# Fast Segmentation via Randomized Hashing

Camillo J. Taylor and Anthony Cowley
GRASP Laboratory
University of Pennsylvania
Philadelphia, PA USA

# The Segmentation Problem

Input Image

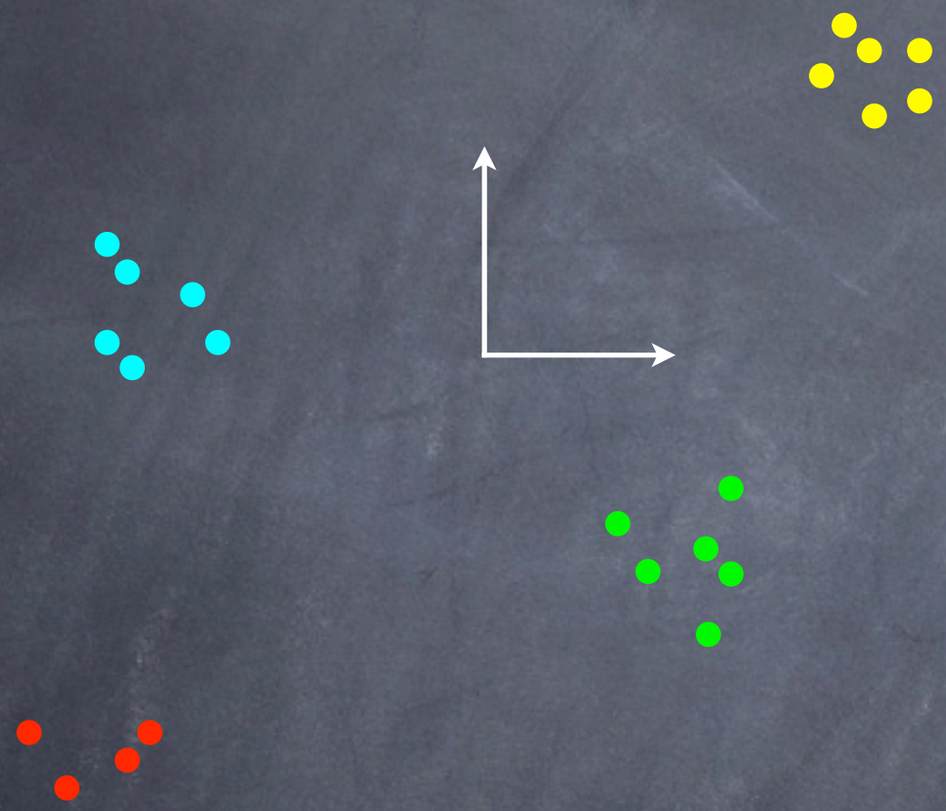
Segmentation

Goal: Break the input image into coherent regions accurately and quickly.

# Prior Work

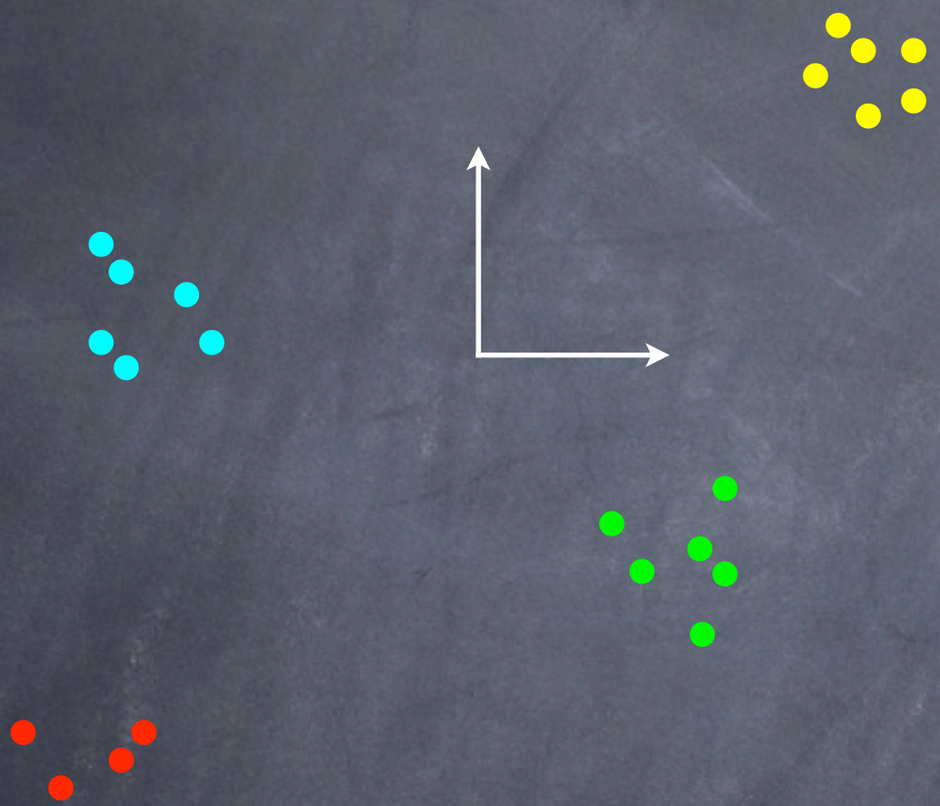
- Graph based methods
  - [Shi & Malik, 97]
  - [Felzenswalb & Huttenlocher, 04]
- Learning based methods
  - [Shotton, Johnson & Cipolla, 08]
  - [Fowlkes, Martin & Malik, 03]
- Feature based Methods
  - [Comaniciu & Meer, 02]
  - [Vazquez, Weijer & Baldrich, 08]

# A Simple Clustering Scenario

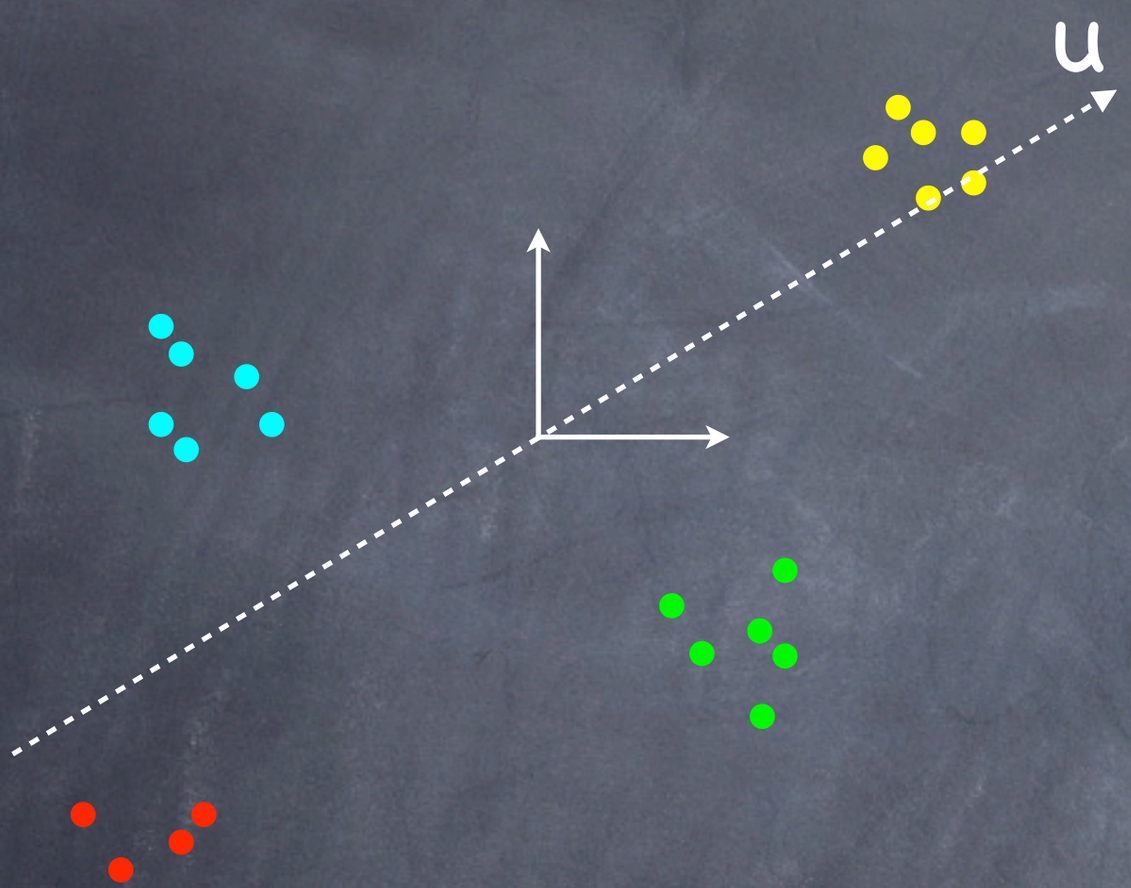


- Consider a set of feature vectors clustered in an m-dimensional feature space.
- Dasgupta's notion of c-separation [Dasgupta, 00] considers the ratio between the inter cluster distances and intra cluster distances.

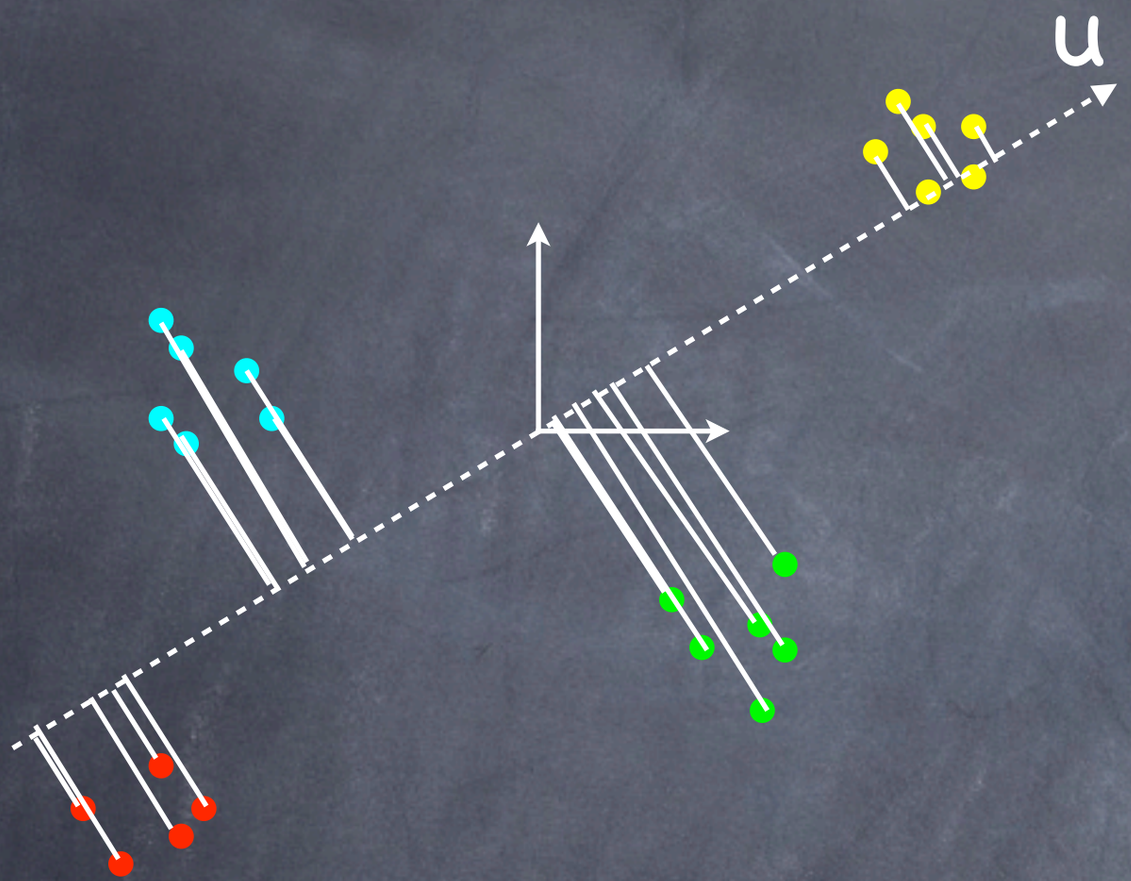
# Random Projection / Splitting



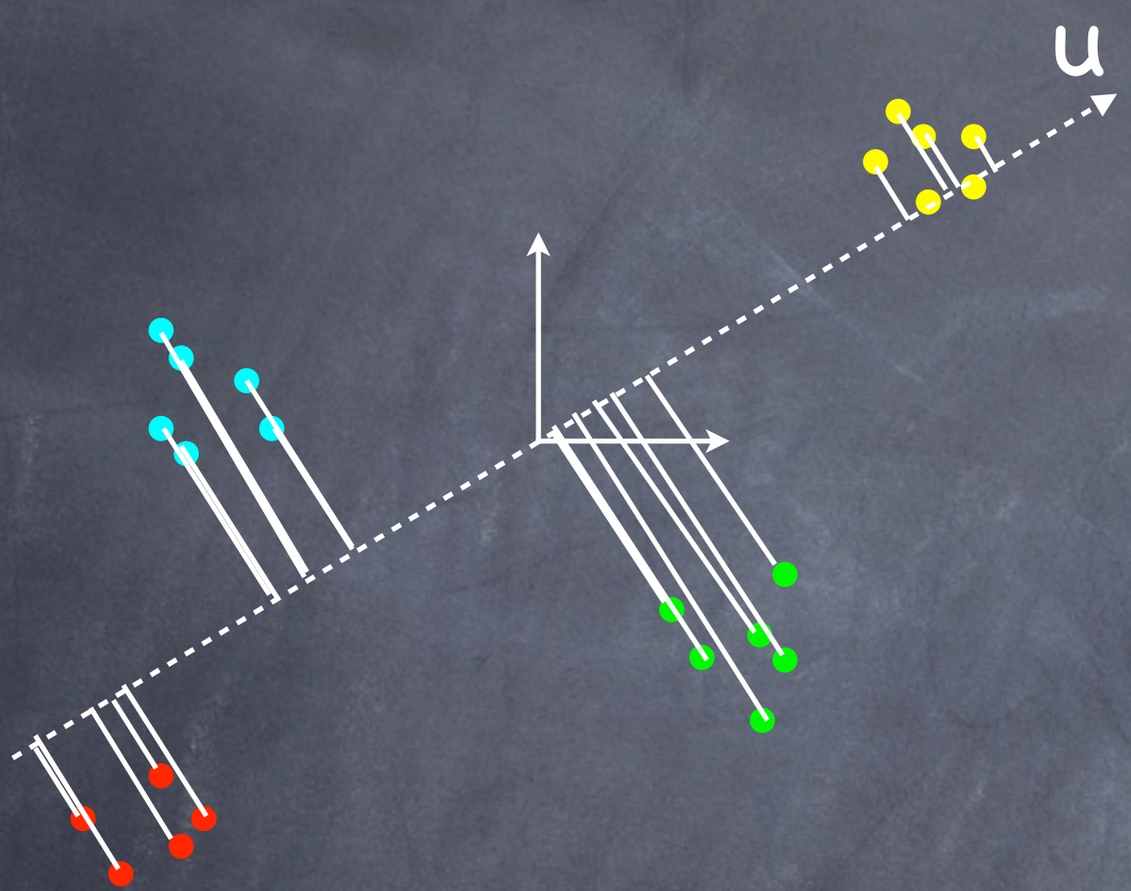
# Random Projection / Splitting

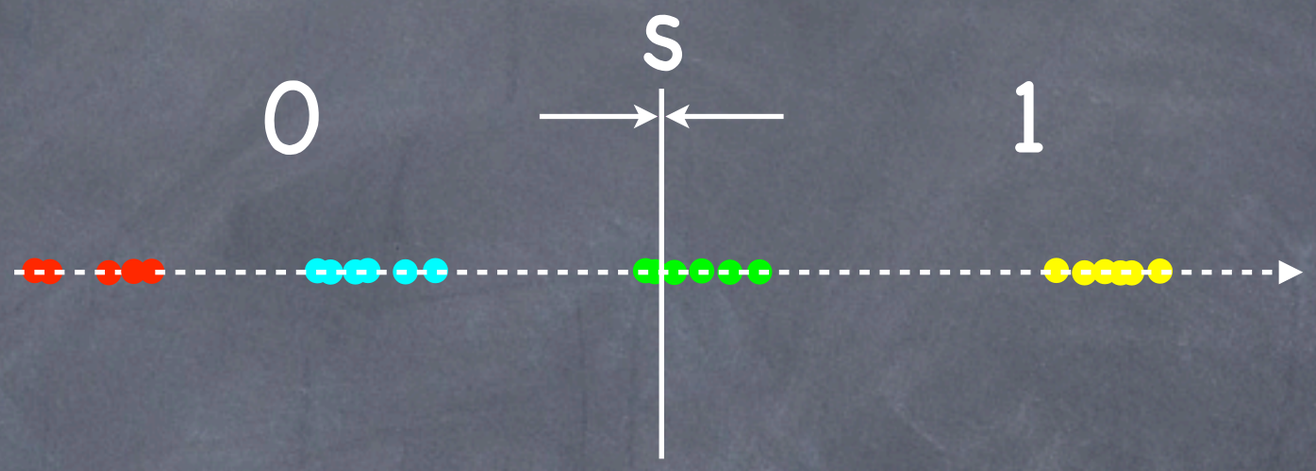


# Random Projection / Splitting

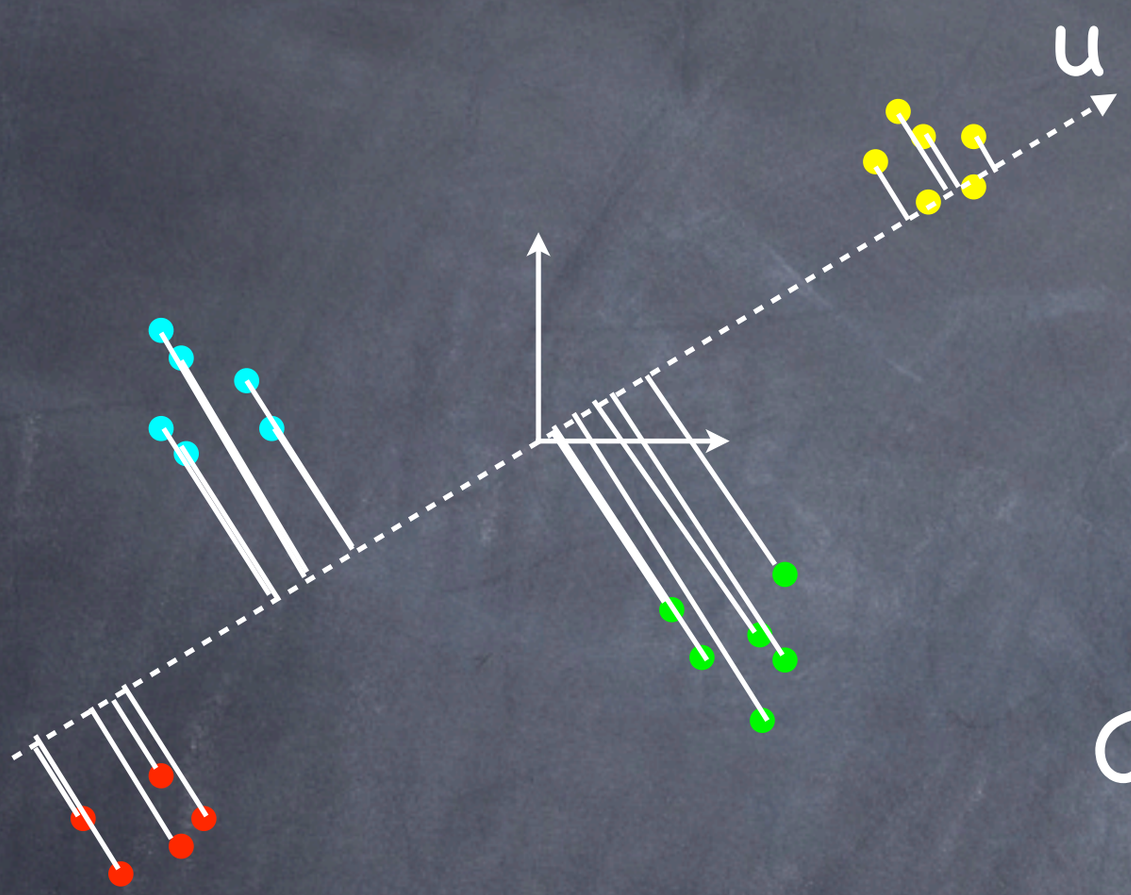


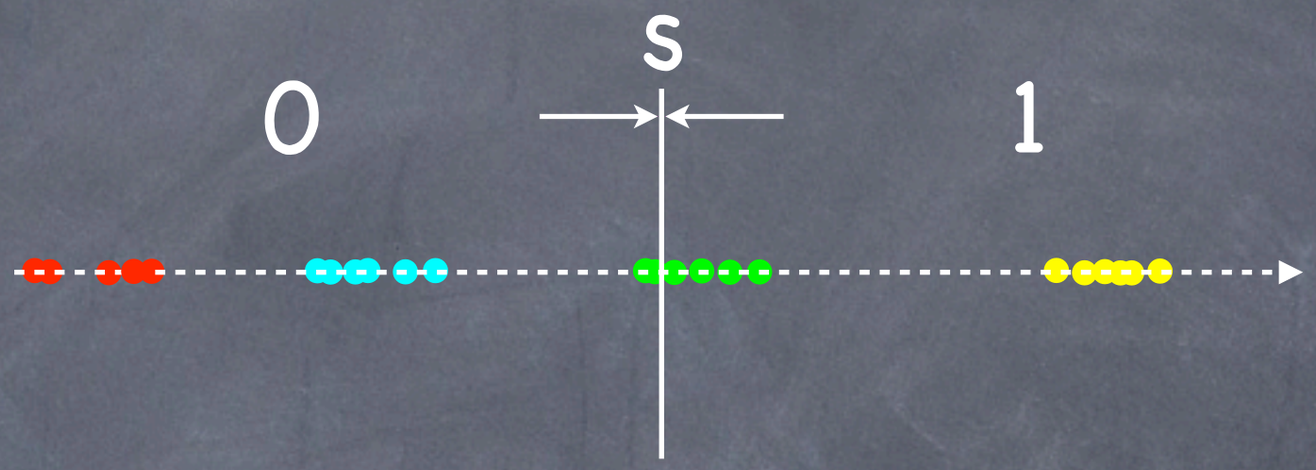
# Random Projection / Splitting



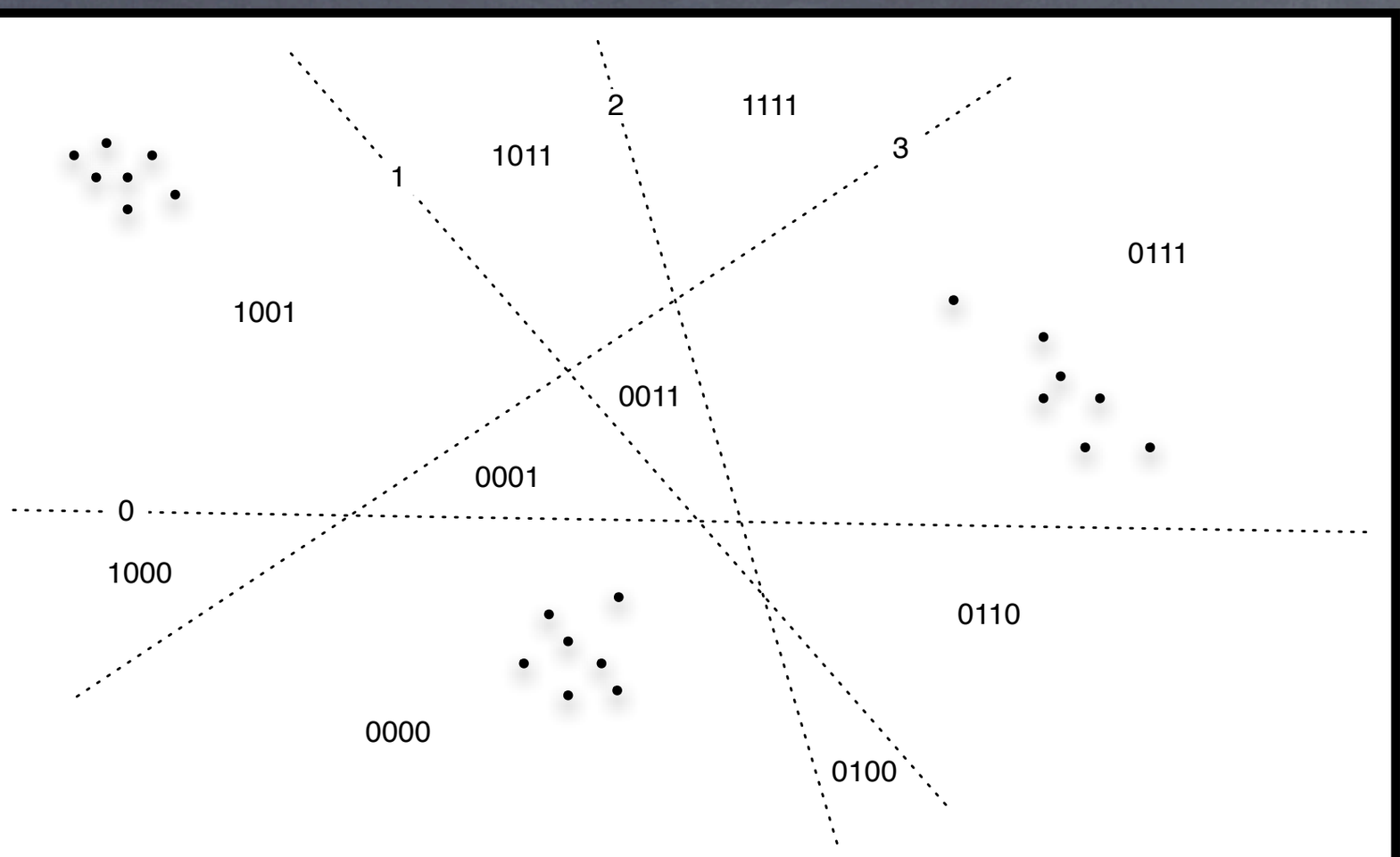
# Random Projection / Splitting

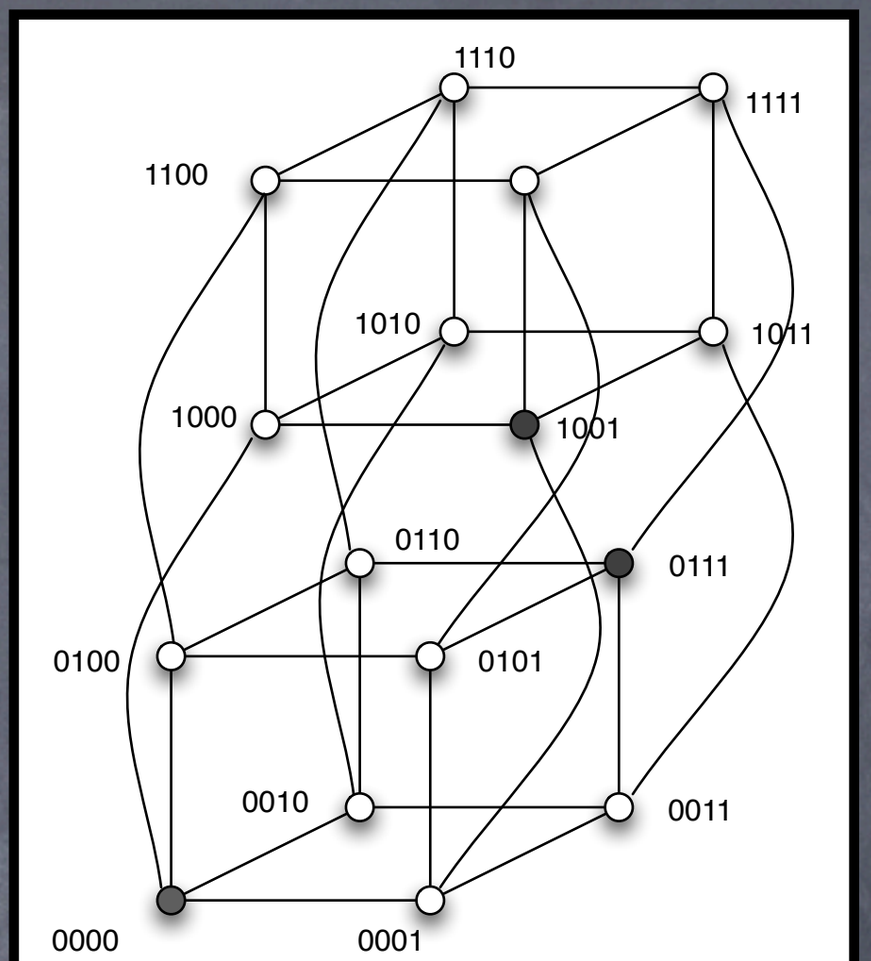



Choices for splitting value, s

- s = mean (u.v)
- s = 0.5*(max(u.v) - min(u.v))




- Each splitting plane contributes a bit to the hash code
- The splitting planes fracture feature space into a collection of convex cells - one for each hash code.
- These cells can be associated with the vertices of a hypercube
- We can look for population maxima in this graph

# Locality Sensitive Hashing

- Key Idea: Points that are close to each other in space will hash to similar codes with high probability. [Indyk & Motwani, 98]

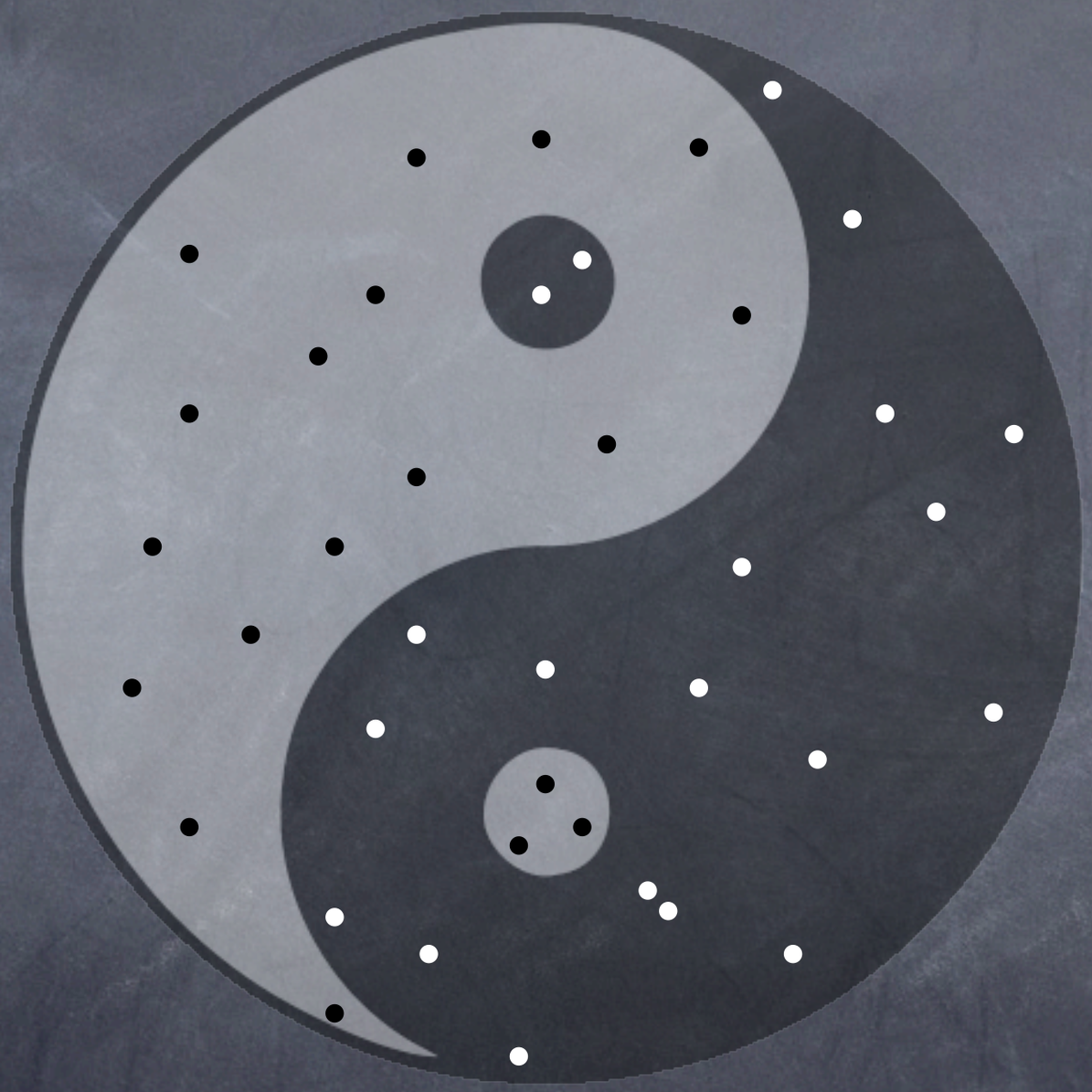
# Pseudo-Code for Segmentation Algorithm

1. Smooth the image and associate a color feature vector with each pixel
2. Hash each feature vector into an n-bit code using n randomly chosen splitting planes
3. Maintain a count of the number of feature vectors mapped to each hash code
4. Identify local maxima in the code space
5. Assign each feature vector to the closest local maxima
6. Run connected components on the labeled pixels to yield coherent image regions.

# Important Parameters

- Number of splitting planes, n
  - Increasing this parameter leads to more segments
- Hamming Distance Threshold, k
  - Used to define local maxima on the hypercube - procedure looks at all nodes in the graph within k hops.
  - Increasing this parameter leads to fewer segments
- Size of smoothing window, w
  - Increasing this parameter leads to increased smoothing, less color variation, and fewer segments

# A Difficult Situation



Feature sets are inextricably intertwined

no good separating hyperplane

# Sample Segmentations
# (The good, the bad and the ugly)

# Sample Segmentations

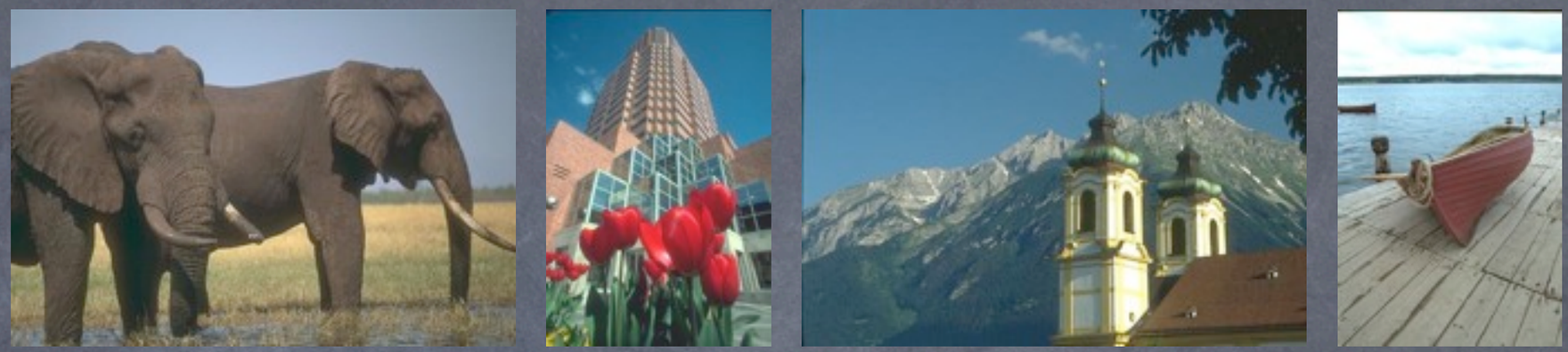
Input

Human

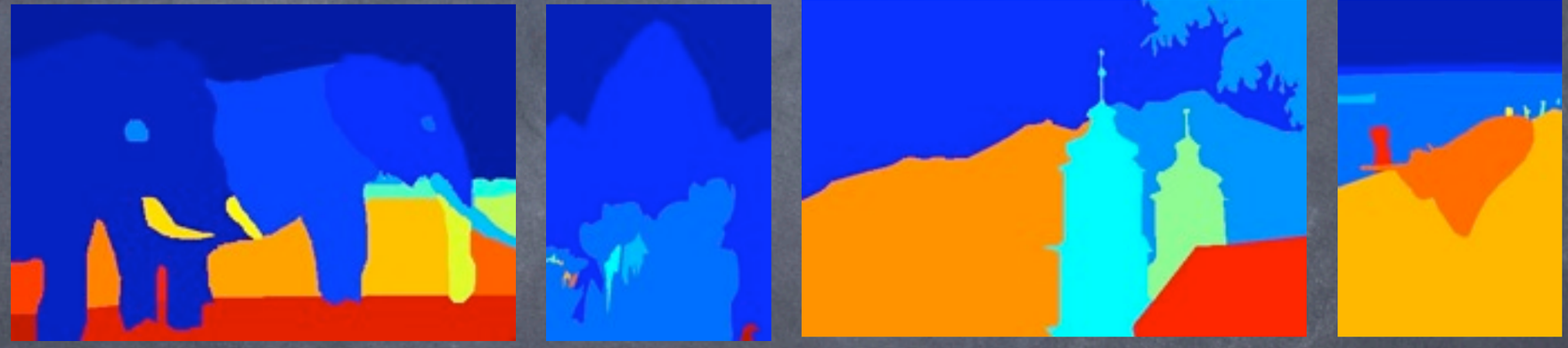
Machine

# Sample Segmentations

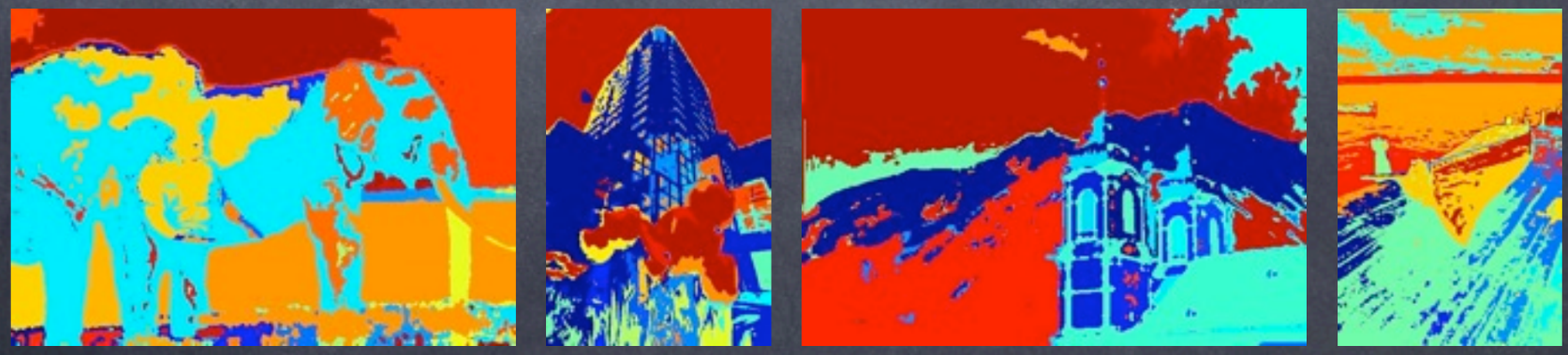
Input



Human

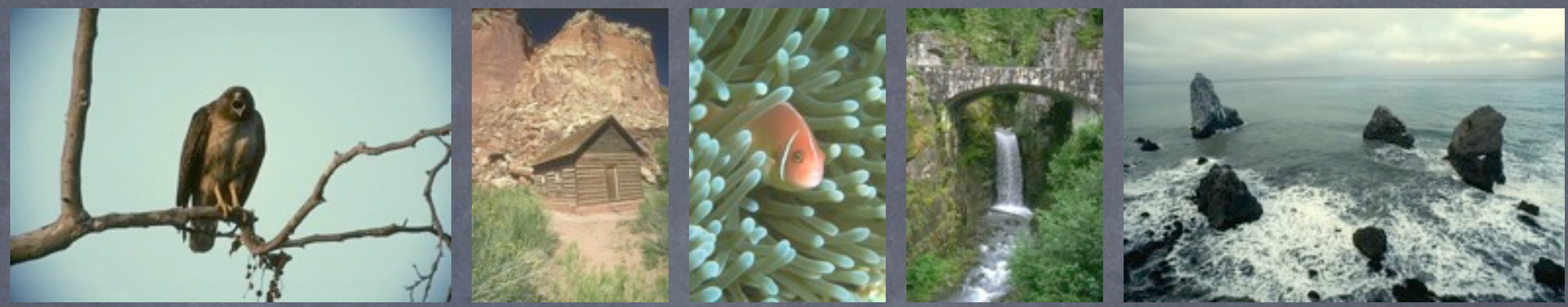


Machine

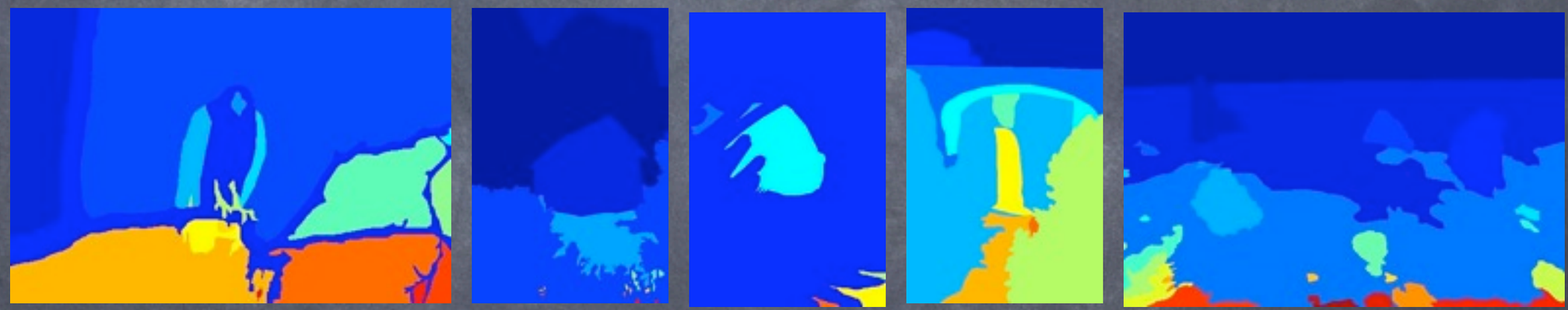


# Sample Segmentations

Input



Human

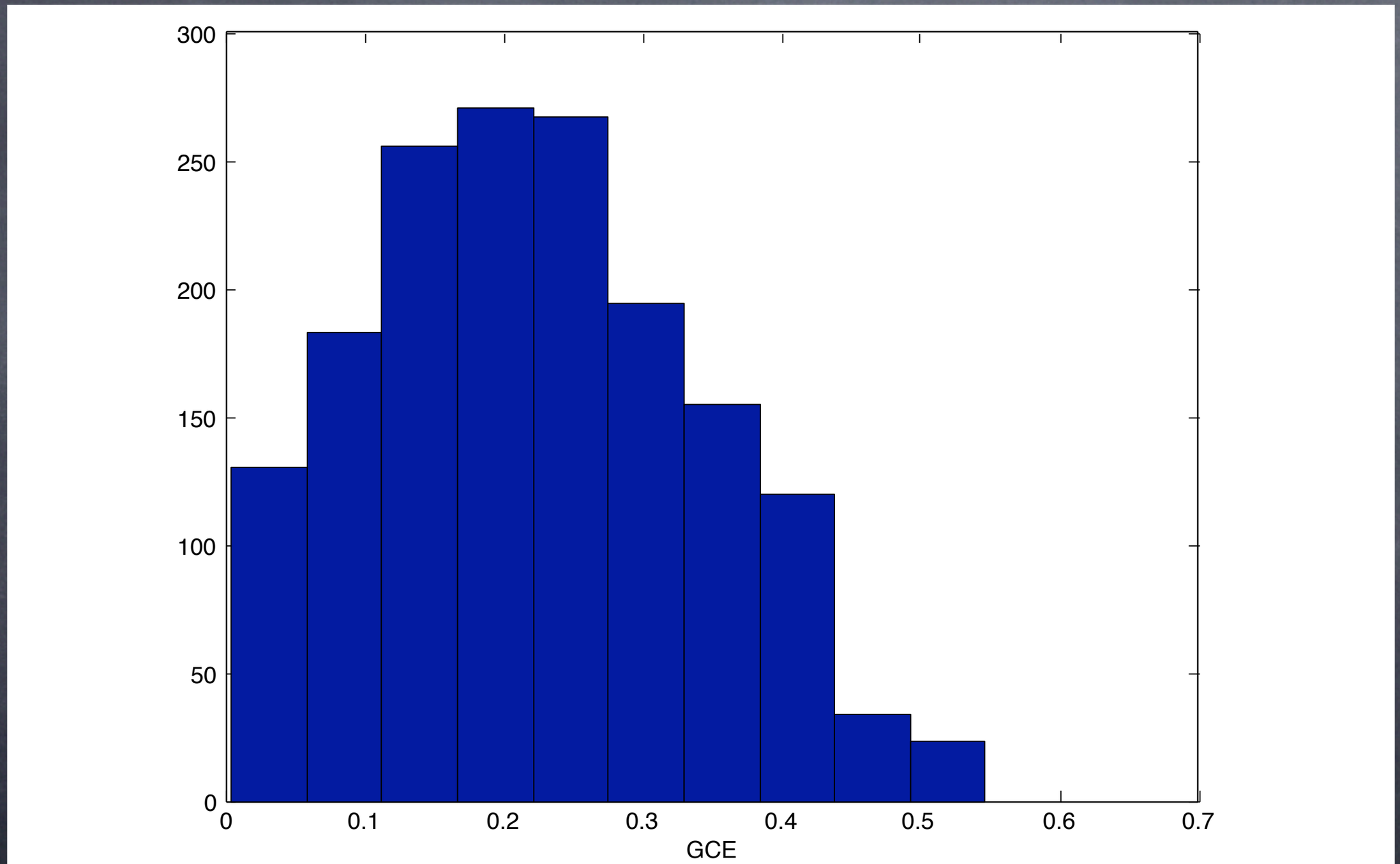


Machine

# Evaluating Segmentations

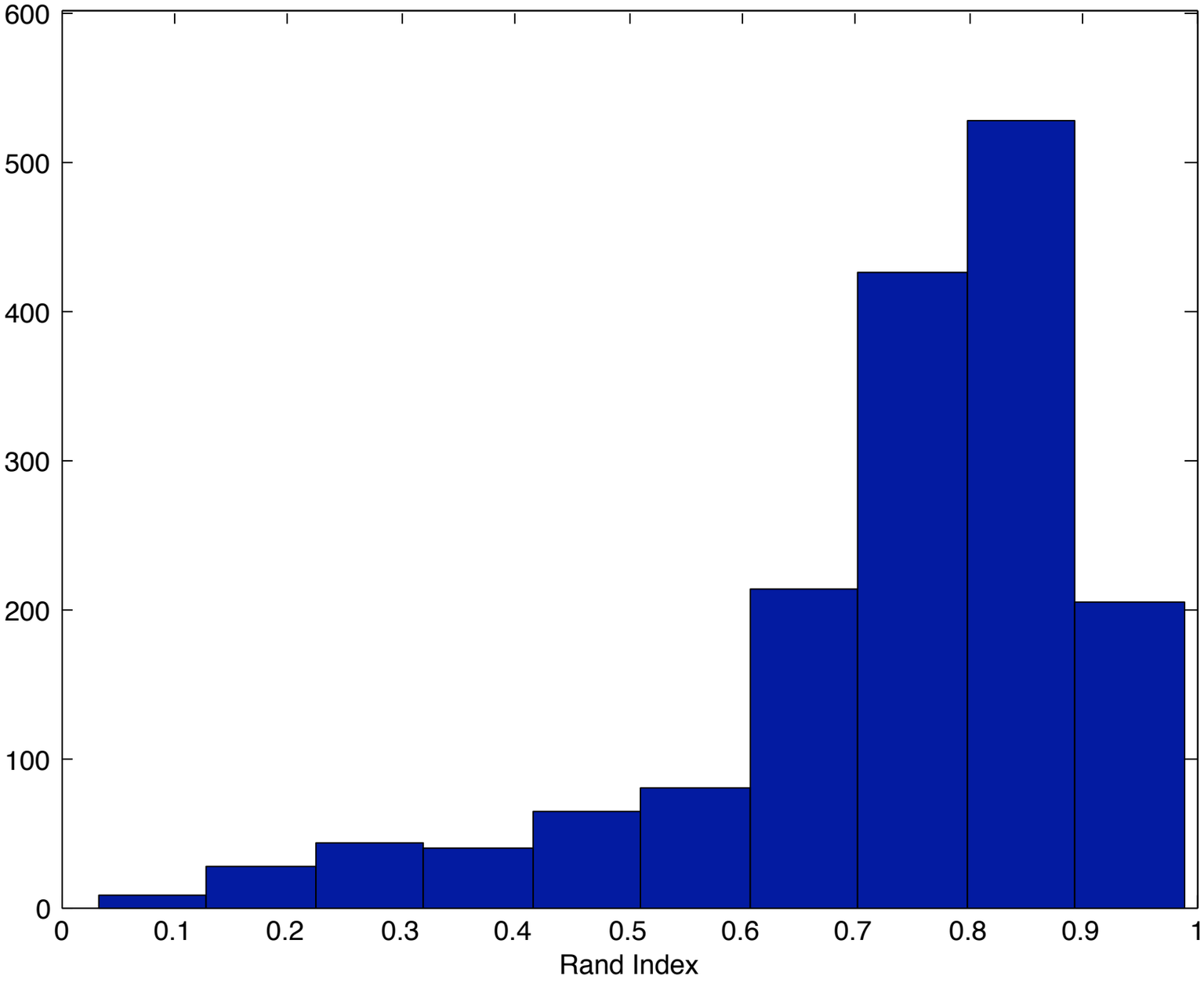
- Comparison to human segmentations in Berkeley Segmentation Database.
- Global Consistency Error (GCE)- considers the fraction of each segment that is labeled inconsistently
  - closer to 0 is better
  - [Martin, Fowlkes, Tal & Malik, 01]
- Rand Index - considers the fraction of pixel pairs that are labeled consistently
  - closer to 1 is better
  - [Rand, 71]
    - [Unnikrishnan, Pantofaru & Hebert, 07]

# Histogram of GCE values over the dataset



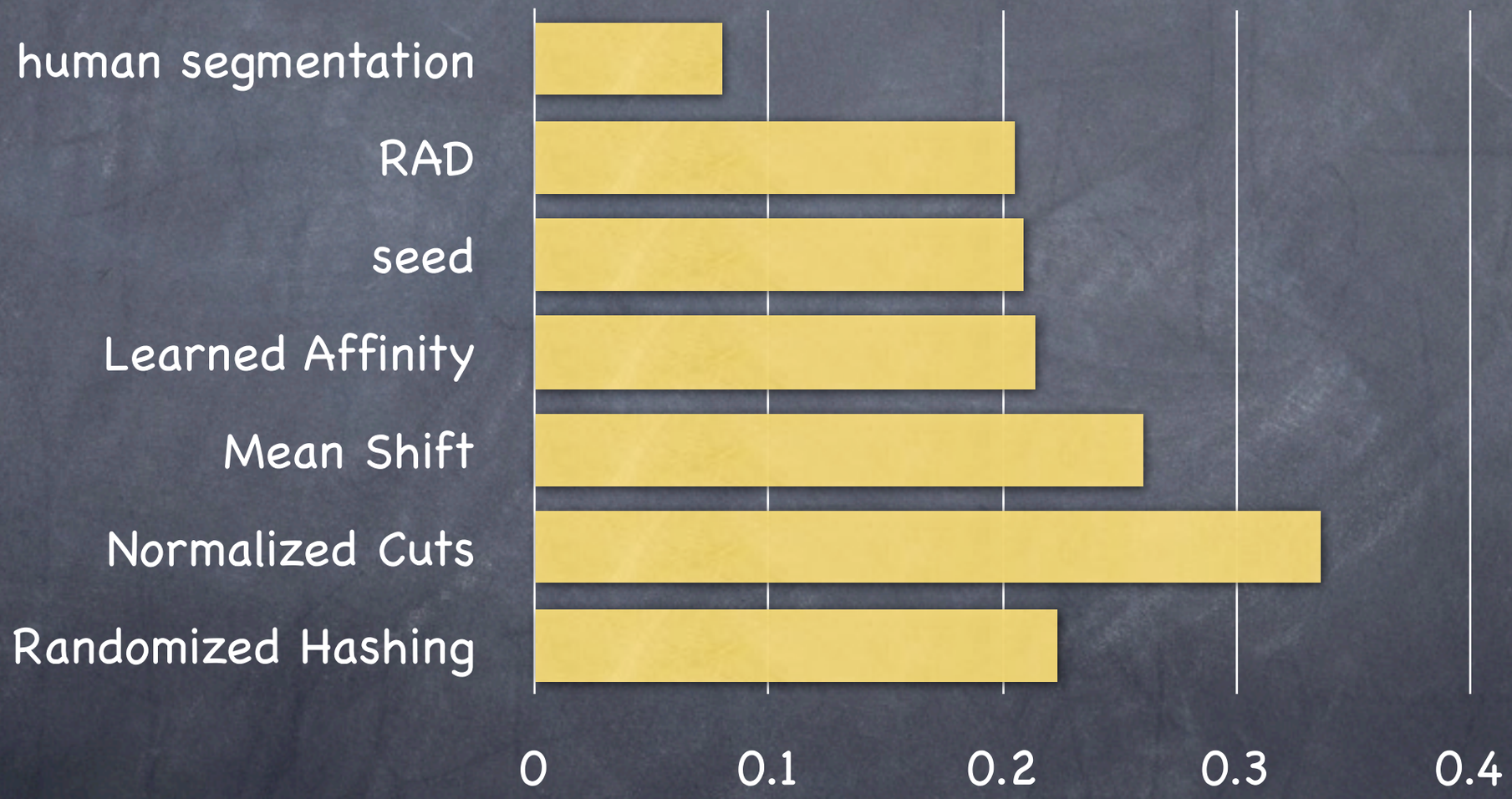

Mean GCE : 0.2235, Median GCE : 0.2157

# Histogram of Rand Index values over the dataset



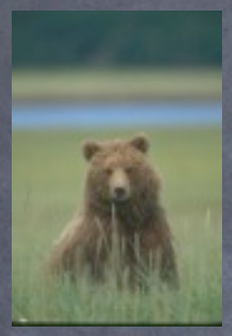
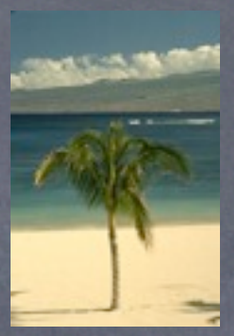
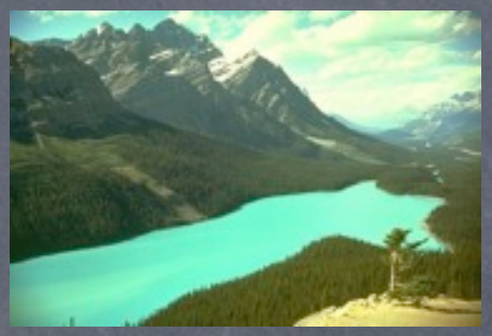
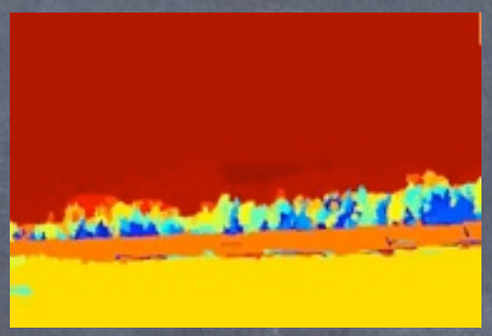
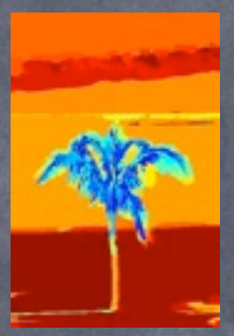
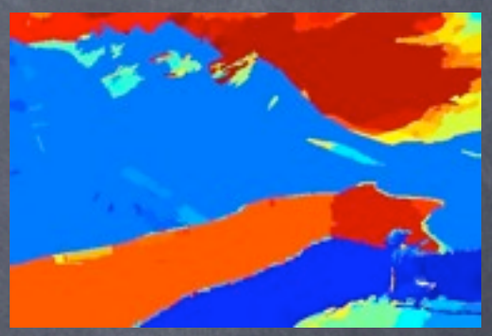
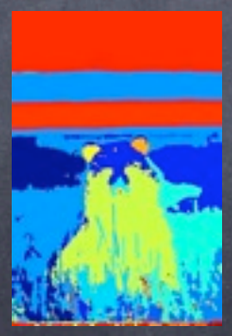
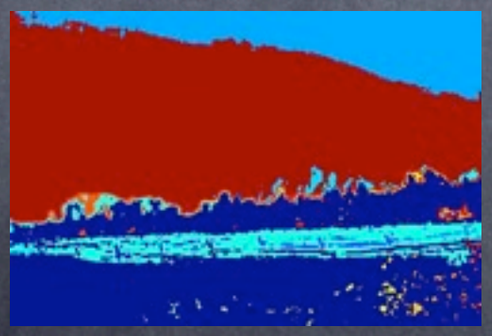
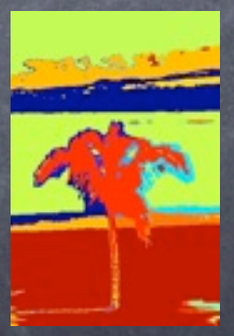
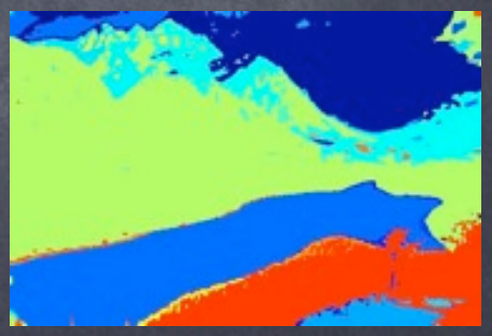

Mean Rand Index : 0.7370, Median RI : 0.7833

# Comparing reported GCE values on Berkeley dataset



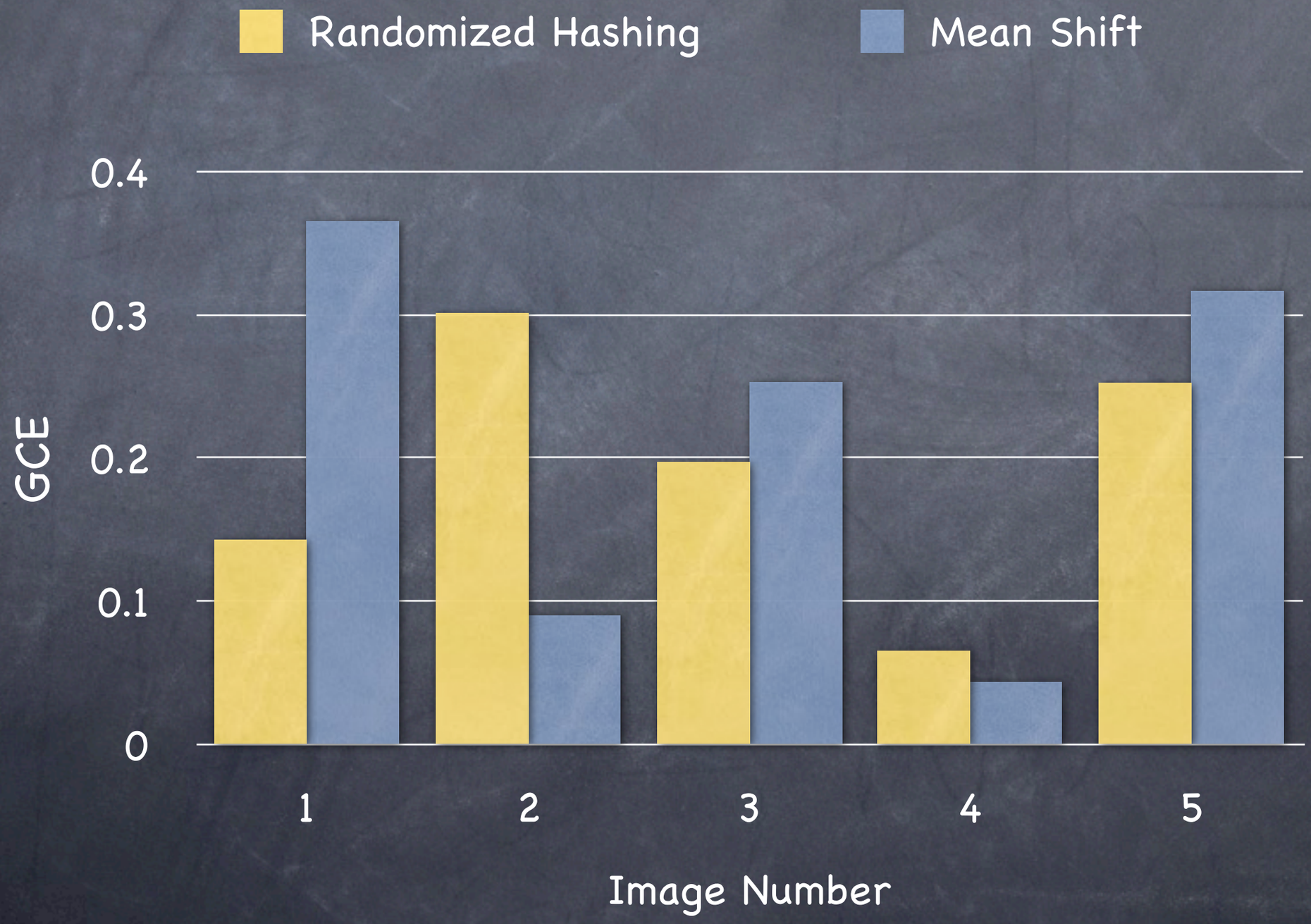

Data from [Vazquez, Weijer & Baldrich, 08]

# Comparison to Mean Shift

| | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Input |  |  |  |  |  |
| Mean Shift |  |  |  |  |  |
| Hashing |  |  |  |  |  |
| Image number | | | | | |

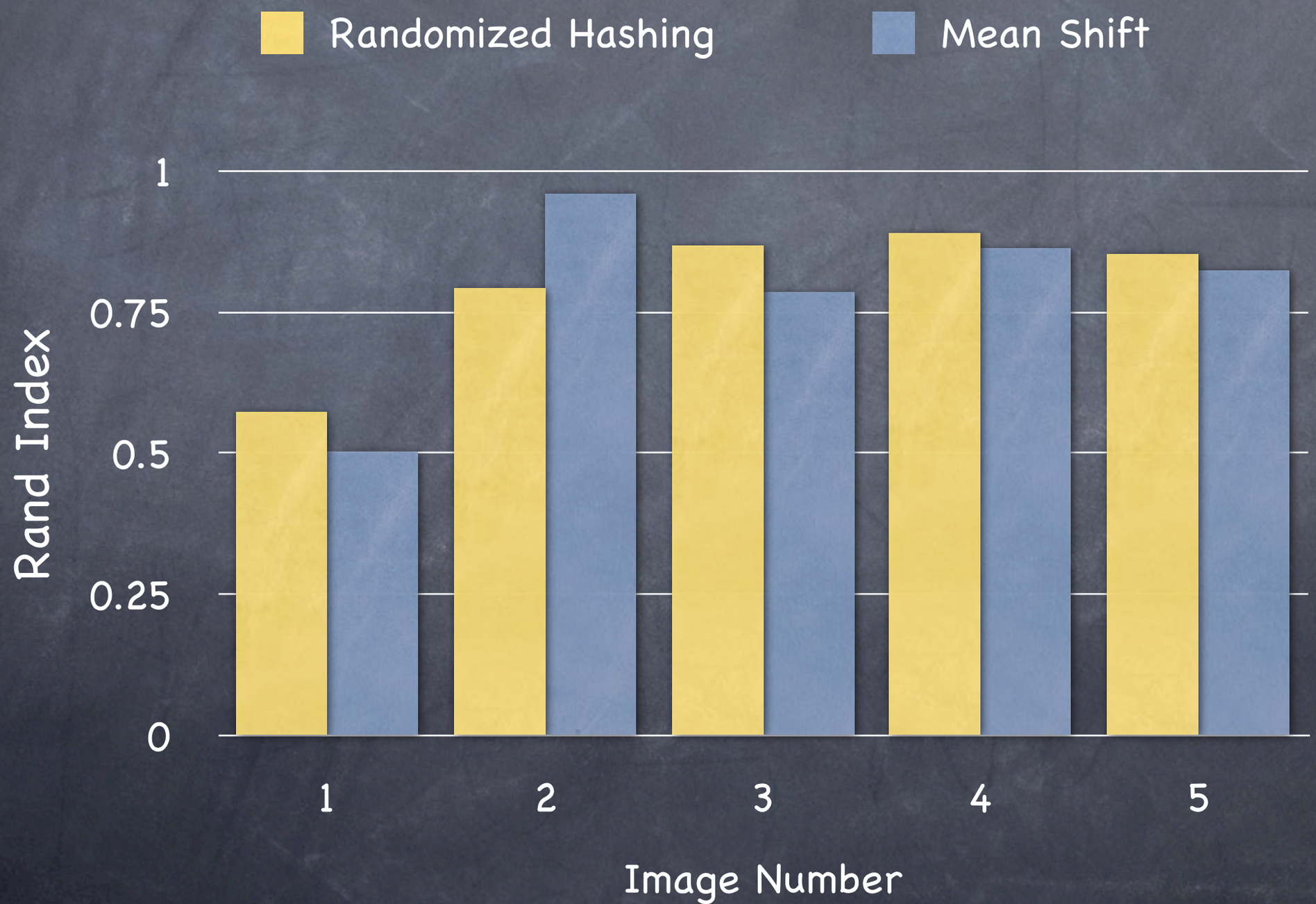
# Comparison to Mean Shift

## GCE



# Comparison to Mean Shift

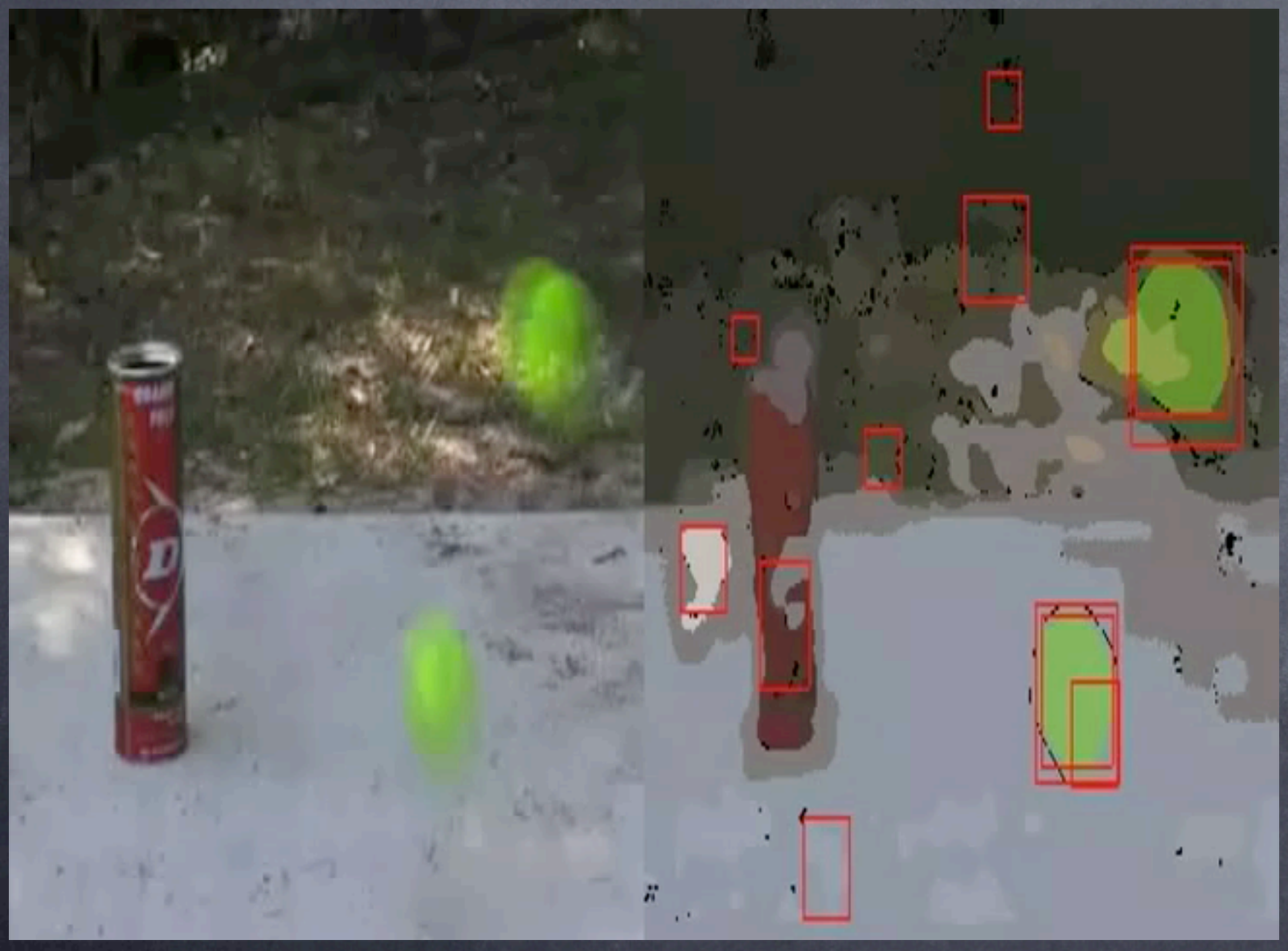
# Rand Index



# Implementation

- Scheme eschews distance computation in favor of hashing which simplifies control flow
- Almost all of the stages in the segmentation procedure are embarrassingly parallel and amenable to GPU implementation
- Implementation on one core of a 2.33 GHz Intel Core 2 Duo can run at 10Hz.

# Simple Ball Detector



- Segment each image in video independently
- Report all segments with appropriate size and aspect ratio.

# Live Demo
# (Always Dangerous)

# Future Work

- Experiments on texture features - higher dimensional vectors
- Applications to tracking?
- Tuning separation planes to content?

# Summary

- New approach to segmentation based on random hashing
- Accuracy is comparable with current methods but speed is at least 10x faster.
- Fast segmentation can be employed to suggest groupings of pixels to higher level interpretation processes.