Haoran Man

ELEC345 HW5

Solving correspondence

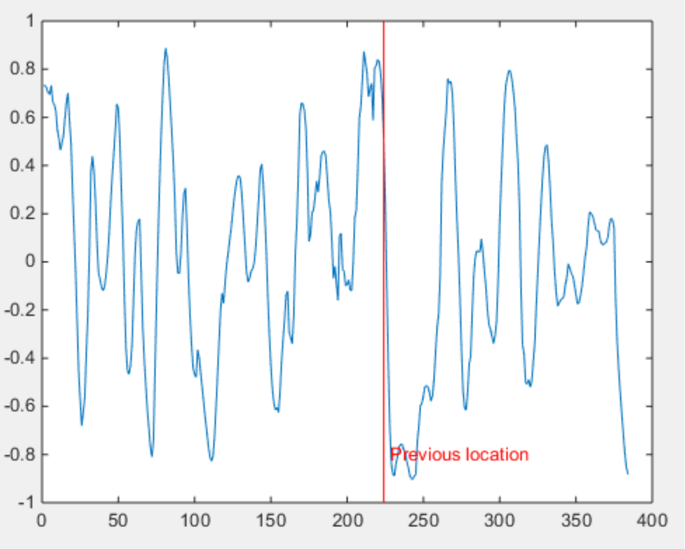
1 I used patch size of 30. In this picture, 30by30 can give us enough detail to distinguish most features in the picture.

2 I used Matlab function normxcorr2 to get normalized cross-correlation, and then find the maximum point in the line on the right figure.

I also used vline function to label a line at the position of the point at left image. (By Brandon Kuczenski for Kensington Labs)

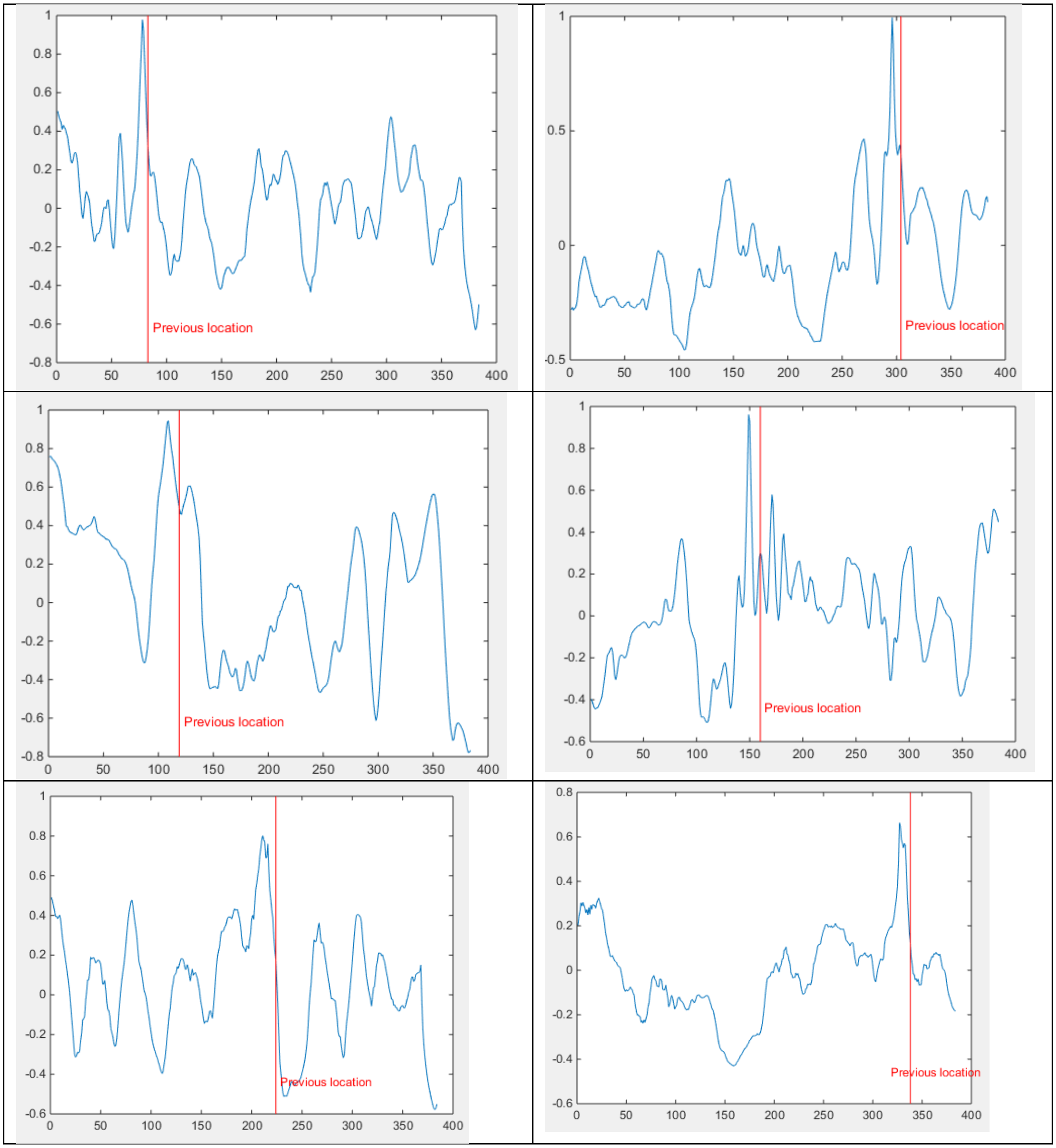
For the first four images, the correlation is pretty good (around 1). For 5 and 6, the correlation is lower. In this case, if we choose a batch size lower than 30, the position for the corresponding point might be off.

For P(5), patch size of 15, we get a line below:



In this case, the global maxima is not the correspondence point.

3 The result for all 6 point is listed on the next page.



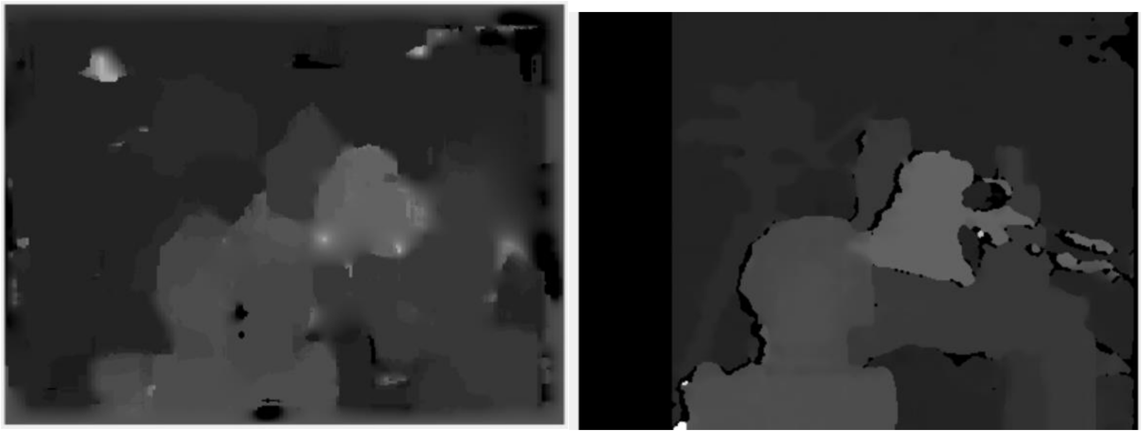
4 For the entire image’s disparity map, I used double loop over all pixel point row by row to get the corresponding point for each pixel on the left image. For the points with normalized cross-correlation larger than 0.6, I kept the disparity value and saved it to the disparity matrix, all other points are marked as NaN. After this, after checking the result, many point got a disparity larger than 150 pixels, which is not real in our case, I used another function setNaN() to get rid of the points with unreasonably large disparity. It took my computer a lot of time(10 min) for the disparity map calculation because my multiple loops. The standard disparity function given by Matlab will produce a much better result for much less time.

5 Then I used inpaint_nans by John D'Errico to interpolate the NaN in my result. I used one of his method which solves for BVP problems using linear function with least square approach.

Here is my disparity map result. On the right is the Matlab disparity function result.

(Both intensity magnified by 7)

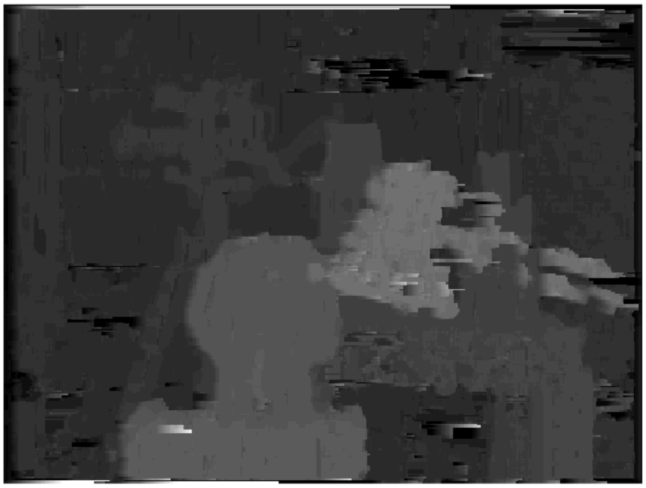
The result looks promising, clearly improvement can be done on my code.



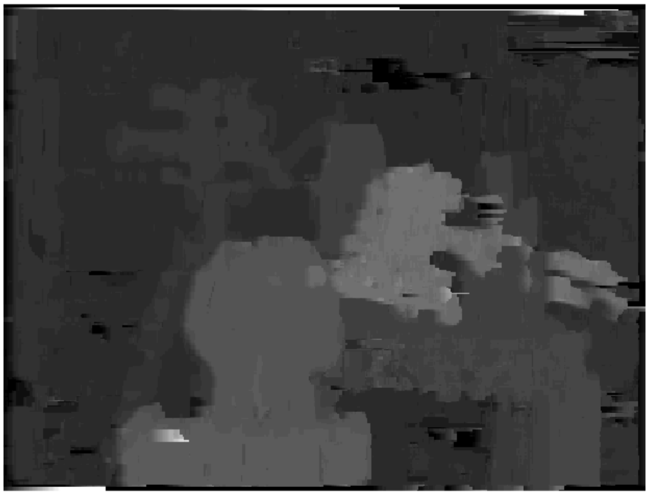
Scanline stereo

1. This part of the paper mainly discussed about the way to do both intra and inter scan line searches to get accurate disparity map. Previous researchers did intra-scanline search first and then using cooperative process to get consistent result. But this method may not yield optimal result. Also, brute force method is not feasible because of the data complexity. By doing dynamical programing, the both intra and inter scanlines can be considered simultaneously to obtain optimal result.
   The ordering constraints will eliminate a lot of unreal pairs and narrow down the selection pool. Smoothness will also help to choose the most possible edge as there is only one true pair in reality.
2. For the implementation. I used dynamic programming to get the optimal disparity for each line. For each line, I used my function DynamicLine(left, right, patchsize, linenumber, dim) to calculate the disparity.
   Basically, I get a matrix of all normalized cross correlation for each point of the left scanline with each point from the right scanline. Then, I used dynamical programing to get the path that maximize the sum of correlation from top left to bottom right. This path will be the disparity for this line. The result is much better than previous method!

Patch size 6:



Patch size 10:



Patch size 16: