Computer Vision: Lecture 7

Carl Olsson

2020-02-11

Today's Lecture

Repetition

- Reconstruction Problems
- A First Reconstruction System

Model Fitting

- Line Fitting and Noise models
- Linear Least Squares
- Total Least Squares
- Outliers and Robust Estimation
- Reprojection error

Repetition: Computing the Camera Matrix

Known

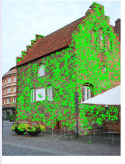
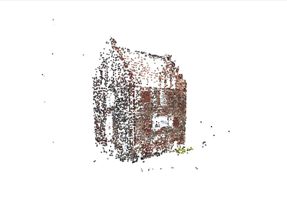
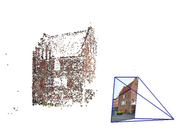


Image points $\mathbf{x}_i$.



Known scene points $\mathbf{X}_i$.

Estimate



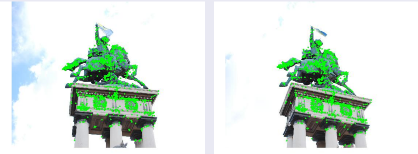
A camera matrix P such that

$$\lambda_i \mathbf{x}_i = P\mathbf{X}_i.$$

Solved using DLT in lecture 3.

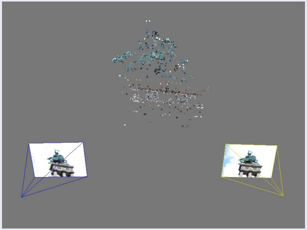
Repetition: Relative Orientation

Known:



Two corresponding point sets $\{\bar{\mathbf{x}}_i\}$ and $\{\mathbf{x}_i\}$.

Sought:



Scene points $\{\mathbf{X}_i\}$ and cameras P_1 P_2, such that

$$\lambda_i \mathbf{x}_i = P_1 \mathbf{X}_i$$
$$\bar{\lambda}_i \bar{\mathbf{x}}_i = P_2 \mathbf{X}_i$$

Repetition: Relative Orientation

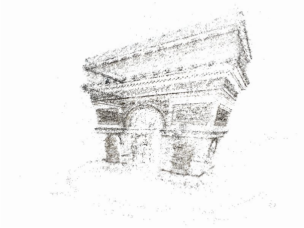
The Fundamental Matrix (see lecture 5)

For cameras $P_1 = \begin{bmatrix} I & 0 \end{bmatrix}$ and $P_2 = \begin{bmatrix} A & t \end{bmatrix}$. The corresponding image points $\mathbf{x}_i$ and $\bar{\mathbf{x}}_i$ fulfills

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

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

- The scene point $\mathbf{X}_i$ has been eliminated.
- Solve F using 8-point alg, compute cameras (lect. 5).



Problem: Projective ambiguity

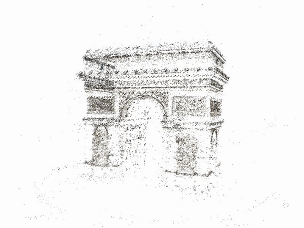
The Essential Matrix (see lecture 5)

For cameras $P_1 = \begin{bmatrix} I & 0 \end{bmatrix}$ and $P_2 = \begin{bmatrix} R & t \end{bmatrix}$. The corresponding image points $\mathbf{x}_i$ and $\bar{\mathbf{x}}_i$ fulfills

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

where, $F = [t]_\times R$.

- The scene point $\mathbf{X}_i$ has been eliminated.
- Solve E using modified 8-point alg, compute cameras (lect. 6).



No projective ambiguity

Repetition: Triangulation

Known

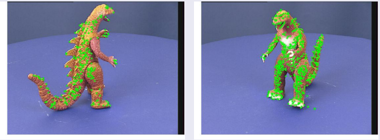
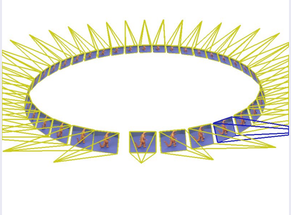


Image points $\{\mathbf{x}_{ij}\}$.



Camera matrices P_j

Sought

3D points $\mathbf{X}_i$, such that

$$\lambda_{ij}\mathbf{x}_{ij} = P_j\mathbf{X}_i$$

See lecture 4.

A First Reconstruction System

Sequential Reconstruction

Given lots of images

...

How do we compute the entire reconstruction?

1. For an initial pair of images, compute the cameras and visible scene points, using 8-point alg.
2. For a new image viewing some of the previously reconstructed scene points, find the camera matrix, using DLT.
3. Compute new scene points using triangulation.
4. If there are more cameras goto step 2.

A First Reconstruction System

Demonstrations...

A First Reconstruction System

Issues

- Outliers.
- Noise sensitivity.
- How to select initial pair.
- Unreliable 3D points.

Will get back to theses issues later in the course.

Model Fitting

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

Examples:

Camera Estimation Given scene points $\mathbf{X}_i$ find P such that $P\mathbf{X}_i$ 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.

Model Fitting

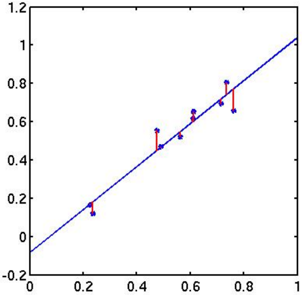
See lecture notes.

Least Squares Line Fitting

$$\min \sum_i (ax_i + b - y_i)^2$$

In matrix form

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

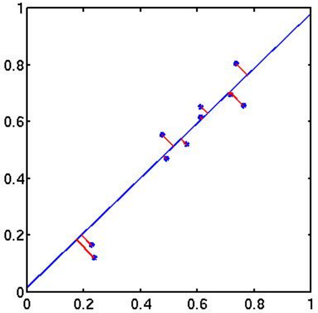


In matlab use

$$A \backslash B$$

Total Least Squares Line Fitting

$$\begin{array}{ll} \min & \sum_i (ax_i + by_i + c)^2 \\ \text{s.t} & a^2 + b^2 = 1 \end{array}$$



Let

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

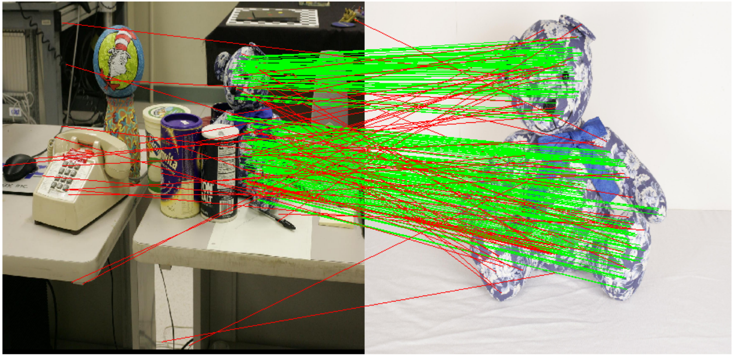
The the optimum fulfills

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

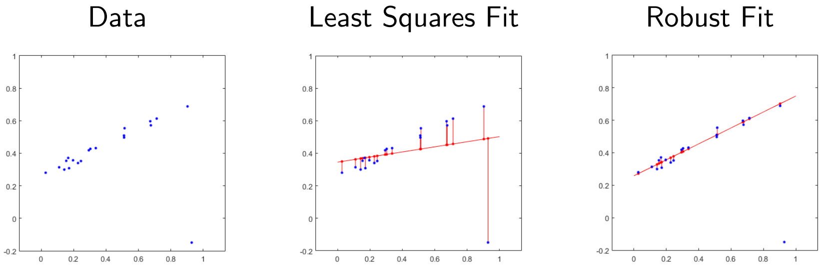
and

$$c = -(a\bar{x} + b\bar{y})$$

Outliers and Robust Loss Functions



Outliers and Robust Loss Functions

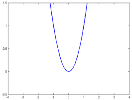


We want to remove measurements that do not obey the Gaussian-noise model before doing least squares fitting.

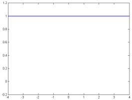
Outliers and Robust Loss Functions

See lecture notes.

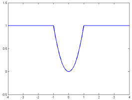
$$\rho_1(\epsilon_i^2) = \epsilon_i^2$$



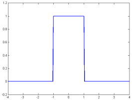
$$\rho_1'(\epsilon_i^2) = 1$$



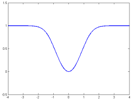
$$\rho_2(\epsilon_i^2) = \min(\epsilon_i^2, \tau^2)$$



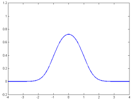
$$\rho_2'(\epsilon_i^2) = \begin{cases} 0 & |\epsilon_i| > \tau \\ 1 & |\epsilon_i| < \tau \end{cases}$$



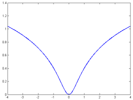
$$\rho_3(\epsilon_i^2) = \frac{\ln(1+\tau^2) - \ln(e^{-\epsilon_i^2} + \tau^2)}{\ln(1+\tau^2)}$$



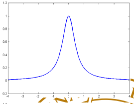
$$\rho_3'(\epsilon_i^2) = \frac{e^{-\epsilon_i^2}}{(e^{-\epsilon_i^2} + \tau^2)\ln(1+\tau^2)}$$



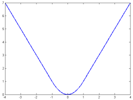
$$\rho_4(\epsilon_i^2) = b^2 \ln(1 + \frac{\epsilon_i^2}{b^2})$$



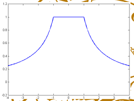
$$\rho_4'(\epsilon_i^2) = \frac{b^2}{b^2 + \epsilon_i^2}$$



$$\rho_5(\epsilon_i^2) = \begin{cases} 2b|\epsilon_i| - b^2 & |\epsilon_i| \geq b \\ \epsilon_i^2 & |\epsilon_i| \leq b \end{cases}$$



$$\rho_5'(\epsilon_i^2) = \begin{cases} \frac{b}{|\epsilon_i|} & |\epsilon_i| \geq b \\ 1 & |\epsilon_i| \leq b \end{cases}$$



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.

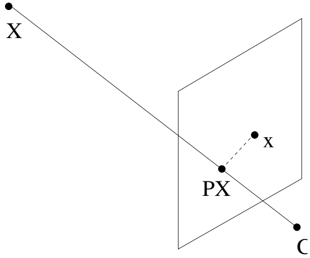
The reprojection error

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

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



Handling Noise - Minimizing Reprojection Error

Demonstration ...

Reprojection Error vs. Algebraic Error

Algebraic Error

Attempts to find an approximate solution to an algebraic equation.
Ex. DLT

$$\min \sum_i ||\lambda_i \mathbf{x}_i - P\mathbf{X}_i||^2,$$

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 solution (next lecture).

Optimal 2-view Triangulation

$\mathbf{x}$,$\bar{\mathbf{x}}$ measured projections in P and $\bar{P}$

$$\min_{\mathbf{X}} \underbrace{\left\| \left(x^1 - \frac{P^1\mathbf{X}}{P^3\mathbf{X}},\ x^2 - \frac{P^2\mathbf{X}_j}{P^3\mathbf{X}} \right) \right\|^2}_{:=d(\mathbf{x},P\mathbf{X})^2} + \underbrace{\left\| \left(\bar{x}^1 - \frac{\bar{P}^1\mathbf{X}}{\bar{P}^3\mathbf{X}},\ \bar{x}^2 - \frac{\bar{P}^2\mathbf{X}}{\bar{P}^3\mathbf{X}} \right) \right\|^2}_{:=d(\bar{\mathbf{x}},\bar{P}\mathbf{X})^2}.$$

Optimal 2-view Triangulation

Lecture 5: There is a 3D-point $\mathbf{X}$ that projects to $\mathbf{y}$ and $\bar{\mathbf{y}}$ in P and $\bar{P}$ respectively if and only if

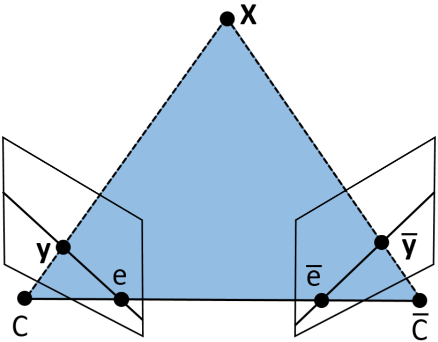
$$\bar{\mathbf{y}}^T F \mathbf{y} = 0.$$

Equivalent formulation:

$$\begin{array}{cl} \min & d(\mathbf{x}, \mathbf{y})^2 + d(\bar{\mathbf{x}}, \bar{\mathbf{y}})^2 \\ \text{such that} & \bar{\mathbf{y}}^T F \mathbf{y} = 0 \end{array}$$

Optimal 2-view Triangulation

Corresponding epipolar lines:
There is a one-to-one correspondence between epipolar lines of image 1 and 2.



There is a one-parameter family of corresponding epipolar lines.

Ex. 1

Compute all epipolar line correspondences for the cameras

$$P = \begin{bmatrix} I & 0 \end{bmatrix} \text{ and } \bar{P} = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}. \tag{1}$$

Optimal 2-view Triangulation

Equivalent formulation:

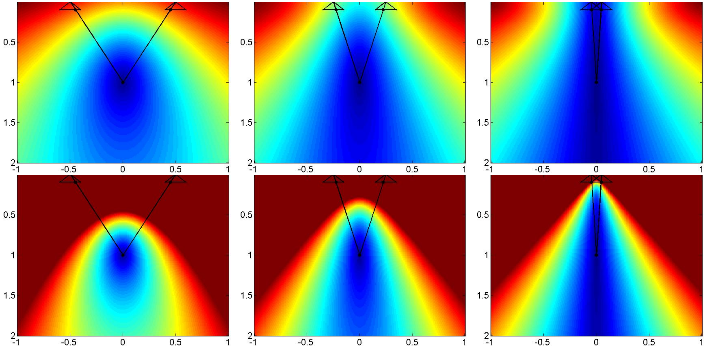
$$\begin{array}{ll} \min & d(\mathbf{x},\mathbf{l})^2 + d(\bar{\mathbf{x}},\bar{\mathbf{l}})^2 \\ \text{such that} & \mathbf{l} \text{ corresponds to } \bar{\mathbf{l}} \end{array}$$

$d(\mathbf{x},\mathbf{l}) =$ orthogonal distance between $\mathbf{x}$ and $\mathbf{l}$.

Ex. 2

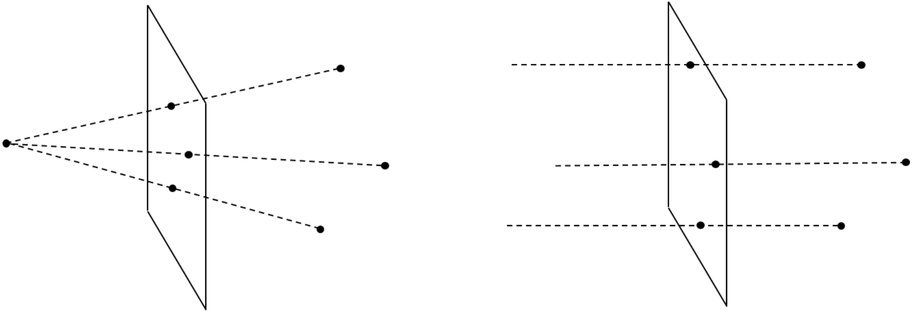
Compute the corresponding epipolar lines that minimize $d(\mathbf{x}, \mathbf{l})^2 + d(\bar{\mathbf{x}}, \bar{\mathbf{l}})^2$ for the cameras in Ex. 1.

Optimal Triangulation vs. DLT



Comparison between the DLT- and the ML objectives for triangulation. First row DLT, second row ML objective (with the same colormap).

Special Case: Affine Cameras



$$P = \begin{bmatrix} A_{2\times3} & t_{2\times1} \\ 0 & 1 \end{bmatrix}$$

Special Case: Affine Cameras

Solving Structure and Motion via Factorization

Suppose x_{ij} is the projection of X_j in image i. The maximum likelihood solution is obtained by minimizing

$$\sum_{ij} ||x_{ij} - (A_i X_j + t_i)||^2$$

The optimal t_i is given by

$$t_i = \bar{x}_i - A_i \bar{X},$$

where $\bar{X} = \frac{1}{m} \sum_j X_j$ and $\bar{x}_i = \frac{1}{m} \sum_j x_{ij}$.

Special Case: Affine Cameras

Solving Structure and Motion via Factorization

Changing coordinates, $\tilde{x}_{ij} = x_{ij} - \bar{x}_i$ and $\tilde{X}_i = X_i - \bar{X}$, gives

$$\sum_{ij} ||\tilde{x}_{ij} - A_i \tilde{X}_j||^2.$$

In matrix form

$$|| \underbrace{\begin{bmatrix} \tilde{x}_{11} & \tilde{x}_{12} & \dots & \tilde{x}_{1m} \\ \tilde{x}_{21} & \tilde{x}_{22} & \dots & \tilde{x}_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{x}_{n1} & \tilde{x}_{n2} & \dots & \tilde{x}_{nm} \end{bmatrix}}_{M} - \underbrace{\begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_n \end{bmatrix} \begin{bmatrix} \tilde{X}_1 & \tilde{X}_2 & \dots & \tilde{X}_m \end{bmatrix}}_{\text{rank 3 matrix}} ||^2$$

Special Case: Affine Cameras

Algorithm

- Re center all images such that the center of mass of the points is zero.
- Form the measurement matrix M.
- Compute the svd:
$$[U, S, V] = svd(M);$$
- A solution is given by the cameras in $U(:, 1:3)$ and the structure in $S(1:3, 1:3) * V(:, 1:3)'$.
- Transform back to the original image coordinates.

Factorization

- Requires all points to be visible in all images.
- Could work for perspective cameras if all points have roughly the same distance to the cameras.

Special Case: Affine Cameras

Demonstration...