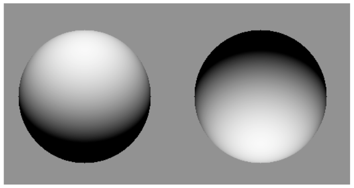
Simple example*

- What you know
  - John is coughing
- What do you conclude?
  - John has a cold
  - John has lung cancer
  - John has stomach problems

*used by Josh Tenenbaum in recent cog sci talk



Notice that the interpretation of the data is ambiguous.

The left image can be a convex with light from above, or concave with light from below.

The right image can be convex with light from below, or concave with light from above.

On average, we resolve the ambiguity by assuming that the light comes from above (prior).

Model Fitting Challenges

- Robustness
  - Squared error grows rapidly as distance increases
  - Since large distance is unlikely given Gaussian assumption, this means that either the assumption or model is likely incorrect!
- How do we know whether a point is on the line?
  - Incremental line fitting
  - K-means line fitting
  - Probabilistic with missing data

Algorithm 15.1: Incremental line fitting by walking along a curve, fitting a line to runs of pixels along the curve, and breaking the curve when the residual is too large

```
Put all points on curve list, in order along the curve
Empty the line point list
Empty the line list
Until there are too few points on the curve
  Transfer first few points on the curve to the line point list
  Fit line to line point list
  While fitted line is good enough
    Transfer the next point on the curve
      to the line point list and refit the line
  end
  Transfer last point(s) back to curve
  Refit line
  Attach line to line list
end
```

For completeness.
Not covered in 2006

Algorithm 15.2: K-means line fitting by allocating points to the closest line and then refitting.

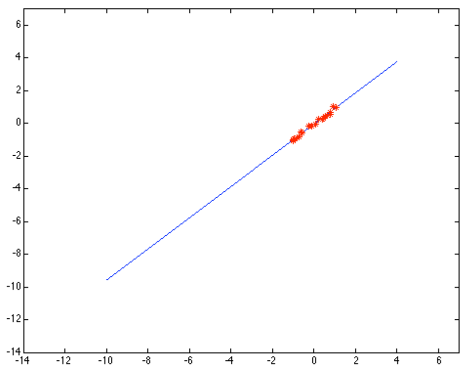
Hypothesize k lines (perhaps uniformly at random)
or
Hypothesize an assignment of lines to points
  and then fit lines using this assignment

Until convergence
  Allocate each point to the closest line
  Refit lines
end

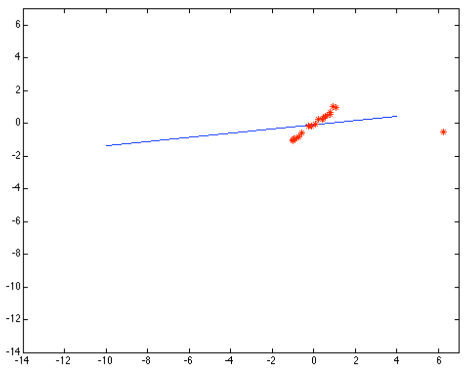
Robustness

- Squared error is a liability when model is wrong
  - One fix is EM - we'll do this shortly
  - Another is an M-estimator
    - Square nearby, threshold far away
  - A third is RANSAC
    - Search for good points

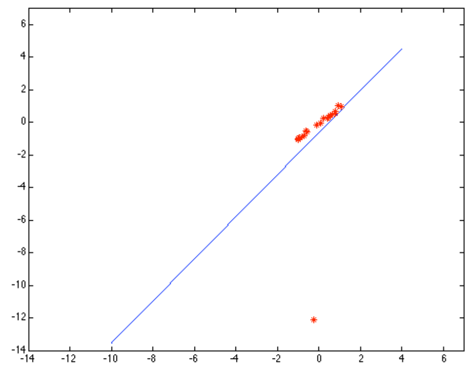
Least squares fit (good example)



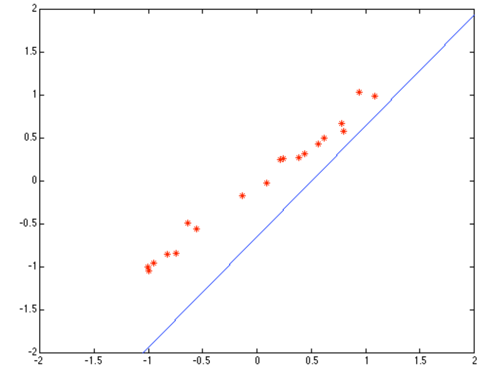
Least squares fit (destroyed by outlier)



Least squares fit (warped by outlier)



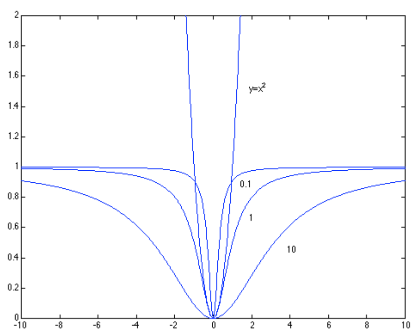
Least squares fit (previous slide zoom in)



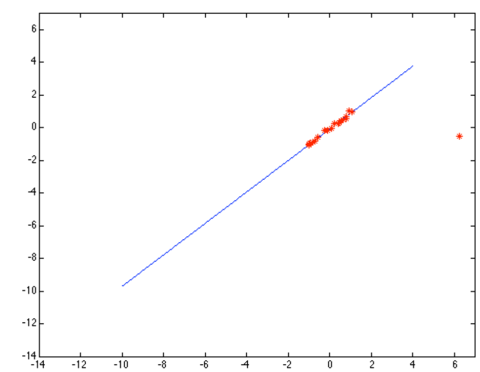
Example of a robust estimator. The effect of outliers are mitigated. After a certain distance, errors count the same.

$y=x^2/(x^2+s^2)$

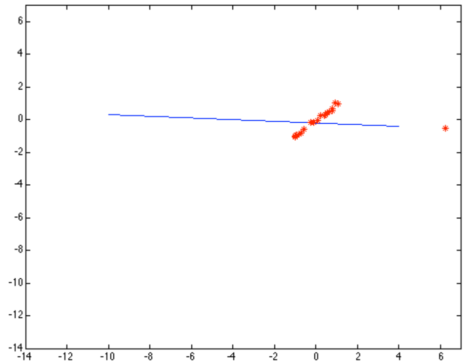
(Curve for different values of *s* shown)



Line fit with estimator with good choice for *s*

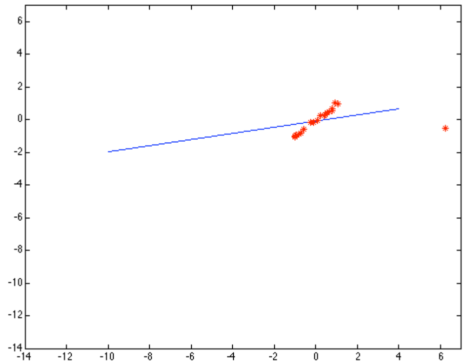


Line fit with estimator with choice for s that is too small



If s is too small, then the data is ignored too much

Line fit with estimator with choice for s that is too big



If s is too big, then we are back towards least squares

RANSAC

- Choose a small subset uniformly at random
- Fit to that
- Anything that is close to result is signal; all others are noise
- Refit
- Do this many times and choose the best

RANSAC

- Issues
  - How many times?
    - Often enough that we are likely to have a good line
  - How big a subset?
    - Smallest possible
  - What does close mean?
    - Depends on the problem
  - What is a good line?
    - One where the number of nearby points is so big it is unlikely to be all outliers

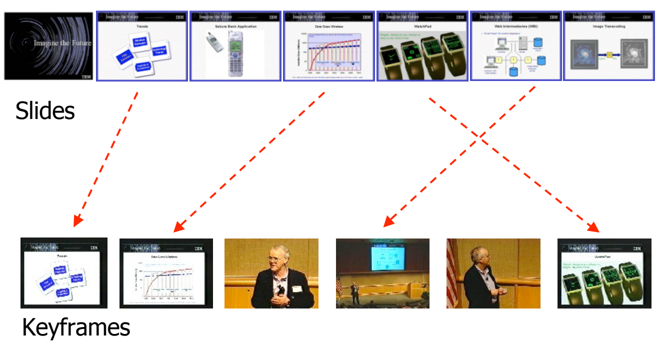
Algorithm 15.4: RANSAC: fitting lines using random sample consensus

Determine:
  n — the smallest number of points required
  k — the number of iterations required
  t — the threshold used to identify a point that fits well
  d — the number of nearby points required
    to assert a model fits well
Until k iterations have occurred
  Draw a sample of n points from the data
    uniformly and at random
  Fit to that set of n points
  For each data point outside the sample
    Test the distance from the point to the line
      against t; if the distance from the point to the line
      is less than t, the point is close
  end
  If there are d or more points close to the line
    then there is a good fit. Refit the line using all
    these points.
end
Use the best fit from this collection, using the
  fitting error as a criterion

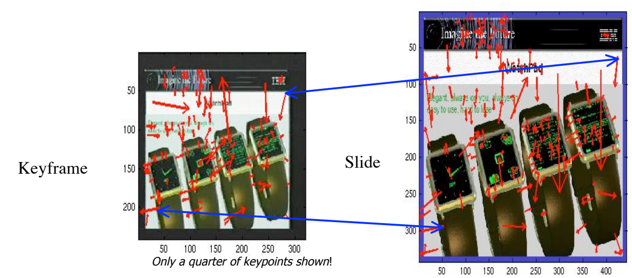
RANSAC and SIFT

- Powerful combination to find objects in images
- Exemplar image and image being studied typically have different camera angle or position.
- Recall that:
  - SIFT descriptors are relatively invariant to camera changes
  - SIFT matching leads to lots of "false" matches
- The main idea is that true matches should "agree"
- For planar objects, the definition of "agree" is quite simple

Matching Slides to Presentation Videos



SIFT (**S**cale **I**nvariant **F**eature **T**ransformation) keypoints review

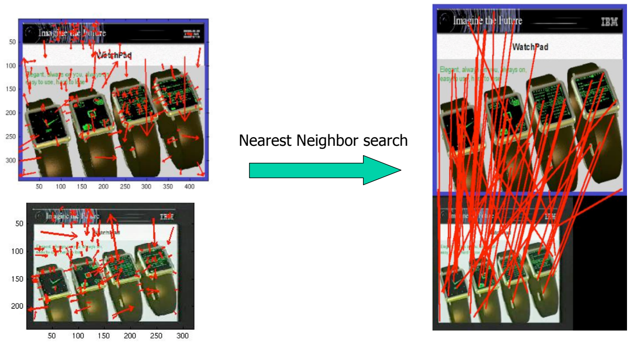


local feature descriptors

location, scale, orientation and a feature vector with 128 elements

Shown as a vector in the image

Nearest neighbor ratio has many outliers



Only a quarter of matches shown!

Planar Homography

Mappings of points on a plane in 3D satisfy a simple relation

$$\begin{bmatrix} x' \\ y' \\ \lambda \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{12} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

frame keypoints

slide keypoints

X' = H X

Optional

Derivation of Planar Homography

Consider a point on a plane given by

$$X = X_o + sX_1 + tX_2$$

under the two projective transforms

$$P = \begin{bmatrix} A & \mathbf{b} \end{bmatrix} \quad \text{and} \quad P' = \begin{bmatrix} A' & \mathbf{b}' \end{bmatrix}$$

This leads to two image points, $\lambda\mathbf{p}$ and $\lambda'\mathbf{p}'$.

Optional

Derivation of Planar Homography

$$\begin{aligned} \lambda\mathbf{p} &= \begin{bmatrix} A & \mathbf{b} \end{bmatrix} \begin{bmatrix} \mathbf{X}_o + s\mathbf{X}_1 + t\mathbf{X}_2 \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} sA\mathbf{X}_1^T + tA\mathbf{X}_2^T + A\mathbf{X}_0^T + \mathbf{b} \end{bmatrix} \\ &= \begin{bmatrix} A\mathbf{X}_1^T & A\mathbf{X}_2^T & A\mathbf{X}_0^T + \mathbf{b} \end{bmatrix} \begin{bmatrix} s \\ t \\ 1 \end{bmatrix} \\ &= V \begin{bmatrix} s \\ t \\ 1 \end{bmatrix} \end{aligned}$$

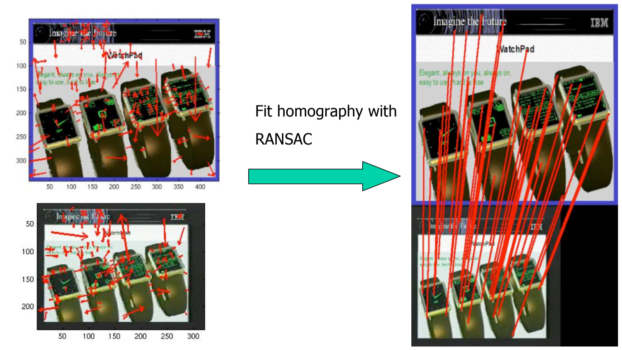
Optional

Derivation of Planar Homography

Similarly, $\lambda' \mathbf{p}' = V' \begin{bmatrix} s \\ t \\ 1 \end{bmatrix}$

and so $\lambda' \mathbf{p}' = V' \; V^{-1} \; \lambda \mathbf{p} = H \lambda \mathbf{p}$

Constraining matches by homography



Only a quarter of matches shown!

RANSAC approach

Repeat many times

- Randomly select enough matches to fit homography
- Compute homography
- Using that homography, measure error on best (say) 50%

Output best one found

Computing Homography

Seek H where $\begin{bmatrix} u' \\ v' \\ w' \end{bmatrix} = H \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$

H is only determined up to a scale factor (eight unknowns).

Let the rows of H be $h_1^T, \; h_2^T, \; h_3^T$.

$x' = \dfrac{u'}{w'}$ so $x'w' = u'$. Similarly, $y'w' = v'$

Also, $u' = h_1^T X$ and $v' = h_2^T X$ and $w' = h_3^T X$

Computing Homography

Each match then gives two linear equations

$$x' h_3^T X = h_1^T X \quad \text{and} \quad y' h_3^T X = h_2^T X$$

Hence four matches are OK.

This can be solved with homogenous least squares, but this is a bit unstable. A better way is the DLT (direct linear transform) method.

Details optional

Direct Linear Transform Method

$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}$ and $H\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$ are parallel, so their cross product should be zero.

This leads to the following more stable equation for homogenous least squares.

$$\begin{bmatrix} 0 & -w' X^T & v' X^T \\ w' X^T & 0 & -u' X^T \\ -v' X^T & u' X^T & 0 \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \\ h_3 \end{bmatrix} = 0$$

Details optional

Direct Linear Transform Method

The previous system has 3 equations per match, but only two of them are independent (one could be omitted, but no need for least squares methods, and hard to characterize the effect of breaking the symmetry).

By adding rows for additional points, we get the DLT method.