# CS195: Computer Vision

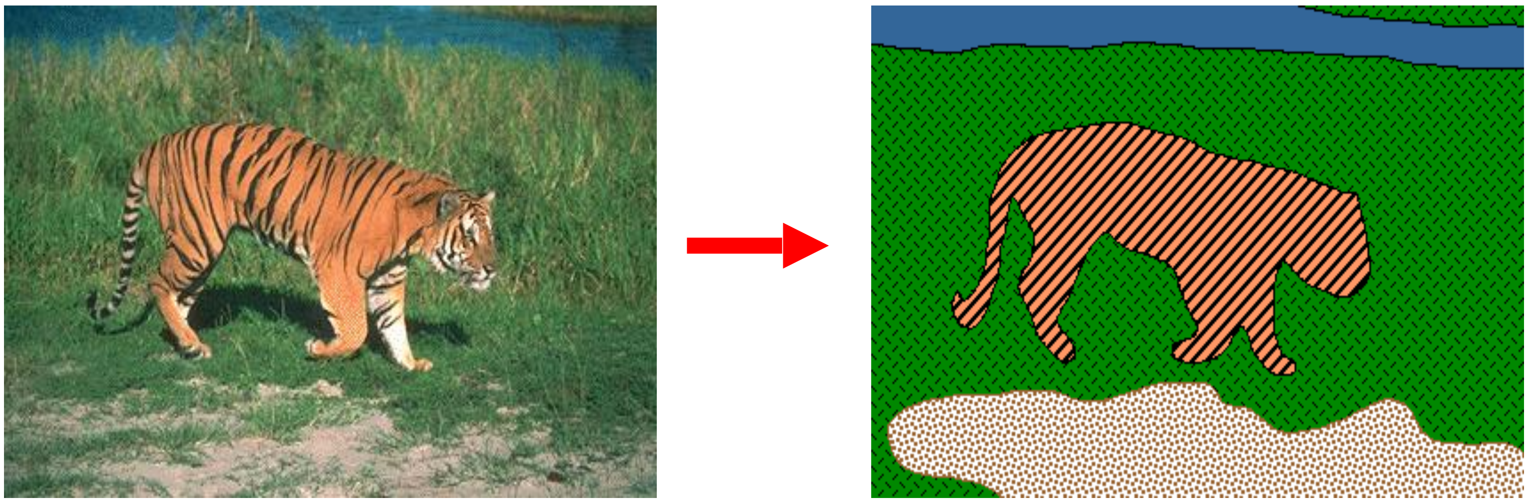
## Image Segmentation

Clustering-based segmentations (k-means, SLIC)

Wednesday, October 16th, 2024

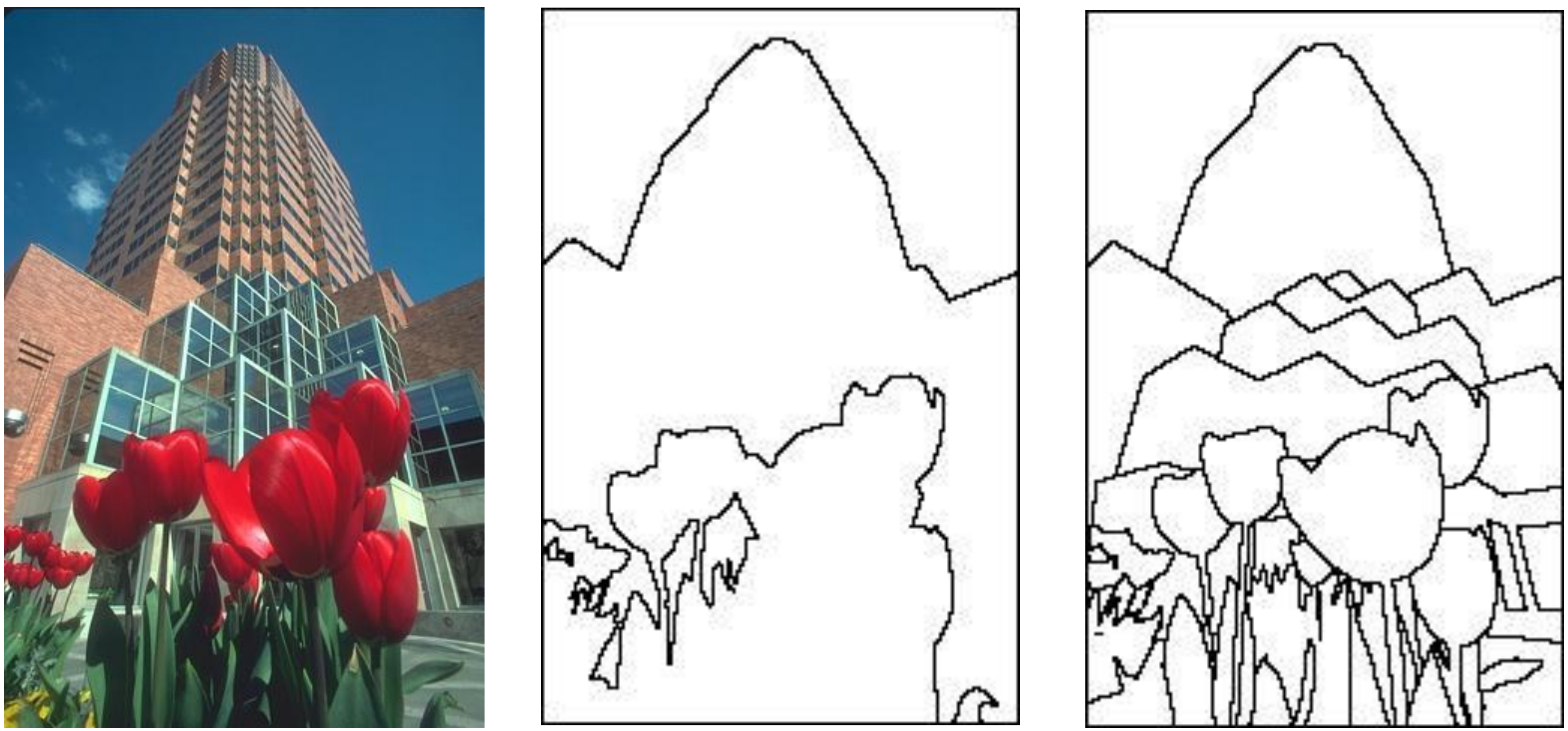
# Image segmentation

- Goal: break apart an image into simpler components



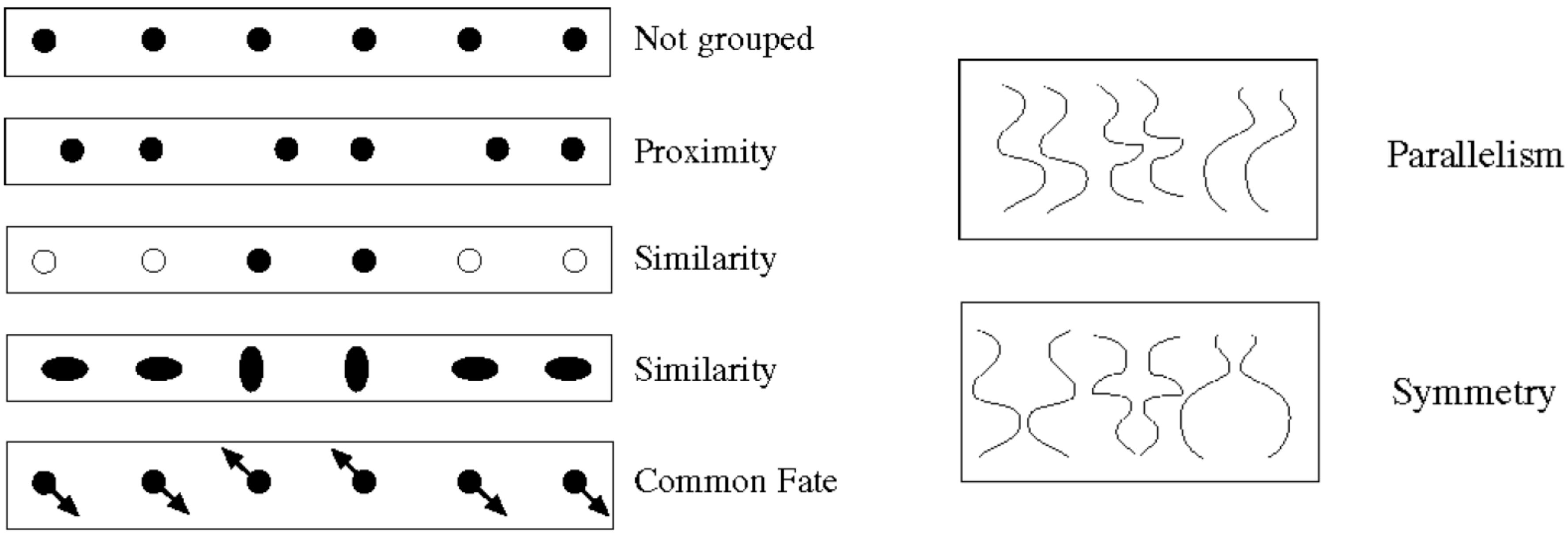
# Hard to judge success

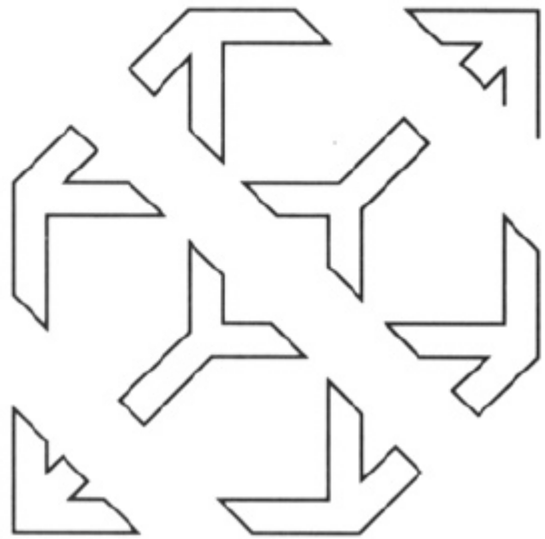
- Which of these segmentations is “correct”?

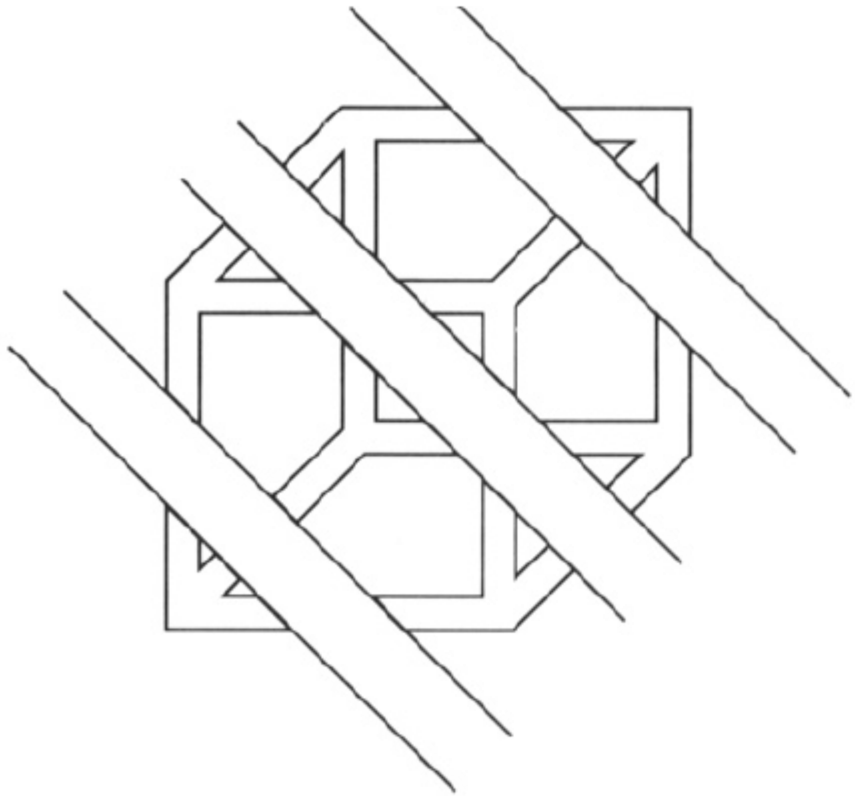


[Martin 2001]

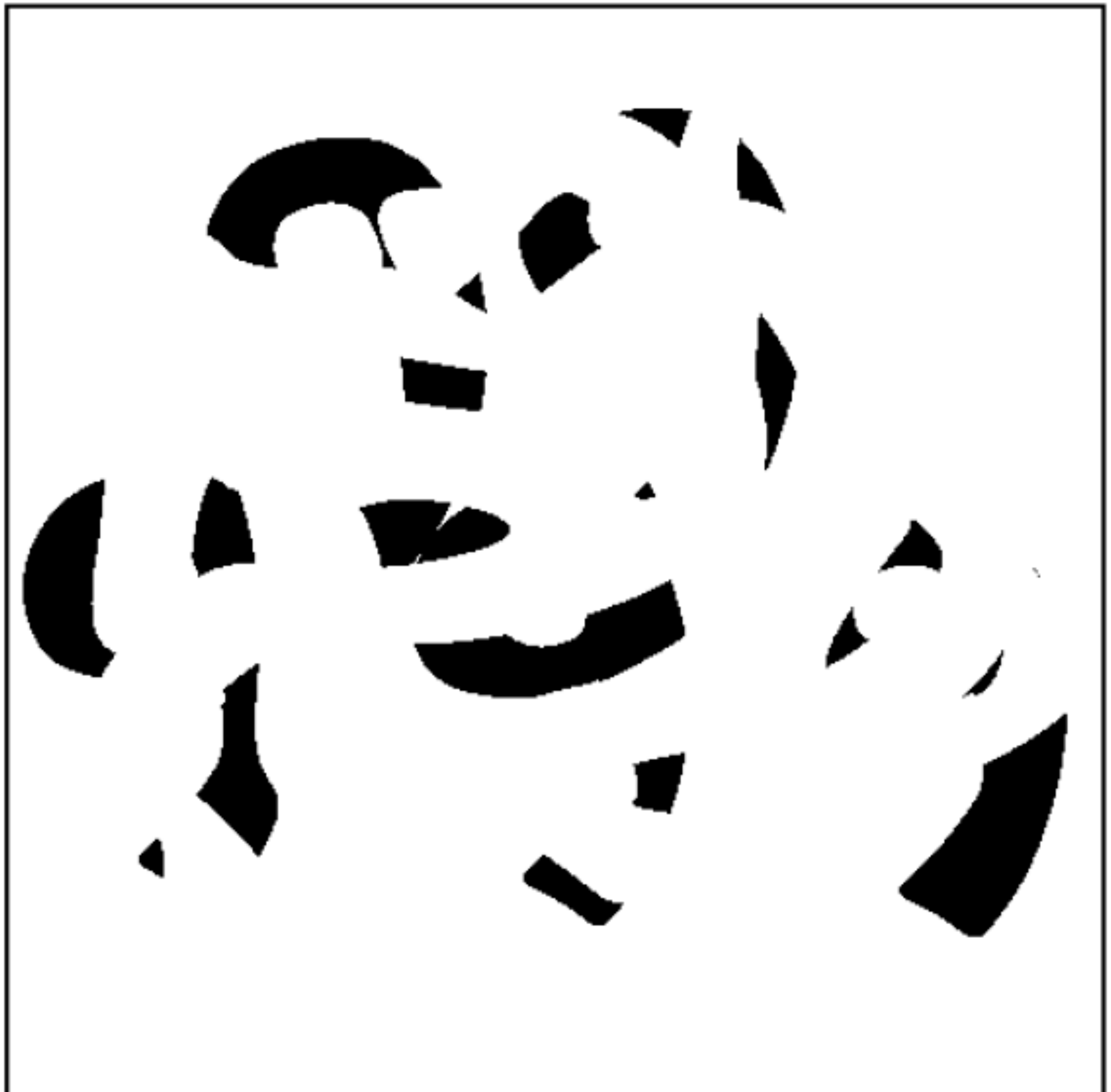
# Some Gestalt factors







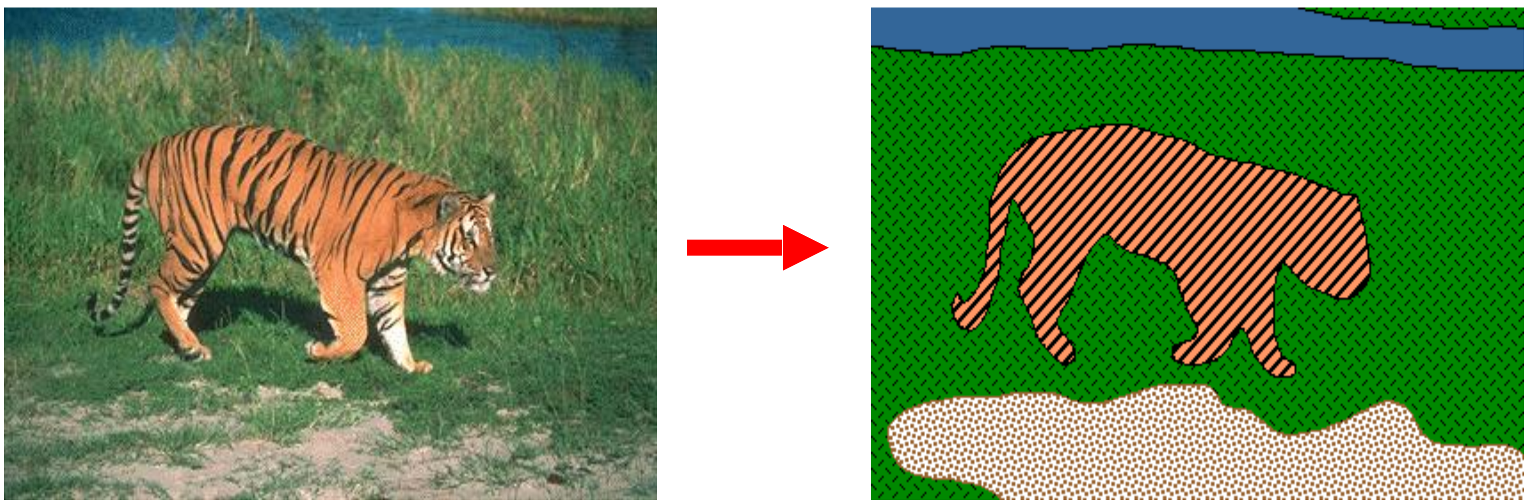
Continuity, explanation by occlusion



D. Forsyth

# Image segmentation

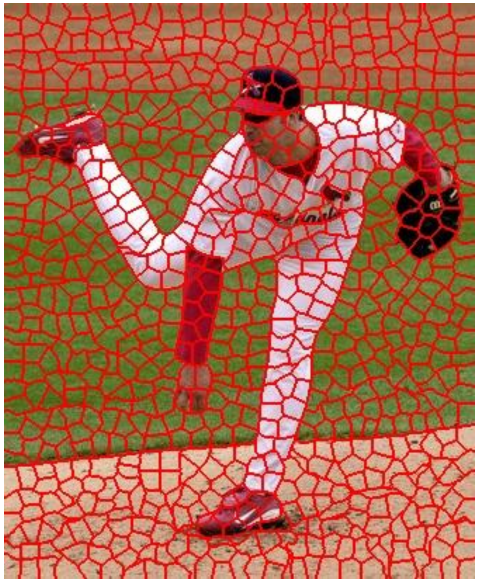
- Goal: identify groups of pixels that go together.

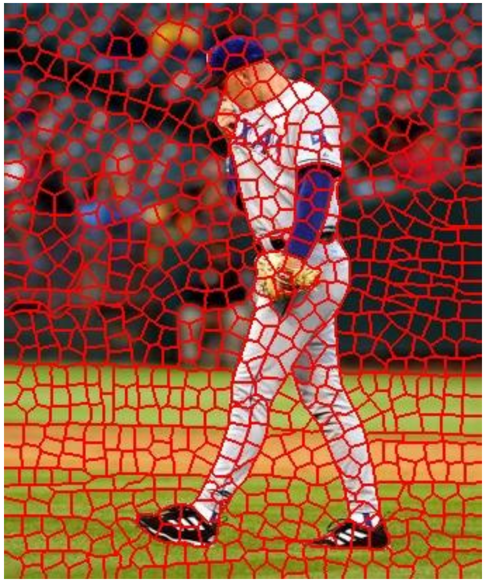


# The goals of segmentation

- Separate image into coherent “objects”
- Group together ***similar-looking*** pixels for efficiency of further processing
  - How can you measure similarity?

“superpixels”





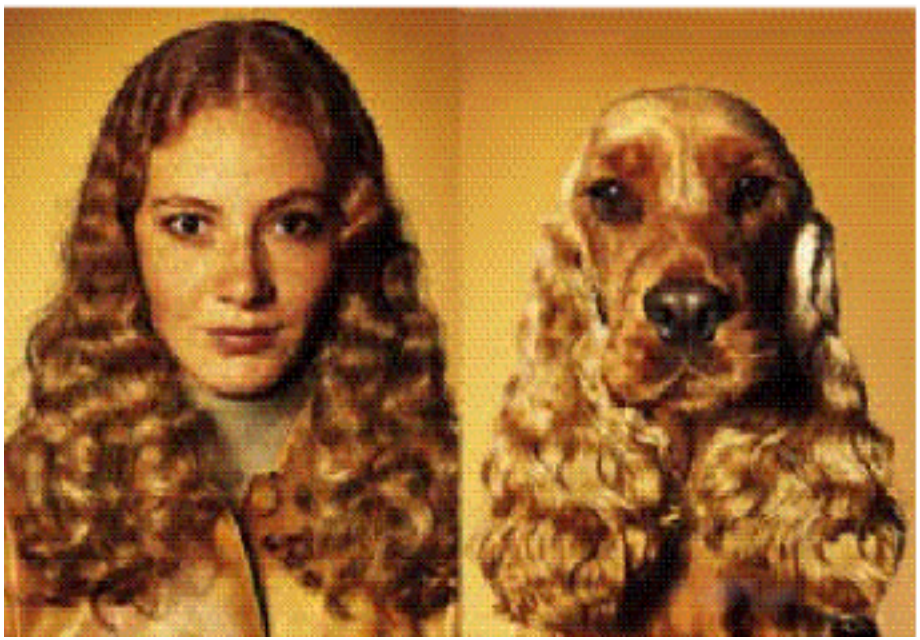
X. Ren and J. Malik. [Learning a classification model for segmentation.](#) ICCV 2003.

Source: Lana Lazebnik

# What is Similarity?

The quality or state of being similar; likeness; resemblance; as, a similarity of features.

**Webster's Dictionary**

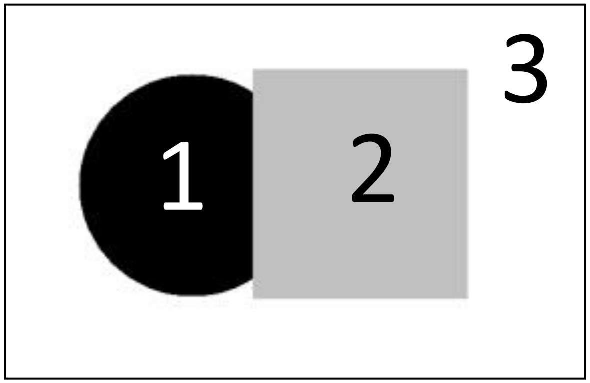


Similarity is hard to define, but…
*"We know it when we see it"*

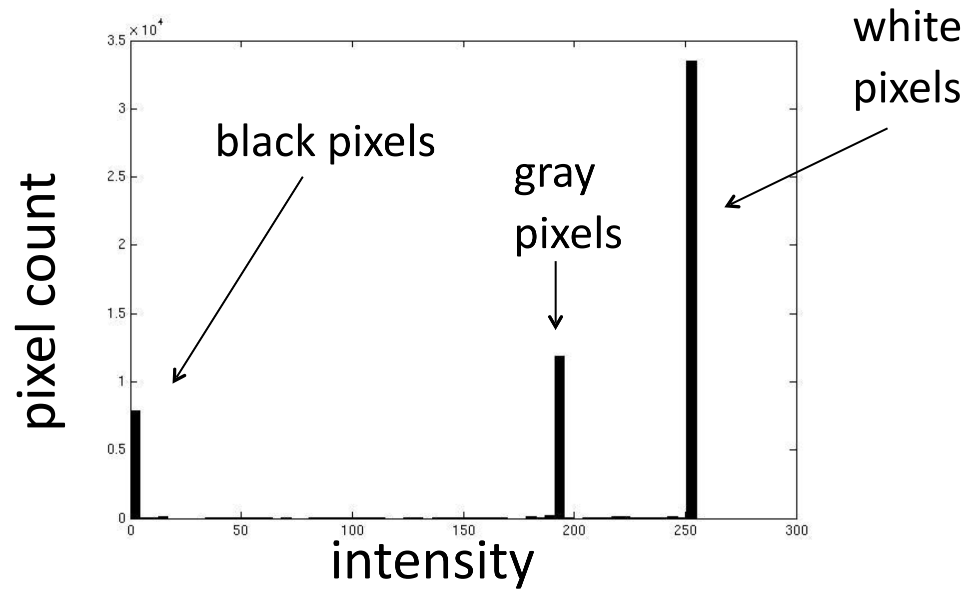
The real meaning of similarity is a philosophical question. We will take a more pragmatic approach.

Slide by E. Keogh

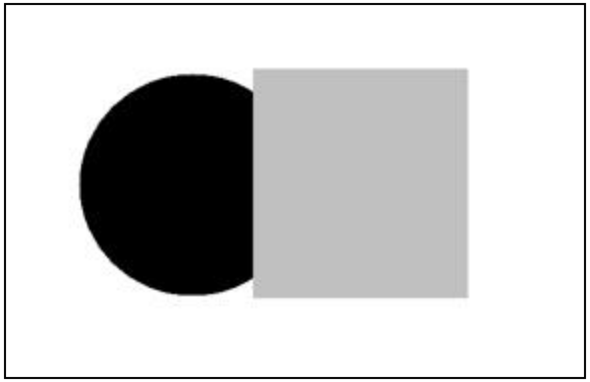
# Image segmentation: toy example



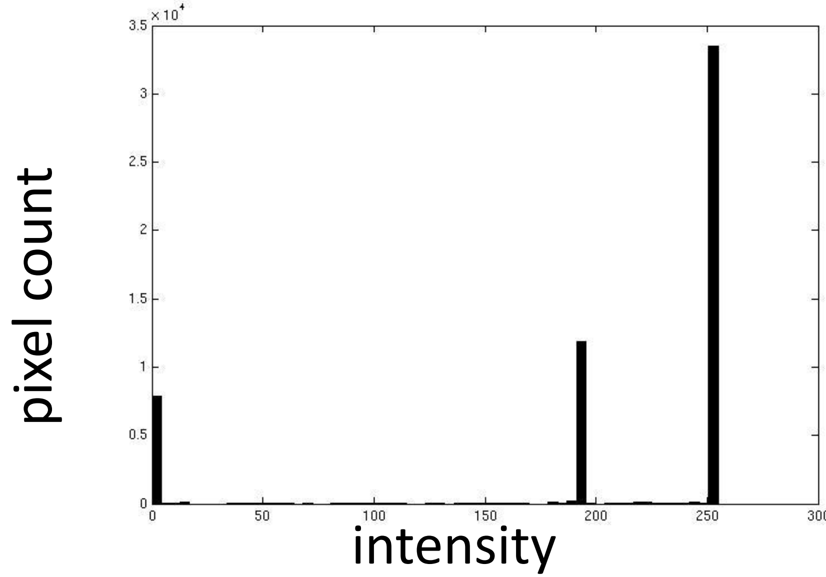

input image

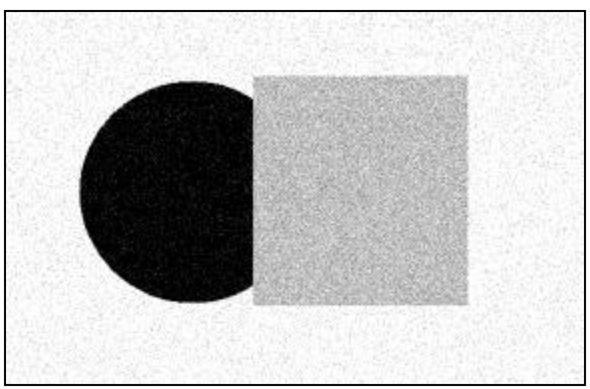



- Three intensities define the three groups.
- We could label every pixel in the image according to its greyscale value.

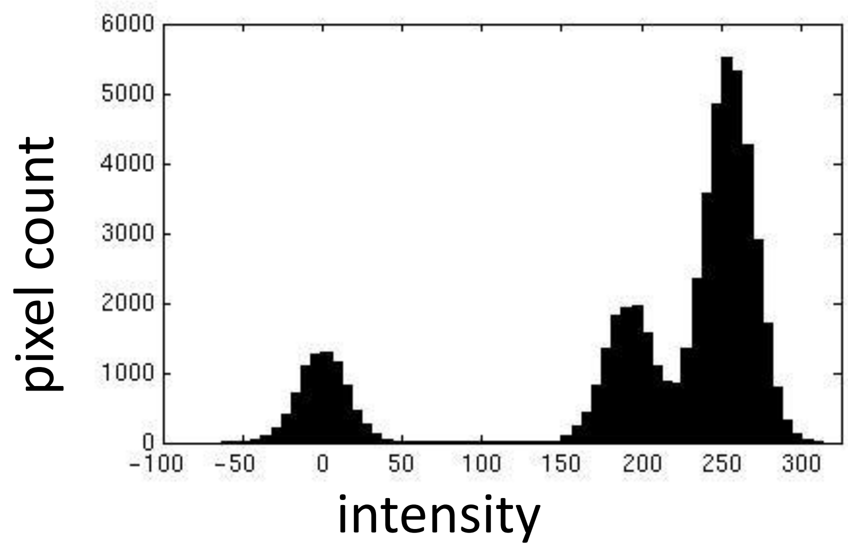


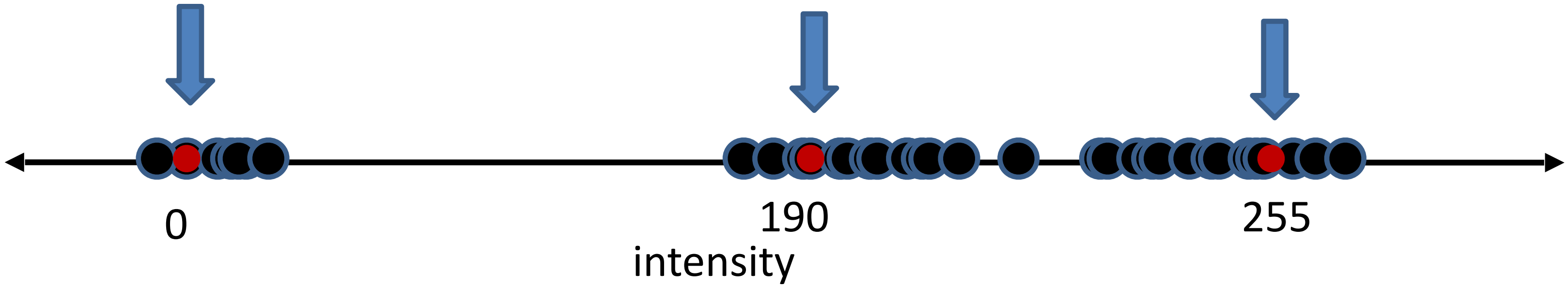
input image

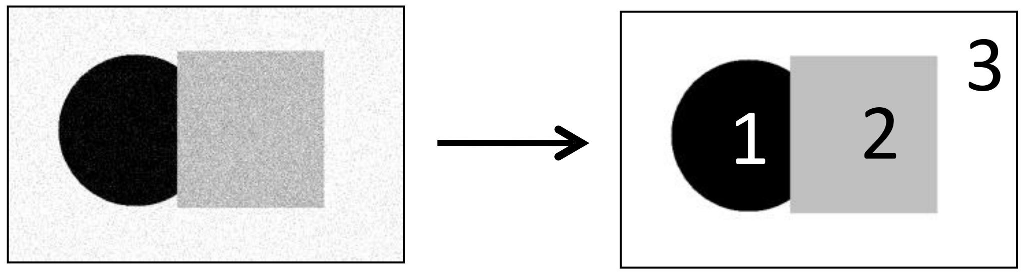


input image





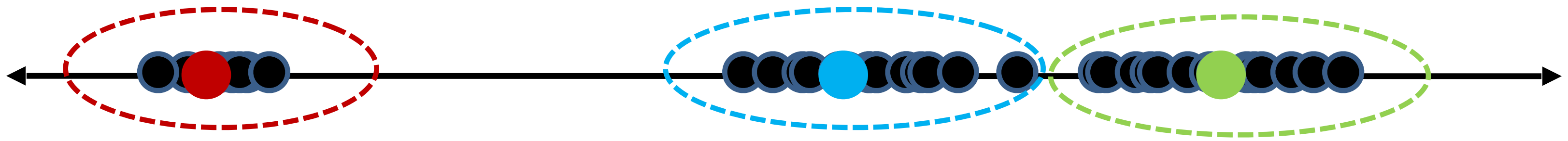

- Goal: choose 3 “centers” as representative intensities, label each pixel according to which center is nearest.
- Best cluster centers minimize sum squared difference between each point and its nearest cluster center $c_i$:

$$\text{minimize} \sum_{\text{clusters } i} \sum_{\text{points p in cluster } i} \|p - c_i\|^2$$

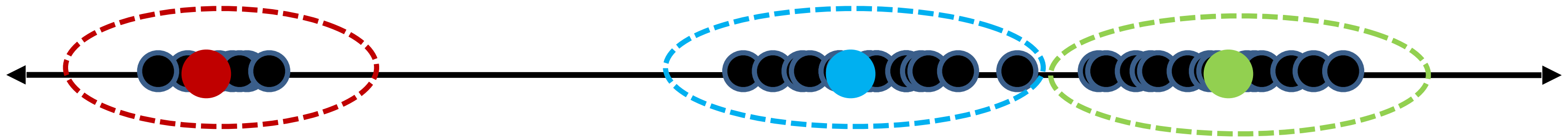
Similarity measure: Euclidean distance

# A chicken and egg problem!

– If we knew the **cluster centers**, we could allocate points to groups by assigning each to its closest center.



– If we knew the **group memberships**, we could get the centers by computing the mean per group.



# K-means clustering

- Basic idea: randomly initialize the *k* cluster centers, and iterate between the two steps we just saw.

1. Randomly initialize the cluster centers, $c_1, \ldots, c_K$
2. Given cluster centers, determine points in each cluster
   - For each point p, find the closest $c_i$. Put p into cluster i
3. Given points in each cluster, solve for $c_i$
   - Set $c_i$ to be the mean of points in cluster i
4. If $c_i$ have changed, repeat Step 2

Properties

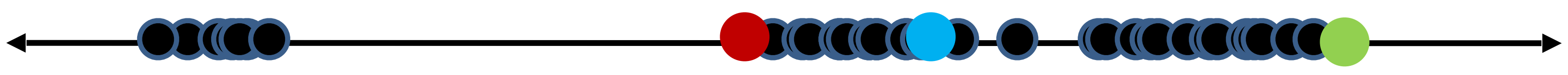
- Will always converge to *some* solution
- Might be a local minimum; i.e. not a global minimum of:

$$\sum_{\text{clusters } i} \sum_{\text{points p in cluster } i} ||p - c_i||^2$$

Source: Steve Seitz

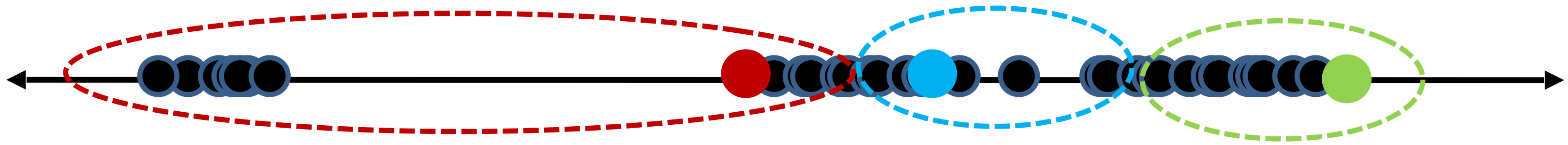
# K-means

1. Randomly initialize the cluster centers, $c_1, \ldots, c_K$



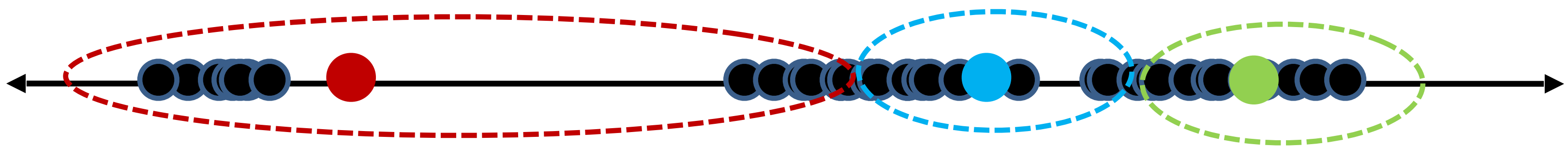
# K-means

1. Randomly initialize the cluster centers, $c_1, \ldots, c_K$
2. Given cluster centers, determine points in each cluster



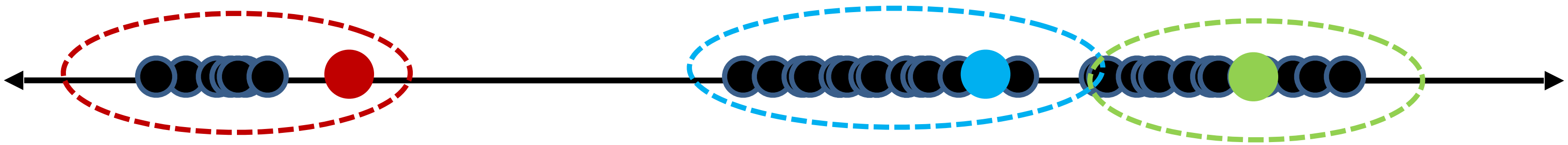
# K-means

1. Randomly initialize the cluster centers, $c_1, \ldots, c_K$
2. Given cluster centers, determine points in each cluster
3. Given points in each cluster, solve for $c_i$



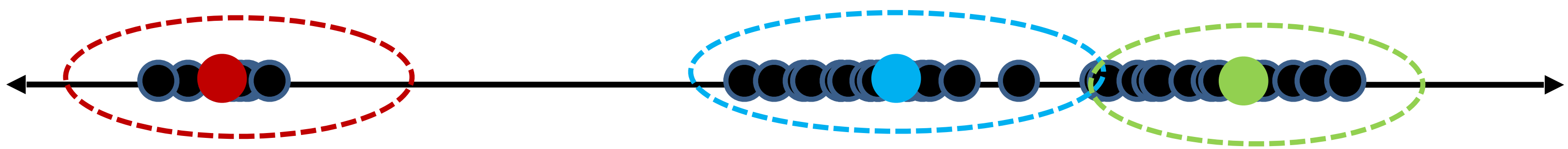
# K-means

1. Randomly initialize the cluster centers, $c_1, \ldots, c_K$
2. Given cluster centers, determine points in each cluster
3. Given points in each cluster, solve for $c_i$



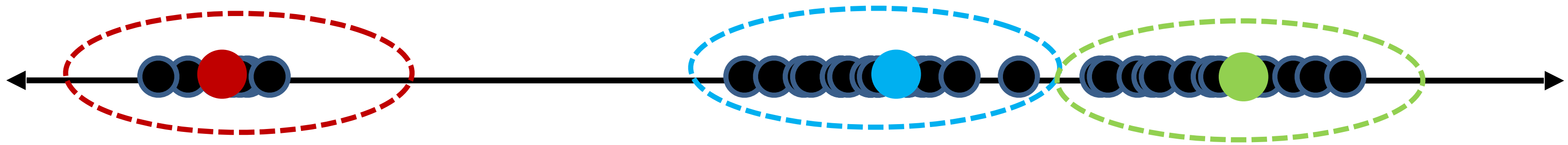
# K-means

1. Randomly initialize the cluster centers, $c_1, \ldots, c_K$
2. Given cluster centers, determine points in each cluster
3. Given points in each cluster, solve for $c_i$



# K-means

1. Randomly initialize the cluster centers, $c_1$, ..., $c_K$
2. Given cluster centers, determine points in each cluster
3. Given points in each cluster, solve for $c_i$

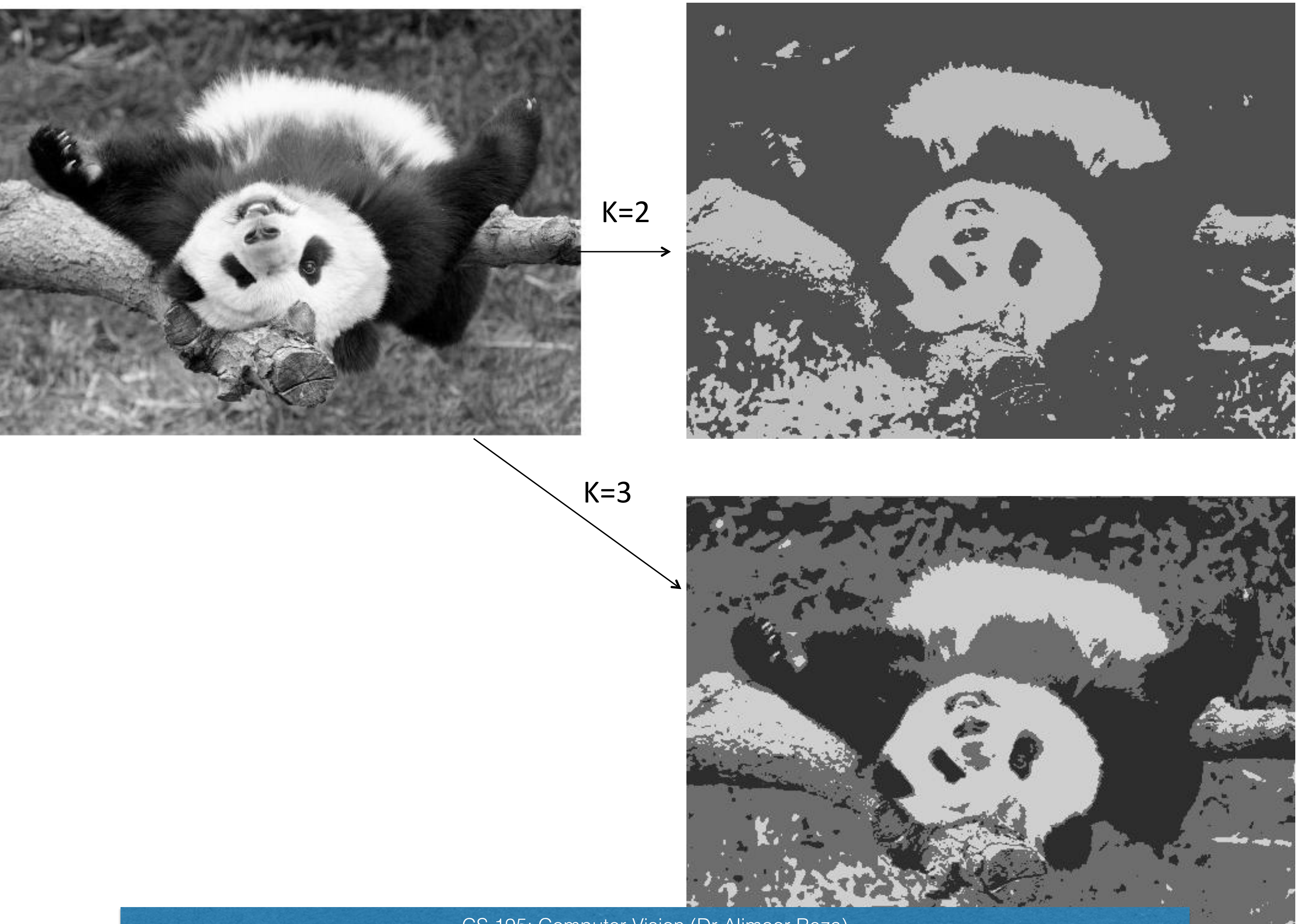


If we run it for enough iterations, it will converge to this state, when the centers won't be changing anymore. Clustering mechanism stops.

# Segmentation as clustering

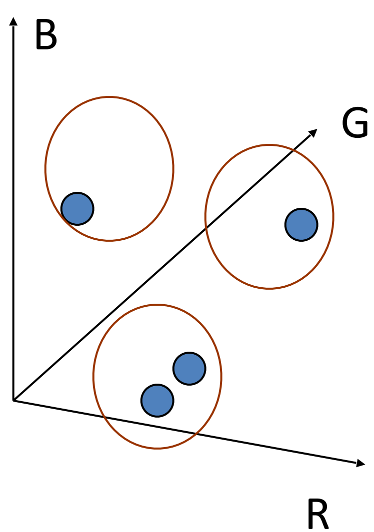
We can customize the clustering by changing the feature space.

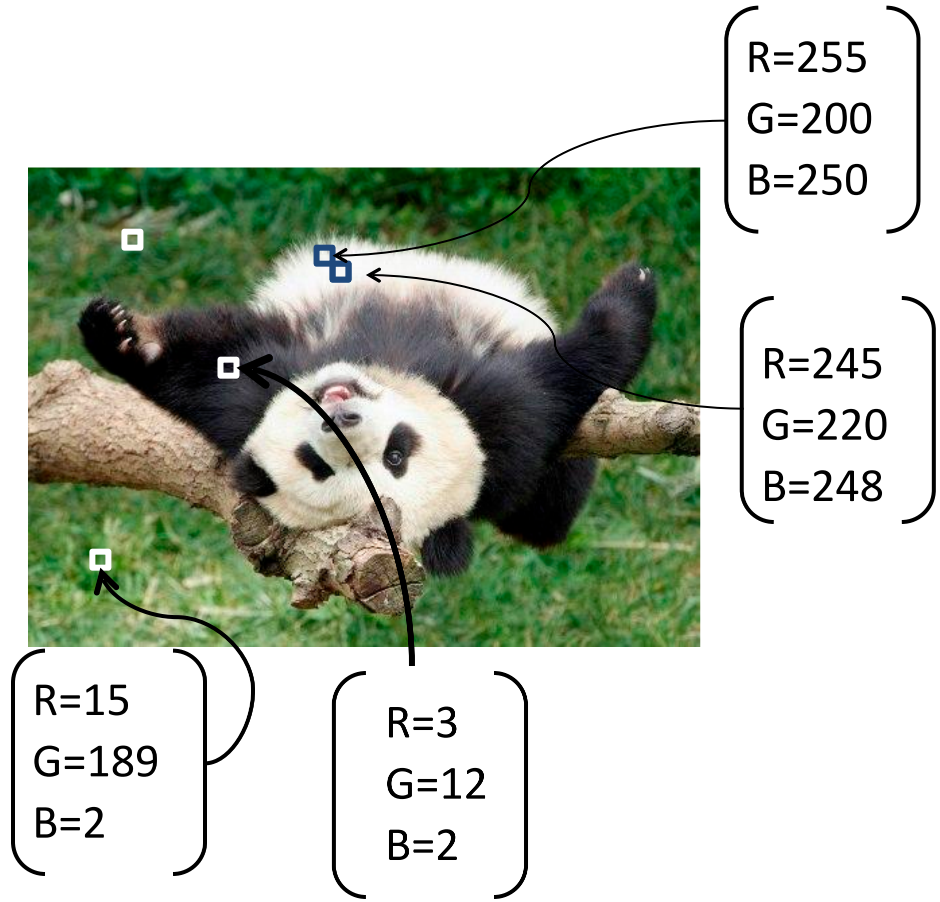
e.g. Grouping pixels based on intensity similarity


K=2
K=3

# Segmentation as clustering

e.g. Grouping pixels based on color similarity

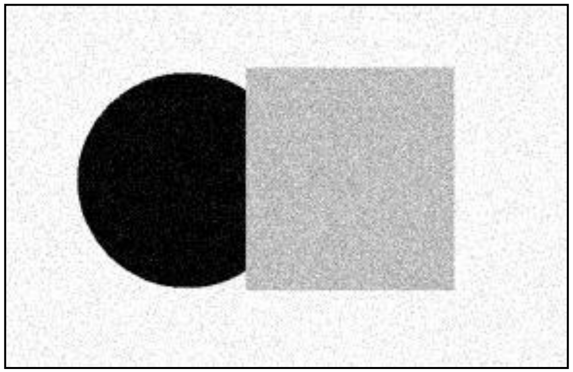


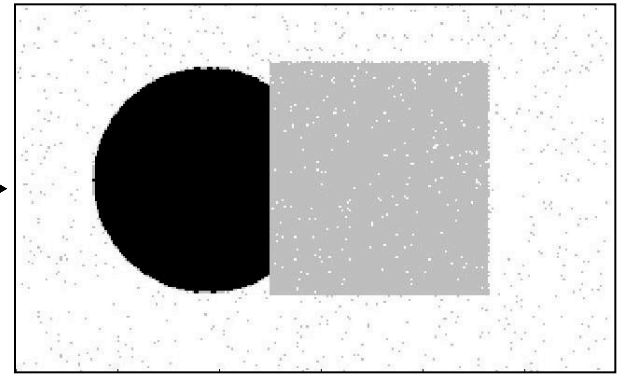

Feature space: color value (3-d)

# Smoothing out cluster assignments

- Assigning a cluster label per pixel may yield outliers:

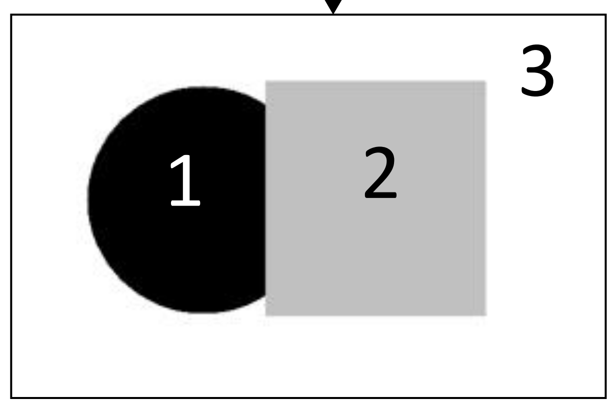


original



labeled by cluster center's intensity

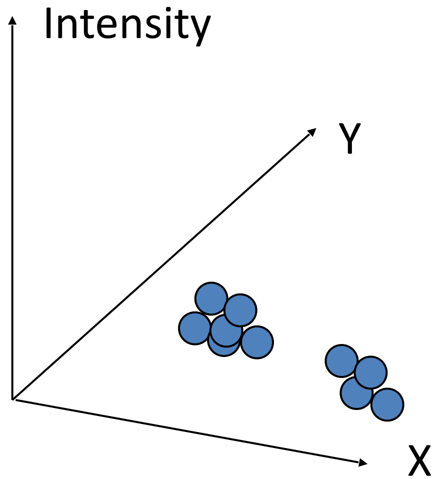
?

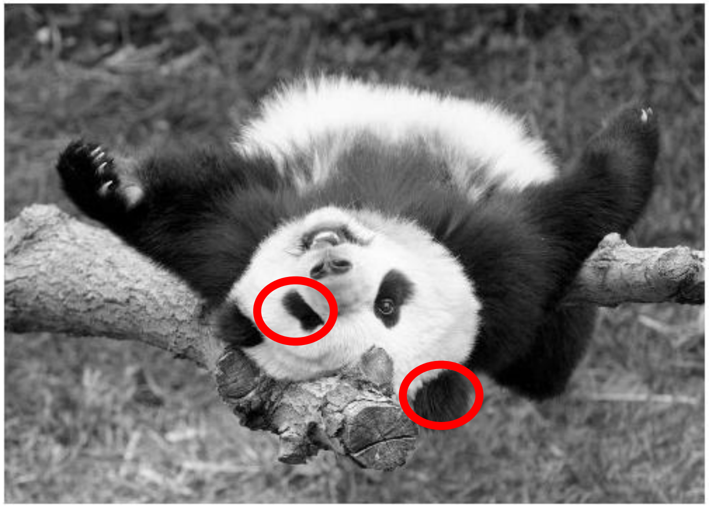


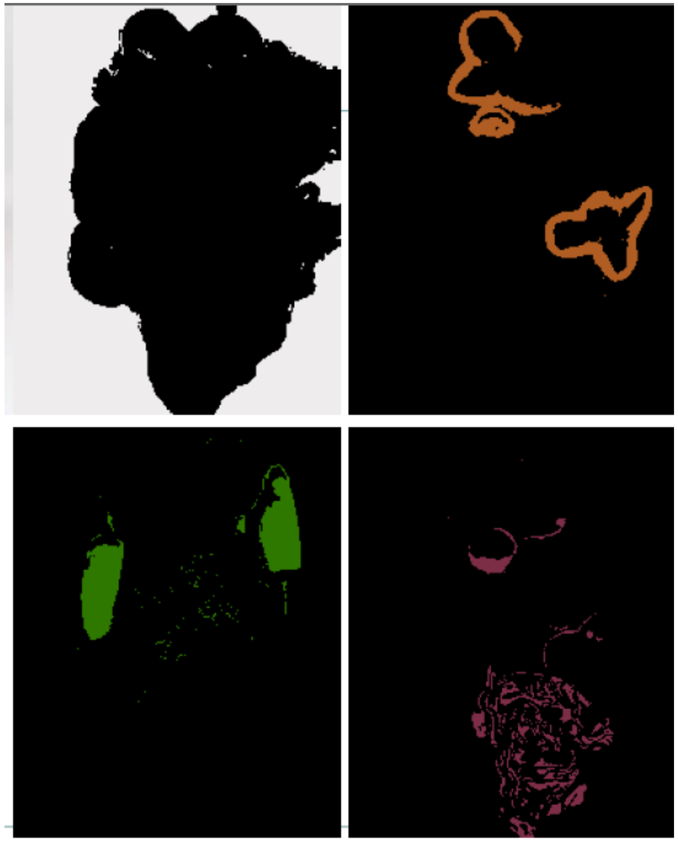

- How to ensure they are spatially smooth?

# Segmentation as clustering

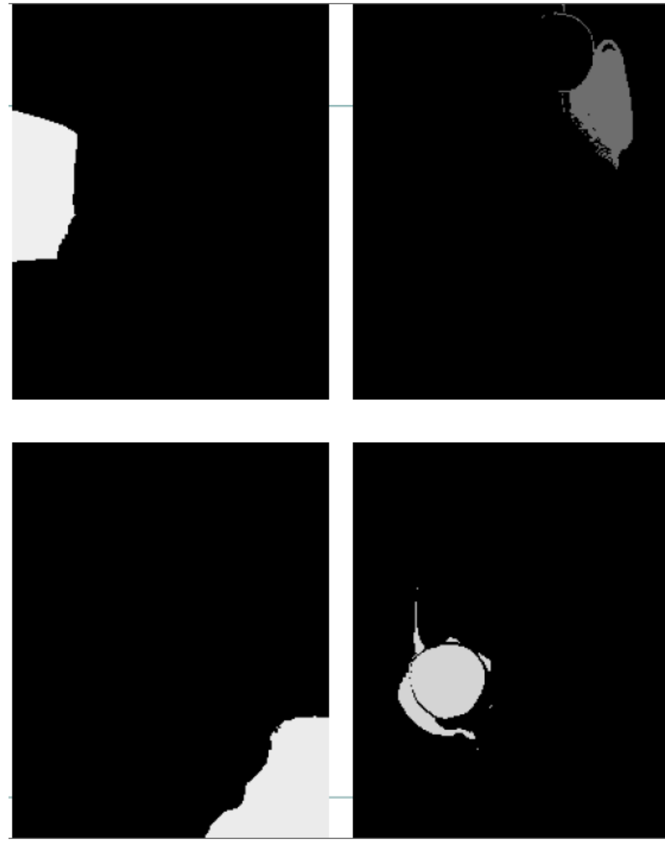
e.g. Grouping pixels based on intensity+position similarity





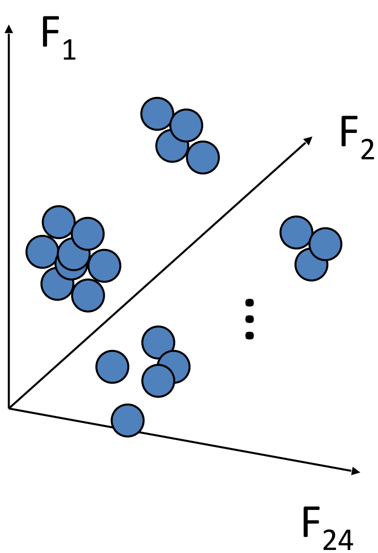
Clustering pixels based on color alone

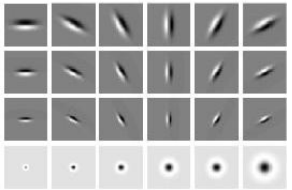


Clustering pixels based on color and position

# Segmentation as clustering

e.g. Grouping pixels based on texture similarity



Filter bank of 24 filters

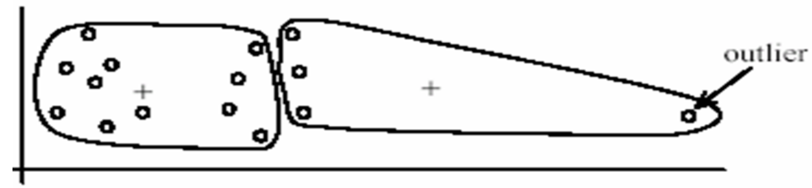
# K-means: pros and cons

## Pros

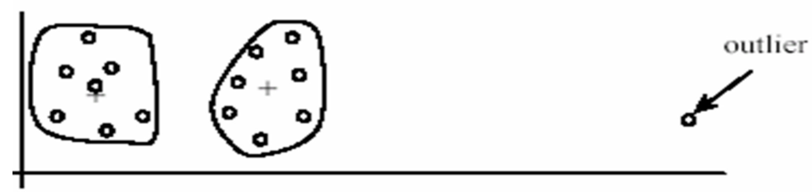
- Simple, fast to compute
- Converges to local minimum of within-cluster squared error

## Cons/issues

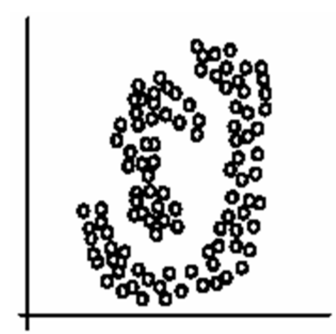
- Setting k?
- Sensitive to initial centers
- Sensitive to outliers
- Detects spherical clusters



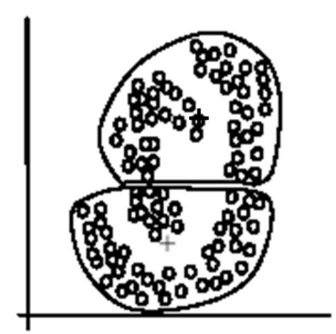

(A): Undesirable clusters




(B): Ideal clusters



(A): Two natural clusters



(B): *k*-means clusters

# Mean Shift: Object Proposal Generation in 3D point cloud

## Multiview RGB-D Dataset for Object Instance Detection

Georgios Georgakis, Md Alimoor Reza, Arsalan Mousavian, Phi-Hung Le, Jana Košecká
Department of Computer Science
George Mason University
{ggeorgak,mreza,amousavi,ple13,kosecka}@gmu.edu



### Abstract

*This paper presents a new multi-view RGB-D dataset of nine kitchen scenes, each containing several objects in realistic cluttered environments including a subset of objects from the BigBird dataset [26]. The viewpoints of the scenes are densely sampled and objects in the scenes are annotated with bounding boxes and in the 3D point cloud. Also, an approach for detection and recognition is presented, which is comprised of two parts: i) a new multi-view 3D proposal generation method and ii) the development of several recognition baselines using AlexNet [14] to score our proposals, which is trained either on crops of the dataset or on synthetically composited training images. Finally, we compare the performance of the object proposals and a detection baseline to the Washington RGB-D Scenes (WRGB-D) dataset [15] and demonstrate that our Kitchen scenes dataset is more challenging for object detection and recognition. The dataset is available at:* `http://cs.gmu.edu/~robot/gmu-kitchens.html`.

### 1. Introduction

The problem ...

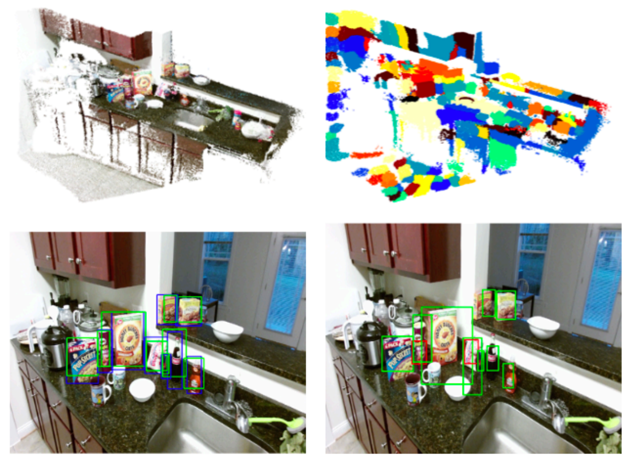


Figure 1: Top left: Example reconstructed scene from the Kitchen scenes dataset. Top right: Generation of 3D object proposals using a simple clustering algorithm. Bottom left: Proposals projected on a 2D image in the scene. For clarity only proposals with high overlap with the ground truth are shown. Bottom right: Object recognition using our projected proposals. The blue bounding boxes are the ground truth, the green are correct detections, and the red are false detections.

Multiview RGB-D dataset for object instance detection
G Georgakis, MA Reza, A Mousavian, PH Le, J Košecká
2016 Fourth International Conference on 3D Vision (3DV), 426-434

# Mean Shift: Object Proposal Generation in 3D point cloud

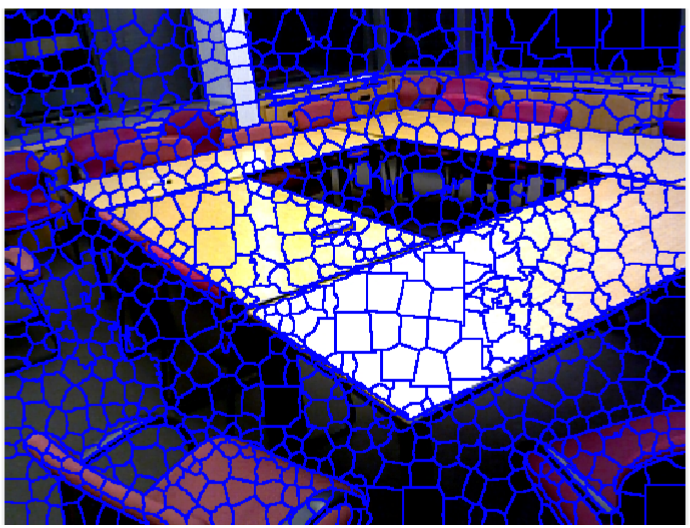

Figure 4: 3D Object proposal generation: (a) Dense point cloud of a scene; (b) output of the plane detection algorithm; (c) output of the Mean Shift clustering given $radius = 0.4$ after removing 33% of the large planes; (d) extracted 3D proposals after cuboid fitting.

# More Clustering Based Segmentation

# Simple Linear Iterative Clustering (SLIC)

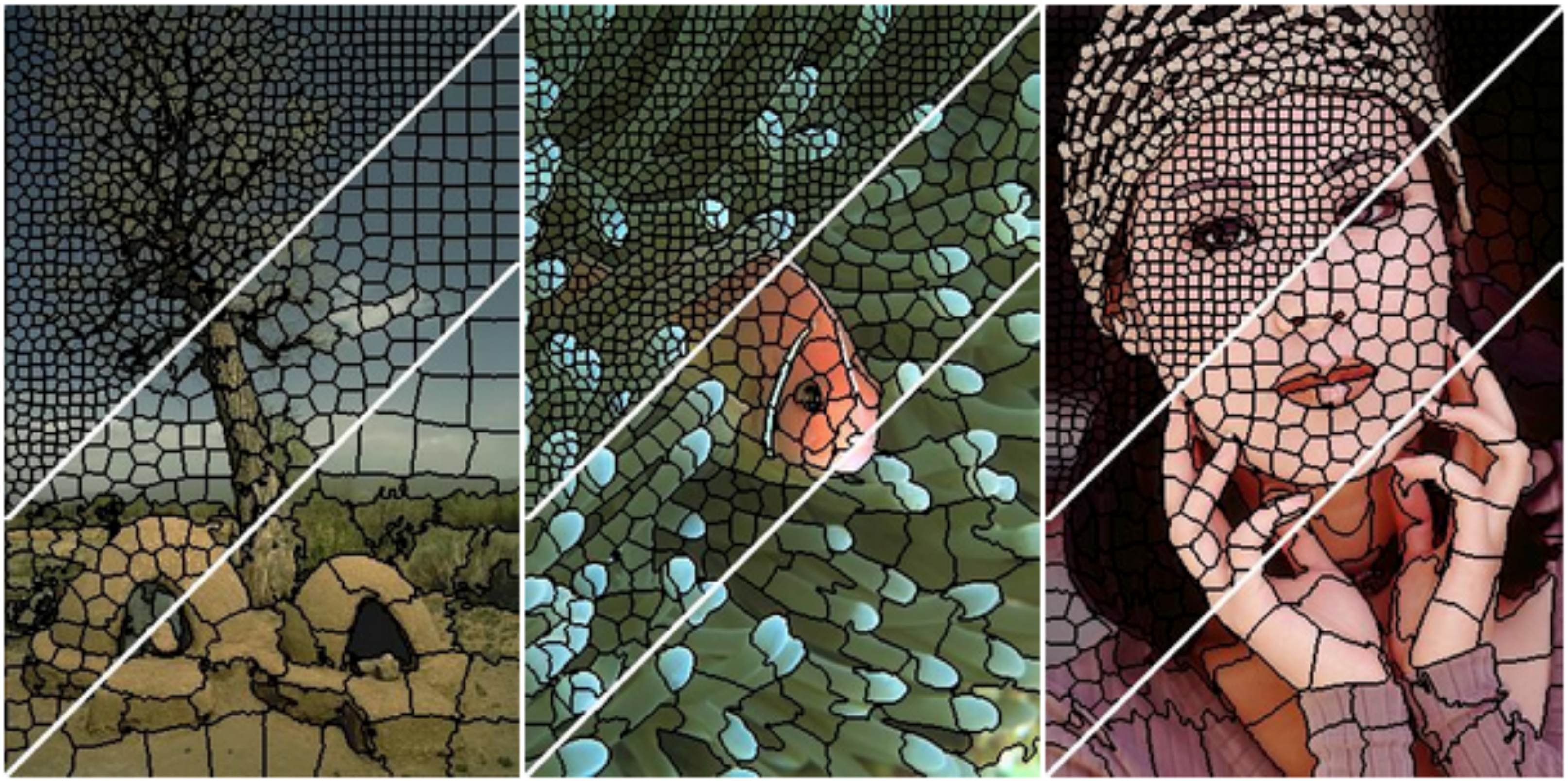
- SLIC clusters pixels in the combined five-dimensional color and image plane space to generate compact, nearly uniform segments (also called superpixels)
- Very fast, can generate superpixels in less than a second.

RGB



SLIC superpixels

# Simple Linear Iterative Clustering (SLIC)

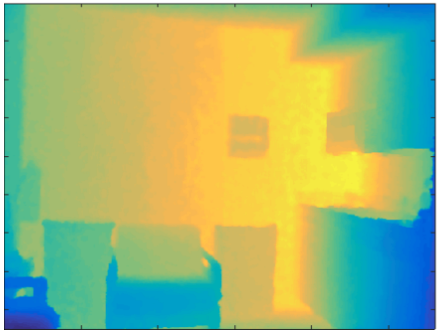


SLIC superpixels with different parameters

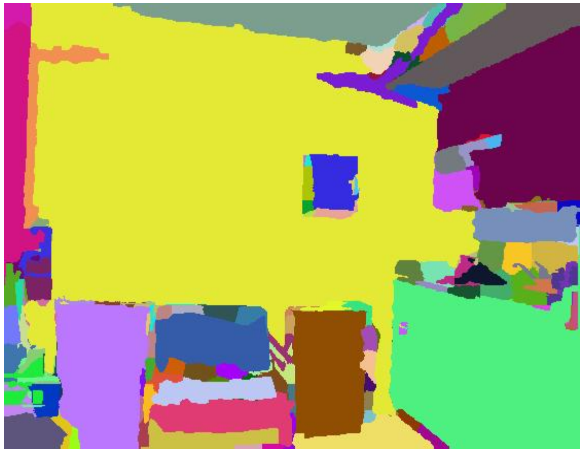
# Segmentation by Reza et al. RSSw'2014

- Utilize the depth information to find dominant planar surfaces by fitting planes[1] from 3D points
- represent remaining as compact[2] regions in 2D image space

SLIC segmentation on RGB image



plane extraction from depth image



Color coded segmentation

**M. Reza** and J. Kosecka, "Object Recognition and Segmentation in Indoor Scenes from RGB-D Images", **RSS-W-2014**

# Activity: Clustering-based Segmentation

## SLIC clustering based algorithm

- n_segment -- (approximate) number of labels in the segmented output image.
- compactness -- balances color proximity and space proximity

```
[ ]  from skimage.segmentation import slic
     #img = skimage.io.imread('data/'+image_names[1])
     img = coffee()
     # --------------- SLIC ---------------------
     segments = slic(img, n_segments=200, compactness=10, sigma=1)
     plt.imshow(np.asarray(segments))
     plt.imshow(mark_boundaries(img, segments))
     plt.title('SLIC clustering-based segmentation')
```

```
Text(0.5, 1.0, 'SLIC clustering-based segmentation')
```