Qualitative Image Localization HoG v. SIFT

Presented By:
Sonal Gupta

Problem Statement

- Given images of interior of a building, how much can a robot recognize the building later
- Qualitative Image Localization

I am in Corridor 4
but I do not know
the exact location

Global v. Local approach

- Global - Histogram of Oriented Gradients
  - Introduced by Dalal & Triggs, CVPR 2005
  - Extended by Bosch et. al., CIVR 2007 - pyramid of HoG - used in the experiments with no pyramids
  - Kosecka et. al., CVPR 2003 uses simpler version of HoG for image based localization
- Local - SIFT features
  - Kosecka et. al., CVPR Workshop 2004

Basic HoG algorithm

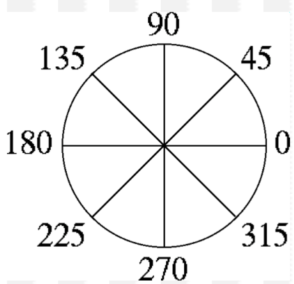
- Divide the image into cells
  - In our case, every pixel is a cell
- Compute edges of the image
  - canny edge detector
- Compute the orientation of each edge pixel
- Compute the histogram
  - Each bin in the histogram represents the number of edge pixels having orientations in a certain range

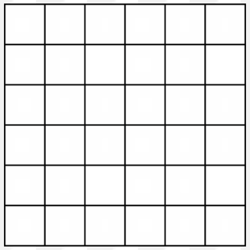
Parameters to HoG

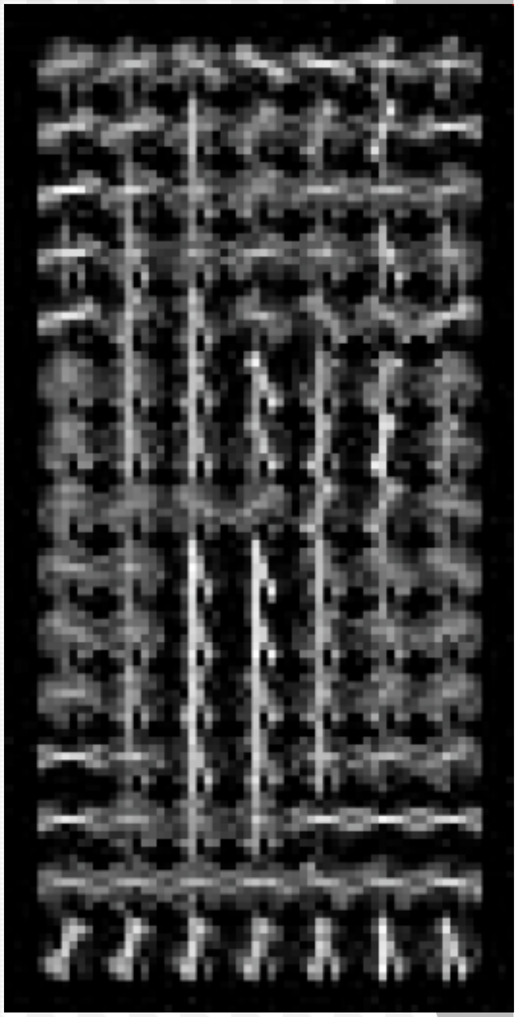
- Number of Bins of the Histogram
- Angle - 180° or 360°,
  - 180° - contrast sign of the gradient is ignored
    - used in the experiments
  - 360° - uses all orientations as in SIFT

- Histogram of gradient orientations

  -Orientation -Position







- Weighted by magnitude

Different HoGs

- Difference between level 0 of pyramid HoG in Bosch et. al. versus Kosecka et. al. implementation of HoG
  - The vote of each edge pixel is linearly distributed across the two neighboring orientation bins according to the difference between the measured and actual bin orientation - soft voting
  - Eg.: Bins - 10°, 20°, 30°; measured value - 17°,
  - vote for: Bin 10° - .15, Bin 30° - .15, Bin 20°- .75

Distance Metric

Chi-Square distance

$$\chi^2(h_i, h_j) = \sum_k \frac{(h_i(k) - h_j(k))^2}{h_i(k) + h_j(k)}$$

h_i and h_j are histograms of two frames
k is the number of histogram bins

Kosecka et. al., CVPR 2003

Benefits of HoG

- Computed globally
- Occlusions caused by walking people, misplaced objects have minor effects
- Can generalize well
- Has worked really well for finding pedestrians on the street

Dataset

Dataset

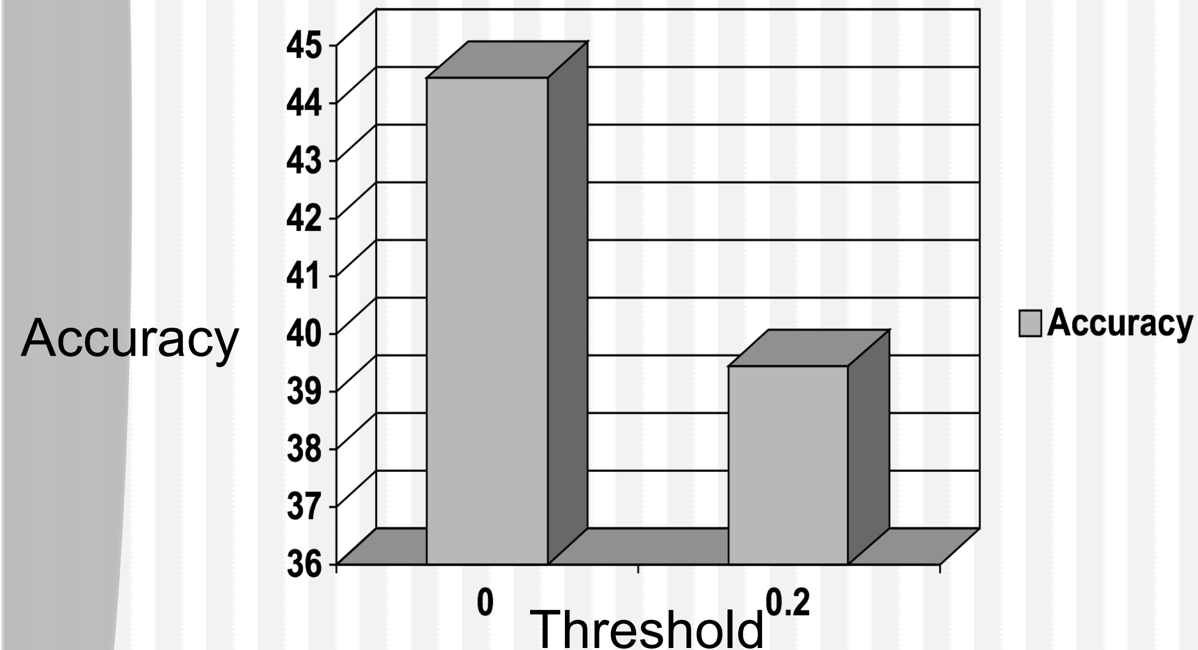
- Total number of images: 92
- Randomly selected 80% to form the training set
- Rest 20% is the test set
- Number of classes: 12
- Ran HoG and SIFT ten times

HoG Experiments

- Effect of a threshold - how much is the nearest image in the training set far from the next nearest
  - ratio of matching features in both the training images
- Effect of Quantization - One representative or prototype view of every class
- Effect of number of bins

Accuracy - Vary Threshold

- Effect of varying the threshold
- Number of Bins = 10

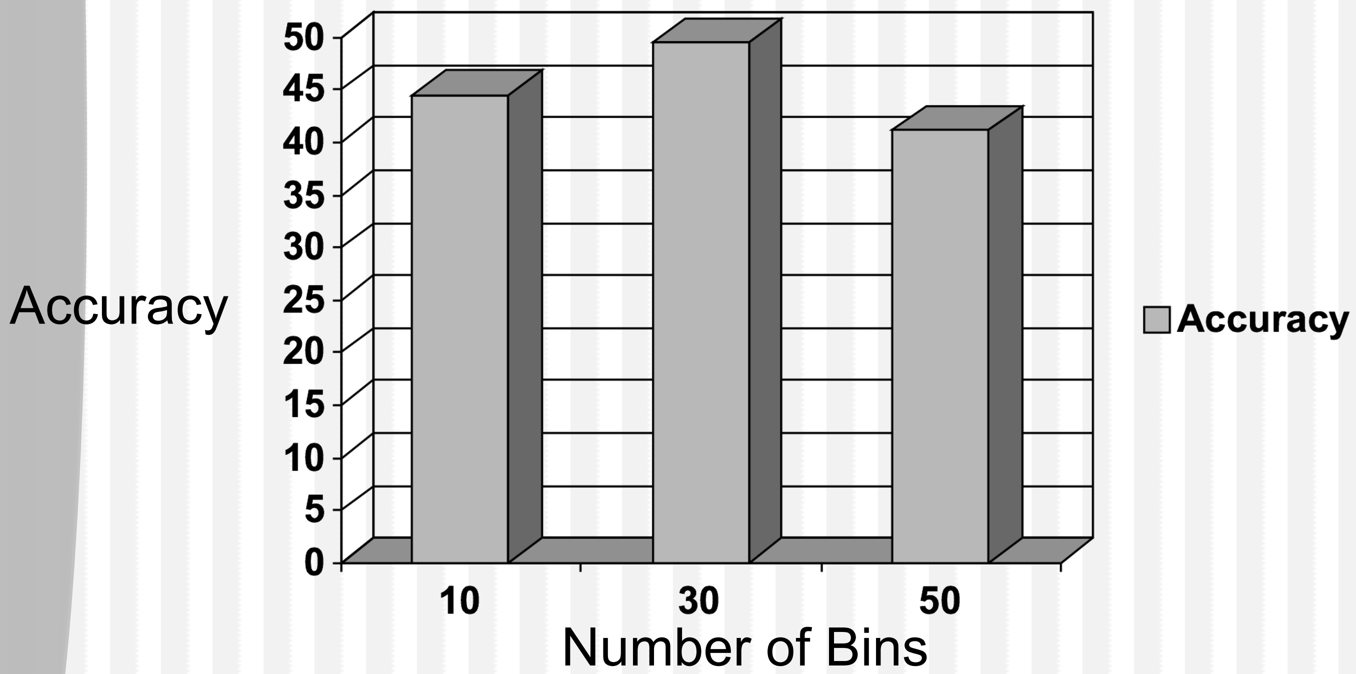


For threshold = 0.2,
Undecided but would have been
•correctly classified - 10!!
•wrongly classified - 8

Many images in the training set have nearly same histogram of oriented gradients

Accuracy - Vary Bins

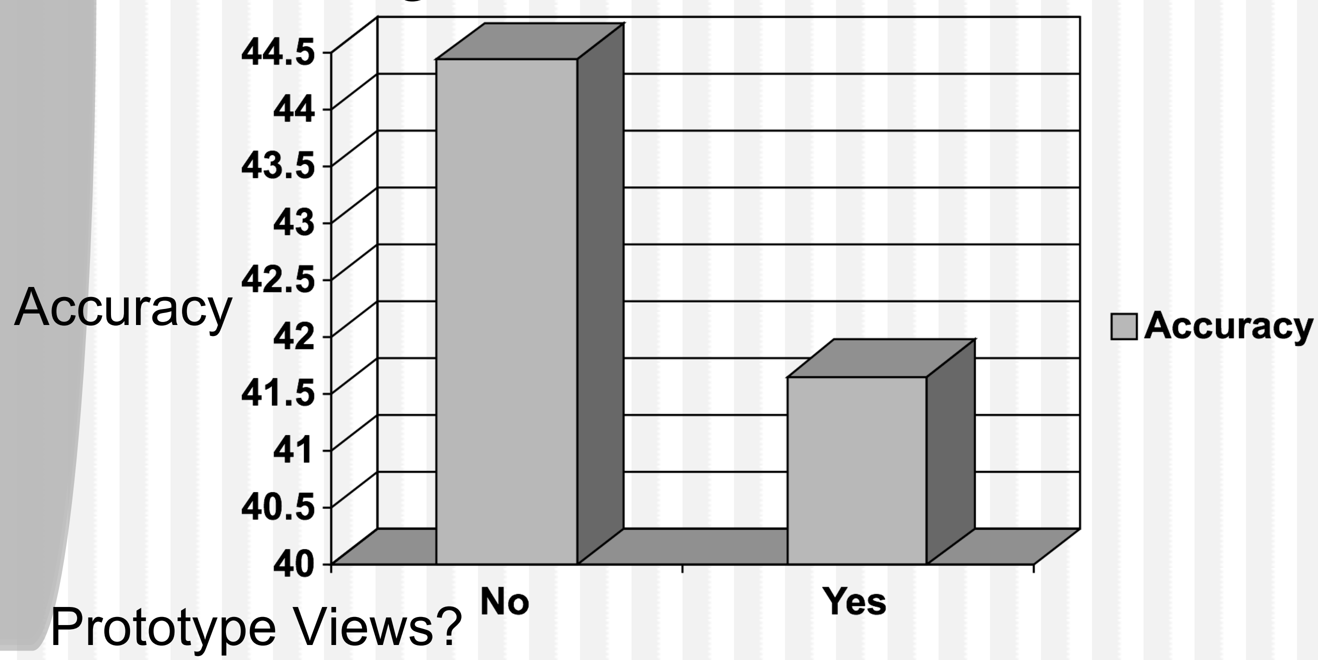
Effect of varying the number of bins
Threshold = 0



Less number of bins - Too much quantization of orientations
Large number of bins - Very less quantization of orientations

Accuracy - Prototype Views

- Threshold = 0, Bins = 10, One prototype image per class
- Prototype image computed by taking mean of images of same class



Best Combination

Best Combination

- Threshold = 0
- Bins = 30
- No prototype views

HoG Results

Test	Result	
		Correct
		Correct

Obvious answers

Test

Result

Wrong

Wrong

Some images are just hard to classify…

Test

Result

Guess?

Test

Result

Guess?

Test

Result

Confused?

Test

Result

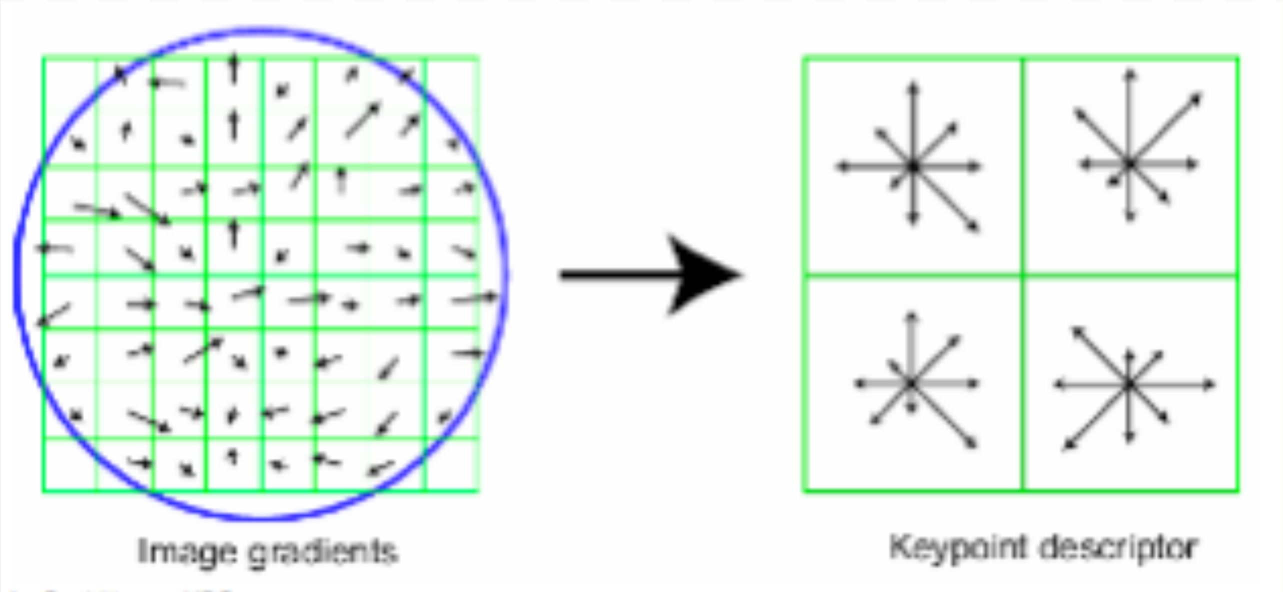
All are wrongly classified, though they look so similar…

SIFT

- Scale & affine invariant feature detection
  - Combines edge detection with Laplacian-based automatic scale selection
  - Mikolajczyk et. al. CVPR'06, BMVC '03
- SIFT descriptor

SIFT Vector Formation

- Threshold image gradients are sampled over 16x16 array of locations in scale space
- Create array of orientation histograms
- 8 orientations x 4 x 4 histogram array = 128 bit vector



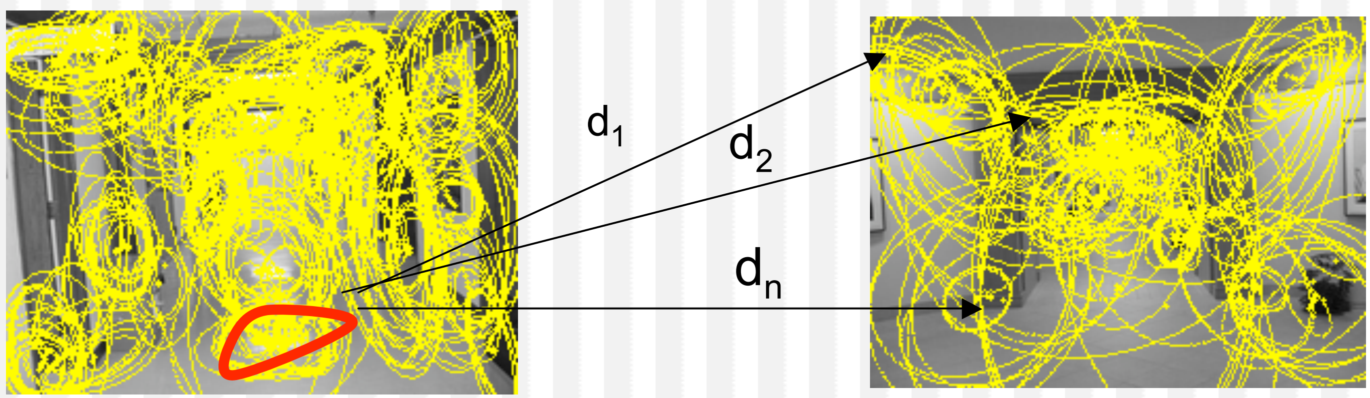
Algorithm

- For every test image
  - For every training image
    - Find the nearest matching feature
    - Find the second nearest matching feature
    - If nearest neighbor 0.6 times closer than the second nearest neighbor
      - Number_of_matching_features ++
  - Find the training image with most number of matching features

How features are matched

Test Images

Each training image



Let di be the minimum distance and dj be the second minimum then

$feature_{test}$ matches $feature_i$ if $d_i < 0.6*d_j$

Two Types of Threshold

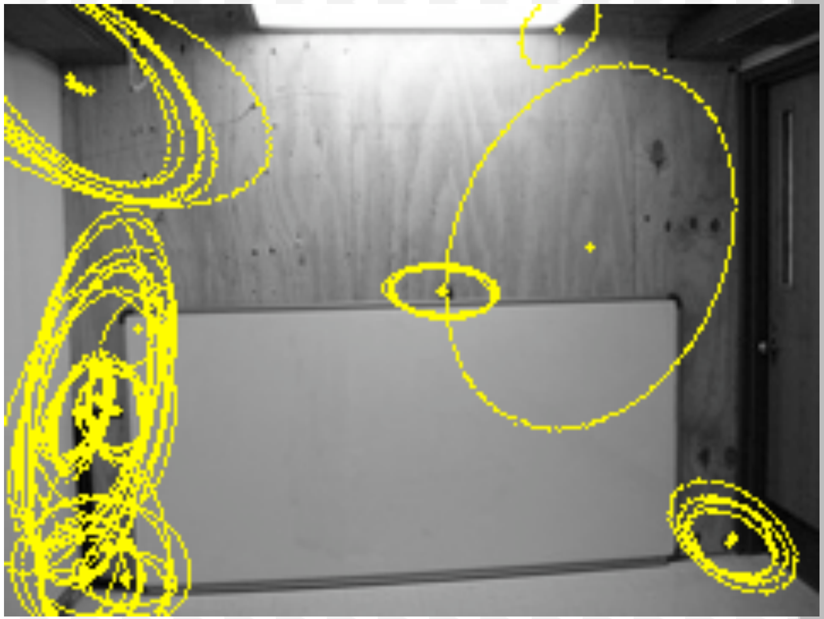
- One is to check whether there is a matching feature in the given training image or not
  - Fixed - 0.6
- One is to check whether the nearest image is far away from the next nearest image or not
  - Experimented for various values

Results - Numbers

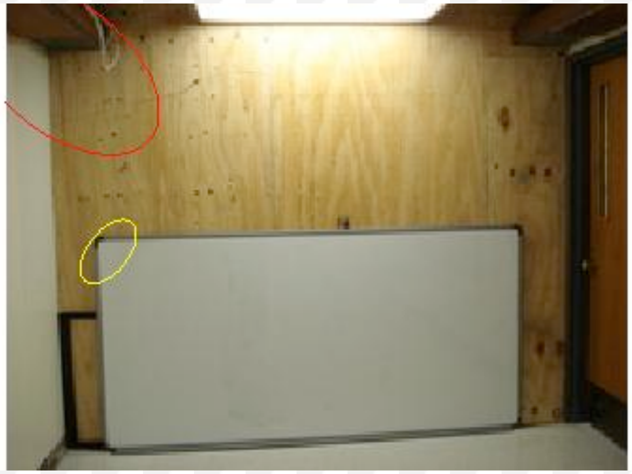
- SIFT
  - Correctly Classified - 99
  - Wrongly Classified - 81
  - Accuracy - 55%

Better than HoG!

SIFT - One bad image ruined the accuracy!



Reason

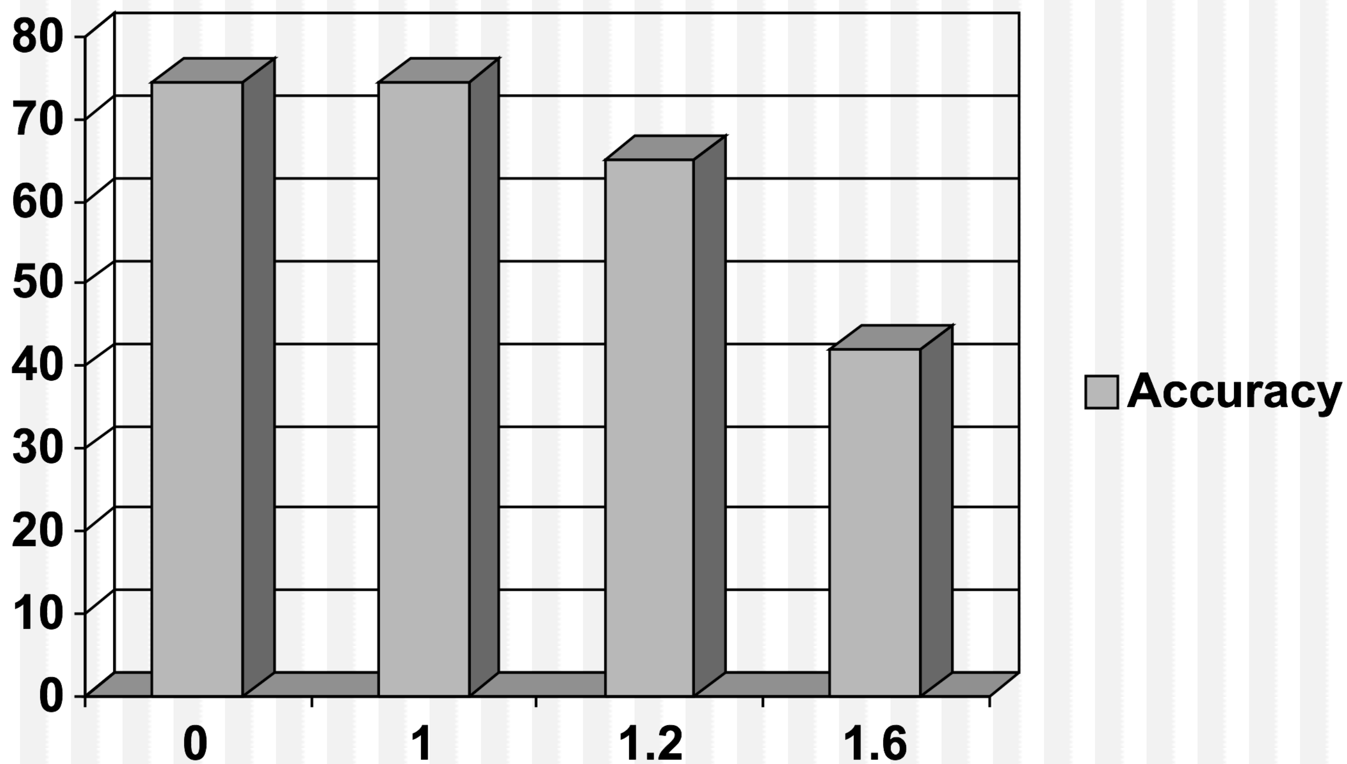


New Results for SIFT

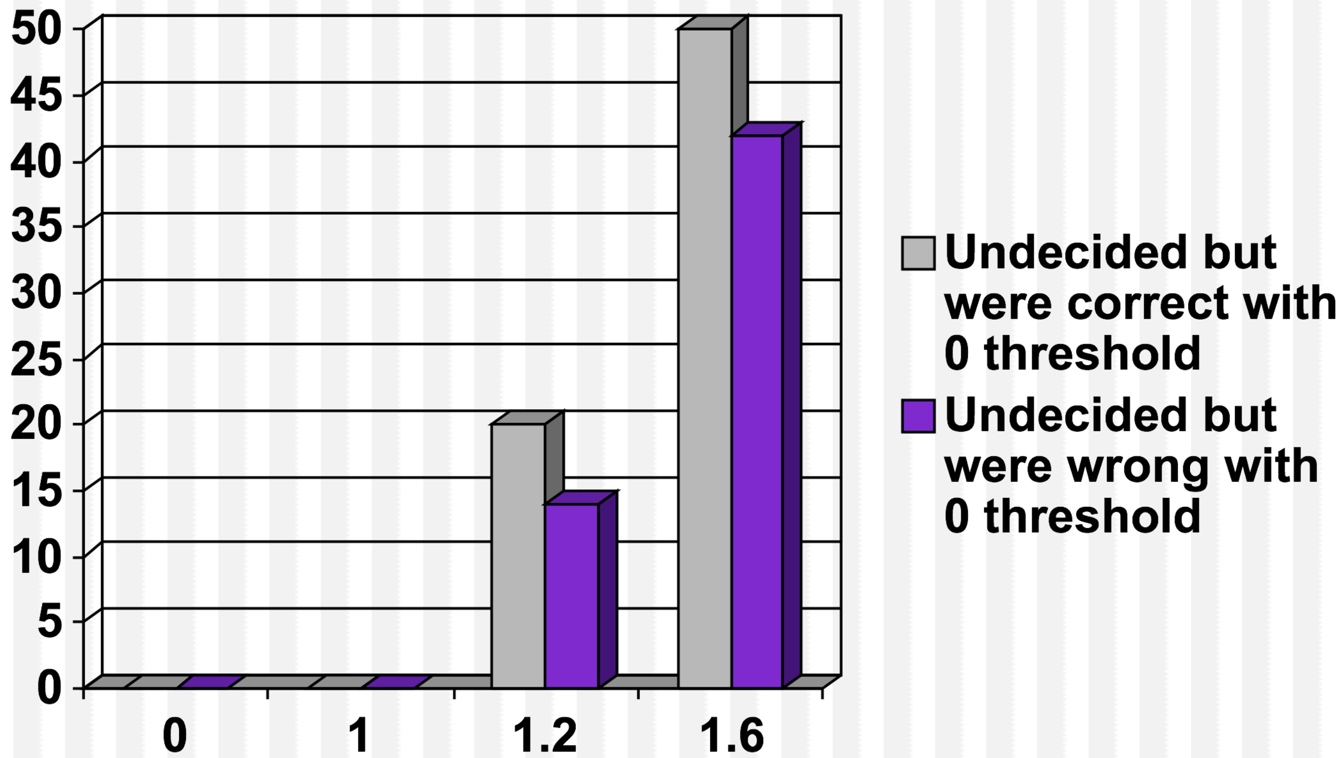
- Removed the image
  - Avg. no. of images correctly classified: 134
  - Avg. no. of images wrongly classified: 46
  - Accuracy 74.4%
  - Earlier accuracy 55%
    - 19.44% higher accuracy!!

Result

- Varying the threshold



Threshold is not good

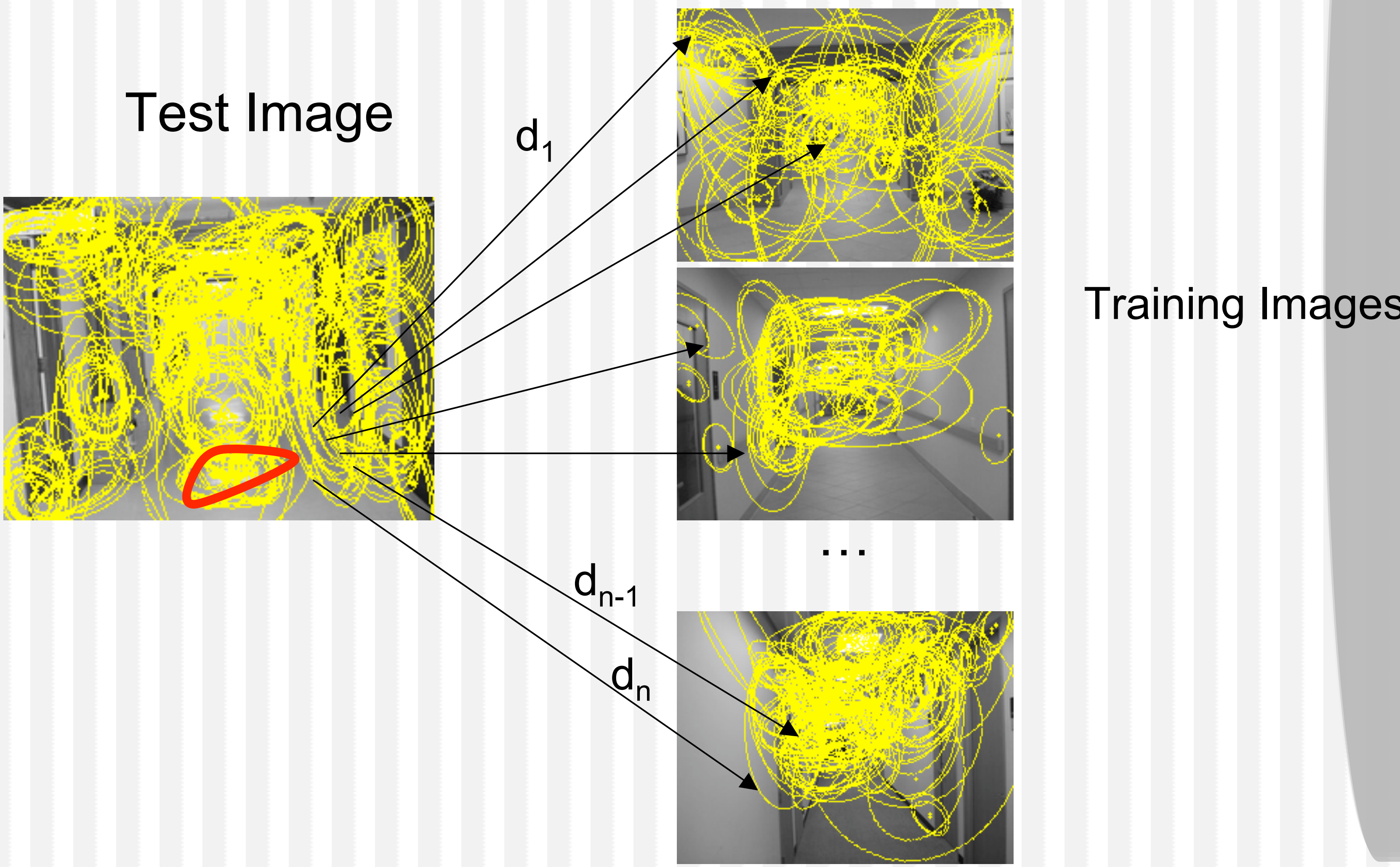


Modified feature matching in SIFT

- For every test feature, find nearest and second nearest feature from ALL the training images' features
- A feature is matching if

  nearest_distance < 0.6*second_nearest_distance
- Find the training image that has most features matching with the test image
- Call this one $SIFT_2$ and the earlier one $SIFT_1$

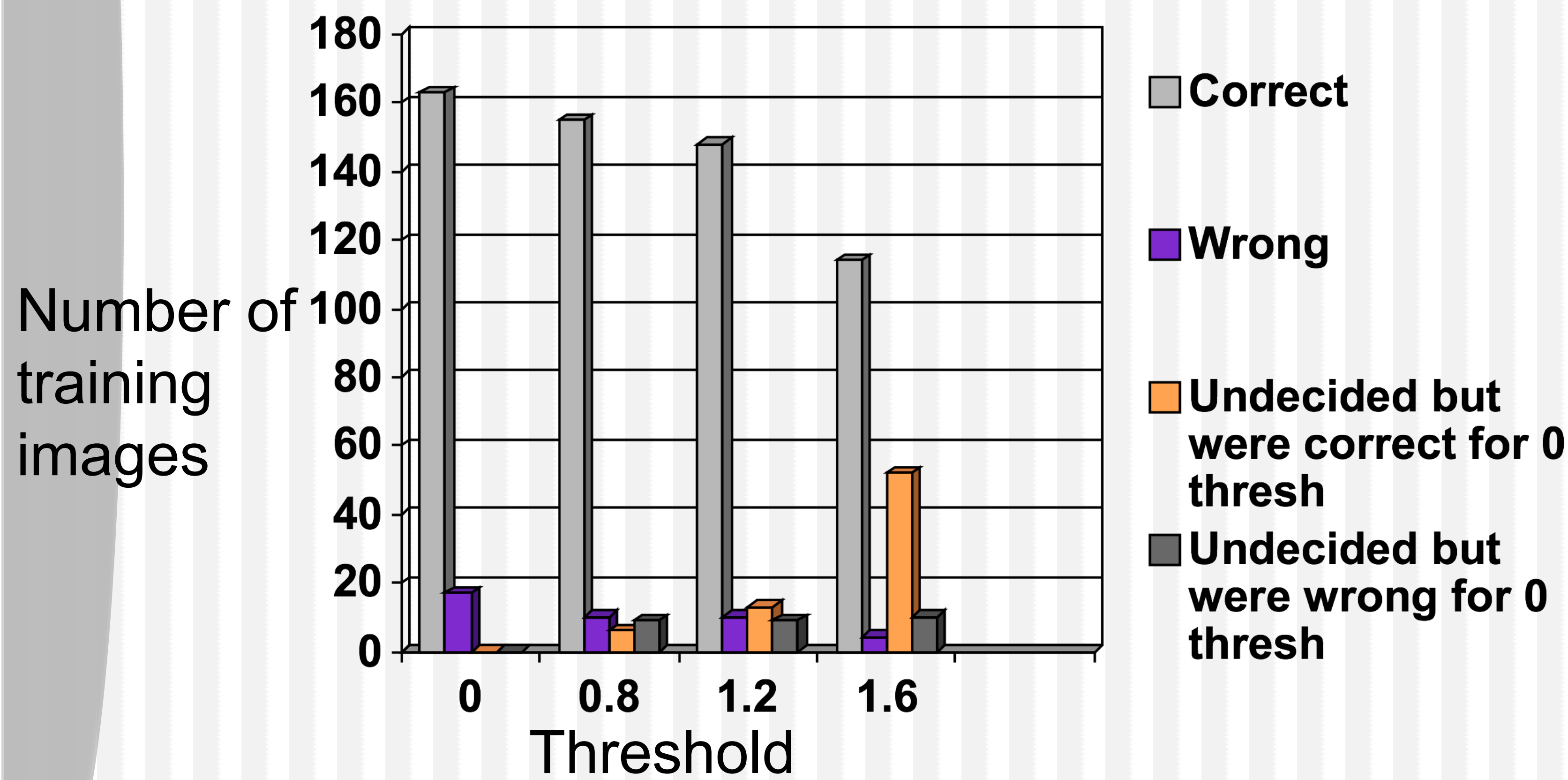
Modified feature matching



Result of $SIFT_2$

- Threshold = 0
  - Correct - 163
  - Wrong - 17
  - Accuracy - 90. 5%
  - Accuracy of $SIFT_1$ = 74.4% -- 16.1% higher!!
- Also, the one bad image problem gets removed!

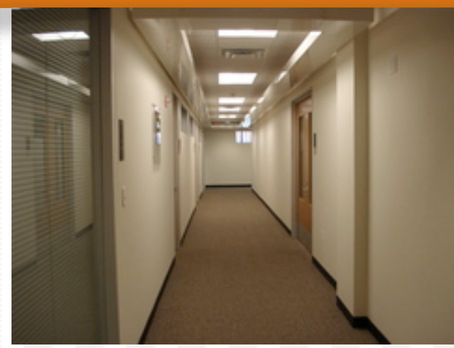
Vary Threshold in $SIFT_2$

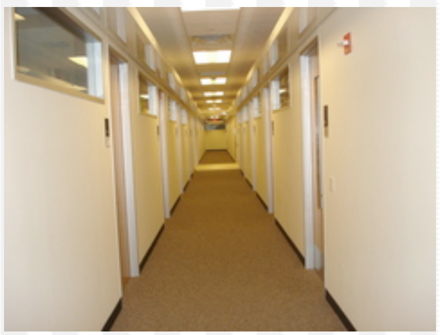


Another dataset

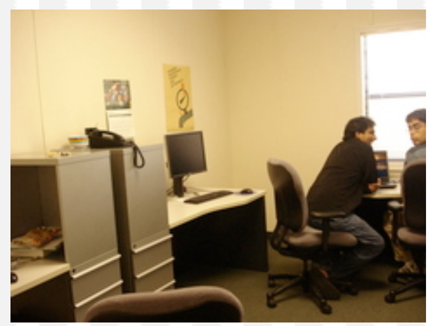
- Till now we had images of the SAME building in our training set
- What if Robot is shown a DIFFERENT building?
  - Can it recognize if an image is a corridor or an office?
- Test dataset has images from different floor and different buildings
  - ACES 5th floor and Taylor hall's corridor
  - Removed the Taylor Hall's corridor images from the training set

Dataset - II







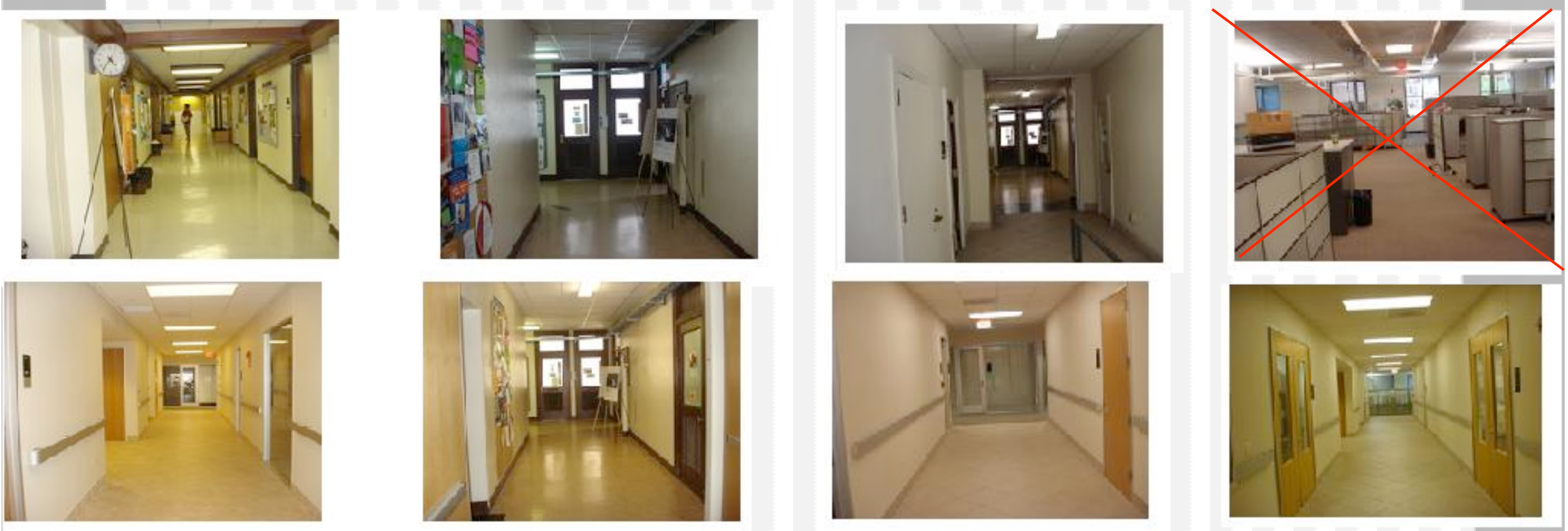
Result

Test | HoG | $SIFT_1$ | $SIFT_2$

No clear winner but $SIFT_2$ = -1

Results

Test HoG $SIFT_1$ $SIFT_2$



No clear winner, but $SIFT_2$= -2

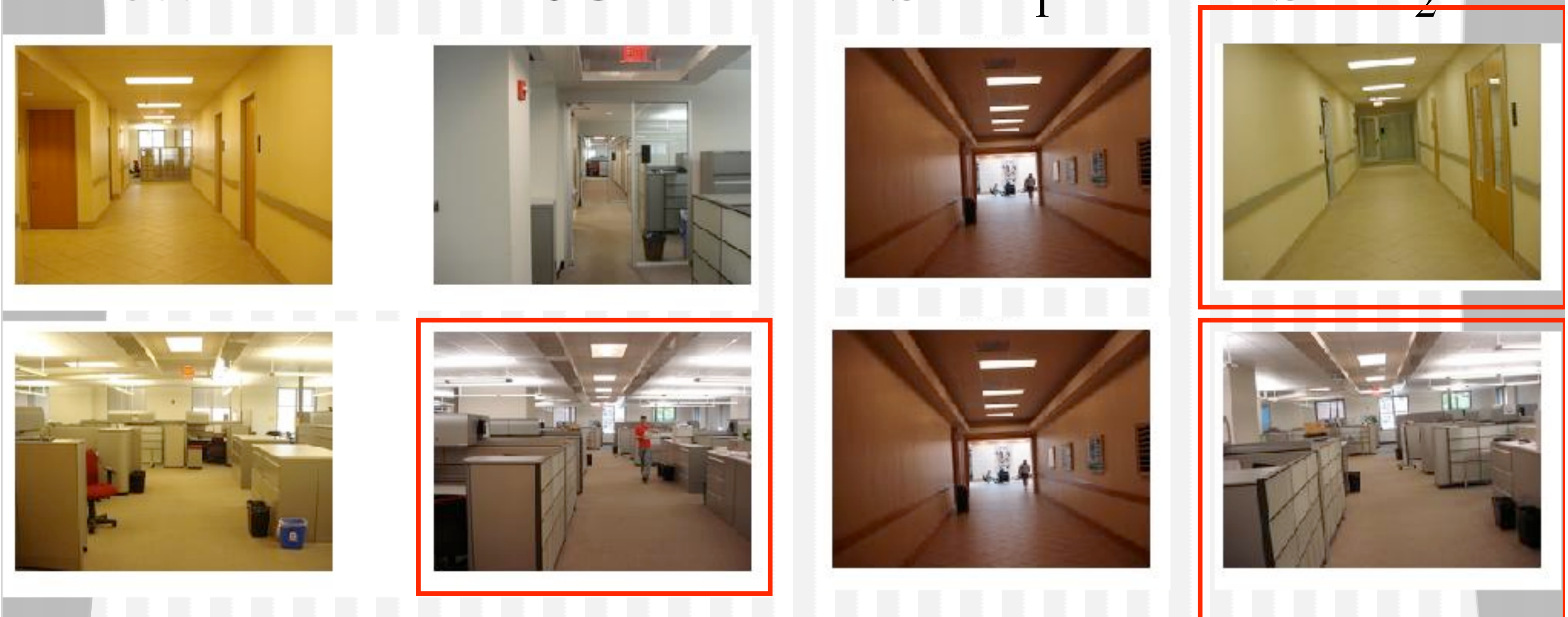
Results

Test HoG $SIFT_1$ $SIFT_2$

HoG = 1; $SIFT_1$ =1, $SIFT_2$= -2+1 = -1

Results

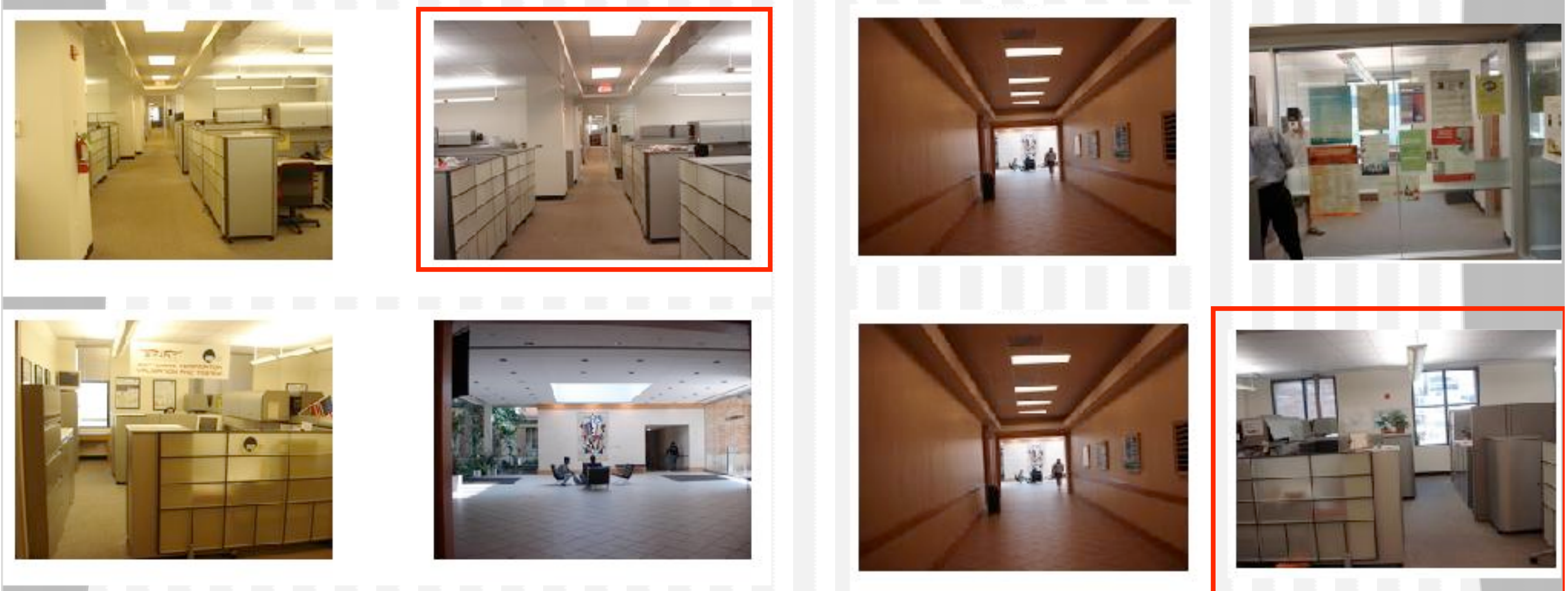
Test HoG $SIFT_1$ $SIFT_2$



HoG = 2; $SIFT_1$=1, $SIFT_2$=-1+2=1

Results

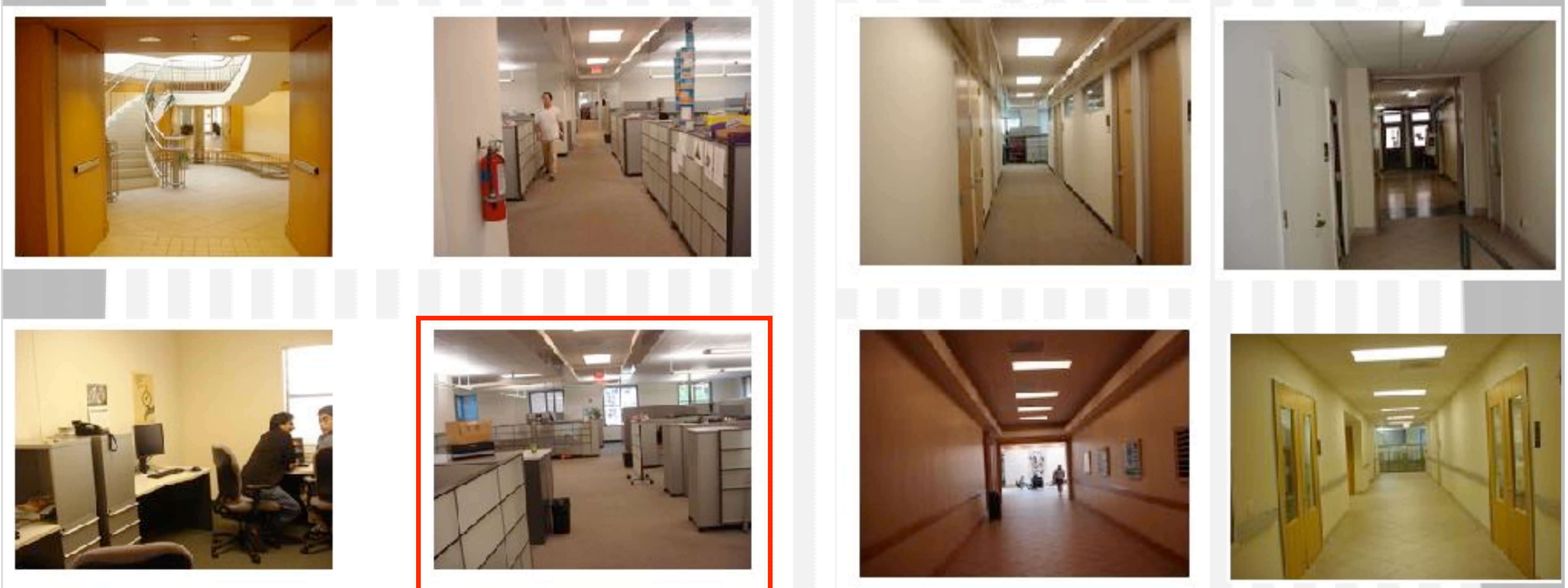
Test HoG $SIFT_1$ $SIFT_2$



HoG = 3; $SIFT_1$ = 1, $SIFT_2$ = 2

Results

Test HoG $SIFT_1$ $SIFT_2$



HoG = 4, $SIFT_1$ = 1, $SIFT_2$ = 2

HoG better than SIFT!

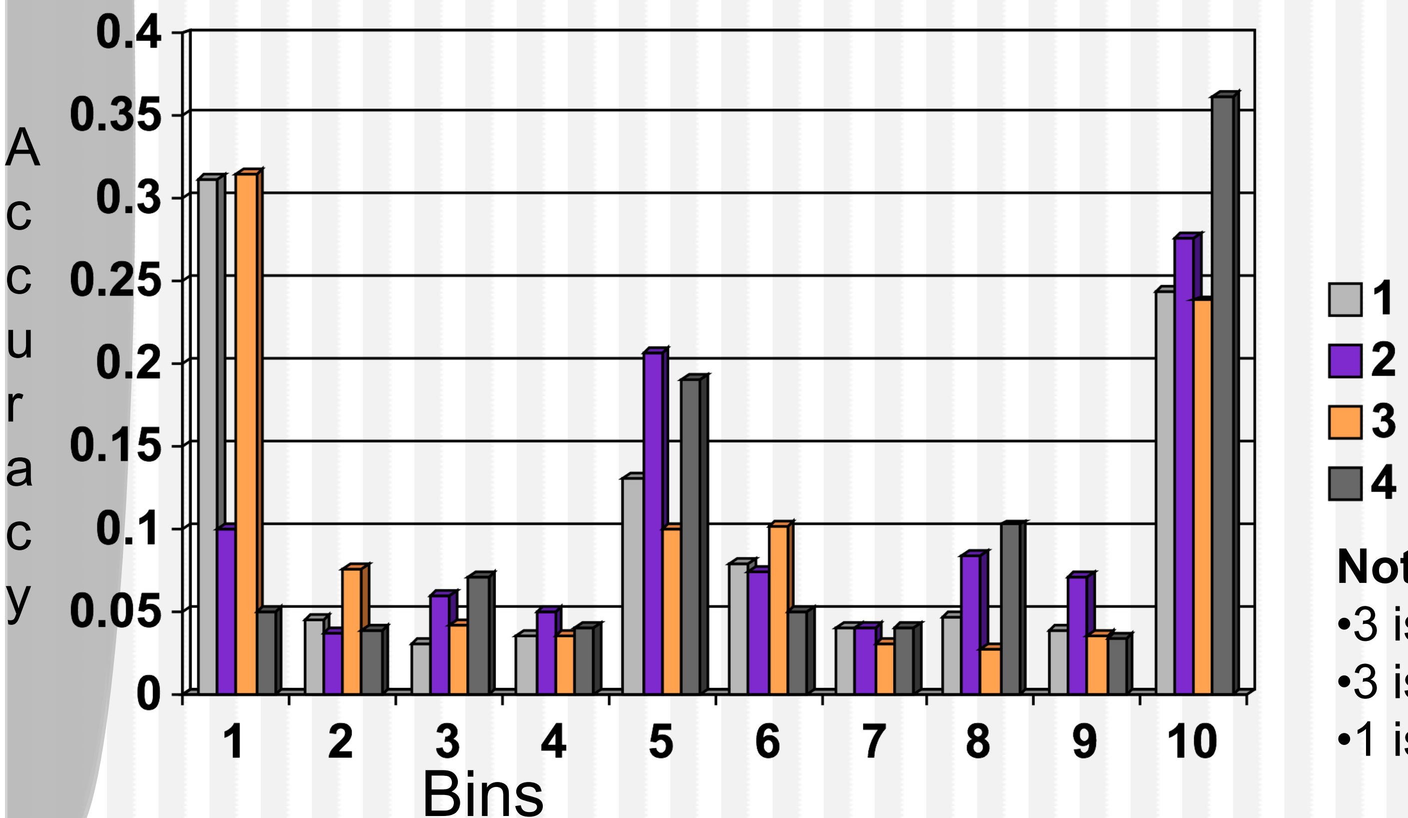
Explanation

- HoG captures the global distinctiveness of a category
- Lets see histograms of some of the images

Result of HoG Of same class as 1 Test Result of SIFT

1 2 3 4



Note

- 3 is similar to 1
- 3 is not similar to 4
- 1 is not very similar to 2

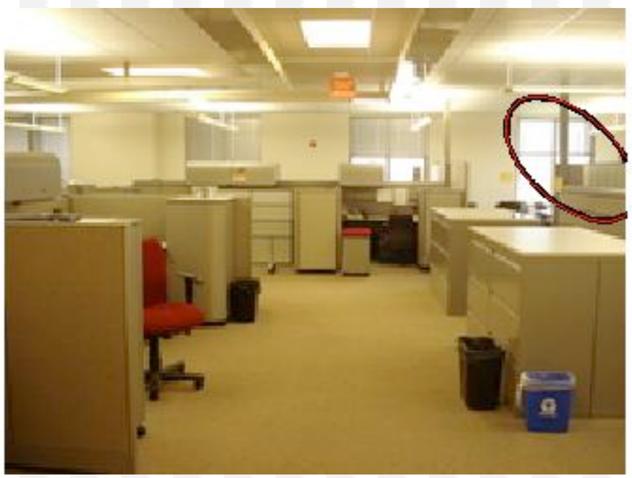
SIFT Explanation

- 20 matching points between test and result images

Test

Result of SIFT_1

Test Image

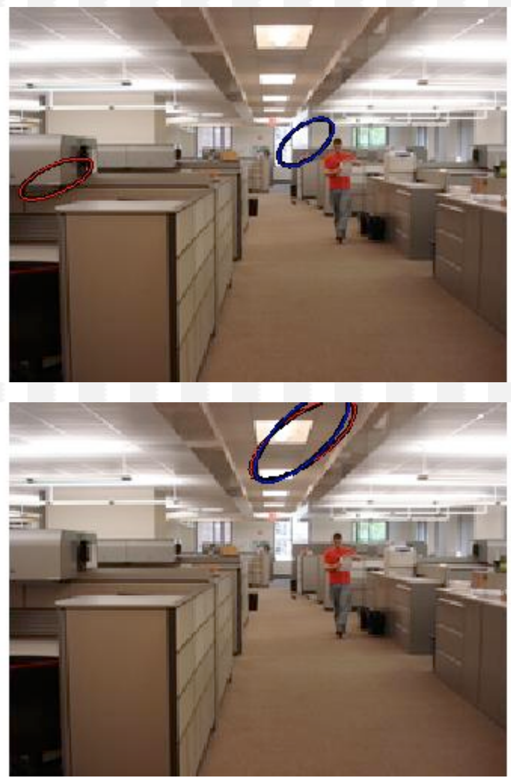


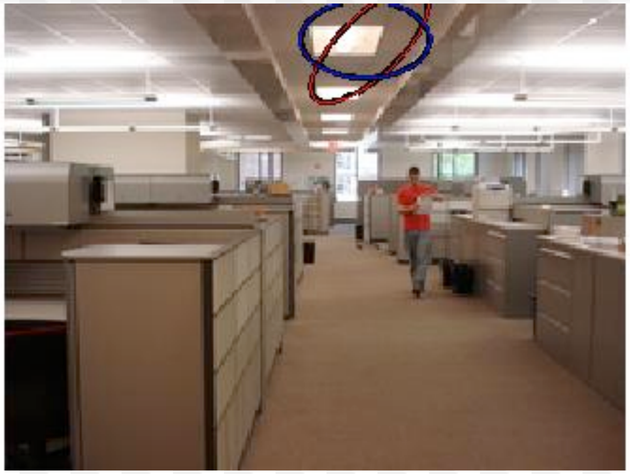
Result Image

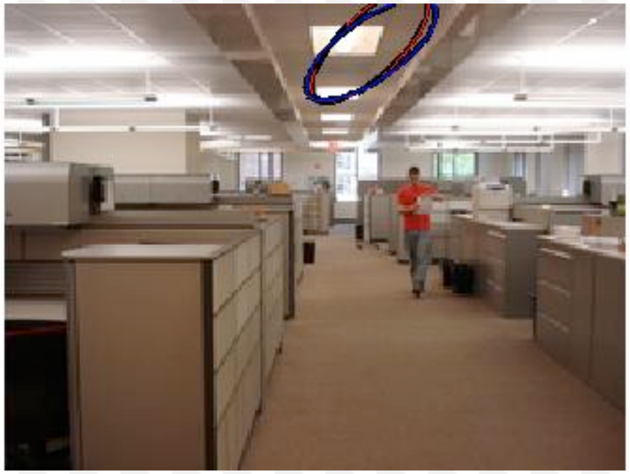
- Only 6 matching points between test image and the result produced by HoG(correct)

Test

Result by HoG





Conclusion

- SIFT performs better than HoG in previously seen building
  - Local descriptor - gets the distinguishing local features
- HoG performs better than SIFT in previously unseen building!
  - Global descriptor - gets the essence
  - Better than SIFT in formal setting of the environment -- Buildings are never at 30°!!
  - Rotation invariance of SIFT results in worse accuracy

Conclusion

- Matching features across all the training images ($SIFT_2$) is better than matching features image by image ($SIFT_1$)
- $SIFT_2$ performs better than $SIFT_1$ in both previously seen and unseen buildings
- Quantization by taking mean in HoG gives poorer performance
- If we are performing 1-NN approach in classification using $SIFT_1$, then one bad image can deteriorate the results

Discussion Points

- Will threshold for selecting nearest images over next nearest image work when we quantize the image?
  - Since only one image per class
- Modify the threshold criteria by calculating ratio of number of matching features of nearest neighbor and for next nearest neighbor of *different* class
- Rotation invariance of SIFT is sometimes hurting the performance. Can we make it partially invariant for this task?
- What can be other matching algorithms than SIFT and HoG?

References and Resources

- Kosecka et. al., Qualitative Image Based Localization in Indoor Environments, CVPR 2003
- Dalal and Triggs, Histograms of Oriented Gradients for Human Detection, CVPR 2005
- Kosecka et. al., Location Recognition and Global Localization Based on Scale-Invariant Keypoints, CVPR Workshop 2004
- Pyramid of Histogram of Oriented Gradients
  - http://www.robots.ox.ac.uk/~vgg/research/caltech/phog.html
- Local features detector and descriptor
  - http://www.robots.ox.ac.uk/~vgg/research/affine/detectors.html