

# Lecture Narrative: Image Segmentation

## Recording Notes

Each subsection corresponds to one slide in the Image Segmentation lecture. Slide numbers follow the order of the presentation from title to the final clustering discussion.

---

### Slide 1: Image Segmentation

This lecture introduces image segmentation, which is the task of partitioning an image into meaningful regions. Segmentation is a foundational problem in computer vision and often serves as the first step toward higher-level understanding.

### Slide 2: Source

This material is adapted from lecture notes available at Imperial College London. The focus here is on classical segmentation methods rather than modern deep learning approaches.

### Slide 3: Introduction to Image Segmentation

The goal of image segmentation is to divide an image into regions that are meaningful for a particular application. The definition of “meaningful” depends heavily on the task we are trying to solve.

### Slide 4: Measurements Used for Segmentation

Segmentation can be based on many different image measurements, including greylevel intensity, color, texture, depth, or motion. Different applications require different cues.

### Slide 5: Segmentation as an Initial Step

Segmentation is usually an early and crucial step in a larger vision pipeline. Errors at this stage can propagate and negatively affect later processing.

### Slide 6: Applications of Segmentation

Applications include identifying objects for measurement, segmenting moving objects for video compression, and separating objects by depth for robotic navigation.

### **Slide 7: Greyscale-Based Segmentation Example**

This example shows segmentation based purely on greyscale intensity. Using a very simple intensity model often leads to incorrect object labeling.

### **Slide 8: Limitations of Greyscale Segmentation**

Greyscale-based segmentation struggles when object and background intensities overlap, resulting in ambiguous boundaries.

### **Slide 9: Texture-Based Segmentation Example**

Segmentation based on texture can distinguish regions even when greyscale values vary within objects.

### **Slide 10: Motivation for Texture Segmentation**

Texture allows us to capture local patterns rather than raw intensity, making segmentation more robust in complex scenes.

### **Slide 11: Motion-Based Segmentation**

Segmentation based on motion requires estimating optical flow, which introduces uncertainty. The segmentation relies on estimated motion rather than true motion.

### **Slide 12: Challenges in Motion Segmentation**

Errors in optical flow estimation directly affect segmentation accuracy, making this a challenging problem.

### **Slide 13: Depth-Based Segmentation**

Depth segmentation uses distance measurements, often from sensors like laser range finders, to separate objects by distance.

### **Slide 14: Robotics Motivation**

Depth-based segmentation is especially useful in robotics, where understanding object distance enables navigation and path planning.

### **Slide 15: Example Images**

This slide shows an original image, its corresponding range image, and the resulting segmented image.

## **Slide 16: Histogram-Based Segmentation**

We now focus on simple segmentation techniques based on the greylevel histogram of an image, specifically thresholding and clustering.

## **Slide 17: Test Images**

We will examine noise-free, low-noise, and high-noise images to understand how noise affects segmentation performance.

## **Slide 18: Histogram Interpretation**

Histograms allow us to visualize the distribution of greylevels and assess how well object and background are separated.

## **Slide 19: Noise-Free Histogram**

In the noise-free case, the histogram consists of two spikes corresponding to object and background intensities.

## **Slide 20: Low-Noise Histogram**

With low noise, the spikes become broader peaks, but the object and background are still distinguishable.

## **Slide 21: High-Noise Histogram**

With high noise, the peaks merge into a single distribution, making segmentation much harder.

## **Slide 22: Signal-to-Noise Ratio**

We define the signal-to-noise ratio in terms of the mean greylevels of object and background and the noise standard deviation.

## **Slide 23: SNR for Test Images**

For the noise-free image, the SNR is infinite. For the low-noise image, it is approximately 5, and for the high-noise image, it drops to around 2.

## **Slide 24: Greylevel Thresholding**

Thresholding is one of the simplest segmentation methods. A single threshold value separates object from background.

### **Slide 25: Histogram Valley**

In the low-noise case, there is a clear valley between the object and background peaks where a threshold can be placed.

### **Slide 26: Threshold Definition**

The thresholding rule assigns pixels to object or background depending on whether their greylevel is below or above the threshold.

### **Slide 27: Choosing the Threshold**

The key challenge is selecting an appropriate threshold value. Several strategies exist, including interactive and adaptive methods.

### **Slide 28: Minimization Approach**

We focus on a minimization approach that selects the threshold by minimizing the within-group variance.

### **Slide 29: Idealized Histogram**

An ideal object-background histogram helps illustrate how thresholding divides pixels into two groups.

### **Slide 30: Within-Group Variance**

Each threshold defines two groups, each with its own mean and variance. The goal is to make each group as homogeneous as possible.

### **Slide 31: Optimal Threshold**

The optimal threshold minimizes the weighted sum of variances within the object and background groups.

### **Slide 32: Group Definitions**

Object pixels have greylevels less than or equal to the threshold, while background pixels exceed it.

### **Slide 33: Prior Probabilities**

The prior probabilities of object and background are computed directly from the histogram.

### **Slide 34: Group Statistics**

The mean and variance of each group can be derived from the histogram and threshold.

### **Slide 35: Optimization**

The within-group variance is evaluated for all possible thresholds, requiring only 256 comparisons for an 8-bit image.

### **Slide 36: Threshold Result**

For the low-noise image, the optimal threshold is approximately 124, roughly midway between the two peaks.

### **Slide 37: Applying the Threshold**

Applying this threshold segments both the low-noise and high-noise images.

### **Slide 38: Low-Noise Segmentation Result**

The segmentation works reasonably well for the low-noise image.

### **Slide 39: High-Noise Segmentation Result**

In the high-noise case, significant pixel misclassification is visible.

### **Slide 40: Misclassification Analysis**

Misclassification arises from overlap between object and background histograms.

### **Slide 41: Fundamental Limitation**

No threshold can perfectly separate object and background when their greylevel distributions overlap.

### **Slide 42: Error Probability**

The probability of error increases as histogram overlap increases.

### **Slide 43: Introduction to Greylevel Clustering**

We now consider greylevel clustering as an alternative to thresholding.

### **Slide 44: Clustering Idea**

Clustering separates greylevels into groups based on proximity to cluster centers.

### **Slide 45: Cluster Centers**

Two cluster centers represent object and background intensities.

### **Slide 46: Nearest Neighbor Assignment**

Each greylevel is assigned to the cluster whose center is closest.

### **Slide 47: Relation to K-Means**

This approach is a simple case of K-means clustering, which is widely used in practice.

### **Slide 48: Partitioning the Data**

The greylevels are partitioned into two sets corresponding to object and background.

### **Slide 49: Updating Cluster Means**

Cluster centers are updated as the mean greylevel of assigned pixels.

### **Slide 50: Chicken-and-Egg Problem**

Cluster assignments depend on cluster means, and cluster means depend on assignments.

### **Slide 51: Iterative Solution**

This dependency is resolved using an iterative algorithm.

### **Slide 52: Iterative Algorithm Steps**

The algorithm alternates between updating cluster means and reassigning pixels.

### **Slide 53: Convergence Questions**

Two key questions arise: does the algorithm converge, and what does it converge to?

### **Slide 54: Cost Function**

We define a cost function measuring within-cluster variance.

### **Slide 55: Updating the Cost**

Each iteration reduces or maintains the cost function value.

### **Slide 56: Proof of Convergence**

Because the cost is non-increasing and bounded below, the algorithm must converge.

### **Slide 57: Interpretation of the Result**

The algorithm converges to a local minimum of within-cluster variance.

### **Slide 58: Relation to Thresholding**

For well-separated clusters, clustering and thresholding yield similar results.

### **Slide 59: Performance Comparison**

Both methods degrade as object and background distributions overlap.

### **Slide 60: Key Takeaways**

Simple histogram-based methods are intuitive and efficient but fundamentally limited by noise and overlap.

### **Slide 61: Transition Forward**

These limitations motivate more advanced segmentation methods based on spatial context, texture, and learning-based models.