IMPLEMENT MORPHOLOGICAL EDGE DETECTION PROCEDURES

ABSTRACT

Identifying edges in images remains a difficult task in many applications, despite there are various techniques such as Sobel, Robert, Prewitt, Canny, and Cellular Automata. This Paper proposes an approach for edge detection that uses Morphological operations, using Erosion, Dilation, Gradient processes to locate edges. Specifically, we use morphological operators to identify thin edges in binary images obtained through Otsu's thresholding technique. This Process has been tested on few real images from various domains. This Process is also capable of producing continuous, one-pixel-width edges with precise positioning while preserving small image details.

Keywords: Morphological Operations, Edge Detection, Otsu's method

1. INTRODUCTION

The edges detection in digital images is important aspect of Digital image processing, with ongoing research aimed at developing new techniques. Edge detection is defined as the process of identifying sharp changes in pixel intensity and it is important in various fields like image segmentation, feature recognition and extraction, computer-guided surgery, and motion estimation.

Edges represent abrupt changes in pixel intensities that indicate object boundaries or the boundaries between objects and their background. Edge detection techniques are commonly used to reduce the amount of data to be processed by filtering out irrelevant information while preserving the image's important structural properties. However, obtaining ideal edges from real images with moderate complexity is not always possible.

Most edge detection techniques can be categorized into three main groups:

gradient-based methods, Laplacian-based methods, and model-based methods.

Gradient-Based Methods: This method detects edges by searching for maximums and minimums in the first derivative of the image.

Examples of gradient-based are Sobel, Prewitt, and Robert.

Laplacian-Based Methods: This method searches for zero-crossings in the second-order derivative expression computed from the image to find edges. Laplacian operator is the most common operator used with zero-crossing method.

Model-Based Methods: This method uses different models or algorithm to locate edges from images some are very application dependent such as using wavelets, fuzzy, SVM , Cellular Automata, and edge detection methods.

2. MORPHOLOGICAL OPERATIONS

Morphological edge detection is a technique used to extract the edges of objects in an image by applying morphological operations. Morphological operations are mathematical operations that are performed on an image to extract or enhance specific features of interest.

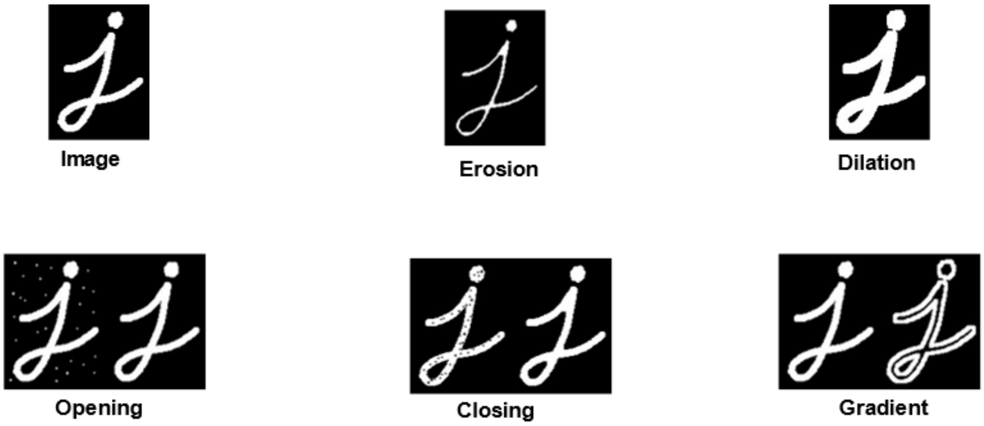
Morphological edge detection is a type of non-gradient based method for extracting edges from images. It is based on the use of morphological operations, such as dilation, erosion, and gradient, to obtain the boundaries of objects in the image. These boundaries are then thresholded to obtain the edges.

Morphological edge detection is to take the morphological gradient of an image, which is defined as the difference between the dilation and erosion of the image using a structuring element. The dilation operation expands the boundaries of the object in the image, while the erosion operation shrinks them. Therefore, taking the difference between the dilation and erosion operations highlights the edges of the objects in the image.

Dilation and erosion are two fundamental operations in morphological image processing. They are used to modify the shape and size of objects in a binary image and are often used in combination with each other to achieve specific image processing goals.

- Dilation: Dilation is a morphological operation that is used to expand or grow binary objects in an image. It works by applying a structuring element to each pixel in the binary image and setting the output pixel to 1 if any of the pixels in the structuring element is 1. This has the effect of making the binary objects larger and more connected. It is used to fill in gaps between objects, connect disjoint objects, and smooth out the edges of objects. It can also be used to remove small holes or gaps within objects.
- Erosion: Erosion is a morphological operation that is used to shrink or erode binary objects in an image. It works by applying a structuring element to each pixel in the binary image and setting the output pixel to 0 if any of the pixels in the structuring element is 0. This has the effect of making the binary objects smaller and more separated. It is used to separate touching objects, remove small objects, and smooth out the edges of objects. It can also be used to remove small protrusions or spikes from objects.
- Dilation and erosion can be used in combination to achieve specific image processing goals. One common technique is to perform dilation followed by erosion, which is known as opening. This has the effect of removing small objects and smoothing out the edges of larger objects, another technique is to perform erosion followed by dilation, which is known as closing. This has the effect of filling in gaps between objects and connecting disjoint objects.

Morphological Operations :



Source : https://docs.opencv.org/4.x/d9/d61/tutorial_py_morphological_ops.html

Figure 1

Automatic thresholding using Otsu's method is a popular image processing technique for determining the optimal threshold value for image segmentation. The goal of image segmentation is to partition an image into distinct regions or objects based on their properties such as intensity, color, or texture. Thresholding is a common technique used in image segmentation, where pixels with intensity values above or below a certain threshold value are classified as foreground or background pixels.

Otsu's method is a histogram-based technique that automatically determines the optimal threshold value by minimizing the variance between the foreground and background pixels. The algorithm works by computing the histogram of the image and the cumulative distribution function (CDF) of the normalized histogram. Then, for each possible threshold value, the algorithm computes the between-class variance, which measures the separation between the foreground and background pixels. The threshold value that maximizes the between-class variance is selected as the optimal threshold value.

A 3x3 square structuring element is a small matrix or kernel that is used in morphological image processing operations such as dilation, erosion, opening, and closing. The structuring element defines the shape and size of the neighborhood over which the morphological operation is applied.

In the case of a 3x3 square structuring element, the kernel is a square matrix with a size of 3 pixels on each side, and all its elements are set to 1. The structuring element is placed at each pixel location in the image, and the operation is applied to the neighborhood of pixels that fall within the structuring element.

3. METHODOLOGY

i. Load the input image as a binary image!
The first step is to load the input image as a binary image. A binary image is an image that consists of only two-pixel values: 0 and 1. In this case, we can convert a grayscale or color image to a binary image using a thresholding operation. A thresholding operation sets all pixel values above a certain threshold to 1 and all pixel values below the threshold to 0. In the example code, we use the ncv2.threshold function to perform this operation. The cv2.IMREAD_GRAYSCALE flag is used to load the input image as a grayscale image, and the cv2.THRESH_BINARY | cv2.THRESH_OTSU flags are used to apply automatic thresholding using Otsu's method.

ii. Define a structuring element!
A structuring element is a binary image that defines the shape and size of the neighborhood used for the morphological operations. In this example, we define a 3x3 square structuring element using the np.ones function in NumPy. A square structuring element is often used for edge detection because it can capture edges of different orientations.

iii. Perform dilation and erosion operations on the input image using the structuring element!
The dilation operation expands the boundaries of the objects in the image, while the erosion operation shrinks them. In this example, we perform a dilation operation and an erosion operation on the binary input image using the square structuring element. We use the cv2.dilate and cv2.erode functions to perform these operations. The iterations parameter is used to specify the number of times the operation is applied. In this case, we perform the operation once.

iv. Take the difference between the dilated and eroded images to obtain the gradient image!
The morphological gradient is defined as the difference between the dilation and erosion of the image using the structuring element. Taking the difference between the dilated and eroded images highlights the edges of the objects in the image. In this example, we obtain the gradient image by taking the absolute difference between the dilated and eroded images using the cv2.absdiff function.

v. Threshold the resulting gradient image to obtain the edges!
The gradient image obtained in the previous step is a grayscale image that contains intensity values that correspond to the strength of the edges in the original image. To obtain the actual edges, we need to threshold the gradient image to obtain a binary image where edge pixels are set to 1 and non-edge pixels are set to 0. In this example, we use the cv2.threshold function to perform thresholding. We use the same automatic thresholding method as before, which computes the threshold value based on the image histogram.

vi. Display the resulting edges!
Finally, we display the resulting edges using the cv2.imshow function. The edges are displayed as a binary image where edge pixels are white and non-edge pixels are black. The cv2.waitKey function is used to wait for a key press before closing the window.

4. Results

Input image Binary Image Edges

Source : Using morphological operations — Erosion based algorithm for edge detection

Figure -2

Input image Binary Image Edges

Source : Using morphological operations — Erosion based algorithm for edge detection

Figure-3

We have used only three morphological operations to obtain the edges of various images. In result we have shown the input image, binary image, edges. I have taken the four various images and their results are shown above.
The first image is image of a person taking photograph, I have used above methodology to obtain the edges.
The second image is the Noisy fingerprint image in which I have used the same methodology to obtain the edges.
The Third image is the Lena's image in which also I have used same methodology to obtain the edges.
The Fourth image is the Anime image which I also used to check this methodology to obtain the edges.
Since I have edges obtained in first, second, third image are better, but for fourth image is also good.
The result obtained from this method is quite better in comparison to other methods.

5.CONCLUSION

- This method provides us with better results in Edge extraction.
- This method provides us with clear, thin, and continuous edges.
- The results obtained from this method are better than classical edge detection methods.
- This Methods require less time processing time so, it is time efficient.
- This method is used when there are some more complex edges and contrast base challenges.

REFERENCES

- Gonzalez, C. R., & Woods, R. E. (2008). Digital Image Processing. Pearson Education International
- Rashmi, Mukesh Kumar, Rohini Saxena, "Algorithm and Technique on Various Edge Detection: a survey," Signal & Image Processing: An International Journal (SIPIJ) Vol.4, No.3, pages 65-75, June 2013.
- C. Sujathaand and Dr. D. Selvathi, "An optimal solution for image edge detection problem using simplified Gabor wavelet," International Journal of Computer Science, Engineering, and Information Technology (IJCSEIT), vol. 2, No.3, June 2012.
- Abhradita Deepak Borkar, Mithilesh Atulkar, "Detection of edges using fuzzy inference system," International Journal of Innovative Research in Computer and Communication Engineering (2320 – 9798). Volume 1, issue1 March 2013.
- Sheng Zheng, Jian Liu, and Jin Wen Tian, "A new efficient SVM-based edge detection method," Pattern Recognition Letters, Volume 25 Issue 10, Pages 1143 - 1154, 16 July 2004.