**CSE 5524 - Homework #9 11/1/2016**

**Chandrasekar Swaminathan (swaminathan.42)**

Template Matching:

1) There’s an elephant in the room. Can you find it? Search for the template

template.png in the search image search.png using color-based SSD, SAD, and NCC

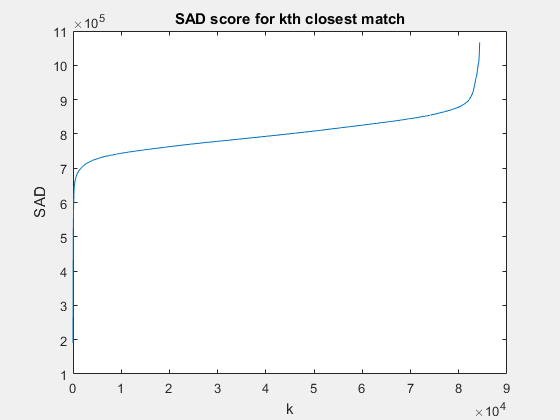
(make sure the standard deviation is “unbiased” with N-1). Assume the origin is in

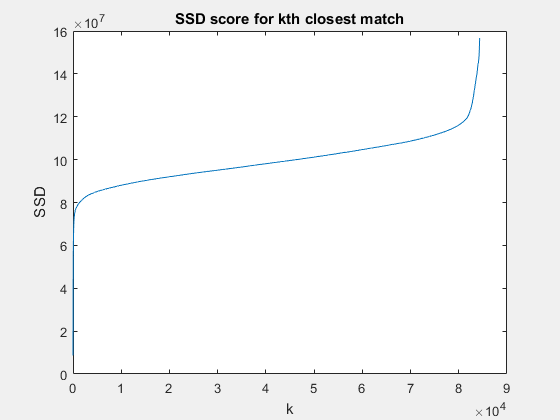
the center of the template image for each approach (Note: there should be a border

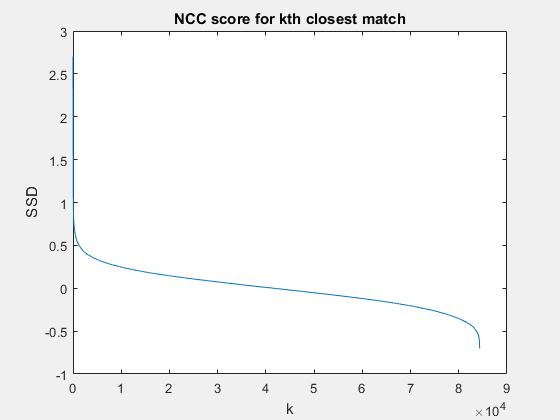
around the search image where the metrics cannot be computed).

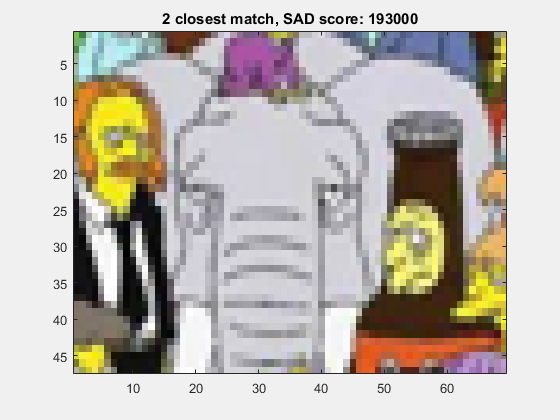
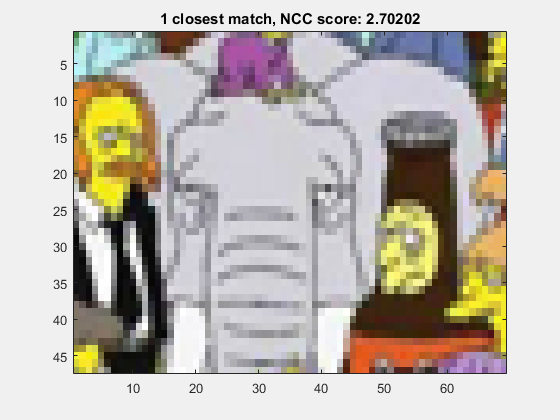
Sort the resulting scores from best to worst for each of the approaches. Plot the sorted

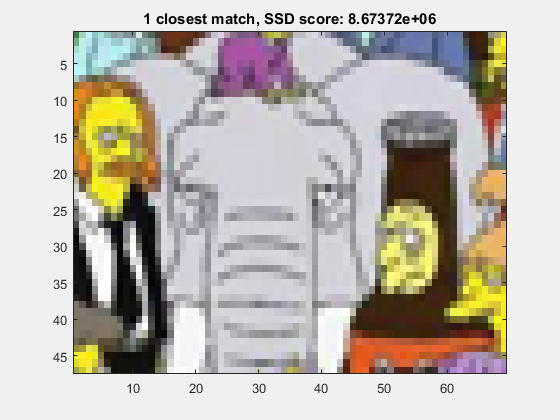
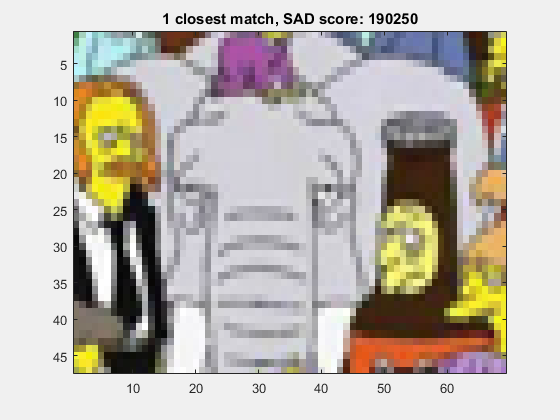
scores and show the patches corresponding to the 1st, 2nd, 5th, 10th, 100th, and 500th

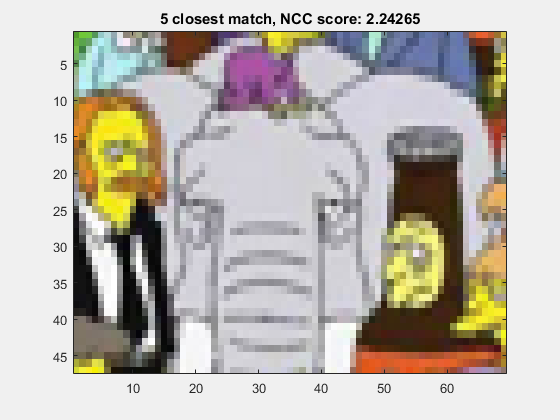
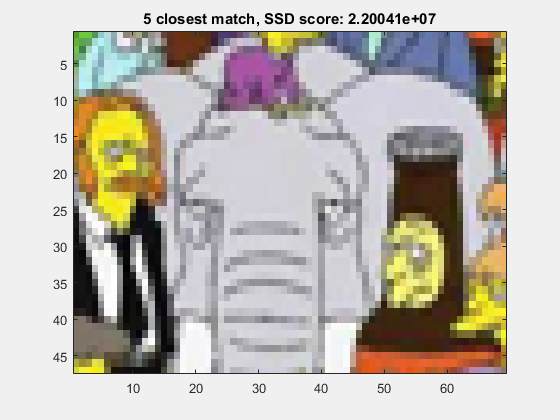
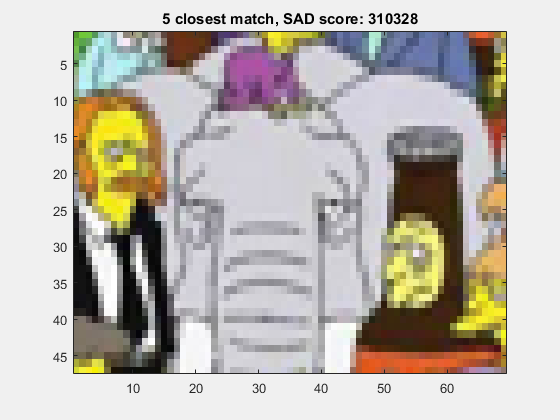
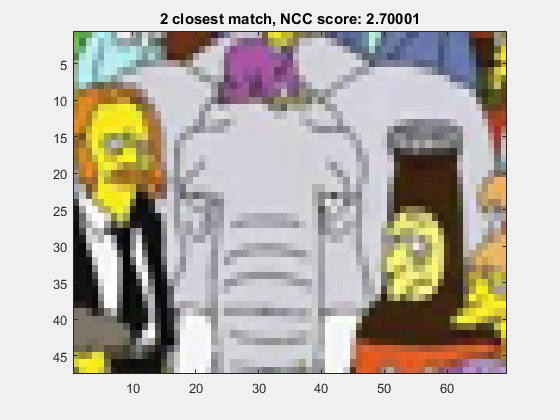
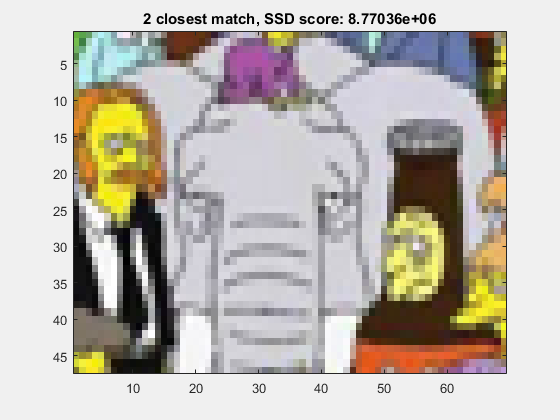
closest matches. Compare your results.

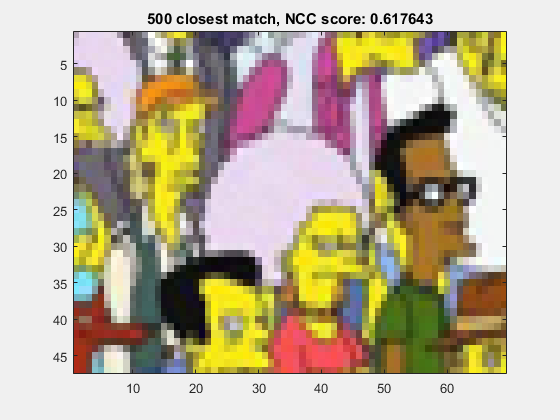
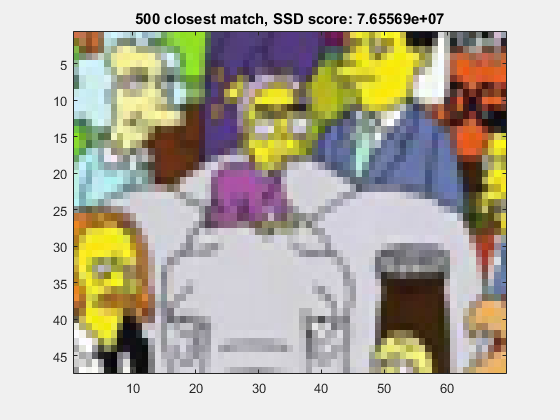
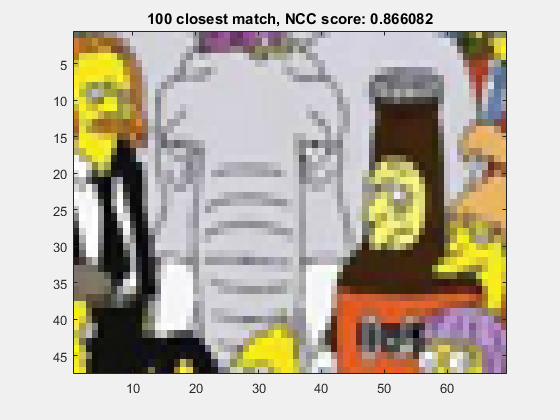
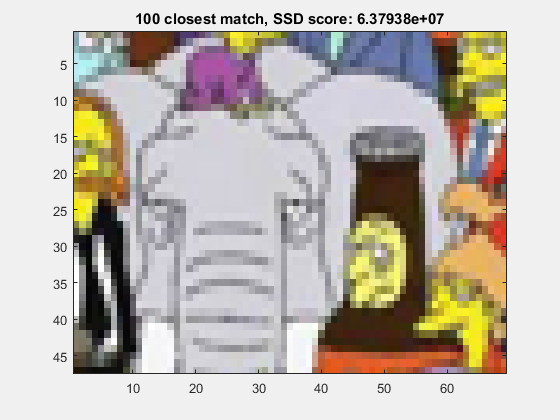
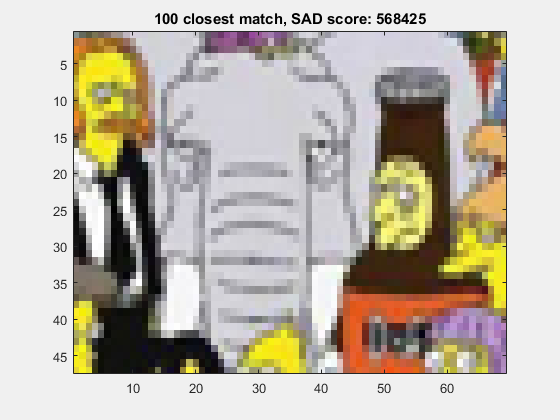
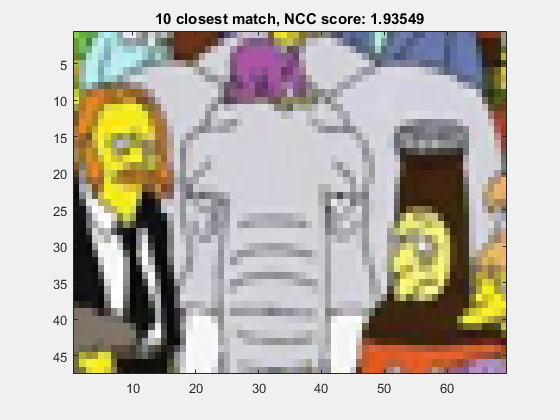
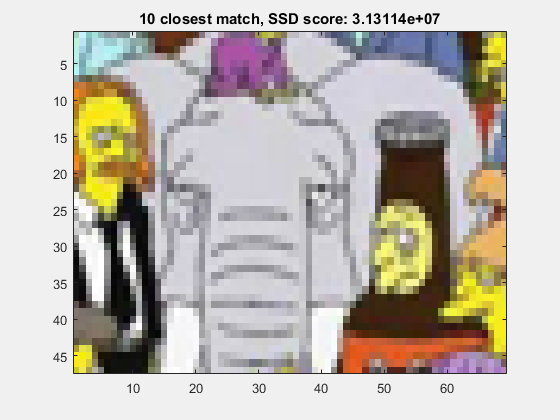
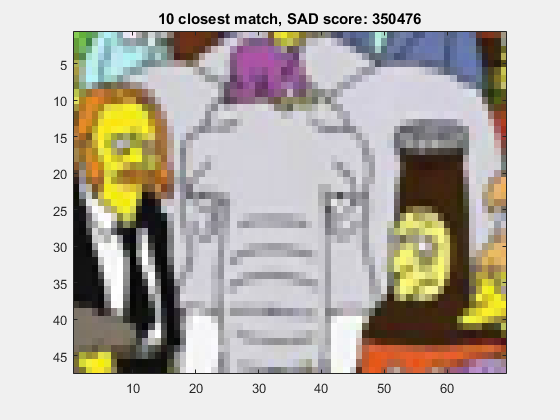


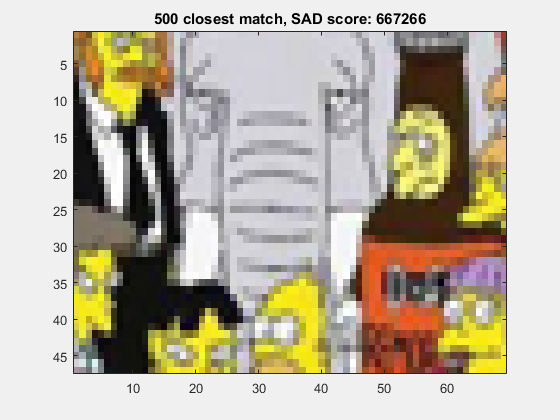












Stereo/Disparity:

2) Compute a disparity map for the images left.png and right.png (having parallel

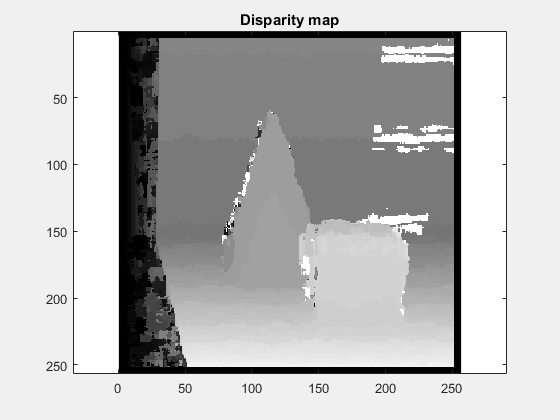
optical axes) using the basic stereo matching algorithm. Use your NCC function to

perform the template matching for each patch in the left image searching in the right

image (search only leftward from the starting point along each row!), and use a

window size of 11x11 pixels. Use the following code to display the disparity map D

with a gray colormap and clip the disparity values at 50 pixels.



Classification:

3) Write an implementation of the simple k-Nearest Neighbors (kNN) algorithm to

classify data points.   
Use the points in file train.txt as training data (this file contains 1 row for each data

point where the first two columns are x-y coordinates and the third column is the

ground truth classification label). Classify all the test data points in the file test.txt

(formatted in the same way) using K=1. Calculate and report the accuracy of your

algorithm (compared to the third column of the test data). Plot the test data points,

color coded by the class label your algorithm gives (use plot() options 'r.' and

'b.'). On the same figure (use hold on/off), (re)plot the points which are

misclassified (use plot() option 'ko'). Repeat this for K=5, 11, and 15. Compare the

results for different values of K.

