The slides are from several sources through James Hays (Brown); Srinivasa Narasimhan (CMU); Silvio Savarese (U. of Michigan); Bill Freeman and Antonio Torralba (MIT), including their own slides.

LEAST SQUARES. RANSAC. HOUGH TRANSFORM.
Homogeneous coordinates

Conversion (see also lecture 2)

Converting to *homogeneous* coordinates

\[(x, y) \Rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}\]

homogeneous image coordinates

\[(x, y, z) \Rightarrow \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}\]

homogeneous scene coordinates

Converting *from* homogeneous coordinates

\[
\begin{bmatrix} x \\ y \\ w \end{bmatrix} \Rightarrow \left( \frac{x}{w}, \frac{y}{w} \right)
\]

to Cartesian coordinates

\[
\begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} \Rightarrow \left( \frac{x}{w}, \frac{y}{w}, \frac{z}{w} \right)
\]
Homogeneous coordinates are invariant to scaling.

\[
\begin{bmatrix}
  x \\
  y \\
  w
\end{bmatrix}
= \begin{bmatrix}
  kx \\
  ky \\
  kw
\end{bmatrix}
\Rightarrow
\begin{bmatrix}
  \frac{kx}{kw} \\
  \frac{ky}{kw}
\end{bmatrix}
= \begin{bmatrix}
  \frac{x}{w} \\
  \frac{y}{w}
\end{bmatrix}
\]

Homogeneous Coordinates  Cartesian Coordinates

Point in Cartesian coordinates is ray in homogeneous coordinates.
Lines in a 2D plane

\[ ax + by + c = 0 \]

\[ l = \begin{bmatrix} a \\ b \\ c \end{bmatrix} \]

If \( x = [x_1, x_2]^T \in \text{line} \)

nonhomogeneous (Cartesian) coordinates
what you measure

\[
\begin{bmatrix} x_1 \\ x_2 \\ 1 \end{bmatrix}^T \begin{bmatrix} a \\ b \\ c \end{bmatrix} = 0
\]

homogeneous...
The point is the cross-product of two intersecting lines.

\[ x = l \times l' \]

Proof

\[ l \times l' \perp l \implies (l \times l') \cdot l = 0 \implies x \in l \]
\[ l \times l' \perp l' \implies (l \times l') \cdot l' = 0 \implies x \in l' \]

\[ \implies x \text{ is the intersection point in homogeneous coordinates} \]

also \[ l = X_1 \times X_2 \]
Points at infinity (ideal points)

\[ x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}, \quad x_3 \neq 0 \]

\[ x_\infty = \begin{bmatrix} x_1 \\ x_2 \\ 0 \end{bmatrix} \quad \text{ideal point} \]

Intersection of two parallel lines is a point at infinity.

\[ l^T [b\ -a\ 0]^T = 0 \quad \text{line slope } -a/b \]
Lines infinity $1_\infty$

Set of ideal points lies on a line called the line at infinity. A line has a value at the third element only.

$$1_\infty = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

since

$$\begin{bmatrix} x_1 \\ x_2 \\ 0 \end{bmatrix}^T \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = 0$$

$$l_{\text{inf}} = X_{1\text{inf}} \times X_{2\text{inf}}$$
Fitting

Critical issues: noisy data; outliers; missing data etc.
Critical issues: noisy data
Critical issues: intra-class variability
Critical issues: outliers
Critical issues: missing data (occlusions)
Ordinary least squares line fitting

Data: \((x_1, y_1), \ldots, (x_n, y_n)\)

Line equation: \(y_i = mx_i + b\)

Find \((m, b)\) to minimize

\[
E = \sum_{i=1}^{n} (y_i - mx_i - b)^2
\]

\[
Y = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} \quad X = \begin{bmatrix} x_1 & 1 \\ \vdots & \vdots \\ x_n & 1 \end{bmatrix} \quad B = \begin{bmatrix} m \\ b \end{bmatrix}
\]

\[
E = \|Y - XB\|^2 = (Y - XB)^T (Y - XB) = Y^T Y - 2(XB)^T Y + (XB)^T (XB)
\]

\[
\frac{dE}{dB} = 2X^T XB - 2X^T Y = 0
\]

\[
X^T XB = X^T Y \quad \text{Normal equations: least squares solution to} \quad XB = Y
\]
Ordinary least squares method
fitting a line

\[ E = \sum_{i=1}^{n} (y_i - mx_i - b)^2 \]

\[ B = \left( X^T X \right)^{-1} X^T Y \]

Limitations since the noise in x is neglected.

Ex. *Fails completely for ideal vertical lines.* Say, x=1. The X is singular and B is not completely defined. Gives a point \((1,y)\).
Total least squares

Distance between point \((x_i, y_i)\) and line \(ax + by = d (a^2 + b^2 = 1)\): \(|ax_i + by_i - d|\)

Will be '-d' instead of 'd'.

Both \(x\) and \(y\) corrupted with noise.
**Total least squares**

Distance between point \((x_i, y_i)\) and line \(ax + by = d\) \((a^2 + b^2 = 1)\): \(|ax_i + by_i - d|\)

Find \((a, b, d)\) to minimize the sum of squared perpendicular distances

\[
E = \sum_{i=1}^{n} (ax_i + by_i - d)^2
\]

Unit normal:

\[
(x_i, y_i)^T \quad N = (a, b)^T
\]
Total least squares

Distance between point \((x_i, y_i)\) and line \(ax + by = d\) (\(a^2 + b^2 = 1\)): \(|ax_i + by_i - d|\)

Find \((a, b, d)\) to minimize the sum of squared perpendicular distances

\[
E = \sum_{i=1}^{n} (ax_i + by_i - d)^2
\]

\[
\frac{\partial E}{\partial d} = \sum_{i=1}^{n} -2(ax_i + by_i - d) = 0
\]

\[
E = \sum_{i=1}^{n} (a(x_i - \bar{x}) + b(y_i - \bar{y}))^2 = \left\| \begin{bmatrix} x_1 - \bar{x} & y_1 - \bar{y} \\ \vdots & \vdots \\ x_n - \bar{x} & y_n - \bar{y} \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} \right\|^2 = (UN)^T (UN)
\]

\[
\frac{dE}{dN} = 2(U^T U)N = 0
\]

Solution to \((U^T U)N = 0\), subject to \(||N||^2 = 1\): eigenvector of \(U^T U\) associated with the smallest eigenvalue (least squares solution to homogeneous linear system \(UN = 0\))
The matrix $U^T U$ is a covariance matrix. Therefore, the eigenvectors of $U^T U$ are also the singular vectors of $V$ from the singular value decomposition, $UDV^T$ of $U(!)$. **Be aware the two $U$-s are different...** this is when I cannot modify some of the slides. See lecture 2 too.

In the case of a 2D line, the solution is $v_2$. The parameter $d$ is (the mean of $x$ and $y$) $^T \ast v_2$

The fit goes through the centroid of the data.
Total least squares

\[
U = \begin{bmatrix}
    x_1 - \bar{x} & y_1 - \bar{y} \\
    \vdots & \vdots \\
    x_n - \bar{x} & y_n - \bar{y}
\end{bmatrix}
\]

\[
U^T U = \begin{bmatrix}
    \sum_{i=1}^{n} (x_i - \bar{x})^2 & \sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y}) \\
    \sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y}) & \sum_{i=1}^{n} (y_i - \bar{y})^2
\end{bmatrix}
\]

second moment matrix
Total least squares

\[
U = \begin{bmatrix}
    x_1 - \bar{x} & y_1 - \bar{y} \\
    \vdots & \vdots \\
    x_n - \bar{x} & y_n - \bar{y}
\end{bmatrix}
\]

\[
U^T U = \begin{bmatrix}
    \sum_{i=1}^{n} (x_i - \bar{x})^2 & \sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y}) \\
    \sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y}) & \sum_{i=1}^{n} (y_i - \bar{y})^2
\end{bmatrix}
\]

second moment matrix

\[N = (a, b)\top\]

\[(x_i - \bar{x}, y_i - \bar{y})\top\]
TLS robustness to inlier noise...
...but not to outlier noise.

Squared error always takes into account all inliers and outliers. Least square is *not robust* to outliers.
Select one match, count inliers
Repeat many times.
Keep match with largest set of inliers based on a standard deviation given by the user.

Basic philosophy: the voting scheme for almost any elemental subset based estimation.

- *Elemental subset* (minimum number of points) randomly picked up for each hypothesis.

- *The standard deviation of the inlier noise has to be given before by the user.*

- Assumption 1: Outlier features will not vote consistently for any single model.

- Assumption 2: There are enough features to agree on a good model.
RANSAC

Algorithm:

1. Select random sample of minimum required size to fit model
2. Compute a putative model from sample set
3. Compute the set of inliers to this model from whole data set

Repeat 1-3 until model with the most inliers over all samples is found

Sample set = set of points in 2D

sigma is given
RANSAC for line fitting example

Inliers and outliers. Sigma has to be given at the beginning.

Source: R. Raguram
RANSAC for line fitting example

Least-squares fit

Source: R. Raguram
RANSAC for line fitting example

1. Randomly select minimal subset of points (= 2).

Source: R. Raguram
RANSAC for line fitting example

1. Randomly select minimal subset of points
2. Hypothesize a model

Source: R. Raguram
1. Randomly select minimal subset of points
2. Hypothesize a model
3. Compute error function

Source: R. Raguram
RANSAC for line fitting example

1. Randomly select minimal subset of points
2. Hypothesize a model
3. Compute error function
4. **Select points consistent with model (sigma)**

Source: R. Raguram
RANSAC for line fitting example

1. Randomly select minimal subset of points
2. Hypothesize a model
3. Compute error function
4. Select points consistent with model
5. Repeat hypothesize-and-verify loop

Source: R. Raguram
RANSAC for line fitting example

1. Randomly select minimal subset of points
2. Hypothesize a model
3. Compute error function
4. Select points consistent with model
5. Repeat *hypothesize-and-verify* loop

Source: R. Raguram
RANSAC for line fitting example

1. Randomly select minimal subset of points
2. Hypothesize a model
3. Compute error function
4. Select points consistent with model
5. Repeat hypothesis-and-verify loop

Source: R. Raguram
RANSAC for line fitting example

The best inlier structure
largest number of inliers.

1. Randomly select minimal subset of points
2. Hypothesize a model
3. Compute error function
4. Select points consistent with model
5. Repeat hypothesize-and-verify loop

Do least-square fit on all the inliers.

Source: R. Raguram
RANSAC
RANdom SAmple Consensus

...in general.

Algorithm:

1. Select random sample of minimum required size to fit model
2. Compute a putative model from sample set
3. Compute the set of inliers to this model from whole data set

Repeat 1-3 until model with the most inliers over all samples is found
Algorithm:
1. Select random sample of minimum required size to fit model
2. Compute a putative model from sample set
3. Compute the set of inliers to this model from whole data set
   Repeat 1-3 until model with the most inliers over all samples is found
Algorithm:
1. Select random sample of minimum required size to fit model
2. Compute a putative model from sample set
3. Compute the set of inliers to this model from whole data set
Repeat 1-3 until model with the most inliers over all samples is found

standard deviation of the inlier noise has to be given

\begin{align*}
|O| &= 6
\end{align*}
Algorithm:

1. Select random sample of minimum required size to fit model
2. Compute a putative model from sample set
3. Compute the set of inliers to this model from whole data set

Repeat 1-3 until model with the most inliers over all samples is found

\[ |O| = 14 \]
• RANSAC fit a homography (later lecture) mapping SIFT features from image 1 to 2.

• Majority of bad matches will be labeled as outliers.
This is a robust fit... as we also said in lecture on SIFT.
RANSAC - conclusions

Good
• Robust to outliers if there are not too many.
• The number of hypotheses $N$ is taken sufficiently large (hundreds to thousands) that RANSAC gives very similar results every time.

Bad
• Computational time grows quickly with fraction of outliers and number of parameters.
• Not good for getting multiple inlier structures.

Some applications
• Computing a homography (e.g., image stitching)
• Estimating fundamental matrix (relating two views), etc.
Hough transform


Given a set of points, find the line or curve that explains the data points best.

Applied to lines, circles and sometime ellipses.
Hough transform - for lines

Use a polar representation for the parameter space. *Each point* will add a sinusoid in the \((\theta, \rho)\) parameter space. The (two) dimensions have different thresholds. *Mistakes can give nonexisting features.*

\[
x \cos \theta + y \sin \theta = \rho
\]
In theory...
...but the effect of noise is very important.

Peak gets fuzzy and hard to locate.
Spurious peaks due to uniform noise.
A nice example. In general not so...

...because mistakes in labeling appear. Lines: 5-- and 5 |
Hough transform - conclusions

Good:

• All points are processed independently, so can cope with occlusion/outliers.
• Some robustness to noise: noise points unlikely to contribute consistently to any single bin.

Bad:

• Spurious peaks due to uniform noise.
• Trade-off noise vs. grid size. Hard to find sweet points for each threshold when multiple features are detected.

Is used in the industry for repeated processing only.