Registration

D.A. Forsyth, UIUC

Registration

- Place a geometric model in correspondence with an image
  - could be 2D or 3D model
  - up to some transformations
  - possibly up to deformation
- Applications
  - very important in medical imaging
  - building mosaics
  - representing shapes
  - form of object recognition

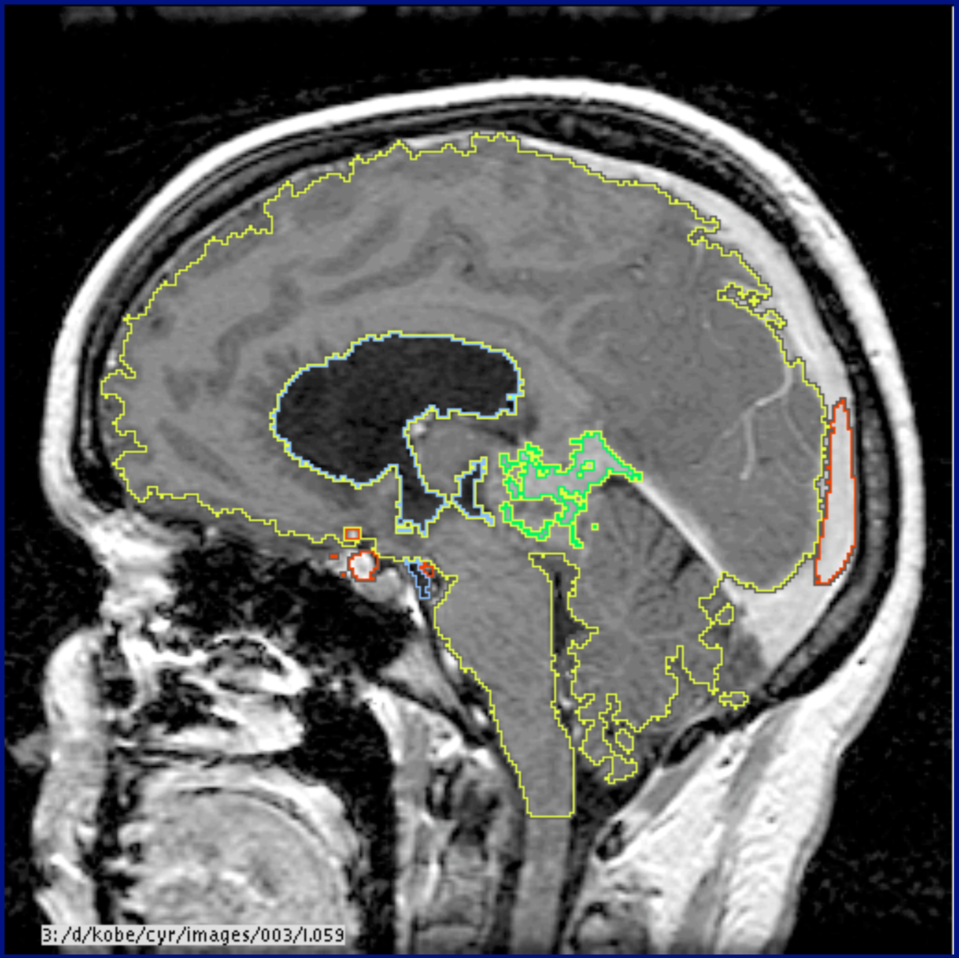
Correspondence

- Registration implies correspondence
  - because once they're in register, correspondence is easy
- Correspondence yields registrations
  - take correspondences and solve for best registration

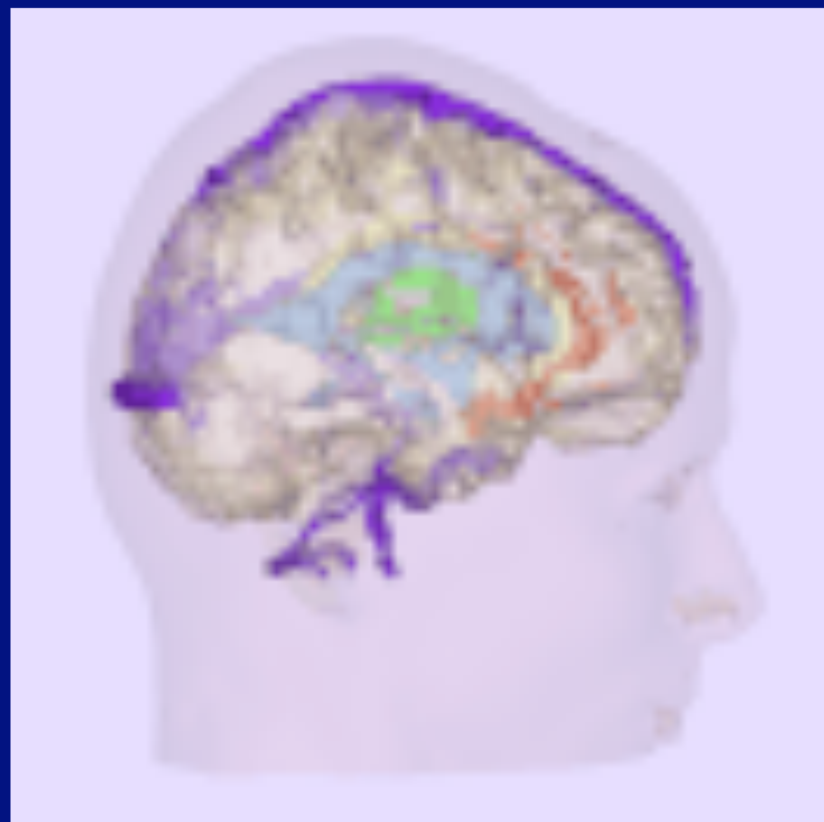
- Interact in a variety of ways in the main algorithms

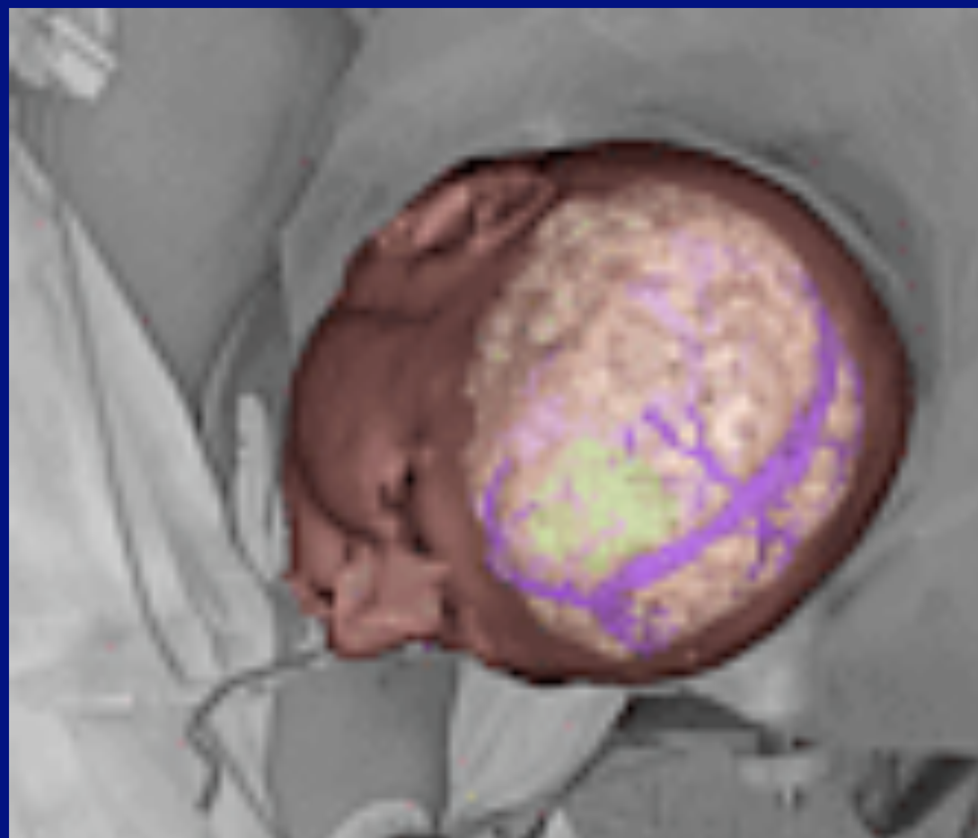
Medical Application

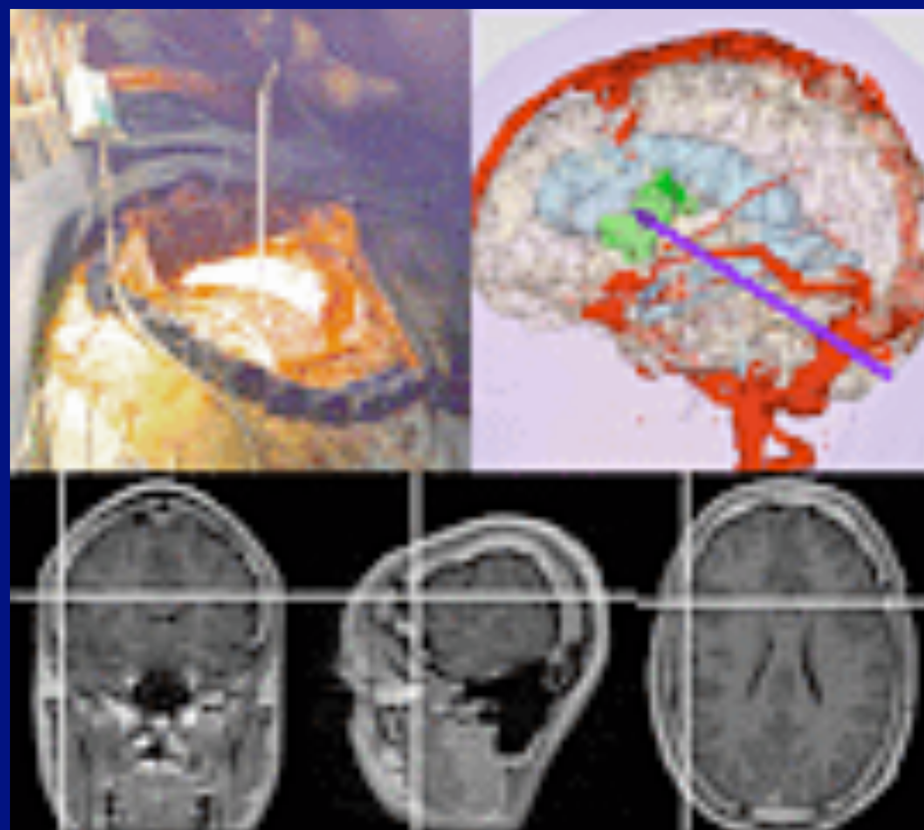
- Register scan of patient to actual patient
  - To remove only affected tissue
  - To minimize damage by operation planning
  - To reduce number of operations by planning surgery
- Register viewing device to actual patient
  - virtual reality displays



Images courtesy of Eric Grimson







Algorithms

- Hypothesize and test
- Iterative closest point
- Coarse-to-fine search

Registration by Hypothesize and Test

- General idea
  - Hypothesize correspondence
  - Recover pose
  - Render object in camera (widely known as backprojection)
  - Compare to image
- Issues
  - where do the hypotheses come from?
  - How do we compare to image (verification)?
- Simplest approach
  - Construct a correspondence for all object features to every correctly sized subset of image points
  - These are the hypotheses
  - Expensive search, which is also redundant.

Correspondences yield transformations

- 2D models to 2D images
  - Translation
    - one model point-image point correspondence yields the translation
  - Rotation, translation
    - one model point-image point correspondence yields the translation
    - one model direction-image direction correspondence yields the rotation
  - Rotation, translation, scale
    - two model point-image point correspondences

Correspondences yield transformations

- 3D models to 3D info
  - Translation
    - one model point-image point correspondence yields the translation
  - Rotation, translation
    - points, directions
      - one model point-image point correspondence yields the translation
      - two model direction-image direction correspondences for rotation
  - Rotation, translation, scale
    - points, directions
      - two model point-image point correspondences and one direction
    - lines
      - two disjoint line correspondences yield rotation, translation, scale
- Many other correspondences work

Correspondences yield transformations

- 3D models, 2D images, calibrated orthographic camera
  - Translation
    - one model point-image point correspondence yields all that can be known
  - Translation, rotation
    - three model point-image point correspondence yields all that can be known
- Etc (perspective cameras, and so on)

Pose consistency

- A small number of correspondences yields a camera
- Strategy:
  - Generate hypotheses using small numbers of correspondences (e.g. triples of points for a calibrated perspective camera, etc., etc.)
  - Backproject and verify
    - Notice that the main issue here is camera calibration
    - Appropriate groups are "frame groups"

```
For all object frame groups O
  For all image frame groups F
    For all correspondences C between
      elements of F and elements
      of O

      Use F, C and O to infer the missing parameters
      in a camera model

      Use the camera model estimate to render the object

      If the rendering conforms to the image,
        the object is present
    end
  end
end
```

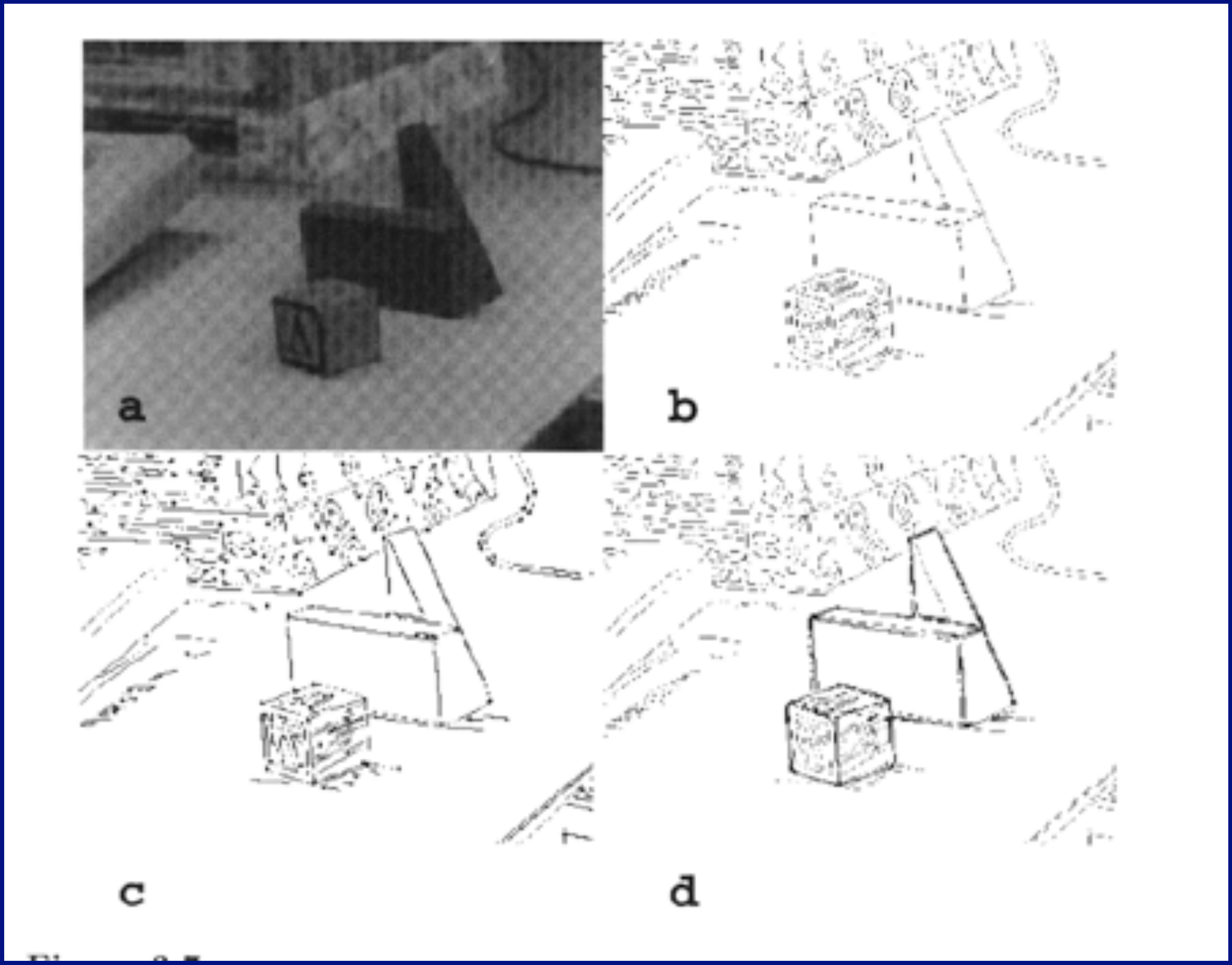
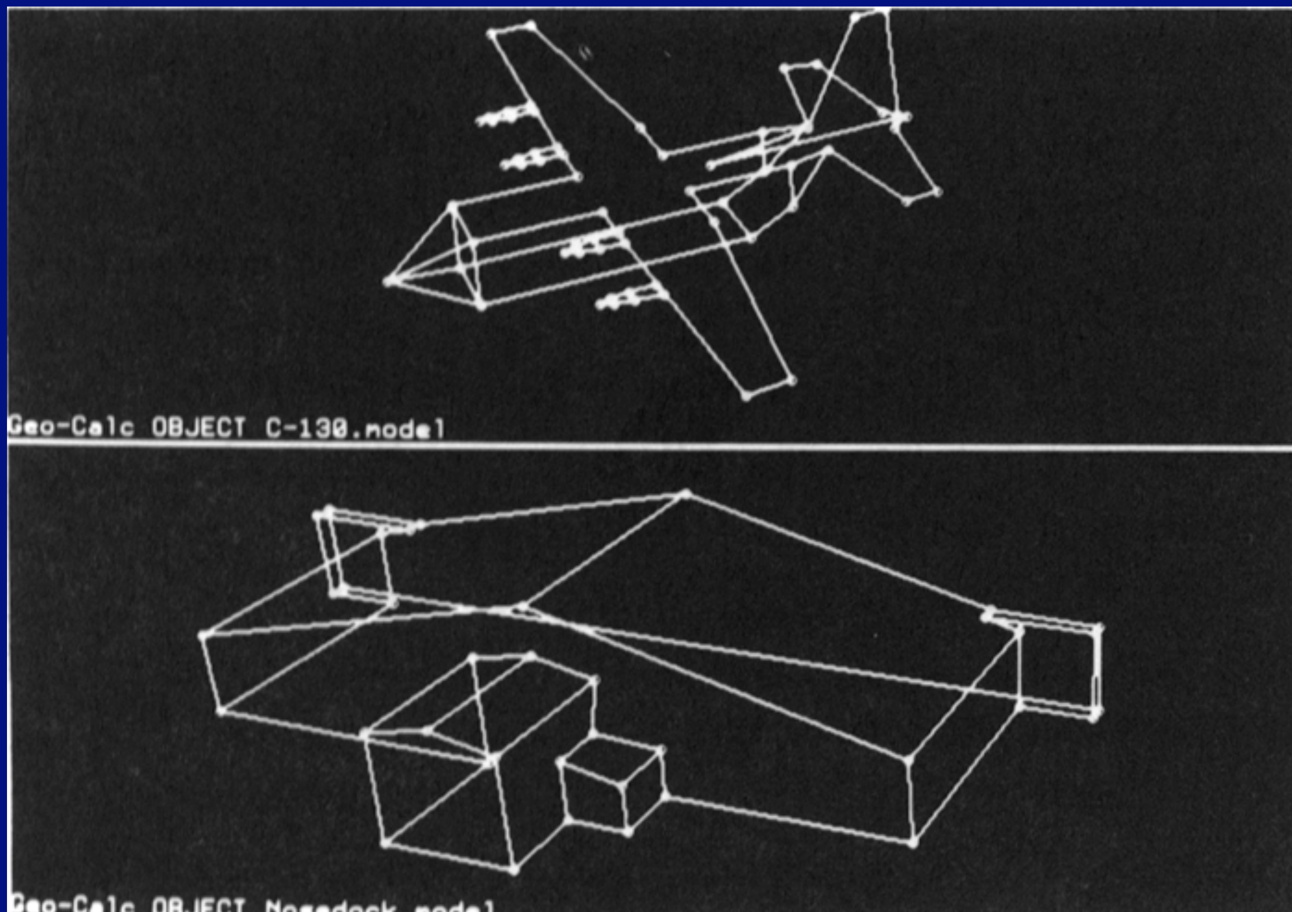


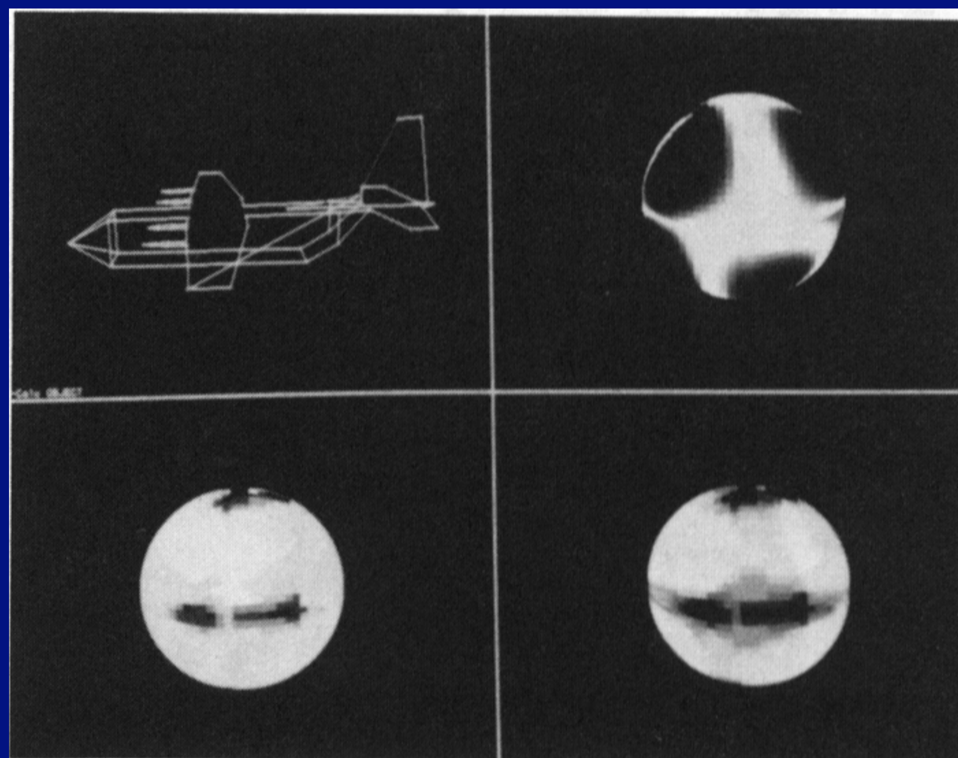
Figure from Huttenlocher+Ullman 1990

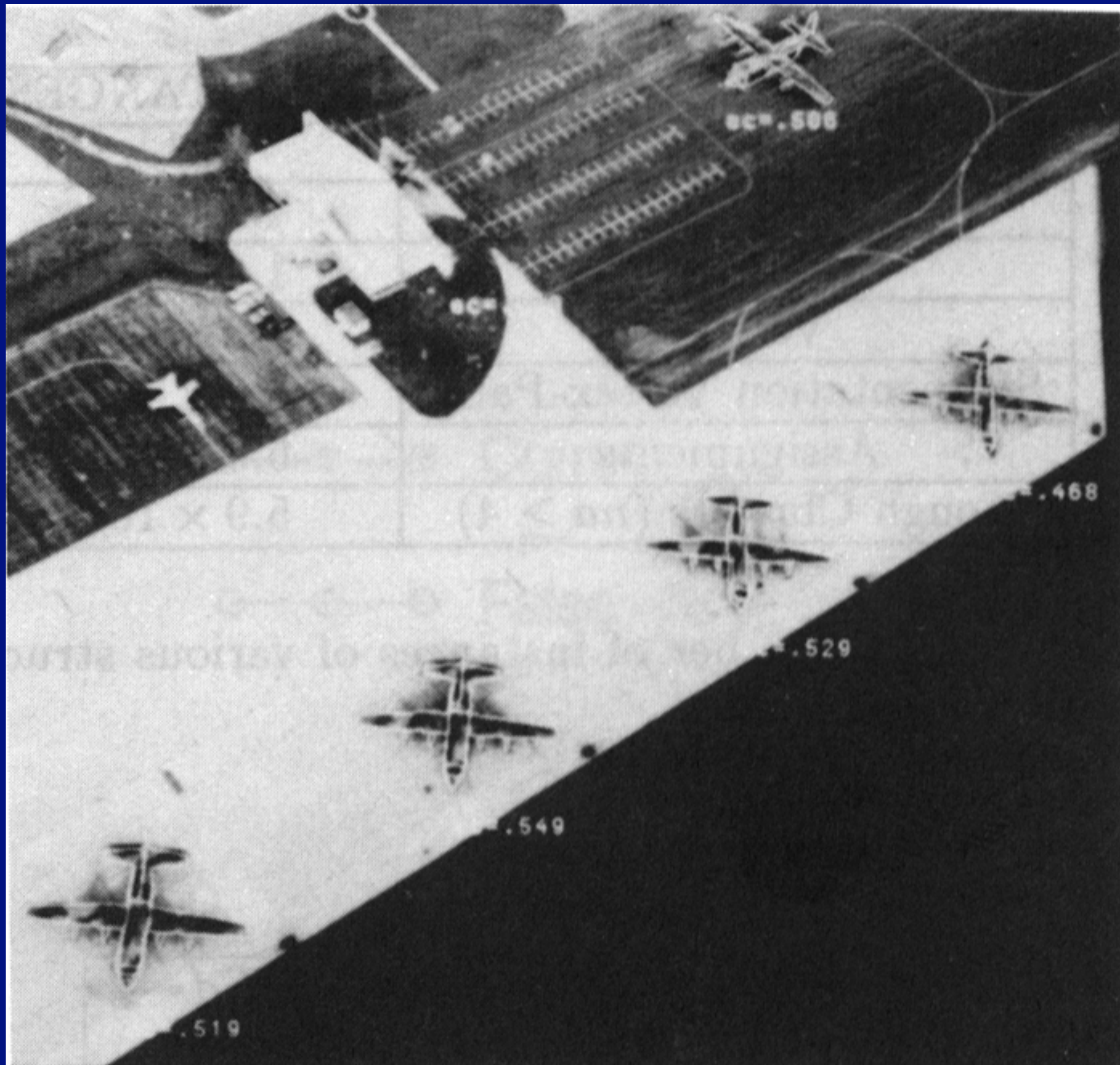
Voting on Pose

- Each model leads to many correct sets of correspondences, each of which has the same pose
  - Vote on pose, in an accumulator array
  - This is a hough transform, with all it’s issues.

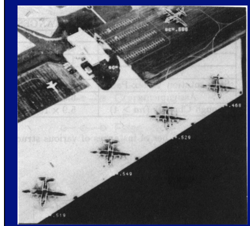
```
For all objects O
  For all object frame groups F(O)
    For all image frame groups F(I)
      For all correspondences C between
        elements of F(I) and elements
        of F(O)

        Use F(I), F(O) and C to infer object pose P(O)

        Add a vote to O's pose space at the bucket
        corresponding to P(O).
      end
    end
  end
end
For all objects O
  For all elements P(O) of O's pose space that have
    enough votes

    Use the P(O) and the
    camera model estimate to render the object

    If the rendering conforms to the image,
    the object is present
  end
end
```

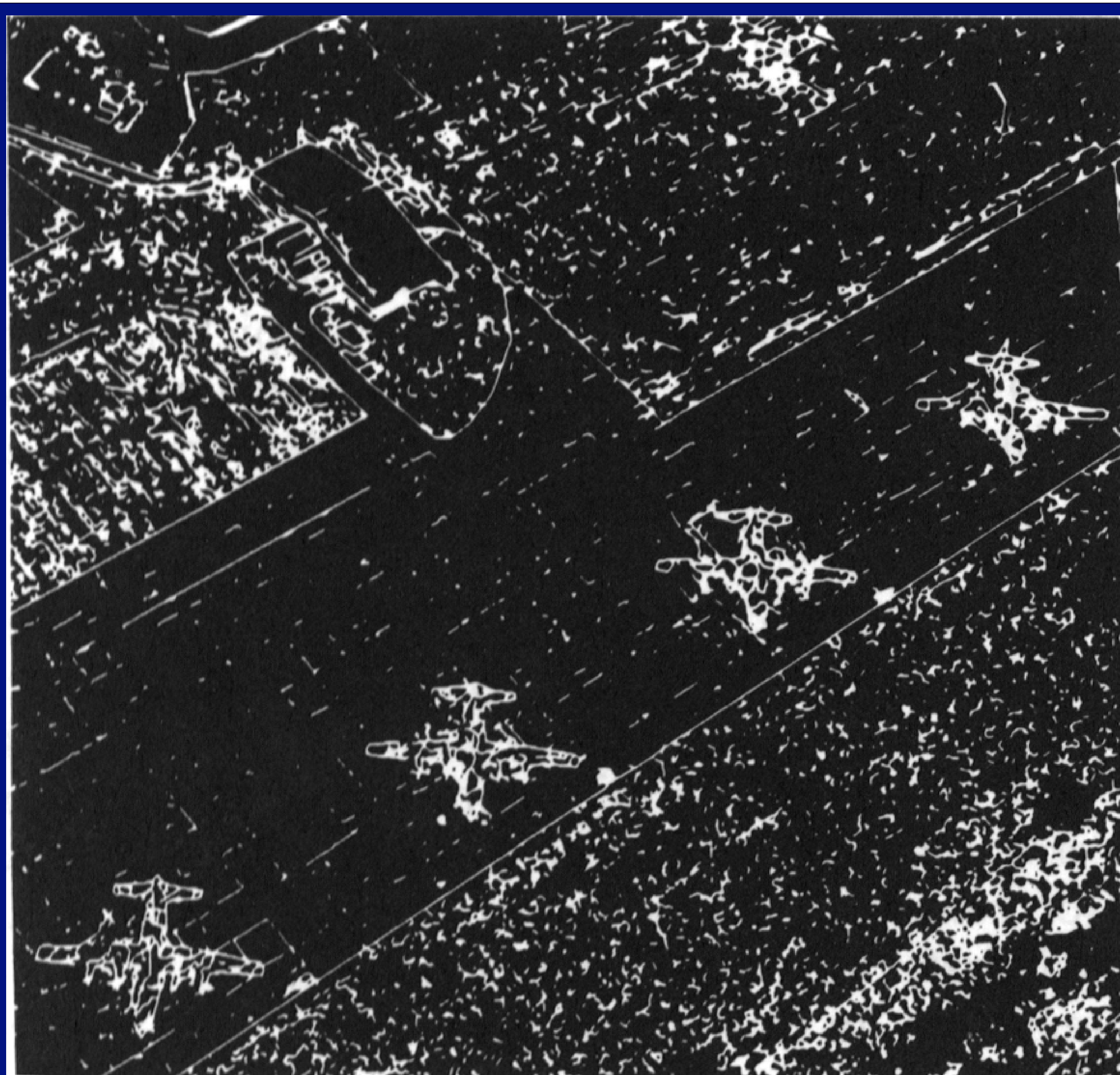

Geo-Calc OBJECT C-130.model
Geo-Calc OBJECT Nosedock.model

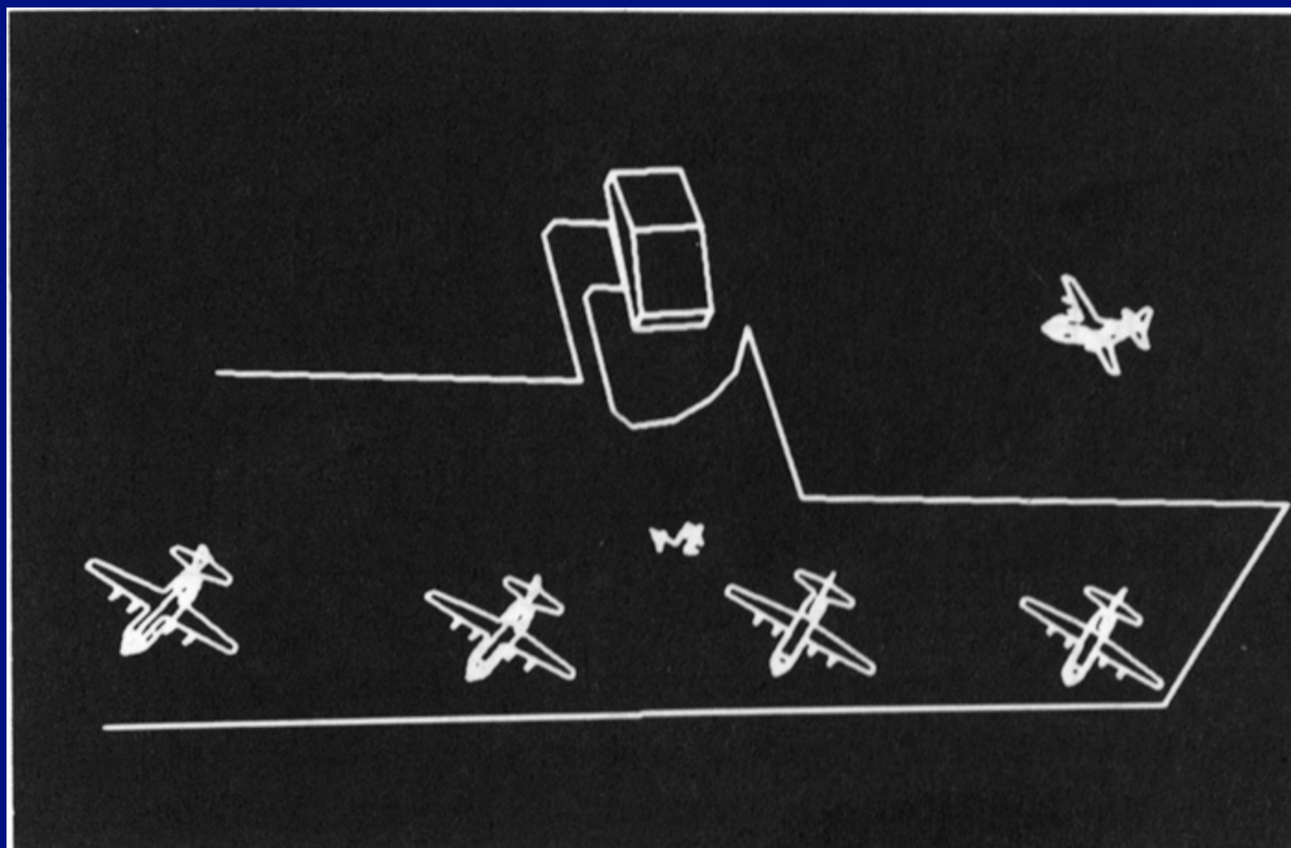




ec=.506
ec=
.468
.529
.549
.519







Verification

- Is the object actually there?
- Edge based
  - project object model to image, score whether image edges lie close to object edges
- Orientation based
  - project object model to image, score whether image edges lie close to object edges at the right orientation
- More sophisticated
  - Opportunity!

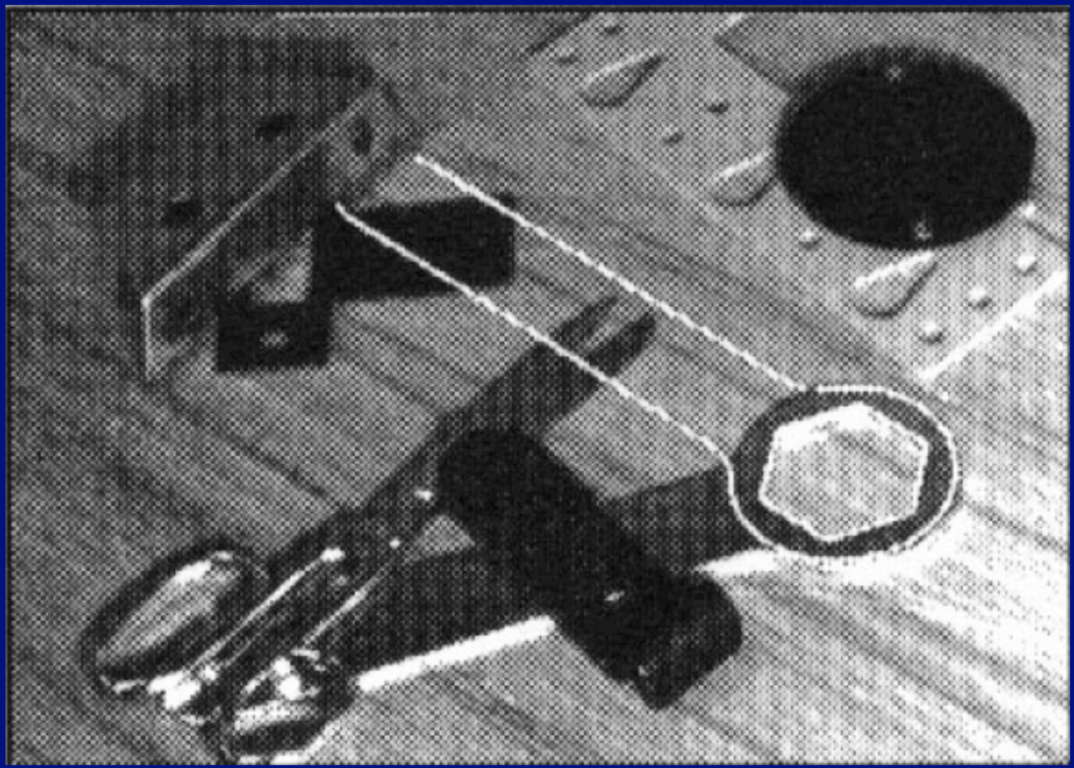
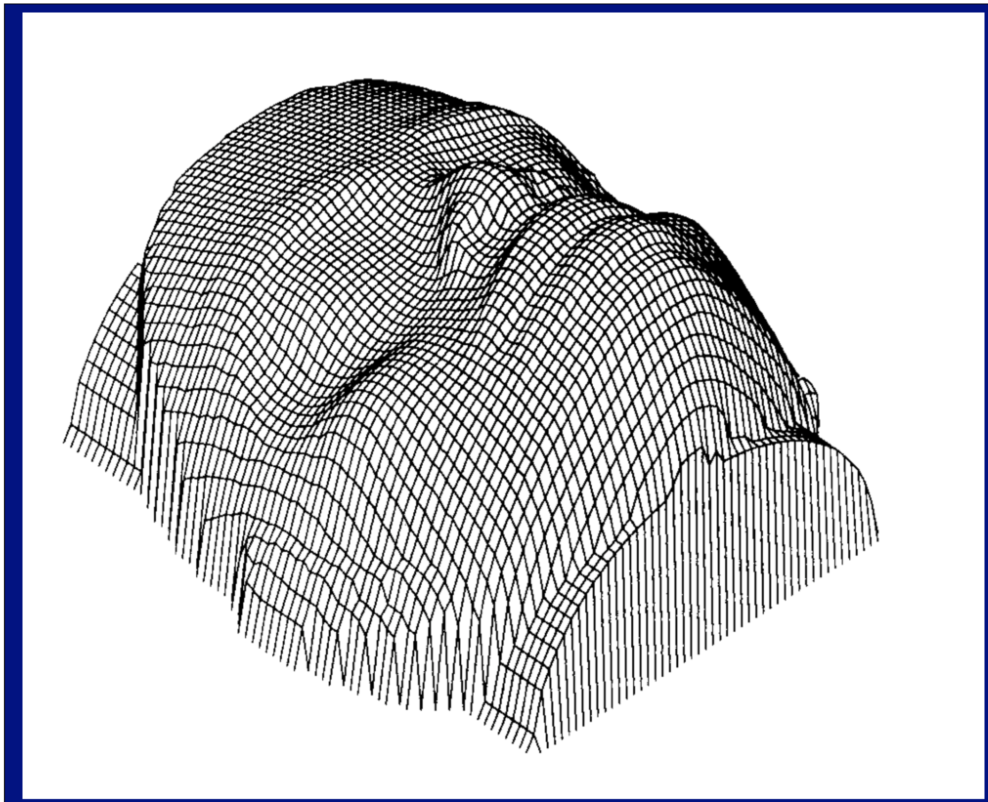


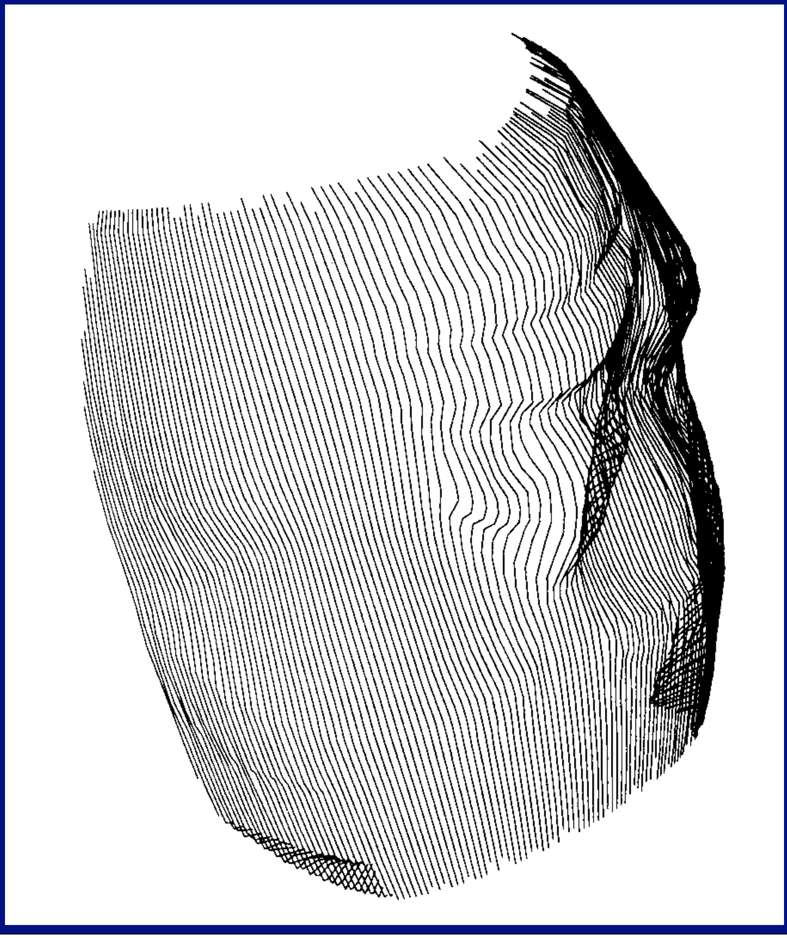
Figure from Rothwell et al, 1992

Iterative closest point

- For registering 2D-2D or 3D-3D point sets
  - typically under translation, rotation and scale
- Iterate
  - Find closest point on measurement to each point on model
    - using current pose
  - Minimize sum of distances to closest points as a function of pose
- Variants
  - model consists of lines, surface patches, etc.

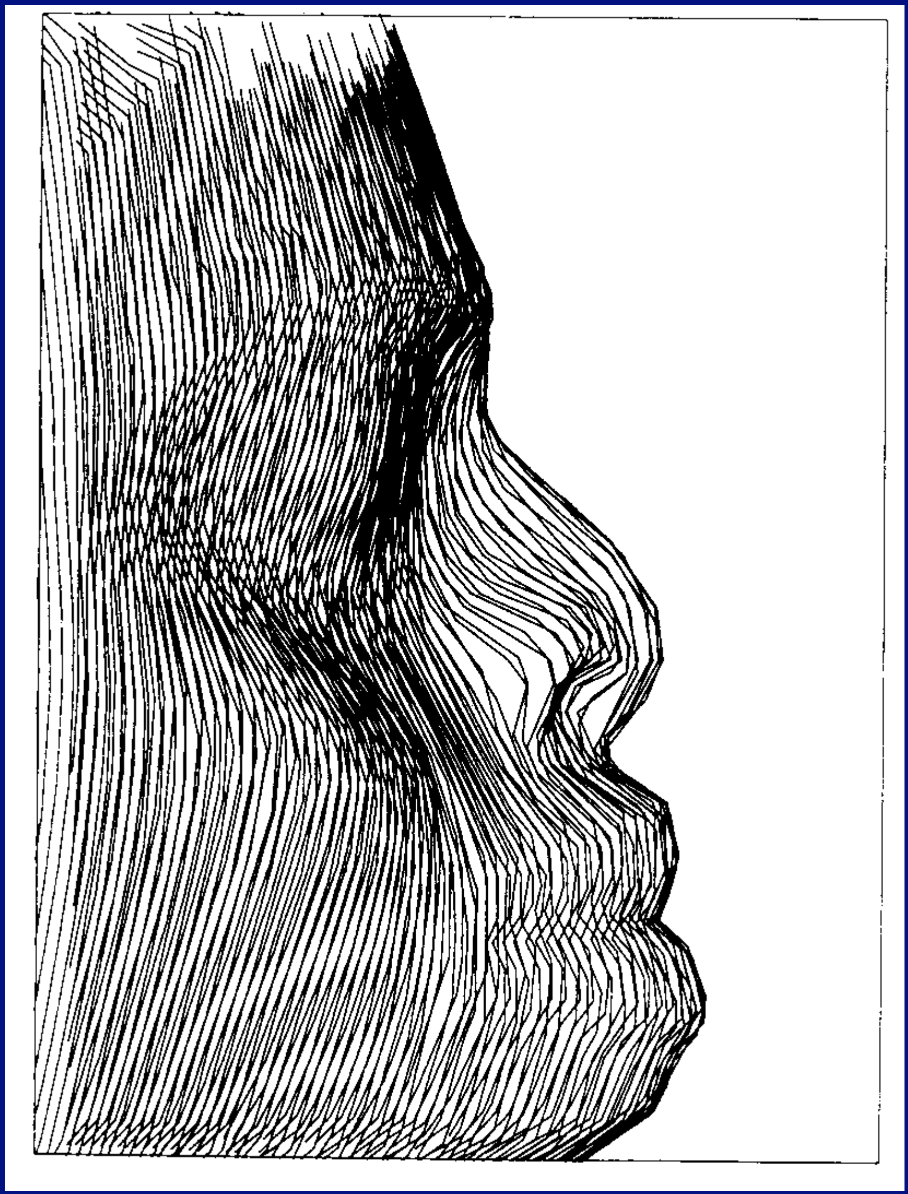


Model: triangle set of 8442 triangles



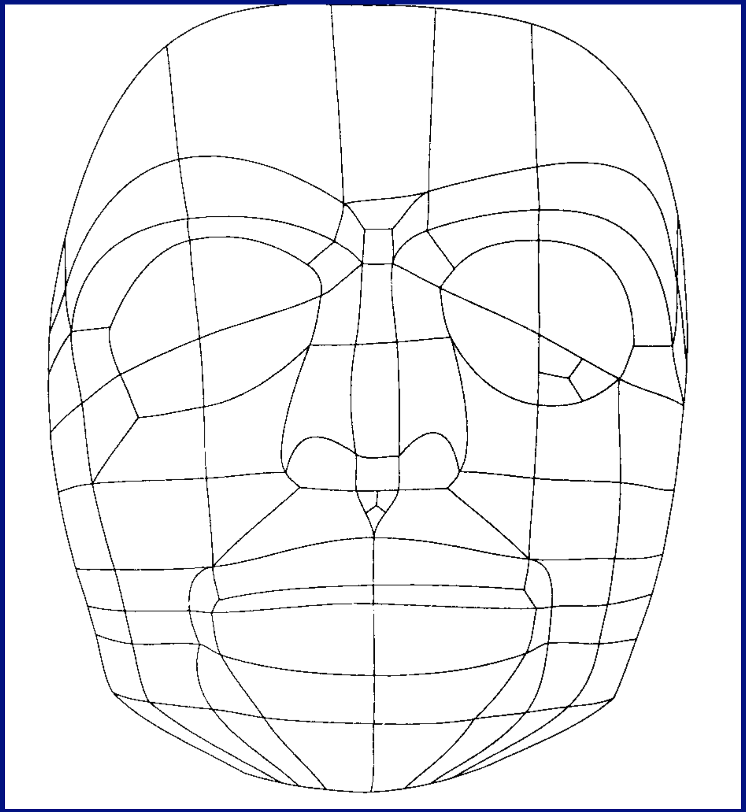
Point set

Figure from Besl+McKay, 1992

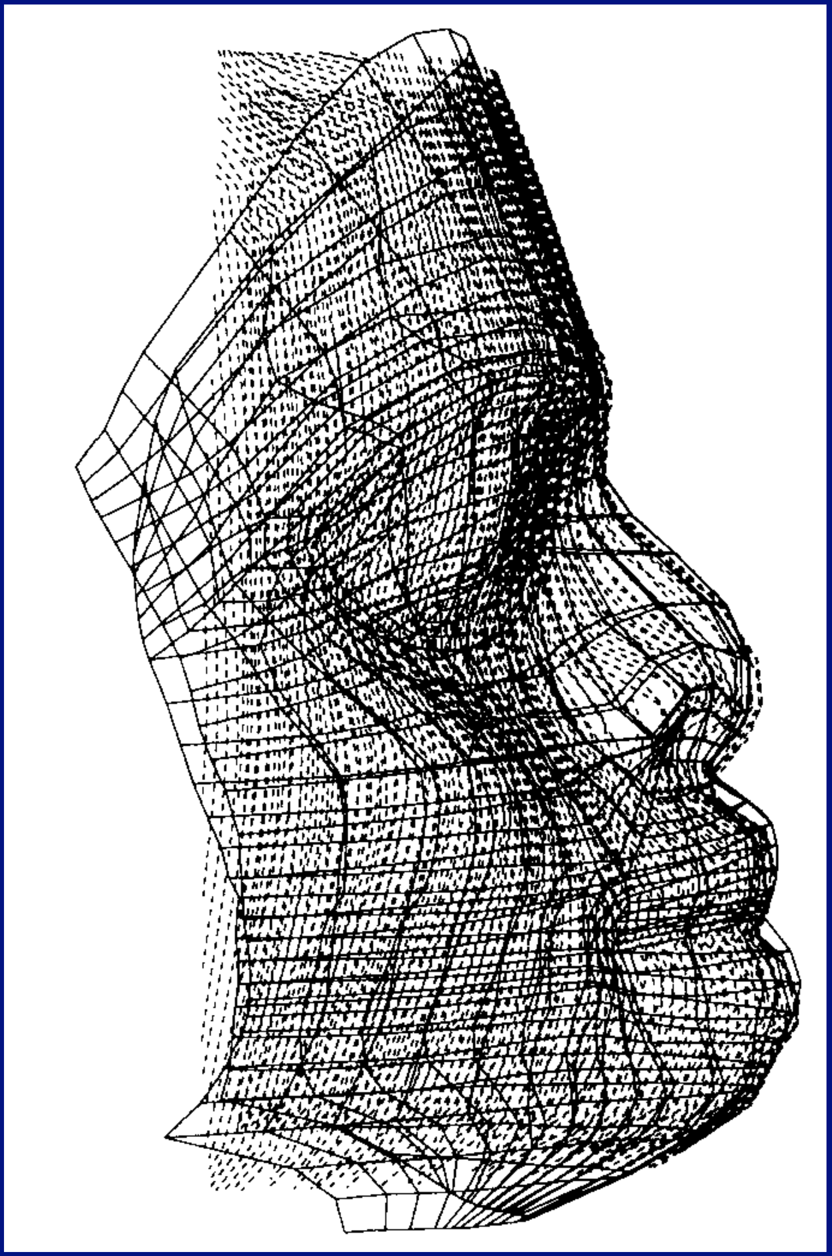


Points registered to triangles

Figure from Besl+McKay, 1992



Model: Bezier patches

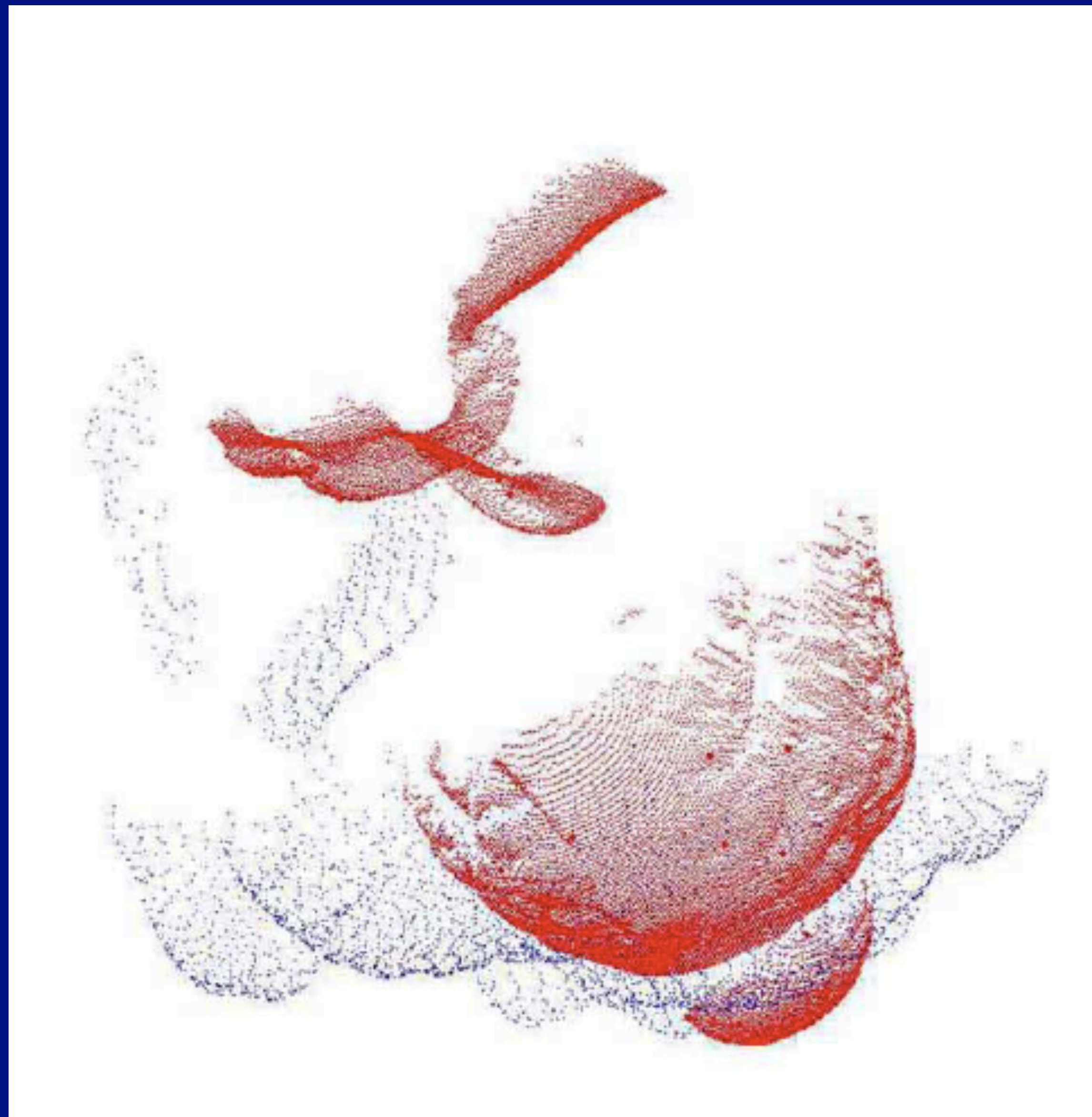


Registered to point set

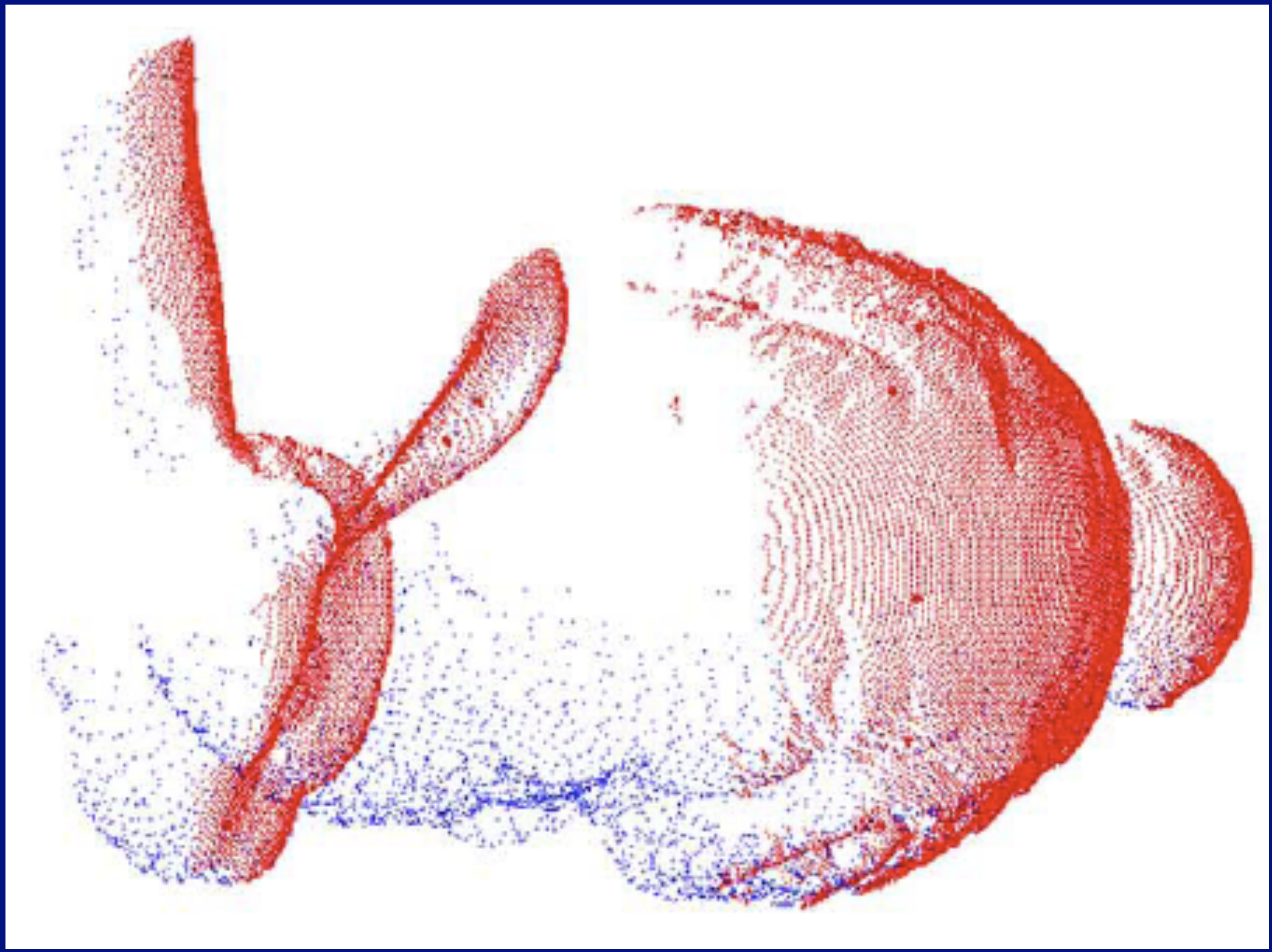
Figure from Besl+McKay, 1992

Variants

- Use Levenberg-Marquardt on robust error measure
  - ignore failures of differentiability caused by correspondence
  - Fitzgibbon 2003



Initial alignment (Fitzgibbon, 2003, red rabbit to blue rabbit)



Solution (Fitzgibbon, 2003, red rabbit to blue rabbit)

Coarse to fine search

- General idea:
  - many minima may be available for registration problems
    - eg ICP for 2D object to points on image edges
  - search a coarse representation at multiple points
  - take each local minimum, search a refined representation
    - possibly repeat multiple times
- Advantage
  - coarse representation is fast to search
    - so you can look at many poses
  - fine representation gives accurate estimates

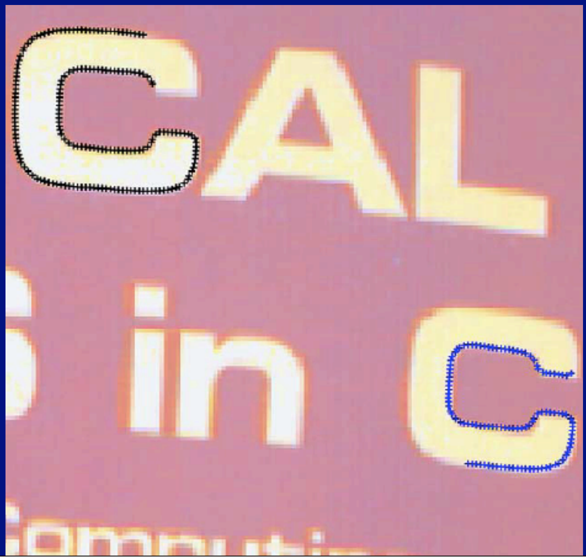


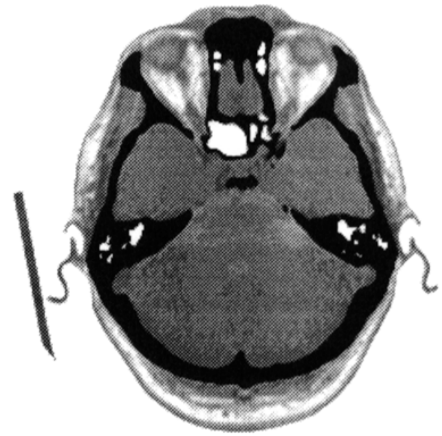
Figure from Fitzgibbon, 2003

Registration and deformation

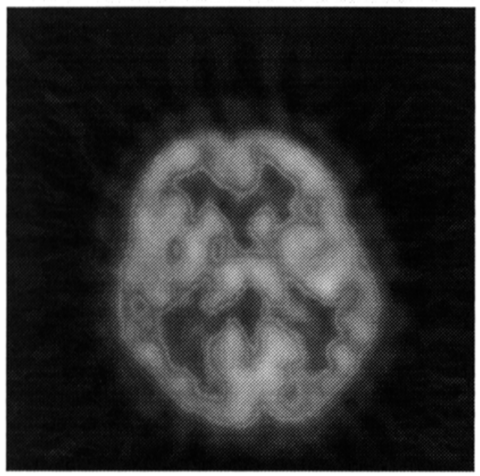
- Medical applications often deal with deformable objects
- Real objects often deform, too
    - equivalently, deformation is an important part of matching
    - e.g.
        - matching one car to another
        - matching one flower to another, etc.
- Idea:
    - build parametric deformation model into registration process

MRI

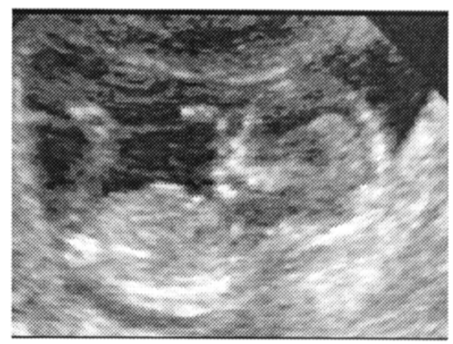
CTI



NMI



USI



Parametric deformation models

- Assume we have a set of points (x, y) which should deform to (u, v)

$$u = f_1(x, y, \theta), v = f_2(x, y, \theta)$$

- Good models
  - Affine

$$u = a_{00}x + a_{01}y + a_2, v = a_{10}x + a_{11}y + a_3$$

  - More deformation (here the f's are "small")

$$u = f_1(x, y, \theta) + a_{00}x + a_{01}y + a_2, v = f_2(x, y, \theta) + a_{10}x + a_{11}y$$

Radial basis function deformations

- Choose some special points in x, y space (x_i^*, y_i^*)
- deformation functions become:

$$f_l(x, y, \theta) = \sum_i \theta_i \phi(x, y; x_i^*, y_i^*)$$

  - where phi depends only on distance:
    - eg

$$\phi(x, y; x_i^*, y_i^*) = \frac{1}{(x - x_i^*)^2 + (y - y_i^*)^2 + \epsilon^2}$$

Radial basis function deformation

- We must choose theta, a's
- least squares

$$\sum_{j \in \text{points}} \left[\begin{array}{c} (u_j - \{[\sum_i \theta_{i,1}\phi(x_j, y_j; x_i^*, y_i^*)] + a_{00}x + a_{01}y + a_2\})^2 + \\ (v_j - \{[\sum_i \theta_{i,2}\phi(x_j, y_j; x_i^*, y_i^*)] + a_{00}x + a_{01}y + a_2\})^2 \end{array} \right]$$

- Solving this gives a linear system!
  - but we might get f's that are too big

Radial basis function deformation

- Penalize the least squares

$$\sum_{j \in \text{points}} \left[\begin{array}{c} (u_j - \{[\sum_i \theta_{i,1} \phi(x_j, y_j; x_i^*, y_i^*)] + a_{00}x + a_{01}y + a_2\})^2 + \\ (v_j - \{[\sum_i \theta_{i,2} \phi(x_j, y_j; x_i^*, y_i^*)] + a_{00}x + a_{01}y + a_2\})^2 \end{array} + \lambda(\theta_{i,1}^2 + \theta_{i,2}^2) \right]$$

- And we still have a linear system!

- ICP matching: Iterate
  - Fix a's, thetas, choose correspondences
  - Solve for a's, thetas

Deformation is like flow

- Notice the similarity between
  - estimating deformation   I_1(x, y) -> I_2(u(x, y), v(x, y))
  - estimating flow field   I(x, y, t) -> I(x+a(x, y), y+b(x,y), t+1)
- Recall
  - accurate local estimates of flow are hard (no good local description)
  - options
    - parametric flow model
    - smooth

Deformation is like flow

- Plausible flow model

$$I(x,y) \rightarrow I(x + m_1 x + m_2 y + m_3, y + m_4 x + m_5 y + m_6)$$

- Not much help if the m's are fixed
- Idea: let the m's vary with space, and penalize derivatives

- Cost function:

$$\sum_{x,y} (I(x,y) - I(x + m_1 x + m_2 y + m_3, y + m_4 x + m_5 y + m_6))^2$$

  - simplify to first order term in Taylor series

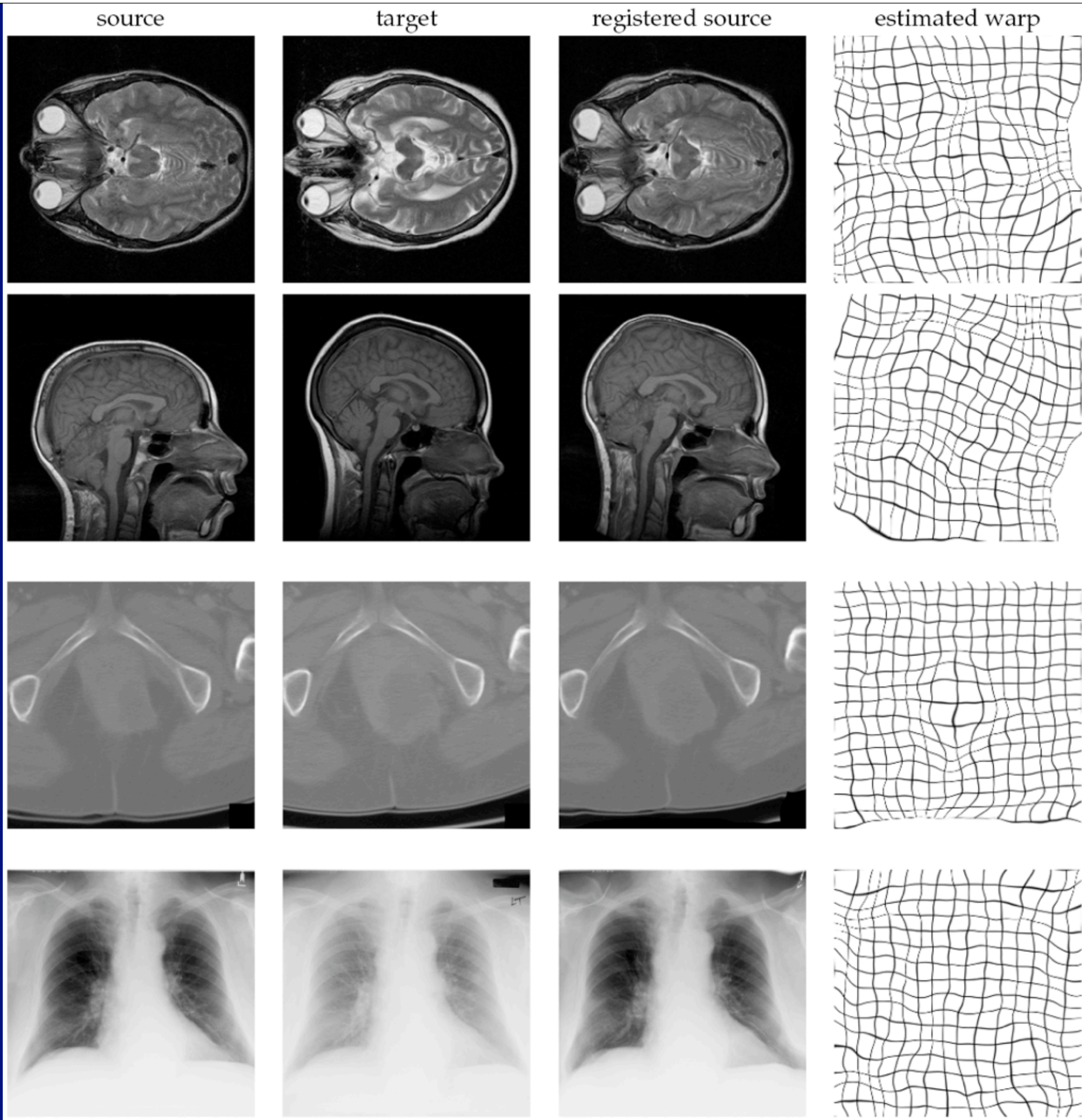
$$\sum_{x,y} ((m_1 x + m_2 y + m_3)\frac{\partial I}{\partial x} + (m_4 x + m_5 y + m_6)\frac{\partial I}{\partial y})^2$$

Deformation is like flow

- Overall cost function

$$\sum_{x,y}\left[\begin{array}{c}((m_1x+m_2y+m_3)\frac{\partial I}{\partial x}+(m_4x+m_5y+m_6)\frac{\partial I}{\partial y})^2+\\ \sum_l(\frac{\partial m_l}{\partial x}^2+\frac{\partial m_l}{\partial y}^2)\end{array}\right]$$

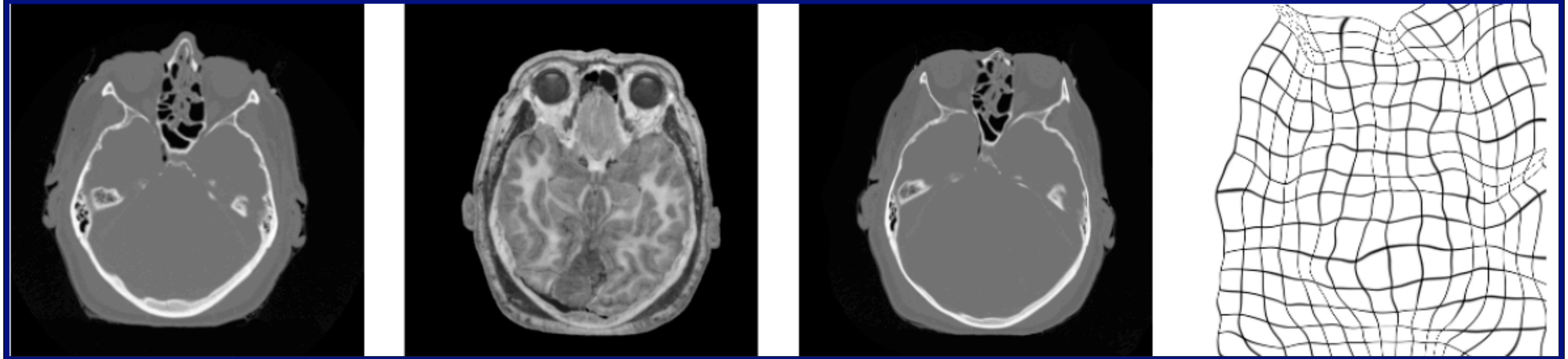
Periaswamy
Farid 03



What about multiple modes?

- We could model the change in intensity
  - eg I_1 -> a I_1 + b
  - then bung it in minimizer
- Use mutual information
  - (loosely) geometric registration between images gives a model of sensors
  - P(s_1=a, s_2=b)
  - maximize the mutual information in this model

$$I(A, B) = H(B) - H(B \mid A).$$



Periaswamy
Farid 03

Periaswamy
Farid 06