Computer Vision: Lecture 9

2023-11-27

Today's Lecture

Reconstruction and Optimization

- Reprojection error: noise models
- Maximum likelihood estimation
- Principles of Local Optimization

- Szeliski, Sec A.2 and A.3 (least squares)
- Szeliski, Sec B.1 and B.2 (MLE)

Discussion

Question

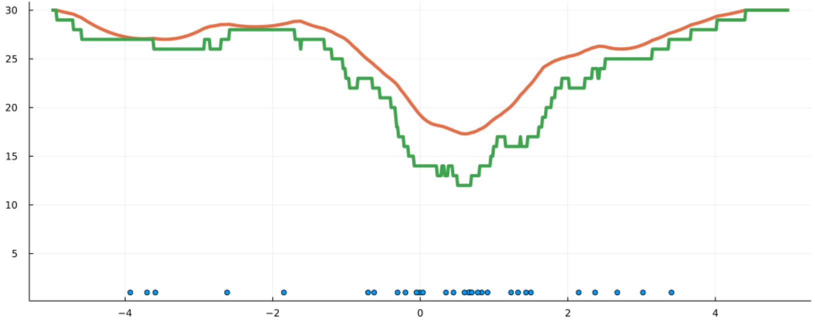
Is the solution returned by RANSAC the best solution we can ask for?

Discussion

Question

Is the solution returned by RANSAC the best solution we can ask for?

- No: we want a revised model taking into account all inliers
- Inliers are still affected by (well-behaved) noise
- RANSAC can be seen as smart way to find an initial solution for a very difficult optimization problem



Model Fitting

Model fitting task

Given a set of model parameters find the parameter values that give the "best" fit to the data.

Examples:

Camera Estimation Given scene points $\mathbf{X}_i$ find $\mathtt{P}$ such that $\mathtt{P}\mathbf{X}_i$ gives the best fit to the detected image points $\mathbf{x}_i$.

Line Fitting Find the line that best fits a set of 2D-points (x_i, y_i).

What is the "best" fit? Depends on the *noise model*.

Least Squares Line Fitting

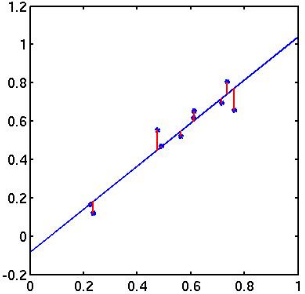
$$\min_{a,b} \sum_i (ax_i + b - y_i)^2$$

In matrix form

$$\min \left\| \underbrace{\begin{pmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \vdots \\ x_n & 1 \end{pmatrix}}_{\mathtt{A}} \begin{pmatrix} a \\ b \end{pmatrix} - \underbrace{\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}}_{\mathtt{B}} \right\|^2$$

Noise only in y, no noise in x

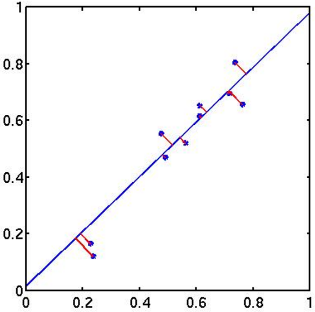
In Matlab use `A\B`



Total Least Squares Line Fitting

$$\min_{a,b,c} \sum_i (ax_i + by_i + c)^2$$

$$\text{s.t. } a^2 + b^2 = 1$$



Let

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \text{ and } \bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$$

The the optimum fulfills

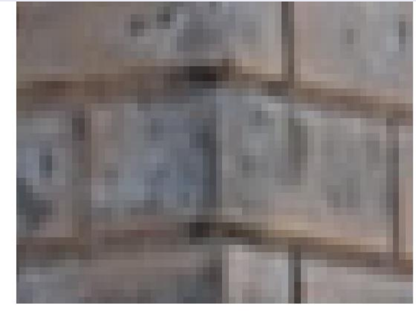
$$\sum_{i=1}^{m} \begin{bmatrix} (x_i - \bar{x})(x_i - \bar{x}) & (y_i - \bar{y})(x_i - \bar{x}) \\ (x_i - \bar{x})(y_i - \bar{y}) & (y_i - \bar{y})(y_i - \bar{y}) \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \lambda \begin{bmatrix} a \\ b \end{bmatrix}$$

and $c = -(a\bar{x} + b\bar{y})$
$(a,b)^\top$ is eigenvector corresponding to smallest eigenvalue

Handling Noise: Minimizing Reprojection Error

2D Position Noise

When outliers have been removed, measurements are still corrupted by noise. The exact position of a feature may be difficult to determine.



Handling Noise: Minimizing Reprojection Error

Under the assumption that image points are corrupted by Gaussian noise, minimize the reprojection error.

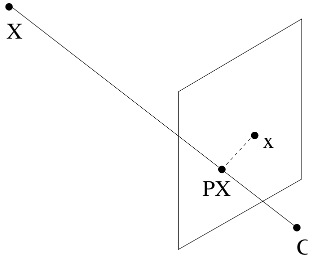
The reprojection error

In regular coordinates the projection is

$$\left(\frac{\mathtt{P}_{1,:}\mathbf{X}}{\mathtt{P}_{3,:}\mathbf{X}}, \frac{\mathtt{P}_{2,:}\mathbf{X}}{\mathtt{P}_{3,:}\mathbf{X}}\right)^\top = \pi(\mathtt{P}\mathbf{X})$$

$\mathtt{P}_{1,:}, \mathtt{P}_{2,:}, \mathtt{P}_{3,:}$ are the rows of $\mathtt{P}$.
The reprojection error is

$$\left\| \left(x_1 - \frac{\mathtt{P}_{1,:}\mathbf{X}}{\mathtt{P}_{3,:}\mathbf{X}}, x_2 - \frac{\mathtt{P}_{1,:}\mathbf{X}}{\mathtt{P}_{3,:}\mathbf{X}} \right) \right\|^2$$



Minimizing Reprojection Error

(Uncalibrated) Structure and Motion

Given image projections $\{\mathbf{x}_{ij} = (x_{ij}, y_{ij})\}$ (i = image nr, j = point nr), find 3D points $\mathbf{X}_j$ and camera matrices $\mathtt{P}_i$ such that

$$\sum_{ij} \|\mathbf{x}_{ij} - \pi(\mathtt{P}_i \mathbf{X}_j)\|^2$$

is minimized.

Calibrated Structure and Motion

Given (normalized) image projections $\{\mathbf{x}_{ij} = (x_{ij}, y_{ij})\}$ (i = image nr, j = point nr), find 3D points $\mathbf{X}_j$ and cameras $\mathtt{P}_i = (\mathtt{R}_i \mid \mathbf{T}_i)$ such that

$$\sum_{ij} \|\mathbf{x}_{ij} - \pi(\mathtt{P}_i \mathbf{X}_j)\|^2 = \sum_{ij} \left\| \begin{pmatrix} x_{ij} \\ y_{ij} \end{pmatrix} - \frac{(\mathtt{R}_i \mathbf{X}_j + \mathbf{T}_i)_{1,2}}{(\mathtt{R}_i \mathbf{X}_j + \mathbf{T}_i)_3} \right\|^2$$

is minimized.

- Complicated non linear expression.
- No closed form solution.

Reprojection Error vs. Algebraic Error

Algebraic Error

Attempts to find an approximate solution to an algebraic equation, e.g. DLT,

$$\min \sum\nolimits_j \left\| \lambda_j \mathbf{x}_j - \mathtt{P}\mathbf{X}_j \right\|^2$$

8-point algorithm etc.

Reprojection Error

- Gives most probable solution (least squares)
- Geometrically meaningful
- Nonlinear equations, difficult to optimize
  Often requires starting solutions

Algebraic Error

- No clear geometrical meaning
- May produce poor solutions
- Easy to optimize, using e.g. SVD

Use algebraic solution as starting point

Maximum Likelihood Estimation

General framework to infer unknown parameters from noisy observations

Maximum likelihood estimator (MLE)

Let $p(\cdot|\theta)$ be a pdf/pmf of a RV with parameters θ. For given observation $\mathbf{x}$ the MLE is given by

$$\theta^*_{MLE} = \arg\max_\theta p(\mathbf{x}|\theta) = \arg\max_\theta \log p(\mathbf{x}|\theta).$$

Maximum a-posteriori estimator (MAP)

Let $p(\cdot|\theta)$ be a pdf/pmf of a RV with parameters θ and $p_0(\theta)$ represent prior knowledge about θ. For given observation $\mathbf{x}$ the MAP is given by

$$\theta^*_{MAP} = \arg\max_\theta p(\mathbf{x}|\theta) p_0(\theta) = \arg\max_\theta \log p(\mathbf{x}|\theta) + \log p_0(\theta).$$

Maximum Likelihood Estimation

- We always use log-likelihood or negative log-likelihood ("energy")

$$\theta^*_{MLE} = \arg\max_\theta \log p(\mathbf{x}|\theta) = \arg\min_\theta -\log p(\mathbf{x}|\theta).$$

- Reason 1: numerical stability
- Reason 2: multiple observations, noise is assumed independent

$$\begin{aligned}\theta^*_{MLE} &= \arg\max_\theta \log p(\mathbf{x}_1, \dots, \mathbf{x}_n|\theta) \\ &= \arg\max_\theta \log \prod_{i=1}^n p(\mathbf{x}_i|\theta) \\ &= \arg\max_\theta \sum_{i=1}^n \log p(\mathbf{x}_i|\theta)\end{aligned}$$

Maximum Likelihood Estimation

- We use a deterministic forward model for the imaging process, e.g.

$$\mathbf{x} = \pi(\mathrm{P}\mathbf{X})$$

- Measuring image points is subjected to *known* noise model

$$\mathbf{x} = \pi(\mathrm{P}\mathbf{X}) + \varepsilon \qquad\qquad \varepsilon \sim p_{\text{noise}}$$

- Therefore

$$\begin{aligned} p(\mathbf{x}) &= \left(\delta_{\pi(\mathrm{P}\mathbf{X})} * p_{\text{noise}}\right)(\mathbf{x}) \\ &= \int \delta_{\pi(\mathrm{P}\mathbf{X})}(\mathbf{x} - \varepsilon) p_{\text{noise}}(\varepsilon)\, d\varepsilon \\ &= p_{\text{noise}}(\mathbf{x} - \pi(\mathrm{P}\mathbf{X})) \end{aligned}$$

Maximum Likelihood Estimation

- We use a deterministic forward model for the imaging process, e.g.

$$\mathbf{x} = \pi(\mathrm{P}\mathbf{X})$$

- Measuring image points is subjected to *known* noise model

$$\mathbf{x} = \pi(\mathrm{P}\mathbf{X}) + \varepsilon \qquad \qquad \varepsilon \sim p_{\text{noise}}$$

- Therefore

$$\begin{aligned} p(\mathbf{x}) &= \left(\delta_{\pi(\mathrm{P}\mathbf{X})} * p_{\text{noise}}\right)(\mathbf{x}) \\ &= \int \delta_{\pi(\mathrm{P}\mathbf{X})}(\mathbf{x} - \varepsilon) p_{\text{noise}}(\varepsilon)\, d\varepsilon \\ &= p_{\text{noise}}(\mathbf{x} - \pi(\mathrm{P}\mathbf{X})) \end{aligned}$$

Maximum Likelihood Estimation

- We use a deterministic forward model for the imaging process, e.g.

$$\mathbf{x} = \pi(\mathtt{P}\mathbf{X})$$

- Measuring image points is subjected to *known* noise model

$$\mathbf{x} = \pi(\mathtt{P}\mathbf{X}) + \varepsilon \qquad\qquad \varepsilon \sim p_{\text{noise}}$$

- Therefore

$$\begin{aligned} p(\mathbf{x}) &= \left(\delta_{\pi(\mathtt{P}\mathbf{X})} * p_{\text{noise}}\right)(\mathbf{x}) \\ &= \int \delta_{\pi(\mathtt{P}\mathbf{X})}(\mathbf{x} - \varepsilon) p_{\text{noise}}(\varepsilon)\, d\varepsilon \\ &= p_{\text{noise}}\big(\mathbf{x} - \pi(\mathtt{P}\mathbf{X})\big) \end{aligned}$$

Noise Models for Image Points

Common noise assumptions (ε is 2-vector)

- Isotropic Gaussian: $p_{\text{noise}}(\varepsilon) = \mathcal{N}(\mathbf{0}, \sigma^2 \mathtt{I})$

$$-\log p_{\text{noise}}(\varepsilon) \doteq \tfrac{1}{2\sigma^2} \|\varepsilon\|^2$$

- Laplace distribution: $p_{\text{noise}}(\varepsilon) = \mathrm{Laplace}(0, b)$

$$-\log p_{\text{noise}}(\varepsilon) \doteq \tfrac{1}{b} \|\varepsilon\|$$

- Uniform distribution: $p_{\text{noise}}(\varepsilon) = \mathcal{U}_{\mathcal{B}(b)}$ ($\mathcal{B}(b)$ is disk with radius b)

$$-\log p_{\text{noise}}(\varepsilon) \doteq \imath_{\mathcal{B}(b)}(\varepsilon) = \begin{cases} 0 & \text{if } \|\varepsilon\| \le b \\ \infty & \text{otherwise} \end{cases}$$

- Zero-mean assumption
- Usually symmetric distribution

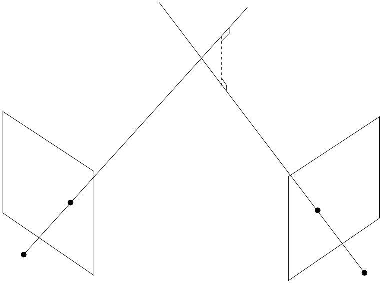
Optimal Triangulation

Triangulation

For corresponding image points $\mathbf{x}_i$ in N cameras $\mathtt{P}_i$ determine $\mathbf{X}$ such that

$$-\sum_i \log p_{\text{noise}}(\mathbf{x}_i - \pi(\mathtt{P}_i \mathbf{X}))$$

is minimal (i.e. $\mathbf{X}$ is maximal likely).



Optimal Triangulation

Triangulation

For corresponding image points $\mathbf{x}_i$ in N cameras $\mathtt{P}_i$ determine $\mathbf{X}$ such that

$$-\sum_i \log p_{\mathsf{noise}}(\mathbf{x}_i - \pi(\mathtt{P}_i\mathbf{X}))$$

is minimal (i.e. $\mathbf{X}$ is maximal likely).

- Gaussian noise (same in all images)

$$\min_{\mathbf{X}} \sum_i \|\mathbf{x}_i - \pi(\mathtt{P}_i\mathbf{X})\|^2$$

- Laplacian noise (same in all images)

$$\min_{\mathbf{X}} \sum_i \|\mathbf{x}_i - \pi(\mathtt{P}_i\mathbf{X})\|$$

- Uniform noise (same in all images)

$$\text{Find } \mathbf{x} \text{ such that } \forall i : \|\mathbf{x}_i - \pi(\mathtt{P}_i\mathbf{X})\| \le b$$

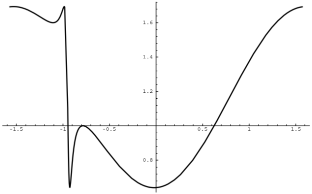
Optimal Triangulation: Gaussian Noise

Optimal L_2 triangulation

Find $\mathbf{X}$ that minimizes

$$\sum_{i=1}^{N} \left\| \mathbf{x}_i - \pi(\mathrm{P}_i \mathbf{X}) \right\|^2$$

- $N = 1$: under-constrained
- $N = 2$ (two-view triangulation)
  - Hartley-Sturm method
  - Solve polynomial of degree 6
  - Up to 3 local minima



- $N \geq 3$: iterative methods, local optimization

Optimal Triangulation: Laplacian Noise

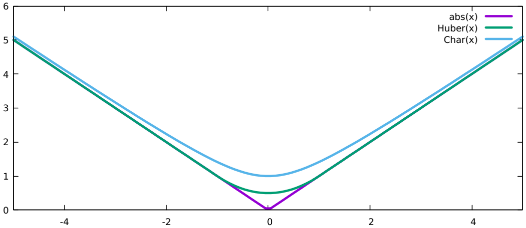
Optimal L_1 triangulation

Find $\mathbf{X}$ that minimizes $\sum_{i=1}^{N} \|\mathbf{x}_i - \pi(\mathtt{P}_i \mathbf{X})\|$.

- Non-convex & non-differentiable objective
- Differentiable surrogate: local optimization
  - Huber cost

$$H_\rho(\mathbf{r}) = \begin{cases} \|\mathbf{r}\|^2/(2\rho) + \rho/2 & \text{if } \|\mathbf{r}\| \le \rho \\ \|\mathbf{r}\| & \text{if } \|\mathbf{r}\| \ge \rho \end{cases}$$

  - Charbonnier (or L_1-L_2) cost: $\sqrt{\|\mathbf{r}\|^2 + \rho^2}$



Optimal Triangulation: Uniform Noise

Optimal L_∞ triangulation

For $b > 0$ find $\mathbf{X}$ that satisfies

$$\|\mathbf{x}_i - \pi(\mathtt{P}_i\mathbf{X})\| \leq b.$$

- $\left\{\mathbf{X} : \|\mathbf{x}_i - \pi(\mathtt{P}_i\mathbf{X})\| \leq b \wedge (\mathtt{P}_i\mathbf{X})_3 \geq 0\right\}$ is a convex cone

$$\|\mathbf{x}_i - \pi(\mathtt{P}_i\mathbf{X})\| \leq b \iff \|(\mathtt{P}_i\mathbf{X})_3\mathbf{x}_i - (\mathtt{P}_i\mathbf{X})_{1,2}\| - b \cdot (\mathtt{P}_i\mathbf{X})_3 \leq 0$$

- Intersection of convex sets is convex
- For fixed $b \geq 0$: convex feasibility problem!
  - Global solution
  - Iterative convex optimization algorithm, no closed-form solution

Optimal Triangulation: Uniform Noise

Optimal L_∞ triangulation

For $b > 0$ find $\mathbf{X}$ that satisfies

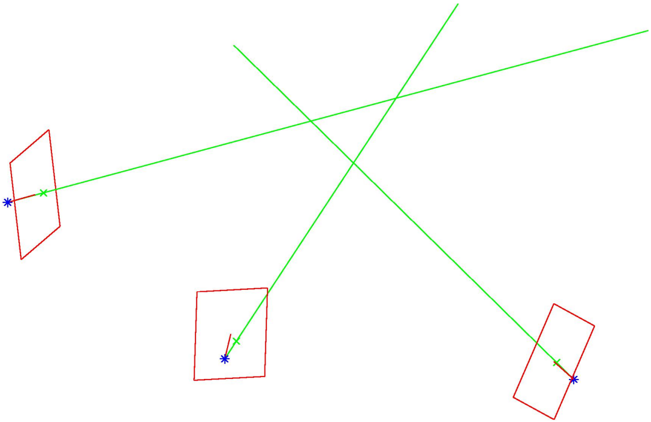
$$\|\mathbf{x}_i - \pi(\mathtt{P}_i\mathbf{X})\| \leq b.$$

- $\left\{\mathbf{X} : \|\mathbf{x}_i - \pi(\mathtt{P}_i\mathbf{X})\| \leq b \wedge (\mathtt{P}_i\mathbf{X})_3 \geq 0\right\}$ is a convex cone

$$\|\mathbf{x}_i - \pi(\mathtt{P}_i\mathbf{X})\| \leq b \iff \|(\mathtt{P}_i\mathbf{X})_3\mathbf{x}_i - (\mathtt{P}_i\mathbf{X})_{1,2}\| - b \cdot (\mathtt{P}_i\mathbf{X})_3 \leq 0$$

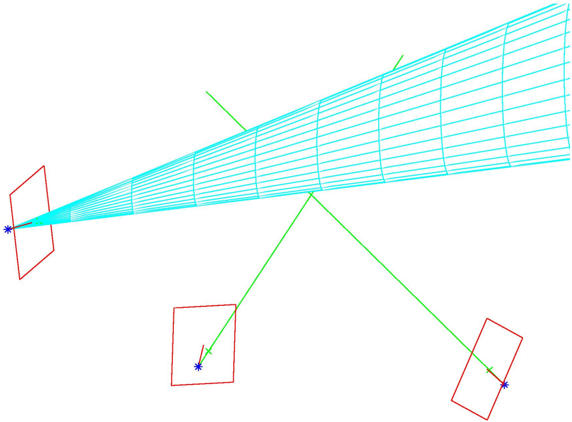
- Intersection of convex sets is convex
- For fixed $b \geq 0$: convex feasibility problem!
  - Global solution
  - Iterative convex optimization algorithm, no closed-form solution

Optimal Triangulation: Uniform Noise



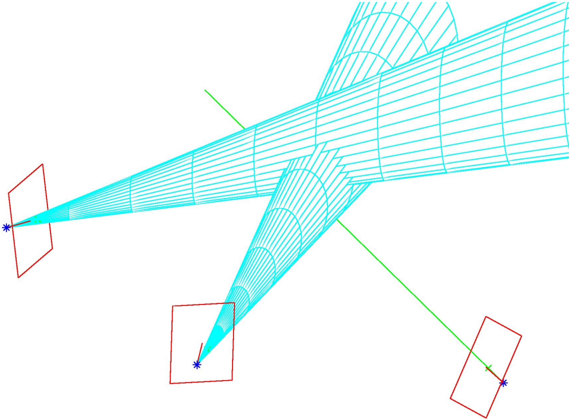
The 3D point $\mathbf{X}$ must lie in the intersection of these cones

Optimal Triangulation: Uniform Noise



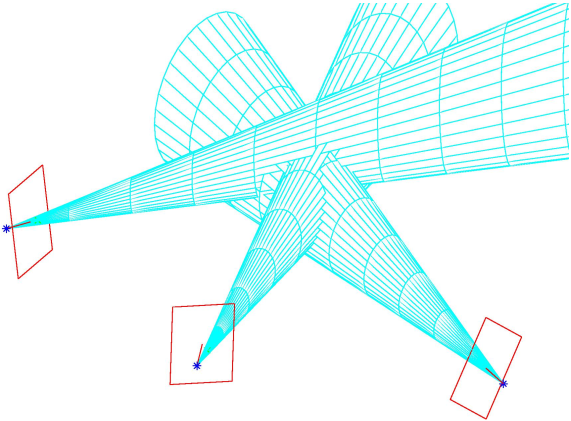
The 3D point $\mathbf{X}$ must lie in the intersection of these cones

Optimal Triangulation: Uniform Noise



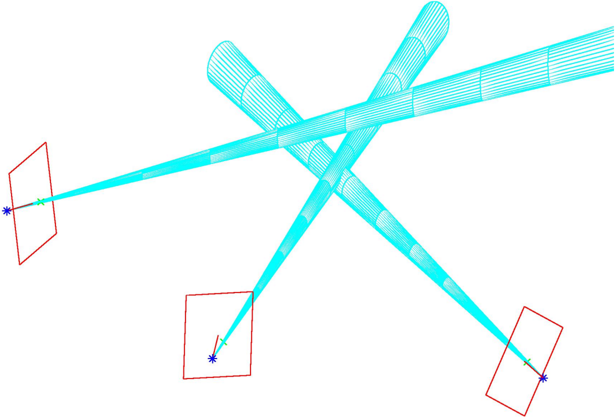
The 3D point $\mathbf{X}$ must lie in the intersection of these cones

Optimal Triangulation: Uniform Noise



The 3D point $\mathbf{X}$ must lie in the intersection of these cones
Many solutions in a convex set

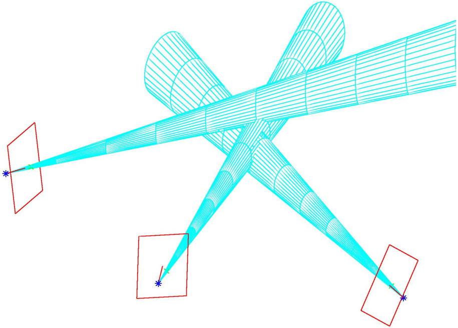
Optimal Triangulation: Uniform Noise



The 3D point $\mathbf{X}$ must lie in the intersection of these cones

No solution

Optimal Triangulation: Uniform Noise



- Reduce the size of the cones $b \implies$ lower the permitted error
- Find the smallest $b \geq 0$ such that the intersection of cones is non-empty

Global Optimization: Triangulation

Algorithm via bisection search

Minimizes the maximal reprojection error. Finds the smallest possible b for which there is a solution $\mathbf{X}$ with all reprojection errors is less than b. That is, solves

$$\min_{\mathbf{X}} \max_i r_i(\mathbf{X}) = \min_{\mathbf{X}} \max_i \|\mathbf{x}_i - \pi(\mathtt{P}_i \mathbf{X})\|$$

1. Let b_l and b_u be lower and upper bound on the optimal error.
2. Check if there is a solution $\mathbf{X}$ such that

$$r_i(\mathbf{X}) = \|\mathbf{x}_i - \pi(\mathtt{P}_i \mathbf{X})\| \leq \frac{b_u + b_l}{2}, \qquad \forall i$$

   (convex optimization problem).
3. If there is set $b_u = \frac{b_u + b_l}{2}$, otherwise set $b_l = \frac{b_u + b_l}{2}$.
4. If $b_u - b_l > tol$ (some predefined tolerance) goto 2.

Global Optimization

Generalizations

Works for other problems as well:

- Camera resectioning

$$\underbrace{\mathbf{x}_i}_{\text{known}} \sim \underbrace{\mathtt{P}}_{\text{unknown}} \underbrace{\mathbf{X}_i}_{\text{known}}$$

- Homography estimation

$$\underbrace{\mathbf{y}_i}_{\text{known}} \sim \underbrace{\mathtt{H}}_{\text{unknown}} \underbrace{\mathbf{x}_i}_{\text{known}}$$

- Structure and motion if camera rotations are known

$$\underbrace{\mathbf{x}_{ij}}_{\text{known}} \sim \left(\underbrace{\mathtt{R}_i}_{\text{known}} \;\middle|\; \underbrace{\mathbf{T}_i}_{\text{unknown}} \right) \underbrace{\mathbf{X}_j}_{\text{unknown}} = \mathtt{R}_i \mathbf{X}_j + \mathbf{T}_i$$

Discussion

Noise assumption

- Is the noise on image points independent?
- Is the zero-mean assumption for noise valid?

Discussion

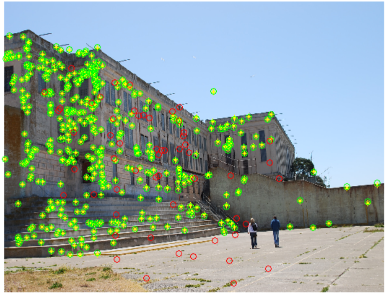
Noise assumption

- Is the noise on image points independent?
  - Not really. There is noise and there are systematic errors.
  - Unhandled lens distortion
- Is the zero-mean assumption for noise valid?

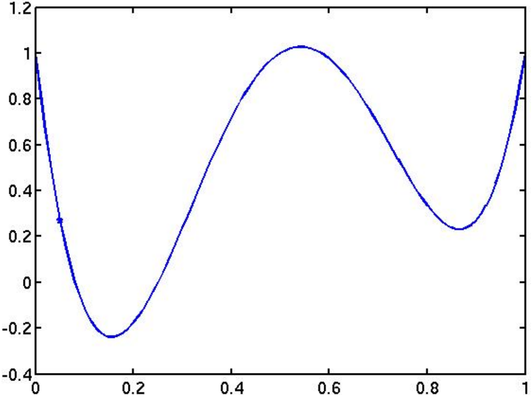
Discussion

Noise assumption

- Is the noise on image points independent?
- Is the zero-mean assumption for noise valid?
  - Errors over the image tend to cancel out on average
  - Systematic bias for unbalanced keypoint distribution

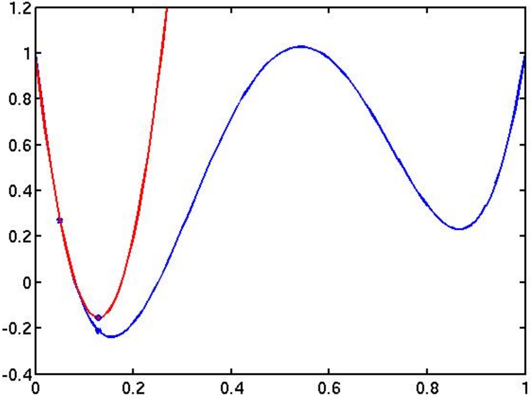


Local Optimization



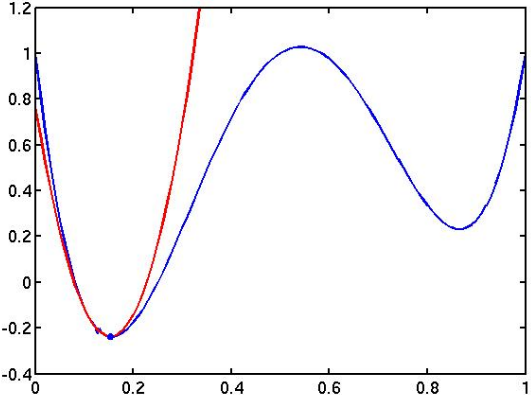
- Pick a starting point.

Local Optimization



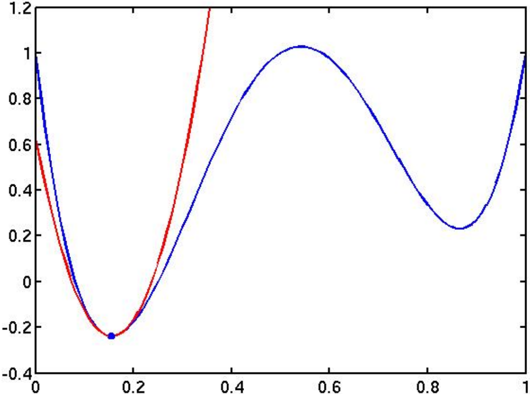
- Approximate the function using 2nd order Taylor expansion and minimize.

Local Optimization



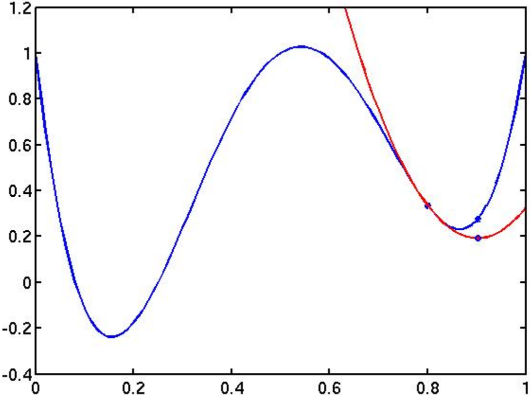
- Repeat.

Local Optimization



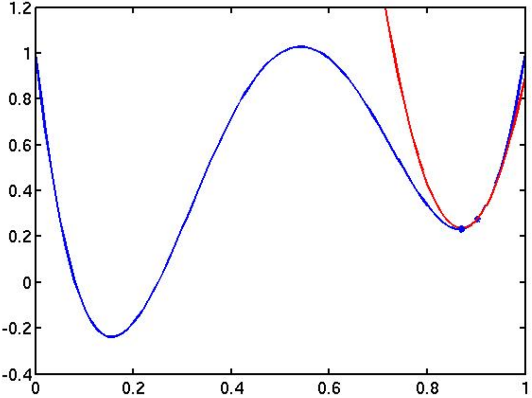
Newtons method.

Local Optimization



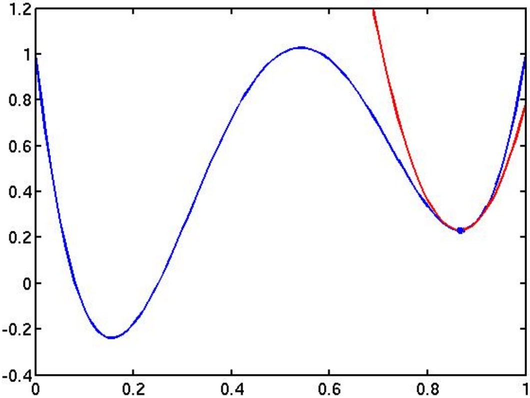
Different starting point.

Local Optimization



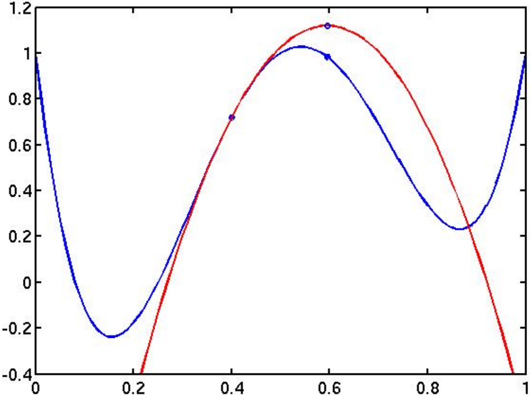
Different starting point.

Local Optimization



Leads to local minimum.

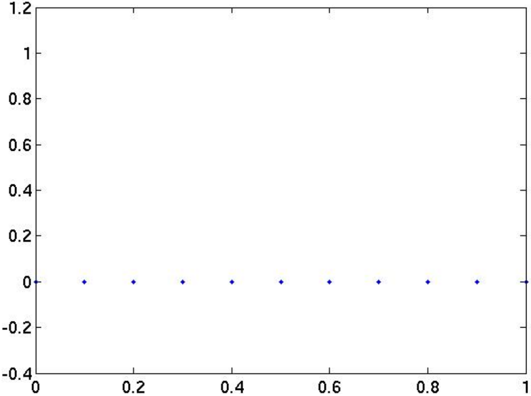
Local Optimization



Third starting point, leads to local maximum.

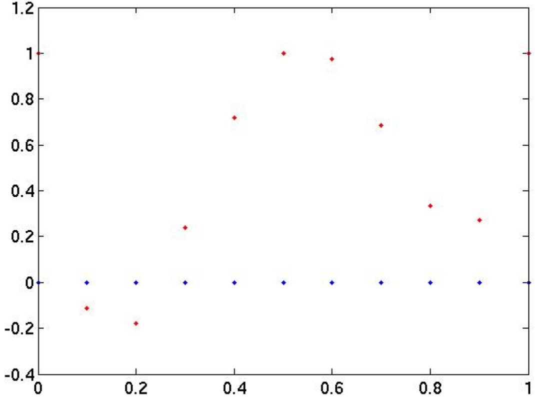
Local Optimization

Why not just sample the function?
One dimensional function:



Local Optimization

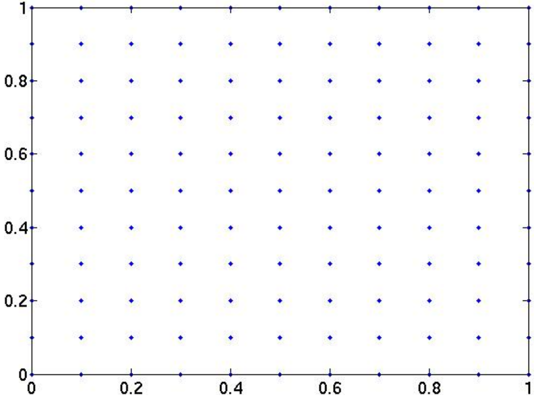
Why not just sample the function?
One dimensional function:



Sample 10 points, pick lowest value. Probably works.

Local Optimization

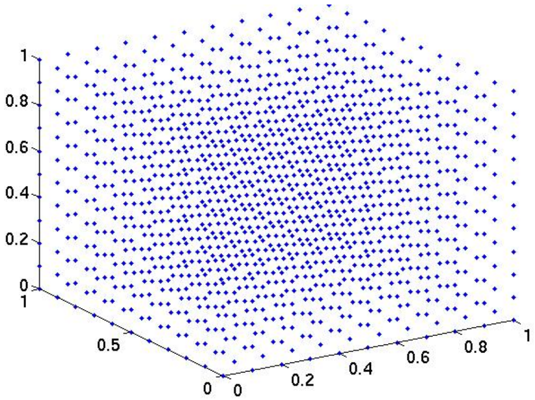
Why not just sample the function?
Two dimensional function:



10^2 samples.

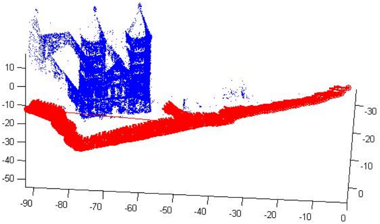
Local Optimization

Why not just sample the function?
Three dimensional function:



10^3 samples.

Local Optimization



How many variables do we have?

The cathedral dataset:

- 480 camera matrices $(\mathtt{R}_i \mid \mathbf{T}_i)$.
  Rotation part 3 dof, translation part 3 dof.
  In total: $480(3+3) = 2880$.
- 91178 3D points.
  3 dof each.
  In total: $91178 \cdot 3 = 273534$

Local Optimization

Goal: 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 many computer vision problems

- Newton's method: 2nd-order 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)
Sometimes a good choice, but specialized methods are preferable

Local Optimization

Goal: 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 many computer vision problems

- Newton's method: 2nd-order 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)
  Sometimes a good choice, but specialized methods are preferable

Local Optimization

Goal: 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 many computer vision problems

- Newton's method: 2nd-order 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)
  Sometimes a good choice, but specialized methods are preferable

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 $\mathbf{f}(x)$

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

- Gauss-Newton method: optimal update δ

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

  Large steps for δ

Non-Linear Least-Squares Optimization

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

- 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 $\mathbf{f}(x)$

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

- Gauss-Newton method: optimal update δ

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

Large steps for δ

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 $\mathtt{H}_F$

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

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

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

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

- Gauss-Newton method: optimal update δ

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

  Large steps for δ

Non-Linear Least-Squares Optimization

Gauss-Newton Method

Task: determine

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

Method: iterate

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

until a stopping criteration is reached

- Problem 1: $\mathsf{J}^\top \mathsf{J}$ may be singular
- Problem 2: $x^{(t)}$ is not guaranteed to converge

Non-Linear Least-Squares Optimization

Levenberg-Marquardt method = Gauss-Newton with adaptive damping

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 $1/\mu$

Non-Linear Least-Squares Optimization

Levenberg-Marquardt method = Gauss-Newton with adaptive damping

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 $1/\mu$

Minimizing the Reprojection Error

General objective with N cameras and M 3D points:

$$\sum_{i,j} m_{ij}\psi\big(\mathbf{x}_{ij} - \pi(\mathtt{P}_i\mathbf{X}_j)\big)$$

$m_{ij} \in \{0,1\}$ indicates whether $\mathbf{X}_j$ is visible in camera $\mathtt{P}_i$.

- Which are the unknowns
  - Only $\{\mathbf{X}_j\}$: triangulation
  - Only $\{\mathtt{P}_i\}$: camera resectioning
  - $\{\mathtt{P}_i\}$ and $\{\mathbf{X}_j\}$: structure from motion (SfM), bundle adjustment
- What is ψ
  - $\psi(\mathbf{r}) = \|\mathbf{r}\|_2^2$: Gaussian noise, non-robust
  - $\psi(\mathbf{r}) = \imath\{\|\mathbf{r}\|_2 \le b\}$: uniform noise, non-robust, may have no solution
  - $\psi(\mathbf{r}) = \min\{\|\mathbf{r}\|_2^2, \tau^2\}$: robust, but hard to minimize
- What is m_{ij}
  - All m_{ij} are 1: no occlusions, every scene point is visible in every image
  - Few m_{ij} are 1: many occlusions / 3D points not in the image

Minimizing the Reprojection Error

General objective with N cameras and M 3D points:

$$\sum_{i,j} m_{ij}\psi\big(\mathbf{x}_{ij} - \pi(\mathtt{P}_i\mathbf{X}_j)\big)$$

$m_{ij} \in \{0,1\}$ indicates whether $\mathbf{X}_j$ is visible in camera $\mathtt{P}_i$.

- Which are the unknowns
  - Only $\{\mathbf{X}_j\}$: triangulation
  - Only $\{\mathtt{P}_i\}$: camera resectioning
  - $\{\mathtt{P}_i\}$ and $\{\mathbf{X}_j\}$: structure from motion (SfM), bundle adjustment
- What is ψ
  - $\psi(\mathbf{r}) = \|\mathbf{r}\|_2^2$: Gaussian noise, non-robust
  - $\psi(\mathbf{r}) = \imath\{\|\mathbf{r}\|_2 \le b\}$: uniform noise, non-robust, may have no solution
  - $\psi(\mathbf{r}) = \min\{\|\mathbf{r}\|_2^2, \tau^2\}$: robust, but hard to minimize
- What is m_{ij}
  - All m_{ij} are 1: no occlusions, every scene point is visible in every image
  - Few m_{ij} are 1: many occlusions / 3D points not in the image

Minimizing the Reprojection Error

General objective with N cameras and M 3D points:

$$\sum_{i,j} m_{ij}\psi\big(\mathbf{x}_{ij} - \pi(\mathtt{P}_i\mathbf{X}_j)\big)$$

$m_{ij} \in \{0,1\}$ indicates whether $\mathbf{X}_j$ is visible in camera $\mathtt{P}_i$.

- Which are the unknowns
  - Only $\{\mathbf{X}_j\}$: triangulation
  - Only $\{\mathtt{P}_i\}$: camera resectioning
  - $\{\mathtt{P}_i\}$ and $\{\mathbf{X}_j\}$: structure from motion (SfM), bundle adjustment
- What is ψ
  - $\psi(\mathbf{r}) = \|\mathbf{r}\|_2^2$: Gaussian noise, non-robust
  - $\psi(\mathbf{r}) = \imath\{\|\mathbf{r}\|_2 \le b\}$: uniform noise, non-robust, may have no solution
  - $\psi(\mathbf{r}) = \min\{\|\mathbf{r}\|_2^2, \tau^2\}$: robust, but hard to minimize
- What is m_{ij}
  - All m_{ij} are 1: no occlusions, every scene point is visible in every image
  - Few m_{ij} are 1: many occlusions / 3D points not in the image

Minimizing the Reprojection Error

General objective with N cameras and M 3D points:

$$\sum_{i,j} m_{ij}\psi\big(\mathbf{x}_{ij} - \pi(\mathtt{P}_i\mathbf{X}_j)\big)$$

$m_{ij} \in \{0,1\}$ indicates whether $\mathbf{X}_j$ is visible in camera $\mathtt{P}_i$.

- Which are the unknowns
  - Only $\{\mathbf{X}_j\}$: triangulation
  - Only $\{\mathtt{P}_i\}$: camera resectioning
  - $\{\mathtt{P}_i\}$ and $\{\mathbf{X}_j\}$: structure from motion (SfM), bundle adjustment
- What is ψ
  - $\psi(\mathbf{r}) = \|\mathbf{r}\|_2^2$: Gaussian noise, non-robust
  - $\psi(\mathbf{r}) = \imath\{\|\mathbf{r}\|_2 \le b\}$: uniform noise, non-robust, may have no solution
  - $\psi(\mathbf{r}) = \min\{\|\mathbf{r}\|_2^2, \tau^2\}$: robust, but hard to minimize
- What is m_{ij}
  - All m_{ij} are 1: no occlusions, every scene point is visible in every image
  - Few m_{ij} are 1: many occlusions / 3D points not in the image

Minimizing the Reprojection Error

Main goal

For given $\{\mathbf{x}_{ij}\}$ and $\{m_{ij}\}$ find a minimizer

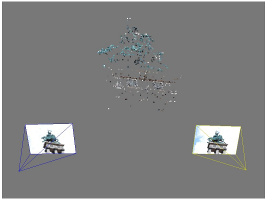
$$\sum_{i,j} m_{ij}\psi\big(\mathbf{x}_{ij} - \pi(\mathtt{P}_i\mathbf{X}_j)\big) \to \min_{\{\mathtt{P}_i\},\{\mathbf{X}_j\}} .$$

Local optimization needs good starting point

$N = 2$: Recall lecture 5+6

- Compute $\mathtt{F}/\mathtt{E}$
- Compute a pair of camera matrices: $\mathtt{P}_1 = (\mathtt{I} \mid \mathbf{0})$, $\mathtt{P}_2 = ([\mathbf{e}_2]_\times \mathtt{F} \mid \mathbf{e}_2)$
- Triangulate $\mathbf{X}_j$ using DLT





Minimizing the Reprojection Error

Main goal

For given $\{\mathbf{x}_{ij}\}$ and $\{m_{ij}\}$ find a minimizer

$$\sum_{i,j} m_{ij}\psi\big(\mathbf{x}_{ij} - \pi(\mathtt{P}_i\mathbf{X}_j)\big) \to \min_{\{\mathtt{P}_i\},\{\mathbf{X}_j\}} .$$

Local optimization needs good starting point

$N > 2$: Sequential / incremental SfM

. . .

How do we compute the entire reconstruction?

1. For a pair of images compute initial reconstruction ($N = 2$)
2. For a new image viewing some of the previously reconstructed scene points, find the camera matrix (via DLT)
3. Compute new scene points using triangulation (DLT)
4. If there are more cameras/images goto step 2

Minimizing the Reprojection Error

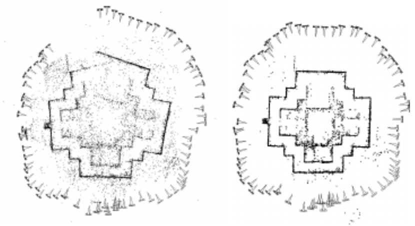
Main goal

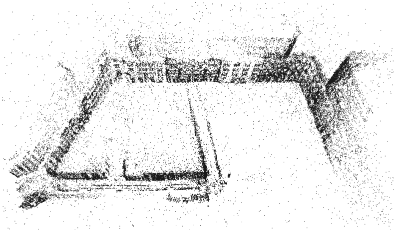
For given $\{\mathbf{x}_{ij}\}$ and $\{m_{ij}\}$ find a minimizer

$$\sum_{i,j} m_{ij}\psi\big(\mathbf{x}_{ij} - \pi(\mathtt{P}_i\mathbf{X}_j)\big) \to \min_{\{\mathtt{P}_i\},\{\mathbf{X}_j\}} .$$

Local optimization needs good starting point

$N > 2$: Sequential / incremental SfM
Main drawback: drift due to error accumulation





Minimizing the Reprojection Error

Main goal

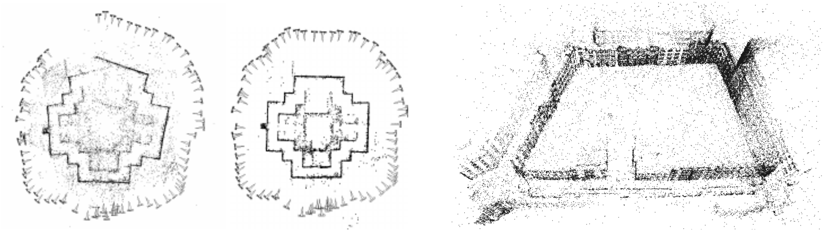
For given $\{\mathbf{x}_{ij}\}$ and $\{m_{ij}\}$ find a minimizer

$$\sum_{i,j} m_{ij}\psi\big(\mathbf{x}_{ij} - \pi(\mathtt{P}_i\mathbf{X}_j)\big) \to \min_{\{\mathtt{P}_i\},\{\mathbf{X}_j\}} .$$

Local optimization needs good starting point

$N > 2$: Sequential / incremental SfM

Main drawback: drift due to error accumulation



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?

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

To do

- Work on assignment 3
- Next time: non-sequential methods for SfM and project presentation!

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