Computer Vision (CS763)

Teaching cameras to "see"

Robust Methods in Computer Vision

Arjun Jain

Most slides courtesy Ajit Rajwade

https://www.cse.iitb.ac.in/~ajitvr/CS763_Spring2017/ImageAlignment.pdf

Dealing with Outliers

1. Heavier Tailed Distributions

2. **RANSAC**

3. LMeds

RANSAC: Random Sample Consensus

1. Arbitrarily choose *k* out of N points where *k* is the smallest number of points required to determine **a**. Call this set C.

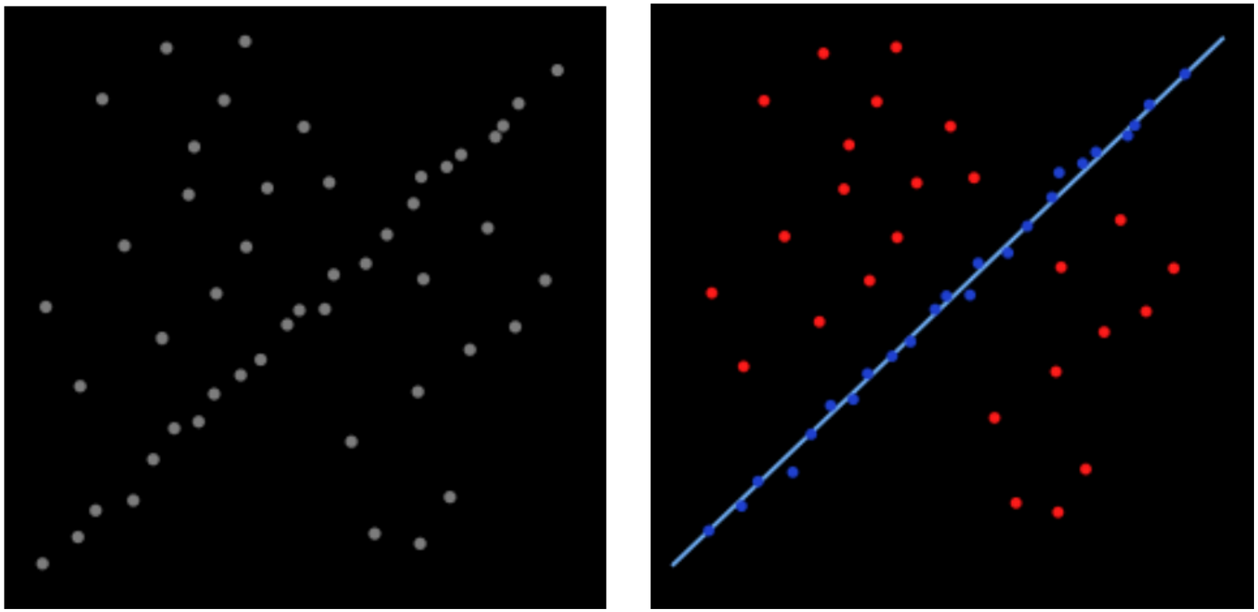
2. Determine **a** using an inverse (say) from C.

3. Determine and remember the squared residual errors for all the other N-k points, i.e. compute
$\{e_i = (y_i - f(x_i; \boldsymbol{a}))^2\}_{i \neq C}$

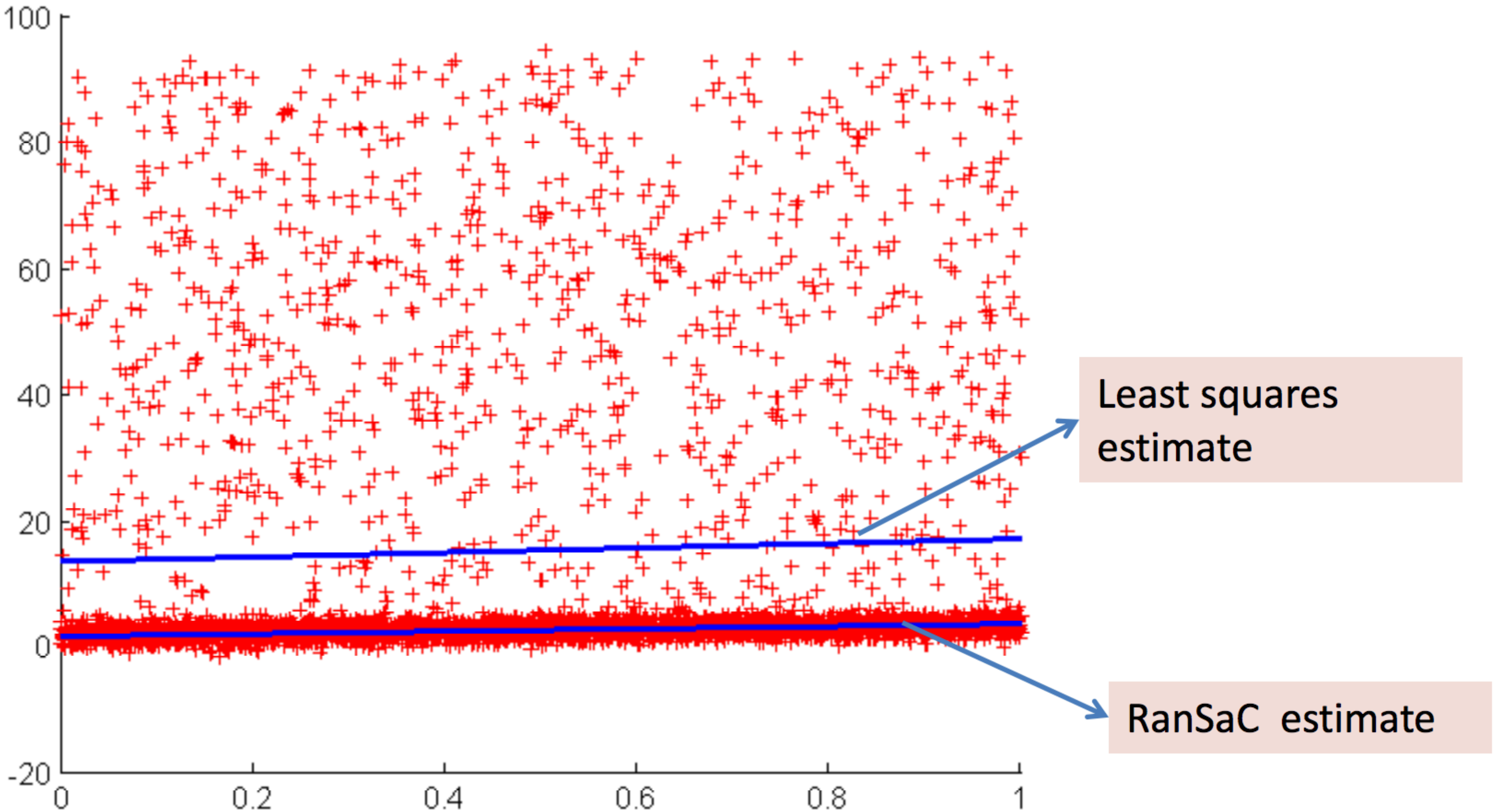
RANSAC: Random Sample Consensus

4. Count the number of points for which $e_i < \lambda$ where λ is some threshold. These points form the "consensus set" for the chosen model.

Sample result with RanSaC for line fitting.



RANSAC: Random Sample



RANSAC: Complexity Analysis

- What's the time complexity of this algorithm?

- **S** = number of subsets to be sampled. What should be the minimum value of **S**?

- Let's say that some fraction p ($0 < p < 1$) of the N points are inliers ("good points"), and somehow we know this p

RANSAC: Complexity Analysis

- The probability that at all points in a chosen set of k points are inliers $= p^k$

- The probability that at least one point in a chosen set of k points is an outlier $= 1 - p^k$

- Then the probability that at least one of the **S** different subsets contains all inliers i.e. yields good estimate of **a** is:
  P $= 1 - \left(1 - p^k\right)^S$

RANSAC: Complexity Analysis

- (copied from last slide): Then the probability that at least one of the **S** different subsets contains all inliers i.e. yields good estimate of **a** is:
  P $= 1 - \left(1 - p^k\right)^S$

- Fix **P** to 0.99 (say) and compute **S** assuming you know p.

- Clearly **S** will increase hugely if either **k** is large (more parameters to determine) and/or if p is small (fewer inliers).

RANSAC: Expected number of iterations

- The probability that at all points in a chosen set of k points are inliers $= p^k$

- The probability that at least one point in a chosen set of k points is an outlier $= 1 - p^k$

RANSAC: Expected number of iterations

- The probability that the i-th set is the **first** set that contains no outliers = $(1-p^k)^{i-1}p^k$ to be denoted as $Q(i) = (1-p^k)^{i-1}p^k$

- The expected number of sets to be drawn required to find the first no-outlier set = $\sum_i i\, Q(i)$

RANSAC: Expected number of iterations

- $Q(i) = \left(1 - p^k\right)^{i-1} p^k$
- And $\sum_i i\, Q(i)$

$$= \sum_i i\,(1 - p^k)^{i-1} p^k$$

$$= p^k \sum_i i\,(1 - p^k)^{i-1}$$

$$= p^k \frac{d}{d(1 - p^k)} \left(\sum_i (1 - p^k)^i \right)$$

RANSAC: Expected number of iterations

- $\sum_i i\, Q(i)$

$$= p^k \frac{d}{d(1-p^k)}\left(\sum_i (1-p^k)^i\right)$$

Let $1-p^k = r$

$$= p^k \frac{d}{dr}\left(\sum_i r^i\right) = p^k \frac{d}{dr}\left(\frac{r}{1-r}\right)$$

$$= p^k \left(\frac{1}{1-r} + \frac{r}{(1-r)^2}\right)$$

$$= p^k \frac{1}{(1-r)^2} = p^k \frac{1}{(1-(1-p^k))^2} = p^{-k}$$

RANSAC Variant: MSAC

- RANSAC gives us the subset **C** with largest number of inliers (i.e. least number of outliers), which is equivalent to picking the subset that minimizes the following cost $J(C)$

$$J(C) = \sum_{i \notin C} \rho(e_i) \qquad \rho(e_i) = \begin{cases} 1 \; if \; {e_i}^2 \geq T(= \lambda^2) \\ 0 \; if \, {e_i}^2 < T \end{cases}$$

RANSAC Variant: MSAC

- One could instead minimize a cost function that gives weights to inliers to see how well they fit the model:

$$\hat{J}(C) = \sum_{i \notin C} \hat{\rho}\,(e_i) \quad \rho(e_i) = \begin{cases} T & if\ e_i{}^2 \geq T \\ e_i{}^2 & if\, e_i{}^2 < T \end{cases}$$

M-estimator: an estimator that weighs inliers by their "quality", and outliers by a fixed constant

MSAC (M-estimator Sample Consensus)

Reminder: 2D Homography

$$\lambda \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix}$$

- Most general 2D transform (projective)
- If camera rotates about its center, then the images are related by a homography irrespective of scene depth.
- If the scene is planar, then images from any two cameras are related by a homography.
- Homography mapping is a 3x3 matrix with 8 degrees of freedom

Reminder: 2D Homography

$$\lambda \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix}$$

- So given correspondences from two images of a coplanar scene taken from two different cameras, how will you determine the planar homography matrix H?
- How many point correspondences will you require?
- How do you make use of more correspondences for a more accurate estimate of H?

Setting Up the Equations for Determining H

- For multiple points, we obtain

$$\begin{bmatrix} x_{2i} \\ y_{2i} \\ 1 \end{bmatrix} = \underset{3\times 3}{\mathrm{H}} \begin{bmatrix} x_{1i} \\ y_{1i} \\ 1 \end{bmatrix} \qquad i = 1, \ldots, N$$

- Write H as a vector *h*, express the correspondences as a Ah=0 equation with the constraint that $\|h\| = 1$ as H only has 8 DOF

Setting Up the Equations for Determining H

$$\begin{pmatrix} x_{21} \\ y_{21} \\ 1 \end{pmatrix} = \begin{pmatrix} \hat{H}_{11} & \hat{H}_{12} & \hat{H}_{13} \\ \hat{H}_{21} & \hat{H}_{22} & \hat{H}_{23} \\ \hat{H}_{31} & \hat{H}_{32} & \hat{H}_{33} \end{pmatrix} \begin{pmatrix} x_{11} \\ y_{11} \\ 1 \end{pmatrix}$$

$$x_{21} = \frac{\hat{H}_{11} x_1 + \hat{H}_{12} y_1 + \hat{H}_{13}}{\hat{H}_{31} x_1 + \hat{H}_{32} y_1 + \hat{H}_{33}}$$

$$y_{21} = \frac{\hat{H}_{21} x_1 + \hat{H}_{22} y_1 + \hat{H}_{23}}{\hat{H}_{31} x_1 + \hat{H}_{32} y_1 + \hat{H}_{33}}$$

$$x_{2i}x_{1i}\hat{H}_{31} + x_{2i}y_{1i}\hat{H}_{32} + x_{2i}\hat{H}_{33} - x_{1i}\hat{H}_{11} - y_{1i}\hat{H}_{12} - \hat{H}_{13} = 0$$

$$y_{2i}x_{1i}\hat{H}_{31} + y_{2i}y_{1i}\hat{H}_{32} + y_{2i}\hat{H}_{33} - x_{1i}\hat{H}_{21} - y_{1i}\hat{H}_{22} - \hat{H}_{23} = 0$$

$$\begin{pmatrix} -x_{1i} & -y_{1i} & -1 & 0 & 0 & 0 & x_{2i}x_{1i} & x_{2i}y_{1i} & x_{2i} \\ 0 & 0 & 0 & -x_{1i} & -y_{1i} & -1 & y_{2i}x_{1i} & y_{2i}y_{1i} & y_{2i} \\ & & & & & & & & \\ & & & & & & & & \end{pmatrix} \begin{pmatrix} \hat{H}_{11} \\ \hat{H}_{12} \\ \hat{H}_{13} \\ \hat{H}_{21} \\ \hat{H}_{22} \\ \hat{H}_{23} \\ \hat{H}_{31} \\ \hat{H}_{32} \\ \hat{H}_{33} \end{pmatrix} = \mathbf{0}$$

$$\mathbf{Ah} = \mathbf{0}, \mathbf{A} \text{ has size } 2N \times 9, \mathbf{h} \text{ has size } 9 \times 1$$

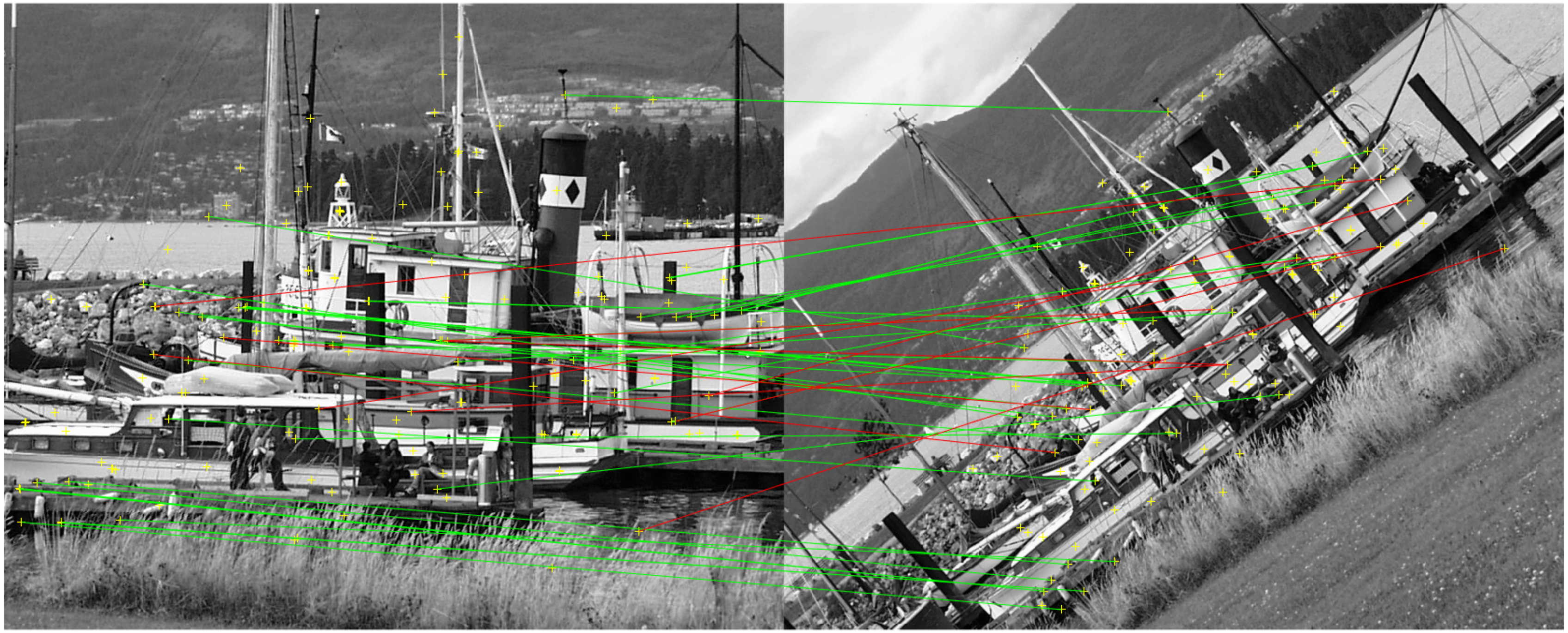
There will be N such pairs of equations (i.e. totally $2N$ equations), given N pairs of corresponding points in the two images

The equation **Ah** = **0** will be solved by computing the SVD of **A**, i.e. **A** = **USV**^{T}. The vector **h** will be given by the singular vector in corresponding to the null singular value (in the ideal case) or the null singular value.

RANSAC: Use for estimating Homography

- Determine sets **Q1** and **Q2** of salient feature points in both images, using the SIFT algorithm
- **Q1** and **Q2** may have different sizes! Determine the matching points between **Q1** and **Q2** using methods such as matching of SIFT descriptors.
- Many of these matches will be near-accurate, but there will be outliers too!

RANSAC: Use for estimating Homography



- Homography of consistent matches are marked in green while red ones are outliers.

RANSAC: Use for estimating Homography

- Pick a set of any k = 4 pairs of points and determine homography matrix H using SVD based method.
- Determine the number of inliers – i.e. those point pairs for which:

  $\|\boldsymbol{q_{1i}} - H\boldsymbol{q_{2i}}\|_2^2 \leq \varepsilon$

- Compute final H from all inliers when we have maximum inliers!

1st image

2nd image

1st image: warped using estimated **H**

H =

$$
\begin{matrix}
0.57882301155793 & 0.06780863137907 & -28.33314842189324 \\
-0.06084045669542 & 0.56283594396435 & 30.61319941910327 \\
0.00002958152711 & -0.00003144483692 & 0.58195535780312
\end{matrix}
$$

RANSAC result with 41% inliers (threshold on squared distance was 0.1) – point matching done using minor-Eigenvalue method with SSD based matching of 9 x 9 windows in a 50 x 50 neighborhood

1st image

2nd image

1st image: warped using estimated **H** and overlapped/merged with 2nd image – to show accuracy of alignment

Left: Result of warping 1st image using **H** estimated with simple least-squares on the matching points (No RANSAC).
Right: Result merged with 2nd image.
Notice that the estimation is quite poor.

Some cautions with RANSAC

- Consider a dataset with a cloud of points all close to each other (degenerate set). A model created from an outlier point and any point from a degenerate set will have a large consensus set!

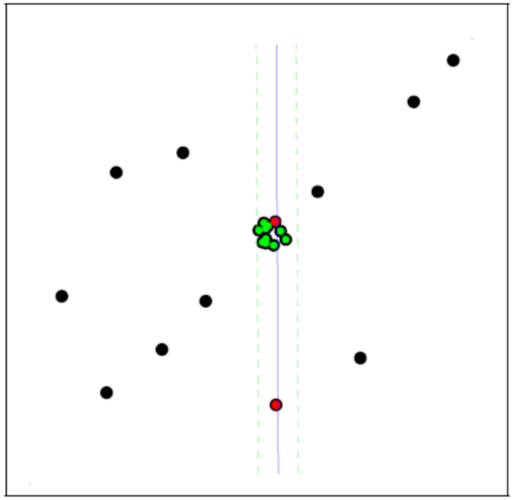


Image taken from Ph.D. thesis of Ondrej Chum, Czech Technical University, Prague

Dealing with Outliers

1. Heavier Tailed Distributions

2. RANSAC

3. **LMeds**

New ways of defining the "mean"

- We know the mean as the one that minimizes the following quantity:

$$E(\mu) = \frac{\sum_{i=1}^{n} (x_i - \mu)^2}{n} \rightarrow \mu = \frac{\sum_{i=1}^{n} x_i}{n}$$

- Changing the error to sum of absolute values, we get:

$$E(\mu) = \frac{\sum_{i=1}^{n} | x_i - \mu |}{n} \rightarrow \mu = median(\{x_i\}_{i=1}^{n})$$

Exam!

New ways of defining the "mean"

- We can also use errors of the following type with $0 < q \leq 1$:

$$E(\mu) = \frac{\left(\sum_{i=1}^{n} | x_i - \mu |^q \right)^{1/q}}{n}$$

- Optimizing the above requires iterative methods (no closed form solutions).
- The mean computed using $0 < q \leq 1$ is quite robust to outliers – with q greater than or equal to 2, the mean is susceptible to outliers.

New ways of defining the "mean"

- The earlier definitions of the mean were for scalars. They can be extended for vectors in some d > 1 dimensions as well.

$$E(\boldsymbol{\mu}) = \frac{\sum_{i=1}^{n} (\mathbf{x}_i - \boldsymbol{\mu})^2}{n} \rightarrow \boldsymbol{\mu} = \frac{\sum_{i=1}^{n} \mathbf{x}_i}{n}$$

- For other q-norms (0 <= q < 1), we have:

$$E(\boldsymbol{\mu}) = \frac{\left(\sum_{i=1}^{n} |\mathbf{x}_i - \boldsymbol{\mu}|^q \right)^{1/q}}{n}$$

LMeds: Least Median of Squares

- It works as follows:

$$J(\mathbf{a}) = \text{median}_{i=1:N} \, (y_i - f(x_i;\mathbf{a}))^2;$$

select $\mathbf{a}$ for which $J(\mathbf{a})$ is minimum.

- This has no closed form solution either and you can't do gradient descent type of techniques as the median is not differentiable.

- But it has an "algorithmic" solution.

LMeds: Algorithm

- **Step 1**: Arbitrarily choose **k** out of **N** points where **k** is the smallest number of points required to determine **a**.
- Call this set of **k** points as **C**.
- E.g.: If you had to do **line** fitting, **k** = ?
- E.g.: If you were doing **circle** fitting, **k** = ?
- E.g.: If you have to find the **affine** transformation between two point sets in 2D, you need **k** = ? correspondences

LMeds: Algorithm

- **Step 2**: Determine **a** using C.
- **Step 3**: Determine the squared residual errors for all the other N - k points, i.e. compute

$$\{e_i = (y_i - f(x_i;\mathbf{a}))^2\}_{i \notin C}$$

- **Step 4**: Compute **medC** = median of $\{e_i\}$.
- Repeat all these four steps for **S** different subsets of **k** points each.

RANSAC versus LMeds

- LMeds needs no threshold to determine what is an inlier unlike RANSAC.

- But RANSAC has one advantage. What?
  - LMeds will need at least 50% inliers (by definition of median).
  - RANSAC can tolerate a smaller percentage of inliers (i.e. larger percentage of outliers).

Slide Information

- The slides have been created by Cyrill Stachniss as part of the photogrammetry and robotics courses.
- **I tried to acknowledge all people from whom I used images or videos. In case I made a mistake or missed someone, please let me know.**
- Ajit Rajwade https://www.cse.iitb.ac.in/~ajitvr/CS763_Spring2017/ImageAlignment.pdf
- Appendix A.7 of Trucco and Verri
- Article on Robust statistics by Chuck Stewart
- Original article on RanSaC by Fischler and Bolles
- Article on RanSaC variants by Torr and Zisserman