**LAPORAN PRAKTIKUM PENGOLAHAN CITRA DIGITAL**

**19. IMAGE THRESHOLDING**



**Disusun oleh :**

**Nama : Garcia Bryan Farrel**

**NPM : 2327250026**

**Kelas : IF41**

**PROGRAM STUDI INFORMATIKA**

**FAKULTAS ILMU KOMPUTER DAN REKAYASA**

**UNIVERSITAS MULTI DATA PALEMBANG**

**2024**

**TUTORIAL : EDGE DETECTION**

**Goal**

The goal of this tutorial is to learn to perform image thresholding using MATLAB

and the IPT.

**Objectives**

* Learn how to visually select a threshold value using a heuristic approach.
* Explore the graythresh function for automatic threshold value selection.
* Learn how to implement adaptive thresholding.

**What You Will Need**

* + gradient\_with\_text.tif



FIGURE 19.1 Segmentation using the morphological watershed transform: (a) complement of the image shown in Figure 19.2; (b) distance transform; (c) watershed ridge lines; (d) result of segmentation.



**FIGURE 19.2** Image thresholding results for the image in Figure 15.1c using iterative threshold selection algorithm (a) and manually selected threshold (b).

**Procedure**

**Global Thresholding**

The first method of thresholding that we will explore involves visually analyzing the histogram of an image to determine the appropriate value of T (the threshold value).

1. Load and display the test image.



|  |  |
| --- | --- |
|  |  |

1. Display a histogram plot of the coins image to determine what threshold level to use.



|  |  |
| --- | --- |
|  |  |

**Question 1** Which peak of the histogram represents the background pixels and which peak represents the pixels associated with the coins?

|  |
| --- |
| Puncak kiri pada histogram mewakili piksel latar belakang warna gelap.  Puncak kanan mewakili piksel koin warna putih atau terang. |

The histogram of the image suggests a bimodal distribution of grayscale values. This means that the objects in the image are clearly separated from the background.



**FIGURE 19.1** Histogram plot with data cursor selection.

We can inspect the X and Y values of the histogram plot by clicking the “inspect” icon and then selecting a particular bar on the graph.

1. Inspect the histogram near the right of the background pixels by activating the data cursor. To do so, click on the “inspect” icon.

**Figure 19.1** illustrates what the selection should look like. The data cursor tool suggests that values between 80 and 85 could possibly be used as a threshold, since they fall immediately to the right of the leftmost peak in the histogram. Let us see what happens if we use a threshold value of 85.

1. Set the threshold value to 85 and generate the new image.



|  |  |
| --- | --- |
|  |  |

**Question 2** What is the purpose of the im2bw function?

|  |
| --- |
| Fungsi im2bw digunakan untuk mengubah citra grayscale menjadi citra biner atau black and white.   * Piksel dengan nilai lebih besar dari threshold akan menjadi putih (1). * Piksel dengan nilai lebih kecil atau sama dengan threshold akan menjadi hitam (0). |

**Question 3** Why do we divide the threshold value by 255 in the im2bw function call?

|  |
| --- |
| Fungsi im2bw mengharapkan nilai threshold dalam rentang 0 hingga 1, karena bekerja dengan citra dalam format normalized mangkanya nilai piksel dibagi 255. |

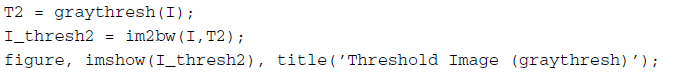
You may have noticed that several pixels—some white pixels in the background and a few black pixels where coins are located—do not belong in the resulting image. This small amount of noise can be cleaned up using the noise removal techniques.

**Question 4** Write one or more lines of MATLAB code to remove the noise pixels in the thresholded image.

|  |  |
| --- | --- |
|  |  |

The thresholding process we just explored is known as the heuristic approach. Although it did work, it cannot be extended to automated processes. Imagine taking on the job of thresholding a thousand images using the heuristic approach! MATLAB’s IPT function graythresh uses Otsu’s method for automatically finding the best threshold value.

1. Use the graythresh function to generate the threshold value automatically.



|  |  |
| --- | --- |
|  |  |

**Question 5** How did the graythresh function compare with the heuristic approach?

|  |
| --- |
| Graythresh menghilakan noise tapi membuat gambar menjadi rusak. Jadi lebih baik heuristic. |

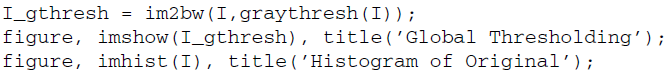
1. Close all open figures and clear all workspace variables.
2. Load the gradient\_with\_text image and prepare a subplot.



|  |  |
| --- | --- |
|  |  |

Let us see what happens when we attempt to threshold this image using the techniques we have learned so far.

1. Globally threshold the image.



|  |
| --- |
|  |
|  |

As you may have noticed, we cannot pick one particular value to set as the threshold value because the image is clearly not bimodal. Adaptive thresholding may help us in this instance. To properly implement adaptive thresholding, we must use the blkproc function to perform an operation on small blocks of pixels one at a time.

In order to use the function, we must specify what is to be done on each block of pixels. This can be specified within a function that we will create manually. Let us first set up this function.

1. Close all open figures.
2. Start a new M-File in the MATLAB Editor.
3. Define the function as well as its input and output parameters in the first line.



This function will be used to define each new block of pixels in our image. Basically all we want to do is perform thresholding on each block individually, so the code to do so will be similar to the code we previously used for thresholding.

1. Add this line of code under the function definition.



When the function is called, it will be passed a small portion of the image, and

will be stored in the variable x. We define our output variable y as a black and white image calculated by thresholding the input.

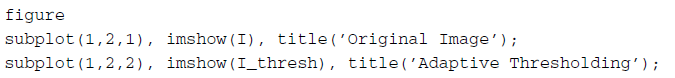
1. Save your function as adapt\_thresh.m in the current directory.

We can now perform the operation using the blkproc function. We will adaptively threshold the image, 10×10 pixel blocks at a time.

1. Perform adaptive thresholding by entering the following command in the command window. Note that it may take a moment to perform the calculation, so be patient.



1. Display the original and new image.



|  |
| --- |
|  |
|  |

The output is not quite what we expected. If you look closely, however, the operation was successful near the text, but everywhere else it was a disaster. This suggests that we need to add an extra step to our function to compensate for this unwanted effect. Over the next few steps, let us examine the standard deviation of the original image where there is text, and where there is not.

1. Calculate the standard deviation of two 10×10 blocks of pixels; one where there is text and another where there is not.



|  |
| --- |
|  |
|  |

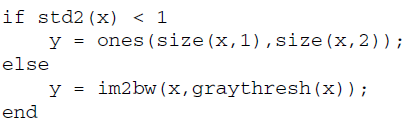
**Question 6** What is the difference between the standard deviation of the two blocks of pixels? Explain.

|  |
| --- |
| Blok piksel pertama memiliki standar deviasi rendah karena merupakan area latar belakang polos dengan intensitas hampir sama. Dan sebaliknya untuk blok kedua memiliki standar deviasi tinggi karena mengandung detail seperti teks atau tepi objek yang menyebabkan variasi piksel besar. |

Since there is such a difference between a block with and without text, we can

use this information to improve our function. Let us replace the one line of code we previously wrote with the new code that will include an if statement: if the standard deviation of the block of pixels is low, then simply label it as background; otherwise, perform thresholding on it. This change should cause the function to only perform thresholding where text exists. Everything else will be labeled as background.

1. Replace the last line of our function with the following code. Save the function after the alteration.



**Question 7** How does our function label a block of pixels as background?

|  |
| --- |
| Fungsi memberikan label sebuah blok piksel sebagai latar belakang jika standar deviasi (std2) dari blok piksel tersebut kurang dari 1 artinya, jika blok tersebut memiliki variasi intensitas yang sangat kecil (piksel-pikselnya hampir sama), maka blok tersebut dianggap sebagai latar belakang dan seluruh piksel dalam blok tersebut diberi nilai 1 (putih). |

1. Now rerun the block process (in the command window) to see the result.

**Question 8** How does the output of the new function compare with the old?

|  |
| --- |
| Output dari fungsi baru lebih adaptif terhadap variasi pencahayaan pada gambar karena ia memeriksa standar deviasi pada setiap blok piksel. Jika blok tersebut std2 < 1, maka seluruh blok dianggap latar belakang, sedangkan pada blok yang memiliki variasi intensitas thresholding dilakukan secara otomatis menggunakan graythresh. Sedangkan fungsi lama biasanya menggunakan threshold global yang sama untuk seluruh gambar sehingga kurang efektif jika ada perbedaan pencahayaan atau kontras di bagian tertentu. |

**Question 9** What is the main limitation of the adaptive thresholding function developed in this tutorial?

|  |
| --- |
| Keterbatasan utama dari fungsi ini adalah ketika standar deviasi dalam sebuah blok sangat rendah fungsi langsung menetapkan seluruh blok sebagai latar belakang tanpa mempertimbangkan kemungkinan adanya detail penting yang sangat halus. Selain itu, fungsi ini hanya menggunakan nilai threshold dari graythresh yang berbasis histogram global, sehingga masih kurang optimal untuk gambar dengan variasi tekstur yang sangat kompleks atau noise yang tinggi. |