Computer Vision
Exercise 3
Amrollah Seifoddini

Firstly, I resized the images and shrink them by factor of 4. Because, they had huge resolution and may PC RAM was not sufficient for processing them.

Part 1)

For this part, I strictly followed the guidelines provided by slides and TA. Firstly, I used gradient function to compute Ix, Iy. Then I blurred three components that were going to be used in Harris matrix. After that, I used the expression suggested by TA to compute the Harris strength values. For the threshold, I played with the value until found a proper value for it. Then using the non-max suppression I selected key points that were max in their neighborhood and not on the border of image.

This procedure repeated for the second image. The results are as follows:

These images are for the case when I used non-max suppression. I could not see the result without applying non-max suppression because my graphic card hangs in this case. So, I could not put any picture of this scenario in my report. I think it is because of high number of keypoints in this case that my old graphic card crashes. It is obvious that when I do not apply non-max suppression, many of keypoints that are not max in their 3*3 neighborhood will be included in the result and will crowd the picture with non-relevant overlapping descriptors that are not discriminative enough and will result in some bad matches.

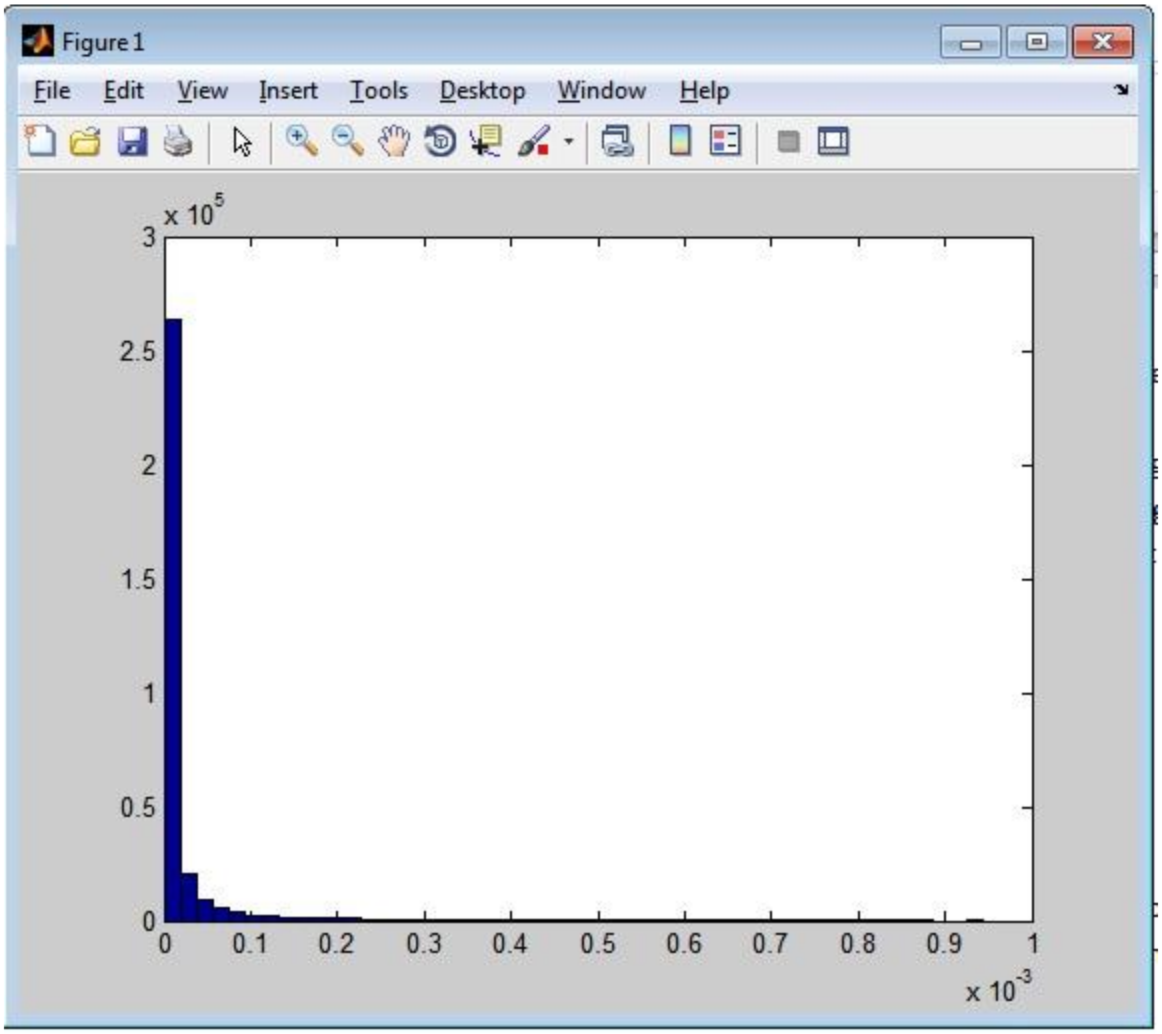
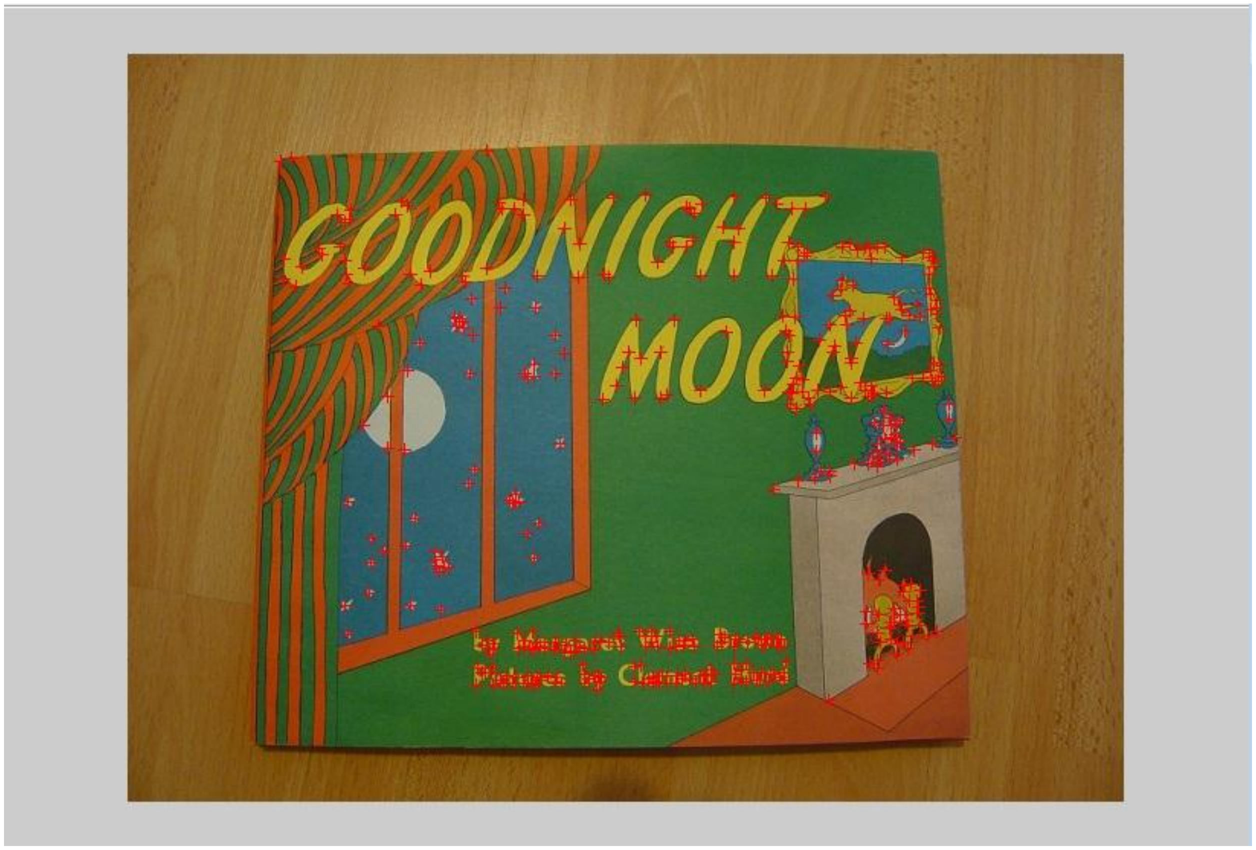
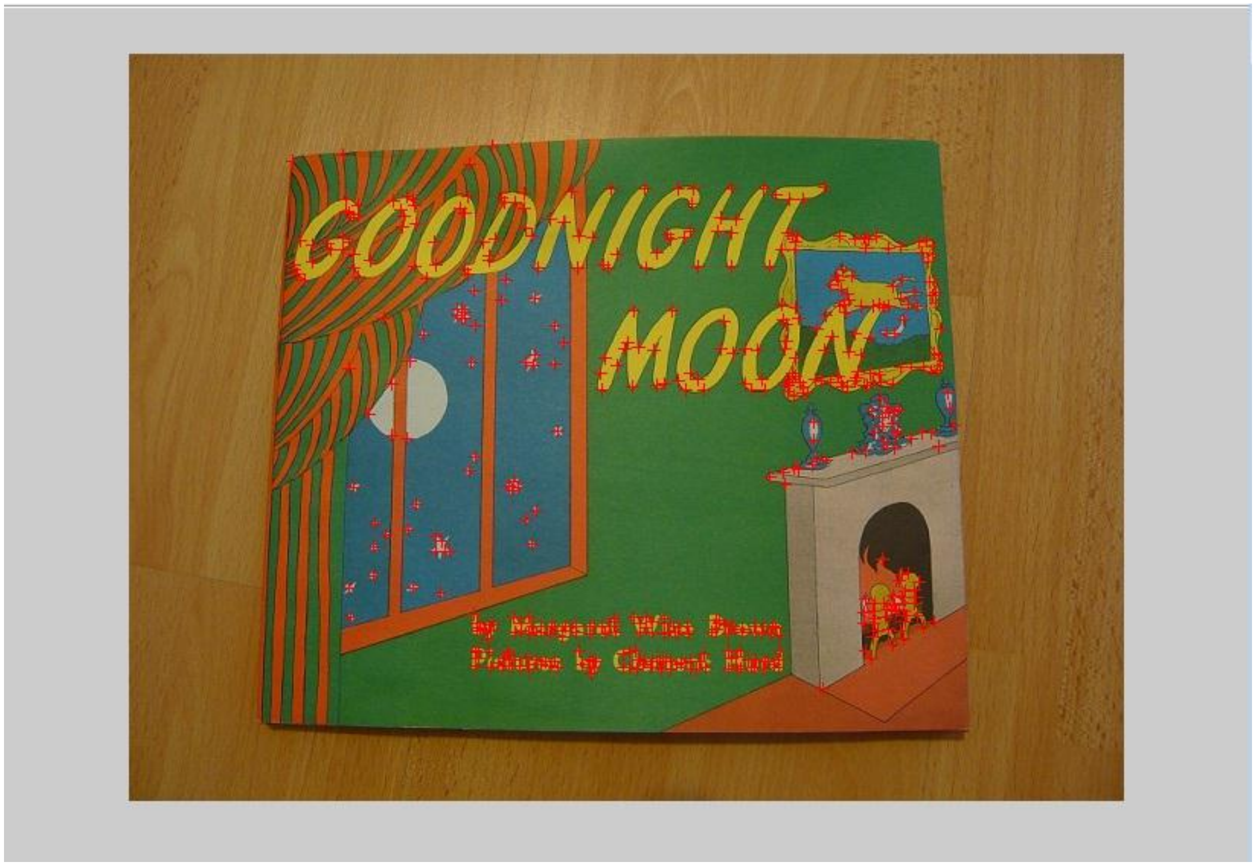


Figure 1
File
Edit
View
Insert
Tools
Desktop
Window
Help
x 10^5
3
2.5
2
1.5
1
0.5
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
x 10^-3


GOODNIGHT MOON
by Margaret Wise Brown
Pictures by Clement Hurd



Part2)

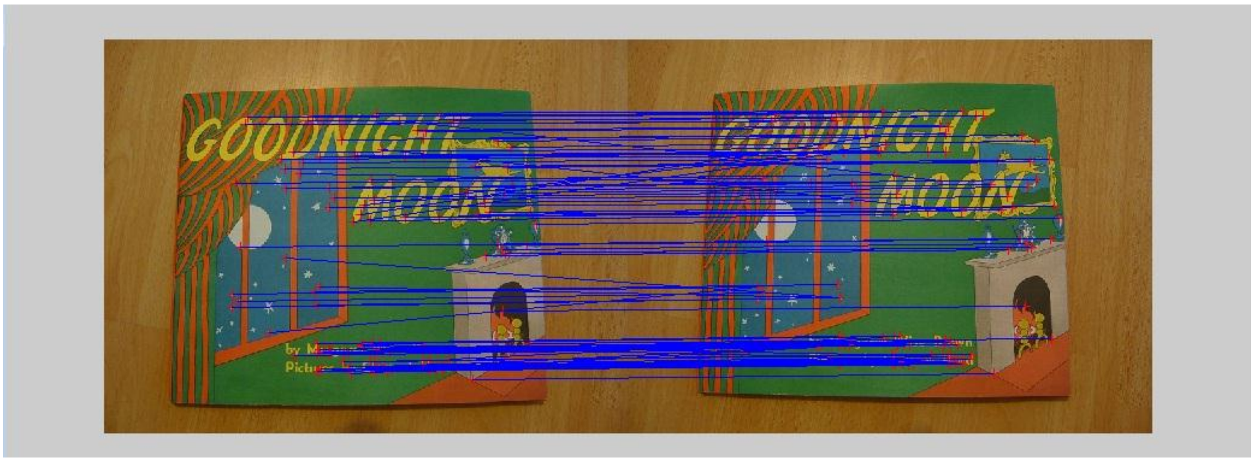
For this part, I formed a 9*9 patch around the keypoints returned from part1. Then with a loop, I extract the image values in that patch, but with condition that the entire patch should fit in the image boundaries.

This procedure repeated for the second image.

Part3)

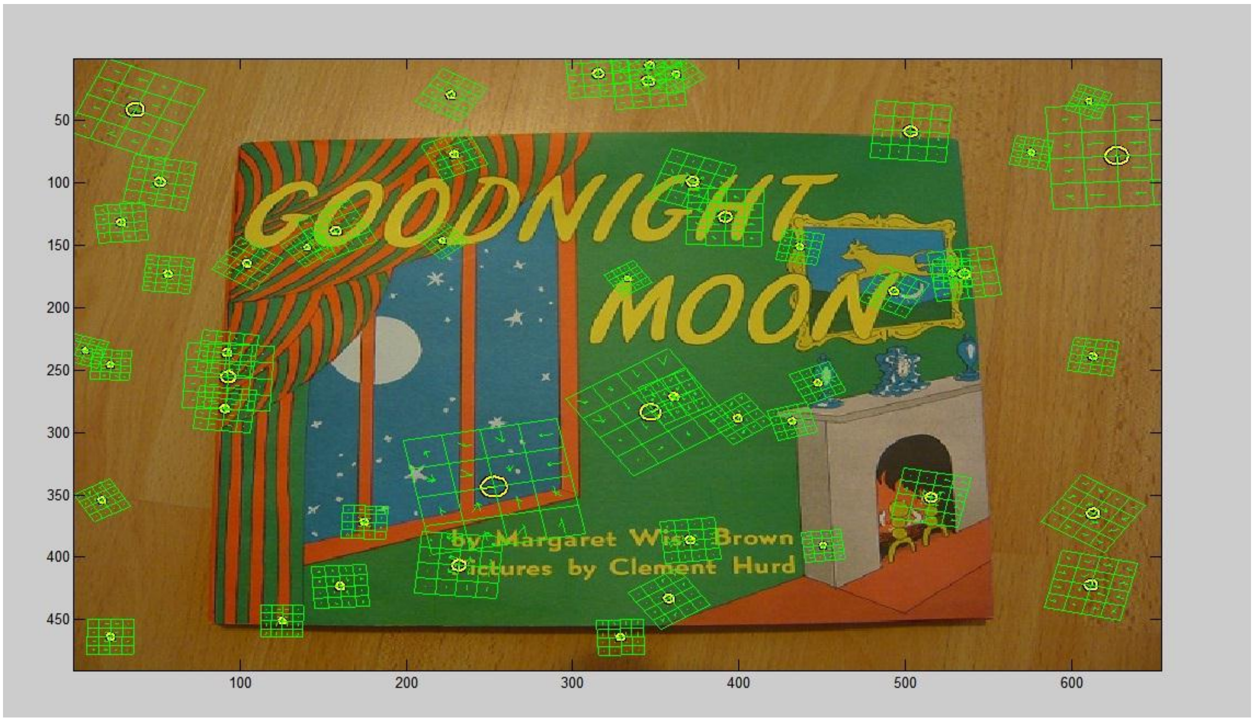
In this part, I just implemented a two level nested loop for iterating over keypoints from image 1 and then found matches in keypoints in keypoints from image 2. The distance measure I used is sdd. The threshold of this part is also set empirically which is 0.2 now. In the inner loop I calculate sdd and store it for all keypoints from second image if the they are below the diss-similarity threshold. After exiting the inner loop, I apply a min function to sdd and find the minimum of that, which give me the best match for ith keypoint in image 1. Of course I store the coordinates of this match if exist. After exiting the outer loop, we have all matched keypoints between two images.

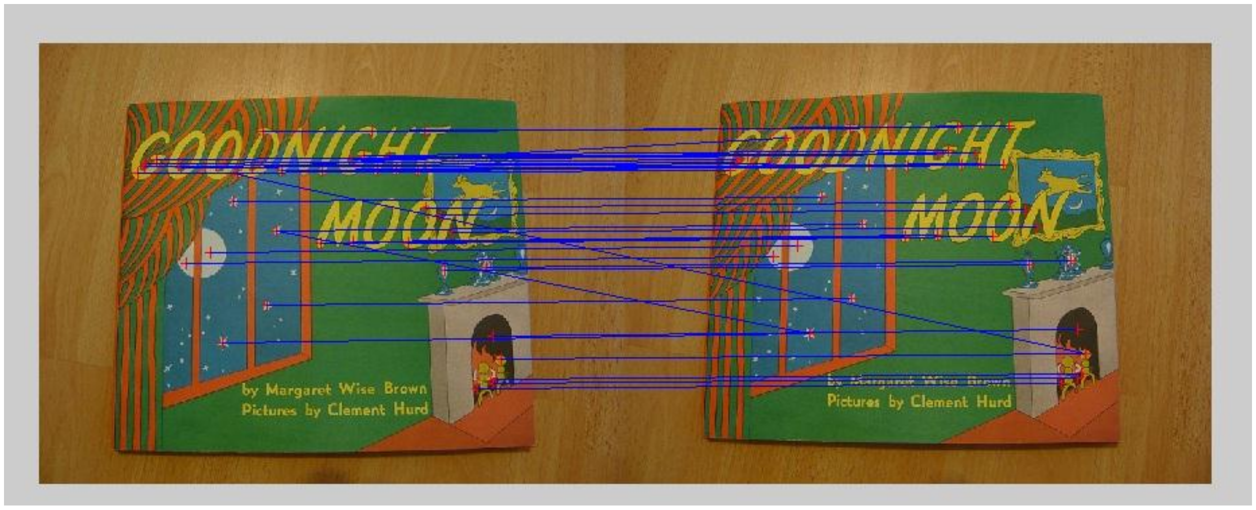
The results are as follows:



Part4)

In this part, I installed the SIFT feature extractor and applied it to my images. I followed the this page (http://www.vlfeat.org/overview/sift.html) for instructions. Firstly, I showed the descriptors and then used the matching algorithm provided by toolbox. The results are shown below. As we can see, the matches are somewhere not correct and the overall quality of matching is not as good as our descriptor. But I think it is because I used the default configuration parameters in the SIFT algorithm. Maybe if I tweaked it the results would have been better.




GOODNIGHT MOON
by Margaret Wise Brown
Pictures by Clement Hurd