Team members: Lijiang Guo, Achal Shah, Sachin Sable, Kushal Kokhe.

1 Intro

This report is a supplementary to the main report for assignment 1, part 2, car detection problem. We implemented template matching with Chamfer distance in Matlab.

2 Car detection algorithm

2.1 Edge detection

For preprocessing, we convert each image to gray scale image. Before running edge detectors, we expanded the image edges by mirroring. Then we ran edge detection operators on the gray scale image. Because of time limit, we were not able to implement non-maximum-supression and hysteresis in Canny edge detection algorithm. After Comparing results from Sobel and Laplacian operator, we choose Laplacian edge detector. See Figure 1b, Figure 3b, Figure 5b for edge detection results.

2.2 Template matching

For each image, we extracted a few cars as templates. Each template has binary pixel values. The template is then convolved with the entire image with a distance measure described below. The image path with minimum distance from the template is consider as potential car locations.

2.3 Chamfer distance transform

The Chamfer distance (Burger et al., 2009) finds the nearest non-zero image pixel to the each non-zero image pixel in the template. The distance can be computed using shorted-path algorithms such as Dijkstra's algorithm with city-block distance. Burger et al. (2009) Algorithm 11.2 also provides a recipe of computer this distance transform. Due to time limit, we were not able to completely implement this function. Instead we used `Matlab` function `bwdist` as suggested in this reference[1]. All other functions are implemented by us in `Matlab`.

After distance transform, the template is convolved with the image to compute mean Chamfer distance for each patch. In Chamfer matching, the observed distances are very similar near a real interest point. Therefore, we want to only keep the local minimum in a local area

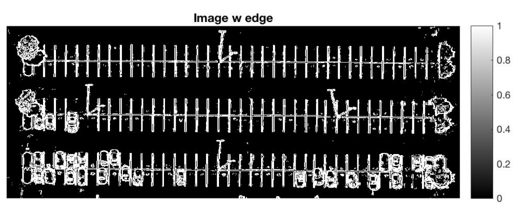
[1] http://www.cs.bc.edu/ hjiang/c342/lectures/matching/fit-I.pdf

and discard the rest in the same local area. This local minimum is the most likely position of a car.

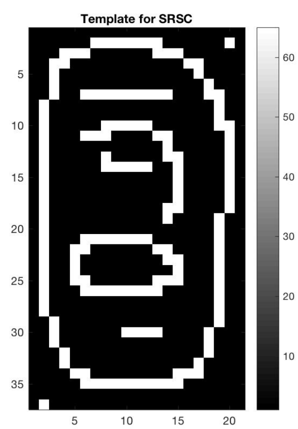
We show the detection results in Figure 2, Figure 4, Figure 6.

3 Results

(a) Input image (SRSC).



(b) Edge detection (SRSC).



(c) Template (SRSC).

Figure 1: Car detection for SRSC image.

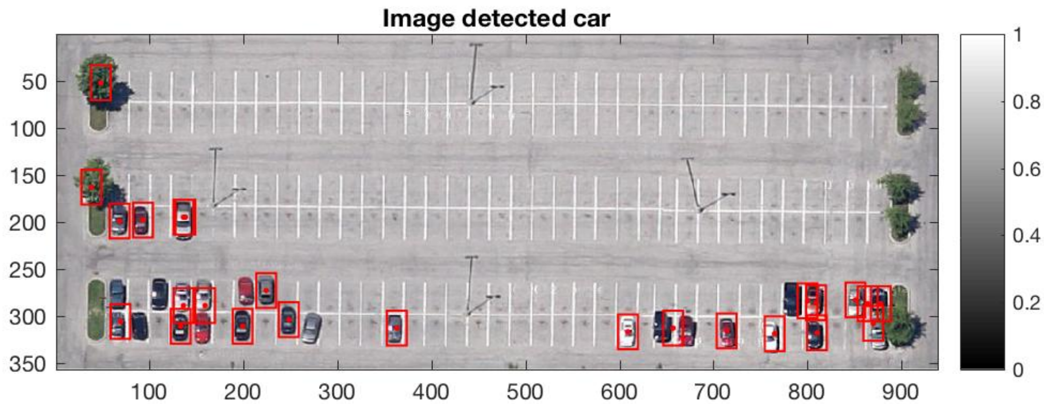
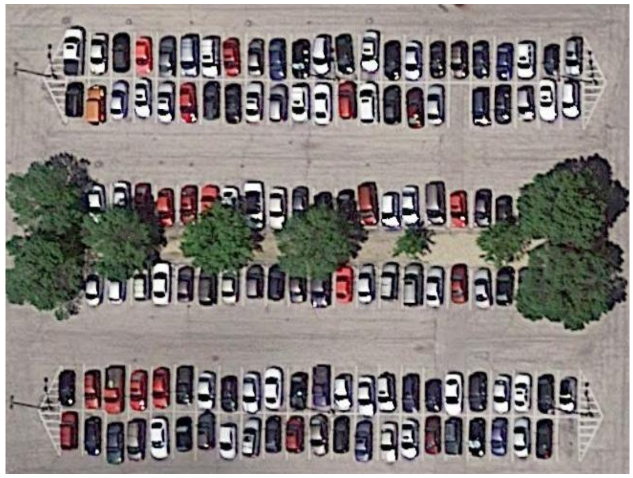


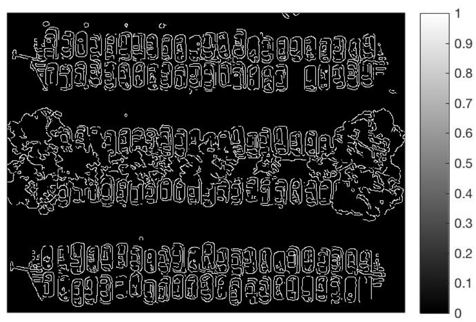
Figure 2: Detection results (SRSC).

References

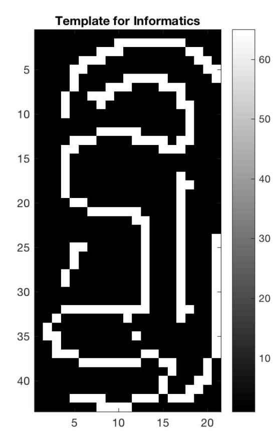
W. Burger, M. J. Burge, M. J. Burge, and M. J. Burge. *Principles of digital image processing.* Springer, 2009.



(a) Input image (Informatics).



(b) Edge detection (Informatics).



(c) Template (Informatics).

Figure 3: Car detection for Informatics image.

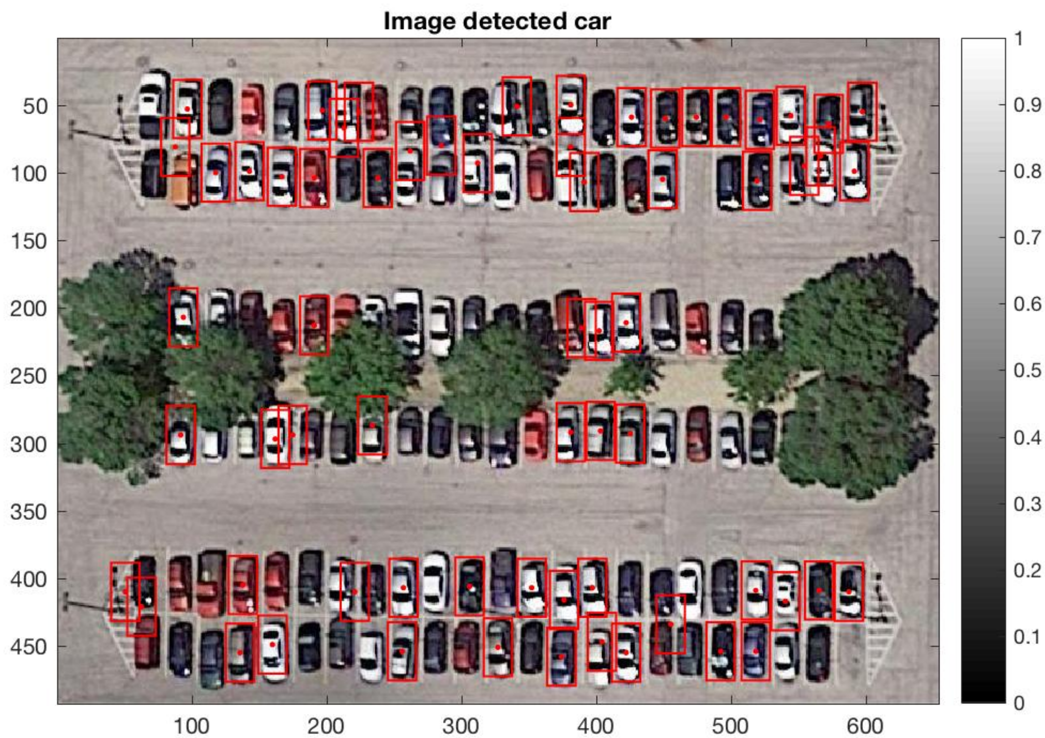
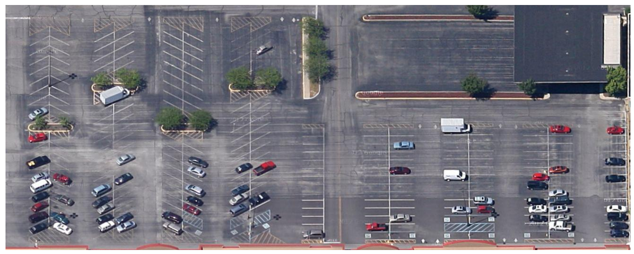
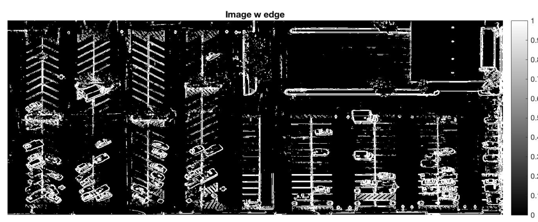


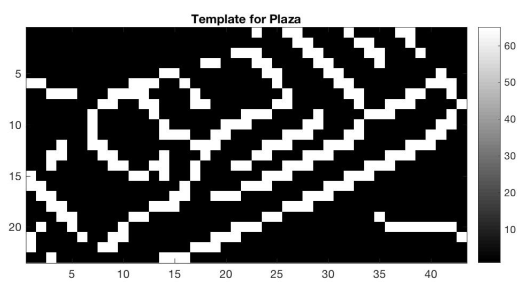
Figure 4: Detection results (Informatics).



(a) Input image (Plaza).



(b) Edge detection (Plaza).



(c) Template (Plaza).

Figure 5: Car detection for Plaza image.

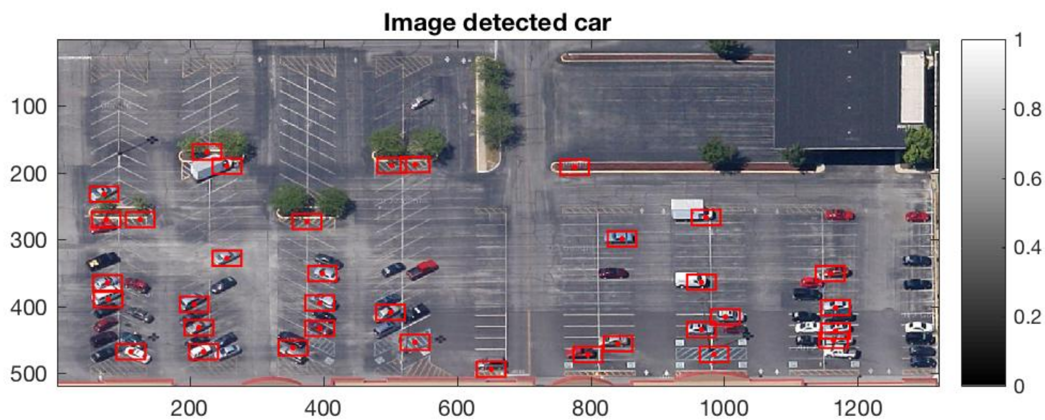


Figure 6: Detection results (Plaza).