RANSAC

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

RANSAC

- How big a subset?
  - Smallest possible for the particular model (for a line, use 2 points)
- What does close mean?
  - Depends on the problem
  - Two strategies
    - Points within some fit threshold
    - Best k% points
- What is a good line?
  - One where the number of nearby points is so big it is unlikely to be all outliers (another threshold decision).

RANSAC

- How many iterations?
  - Often enough that we are likely to have a good model
    - Goes up with model complexity and belief about percentage of outliers
- Following notation from: http://en.wikipedia.org/wiki/RANSAC
  - Let w be the probability of getting an inlier.
  - Assume n points are needed for the model.
  - Suppose you want to be sure of a valid fit with probability, p.
  - Then the number of iterations, k, needed is:

$$k = \frac{\log(1-p)}{\log(1-w^n)}$$

Algorithm 15.4: RANSAC: fitting lines using random sample consensus

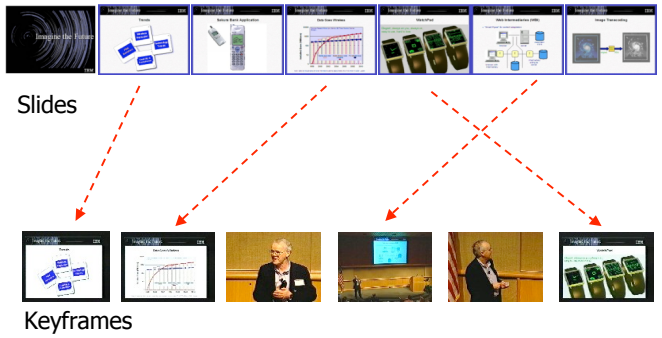
Determine:
  n — the smallest number of points required
  k — the number of iterations required
  t — the threshold used to identify a point that fits well
  d — the number of nearby points required
    to assert a model fits well
Until k iterations have occurred
  Draw a sample of n points from the 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

Note use of threshold

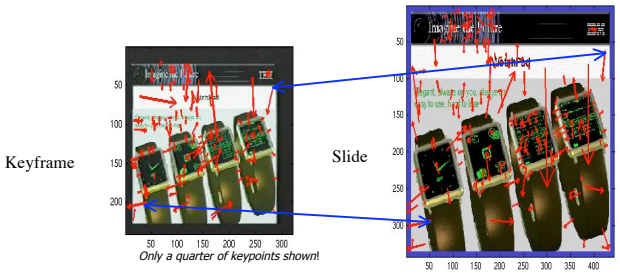
RANSAC and SIFT

- 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

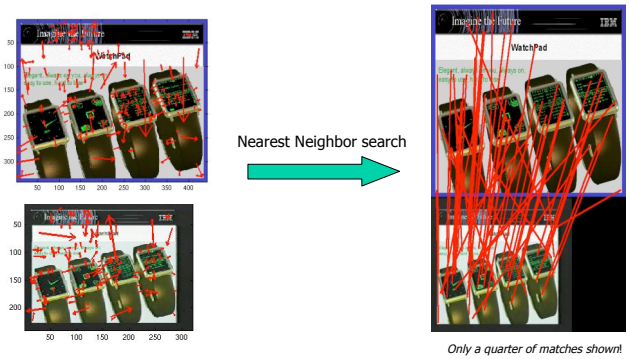


local feature descriptors

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

Shown as a vector in the image

Nearest neighbor ratio has many outliers



Planar Homography

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

$$\lambda' \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

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

$$\lambda\mathbf{p} = \begin{bmatrix} A & \mathbf{b} \end{bmatrix} \begin{bmatrix} \mathbf{X}_o + s\mathbf{X}_1 + t\mathbf{X}_2 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} A & \mathbf{b} \end{bmatrix} \begin{bmatrix} \mathbf{X}_1 & \mathbf{X}_2 & \mathbf{X}_o \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s \\ t \\ 1 \end{bmatrix}$$

3x4 4x3

$$= \begin{bmatrix} A\mathbf{X}_1 & A\mathbf{X}_2 & A\mathbf{X}_0 + \mathbf{b} \end{bmatrix} \begin{bmatrix} s \\ t \\ 1 \end{bmatrix}$$

3x3

$$= V \begin{bmatrix} s \\ t \\ 1 \end{bmatrix}$$

Notes for 2010

- This is a slightly different derivation than the draft notes.
- The draft notes have extra transposes (i.e., X^T) that are incorrect.

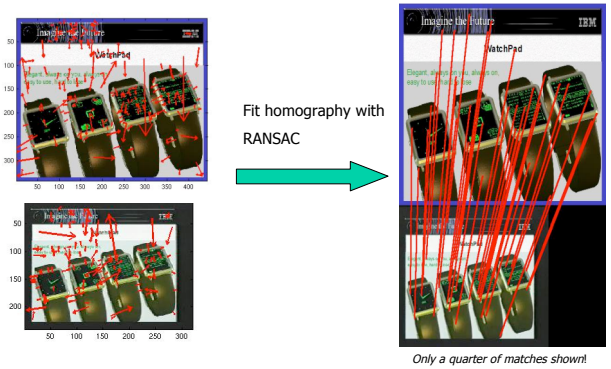
Optional

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

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.

* Doing this is part of the homework. You may want to review how we set up the equations for camera calibration. Possibly helpful hint --- begin by noticing that you can make X the row vector and the h_x column vectors.

Details optional

Direct Linear Transform Method

From before $\lambda' \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = H \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$

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

i.e., $\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} \parallel \begin{bmatrix} h_1^T \bullet X \\ h_2^T \bullet X \\ h_3^T \bullet X \end{bmatrix}$

so, for example, the first component of the cross product gives

$y' h_3^T \bullet X - h_2^T \bullet X = y' X^T \bullet h_3 - X^T \bullet h_2 = 0$

Details optional

Direct Linear Transform Method

This leads to the following more stable set of equations.

$$\begin{bmatrix} 0 & -X^T & y'X^T \\ X^T & 0 & -x'X^T \\ -y'X^T & x'X^T & 0 \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \\ h_3 \end{bmatrix} = 0$$

Details optional

Direct Linear Transform Method

For homogenous least squares.

$$\begin{bmatrix} 0 & -X_1^T & y'X_1^T \\ X_1^T & 0 & -x'X_1^T \\ -y'X_1^T & x'X_1^T & 0 \\ & \dots & \\ 0 & -X_4^T & y'X_4^T \\ X_4^T & 0 & -x'X_4^T \\ -y'X_4^T & x'X_4^T & 0 \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \\ h_3 \end{bmatrix} = 0$$

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.

Segmentation/Grouping by EM

- A segment could be modeled as a Gaussian process that emits feature vectors (which could contain color; or color and position; or colour, texture and position).
- Segment parameters are mean and (perhaps) variance or covariance, and prior probability (was λ in the line fitting example).
- If we knew which segment each point belonged to, estimating these parameters would be easy **(this point should be familiar!)**
- If we know the parameters, we can compute the probabilities that a point belongs to each cluster (**soft clustering**).
- Here the model is a mixture of Gaussians (one for each cluster), and the way to fit the model is Expectation Maximization (EM).

EM flow chart