For this programming assignment you will implement Hausdorff matching for object detection in binary images.

You should turn in your source code and a brief writeup. Your writeup should include example results computed using your implementation. The writeup should also explain how you addressed the scale search in part 3. Please turn in a ZIP file via email with the subject “ENGN1610 HOMEWORK 4”.

Part 1

Write a function to compute the distance transform of a binary image using the manhattan distance, using the two pass algorithm covered in class.

Illustrate the result of the image transform on the binary image of cells shown below and included with the assignment.

Part 2

(a) Write a function for computing the Hausdorff fraction under translations. The function should take a binary image $I$, a template $T$, and a distance $d$. The result should be an image $F$ showing the Hausdorff fraction for each possible translation of $T$,

$$F(i, j) = f_d(T \oplus (i, j), I).$$

where the Hausdorff fraction $f_d(J, I)$ counts the number of points in $J$ that are within distance $d$ from some point in $I$.

Your function should use the distance transform function from part 1 to dilate the image $I$. It should then compute $F$ using the correlation of $T$ and the dilated image.
You can use the matlab function imfilter for computing correlation with various boundary conditions. Using the parameters 'same' and 'symmetric' should lead to peaks in the centers of objects.

Illustrate the results of this function on the cells image by creating a template for a cell and choosing the parameter $d$ appropriately. You can crop a piece of the cells image to create your template.

(b) Write a function to postprocess the result of part (a) to generate a set of detections, by finding local maxima in $F$ and thresholding.

(c) Write a function that will display the detections from (b) overlayed on the input image. Show the results of this function on the cells image by selecting an appropriate threshold that leads to reasonable results.

Part 3

Modify the functions you implemented in Part 2 to include a search over objects of different sizes. You can do this by scaling the template using a range of sizes and combining the Hausdorff matching results at different scales. You should take care to account for the different scales during peak detection, so that each object is detected at a single scale.

Use your implementation to detect the coins of various sizes in the binary image of coins below and included with the assignment. You can create a template by cropping a single coin from the image.