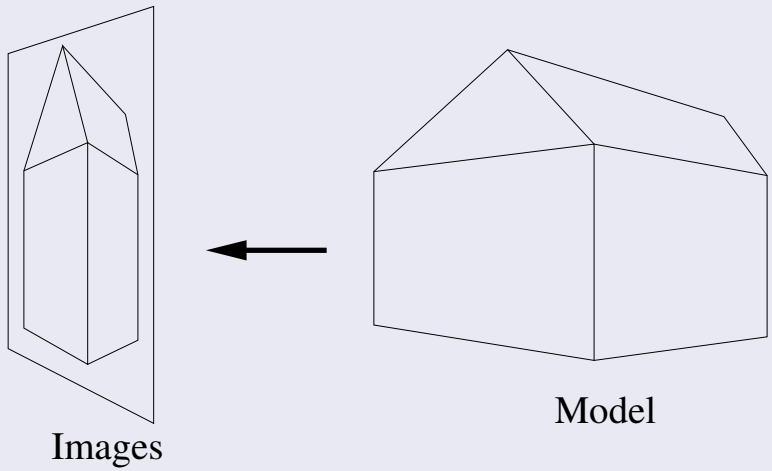
Introduction to Computer Vision

Carl Olsson

2018-10-16

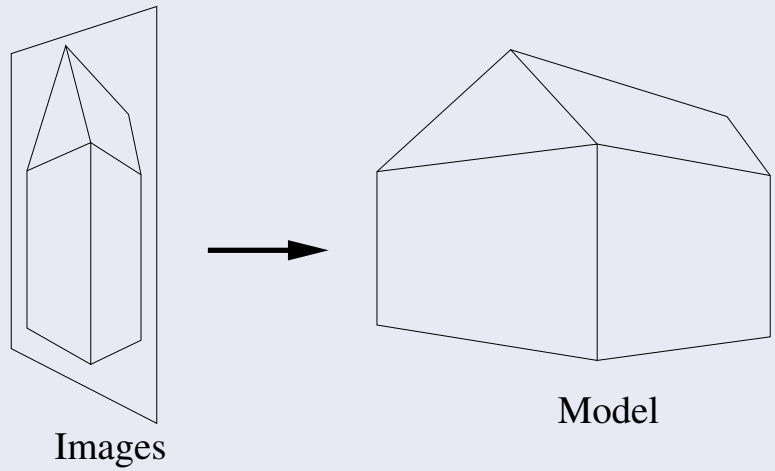
What is Computer Vision?

Computer Graphics



Generate images from a 3D model.

Computer Vision



The inverse problem: Generate 3D model from images.

Multiview Reconstruction

Given Images




4 images out of a sequence with 435 images.

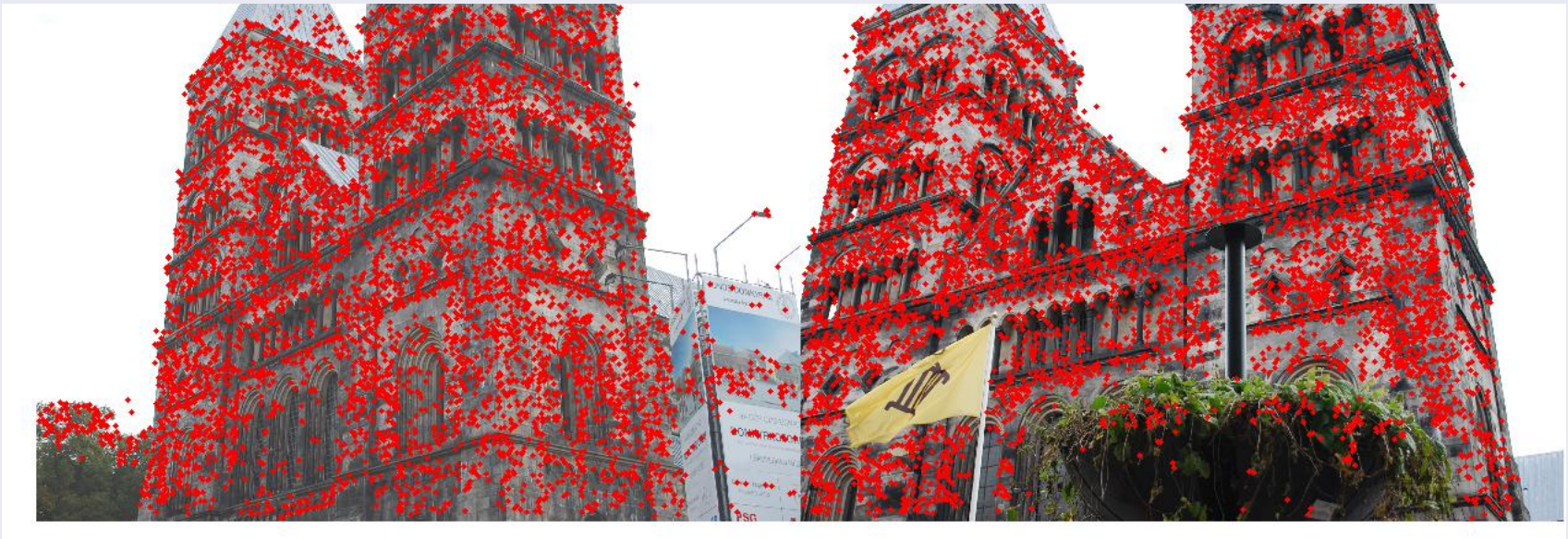
Compute 3D Model

Reconstruction Pipeline

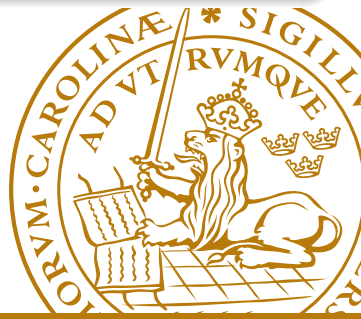
Point Detection and Matching

Reconstruction Pipeline

Point Detection and Matching

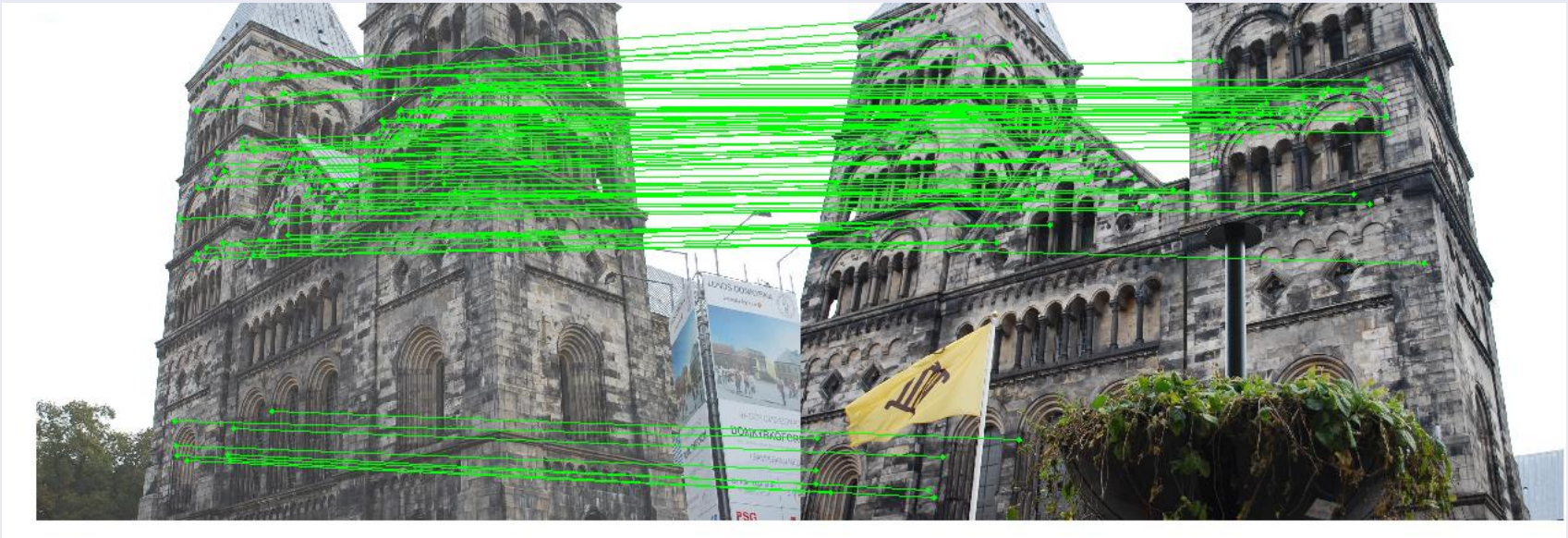


Detect interesting (descriptive) points in all images.

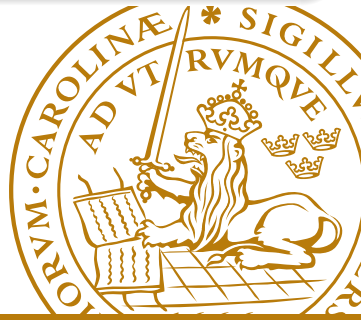


Reconstruction Pipeline

Point Detection and Matching

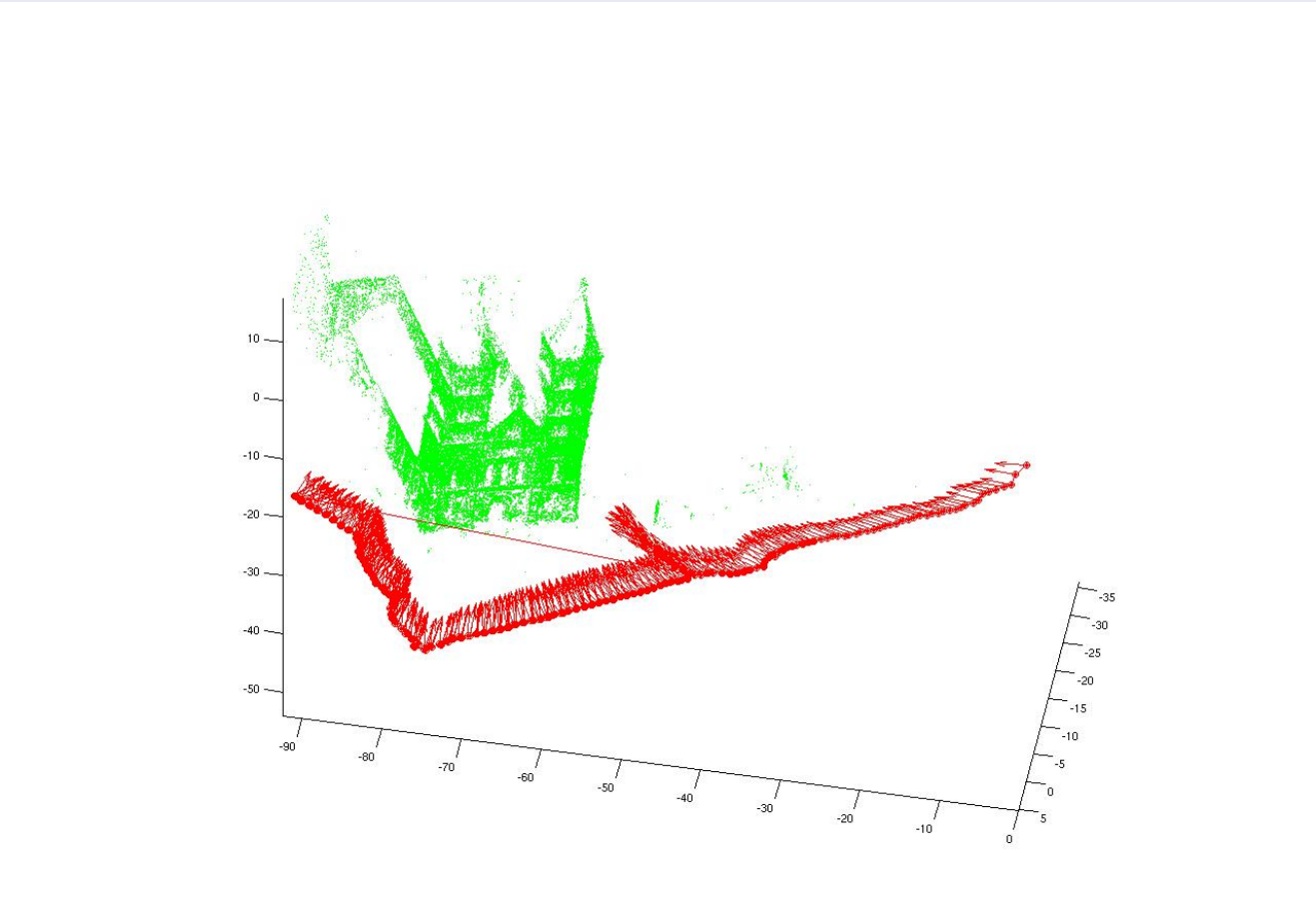


Match points between images.



Reconstruction Pipeline

Geometric Computations (main part of the Computer Vison course!)

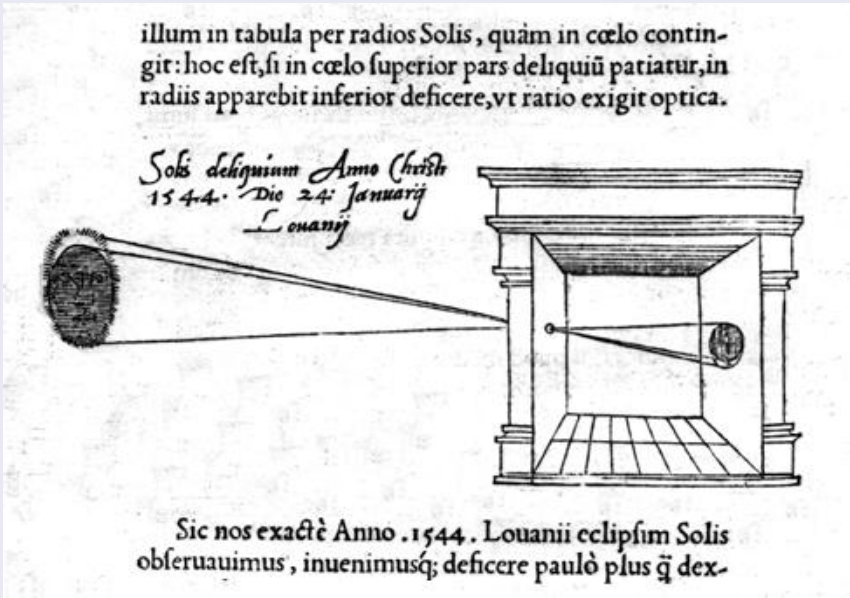


Compute 3D-positions of the matched points, position and orientation of the cameras.

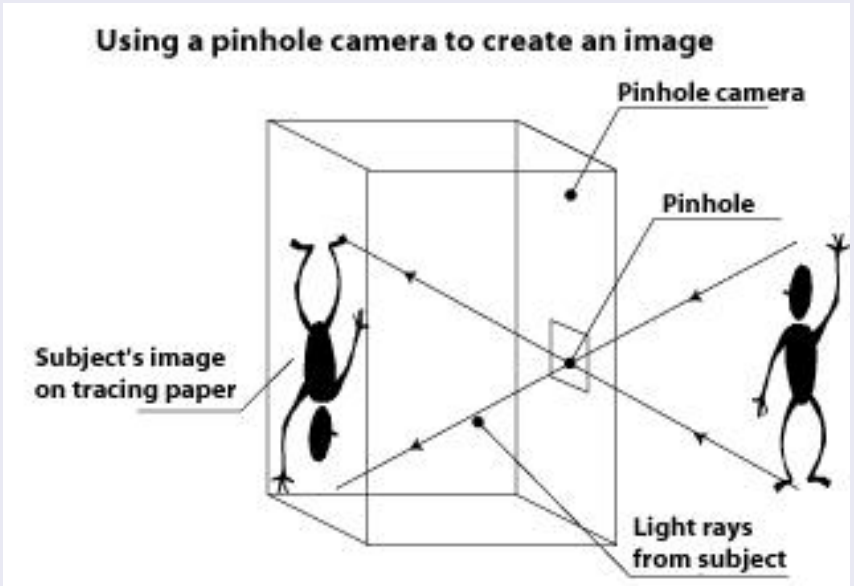
Video

Camera Model

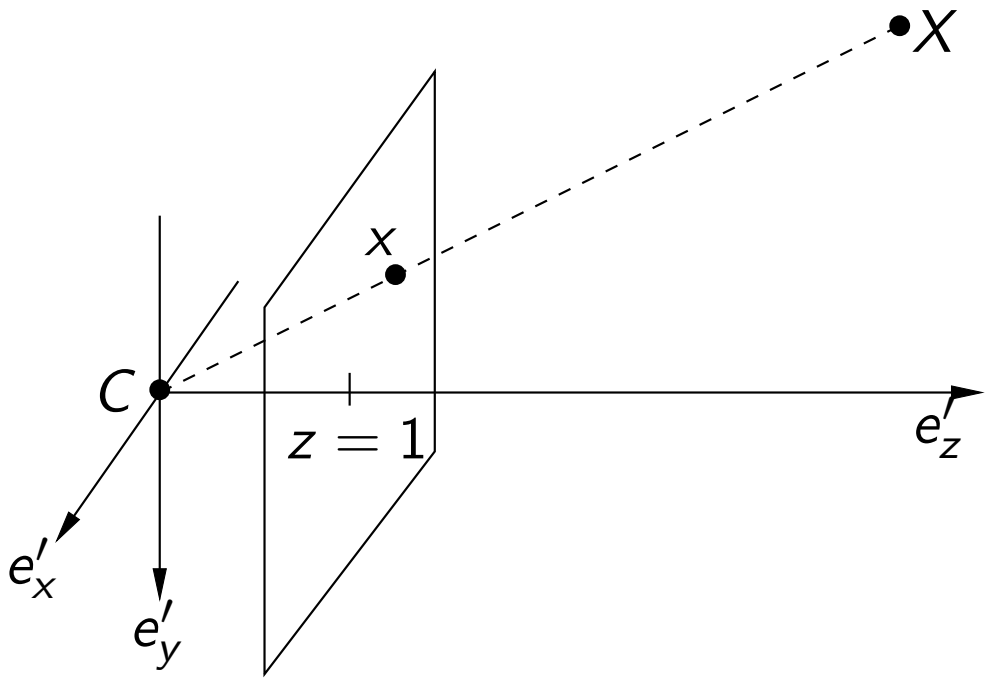
The Pinhole Camera



Reinerus Gemma-Frisus camera obscura from 1544.



Pinhole Projection



$$C = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \Rightarrow \lambda \begin{pmatrix} x_1 \\ x_2 \\ 1 \end{pmatrix} = \begin{pmatrix} X_1 \\ X_2 \\ X_3 \end{pmatrix}, \ \lambda \in \mathbb{R}$$

$$\Rightarrow (x_1, x_2) = (\frac{X_1}{X_3}, \frac{X_2}{X_3})$$

Homogeneous Coordinates & Moving Cameras

Projections with **homogeneous** coord:

$$\lambda \mathbf{x} = \begin{bmatrix} I & 0 \end{bmatrix} \mathbf{X}, \quad \text{where } \mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ 1 \end{pmatrix}, \mathbf{X} = \begin{pmatrix} X_1 \\ X_2 \\ X_3 \\ 1 \end{pmatrix},$$

Camera motion can be modeled using

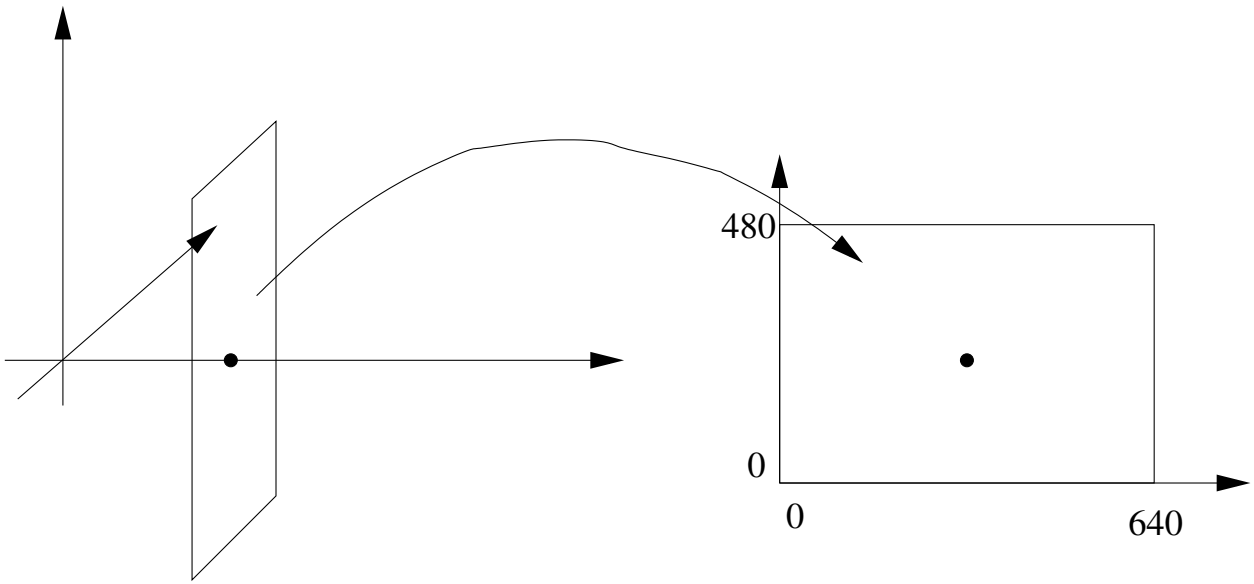
$$\lambda \mathbf{x} = \begin{bmatrix} R & t \end{bmatrix} \mathbf{X}$$

Extrinsic parameters: R-rotation matrix, t-translation vector.

The Inner Parameters -K

$$\begin{pmatrix} fx + x_0 \\ fy + y_0 \\ 1 \end{pmatrix} = \begin{bmatrix} f & 0 & x_0 \\ 0 & f & y_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

f-focal length, (x_0, y_0) principal point. Re-centers and scales (e.g. meters $\rightarrow$ pixels) the image. Typically transforms the point $(0, 0, 1)$ to the middle of the image.



The Inner Parameters -K

The most general version of K is the upper triangular matrix:

$$K = \begin{bmatrix} \gamma f & sf & x_0 \\ 0 & f & y_0 \\ 0 & 0 & 1 \end{bmatrix}.$$

- f - focal length
- γ - aspect ratio
- s - skew
- (x_0, y_0) - principal points

Camera equations:

$$\lambda \mathbf{x} = K \begin{bmatrix} R & t \end{bmatrix} \mathbf{X} = P\mathbf{X}$$

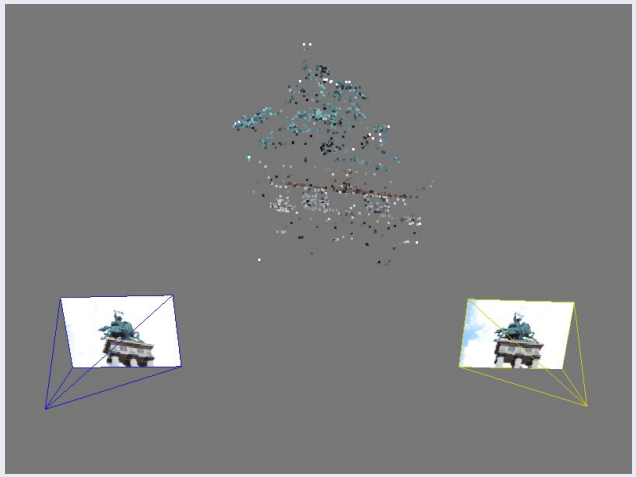
RQ factorization: Any 3×4 matrix P can be written $K \begin{bmatrix} R & t \end{bmatrix}$, with K triangular and R orthogonal.

Relative Orientation: Problem Formulation

Given

Two images and corresponding points.

Compute



The structure (3D-points) and the motion (camera matrices).

Relative Orientation

Problem Formulation

Given corresponding points $\mathbf{x}_i$ and $\bar{\mathbf{x}}_i$, $i = 1, ..., n$ find cameras P_1 and P_2 and 3D points $\mathbf{X}_i$ such that

$$\lambda_i \mathbf{x}_i = P_1 \mathbf{X}_i$$

and

$$\bar{\lambda}_i \bar{\mathbf{x}}_i = P_2 \mathbf{X}_i.$$

Ambiguities (uncalibrated case)

Can always apply a projective transformation H to archive a different solution

$$\lambda_i \mathbf{x}_i = P_1 H H^{-1} \mathbf{X}_i = \tilde{P}_1 \tilde{\mathbf{X}}_i$$

and

$$\bar{\lambda}_i \bar{\mathbf{x}}_i = P_2 H H^{-1} \mathbf{X}_i = \tilde{P}_2 \tilde{\mathbf{X}}_i.$$

Relative Orientation: Problem Formulation

Simplification

If $P_1 = \begin{bmatrix} A_1 & t_1 \end{bmatrix}$ and $P_2 = \begin{bmatrix} A_2 & t_2 \end{bmatrix}$, apply the transformation

$$H = \begin{bmatrix} A_1^{-1} & -A_1^{-1} t_1 \\ 0 & 1 \end{bmatrix}.$$

Then

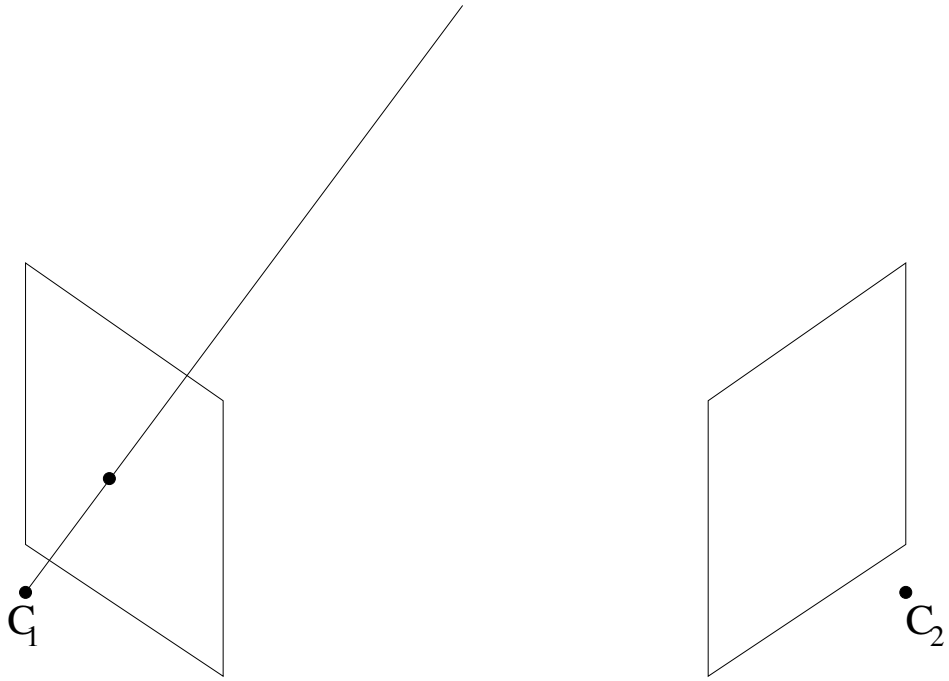
$$P_1 H = \begin{bmatrix} A_1 & t_1 \end{bmatrix} \begin{bmatrix} A_1^{-1} & -A_1^{-1} t_1 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} I & 0 \end{bmatrix}.$$

Hence, we may assume that the cameras are

$$P_1 = \begin{bmatrix} I & 0 \end{bmatrix} \text{ and } P_2 = \begin{bmatrix} A & t \end{bmatrix}$$

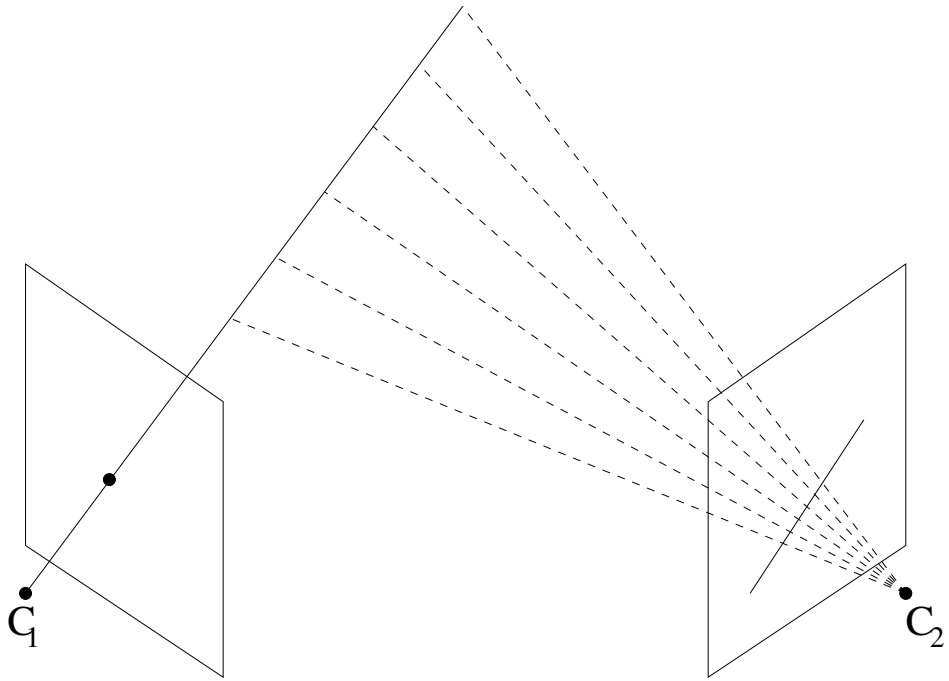
Still hard to solve since P_2 and $\mathbf{X}_i$ are unknown!

Epipolar Geometry



Consider a single point **x** in the first image. Any point on the line projects to this point.

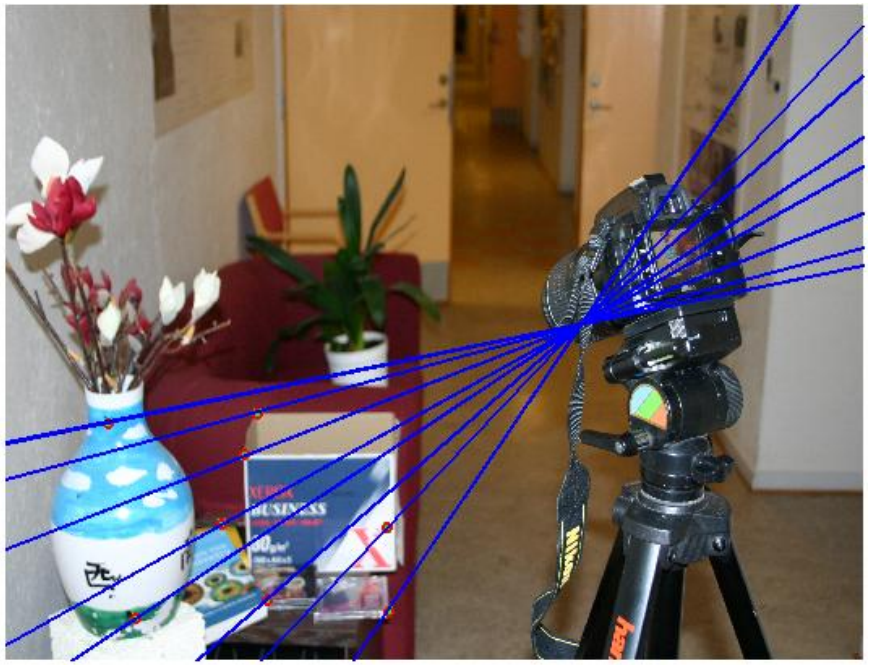
Epipolar Geometry



Any point on the projection of the 3D line can correspond to $\mathbf{x}$.

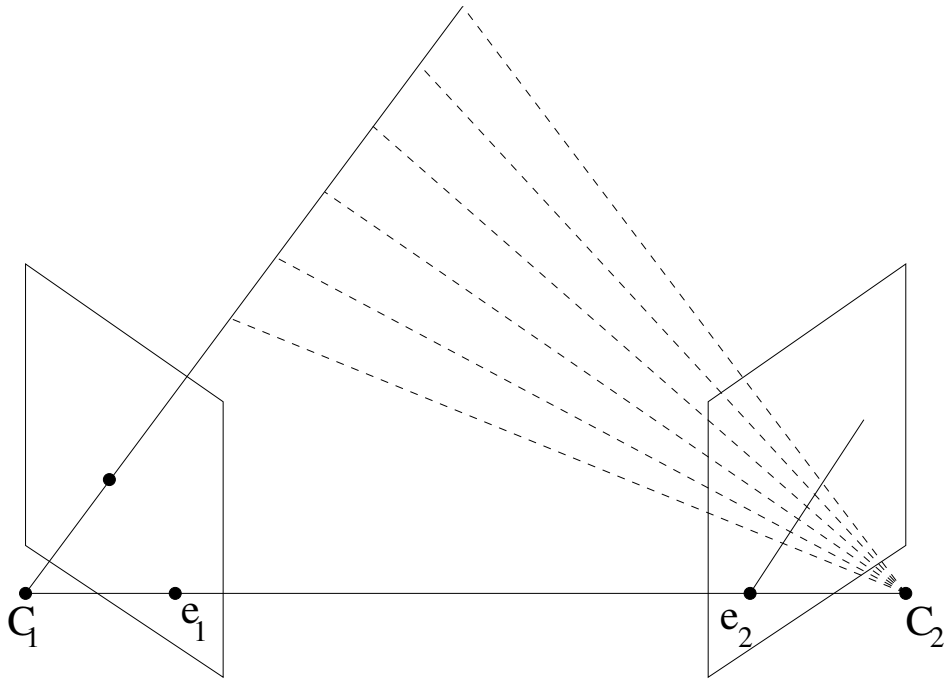
Epipolar Geometry

Epipolar Geometry



The projected lines should all meet in a point. The so called **epipole** is the projection of the camera center of the other camera.

Epipolar Geometry



The epipole e_1 is the projection of the C_2 in P_1.
The epipole e_2 is the projection of the C_1 in P_2.
e_1, e_2 usually outside field of view.

The Fundamental Matrix

If $\mathbf{x}$ and $\bar{\mathbf{x}}$ are projections of X in $P_1 = \begin{bmatrix} I & 0 \end{bmatrix}$ and $P_2 = \begin{bmatrix} A & t \end{bmatrix}$ then

$$\bar{\mathbf{x}}^T F \mathbf{x} = 0,$$

where $F = [t]_\times A$.

- F - Fundamental matrix.
- $\bar{\mathbf{x}}^T F \mathbf{x} = 0$ - epipolar constraint,

$$e_2^T F \mathbf{x} = 0 \quad \forall \mathbf{x} \Rightarrow e_2^T F = 0 \Rightarrow \det(F^T) = \det(F) = 0.$$

Search for the fundamental matrix song on youtube!

The Fundamental Matrix

Estimating F

If $\mathbf{x}_i$ and $\bar{\mathbf{x}}_i$ corresponding points

$$\bar{\mathbf{x}}_i^T F \mathbf{x}_i = 0.$$

If $\mathbf{x}_i = (x_i, y_i, z_i)$ and $\bar{\mathbf{x}}_i = (\bar{x}_i, \bar{y}_i, \bar{z}_i)$ then

$$\begin{aligned}\bar{\mathbf{x}}_i^T F \mathbf{x}_i = \quad & F_{11}\bar{x}_i x_i + F_{12}\bar{x}_i y_i + F_{13}\bar{x}_i z_i \\ & + F_{21}\bar{y}_i x_i + F_{22}\bar{y}_i y_i + F_{23}\bar{y}_i z_i \\ & + F_{31}\bar{z}_i x_i + F_{32}\bar{z}_i y_i + F_{33}\bar{z}_i z_i\end{aligned}$$

The Fundamental Matrix

The epipolar constraints only contain camera information. The 3D-points have been eliminated.

Estimating F

In matrix form (one row for each correspondence):

$$\underbrace{\begin{bmatrix} \bar{x}_1 x_1 & \bar{x}_1 y_1 & \bar{x}_1 z_1 & \dots & \bar{z}_1 z_1 \\ \bar{x}_2 x_2 & \bar{x}_2 y_2 & \bar{x}_2 z_2 & \dots & \bar{z}_2 z_2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \bar{x}_n x_n & \bar{x}_n y_n & \bar{x}_n z_n & \dots & \bar{z}_n z_n \end{bmatrix}}_{M} \begin{bmatrix} F_{11} \\ F_{12} \\ F_{13} \\ \vdots \\ F_{33} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Solve using homogeneous least squares (svd).
F has 9 entries (but the scale is arbitrary). Need at least 8 equations (point correspondences).

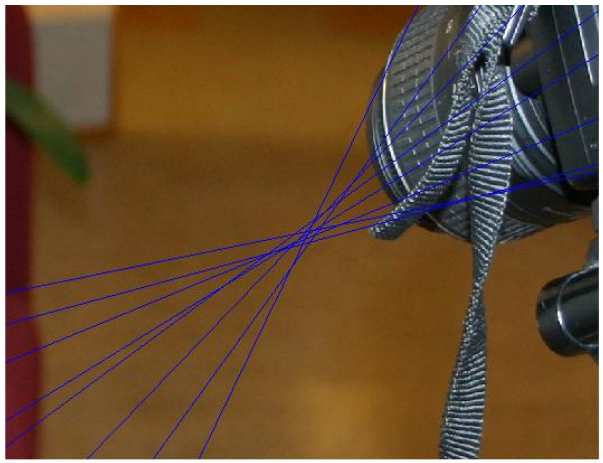
The Fundamental Matrix

Issues

Resulting F may not have $det(F) = 0$.
Pick the closest matrix A with $det(A) = 0$.

Can be solved using svd, in matlab:

$$[U, S, V] = svd(F);$$
$$S(3, 3) = 0;$$
$$A = U * S * V';$$



Extracting cameras from F

A family of solutions:

$$P_1 = \begin{bmatrix} I & 0 \end{bmatrix}$$

$$P_2 = \begin{bmatrix} [e_2]_\times F + e_2 v^T & \lambda e_2 \end{bmatrix},$$

$e_2 \in \text{Null}(F^T)$, $v \in \mathbb{R}^3$, $\lambda \neq 0$.

Calibrated Relative Orientation: Problem Formulation

Simplification

If $P_1 = \begin{bmatrix} R_1 & t_1 \end{bmatrix}$ and $P_2 = \begin{bmatrix} R_2 & t_2 \end{bmatrix}$, apply the transformation

$$H = \begin{bmatrix} R_1^T & -R_1^T t_1 \\ 0 & 1 \end{bmatrix}.$$

Then

$$P_1 H = \begin{bmatrix} R_1 & t_1 \end{bmatrix} \begin{bmatrix} R_1^T & -R_1^T t_1 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} I & 0 \end{bmatrix}.$$

Hence, we may assume that the cameras are

$$P_1 = \begin{bmatrix} I & 0 \end{bmatrix} \text{ and } P_2 = \begin{bmatrix} R & t \end{bmatrix}$$

The Essential Matrix

The Essential Matrix

The camera pair $P_1 = \begin{bmatrix} I & 0 \end{bmatrix}$ and $P_2 = \begin{bmatrix} R & t \end{bmatrix}$ has the fundamental matrix

$$E = [t]_\times R.$$

E is called the essential matrix.

- R has 3 dof, t 3 dof, but the scale is arbitrary, therefore E has 5 dof.
- E has $det(E) = 0$
- E has two nonzero equal singular values.

Solve using the (non-linear) 5-point solver. (Gives 10 solutions.)

Computing the cameras

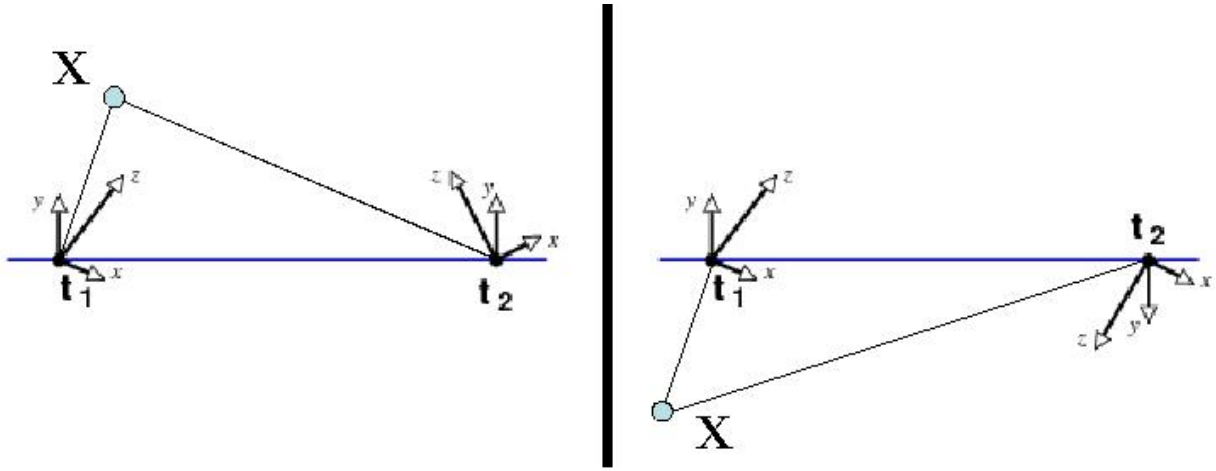
Want to find $P_2 = [R\,t]$ such that $E = [t]_\times R$.
Outline:

- Ensure that E has the SVD

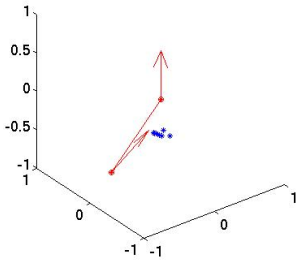
$$E = USV^T$$

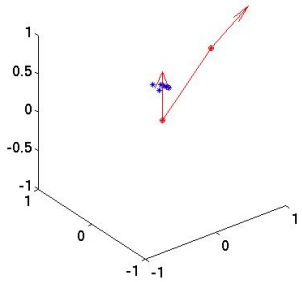
  where $det(UV^T) = 1$.
- Compute a factorization $E = SR$ where S is skew symmetric and R a rotation.
- Compute a t such that $[t]_\times = S$.
- Form the camera $P_2 = [R\,t]$.

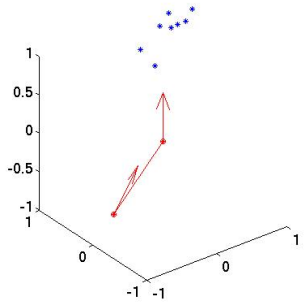
The Twisted Pair

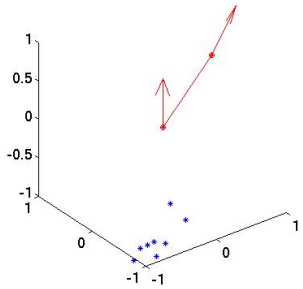


4 Solutions



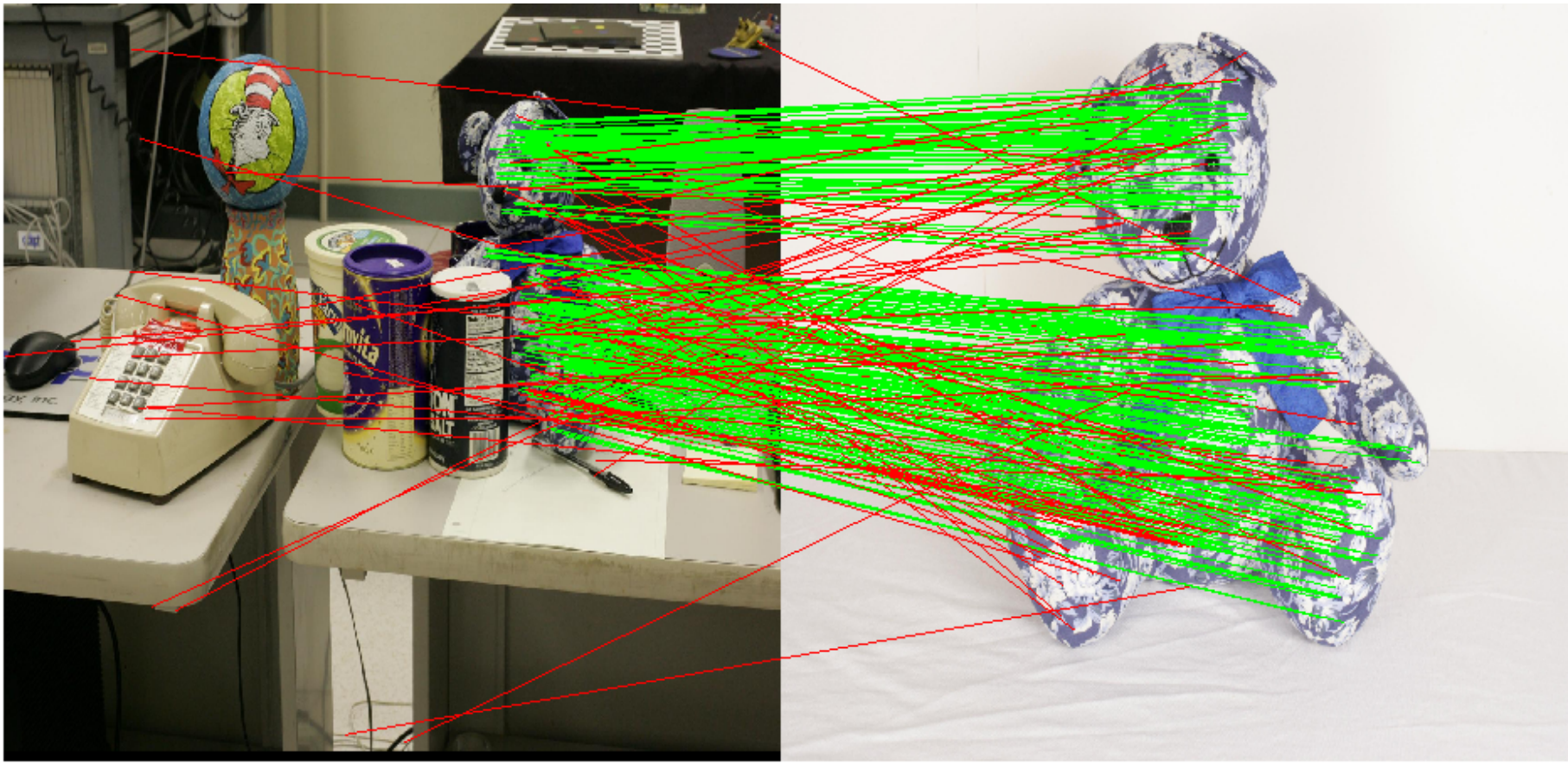






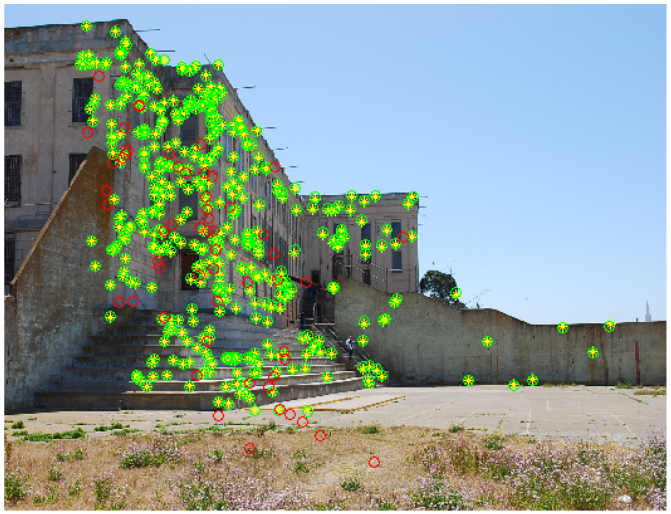
Robust fitting with RANSAC

The outlier problem:

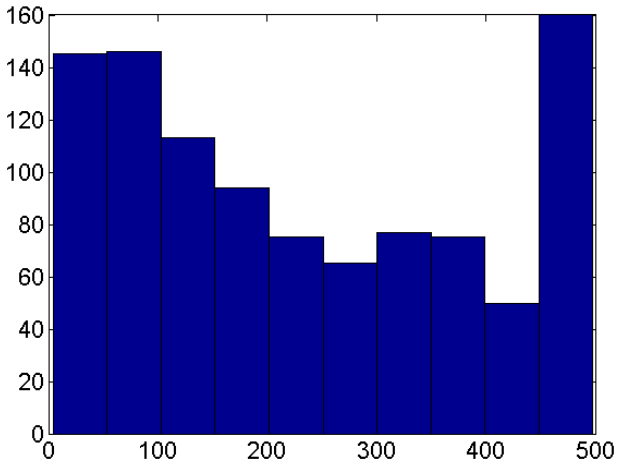


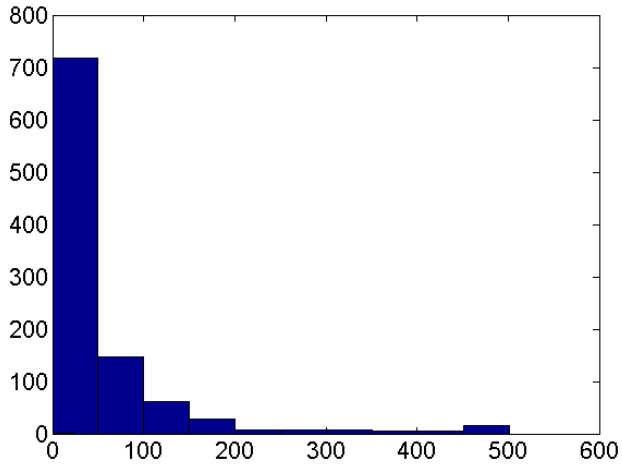
Typical matching result.

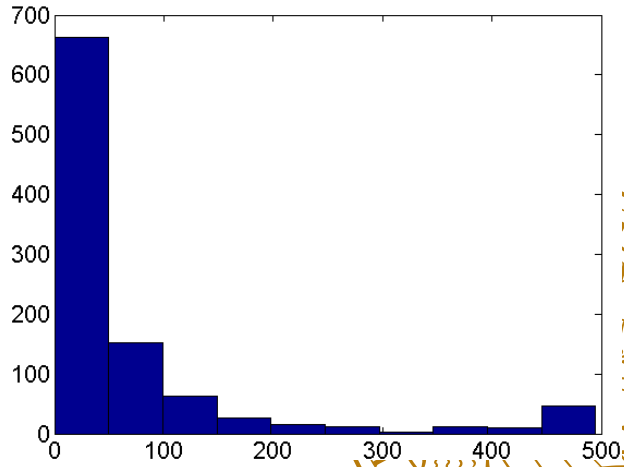
The 5-point solver



Histogram over the size of the consensus set in each iteration of RANSAC (1000 iterations), using 5 points, 8 points and 10 points respectively.







Handling Noise - Minimizing Reprojection Error

Gaussian 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.

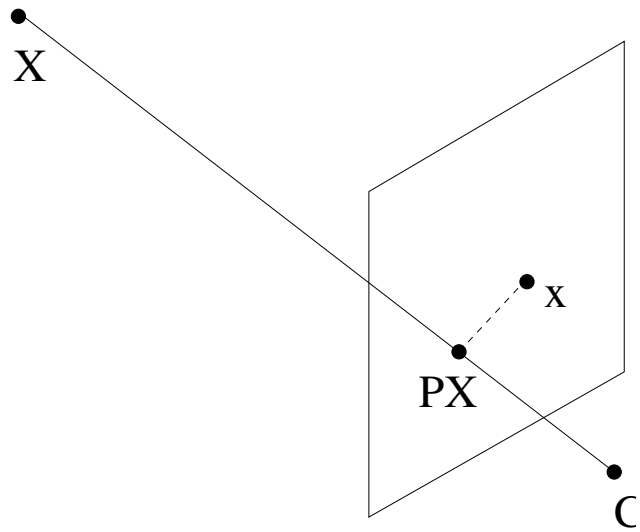
The reprojection error

In regular coordinates the projection is

$$\left(\frac{P^1\mathbf{X}}{P^3\mathbf{X}}, \frac{P^2\mathbf{X}}{P^3\mathbf{X}} \right),$$

P^1, P^2, P^3 are the rows of P.
The reprojection error is

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



Minimizing Reprojection Error

Calibrated Structure and Motion

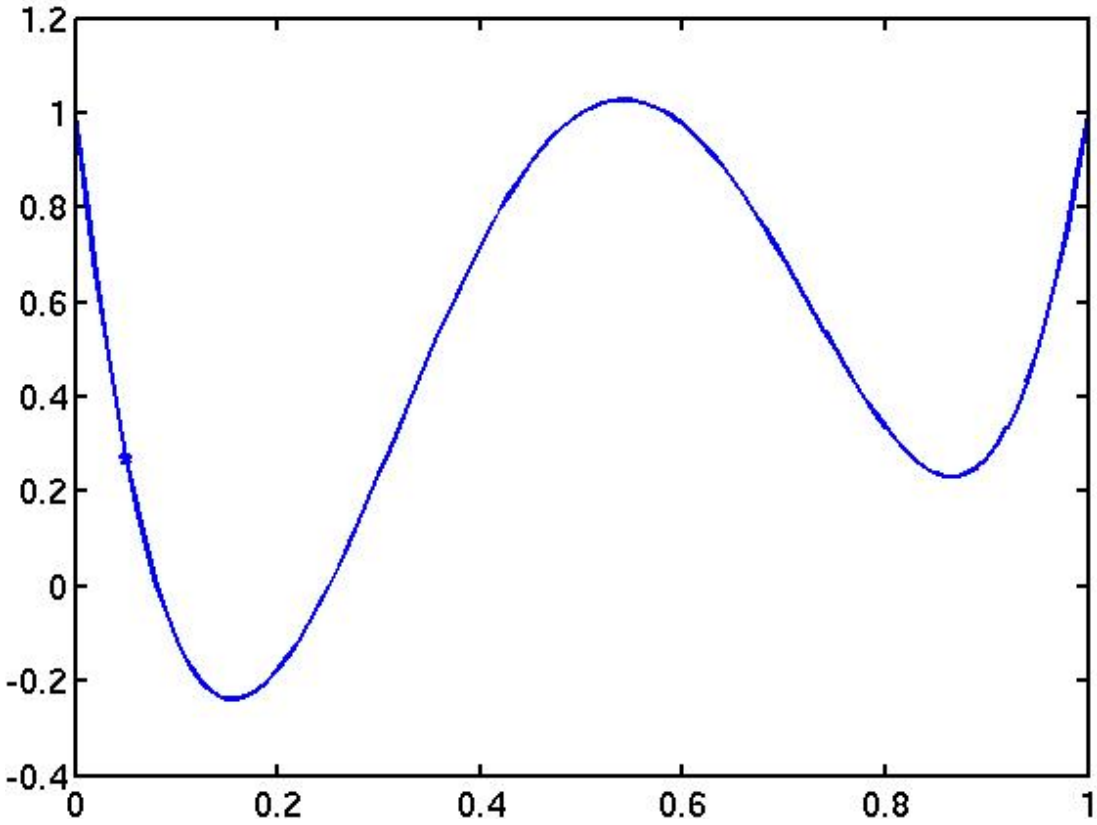
Given image projections $\{(x_{ij}, y_{ij})\}$ (i = point nr, j = image nr), find 3D points X_i and cameras $P_j = \begin{bmatrix} R_j & t_j \end{bmatrix}$ such that

$$\sum_{ij} || \left(x_{ij} - \frac{P_j^1 \mathbf{X}_i}{P_j^3 \mathbf{X}_i}, y_{ij} - \frac{P_j^2 \mathbf{X}_i}{P_j^3 \mathbf{X}_i} \right) ||^2,$$

is minimized.

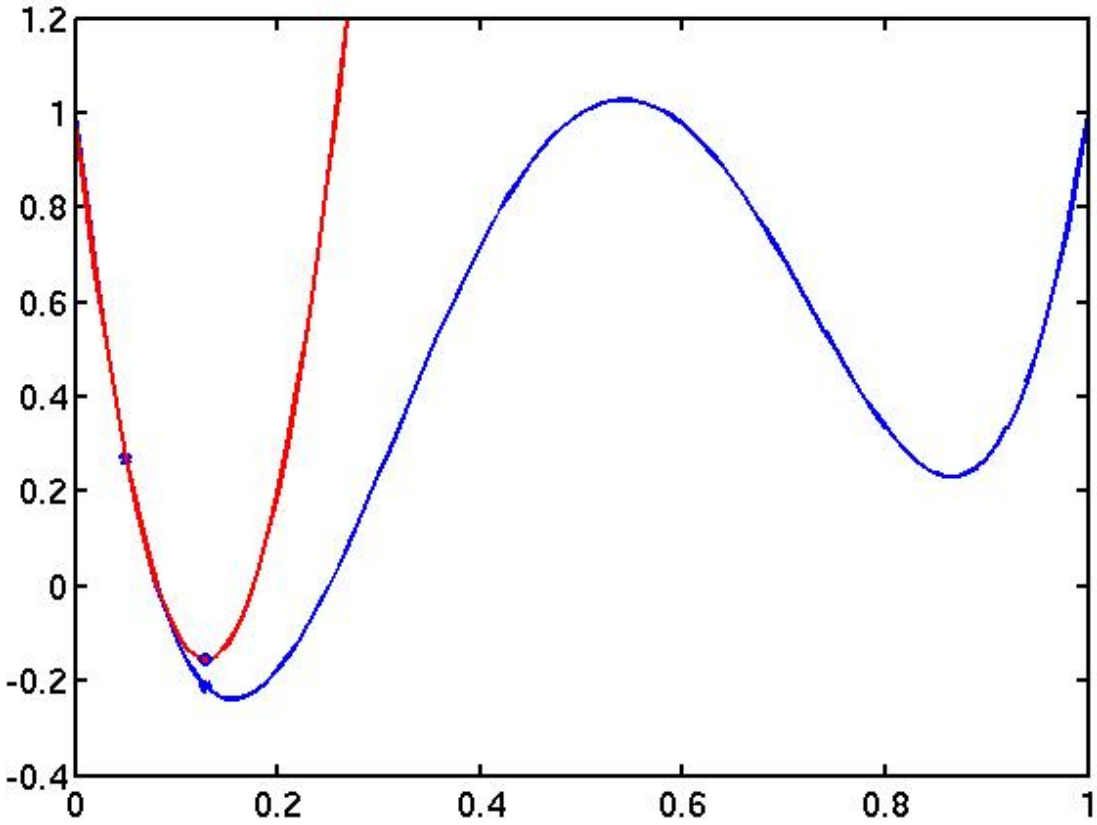
- Complicated non linear expression.
- No closed form solution.

Minimizing Reprojection Error, Locally



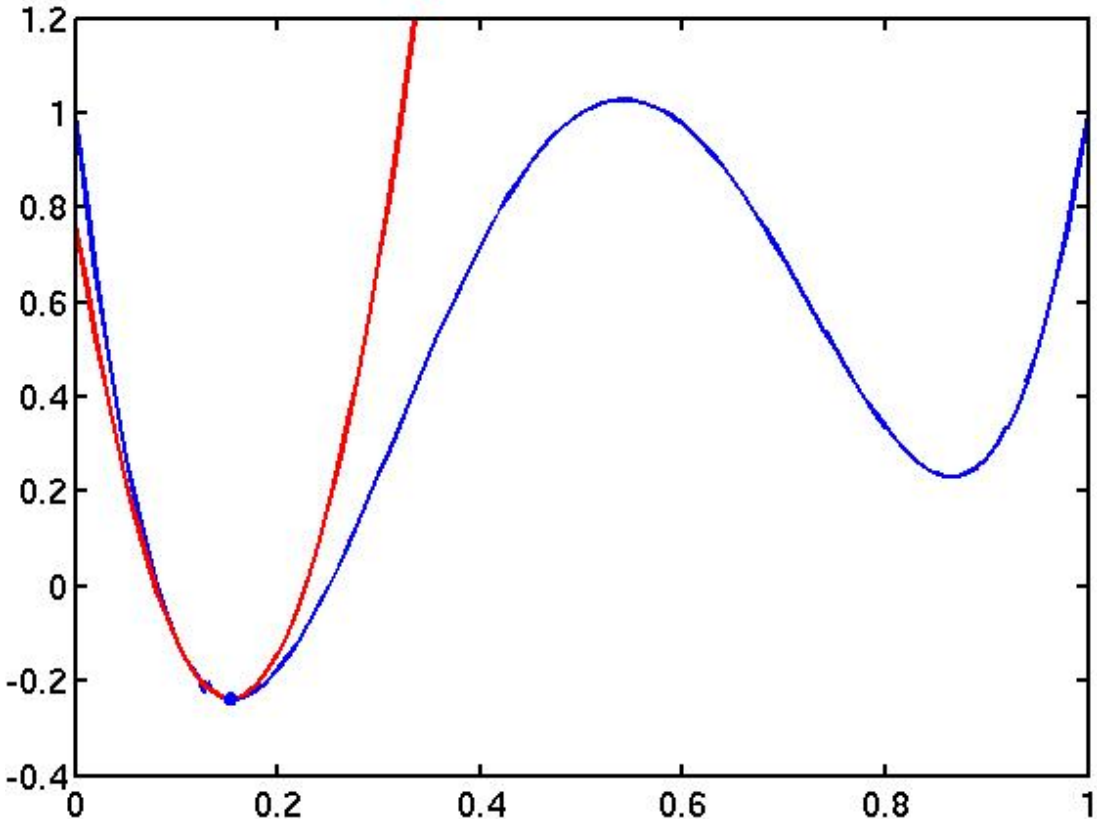
- Pick a starting point.

Minimizing Reprojection Error, Locally



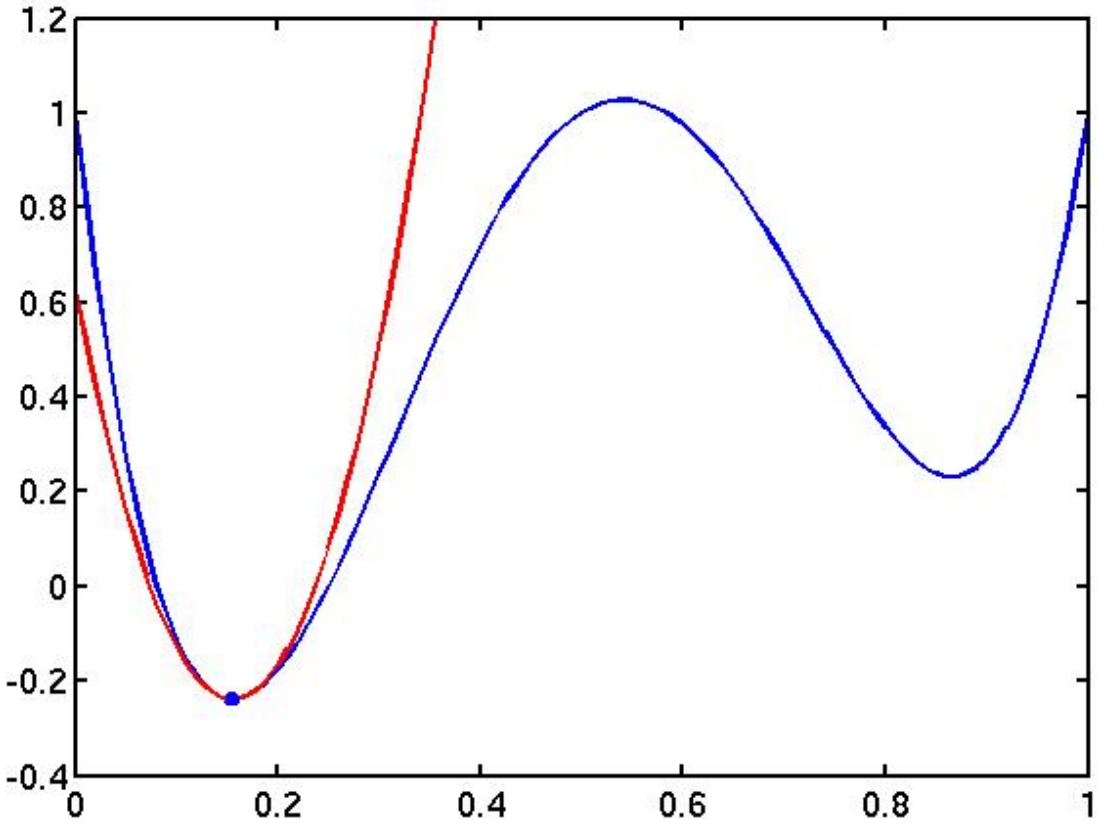
- Approximate the function using 2nd order Taylor expansion and minimize.

Minimizing Reprojection Error, Locally



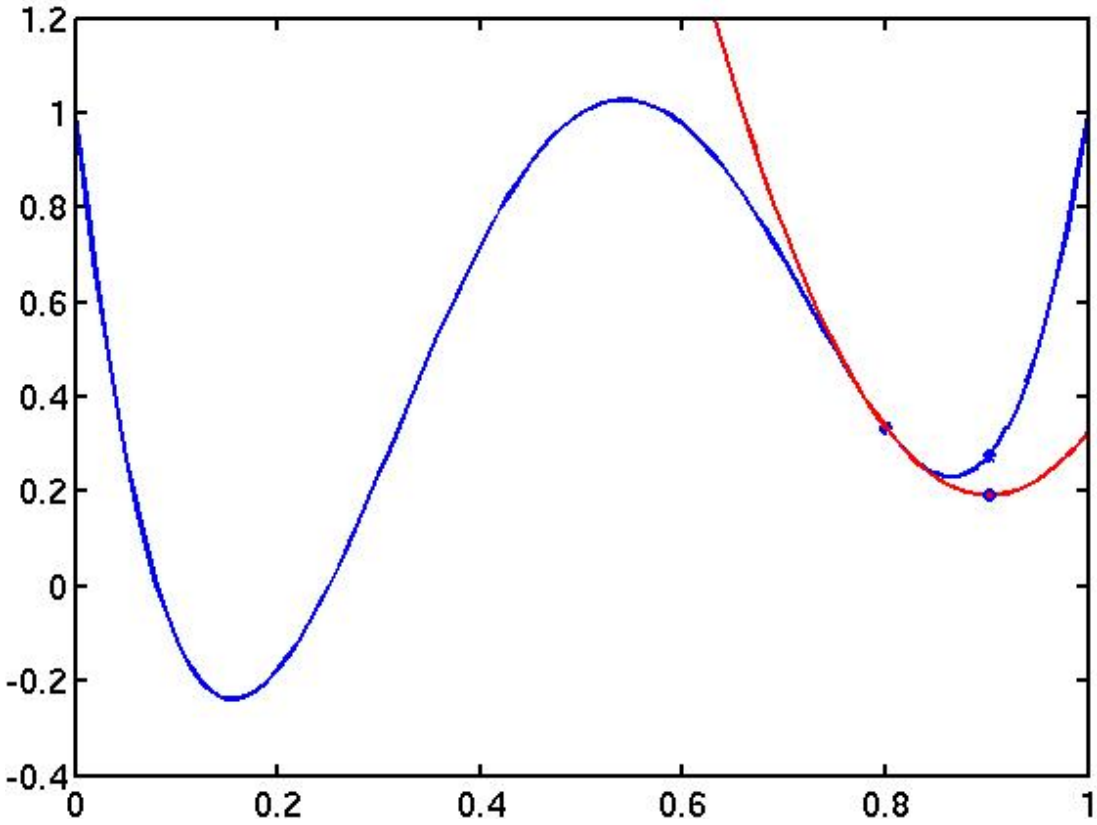
- Repeat.

Minimizing Reprojection Error, Locally



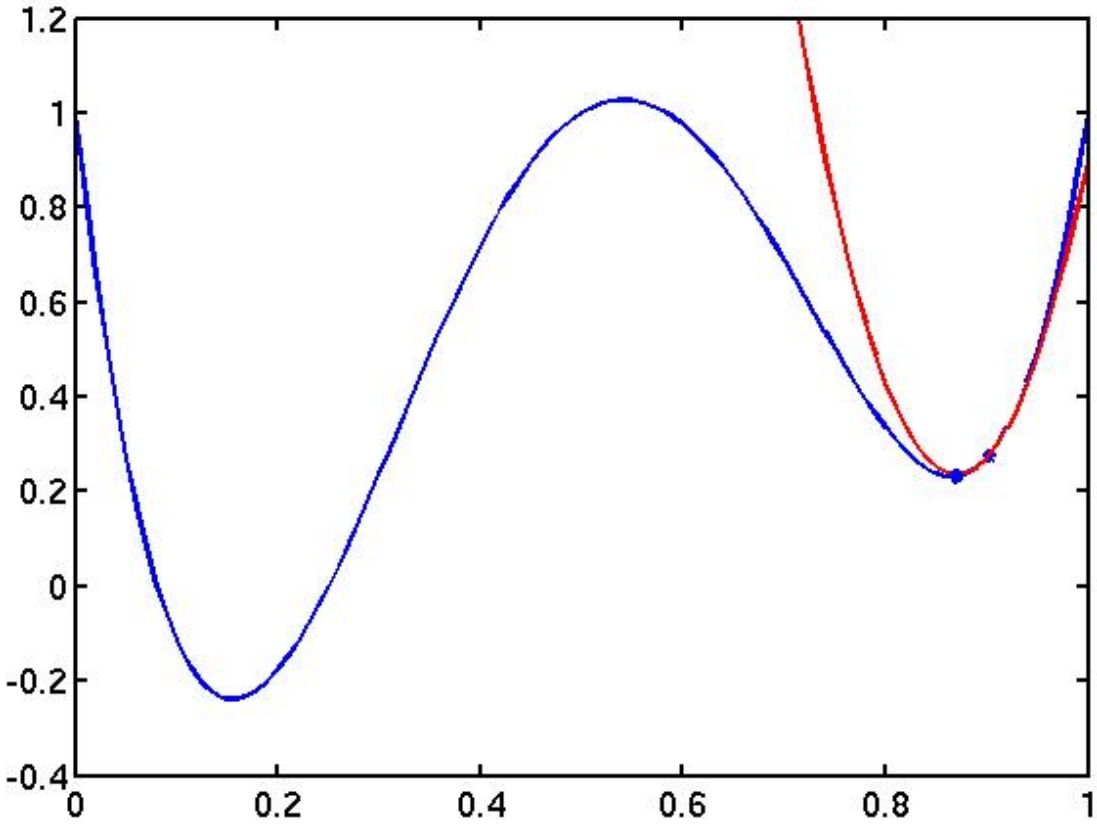
Newtons method.

Minimizing Reprojection Error, Locally



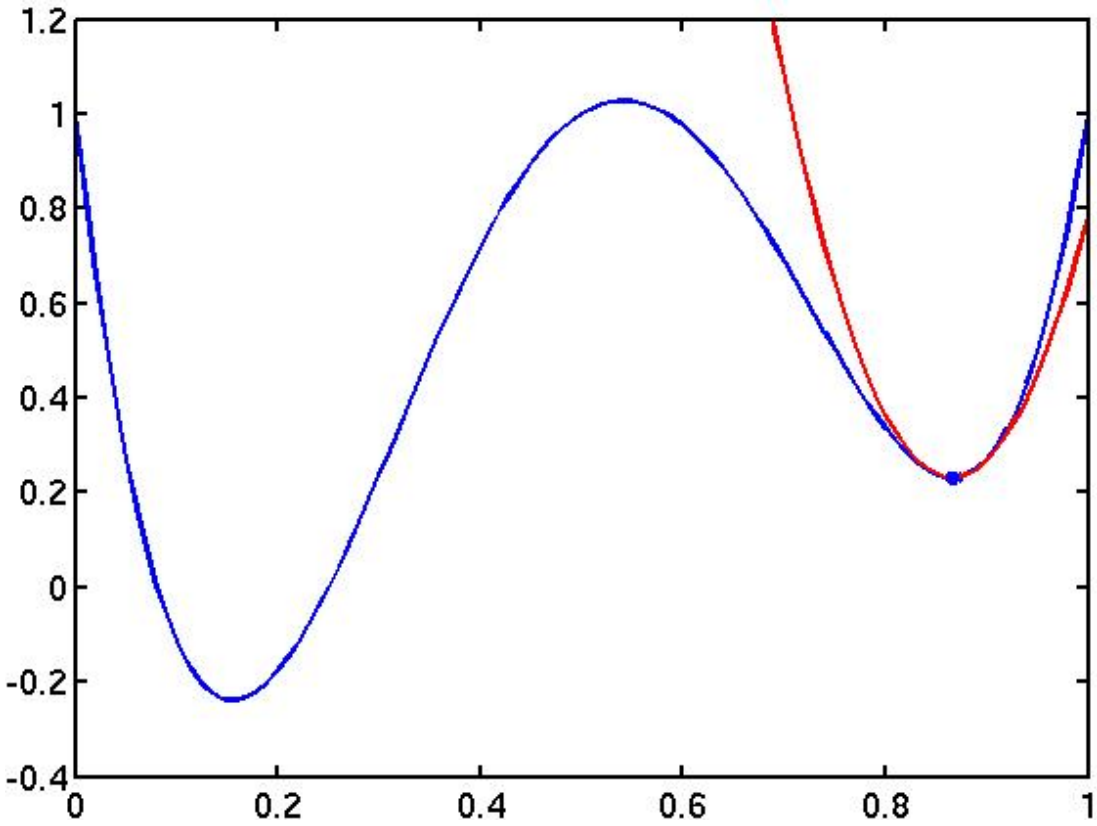
Different starting point.

Minimizing Reprojection Error, Locally



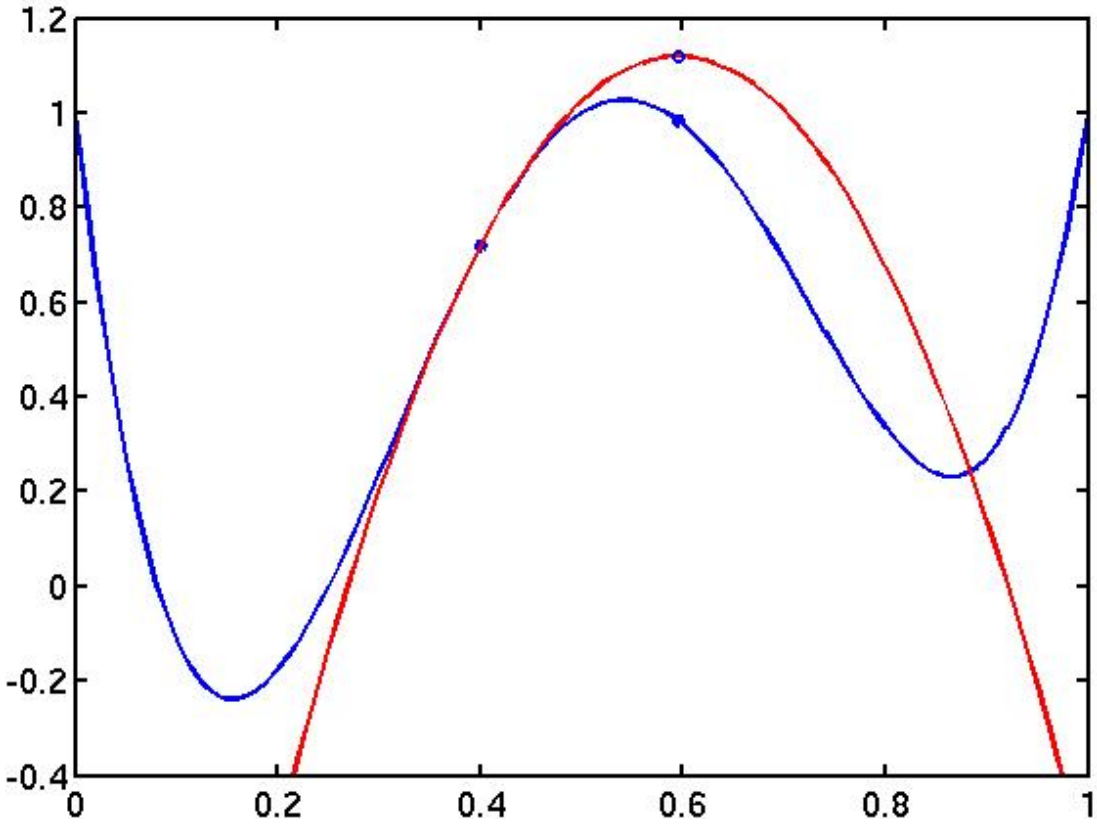
Different starting point.

Minimizing Reprojection Error, Locally



Leads to local minimum.

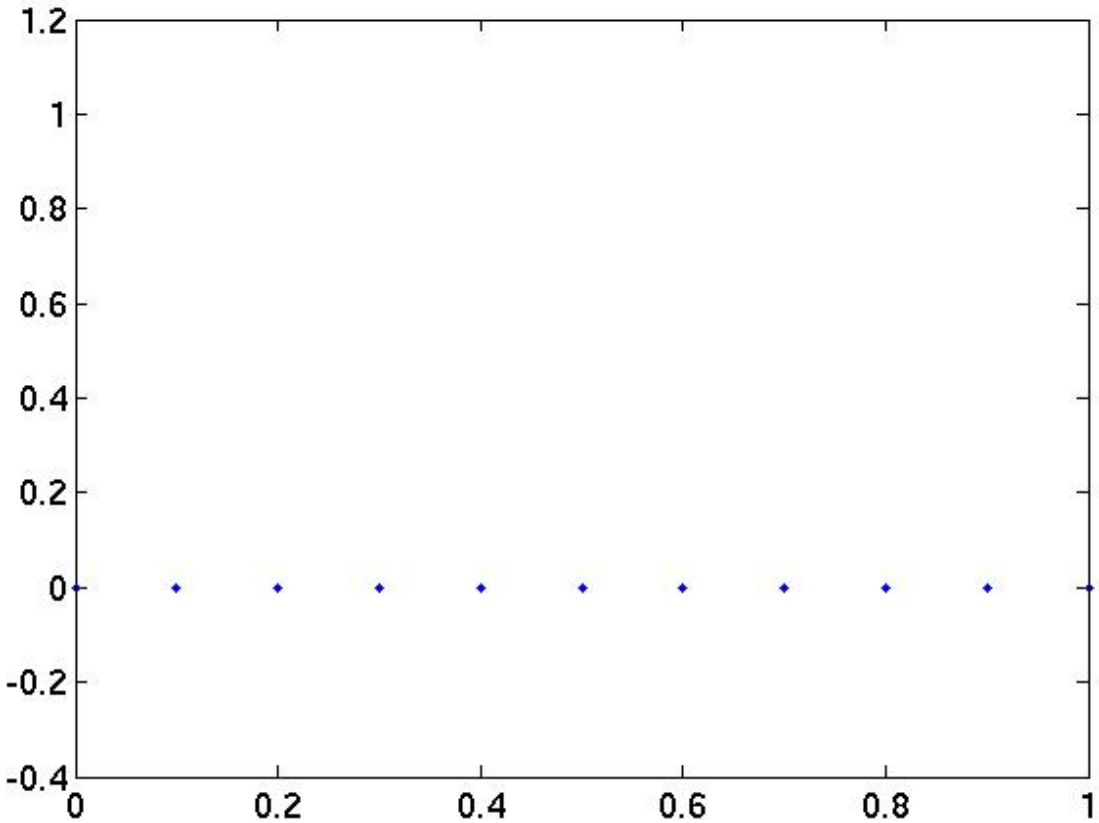
Minimizing Reprojection Error, Locally



Third starting point, leads to local maximum.

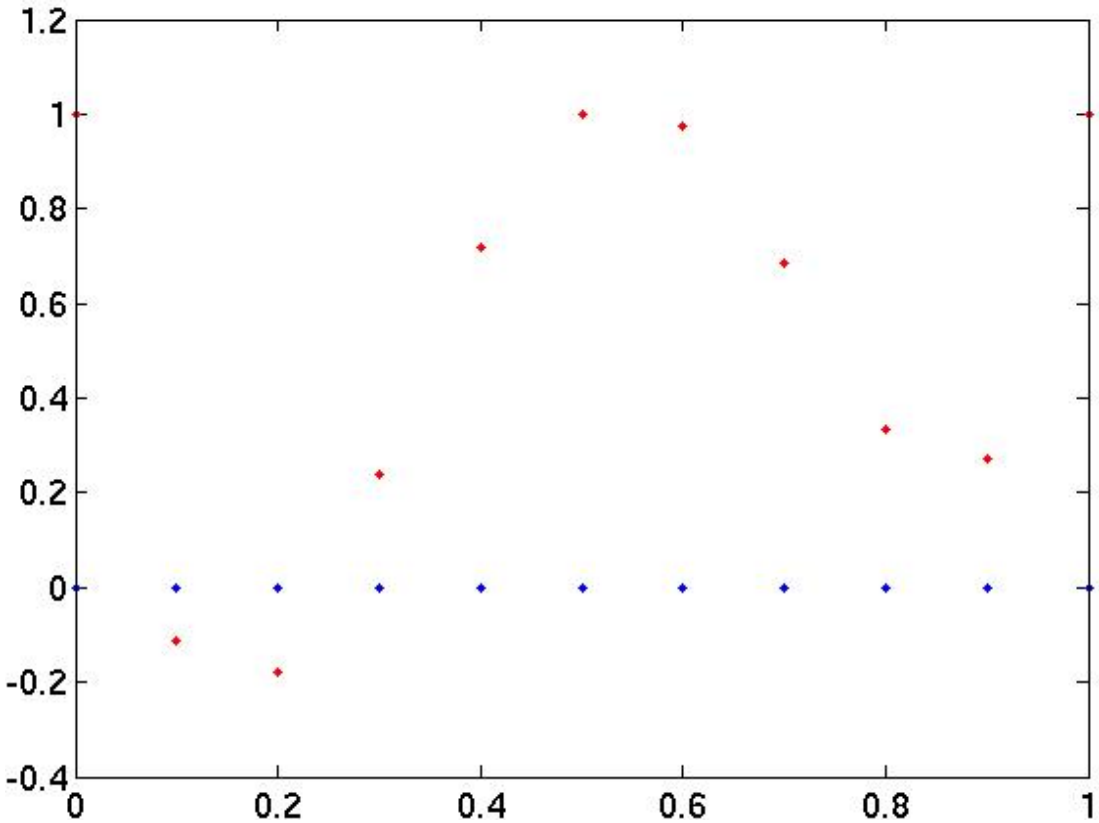
Minimizing Reprojection Error, Locally

Why not just sample the function?
One dimensional function:



Minimizing Reprojection Error, Locally

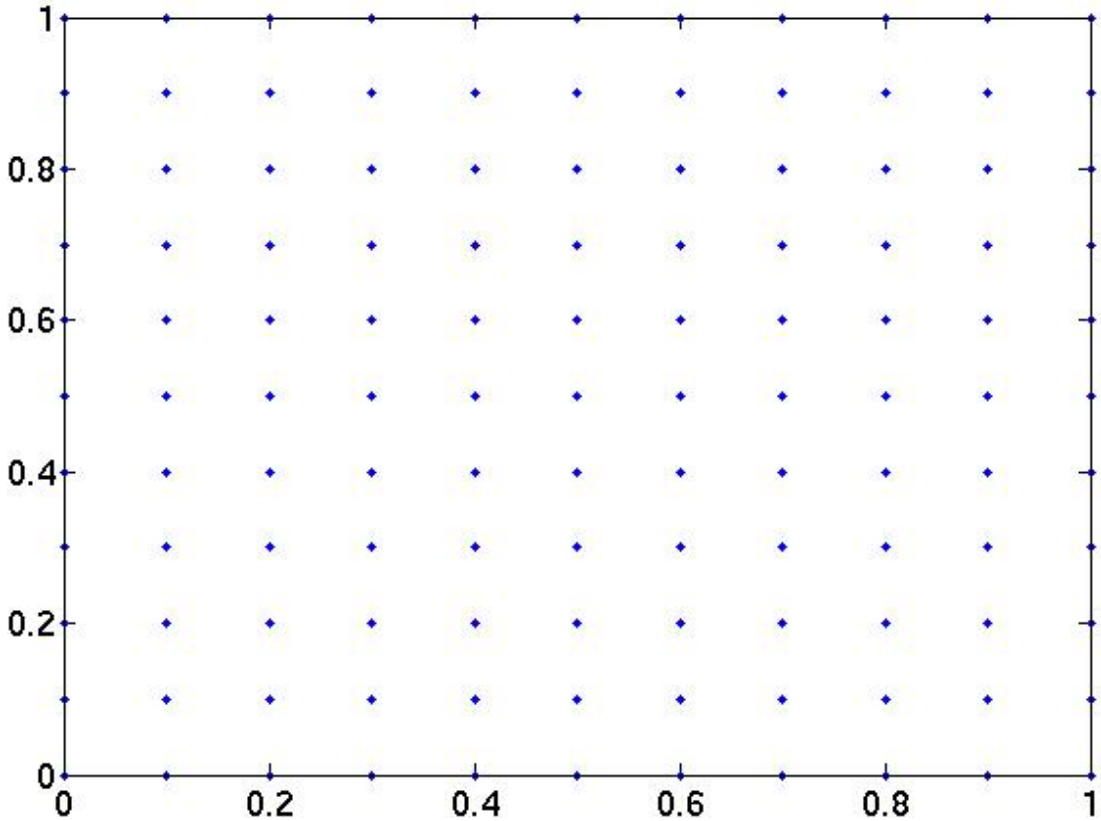
Why not just sample the function?
One dimensional function:



Sample 10 points, pick lowest value. Probably works.

Minimizing Reprojection Error, Locally

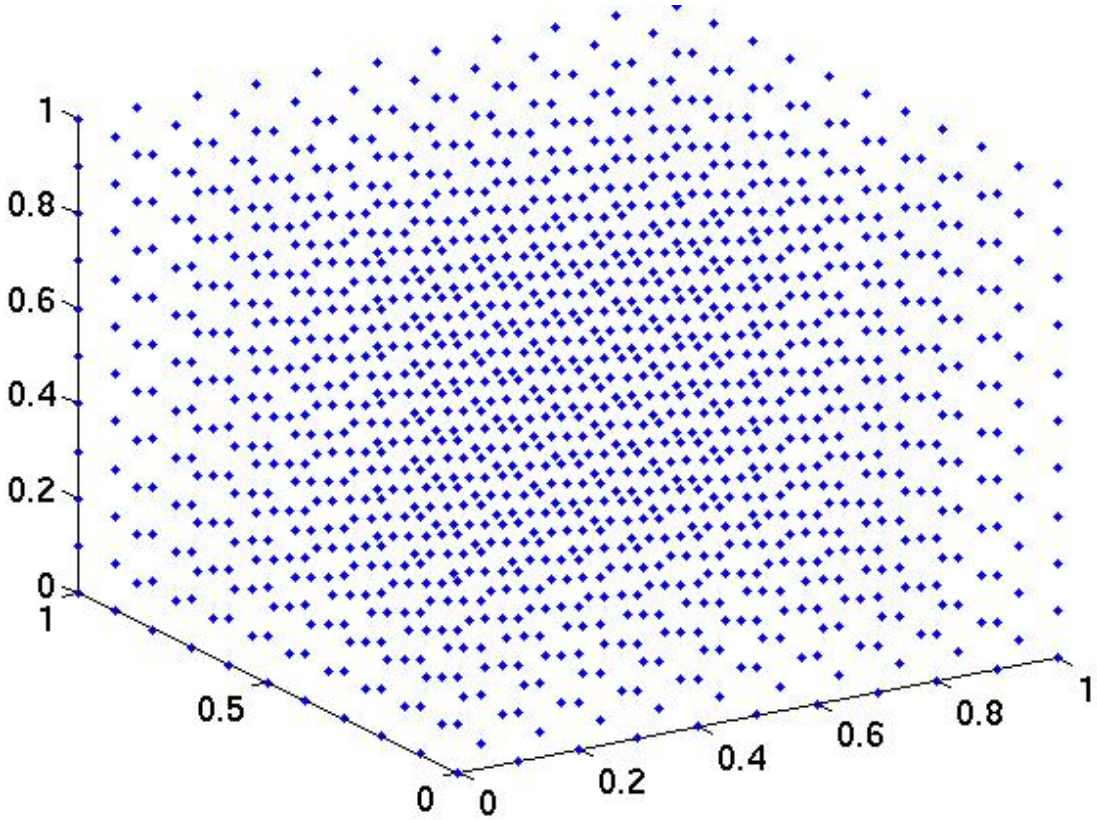
Why not just sample the function?
Two dimensional function:



10^2 samples.

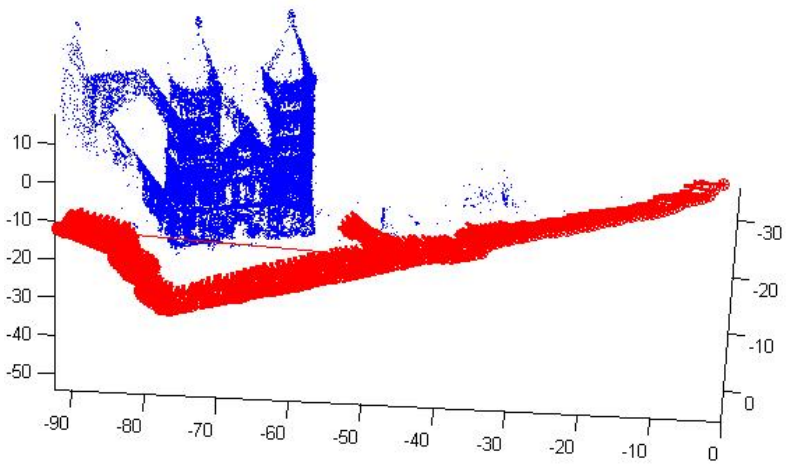
Minimizing Reprojection Error, Locally

Why not just sample the function?
Three dimensional function:



10^3 samples.

Minimizing Reprojection Error, Locally



How many variables do we have?

The cathedral dataset:

- 480 camera matrices $[R_i \quad t_i]$.
  Rotation part 3 dof, translation part 3 dof.
  Totally: $480(3+3) = 2880$.
- 91178 3D points.
  3 dof each.
  Totally: $91178 \cdot 3 = 273534$

Local Optimization (Bundle Adjustment)

$$\min_{\mathbf{v}} \sum_i r_i(\mathbf{v})^2 = \min_{\mathbf{v}} \|\mathbf{r}(\mathbf{v})\|^2$$

Locally around $\mathbf{v}^k$

$$\mathbf{r}(\mathbf{v}) \approx \mathbf{r}(\mathbf{v}^k) + J(\mathbf{v} - \mathbf{v}^k), \quad J = (\mathit{Jacobian})$$

Gauss-Newton:

$$\mathbf{v}^{k+1} = \arg\min_{\mathbf{v}} \|\mathbf{r}(\mathbf{v}^k) + J(\mathbf{v} - \mathbf{v}^k)\|^2$$

Damped Gauss-Newton:

$$\mathbf{v}^{k+1} = \arg\min_{\mathbf{v}} \|\mathbf{r}(\mathbf{v}^k) + J(\mathbf{v} - \mathbf{v}^k)\|^2 + \lambda\|\mathbf{v} - \mathbf{v}^k\|^2$$