# CS 484, Fall 2019
## Homework Assignment 1: Binary Image Analysis

### Due: October 28, 2019

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 can assume that the origin is in the center, or get the coordinate of the origin as a separate input to your functions.

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.

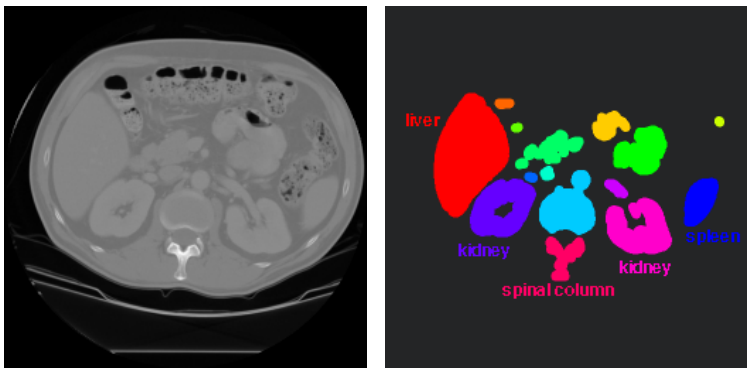
**Submit:** Well-documented source code in ASCII format for `dilation` and `erosion` operations. Also cite the definition you used for the implementation in the code documentation.

## Question 2 (40 pts)

A gray scale CT image is given in Figure 1(a). The organs of interest in this image are the kidneys, the liver, the spleen, and the spinal column as shown in Figure 1(b).

The first problem is to find a threshold that will produce a binary image that has most of the organs separated into distinct regions. You can empirically select a threshold by trial-and-error.

The second problem is to use the morphological operators (`dilation` and `erosion`) you have written, to help separate organs that are connected together or to fill small holes in organs. Once you have a decent binary image, use connected components labeling to produce a labeled image that has a distinct label (integer) for each region.




(a) Gray scale CT image (b) Isolated objects

Figure 1: CT image.

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. Note that you are NOT allowed to use any operator other than the ones listed above.

**Submit:**

1. **A report (pdf file)** that includes the results of thresholding, morphological operations, and connected components labeling (total of three images), 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). Submitting only the outputs will not deserve a full score.

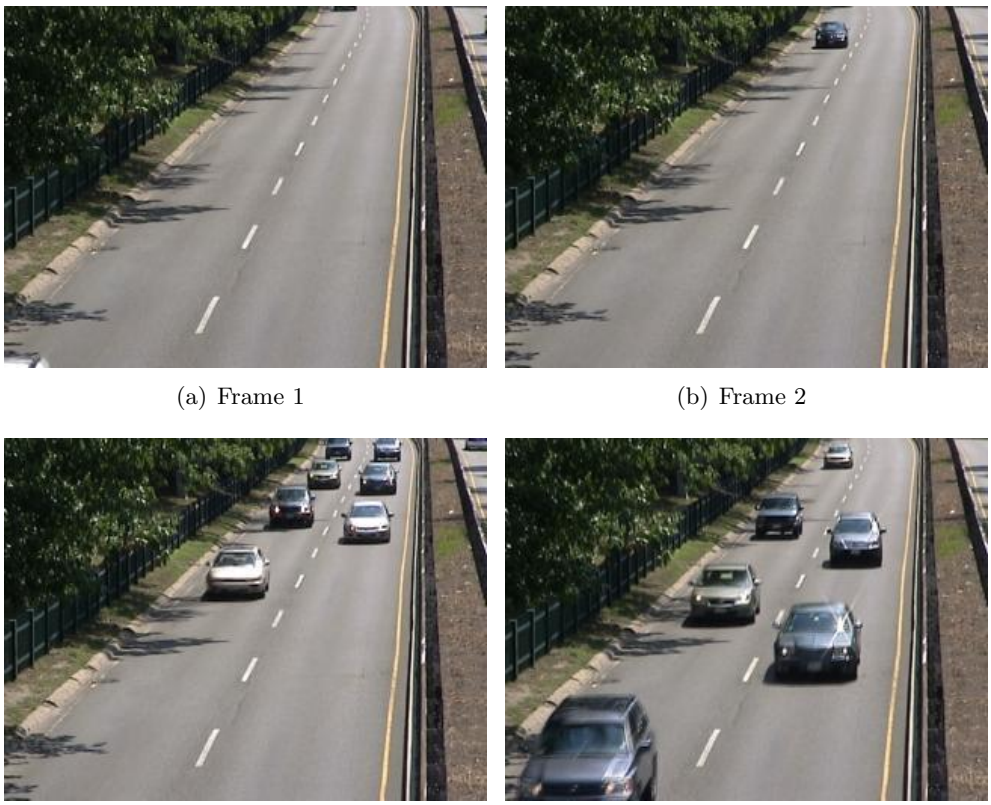
2. **A script** that runs the particular sequence of operations that reproduce the results presented in your report.

## Question 3 (40 pts)

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 2 and 3. 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 cars and people. The frames were taken from the IEEE Workshop on Change Detection (`http://www.changedetection.net`) that was held in conjunction with the IEEE Conference on Computer Vision and Pattern Recognition in 2012.

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.)



(a) Frame 1 (b) Frame 2

(c) Frame 3 (d) Frame 4

Figure 2: Frames from the first sequence.

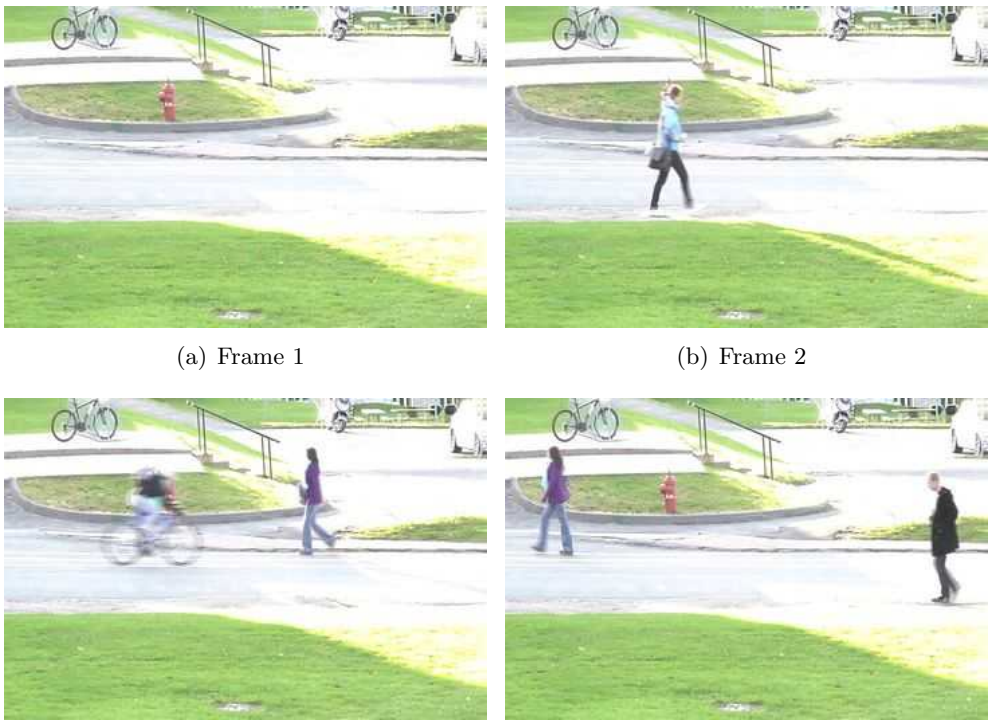
Remember that the changes can occur because of cars or people that appear or disappear but also because of moving leaves, shadows, lighting changes, etc. These noisy changes must be removed from the output. Therefore, the second problem is to use the morphological operators (`dilation` and `erosion`) you have written, to help remove these noisy changes, and to obtain the objects of interest, i.e., the cars and 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.



(a) Frame 1 (b) Frame 2

(c) Frame 3 (d) Frame 4

Figure 3: Frames from the second sequence.

**Notes:**

1. This assignment is due by midnight on Monday, October 28, 2019. 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).