CS 484, Fall 2018
Homework Assignment 1: Binary Image Analysis
Due: October 31, 2018

The goal of this assignment is to find objects of interest in images using binary image analysis techniques.

Question 1 (20 pts)

In this question, you will write your own implementations of the two fundamental morphological operations, namely `dilation` and `erosion`. You will write a separate function for each operation. These functions will take as input a binary image (as a matrix) and a structuring element (also as a matrix), and produce a binary image (another matrix) corresponding to the result of the operation.

These functions should be generic in the sense that they are required to work with any possible image and structuring element pairs. You should use the following prototypes:

```
binary_image = dilation(source_image, struct_el)
binary_image = erosion(source_image, struct_el)
```

You should generate the structuring element as a binary image with an arbitrary shape. Given the structuring element, your code should implement the `dilation` and `erosion` operations using the definitions given in the course slides. Note that the structuring element should be created (as a matrix) outside and given as an input to the dilation/erosion codes so that your code can work with any kind of structuring element.

You are free to use any programming language for the implementation. The representation of the image data and the structuring element data (using data structures such as arrays) will depend on your choice of the language. You MUST write your own implementations of these two morphological operations. Code from other sources is NOT allowed for this part of the assignment (as an exception, you can use the `strel` function in Matlab or the `getStructuringElement` function in OpenCV to generate the arrays containing the structuring elements). Contact the instructor or the TA if you need help with image I/O using different programming languages and environments.

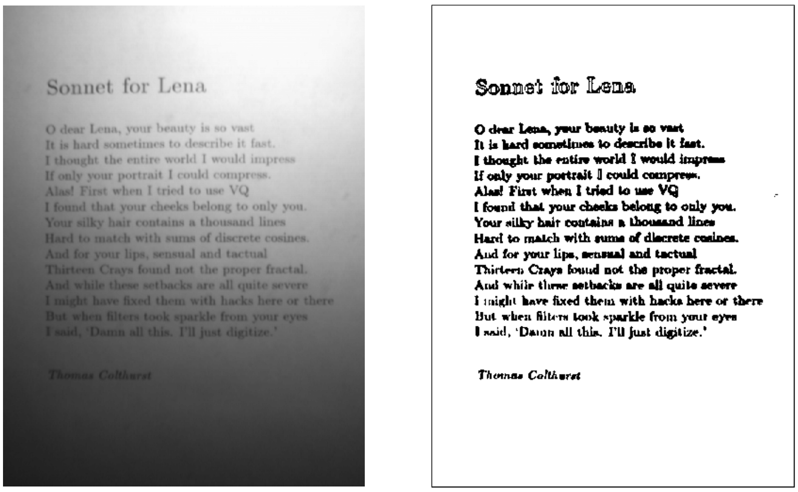
Submit: Well-documented source code in ASCII format for `dilation` and `erosion` operations. Also cite the definition you used for the implementation in the code documentation.

Question 2 (20 pts)

Now, it is time to use the building blocks that you wrote in Question 1. You are given a gray scale image of "Sonnet for Lena"[1], and you are expected to apply a sequence of thresholding and/or morphological operations to come up with its binary version, as shown in Figures 1(a) and 1(b), respectively.

To achieve this goal, one can construct different sequences of operations. For instance, you may start with adaptive thresholding and then follow with morphological operations. Other sequences of operations are also possible. Thus, any pipeline including the combinations of

[1] Taken from `https://homepages.inf.ed.ac.uk/rbf/HIPR2/liblin.htm`



(a) Sonnet for Lena (input)

(b) Example result of thresholding and morphological operations (output)

Figure 1: Sonnet for Lena.

`dilation` and `erosion` as well as thresholding, and arithmetic and logical operations is valid as long as your solution leads to a readable binary version. However, it is also important to clearly report your reasoning and findings along the way. Submitting only the output will not deserve a full score.

You MUST use the `dilation` and `erosion` code from the first question. You CAN use other sources for thresholding, arithmetic and logical operators, and image I/O. Note that you are NOT allowed to use any operator other than the ones listed above.

Submit:

1. **A report (pdf file)** that includes a description (a paragraph) and a result (resulting image) for each step you follow. You are also expected to provide a discussion of the results (e.g., which steps were easy and which were more difficult, what was possible and what was not).

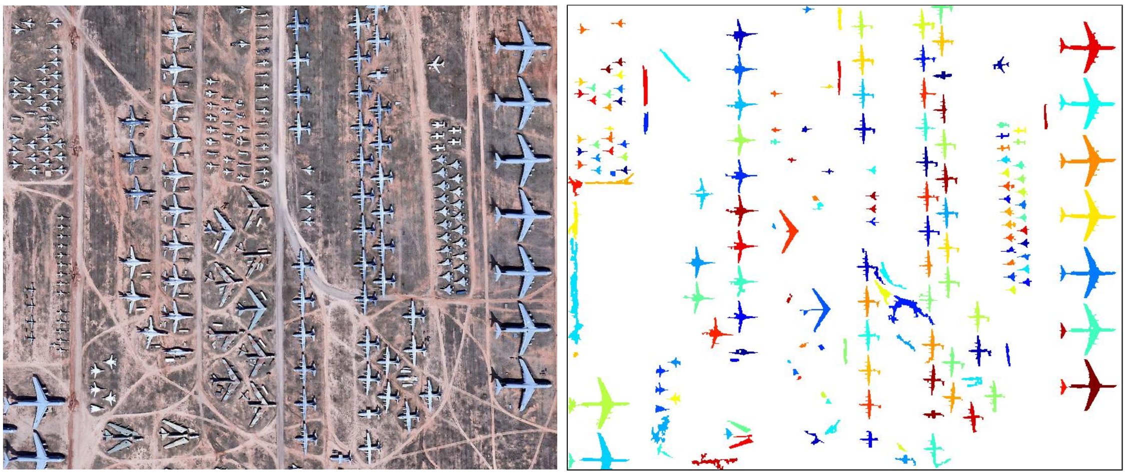
2. **A script** that runs the particular sequence of operations that reproduce the results presented in your report.

Question 3 (20 pts)

In this question, you are given a satellite photograph of an airplane graveyard[2] show in Figure 2. The goal is to identify the individual planes in this image as shown in the example output.

You can start with thresholding (hint: you can check the grayscale image or try to find an appropriate band where the planes are more clear) to obtain a binary image and continue with a sequence of morphological, logical and arithmetic operations to localize the planes. The last step is connected components labeling where each plane (the pixels corresponding to that plane) is assigned a unique integer. The output is a labeled image that has a distinct label

[2]Taken from `http://loworbittourist.tumblr.com/post/162234481509/davis-monthan-air-force-base-aircraft-boneyar`



(a) Airplane graveyard (input)

(b) Example result (in false color) of thresholding and morphological operations (output)

Figure 2: Airplane graveyard.

(integer) for each plane that you could detect. You may want to further process this image to clean up some erroneous results corresponding to very large or very small regions (hint: you can compute the area of each connected component by counting the number of pixels with that label) that do not correspond to an airplane.

You MUST use the `dilation` and `erosion` code from the first question. You CAN use other sources for thresholding, arithmetic and logical operators, connected components labeling, and image I/O.

Submit:

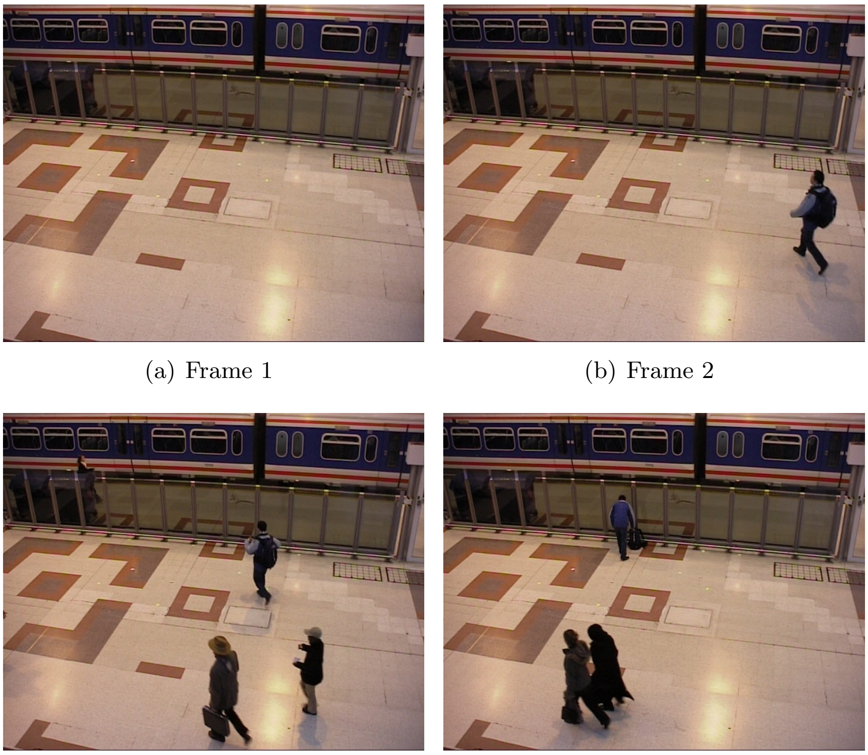
1. **A report (pdf file)** that includes a description (a paragraph) and a result (resulting image) for each step you follow. You are also expected to provide a discussion of the results (e.g., which steps were easy and which were more difficult, what was possible and what was not).

2. **A script** that runs the particular sequence of operations that reproduce the results presented in your report.

Question 4 (40 pts)

In this question, you will work on a bit more complex scenario, *video surveillance*. The goal of video surveillance is to find new objects in a scene by comparing a new image to a known static background image. The scene is monitored using a camera and the frames of the recorded video sequence are compared to the background image.

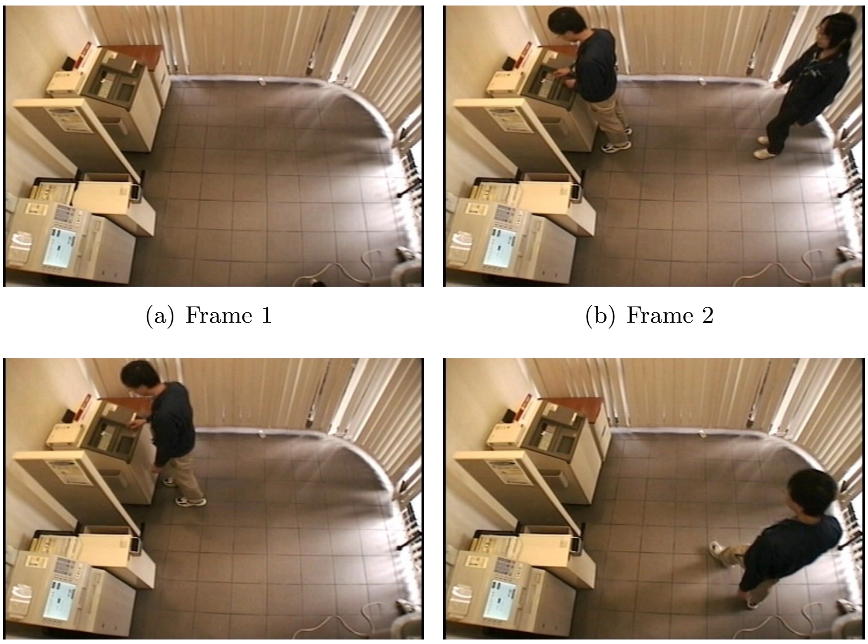
Example frames from two video sequences are shown in Figures 3 and 4. Both of these sequences were taken using static cameras (i.e., the cameras were not moving). Therefore, the differences between a new frame and the background frame are caused by external factors. The objects of interest are the moving people. The frames were taken from the Change Detection 2014 Dataset (`http://wordpress-jodoin.dmi.usherb.ca/dataset2014/`).



(a) Frame 1 (b) Frame 2

(c) Frame 3 (d) Frame 4

Figure 3: Frames from the first surveillance video sequence.



(a) Frame 1 (b) Frame 2

(c) Frame 3 (d) Frame 4

Figure 4: Frames from the second surveillance video sequence.

The first problem is to find the changed areas in a frame. This can be done using the background subtraction technique. Use the first frame in each sequence as the background frame, subtract this background frame from each of the remaining frames, and threshold the difference image to find the changing pixels. Since the original images have RGB bands, you can simply convert them to grayscale and do the subtraction using the grayscale images. (A grayscale image can be obtained as the average of the three RGB bands.)

Remember that the changes can occur because of people that appear or disappear but also because of shadows, lighting changes, etc. These noisy changes must be removed from the output. Therefore, the second problem is to use the morphological operators (`dilation` and `erosion`) you have written, to help remove these noisy changes, and to obtain the objects of interest, i.e., people, as individual connected components. You are expected to obtain the objects of interest as complete as possible (i.e., no under-detection or no over-detection).

You MUST use the `dilation` and `erosion` code from the first question. You CAN use other sources for RGB to grayscale conversion, thresholding, arithmetic and logical operators, connected components labeling, and image I/O. Note that you are NOT allowed to use any operator other than the ones listed above.

Submit:

1. **A report (pdf file)** that contains the following: for each sequence, for three frames (the other one being used as the background), the individual results of background subtraction, thresholding, morphological operations, connected components labeling (total of four images for each frame), as well as a description of the particular sequence of operations you used to obtain these results. You are also expected to provide a discussion of the results (e.g., which steps were easy and which were more difficult, what was possible and what was not).

2. **A script** that runs the particular sequence of operations that reproduce the results presented in your report.

Notes:

1. This assignment is due by midnight on Monday, October 31, 2018. You should upload your solutions as a **single archive file** that contains your **code, resulting image files and descriptions of how you obtained them** using the online submission form on the course web page before the deadline. Please see the course syllabus for a discussion of the late homework policy as well as academic integrity. If you have any questions about what is allowed and what is not allowed in a solution, please check the course syllabus on the course web page.

2. Make sure that the image formats you use to submit your results do not use lossy compression (e.g., do not use jpeg; you can use png, bmp, pgm or ppm).