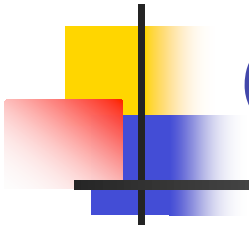
Detecting Shapes in Raven's Matrices

By: Jacqueline Mok

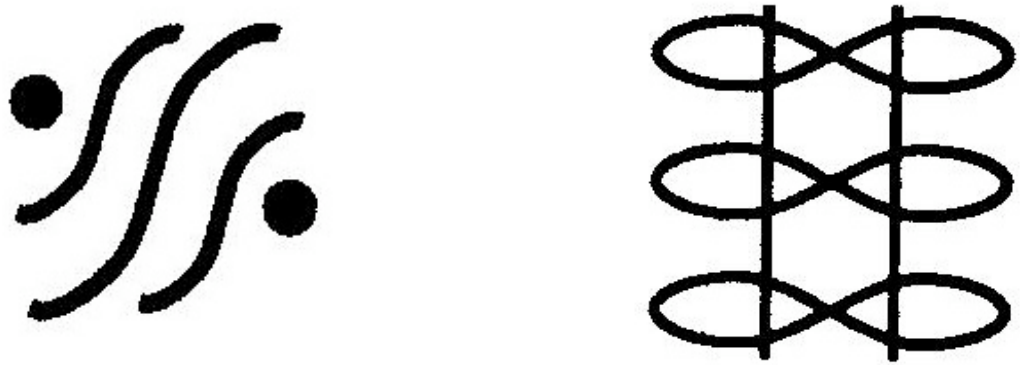


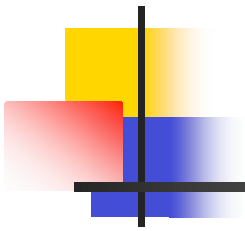
Outline

- Motivation
- Problem
- Techniques used
  - SIFT
  - Hough transform
  - Affine fitting
  - Clustering
- Adaptation to Rasmussen Model
- Conclusions
  - Future problems

Motivation

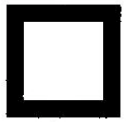
- Learn to recognize objects



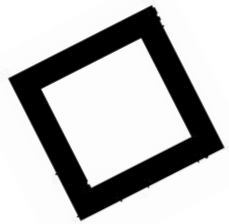


Motivation

- Computers are not THAT intelligent



= square

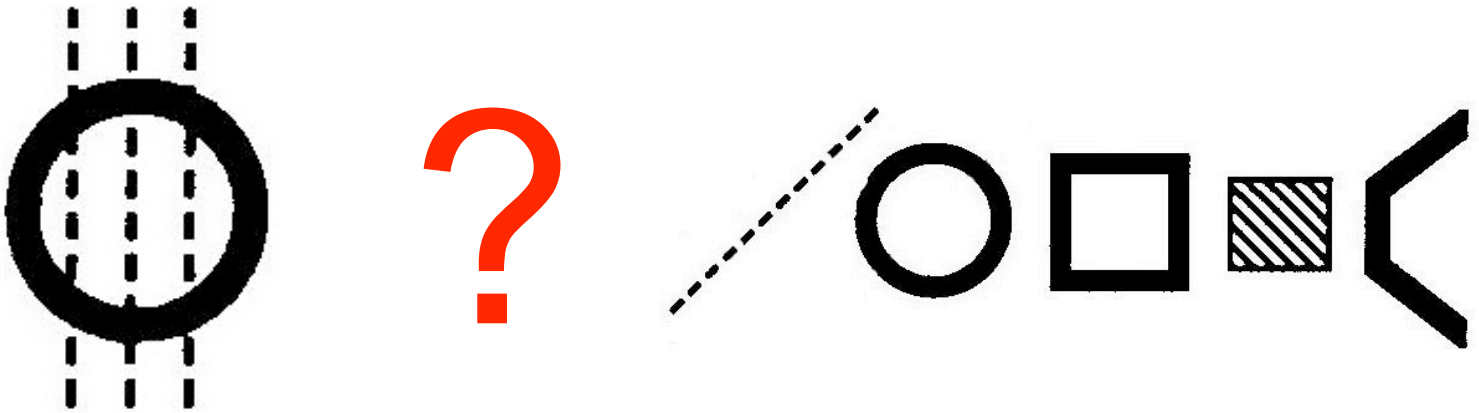


!= square

= object2

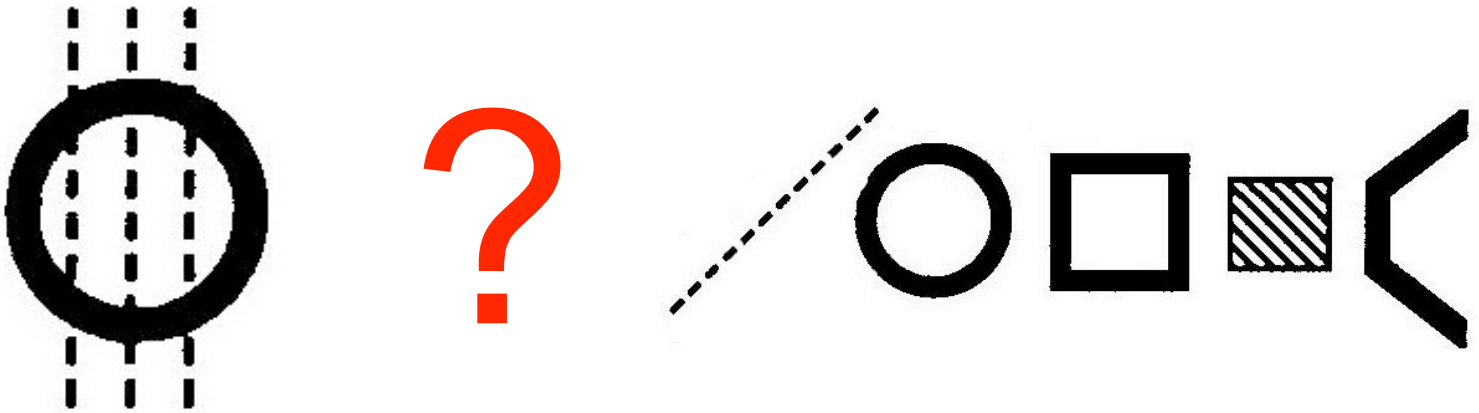
Problem

- Given a test image, is it possible to determine whether or not a model exists in the test image? How many times?



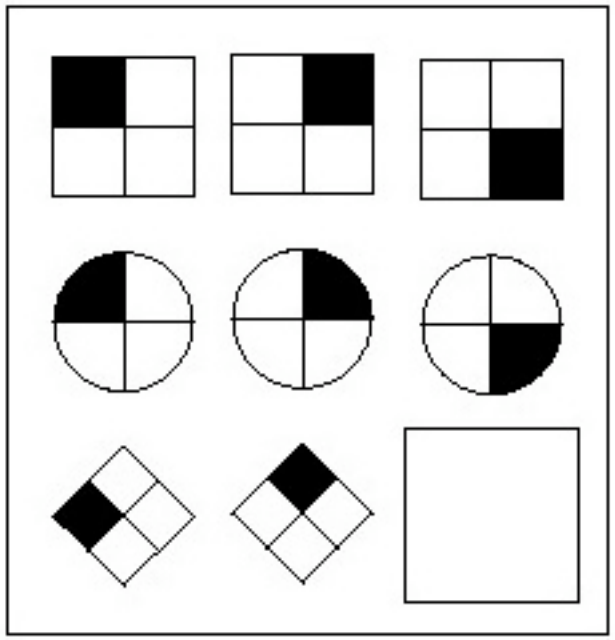
Problem

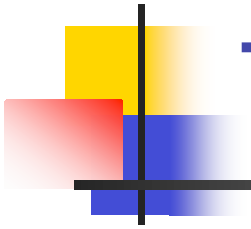
- What is the best (x,y) location, scale and orientation from a model to a given test image?



Problem

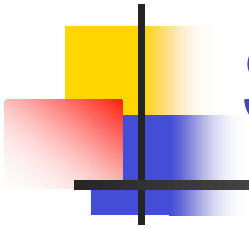
- Used Raven's Matrices





Techniques used

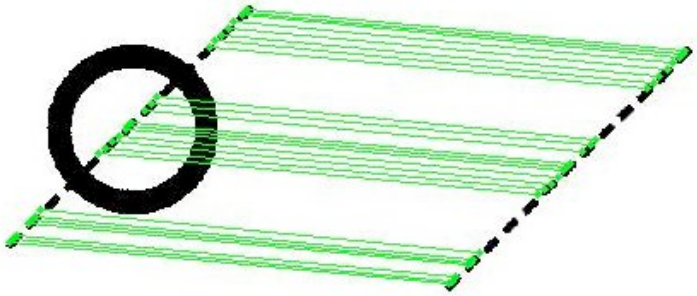
- SIFT (Lowe, 2004)
- Hough Transform
- Affine fitting
- Clustering



SIFT

- Scale Invariant Feature Transform
- David Lowe, 2004
- Match keypoints between test and model
- Provides location, scale, orientation of keypoints

SIFT

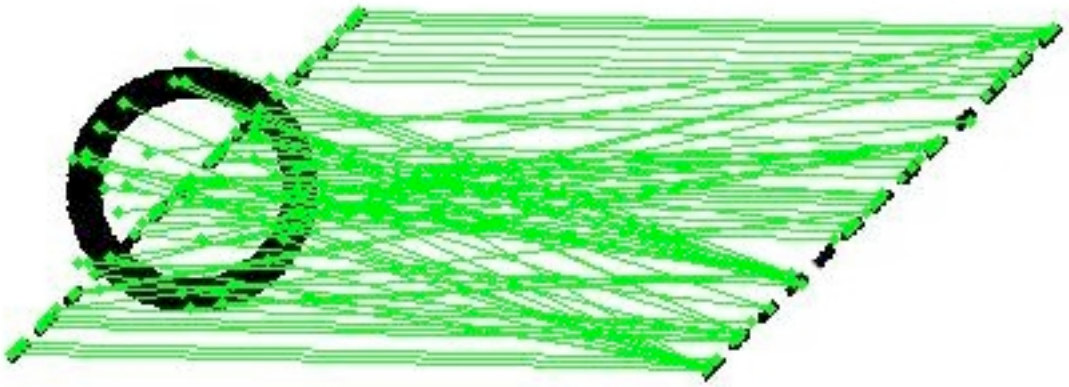


siftmatch(),
plotsiftmatches()

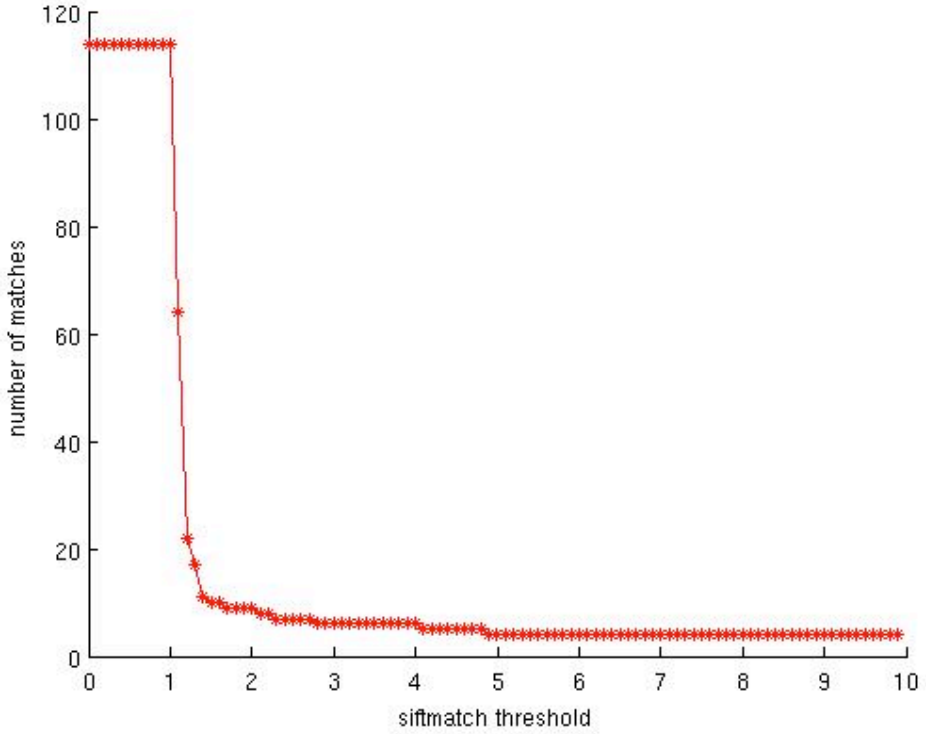
- Andrea Vedaldi, SIFT for Matlab
- sift()
  - Frames: 4xM matrix
  - Descriptors:128-D vector

SIFT

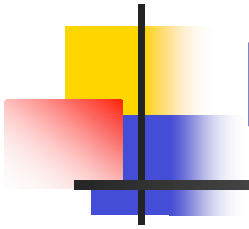
```
[frames_test, test_desc] = sift(test, 'Threshold', THRESHOLD, ...
    'EdgeThreshold', EDGE_THRESHOLD, 'BoundaryPoint', 0);

matches = siftmatch(test_desc, model_desc, SIFTMATCH_THRES);

plotmatches(test, model, frames_test, frames_model, matches);
```



SIFT



Siftmatch()

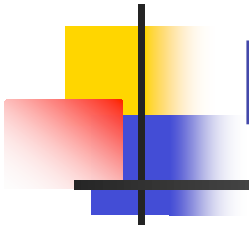


Hough Transform

- Feature extraction technique
- Image analysis, computer vision, digital image processing
- Voting procedure

Hough Transform

- Scene A: knife, blood, person on floor
  - Knife: kitchen, crime, camping
  - Blood: hospital, nose bleed, crime
  - Person on floor: sleeping, star gazing, crime
- Winner: Crime



Hough Transform

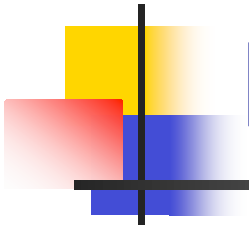
- Input: keypoint matches
- Create bins in Hough space
  - (x,y) location, scale, orientation
- Output:
  - ‘winning’ bin (most votes)
  - No winner (model DNE)
- Multiple winners

Hough Transform

$$\Delta scale = \frac{test\_scale}{model\_scale}$$

$$\Delta\theta = test\_orientation - model\_orientation$$

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} \Delta scale \times cos(\Delta\theta) & -sin(\Delta\theta) & x_t \\ sin(\Delta\theta) & \Delta scale \times cos(\Delta\theta) & y_t \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} -x_m + \frac{model\_width}{2} \\ -y_m + \frac{model\_height}{2} \\ 1 \end{bmatrix}$$



Hough Transform

- Vote 2 closest bins in each dimension
- Total 16 bins
- Hash table, serialize indices
- Track winning bin

Affine Fitting

$$\begin{bmatrix} x & y & 0 & 0 & 1 & 0 \\ 0 & 0 & x & y & 0 & 1 \\ & & \cdots & & & \\ & & \cdots & & & \end{bmatrix} \times \begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \\ l_x \\ l_y \end{bmatrix} = \begin{bmatrix} u \\ v \\ \vdots \end{bmatrix}, m = \begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \\ l_x \\ l_y \end{bmatrix}$$

(Lowe, 2004)

- At least 3 distinct model points mapped to test image

Affine Fitting

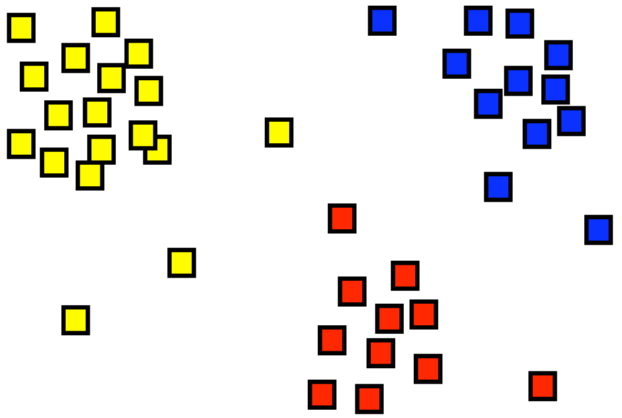
Test image

Model

```
dashed_line2.jpg located at approximately (156.8097,156.8097)

scale: 1.0062 times the model

orientation: 3.1173 radians
```

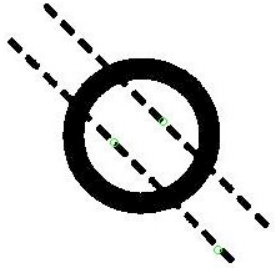
Clustering

- Unsupervised classification of patterns (observations, data, feature vectors) into groups (clusters) (Jain et al., 1999)



Clustering

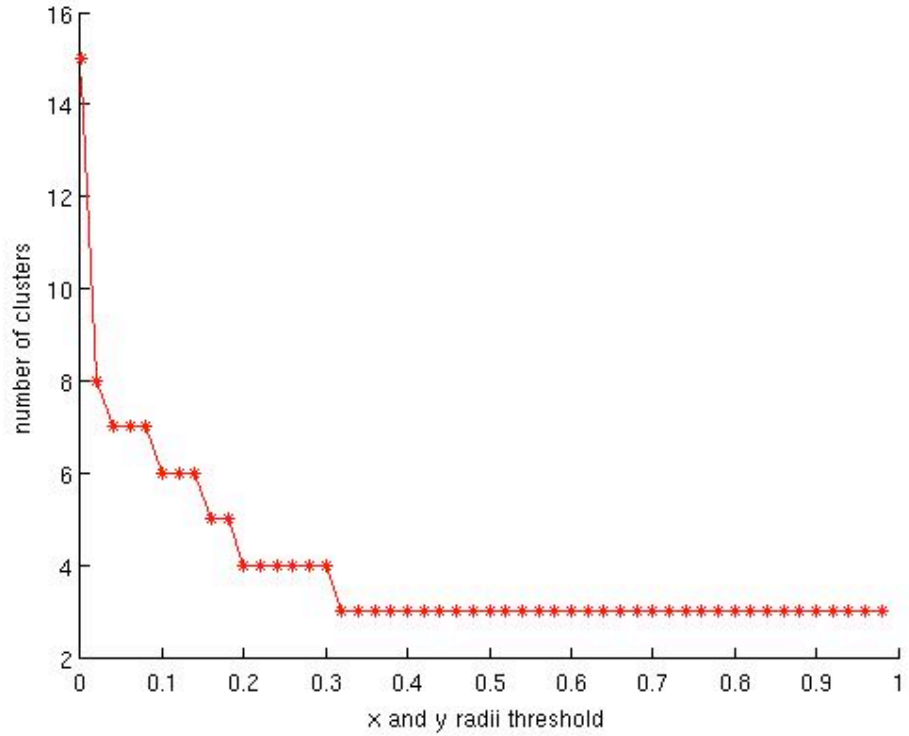
- Track 30 bins with most votes



- If first bin has match between test and model, proceed with clustering
- Matlab's subclust()

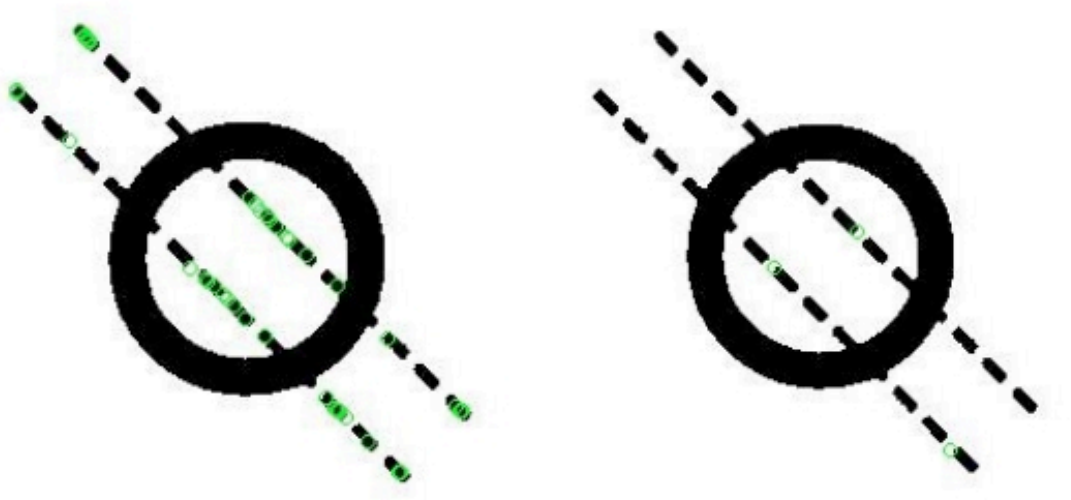
```
[C S] = subclust(X, [xy_radii xy_radii 0.4 0.3 0.3], Xbounds);
```

Clustering



Plot of x and y radii versus number of clusters

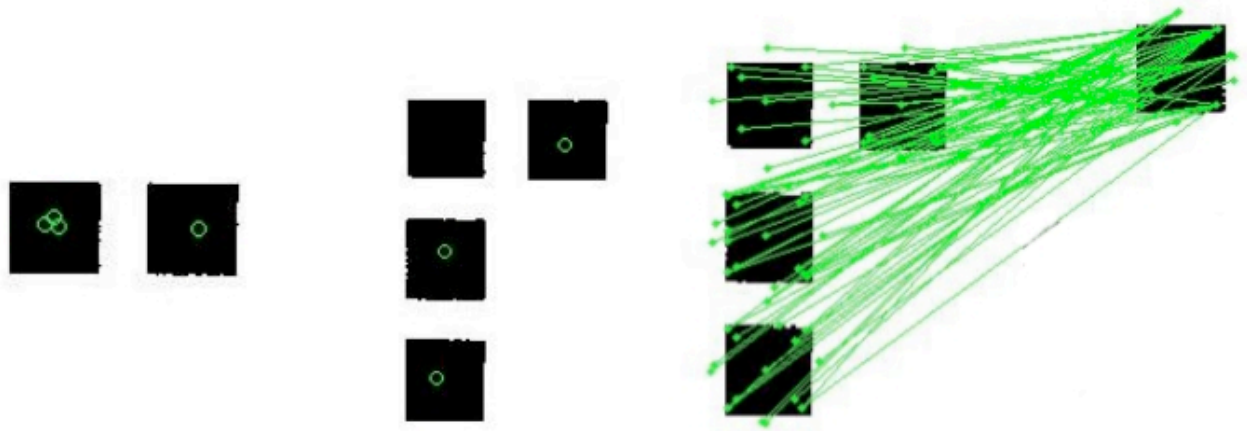
Clustering



(a) Plot of all model points from a dashed line to the image points

(b) Sample showing 2 clusters found in the image (green dots in the center)

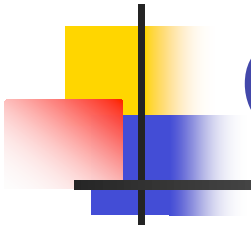
Clustering



(a) Three clusters found in the left square

(b) Top left cluster missing

(c) siftmatch() results between Fig. 8b and the model



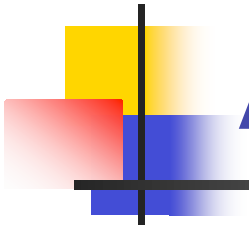
Clustering

There are clusters located at (format: [a b c d e]):

156.8772	166.8502	1.0053	-0.9998	0.0211
196.0512	149.6497	1.0226	0.0021	-1.0000
240.3024	252.6831	0.8730	0.9999	-0.0109

where a and b represent the x-y location, c represents the relative scale from the model to the test image, and d and e represent the angle on a unit circle.

(cos(theta), sin(theta)) = (d,e)

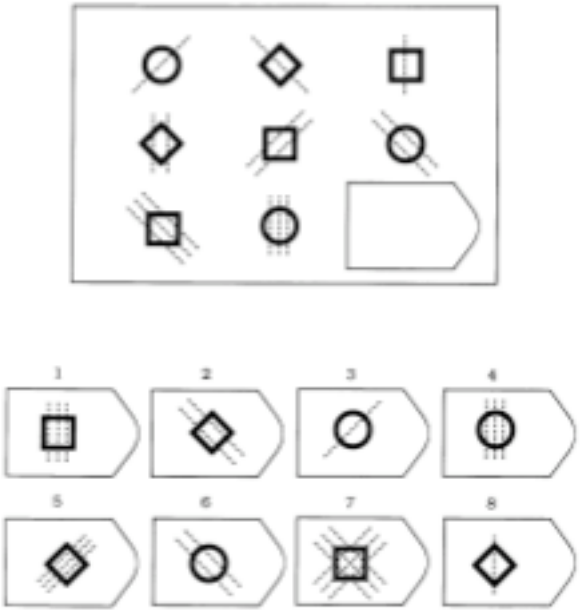


Adaptation to Rasmussen Model

- Basic Idea
  - Apply SIFT/Hough transform on the 8 cells
  - map Rasmussen input with above output
  - If match
    - apply SIFT/Hough transform on answer cells
- theoretical

Adaptation to Rasmussen Model

- Example 1



(a) Raven Matrix 1

(b) Databank of figures

Adaptation to Rasmussen Model

- Cell 1's output from Hough

```
HOUGH IMPLEMENTATION
================================
dashed_line_horizontal.jpg located at approximately (173.7823,173.7823)
scale: 0.99973 times the model
orientation: 5.4982 radians

There are clusters located at (format: [a b c d e]):
  173.6544  157.8504    1.0045    0.7079   -0.7063

where a and b represent the x-y location, c represents
the relative scale from the model to the test image,
and d and e represent the angle on a unit circle.
(cos(theta), sin(theta)) = (d,e)
```

Adaptation to Rasmussen Model

- Cell 1's output from Hough

```
HOUGH IMPLEMENTATION
================================
circle.jpg located at approximately (179.9844,179.9844)
scale: 1.0154 times the model
orientation: 6.2147 radians

There are clusters located at (format: [a b c d e]):
  180.5672  155.7880    1.0035    0.5597   -0.8287
where a and b ...

HOUGH IMPLEMENTATION
================================
no match for square.jpg
```

Adaptation to Rasmussen Model

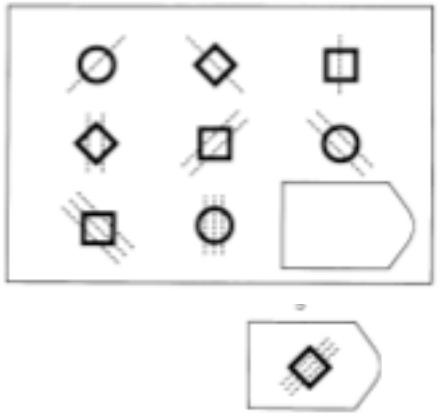
- Modified version of Dan's input

```
#matrix
circle; 1; 45deg
diamond; 1; 135deg
square; 1; 90deg
diamond; 2; 90deg
square; 2; 45deg
circle; 2; 135deg
square; 3; 135deg
circle; 3; 90deg
```

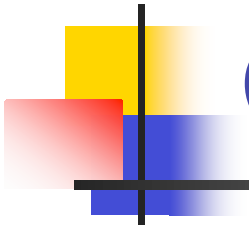
```
#answers
square; 3; 90deg
diamond; 2; 135deg
circle; 1; 45deg
circle; 3; 90deg
diamond; 3; 45deg
circle; 2; 135deg
square; 3; 45deg
diamond; 1; 90deg
```

Adaptation to Rasmussen Model

- logic



```
if cell 1 contains a circle and 1 dashed line at 45deg &&
   cell 2 contains a diamond and 1 dashed line at 135deg &&
   cell 3 contains a square and 1 dashed line at 90deg &&
   ... &&
   cell 8 contains a square and 3 dashed line at 90deg
then
   apply the Hough transform on the answer cells
   if Answer 1 contains a diamond and 3 dashed lines at 45deg
      then output Answer 1
   else if Answer 2 contains a diamond and 3 dashed lines at 45deg
      then output Answer 2
   ...
   else if Answer 8 contains a diamond and 3 dashed lines at 45deg
      then output Answer 8
end
```

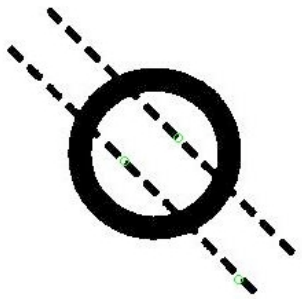


Conclusions

- Is it possible to determine the existence of a model in a given test image?
- YES!!!
  - SIFT, Hough transform
  - Clustering
- Theoretically adapt above methods to Rasmussen Model

Future Problems

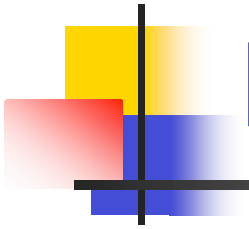
- SIFT has limitations
  - Occlusion
  - Noise



  - Non-detectable features (pixel-based)

Future Problems

- Clustering solutions
  - Similarity transform
  - Prior knowledge
  - First bin approach
  - Adjust more parameters
- Training images



References

Ballard, D. H. (1981). Generalizing the Hough Transform to Detect Arbitrary Shapes. *Pattern Recognition*, 13(2), 111.

Forsyth, D. A., & Ponce, J. (2002). Fitting. *Computer Vision: A Modern Approach* (pp. 436)

Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data clustering: A review. *ACM Computing Surveys*, 31(3)

Jirah. (2007). *Raven's Progressive Matrices.* Retrieved April 3, 2010, from http://en.wikipedia.org/wiki/Raven%27s_Progressive_Matrices

Lowe, D. G. (2004). Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2)

Raven, J. (1962). *Advanced Progressive Matrices (Sets I and II).* London: Lewis.

Shapiro, L., & Stockman, G. (2001). *Computer Vision* Prentice Hall, Inc.

Vedaldi, A. (2006). *An Implementation of SIFT Detector and Descriptor.* Retrieved March 29, 2010 from http://www.vlfeat.org/~vedaldi/code/sift.html

Acknowledgements: Charlie Tang

The End