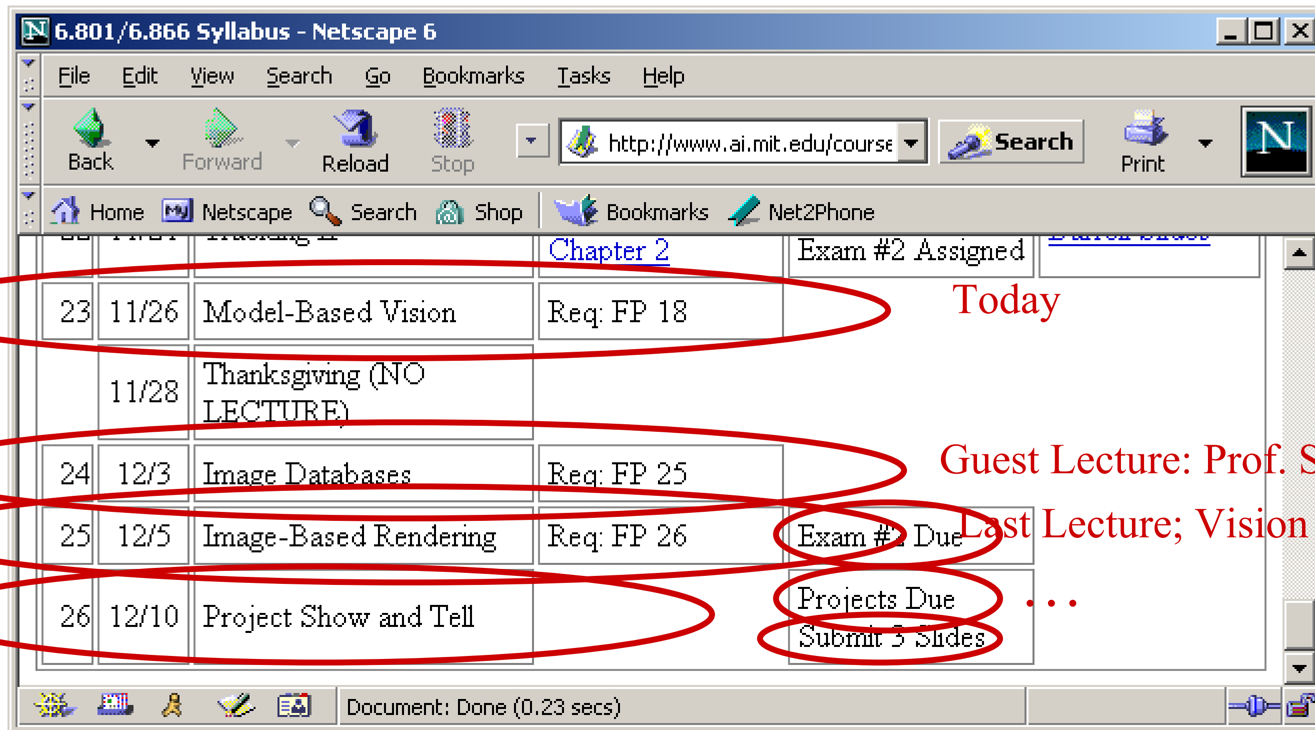
6.801/866

Model-based Vision

T. Darrell

Endgame



Today

Guest Lecture: Prof. Stan Sclaroff

Last Lecture; Vision and Graphics…

. . .

3-5 slides to Louis
Short narration in-class

- PS3 back today; mean = 82, std.dev. = 13
- Next week we'll distribute extra credit problem(s)

Model-based Vision

- Hypothesize and test
- Interpretation Trees
- Alignment
- Pose Clustering
- Invariances
- Geometric Hashing
- Medical Imaging Application

Approach

- Given
  - CAD Models (with features)
  - Detected features in an image
- Hypothesize and test
  - Guess
  - Render
  - Compare

Recognition by Hypothesize and Test

- General idea
  - Hypothesize object identity and correspondence
  - Recover camera (widely known as backprojection)
  - Render object in camera
  - Compare to image
- Issues
  - where do the hypotheses come from?
  - How do we compare to image (verification)?

Recognition by Hypothesize and Test

- 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.

What are the features?

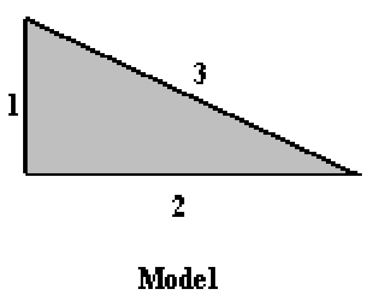
- They have to project like points
  - Lines
  - Conics
  - Other fitted curves
  - Regions (particularly the center of a region, etc.)

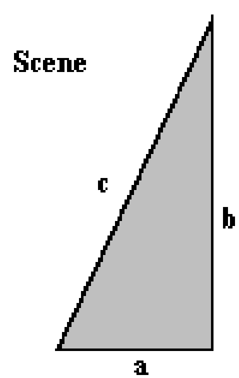
How to generate hypotheses?

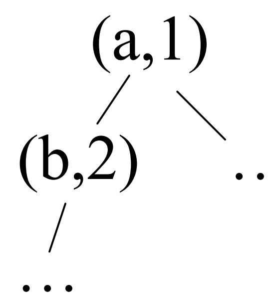
- Brute force
  - L objects with N features
  - M features in image
  - $O(LM^N)$!
- Add geometric constraints to prune search, leading to *interpretation tree search*
- Try subsets of features (frame groups)…

Interpretation Trees

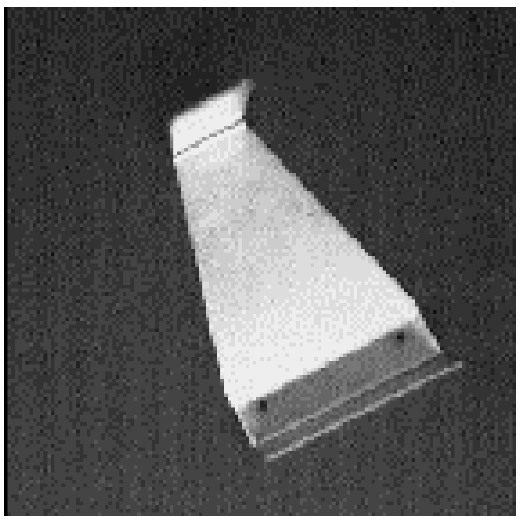
- Tree of possible model-image feature assignments
- Depth-first search
- Prune when unary (binary, …) constraint violated
  - length
  - area
  - orientation
- "Wild cards" handle spurious image features

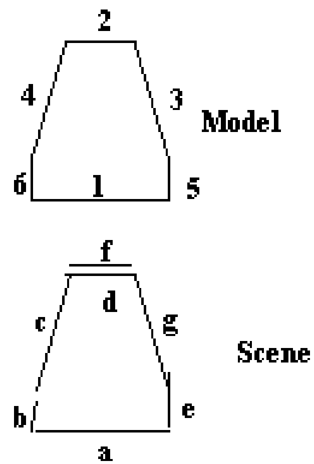


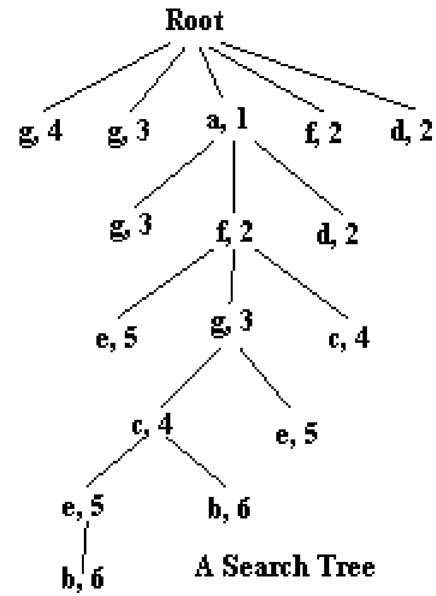




Interpretation Trees







[A.M. Wallace. 1988.]

Interpretation Trees Demo

- http://vision.dai.ed.ac.uk/demos/itreal/

Configuration Search

- Alignment
  - *Model-based RANSAC*
- Pose clustering
  - *Model-based Hough*

These methods search over pose…

another approach computes a measure *invariant* to configuration change.

Pose consistency / Alignment

- Correspondences between image features and model features are not independent.
- A small number of correspondences yields a camera --- the others must be consistent with this.

Pose consistency / Alignment

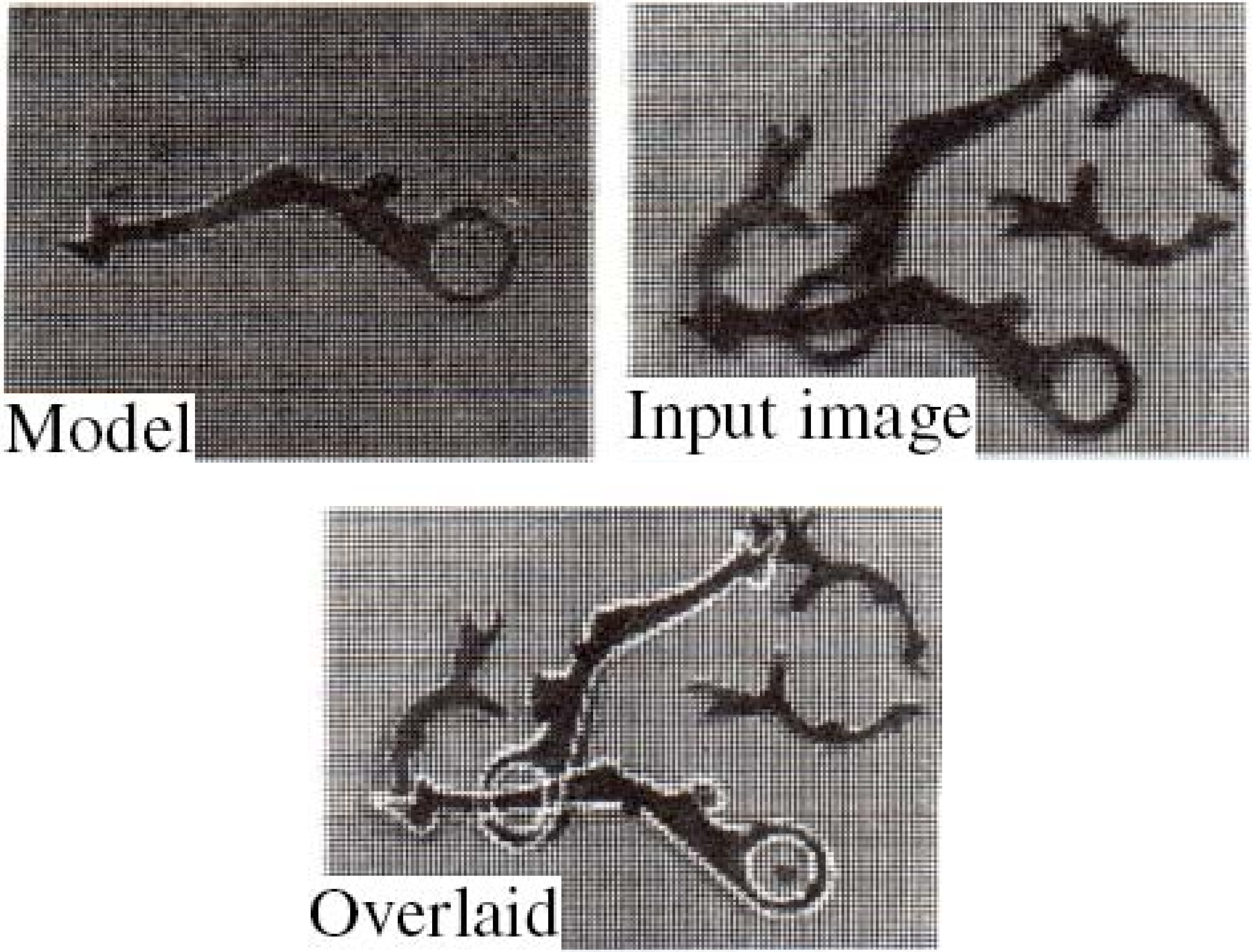
- Strategy:
  - Generate hypotheses using small numbers of correspondences (e.g. triples of points for a calibrated perspective camera, etc., etc.)
  - Backproject and verify
- Notice that the main issue here is camera calibration
- Appropriate groups are "frame groups"

Pose consistency / Alignment

- Given known camera type in some unknown configuration (pose)
- Hypothesize configuration from set of initial features
- Frame group -- set of sufficient correspondences to estimate configuration, e.g.,
  - 3 points
  - 3 directions from 1 point
- Backproject
- Test

Alignment

```
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
```

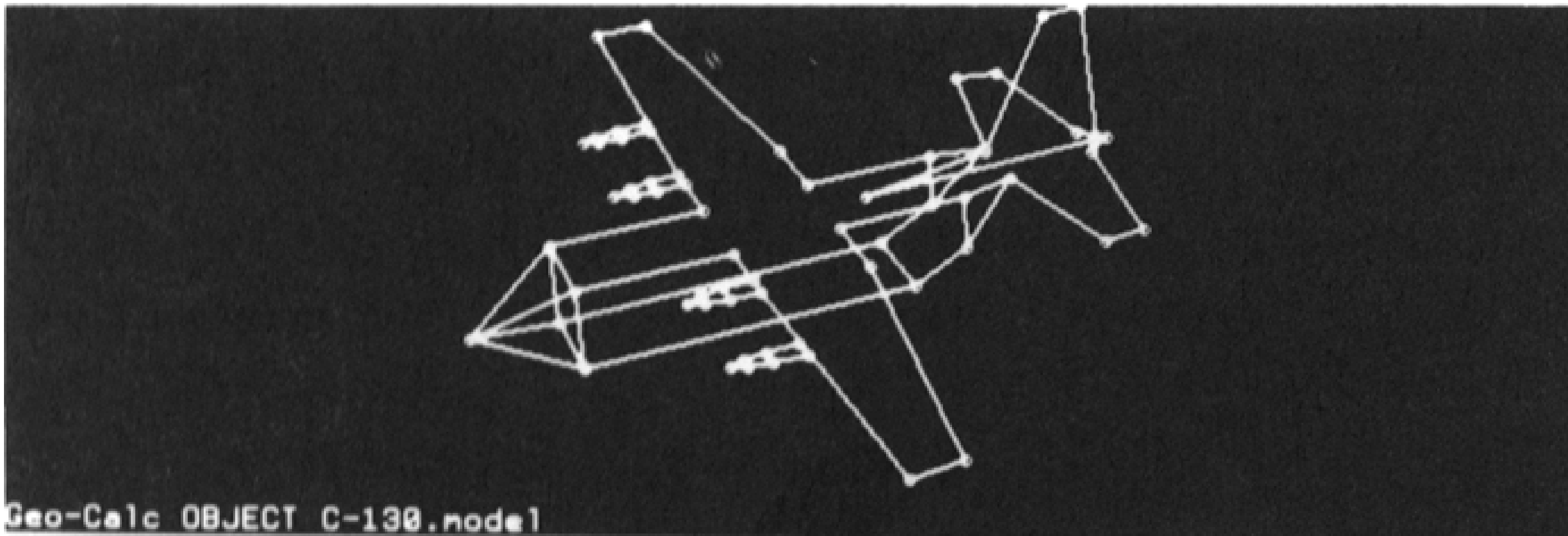

Model
Input image
Overlaid

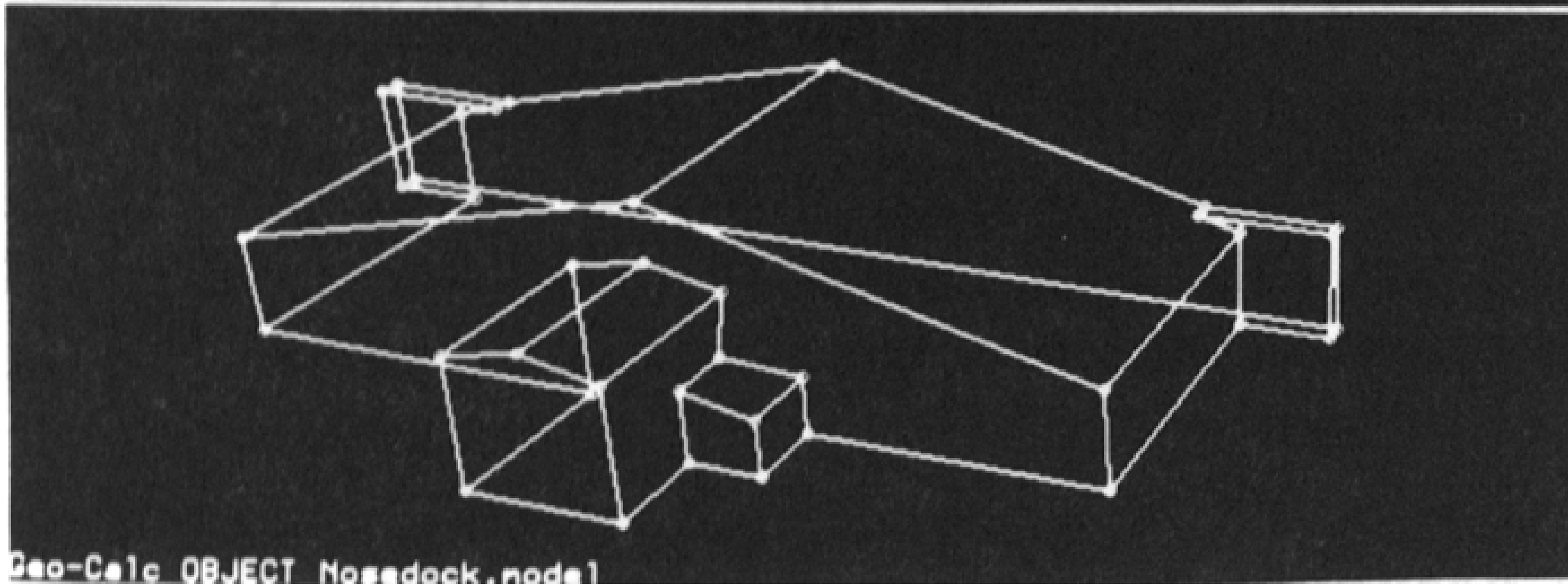
Pose clustering

- 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.

Pose Clustering

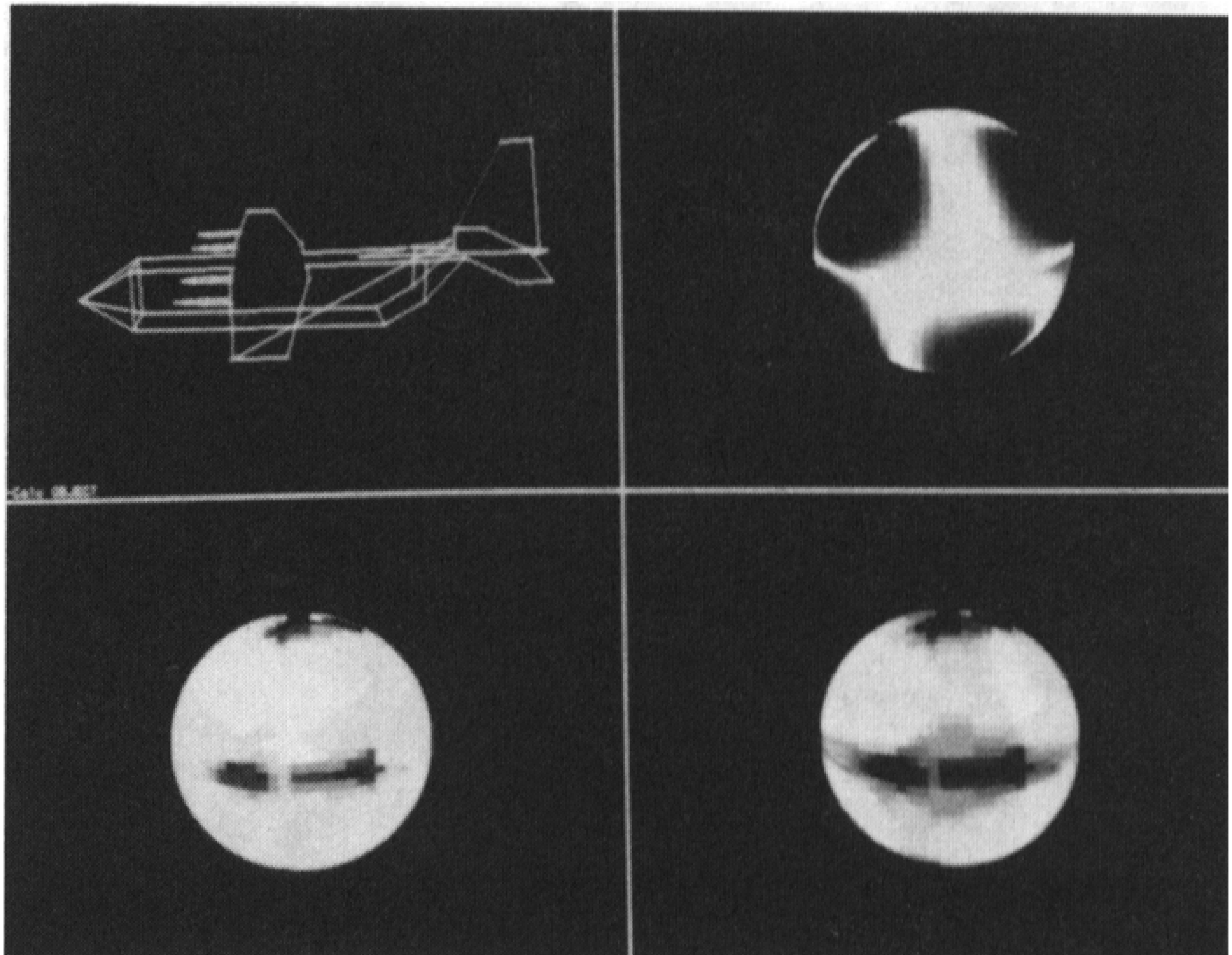
```
For all objects O
  For all object frame groups F(O)
    For all image frame groups F(I)
      For all correspondences C between
        elements of F(I) and elements
        of F(O)

        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
```

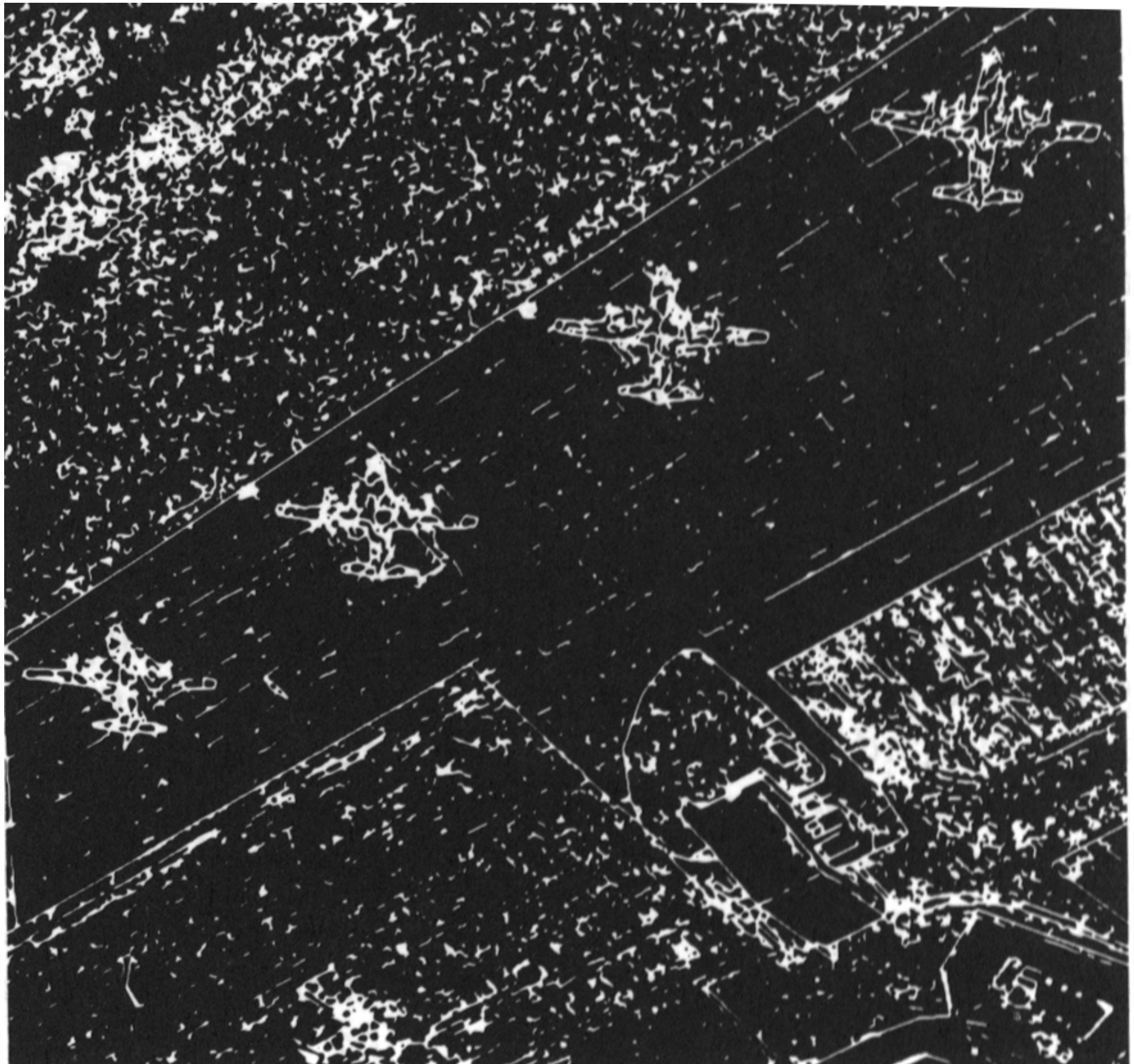

Geo-Calc OBJECT C-130.model

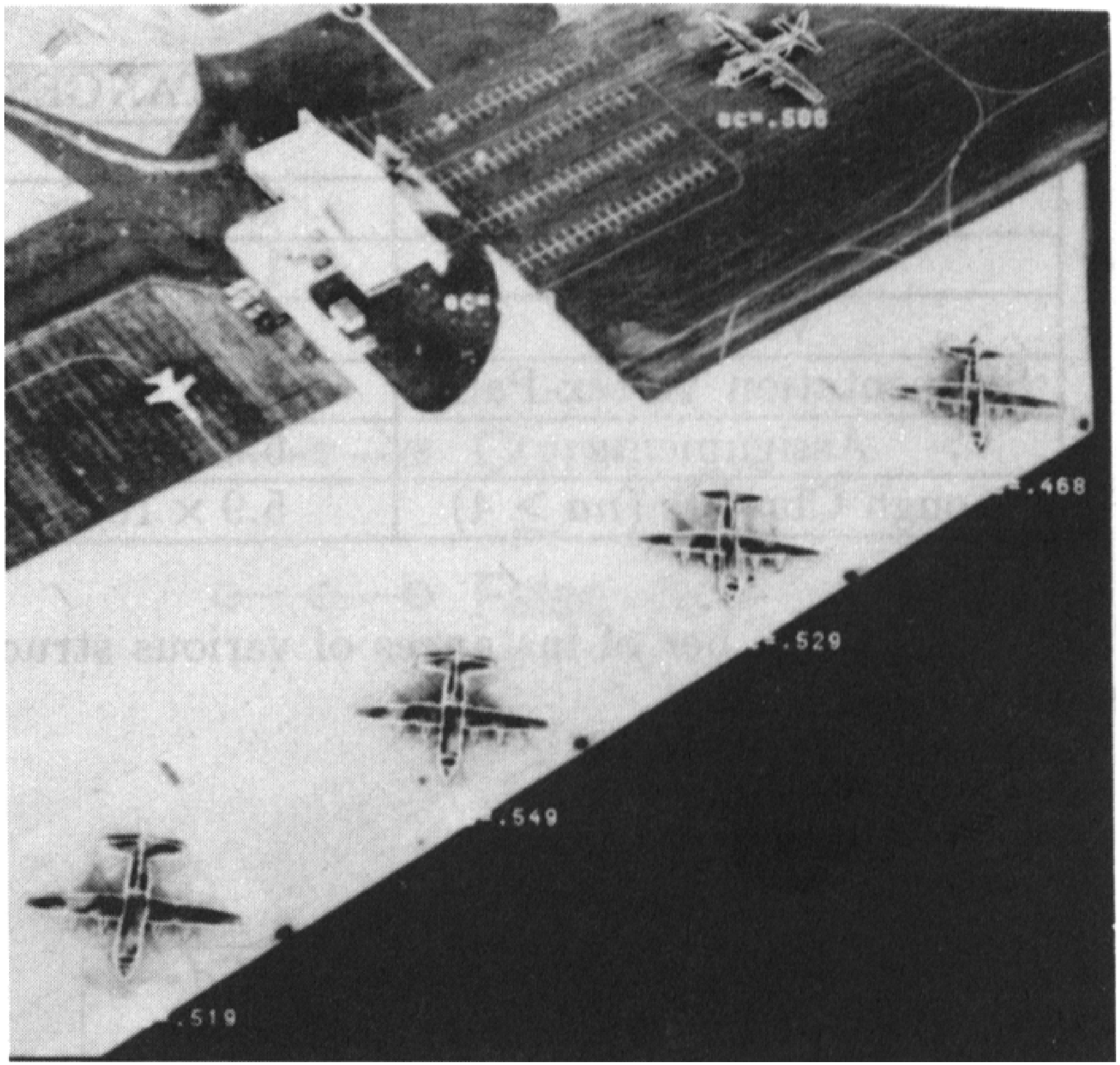

Geo-Calc OBJECT Nosedock.model

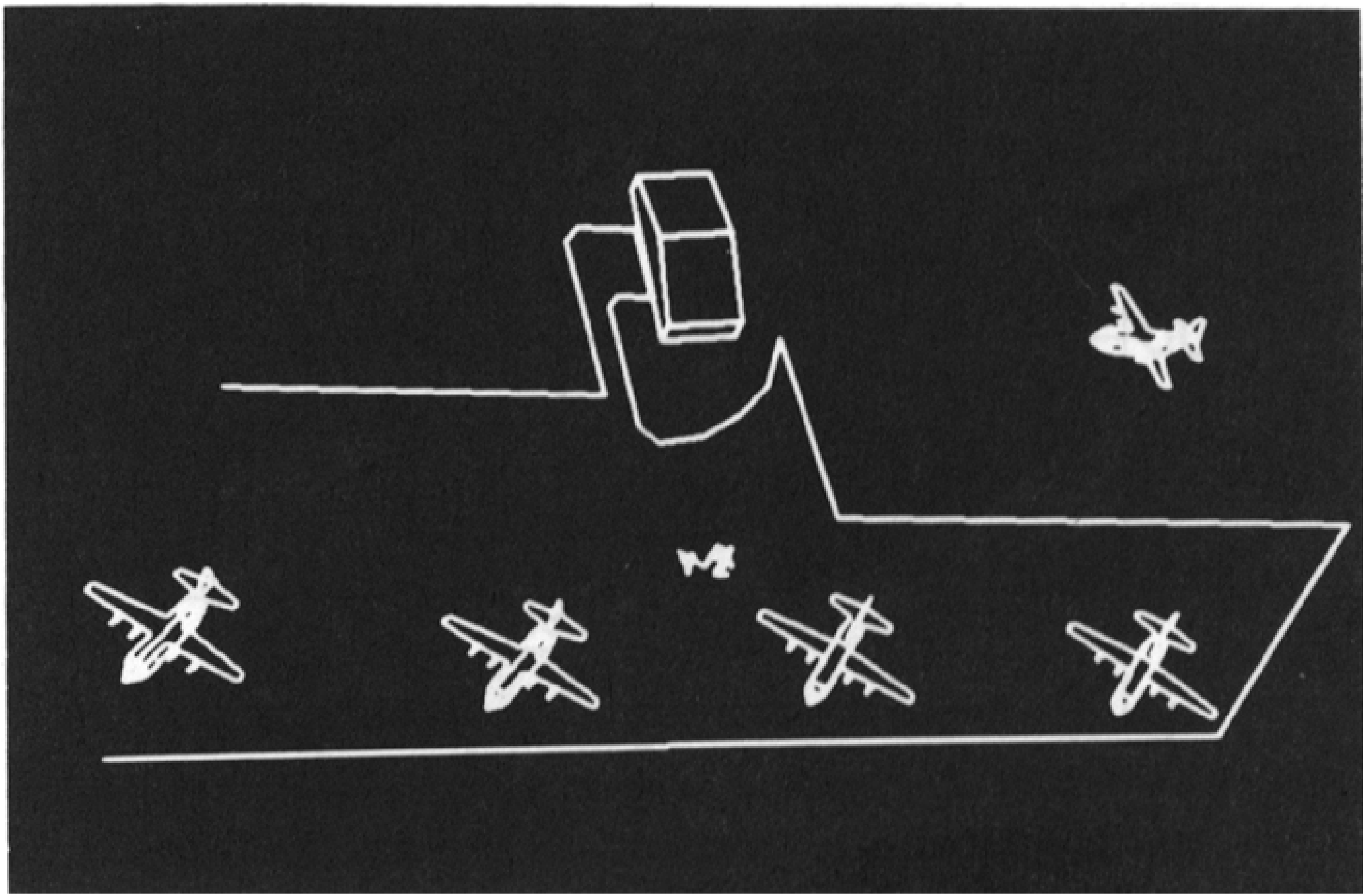
Confidence weighting in Pose clustering

- See where model frame group is reliable (visible!)
- Down-weight / discount votes from frame groups at poses where that frame group is unreliable…






.468
.529
.549
.519



Invariant recognition

- Affine invariants
  - Linear combinations of models
  - Geometric hashing
- Projective invariants
  - Determinant ratio
- Curve invariants

Invariance

- There are geometric properties that are invariant to camera transformations
- Easiest case: view a plane object in scaled orthography.
- Assume we have three base points P_i on the object
  - then any other point on the object can be written as

$$P_k = P_1 + \mu_{ka}(P_2 - P_1) + \mu_{kb}(P_3 - P_1)$$

Invariance

- Now image points are obtained by multiplying by a plane affine transformation, so

$$
\begin{aligned}
p_k &= AP_k \\
&= A\left(P_1 + \mu_{ka}\left(P_2 - P_1\right) + \mu_{kb}\left(P_3 - P_1\right)\right) \\
&= p_1 + \mu_{ka}\left(p_2 - p_1\right) + \mu_{kb}\left(p_3 - p_1\right)
\end{aligned}
$$

Invariance

$$P_k = P_1 + \mu_{ka}\left(P_2 - P_1\right) + \mu_{kb}\left(P_3 - P_1\right)$$

$$\begin{aligned} p_k &= AP_k \\ &= A\left(P_1 + \mu_{ka}\left(P_2 - P_1\right) + \mu_{kb}\left(P_3 - P_1\right)\right) \\ &= p_1 + \mu_{ka}\left(p_2 - p_1\right) + \mu_{kb}\left(p_3 - p_1\right) \end{aligned}$$

- This means that, if I know the base points in the image, I can read off the μ values for the object
  - they're the same in object and in image --- **invariant**
- Suggests a strategy rather like the Hough transform
  - search correspondences, form μ's and vote

Geometric hashing

- Vote on identity and correspondence using invariants
  - Take hypotheses with large enough votes
- Fill up a table, indexed by μ's, with
  - the base points and fourth point that yield those μ's
  - the object identity

Algorithm 18.3: Geometric hashing: voting on identity and point labels

```
For all groups of three image points T(I)
   For every other image point p
      Compute the μ's from p and T(I)
      Obtain the table entry at these values
         if there is one, it will label the three points in T(I)
         with the name of the object
         and the names of these particular points.
      Cluster these labels;
         if there are enough labels, backproject and verify
      end
   end
end
```

Indexing with invariants

- Voting in geometric hashing is superfluous - we could just go ahead and verify if we get a hit.
- It would be nice to have invariants for perspective cameras
- Groups of features with identity information invariant to pose – *invariant bearing groups*
- Easy for perspective views of plane objects --- we write object points in homogenous coordinates, then the object coordinates are multiplied by a 3x3 matrix with non-zero det.

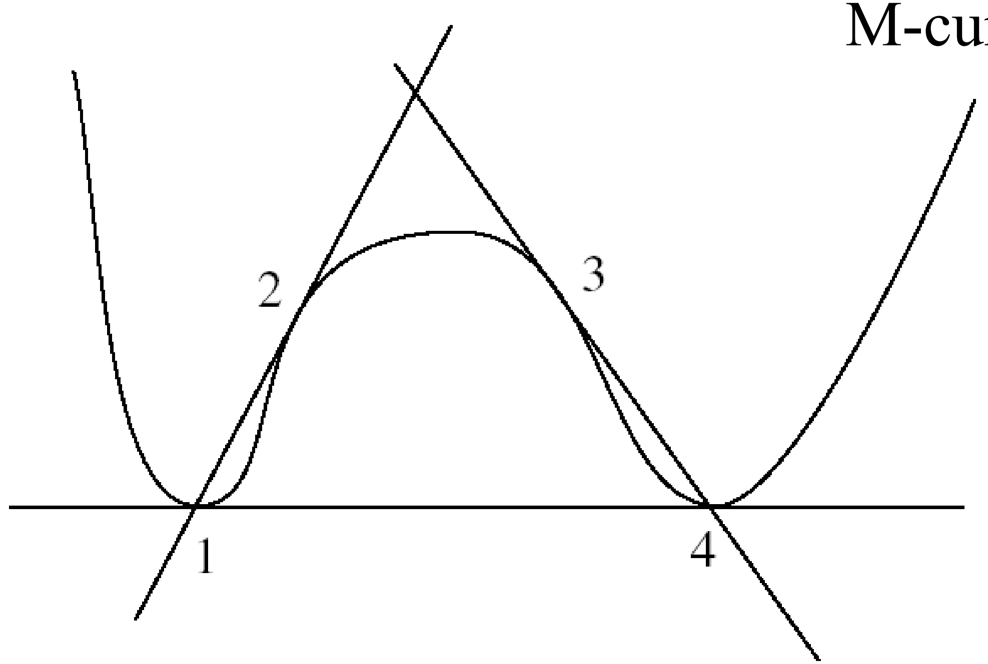
$$
\begin{aligned}
\frac{\det\left(\left[p_i p_j p_k\right]\right)\det\left(\left[p_i p_l p_m\right]\right)}{\det\left(\left[p_i p_j p_l\right]\right)\det\left(\left[p_i p_k p_m\right]\right)}
&= \frac{\det\left(\left[MP_i MP_j MP_k\right]\right)\det\left(\left[MP_i MP_l MP_m\right]\right)}{\det\left(\left[MP_i MP_j MP_l\right]\right)\det\left(\left[P_i MP_k MP_m\right]\right)} \\
&= \frac{\det\left(M\left[P_i P_j P_k\right]\right)\det\left(M\left[P_i P_l P_m\right]\right)}{\det\left(M\left[P_i P_j P_l\right]\right)\det\left(M\left[P_i P_k P_m\right]\right)} \\
&= \frac{\left(\det(M)^2\right)}{\left(\det(M)^2\right)}\frac{\det\left(\left[P_i P_j P_k\right]\right)\det\left(\left[P_i P_l P_m\right]\right)}{\det\left(\left[P_i P_j P_l\right]\right)\det\left(\left[P_i P_k P_m\right]\right)} \\
&= \frac{\det\left(\left[P_i P_j P_k\right]\right)\det\left(\left[P_i P_l P_m\right]\right)}{\det\left(\left[P_i P_j P_l\right]\right)\det\left(\left[P_i P_k P_m\right]\right)}
\end{aligned}
$$

Five points under projective transformations; the text gives several other constructions

Tangent invariance

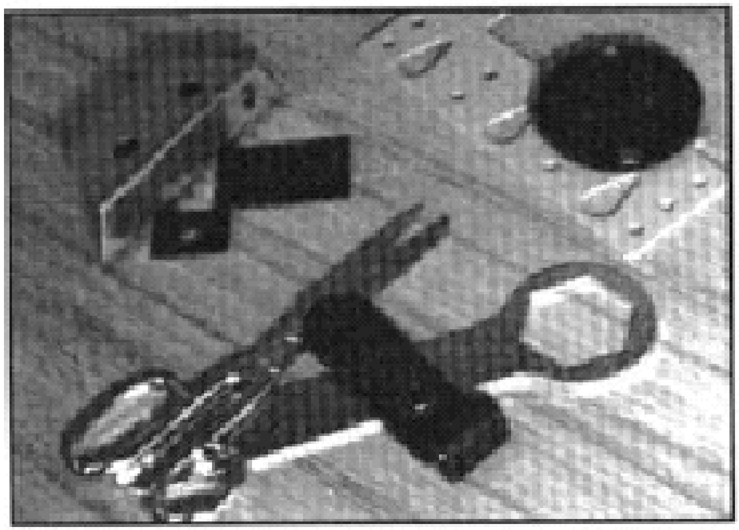
- Incidence is preserved despite transformation

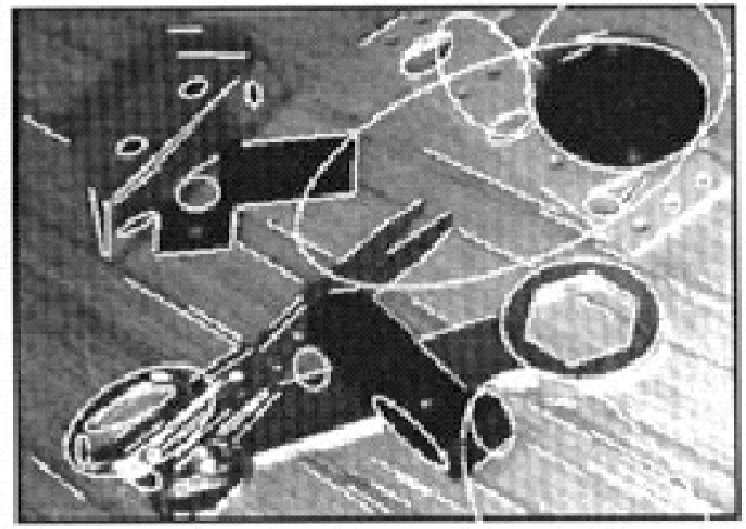
M-curve construction



- Transform four points above to unit square: measurements in this canonical frame will be invariant to pose.

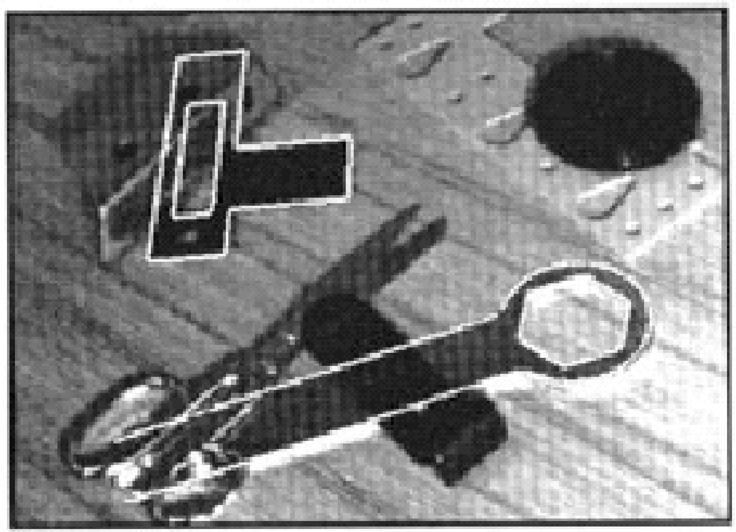
```
For each type T of invariant-bearing group
  For each image group G of type T

  Determine the values V of the invariants of G

    For each model feature group M of type T whose invariants
    have the values V

      Determine the transformation that takes M to G

      Render the model using this transformation

      Compare the result with the image, and accept if
      similar
    end
  end
end
```





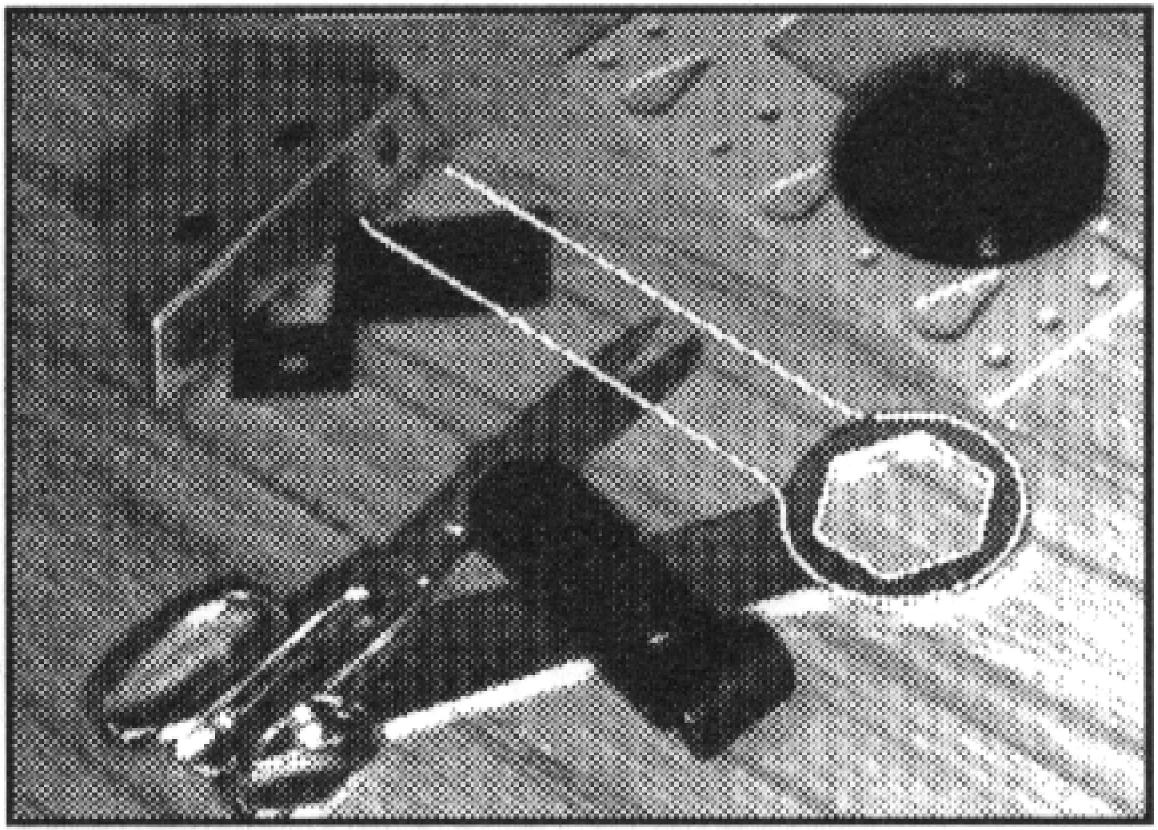
a

b



Verification

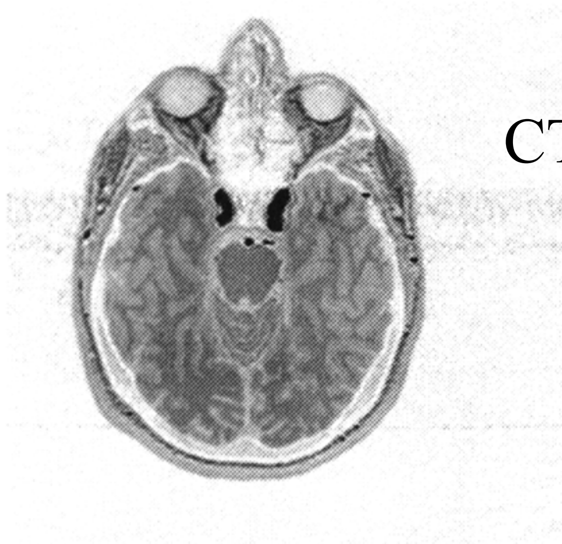
- Edge score
  - are there image edges near predicted object edges?
  - very unreliable; in texture, answer is usually yes
- Oriented edge score
  - are there image edges near predicted object edges with the right orientation?
  - better, but still hard to do well (see next slide)
- No-one's used texture
  - e.g. does the spanner have the same texture as the wood?
- model selection problem
  - more on these later; no-ones seen verification this way, though



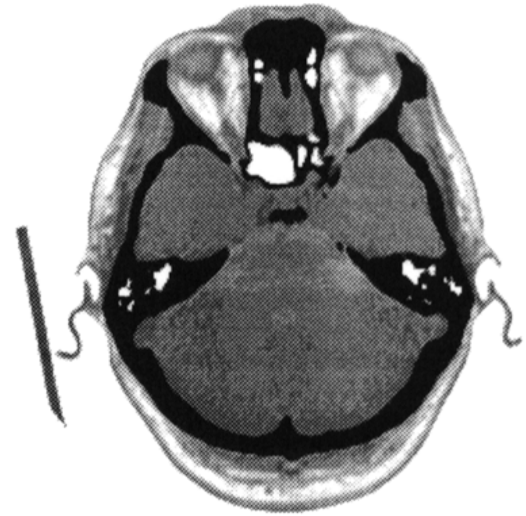
Application: Surgery

- To minimize damage by operation planning
- To reduce number of operations by planning surgery
- To remove only affected tissue
- Problem
  - ensure that the model with the operations planned on it and the information about the affected tissue lines up with the patient
  - display model information supervised on view of patient
  - **Big Issue**: coordinate alignment, as above

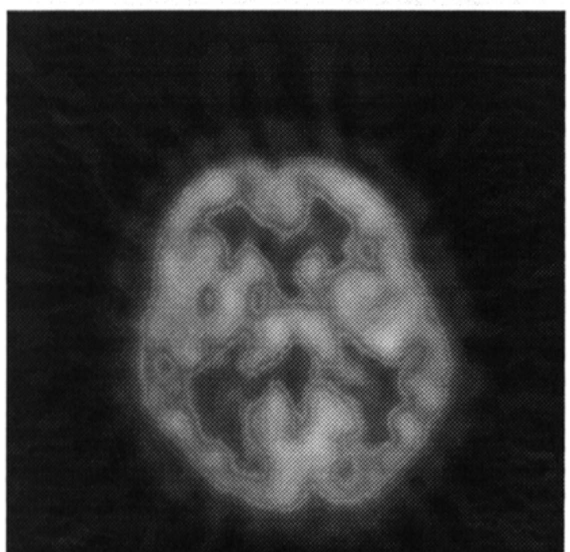
MRI

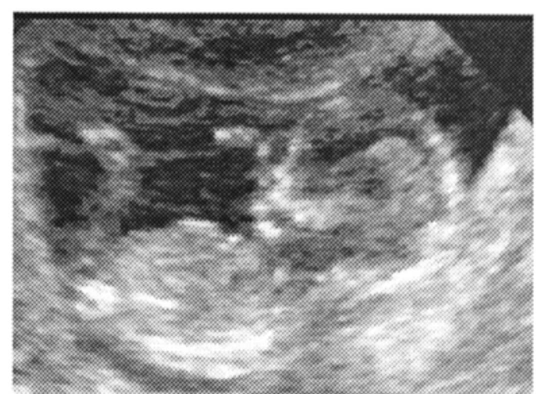


CTI

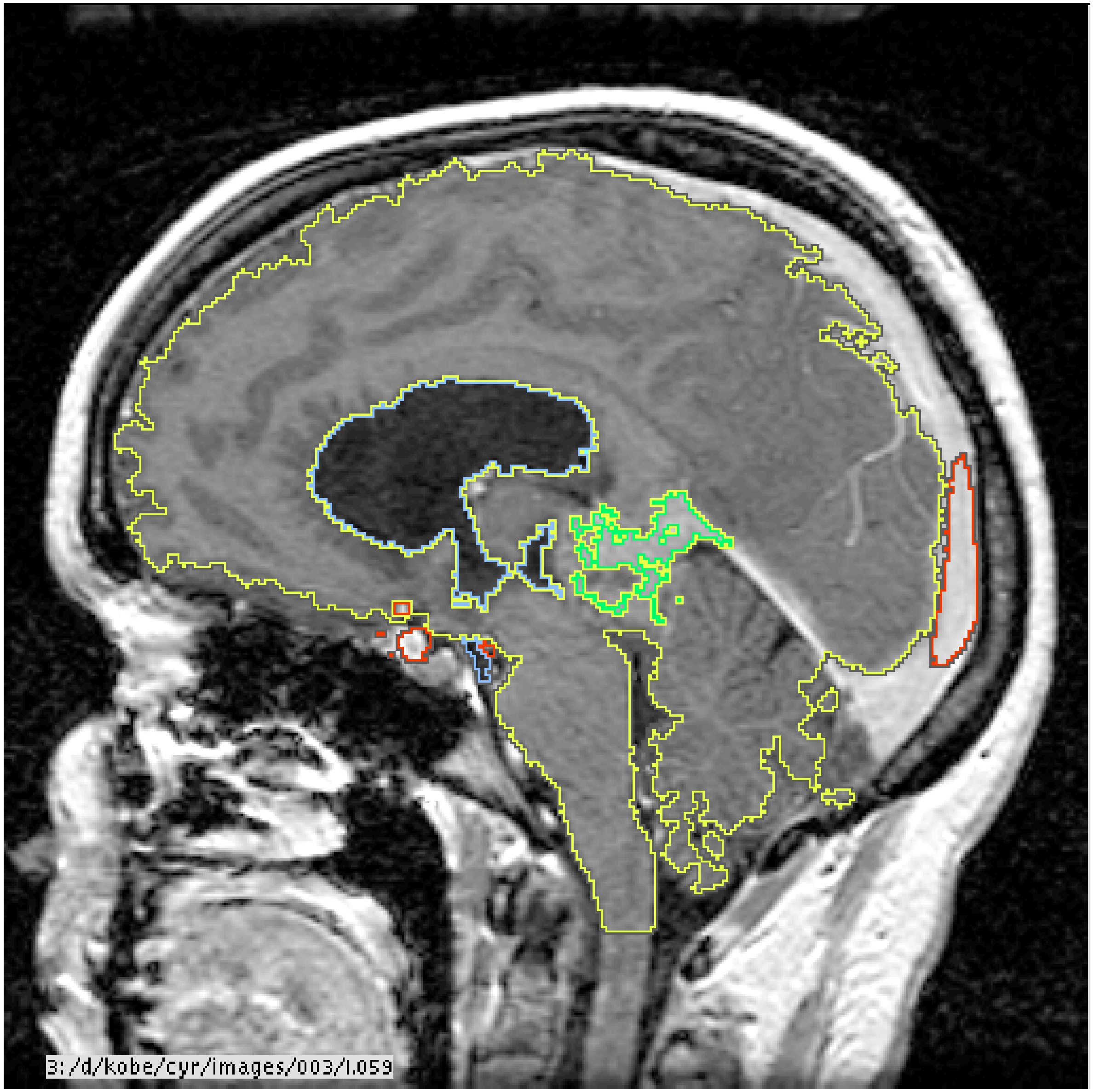


NMI

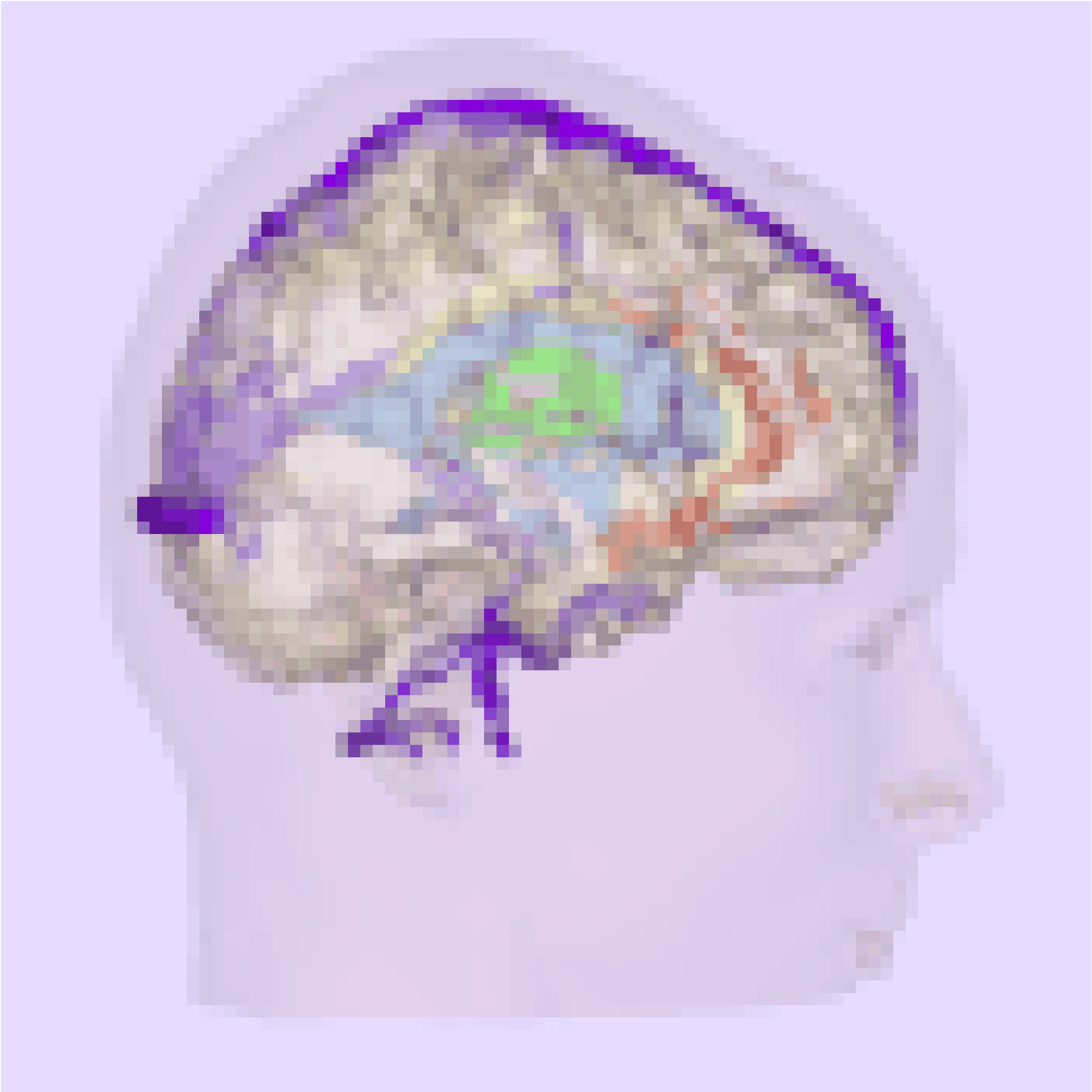




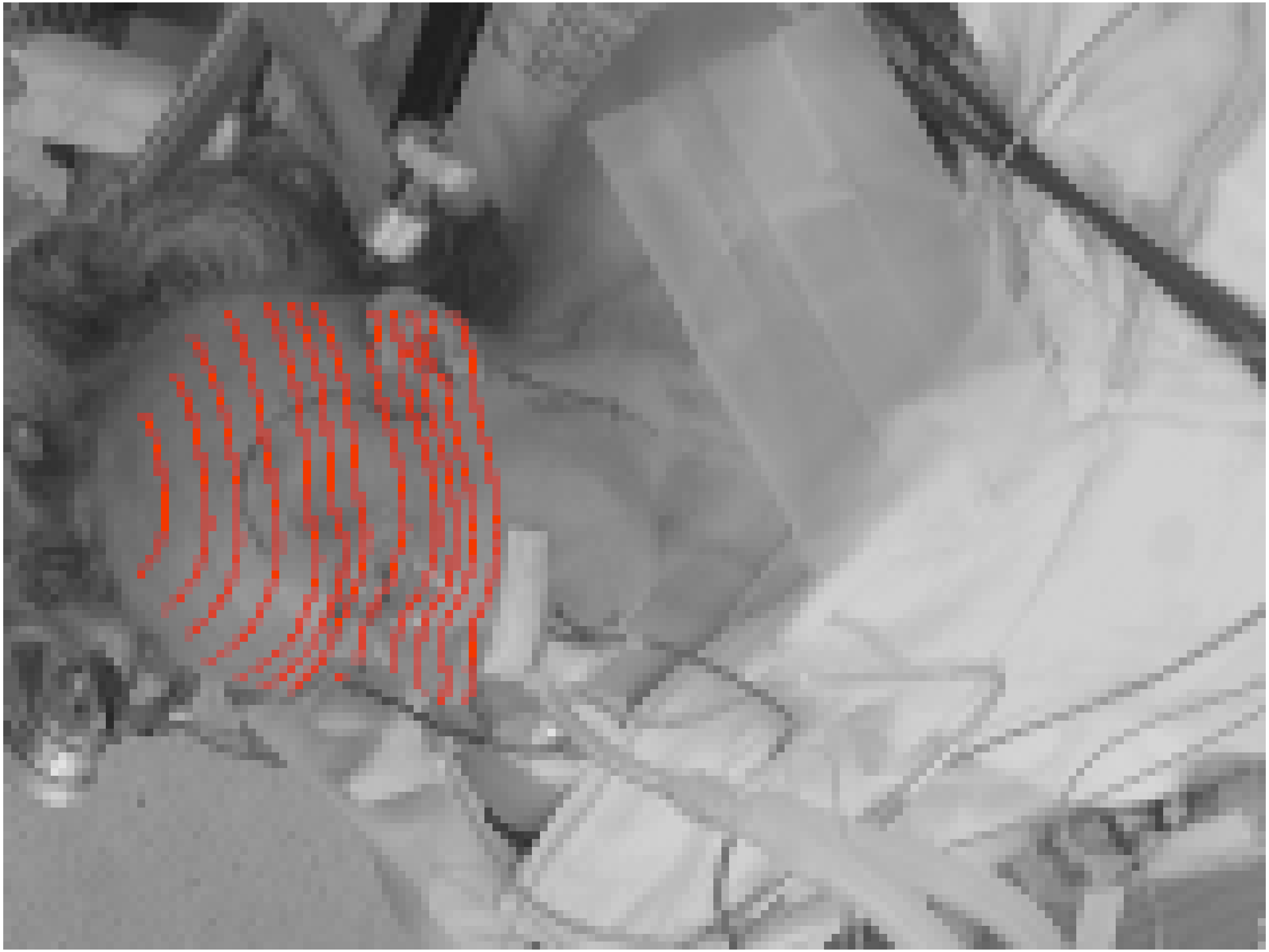
USI



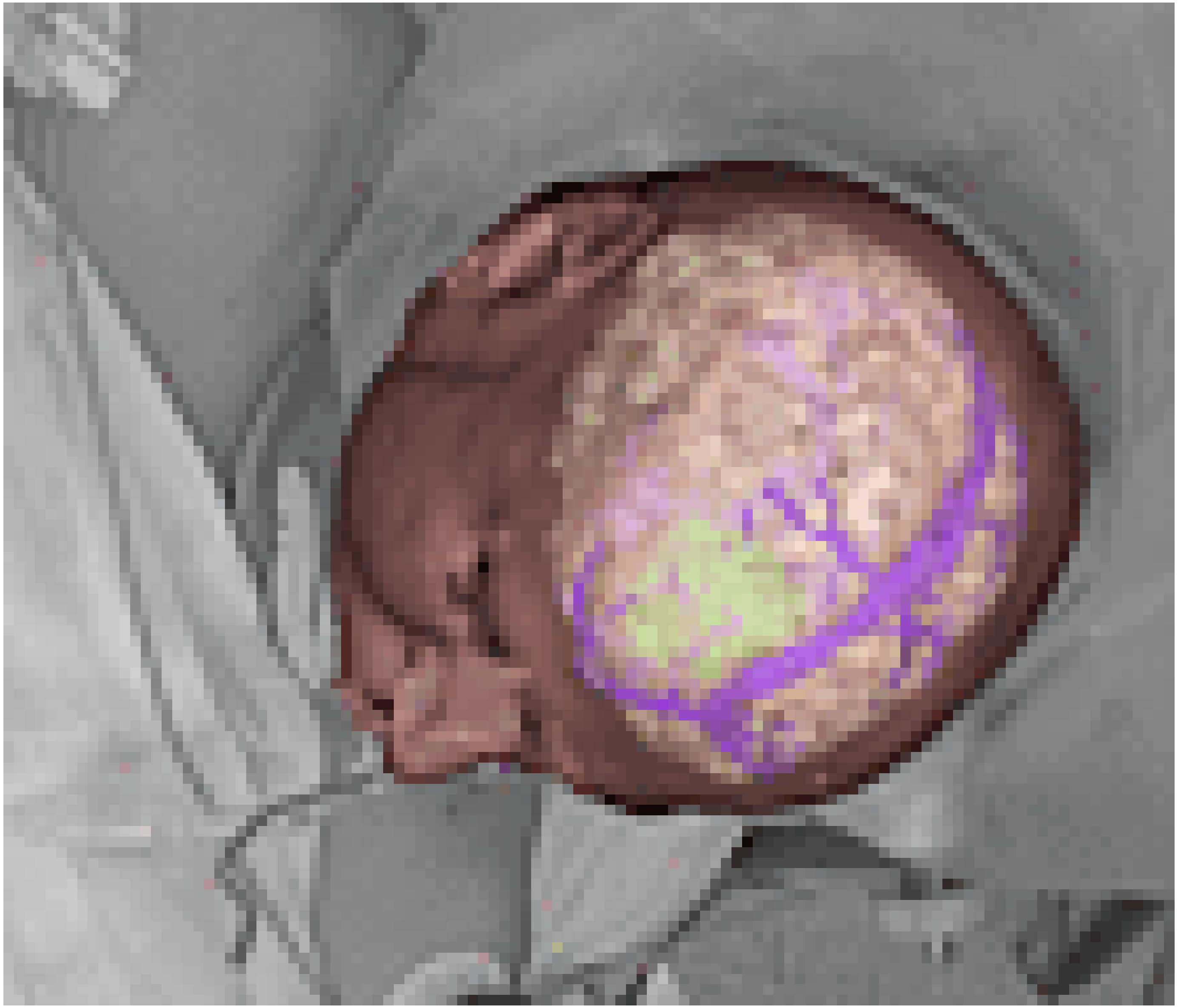
[Eric Grimson; http://www.ai.mit.edu/people/welg/welg.html]



[Eric Grimson; http://www.ai.mit.edu/people/welg/welg.html]



[Eric Grimson; http://www.ai.mit.edu/people/welg/welg.html]



[Eric Grimson; http://www.ai.mit.edu/people/welg/welg.html]

[Eric Grimson; http://www.ai.mit.edu/people/welg/welg.html]

Model-based Vision

- Hypothesize and test
- Interpretation Trees
- Alignment
- Pose Clustering
- Invariances
- Geometric Hashing
- Medical Imaging Application

[Figures from Forsythe & Ponce unless otherwise attributed]