## CSE585/EE555:  Digital Image Processing II

## Computer Project # 1:

## Morphology: Hit-or-Miss Transform

#### Hao Zhou, Fan Yang, Zerui Hua

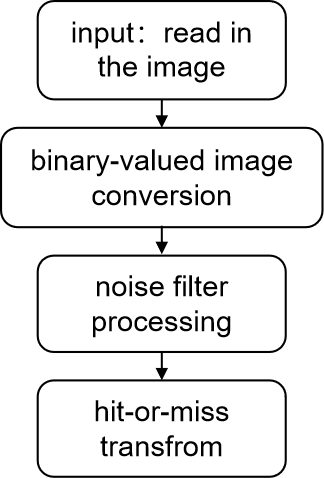
#### Date: 01/25/2023

* + 1. **Objectives** ~~(Length: 1 paragraph 🡪 a few brief statements presented as a list~~

~~Write what the project’s objectives are in a few brief statements~~.

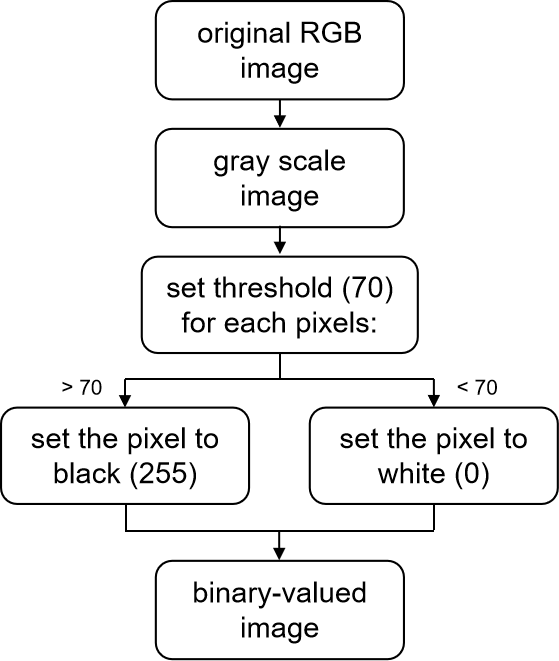
1. Learn how to use MATLAB for digital images processing;
2. Learn how to use math morphology and achieve morphological operations from simple (erosion / dilation) to complex (hit or miss);
3. Learn how the hit-or-miss operation can be used in the object detection in digital images.
   * 1. **Methods** (Length: 3-15 pages – be complete!)
4. Algorithm and Theory

The main function of this project mainly consists of three parts, including the binary-valued image conversion, the noise filtering and the hit-or-miss transform operation, as shown in the flow chart.

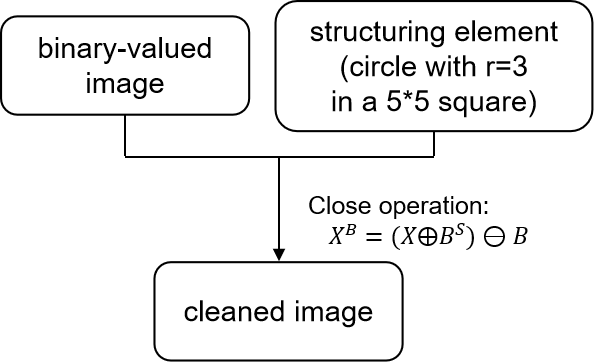


We will discuss these parts in the following report separately.

1. Image conversion: This part can first convert the RGB image into gray scale image. Next set a threshold value, classify the value of each pixels, and then generate the binary-valued image. Finally, for the calculation of the following function, we transfer the data type of the image into bool type.



1. Filter processing: In this part, we set a new function ‘create\_se’ , which can create a symmetric structuring element (a circle inside a square). With it, we can implement a closing operation (• B) in order to fill holes in regions and remove the salt-and-pepper noise. The finally output will be a cleaned image without noise.



1. Hit-or-miss transform:
2. MATLAB Implementation
   * 1. **Results** (Length: Could be many pages, including figures and tables)  
         Give all results here, be they figures, tables, or other numerical results.

Images can be in .gif format (or any other lossless format) – .jpg will affect quality because it is a **lossy** compression method.  The gray-scale images in the “*images.zip*” archive on CANVAS are mostly in .gif format.

For MATLAB, the *imread*, *imwrite*, and *imtool* commands do most of what you need to generate pictorial results.

Figures must have **complete captions (!)**, **giving image names, processing applied, and parameters used – see examples below**.

No more than **4 images** per page!

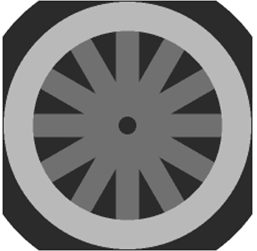


Figure 1. Original "wheel" image.

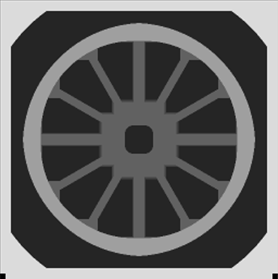


Figure 2. Result after applying an 11X11 erosion (minimum)

operator to the original "wheel" image.

Give detailed observations and/or analysis of all results.

Also, ask yourself the following general questions:

1. Are the results what you were expecting they would be? If not, why not?
2. Do you think it's because the algorithm is not the appropriate one, or because your implementation has a bug?

**Be sure to answer any questions the project write-up may pose**!

It is important that you realize when your results are wrong – this means that you understood the theory behind the project. **Partial credit is given for incorrect results when the error is pointed out and thorough explanations are given** for the incorrect results, along with possible corrections or solutions.

* + 1. **Conclusions** (Length: 1-2 paragraphs)  
       Write a clear short conclusion for your project.

Remember, conclusions **are not** observations.   You draw conclusions **from** what you observe in your results.

This section does not have to be long!