Fitting

- Work with a parametric representation for "objects"
  - (e.g "line", "ellipse").
- Most interesting case is when criterion is not local
  - can't tell whether a set of points lies on a line by looking only at each point and the next
- Three main questions:
  - what object represents a given set of tokens best?
  - which of several objects gets which token? (**correspondence**!)
  - how many objects are there?

Example: Hough Transform for lines

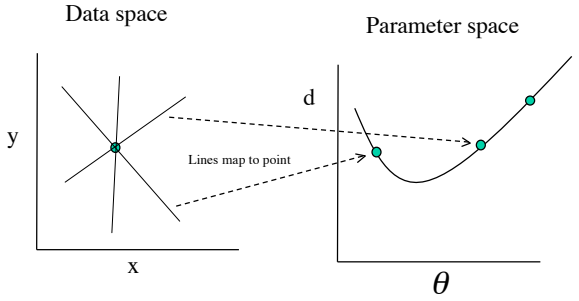
- A line is the set of points (x, y) such that

$$(\sin\theta)x + (\cos\theta)y + d = 0$$

- Different choices of θ, d>0 give different lines
- For any (x, y) there is a family of lines through this point, given by

$$(\sin\theta)x + (\cos\theta)y + d = 0$$

- The choice of θ fixes d. The family of lines has **one** parameter.



$$(\sin\theta)x + (\cos\theta)y + d = 0$$

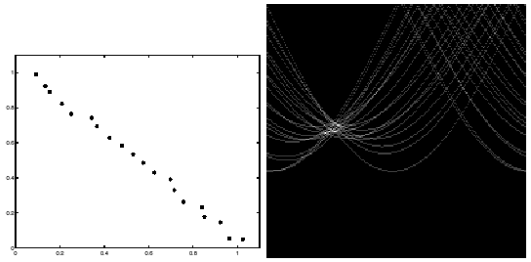
(Note: Curve is **not** accurate)

Example: Hough Transform for lines

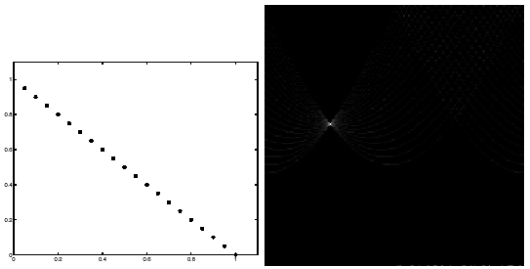
- Main idea: Each observed (x,y) votes for all (θ, d) satisfying

$$(\sin\theta)x + (\cos\theta)y + d = 0$$

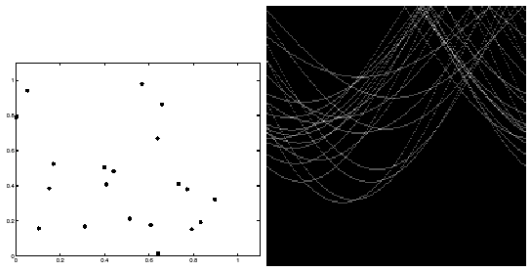
- Discretize the parameter space (θ, d) by an array
- Now each (x,y) leads to a bunch of votes (counts) in a (θ, d) grid (along the curve in the preceding slide).
- To find lines, let all edge points (x,y) vote, and look for (θ, d) cells with lots of votes.



tokens votes



tokens votes

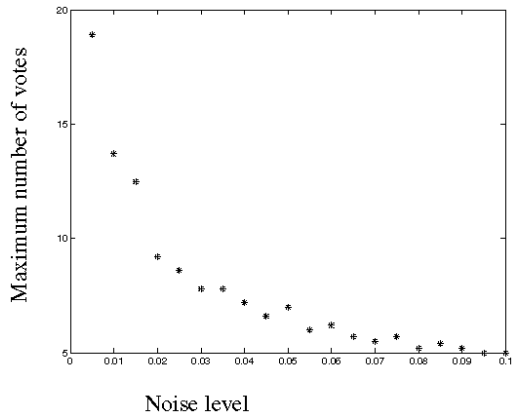


tokens votes

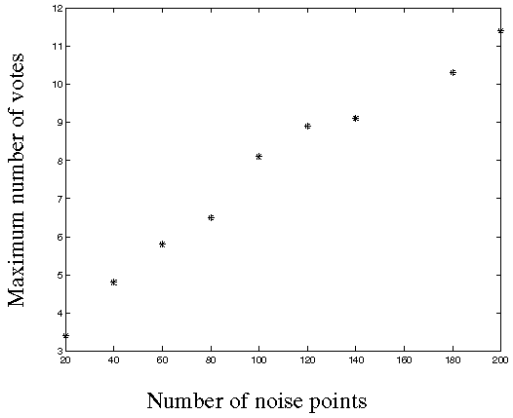
Difficulties with the Hough transform

- How big should the cells be? (too big, and we cannot distinguish between quite different lines; too small, and noise causes lines to be missed)
- How many lines?
  - count the peaks in the Hough array
- Who belongs to which line?
  - tag the votes
- Hough transform is a useful idea, but it is not often satisfactory in practice, because problems with noise and cell size defeat it

Votes for a real line of 20 points versus noise



Votes for a line in a picture that does not have one



Probabilistic Fitting

- Generative probabilistic model
  - Tells a story about how stochastic data comes to be
  - Darts fall around the center of the board, but where exactly?
  - Consider a model with parameters, $\boldsymbol{\theta}$
  - Consider an observation, x_i
  - We denote the probability of seeing x_i under the model by:

$$P(x_i \mid \Theta)$$

Read "given" or "conditioned on"
Restricts to the case of $\boldsymbol{\theta}$

Defined by $P(A \mid B) = \frac{P(A,B)}{P(B)}$

Probabilistic Fitting

- Multiple observations
  - Suppose we have multiple observations, in a vector **x**
  - What is the probability of **x**?

Probabilistic Fitting

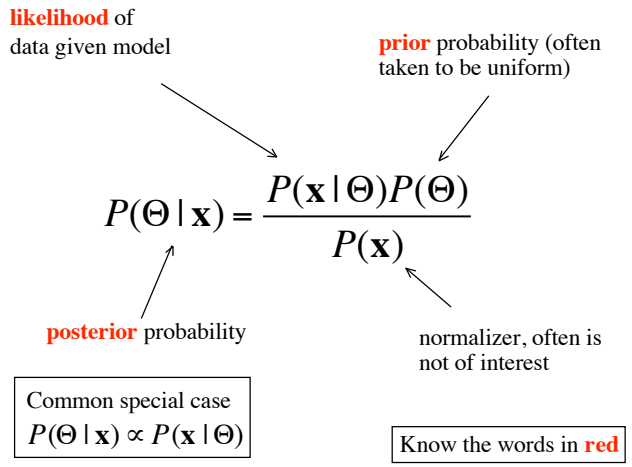
- So, given the model, we have the probability of observing the data

$$P(\mathbf{x} \mid \Theta) = \prod P(x_i \mid \Theta)$$

- But what we really want is the probability of the model (parameters) given the data!
- Bayes rule comes to the rescue!

Bayes Rule

- Bayes rule: $P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}$
- Proof $P(A,B) = P(B \mid A)P(A) = P(A \mid B)P(B)$
- With our notation: $P(\Theta \mid \mathbf{x}) = \frac{P(\mathbf{x} \mid \Theta)P(\Theta)}{P(\mathbf{x})}$



Probabilistic Fitting

- If we assume uniform prior, then find the likelihood for the parameters is represented by:

$$P(\Theta \mid \mathbf{x}) \propto P(\mathbf{x} \mid \Theta)$$

- Now the objective is to find the parameters Θ such that this *likelihood* is maximum
- Note--this is the same as finding the parameters which minimize the **negative log likelihood**

Probabilistic fitting with independence and uniform prior

Finding the "best" model under simple circumstances

$\underset{\Phi}{\text{maximize}}\ P(\Theta \mid \mathbf{x})$ (one definition of best Θ)

$\underset{\Phi}{\text{maximize}}\ P(\mathbf{x} \mid \Theta)$ (by Bayes rule, uniform prior)

$\underset{\Phi}{\text{minimize}}\ -\log(P(\mathbf{x} \mid \Theta))$ (log is monotonic increasing)

$\underset{\Phi}{\text{minimize}}\ -\log\left(\prod P(x_i \mid \Theta)\right)$ (by independence)

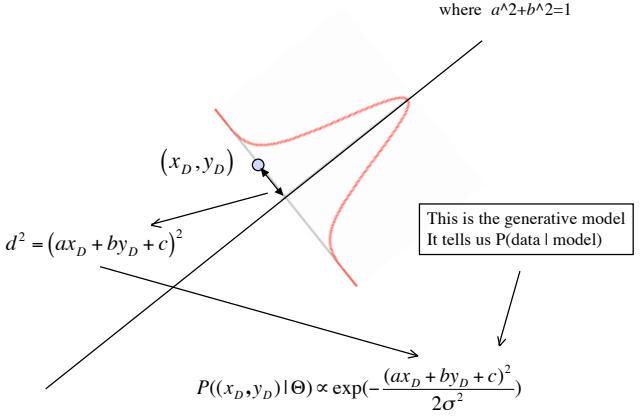
$\underset{\Phi}{\text{minimize}}\ -\sum \log(P(x_i \mid \Theta))$ (high school math)

- Back to lines: ax+bx+c=0 where a^2+b^2=1
- Algebraic fact: Distance squared from (x,y) to this line is (ax+by+c)^2
- **Generative model** for lines: Choose point on line, and then, with probability proportional to p(d), **normally distributed** (Gaussian), go a distance d from the line.
- Now the probability of an observed (x,y) is given by

$$P((x,y) \mid \Theta) \propto \exp\left(-\frac{(ax+by+c)^2}{2\sigma^2}\right)$$

Lines

Convenient formula for line
$ax+by+c=0$
where a^2+b^2=1



We have the probability of the observed (x,y) is given by

$$P((x,y) \mid \Theta) \propto \exp\left(-\frac{(ax+by+c)^2}{2\sigma^2}\right)$$

The negative log is

$$\frac{(ax+by+c)^2}{2\sigma^2}$$

And the negative log likelihood of multiple observations is

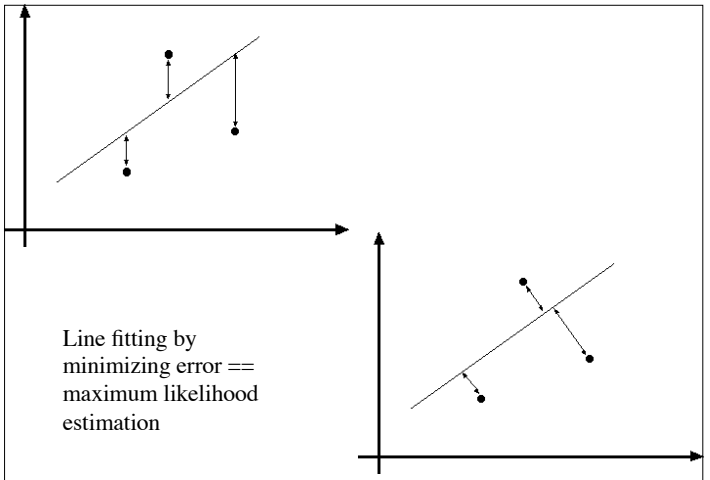
$$\frac{1}{2\sigma^2}\sum_i (ax_i + by_i + c)^2$$

From the previous slide, we had that the negative log likelihood of multiple observations is given by

$$\frac{1}{2\sigma^2}\sum_i (ax_i + by_i + c)^2 \qquad \text{(where } a^2 + b^2 = 1\text{)}$$

This should be recognizable as homogeneous least squares

Thus we have shown that least squares is maximum likelihood estimation under normality (Gaussian) error statistics!



Line fitting by minimizing error == maximum likelihood estimation

Fitting curves other than lines

- In principle, an easy generalization
  - Assuming Gaussian error statistics, Euclidean distance is a good measure
  - The probability of obtaining a point, given a curve, is given by a negative exponential of distance squared
- In practice, this can be hard
  - It can be difficult to compute the distance between a point and a curve
  - Circles, ellipses, and a few others are not too hard
  - Otherwise, craft an approximation
  - §15.3 has more

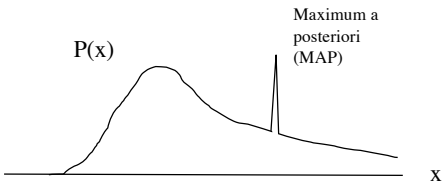
More on the Bayesian Method

- Recall that a generative probabilistic model
  - Tells a story about how stochastic data comes to be
  - Provides likelihood given data given model

$$P(\{\mathbf{x}_i\} \mid \Theta)$$

- Bayes rule
  - Tells us how to go *from* data given model *to* model given data
  - Tell us how to combine prior knowledge and evidence from data
  - Gives a probability distribution for an answer
    - Ideal for further reasoning
    - Supports various estimates (see cartoon on next slide)
    - Supports "risk" functions

Bayesian Estimators



Information from Priors and Data

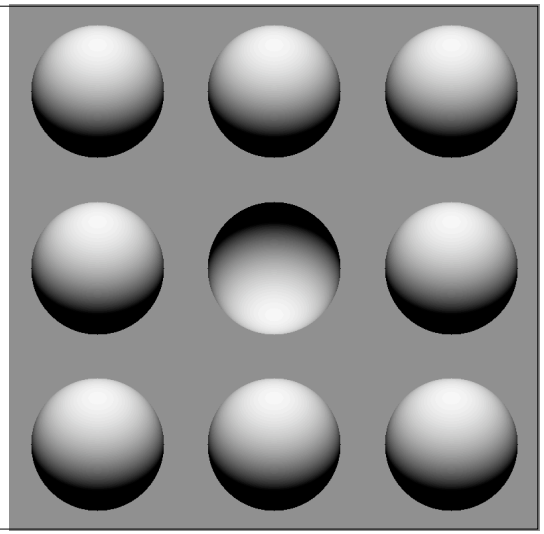
- Recall that vision problems do not have unique solutions!
  - We have to choose solutions suggested both by data and by what we believe (world knowledge)
  - What we believe about the world is the the prior

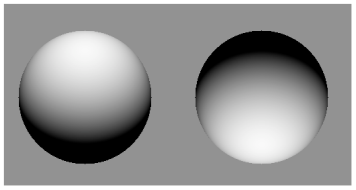
Simple example*

- 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

Vision example





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

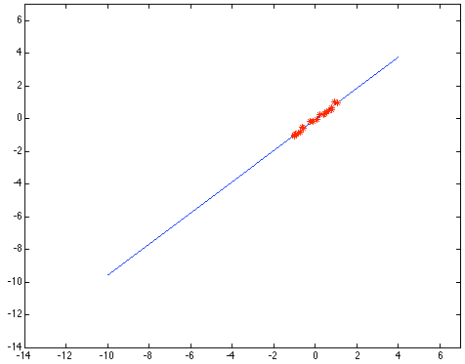
Hypothesize k lines (perhaps uniformly at random)
or
Hypothesize an assignment of lines to points
  and then fit lines using this assignment

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

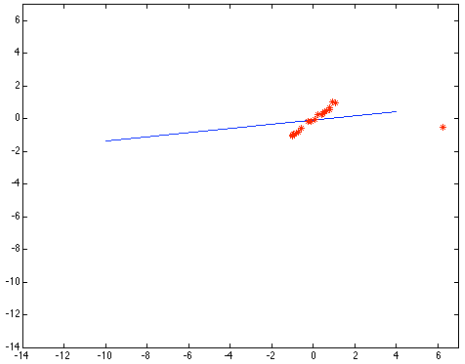
Robustness

- 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

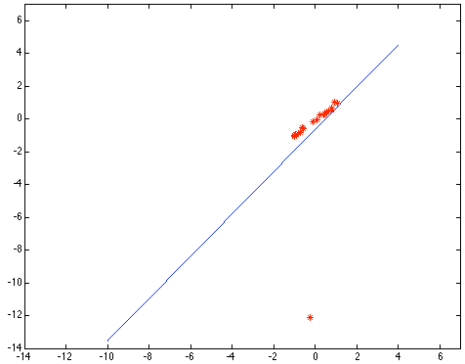
Least squares fit (good example)



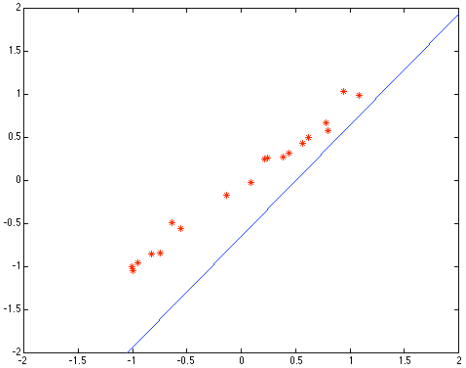
Least squares fit (destroyed by outlier)



Least squares fit (warped by outlier)



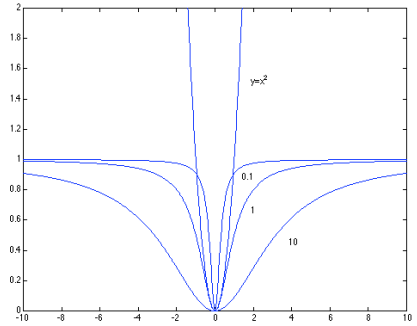
Least squares fit (previous slide zoom in)



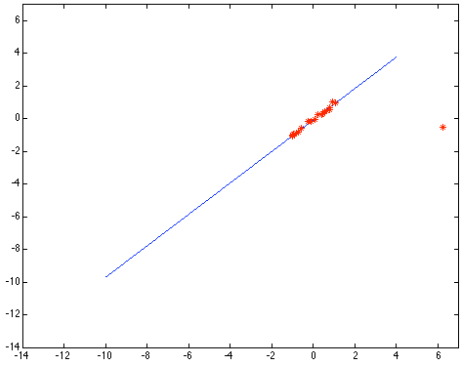
Example of a robust estimator. The effect of outliers are mitigated. After a certain distance, errors count the same.

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

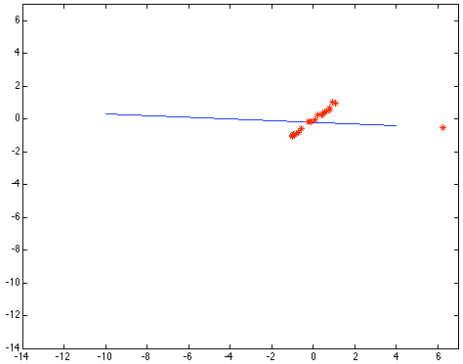
(Curve for different values of *s* shown)



Line fit with estimator with good choice for *s*

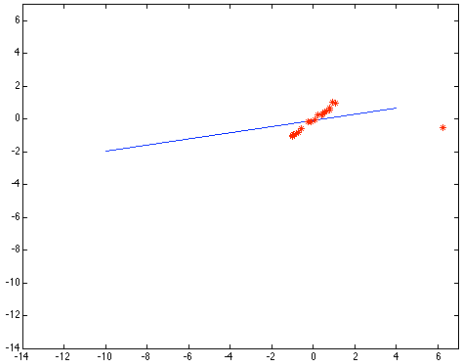


Line fit with estimator with choice for *s* that is too small



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

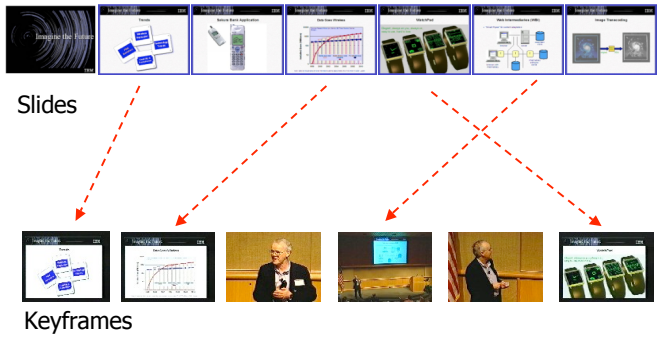
Algorithm 15.4: RANSAC: fitting lines using random sample consensus

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

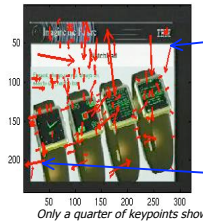
- Powerful combination to find objects in images
- Exemplar image and image being studied typically have different camera angle or position.
- Recall that:
  - SIFT descriptors are relatively invariant to camera changes
  - SIFT matching leads to lots of “false” matches
- The main idea is that true matches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

Matching Slides to Presentation Videos



SIFT (Scale Invariant Feature Transformation) keypoints review

Keyframe



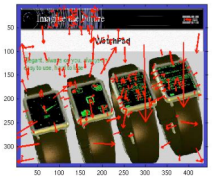
Slide

local feature descriptors

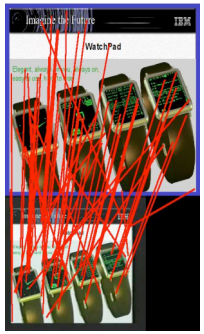
location, scale, orientation and a feature vector with 128 elements

Shown as a vector in the image

Nearest neighbor ratio has many outliers



Nearest Neighbor search



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

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

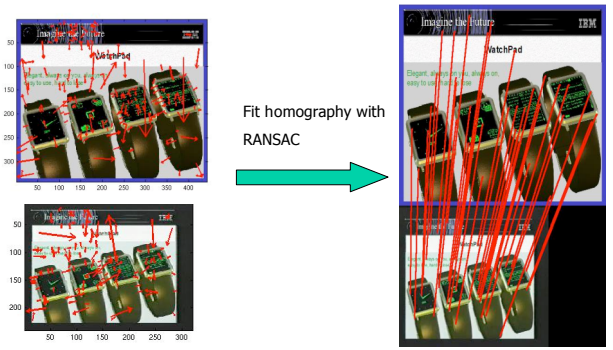
Optional

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.

$x' = \frac{u'}{w'}$ 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 can be omitted).

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