & 0 & x_i \\ 0 & f_i & y_i \\ 0 & 0 & 1 \end{pmatrix}$
  - Maybe also include radial distortion coefficients
  - Metric 3D model

Metric Bundle Adjustment

$\mathrm{P}_i = (\mathrm{R}_i \mid \mathbf{T}_i)$

Main issue: how to parametrize $\mathtt{R}_i \in SO(3)$?

- $\mathtt{R}_i \in \mathbb{R}^{3\times 3}$ has 9 d.o.f.
- $\mathtt{R}_i \in SO(3)$ has 3 d.o.f.
- $SO(3)$ is not a vector space: $\mathtt{R}_i + \delta\mathtt{R}_i \notin SO(3)$ in general
- Use unit quaternion representation: $\mathbf{q} = (q_0, q_1, q_2, q_3)^\top \in \mathbb{S}^3$

$$\mathrm{R}(\mathbf{q}) = \begin{pmatrix} 1-2(q_2^2+q_3)^2 & 2(q_1q_2 - q_0q_3) & 2(q_1q_3 + q_0q_2) \\ 2(q_1q_2 + q_0q_3) & 1-2(q_1^2+q_3)^2 & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_2q_3 + q_0q_1) & 1-2(q_1^2+q_2)^2 \end{pmatrix}$$

- How to enforce $\|\mathbf{q}\| = 1$
  - Model $\mathbf{q} = \mathbf{u}/\|\mathbf{u}\|$
  - Project gradient w.r.t. $\mathbf{q}$ into tangent plane of $\mathcal{S}^3$ at $\mathbf{q}$

$$T_{\mathbf{q}}\mathbb{S}^3 = \{\mathbf{v} \in \mathbb{R}^4 : \mathbf{q}^\top \mathbf{v} = 1\}$$

Normalize $\mathbf{q}$ after update

Metric Bundle Adjustment

$\mathtt{P}_i = (\mathtt{R}_i \mid \mathbf{T}_i)$

Main issue: how to parametrize $\mathtt{R}_i \in SO(3)$?

- $\mathtt{R}_i \in \mathbb{R}^{3\times 3}$ has 9 d.o.f.
- $\mathtt{R}_i \in SO(3)$ has 3 d.o.f.
- $SO(3)$ is not a vector space: $\mathtt{R}_i + \delta\mathtt{R}_i \notin SO(3)$ in general
- Use unit quaternion representation: $\mathbf{q} = (q_0, q_1, q_2, q_3)^\top \in \mathbb{S}^3$

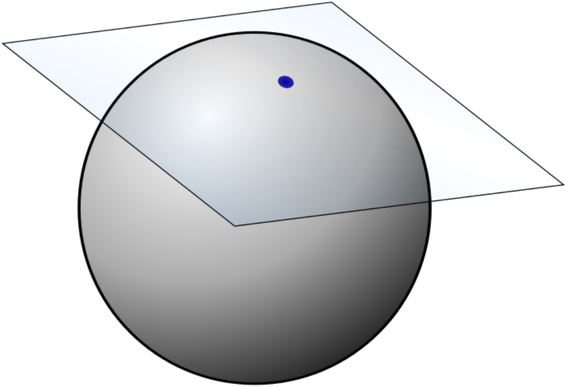
$$\mathtt{R}(\mathbf{q}) = \begin{pmatrix} 1-2(q_2^2+q_3)^2 & 2(q_1q_2 - q_0q_3) & 2(q_1q_3 + q_0q_2) \\ 2(q_1q_2 + q_0q_3) & 1-2(q_1^2+q_3)^2 & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_2q_3 + q_0q_1) & 1-2(q_1^2+q_2)^2 \end{pmatrix}$$

- How to enforce $\|\mathbf{q}\| = 1$
  - Model $\mathbf{q} = \mathbf{u}/\|\mathbf{u}\|$
  - Project gradient w.r.t. $\mathbf{q}$ into tangent plane of $\mathcal{S}^3$ at $\mathbf{q}$

$$T_{\mathbf{q}}\mathbb{S}^3 = \{\mathbf{v} \in \mathbb{R}^4 : \mathbf{q}^\top\mathbf{v} = 1\}$$

  Normalize $\mathbf{q}$ after update

Metric Bundle Adjustment



$$T_{\mathbf{q}}\mathbb{S}^3 = \{\mathbf{v} \in \mathbb{R}^4 : \mathbf{q}^\top \mathbf{v} = 1\}$$

Metric Bundle Adjustment

$\mathtt{P}_i = (\mathtt{R}_i \mid \mathbf{T}_i)$

Main issue: how to parametrize $\mathtt{R}_i \in SO(3)$?

- Axis-angle representation $\mathbf{w} = \theta\mathbf{a} \in \mathbb{R}^3$, $\|\mathbf{a}\| = 1$, $\theta = \|\mathbf{w}\|$
- Use exponential map (matrix exponential)

$$\begin{aligned}
\mathtt{R}(\mathbf{w}) &= \mathrm{expm}([\mathbf{w}]_\times) \\
&= \mathtt{I} + \sin(\theta)[\mathbf{a}]_\times + (1-\cos(\theta))[\mathbf{a}]_\times^2 \\
&= \mathtt{I} + \frac{\sin(\|\mathbf{w}\|)}{\|\mathbf{w}\|}[\mathbf{w}]_\times + \frac{(1-\cos(\|\mathbf{w}\|))}{\|\mathbf{w}\|^2}[\mathbf{w}]_\times^2
\end{aligned}$$

- $\mathbb{R}^3$ is a vector space, so we could use $\mathbf{w}$ directly
- Mapping is very non-linear when $\|\mathbf{w}\| \gg 0$
  - Better behavior when $\|\mathbf{w}\| \approx \mathbf{0}$. How can we handle $\|\mathbf{w}\| \to 0$?
- Avoiding $\|\mathbf{w}\| \gg 0$:
  - Option 1: use $\mathtt{R}(\mathbf{w})\,\mathtt{R}^{\text{init}}$ instead of just $\mathtt{R}(\mathbf{w})$
  - Option 2: use incremental rotations $\mathtt{R}^{(t+1)} = \mathtt{R}(\mathbf{w})\,\mathtt{R}^{(t)}$ in every iteration

Metric Bundle Adjustment

$\mathtt{P}_i = (\mathtt{R}_i \mid \mathbf{T}_i)$

Main issue: how to parametrize $\mathtt{R}_i \in SO(3)$?

- Axis-angle representation $\mathbf{w} = \theta\mathbf{a} \in \mathbb{R}^3$, $\|\mathbf{a}\| = 1$, $\theta = \|\mathbf{w}\|$
- Use exponential map (matrix exponential)

$$\begin{aligned}\mathtt{R}(\mathbf{w}) &= \operatorname{expm}([\mathbf{w}]_\times) \\ &= \mathtt{I} + \sin(\theta)[\mathbf{a}]_\times + (1-\cos(\theta))[\mathbf{a}]_\times^2 \\ &= \mathtt{I} + \frac{\sin(\|\mathbf{w}\|)}{\|\mathbf{w}\|}[\mathbf{w}]_\times + \frac{(1-\cos(\|\mathbf{w}\|))}{\|\mathbf{w}\|^2}[\mathbf{w}]_\times^2\end{aligned}$$

- $\mathbb{R}^3$ is a vector space, so we could use $\mathbf{w}$ directly
- Mapping is very non-linear when $\|\mathbf{w}\| \gg 0$
  - Better behavior when $\|\mathbf{w}\| \approx \mathbf{0}$. How can we handle $\|\mathbf{w}\| \to 0$?
- Avoiding $\|\mathbf{w}\| \gg 0$:
  - Option 1: use $\mathtt{R}(\mathbf{w})\,\mathtt{R}^{\text{init}}$ instead of just $\mathtt{R}(\mathbf{w})$
  - Option 2: use incremental rotations $\mathtt{R}^{(t+1)} = \mathtt{R}(\mathbf{w})\,\mathtt{R}^{(t)}$ in every iteration

Metric Bundle Adjustment

$\mathtt{P}_i = (\mathtt{R}_i \mid \mathbf{T}_i)$

Main issue: how to parametrize $\mathtt{R}_i \in SO(3)$?

- Axis-angle representation $\mathbf{w} = \theta \mathbf{a} \in \mathbb{R}^3$, $\|\mathbf{a}\| = 1$, $\theta = \|\mathbf{w}\|$
- Use exponential map (matrix exponential)

$$\begin{aligned}
\mathtt{R}(\mathbf{w}) &= \operatorname{expm}([\mathbf{w}]_\times) \\
&= \mathtt{I} + \sin(\theta)[\mathbf{a}]_\times + (1 - \cos(\theta))[\mathbf{a}]_\times^2 \\
&= \mathtt{I} + \frac{\sin(\|\mathbf{w}\|)}{\|\mathbf{w}\|}[\mathbf{w}]_\times + \frac{(1 - \cos(\|\mathbf{w}\|))}{\|\mathbf{w}\|^2}[\mathbf{w}]_\times^2
\end{aligned}$$

- $\mathbb{R}^3$ is a vector space, so we could use $\mathbf{w}$ directly
- Mapping is very non-linear when $\|\mathbf{w}\| \gg 0$
  - Better behavior when $\|\mathbf{w}\| \approx \mathbf{0}$. How can we handle $\|\mathbf{w}\| \to 0$?
- Avoiding $\|\mathbf{w}\| \gg 0$:
  - Option 1: use $\mathtt{R}(\mathbf{w})\,\mathtt{R}^{\text{init}}$ instead of just $\mathtt{R}(\mathbf{w})$
  - Option 2: use incremental rotations $\mathtt{R}^{(t+1)} = \mathtt{R}(\mathbf{w})\,\mathtt{R}^{(t)}$ in every iteration

Metric Bundle Adjustment

$\mathtt{P}_i = (\mathtt{R}_i \mid \mathbf{T}_i)$

Main issue: how to parametrize $\mathtt{R}_i \in SO(3)$?

- Axis-angle representation $\mathbf{w} = \theta\mathbf{a} \in \mathbb{R}^3$, $\|\mathbf{a}\| = 1$, $\theta = \|\mathbf{w}\|$
- Use exponential map (matrix exponential)

$$\begin{aligned}\mathtt{R}(\mathbf{w}) &= \mathrm{expm}([\mathbf{w}]_\times) \\ &= \mathtt{I} + \sin(\theta)[\mathbf{a}]_\times + (1 - \cos(\theta))[\mathbf{a}]_\times^2 \\ &= \mathtt{I} + \frac{\sin(\|\mathbf{w}\|)}{\|\mathbf{w}\|}[\mathbf{w}]_\times + \frac{(1 - \cos(\|\mathbf{w}\|))}{\|\mathbf{w}\|^2}[\mathbf{w}]_\times^2\end{aligned}$$

- $\mathbb{R}^3$ is a vector space, so we could use $\mathbf{w}$ directly
- Mapping is very non-linear when $\|\mathbf{w}\| \gg 0$
  - Better behavior when $\|\mathbf{w}\| \approx \mathbf{0}$. How can we handle $\|\mathbf{w}\| \to 0$?
- Avoiding $\|\mathbf{w}\| \gg 0$:
  - Option 1: use $\mathtt{R}(\mathbf{w})\,\mathtt{R}^{\text{init}}$ instead of just $\mathtt{R}(\mathbf{w})$
  - Option 2: use incremental rotations $\mathtt{R}^{(t+1)} = \mathtt{R}(\mathbf{w})\,\mathtt{R}^{(t)}$ in every iteration

Metric Bundle Adjustment

$\mathtt{P}_i = (\mathtt{R}_i \mid \mathbf{T}_i)$

Main issue: how to parametrize $\mathtt{R}_i \in SO(3)$?

- Axis-angle representation $\mathbf{w} = \theta\mathbf{a} \in \mathbb{R}^3$, $\|\mathbf{a}\| = 1$, $\theta = \|\mathbf{w}\|$
- Use exponential map (matrix exponential)

$$\begin{aligned} \mathtt{R}(\mathbf{w}) &= \operatorname{expm}([\mathbf{w}]_\times) \\ &= \mathtt{I} + \sin(\theta)[\mathbf{a}]_\times + (1-\cos(\theta))[\mathbf{a}]_\times^2 \\ &= \mathtt{I} + \frac{\sin(\|\mathbf{w}\|)}{\|\mathbf{w}\|}[\mathbf{w}]_\times + \frac{(1-\cos(\|\mathbf{w}\|))}{\|\mathbf{w}\|^2}[\mathbf{w}]_\times^2 \end{aligned}$$

- $\mathbb{R}^3$ is a vector space, so we could use $\mathbf{w}$ directly
- Mapping is very non-linear when $\|\mathbf{w}\| \gg 0$
  - Better behavior when $\|\mathbf{w}\| \approx \mathbf{0}$. How can we handle $\|\mathbf{w}\| \to 0$?
- Avoiding $\|\mathbf{w}\| \gg 0$:
  - Option 1: use $\mathtt{R}(\mathbf{w})\,\mathtt{R}^{\text{init}}$ instead of just $\mathtt{R}(\mathbf{w})$
  - Option 2: use incremental rotations $\mathtt{R}^{(t+1)} = \mathtt{R}(\mathbf{w})\,\mathtt{R}^{(t)}$ in every iteration

Metric Bundle Adjustment

- Linearize around current value $\mathtt{R}_i^{(t)}$

$$\mathtt{R}_i^{(t+1)} = \mathtt{R}(\mathbf{w}_i)\mathtt{R}_i^{(t)} \qquad \mathbf{w}_i \text{ is the unknown}$$

- Now $\mathbf{w}_i \approx \mathbf{0}$

$$\mathrm{R}(\mathbf{w}) \approx \mathrm{I} + [\mathbf{w}]_\times$$

- Recall

$$[\mathbf{w}]_\times = \begin{pmatrix} 0 & -w_3 & w_2 \\ w_3 & 0 & -w_1 \\ -w_2 & w_1 & 0 \end{pmatrix} = w_1\mathrm{S}_1 + w_2\mathrm{S}_2 + w_3\mathrm{S}_3$$

with

$$\mathrm{S}_1 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix} \qquad \mathrm{S}_2 = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{pmatrix} \qquad \mathrm{S}_3 = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

- Derivative

$$\frac{\partial \mathrm{R}(\mathbf{w})}{\partial w_k}\Big|_{\mathbf{w}=\mathbf{0}} = \mathrm{S}_k \qquad \frac{\partial\left(\mathrm{R}(\mathbf{w})\mathrm{R}^{(t)}\right)}{\partial w_k}\Big|_{\mathbf{w}=\mathbf{0}} = \mathrm{S}_k\mathrm{R}^{(t)}$$

Metric Bundle Adjustment

- Linearize around current value $\mathtt{R}_i^{(t)}$

$$\mathtt{R}_i^{(t+1)} = \mathtt{R}(\mathbf{w}_i)\mathtt{R}_i^{(t)} \qquad\qquad \mathbf{w}_i \text{ is the unknown}$$

- Now $\mathbf{w}_i \approx \mathbf{0}$

$$\mathtt{R}(\mathbf{w}) \approx \mathtt{I} + [\mathbf{w}]_\times$$

- Recall

$$[\mathbf{w}]_\times = \begin{pmatrix} 0 & -w_3 & w_2 \\ w_3 & 0 & -w_1 \\ -w_2 & w_1 & 0 \end{pmatrix} = w_1 \mathrm{S}_1 + w_2 \mathrm{S}_2 + w_3 \mathrm{S}_3$$

with

$$\mathrm{S}_1 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix} \qquad \mathrm{S}_2 = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{pmatrix} \qquad \mathrm{S}_3 = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

- Derivative

$$\frac{\partial \mathrm{R}(\mathbf{w})}{\partial w_k}\Big|_{\mathbf{w}=\mathbf{0}} = \mathrm{S}_k \qquad\qquad \frac{\partial\left(\mathrm{R}(\mathbf{w})\mathrm{R}^{(t)}\right)}{\partial w_k}\Big|_{\mathbf{w}=\mathbf{0}} = \mathrm{S}_k \mathrm{R}^{(t)}$$

Metric Bundle Adjustment

- Linearize around current value $\mathtt{R}_i^{(t)}$

$$\mathtt{R}_i^{(t+1)} = \mathtt{R}(\mathbf{w}_i)\mathtt{R}_i^{(t)} \qquad\qquad \mathbf{w}_i \text{ is the unknown}$$

- Now $\mathbf{w}_i \approx \mathbf{0}$

$$\mathtt{R}(\mathbf{w}) \approx \mathtt{I} + [\mathbf{w}]_\times$$

- Recall

$$[\mathbf{w}]_\times = \begin{pmatrix} 0 & -w_3 & w_2 \\ w_3 & 0 & -w_1 \\ -w_2 & w_1 & 0 \end{pmatrix} = w_1\mathtt{S}_1 + w_2\mathtt{S}_2 + w_3\mathtt{S}_3$$

with

$$\mathtt{S}_1 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix} \qquad \mathtt{S}_2 = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{pmatrix} \qquad \mathtt{S}_3 = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

- Derivative

$$\frac{\partial \mathtt{R}(\mathbf{w})}{\partial w_k}\Big|_{\mathbf{w}=\mathbf{0}} = \mathtt{S}_k \qquad\qquad \frac{\partial\big(\mathtt{R}(\mathbf{w})\mathtt{R}^{(t)}\big)}{\partial w_k}\Big|_{\mathbf{w}=\mathbf{0}} = \mathtt{S}_k\mathtt{R}^{(t)}$$

Metric Bundle Adjustment

- Linearize around current value $\mathtt{R}_i^{(t)}$

$$\mathtt{R}_i^{(t+1)} = \mathtt{R}(\mathbf{w}_i)\mathtt{R}_i^{(t)} \qquad\qquad \mathbf{w}_i \text{ is the unknown}$$

- Now $\mathbf{w}_i \approx \mathbf{0}$

$$\mathtt{R}(\mathbf{w}) \approx \mathtt{I} + [\mathbf{w}]_\times$$

- Recall

$$[\mathbf{w}]_\times = \begin{pmatrix} 0 & -w_3 & w_2 \\ w_3 & 0 & -w_1 \\ -w_2 & w_1 & 0 \end{pmatrix} = w_1 \mathtt{S}_1 + w_2 \mathtt{S}_2 + w_3 \mathtt{S}_3$$

with

$$\mathtt{S}_1 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix} \qquad \mathtt{S}_2 = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{pmatrix} \qquad \mathtt{S}_3 = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

- Derivative

$$\frac{\partial \mathtt{R}(\mathbf{w})}{\partial w_k}\Big|_{\mathbf{w}=\mathbf{0}} = \mathtt{S}_k \qquad\qquad \frac{\partial\big(\mathtt{R}(\mathbf{w})\mathtt{R}^{(t)}\big)}{\partial w_k}\Big|_{\mathbf{w}=\mathbf{0}} = \mathtt{S}_k \mathtt{R}^{(t)}$$

Metric Bundle Adjustment

- Combining everything

$$\frac{\partial(\mathtt{R}_i^{(t)}\mathbf{X}_j^{(t)} + \mathbf{T}_i^{(t)})}{\partial \mathbf{X}_j} = \mathtt{R}_i^{(t)}$$

$$\frac{\partial(\mathtt{R}_i^{(t)}\mathbf{X}_j^{(t)} + \mathbf{T}_i^{(t)})}{\partial \mathbf{T}_i} = \mathtt{I} \qquad \frac{\partial(\mathtt{R}_i^{(t)}\mathbf{X}_j^{(t)} + \mathbf{T}_i^{(t)})}{\partial w_{j,k}} = \mathtt{S}_k\mathtt{R}_i^{(t)} \qquad k \in \{1,2,3\}$$

  Use with the chain rule for handle $\pi(\mathtt{R}_i\mathbf{X}_j + \mathbf{T}_i)$

- Solve augmented normal equations to obtain updates $\delta\mathbf{X}_j$, $\delta\mathbf{T}_i$ and $\mathbf{w}_i$

$$\mathbf{X}_j^{(t+1)} \leftarrow \mathbf{X}_j^{(t)} + \delta\mathbf{X}_j$$

$$\mathbf{T}_i^{(t+1)} \leftarrow \mathbf{T}_i^{(t)} + \delta\mathbf{T}_i \qquad \mathtt{R}_i^{(t+1)} \leftarrow \mathrm{expm}([\mathbf{w}_i]_\times)\mathtt{R}_i^{(t)}$$

- It might be a good idea to project the new matrices $\mathtt{R}_i^{(t+1)}$ to the closest rotation matrix

$$\mathtt{U\Sigma V}^\top = \mathrm{SVD}(\mathtt{R}_i^{(t+1)}) \qquad \mathtt{R}_i^{(t+1)} \leftarrow \mathtt{UV}^\top$$

Metric Bundle Adjustment

- Combining everything

$$\frac{\partial(\mathtt{R}_i^{(t)}\mathbf{X}_j^{(t)} + \mathbf{T}_i^{(t)})}{\partial \mathbf{X}_j} = \mathtt{R}_i^{(t)}$$

$$\frac{\partial(\mathtt{R}_i^{(t)}\mathbf{X}_j^{(t)} + \mathbf{T}_i^{(t)})}{\partial \mathbf{T}_i} = \mathtt{I} \qquad \frac{\partial(\mathtt{R}_i^{(t)}\mathbf{X}_j^{(t)} + \mathbf{T}_i^{(t)})}{\partial w_{j,k}} = \mathtt{S}_k\mathtt{R}_i^{(t)} \qquad k \in \{1,2,3\}$$

Use with the chain rule for handle $\pi(\mathtt{R}_i\mathbf{X}_j + \mathbf{T}_i)$

- Solve augmented normal equations to obtain updates $\delta\mathbf{X}_j$, $\delta\mathbf{T}_i$ and $\mathbf{w}_i$

$$\mathbf{X}_j^{(t+1)} \leftarrow \mathbf{X}_j^{(t)} + \delta\mathbf{X}_j$$

$$\mathbf{T}_i^{(t+1)} \leftarrow \mathbf{T}_i^{(t)} + \delta\mathbf{T}_i \qquad \mathtt{R}_i^{(t+1)} \leftarrow \operatorname{expm}([\mathbf{w}_i]_\times)\mathtt{R}_i^{(t)}$$

- It might be a good idea to project the new matrices $\mathtt{R}_i^{(t+1)}$ to the closest rotation matrix

$$\mathtt{U}\Sigma\mathtt{V}^\top = \operatorname{SVD}(\mathtt{R}_i^{(t+1)}) \qquad \mathtt{R}_i^{(t+1)} \leftarrow \mathtt{U}\mathtt{V}^\top$$

Metric Bundle Adjustment

- Combining everything

$$\frac{\partial(\mathtt{R}_i^{(t)}\mathbf{X}_j^{(t)} + \mathbf{T}_i^{(t)})}{\partial \mathbf{X}_j} = \mathtt{R}_i^{(t)}$$

$$\frac{\partial(\mathtt{R}_i^{(t)}\mathbf{X}_j^{(t)} + \mathbf{T}_i^{(t)})}{\partial \mathbf{T}_i} = \mathtt{I} \qquad \frac{\partial(\mathtt{R}_i^{(t)}\mathbf{X}_j^{(t)} + \mathbf{T}_i^{(t)})}{\partial w_{j,k}} = \mathtt{S}_k \mathtt{R}_i^{(t)} \qquad k \in \{1,2,3\}$$

  Use with the chain rule for handle $\pi(\mathtt{R}_i\mathbf{X}_j + \mathbf{T}_i)$
- Solve augmented normal equations to obtain updates $\delta\mathbf{X}_j$, $\delta\mathbf{T}_i$ and $\mathbf{w}_i$

$$\mathbf{X}_j^{(t+1)} \leftarrow \mathbf{X}_j^{(t)} + \delta\mathbf{X}_j$$

$$\mathbf{T}_i^{(t+1)} \leftarrow \mathbf{T}_i^{(t)} + \delta\mathbf{T}_i \qquad \mathtt{R}_i^{(t+1)} \leftarrow \mathrm{expm}([\mathbf{w}_i]_\times)\mathtt{R}_i^{(t)}$$

- It might be a good idea to project the new matrices $\mathtt{R}_i^{(t+1)}$ to the closest rotation matrix

$$\mathtt{U\Sigma V}^\top = \mathrm{SVD}(\mathtt{R}_i^{(t+1)}) \qquad \mathtt{R}_i^{(t+1)} \leftarrow \mathtt{UV}^\top$$

Metric Bundle Adjustment

- In general

$$\operatorname{expm}([\mathbf{w} + \delta\mathbf{w}]_\times) \neq \operatorname{expm}([\delta\mathbf{w}]_\times)\operatorname{expm}([\mathbf{w}]_\times)$$

  and

$$\operatorname{expm}\left([\delta\mathbf{w}]_\times + \operatorname{logm}(\mathtt{R}^{(t)})\right) \neq \operatorname{expm}([\mathbf{w}]_\times)\mathtt{R}^{(t)}$$

- We modeled

$$(\mathtt{R} \mid \mathbf{T}) \in SO(3) \times \mathbb{R}^3$$

- Often in robotics:

$$(\mathtt{R} \mid \mathbf{T}) \in SE(3) \qquad \text{special Euclidean group}$$

  Tightly links $\mathtt{R}$ and $\mathbf{T}$ in expm etc.

Upcoming

- Next time: factorization methods

Lab sessions today: E-D2480, ES61, ES62 & ES63