Unit IV: IMAGE SEGMENTATION

Image segmentation is a computer vision task that involves dividing an image into multiple regions or segments. Each segment typically corresponds to a meaningful object or region of interest within the image. The goal of image segmentation is to simplify the representation of an image into more understandable and semantically meaningful parts, which can be further analyzed or used in various applications.

The picture can't be displayed.

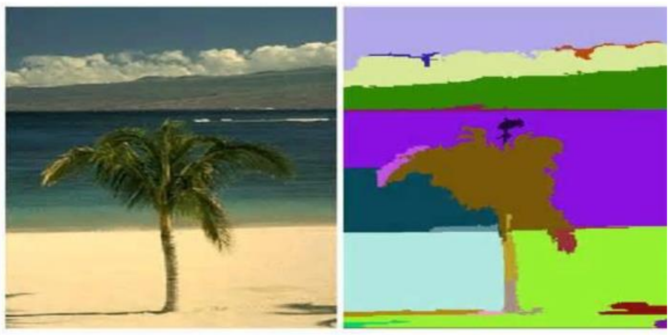


Figure 1: Example of Image Segmentation

There are several types of image segmentation techniques, each with its own approach and characteristics:

1. **Semantic Segmentation:** In this type of segmentation, each pixel in the image is assigned a class label representing the object or region it belongs to. For example, in a street scene, each pixel may be labeled as "car," "pedestrian," "building," etc. This type of segmentation is used in tasks like autonomous driving, object detection, and scene understanding
2. **Instance Segmentation**: Instance segmentation goes beyond semantic segmentation by distinguishing individual instances of objects within the same class. It assigns a unique label to each instance of an object. For instance, in an image with multiple cars, each car would be labeled differently. This technique is useful in scenarios where individual object instances need to be separately identified.
3. **Boundary-based Segmentation:** This approach aims to detect and delineate object boundaries by identifying edges or contours. It's particularly useful when the focus is on the outline of objects rather than the interior.
4. **Region-based Segmentation:** Region-based segmentation groups pixels based on their similarity in color, texture, or other image features. It involves clustering pixels together to form meaningful regions, often through algorithms like k-means or watershed segmentation.
5. **Graph-based Segmentation:** This technique represents the image as a graph, where pixels are nodes, and edges represent connections between neighboring pixels. Graph-based segmentation algorithms find natural boundaries by minimizing a certain cost function related to the graph.
6. **Deep Learning-based Segmentation:** With the advent of deep learning, convolutional neural networks (CNNs) have become the backbone of many modern image segmentation approaches. U-Net, SegNet, and Mask R-CNN are popular CNN architectures used for image segmentation tasks.

Applications of image segmentation are diverse, ranging from medical imaging (e.g., tumor detection) and industrial inspection to object recognition and augmented reality.

Fundamentals

The fundamentals of image segmentation encompass the key concepts and techniques that underlie this important computer vision task. Here are some fundamental aspects:

1) **Image Representation:** Images are typically represented as two-dimensional arrays of pixels, where each pixel contains information about color and intensity. In grayscale images, each pixel is represented by a single intensity value, while in color images, each pixel contains multiple values representing color channels (e.g., red, green, blue).
2) **Object vs. Background:** Image segmentation aims to distinguish between different objects or regions of interest (foreground) and the background. The goal is to separate these meaningful parts of the image for further analysis or understanding.
3) **Feature Extraction:** Image segmentation often relies on extracting relevant features from the image to identify boundaries or similarities between pixels. Features can include color, texture, gradient information, edges, and other visual characteristics.
4) **Thresholding**: Thresholding is a simple segmentation technique where pixels are separated based on their intensity values. A threshold value is set, and pixels with values above or below the threshold are classified as foreground or background, respectively. This technique is effective for binary segmentation tasks.
5) **Clustering:** Clustering algorithms group similar pixels together based on their feature similarity. Common clustering methods include k-means and mean-shift, which can be applied in region-based segmentation.
6) **Edge Detection:** Edge detection algorithms identify abrupt changes in intensity, representing potential object boundaries. Common edge detection methods include Sobel, Canny, and the Scharr operator.
7) **Watershed Transform:** The watershed algorithm treats the image as a topographic map and floods regions from the lowest intensity points. It can segment objects based on their valleys and ridges, but can also lead to oversegmentation.
8) **Graph Theory**: Graph-based segmentation involves representing the image as a graph, with pixels as nodes and edges connecting neighboring pixels. Algorithms like normalized cut or minimum spanning tree can find meaningful segmentation boundaries by analyzing the graph.

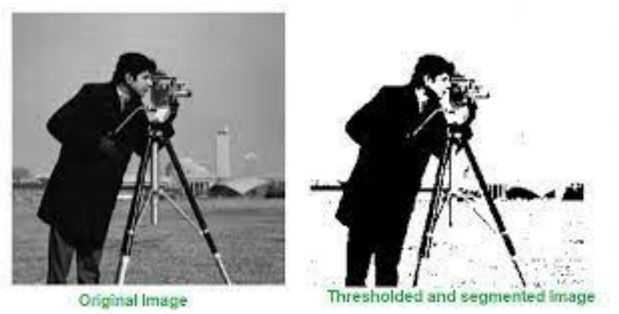
9) **Region Merging and Splitting:** These techniques iteratively merge or split regions based on predefined criteria to achieve more accurate segmentation.
10) **Deep Learning:** Convolutional Neural Networks (CNNs) have revolutionized image segmentation. Deep learning-based segmentation methods, such as U-Net and Mask R-CNN, use CNN architectures to learn and predict pixel-wise segmentation masks from labeled training data.

Evaluation Metrics: To assess the performance of an image segmentation algorithm, various evaluation metrics are used, such as Intersection over Union (IoU), Dice coefficient, and pixel accuracy.

<u>**point, line, edge detection**</u>

Point, line, and edge detection are fundamental image processing techniques used to identify and extract specific features from an image. Each of these techniques focuses on detecting different types of visual elements within the image.

1. **Point Detection:**Point detection, also known as blob detection, aims to identify individual points or isolated areas of interest in an image. These points could represent key locations, corners, or interest points with unique characteristics. Point detection algorithms highlight regions of the image where there is a significant change in pixel intensity compared to its neighboring pixels. Some popular methods for point detection include the Harris Corner Detector and the Shi-Tomasi Corner Detector.
2. **Line Detection:**Line detection involves identifying straight or curved lines within an image. These lines can represent various structures, edges, or patterns. Line detection algorithms typically analyze the intensity and gradient information across the image to identify regions where the pixels align in a straight or nearly straight line. One common algorithm for line detection is the Hough Transform, which can detect lines in both polar and Cartesian coordinate systems.
3. **Edge Detection:**Edge detection is a more general technique that aims to identify abrupt changes in pixel intensity, which often correspond to object boundaries or significant image structures. Edges represent regions of high gradient magnitude and are crucial for various computer vision tasks, including image segmentation and object recognition. Some well-known edge detection algorithms include the Canny edge detector, the Sobel operator, and the Laplacian of Gaussian (LoG) operator.


Original Image   Thresholded and segmented image

<u>Thresholding</u>

Thresholding is a basic image processing technique used to create binary images from grayscale images. It involves converting a grayscale image into a binary image, where pixels are classified into two groups based on a specific threshold value: those above the threshold are set to one (white), and those below or equal to the threshold are set to zero (black).

<u>The process of thresholding can be summarized as follows:</u>

1) **Grayscale Image**: The input image is typically a grayscale image, where each pixel represents the intensity or brightness level. Grayscale images are usually represented by values ranging from 0 (black) to 255 (white).
2) **Threshold Selection:** The threshold value is a crucial parameter that determines the separation between the two classes of pixels (foreground and background) in the binary image. Choosing the appropriate threshold value depends on the characteristics of the image and the specific task.
3) **Thresholding Operation:** Each pixel in the grayscale image is compared to the threshold value. If the pixel's intensity is greater than the threshold, it is assigned a value of 1 (white) in the binary image. Otherwise, if the pixel's intensity is less than or equal to the threshold, it is assigned a value of 0 (black).
4)

<u>Mathematically, the thresholding operation can be represented as follows:</u>

- Binary Image(x, y) = 1 if Grayscale Image(x, y) > Threshold

= 0 if Grayscale Image(x, y) <= Threshold

<u>Thresholding is useful for various image processing tasks, including:</u>

- **Object Segmentation:** By setting an appropriate threshold, it is possible to separate objects or regions of interest from the background, creating binary masks for further processing.
- **Image Enhancement:** Thresholding can be used to enhance certain features or structures in an image by isolating them from the rest of the image.
- **Image Preprocessing**: Thresholding can simplify the image and reduce the complexity of subsequent processing steps.
- **Feature Extraction:** In some cases, thresholding can be used to extract specific features from the image.

However, choosing the correct threshold value can be challenging, especially when dealing with images with non-uniform lighting conditions or high levels of noise. In such cases, adaptive thresholding techniques that automatically adjust the threshold value based on local image characteristics can be used.

region –based segmentation

Region-based segmentation is a type of image segmentation technique that groups pixels into meaningful regions based on their similarity in terms of color, texture, intensity, or other feature characteristics. The primary goal of region-based segmentation is to partition an image into distinct regions or regions of interest (ROIs) that correspond to different objects or entities present in the image.

The process of region-based segmentation can be summarized as follows:

1. **Image Preprocessing:** The first step often involves pre-processing the image to enhance features or reduce noise. Common pre-processing techniques include smoothing, contrast enhancement, and noise reduction.
2. **Region Growing or Region Merging:** The main idea in region-based segmentation is to start with a seed pixel or region and iteratively grow or merge regions based on similarity criteria. Region growing starts with one or more seed points and expands each region by adding neighboring pixels that are similar to the region's seed points according to a predefined criterion (e.g., intensity or color similarity). Region merging, on the other hand, starts with each pixel as a separate region and iteratively merges regions based on their similarity until the desired segmentation is achieved.
3. **Region Similarity Criteria:** The similarity criteria used for region-based segmentation play a crucial role in the final result. Common similarity measures include:

- **Intensity or color similarity:** Pixels with similar intensity or color values are grouped together.
- **Texture similarity:** Regions with similar texture patterns, such as smooth or rough areas, are merged or grown together.
- **Gradient similarity:** Regions with similar gradients or edges are combined.

4. **Stopping Criteria:** The region growing or merging process continues until a stopping criterion is met. This criterion can be based on the size of regions, the similarity between regions, or other specific application requirements.
5. **Post-processing:** After obtaining the initial segmentation, post-processing steps may be applied to refine the results and handle any inconsistencies or artifacts. Techniques like morphological operations (e.g., erosion and dilation) can be used to improve the segmentation's accuracy and smooth the boundaries.

Region-based segmentation is particularly useful when dealing with images containing objects with homogeneous textures and well-defined boundaries. It is less sensitive to noise compared to edge-based segmentation methods and can produce more coherent and meaningful regions. However, region-based segmentation may encounter challenges in cases where the regions have irregular shapes, overlapping boundaries, or varying intensity levels.

MORPHOLOGICAL IMAGE PROCESSINGs

Morphological image processing is a branch of image processing that deals with the analysis and processing of images based on their shapes and structures. It involves using a set of operations derived from mathematical morphology to manipulate images and extract useful information. These operations are typically applied to binary or grayscale images and are particularly effective for tasks like noise removal, image enhancement, object detection, and image segmentation.

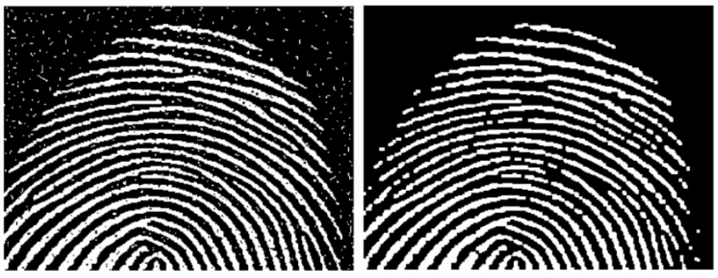


Image after segmentation Image after segmentation and morphological processing

The fundamental morphological operations are based on two basic operations: dilation and erosion. These operations are defined using small structuring elements or kernels, which are typically small shapes (e.g., squares, circles, or lines). The structuring element is applied to each pixel in the image, influencing the pixel's value based on its neighbors.

1) **Dilation:** Dilation is a morphological operation that enlarges bright regions (white regions in binary images) and thickens edges or boundaries. It involves moving the structuring element over the image, and for each position, if any part of the structuring element overlaps with the white pixels, the corresponding output pixel is set to white.
2) **Erosion:** Erosion is the opposite of dilation. It shrinks bright regions and thins edges. It involves moving the structuring element over the image, and for each position, if all the pixels under the structuring element are white, the corresponding output pixel is set to white; otherwise, it is set to black.

Combining dilation and erosion allows for more complex operations:

- **Opening:** Opening is an operation that first applies erosion and then dilation. It is useful for removing small noise regions and separating connected objects. Opening can preserve the shape and size of larger regions while removing smaller details.
- **Closing**: Closing is an operation that first applies dilation and then erosion. It is useful for closing small gaps in regions and joining nearby objects. Closing can help in filling small holes within objects.

- **Morphological Gradient:** The morphological gradient is the difference between dilation and erosion. It highlights the boundaries of objects in the image.
- **Top Hat:** The top hat operation is the difference between the input image and its opening. It can reveal small, bright structures that are not well-connected to the larger objects.
- **Black Hat:** The black hat operation is the difference between the closing of the input image and the input image itself. It can highlight small, dark structures on a bright background.

Morphological image processing is a powerful tool for various image analysis tasks, especially when dealing with binary or grayscale images with well-defined structures.

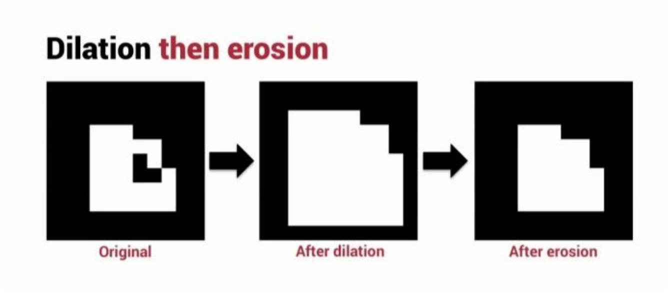
Preliminaries

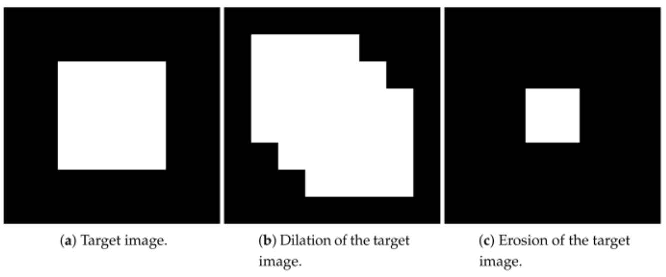
Before delving into morphological image processing operations, it's important to understand some preliminary concepts:

Binary Images: Binary images are images where each pixel has only two possible values, typically represented as 0 (black) and 1 (white). These images are commonly used in morphological image processing, as the operations are designed to work effectively with binary data.
Grayscale Images: Grayscale images are images where each pixel has a single intensity value, typically represented by an 8-bit value ranging from 0 (black) to 255 (white). Grayscale images can be thresholded to create binary images for morphological operations.
Structuring Element: A structuring element is a small shape or neighborhood used in morphological operations. It defines the region of influence around each pixel during the operation. Common structuring elements include squares, rectangles, circles, and lines. The size and shape of the structuring element influence the behavior of the morphological operations.
Image Convolution: Morphological operations involve moving the structuring element over the image and applying a specific operation at each position. This process is similar to convolution in image processing, where the structuring element acts as a kernel that slides over the image.
Boundary Conditions: When applying morphological operations near the image borders, the structuring element might extend beyond the image's boundaries. Different boundary conditions can be used to handle such cases, including zero-padding, reflection, and extension of the nearest valid pixel value.
Hit-and-Miss Transform: The hit-and-miss transform is an advanced morphological operation used for pattern matching in binary images. It can detect specific patterns defined by two structuring elements: one representing the foreground and the other representing the background of the pattern.
Connected Components: In binary images, connected components are sets of adjacent white pixels that form a connected region. Morphological operations like erosion and dilation can help in segmenting and manipulating connected components.
Morphological image processing operates on the idea of transforming an image based on the shape and structure of its pixel neighborhoods. These operations are particularly useful for tasks involving binary or grayscale images with well-defined structures, such as object detection, image segmentation, noise removal, and feature extraction.

Erosion and dilation

Erosion and dilation are fundamental morphological operations used in image processing, particularly for processing binary images. They involve applying a small structuring element or kernel to the image to modify or extract specific features.




(a) Target image. (b) Dilation of the target image. (c) Erosion of the target image.

Erosion: Erosion is a morphological operation that shrinks or erodes the boundaries of bright regions (white regions) in a binary image. It works by moving the structuring element over the image and checking if all the pixels under the structuring element are white. If they are, the corresponding output pixel is set to white; otherwise, it is set to black. The operation causes the white regions to decrease in size and remove small noise regions or thin structures.
Erosion is effective for:
- Removing small noise or isolated pixels in a binary image.
- Separating or breaking apart connected regions.

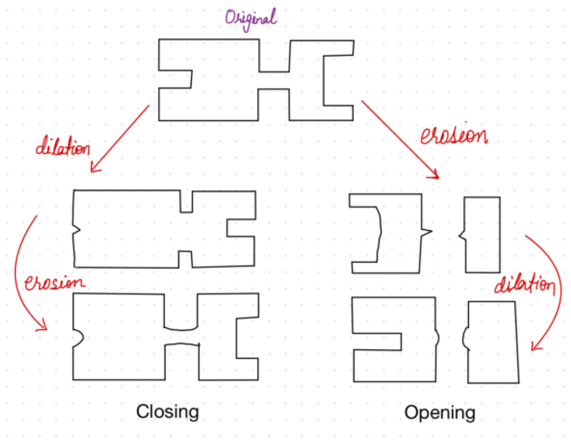
Dilation: Dilation is the opposite of erosion. It expands or dilates bright regions in a binary image. Like erosion, it also involves moving the structuring element over the image, but this time, the output pixel is set to white if any part of the structuring element overlaps with a white pixel in the image. This causes the white regions to increase in size and fill gaps or enlarge structures.
Dilation is useful for:
- Closing small gaps between white regions.
- Enlarging objects or structures in the image.

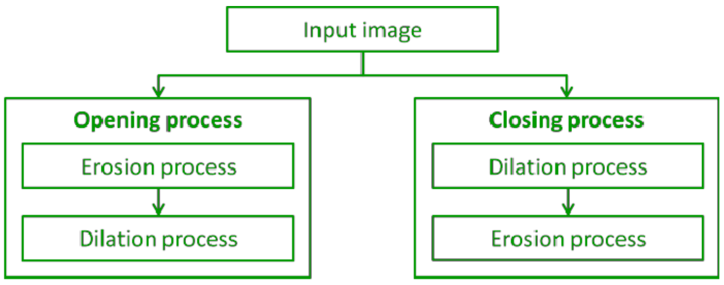
Both erosion and dilation can be used for image preprocessing, feature extraction, and segmentation tasks. These operations are often combined to perform more complex morphological operations like opening and closing:

Opening: Opening is an erosion followed by a dilation. It is useful for removing small noise regions and separating or smoothing connected regions.
Closing: Closing is a dilation followed by an erosion. It is useful for closing small gaps between white regions and joining nearby objects.
The choice of the structuring element's size and shape affects the extent of erosion and dilation. A larger structuring element will cause a more significant effect, while a smaller one will have a more localized effect on the image. The structuring element's shape can also be customized to match the desired features to be enhanced or extracted.

opening and closing

Opening and closing are two morphological operations that involve combining erosion and dilation to process binary images. These operations are commonly used to manipulate the shapes and structures in binary images, especially for noise removal, smoothing, and region segmentation.





Opening:

Opening is a morphological operation that consists of applying erosion followed by dilation. The process involves moving the structuring element over the binary image and performing erosion first and then dilation. Opening is particularly useful for removing small noise regions, separating connected regions, and smoothing the contours of objects.

The main steps of the opening operation are as follows:
a. Erosion: The structuring element is moved over the image, and at each position, if all the pixels under the structuring element are white, the corresponding output pixel is set to white; otherwise, it is set to black. This step removes small noise regions and thins the edges of objects.
b. Dilation: Next, the dilation operation is applied using the same structuring element. The structuring element is again moved over the eroded image, and at each position, if any part of the structuring element overlaps with a white pixel in the image, the corresponding output pixel is set to white. This step helps in restoring the size and shape of the objects while still keeping the small noise regions removed in the previous step.

- The opening operation can be represented as Opening = Dilation(Erosion(Image)).

Opening is useful for tasks such as:

- Removing small isolated noise pixels from the binary image.
- Separating connected objects or regions that are too close to each other.

Closing:

Closing is another morphological operation that consists of applying dilation followed by erosion. Similar to opening, the structuring element is moved over the binary image, but this time dilation is applied first, followed by erosion. Closing is useful for closing small gaps between white regions and joining nearby objects that are separated by narrow spaces.

The main steps of the closing operation are as follows:
a. Dilation: The structuring element is moved over the image, and at each position, if any part of the structuring element overlaps with a white pixel in the image, the corresponding output pixel is set to white. This step helps in closing small gaps between white regions.
b. Erosion: Next, the erosion operation is applied using the same structuring element. The structuring element is moved over the dilated image, and at each position, if all the pixels under the structuring element are white, the corresponding output pixel is set to white; otherwise, it is set to black. This step helps in restoring the size and shape of the objects while keeping the gaps closed in the previous step.

- The closing operation can be represented as Closing = Erosion(Dilation(Image)).

Closing is useful for tasks such as:

- Closing small gaps between white regions or objects.

- Joining nearby objects that are separated by narrow spaces.

basic morphological algorithms for boundary extraction, thinning

Boundary extraction and thinning are two important morphological algorithms used for processing binary images. They help in extracting essential features and enhancing the representation of objects in the image.

Boundary Extraction:

Boundary extraction, also known as contour extraction, is a morphological operation that isolates the boundaries or contours of objects in a binary image. The operation highlights the regions where there is a transition from foreground (white pixels) to background (black pixels) or vice versa. This is achieved by subtracting the original image from its dilation. The basic steps for boundary extraction are as follows:

a. Dilation: Perform dilation on the binary image using an appropriate structuring element. Dilation enlarges the white regions and smoothens the boundaries.

b. Subtraction: Subtract the original binary image from the dilated image. This process results in an image with only the boundary pixels remaining.

Boundary extraction is useful for tasks such as:

- Object detection and recognition based on their contours.
- Feature extraction, where contours are essential for characterizing the shape of objects.
- Image analysis and object tracking by monitoring changes in object boundaries.

Thinning:

Thinning is a morphological operation that reduces the thickness of individual white regions in a binary image to a single-pixel width while preserving the connectivity of the shapes. Thinning is often used to simplify the representation of objects and extract their skeletons. The basic steps for thinning are as follows:

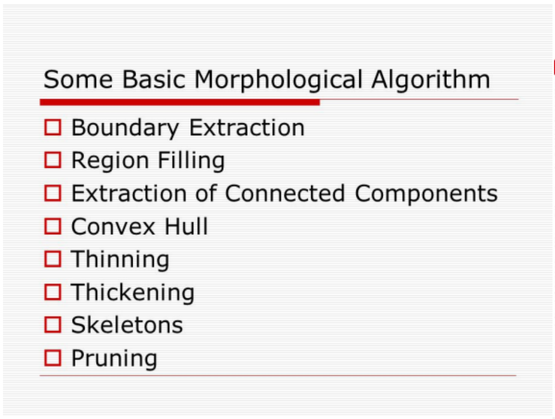
a. Erosion: Perform erosion on the binary image using an appropriate structuring element. Erosion shrinks the white regions and removes the outer pixels.

b. Subtraction and Pruning: Subtract the eroded image from the original binary image. Then, iteratively prune the pixels that do not affect the connectivity of the objects until the desired thinning is achieved.

Thinning is useful for tasks such as:

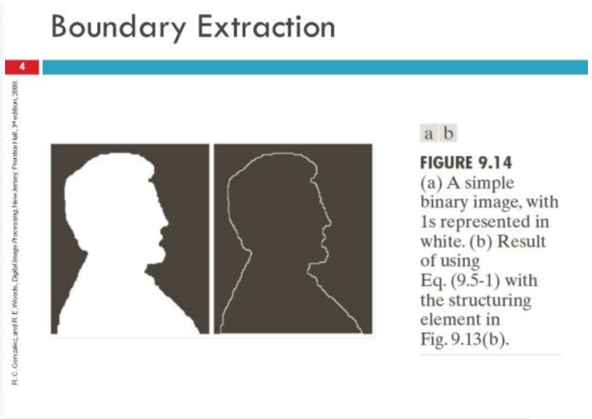
- Skeleton extraction, where the simplified representation of objects is required for shape analysis or pattern recognition.
- Feature extraction, where the skeleton can provide valuable information about the structure of objects.

Both boundary extraction and thinning are morphological algorithms that operate on binary images. They are commonly used in various image processing and computer vision applications to enhance object representations and extract meaningful information from binary images. The choice of the structuring element and the specific pruning criteria for thinning can impact the results and the performance of these algorithms for different applications.



Some Basic Morphological Algorithm

- Boundary Extraction
- Region Filling
- Extraction of Connected Components
- Convex Hull
- Thinning
- Thickening
- Skeletons
- Pruning



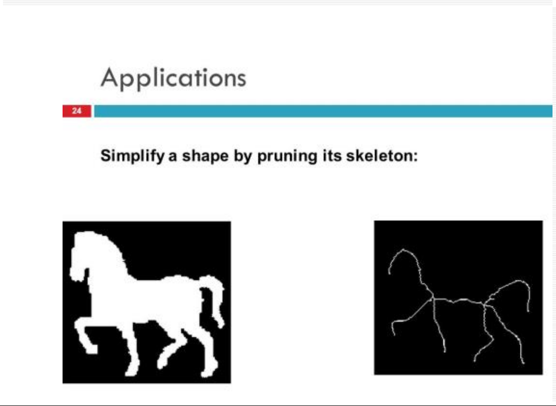
Boundary Extraction

4

R. C. Gonzalez and R. E. Woods, Digital Image Processing, New Jersey: Prentice Hall, 3rd edition, 2008

a b

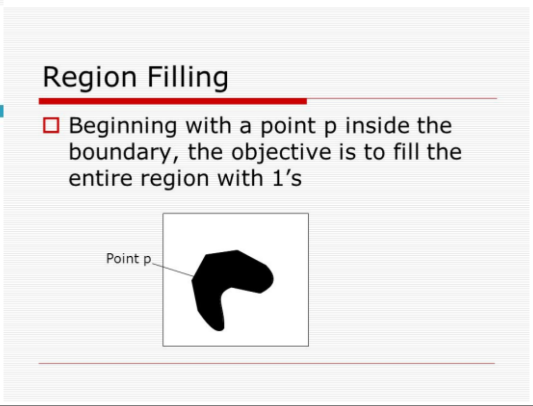
FIGURE 9.14 (a) A simple binary image, with 1s represented in white. (b) Result of using Eq. (9.5-1) with the structuring element in Fig. 9.13(b).



Applications

24

Simplify a shape by pruning its skeleton:



Region Filling

- Beginning with a point p inside the boundary, the objective is to fill the entire region with 1's