Exam 1

Oscar Galindo

1. For this problem the first thing to do is to compute the red index of every pixel (i.e. R\_i = 2R – G-B). After computing such an image then what is to be done is to compute the integral image using the summed area algorithm. For the sake of finding a stop sign I will assume a single stop sign exists. Having computed the integral image, I would then compute a range of windows (e.g. 128x128, 64x64,…) and then finally stop when the difference of the area inside of the current window is around 40/60 when compared to the outside window. For the sake of this problem I will only select a predefined size to be around 10%x10% of the original image.
2. For this problem given that the background of the images is uniform then what can be done is that the image may be segmented. By doing segmentation we can extract the number of distinct regions -1 from the segmentation since ideally all the coins would have their own region and the background would too.

Extra:

For this problem I would first segment the image. Then from every group/segmentation cluster I would create a representative vector/3-channel pixel. Such average pixel would then be used to calculate the “share” intensity of the green channel with respect to the combined intensity of the three channels. If the intensity is above 40% then normally the pixel is said to be green, hence the are must also be green. By knowing the “tag” of the green cluster I could then compute the coordinates of the all the pixels in the cluster and then find opposite corners. Since not all corners can be found at once then I would use a second segmentation approximation of the image to find points that are close to the true corners of the green area. By having pairs of points all that is needed is to calculate the “m” of every pair of points and approximate the y coordinate of the point directly below the opposite corner of the image. For example:

With the X coordinate of this point approximate the Y coordinate of the edge below.

Calculate delta Y by multiplying m times x of the point above

Approximate slope of line by using the pair of points

The previous process could then be repeated to obtain the other corner (top-left-corner). Once the corners are calculated then it is possible to match the corners of the green area (as shown previously) to the corners of the image that is to be shown there (frame taken by the camera) and compute the homography to transform the frame taken by the webcam into the size and shape of the TV with a projective transformation.

Lastly, this can be done repeatedly to show a live feed at the green background.

1. For this problem I would first allow the user to select the area of the green screen. Then I would perform an analysis on the selected region during which I would keep as “is” all the pixels that are not mostly green (i.e. have an ok threshold of green woth respect to the other channels) and would make black all those pixels that are mostly green. Then I would apply the reversed mask that is applied to the target image keeping as is all the corresponding pixels of the src (video frame) image that are “green” in the target image. And, correspondingly making all pixels that are not green in the target image black for the video frame that was warped. Finally, I would simply save a frame that contains the addition of the masked target and the masked src for every frame in the video.
2. For this problem I would basically allow RANSAC to take the correspondences built by the bruteforce matcher and then clean them based on what set of points or model ends up having the most matches from the points that were found by the matcher.
3. For this I problem I would simply create the histogram of gradients of the image using 8 different buckets. Then I would observe what’s the entire sum of gradients among all intervals of direction and if the intervals between 0-45 and 46-90 have high values probably above 30% of the combined value then the image does in fact contain a man-made structure.