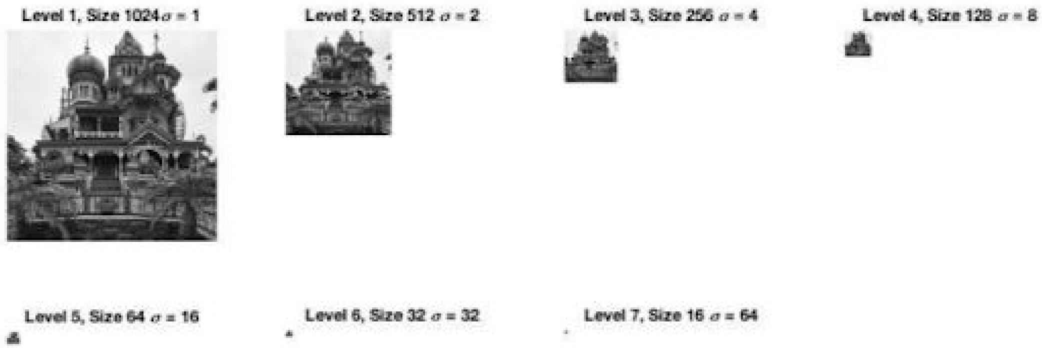
Aidan Wadin - 260716182
COMP 558
Assignment 2

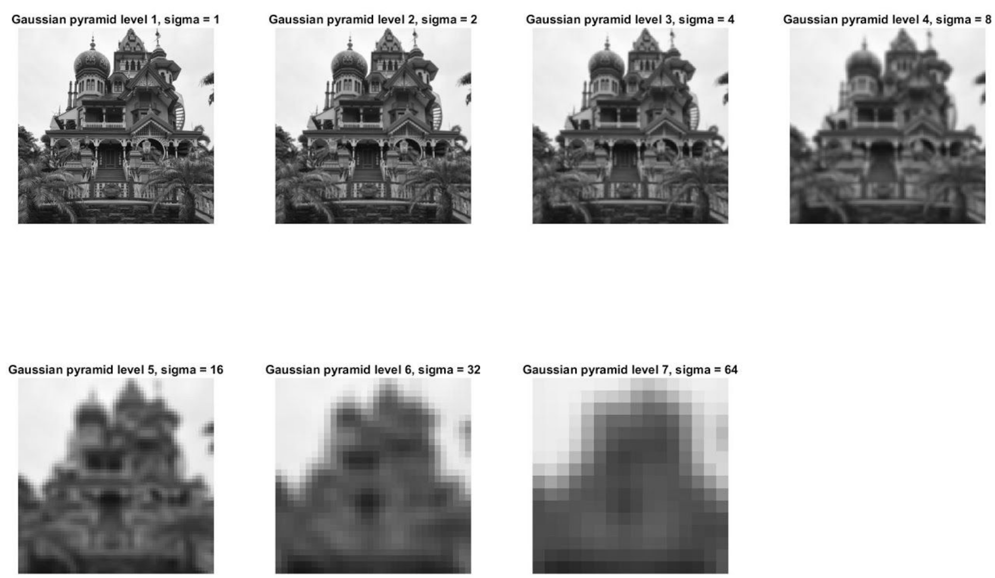
<u>Running my "main" method will return all of these images as well as extras. Some images from that output are not included to keep this report small, but it can easily be seen what each image represents as it follows the order of this paper.</u>

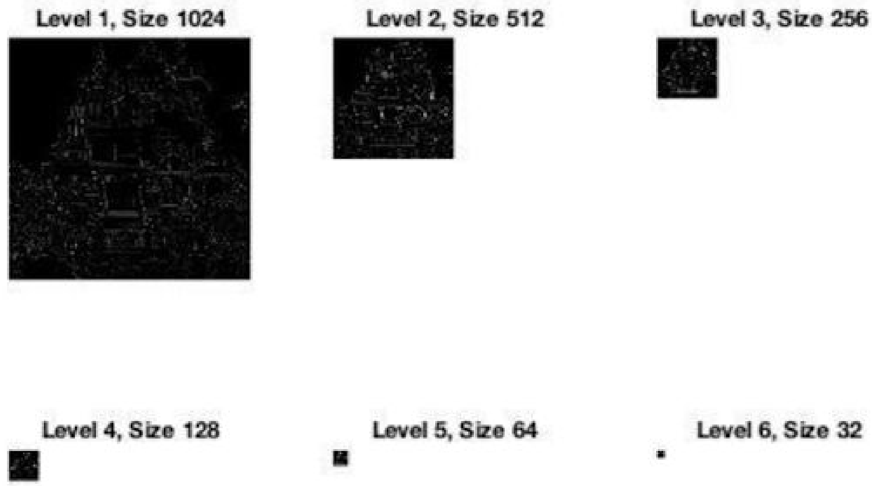
1.

Here is the original picture at its original size of 1024. Below are the gaussian pyramid levels:



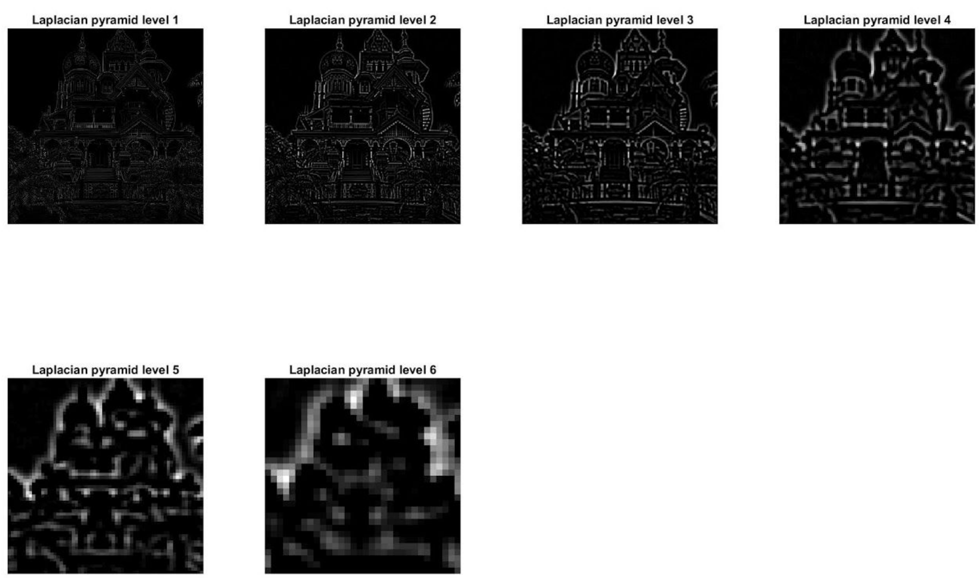
Or, in equal sizes (They are still downsampled, but kept the same size to see the blurring effect):



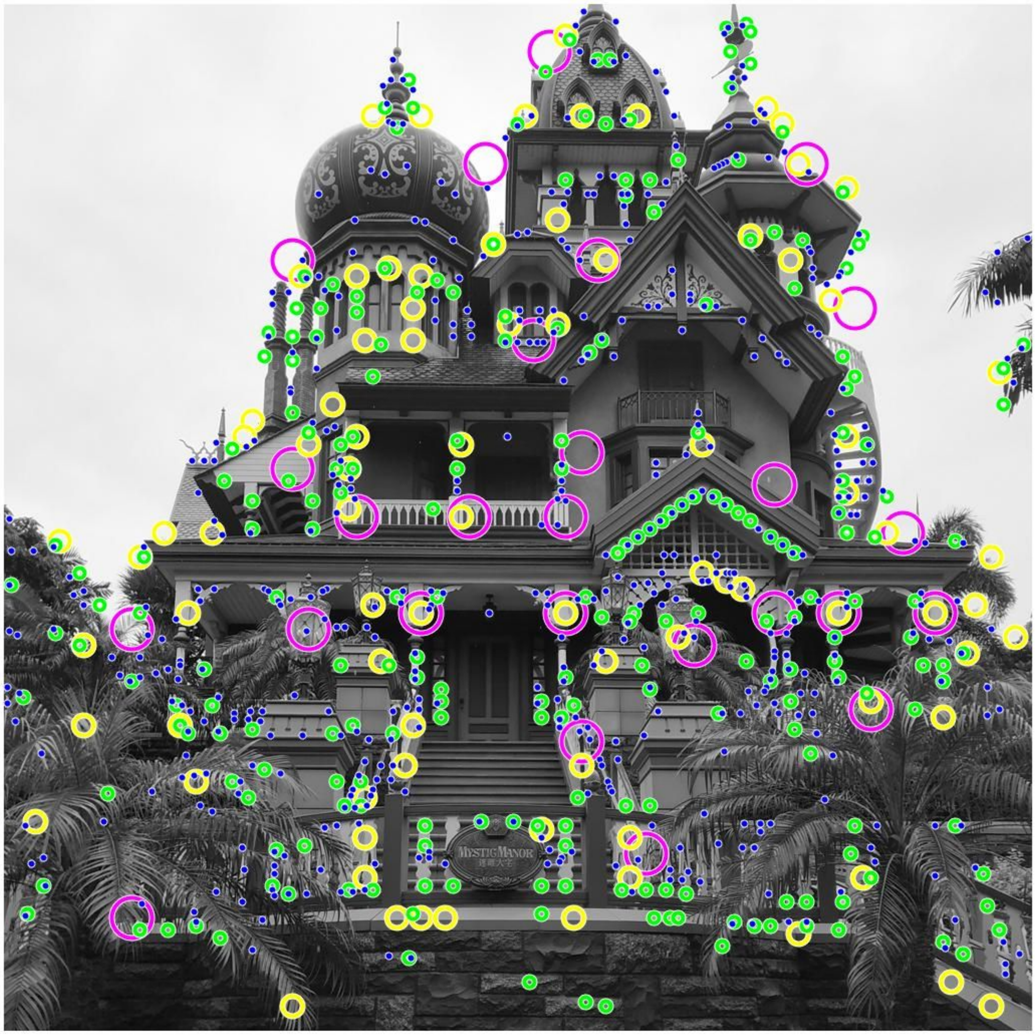


2.

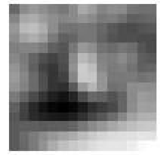
This is the Laplacian Pyramid.
Or, in all equal sizes:

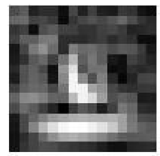


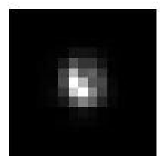
3. Sift Keypoints:



Please NOTE: In my assignment, my thresholding occurs early.. My thresholding occurs when looking at the Laplacian pyramid. It simply ignores pixels that are darker than 20 (mostly the sky). I understand thresholding should occur when computing if a point is a max/min locally and within its given levels (max > point + thresholding or min < point - thresholding), I ran out of time to refactor my code :(. ALSO, there are only 4 levels here due to the nature of the SIFT finding algorithm needing a level both above and below, so it uses Laplacain levels 2-5.



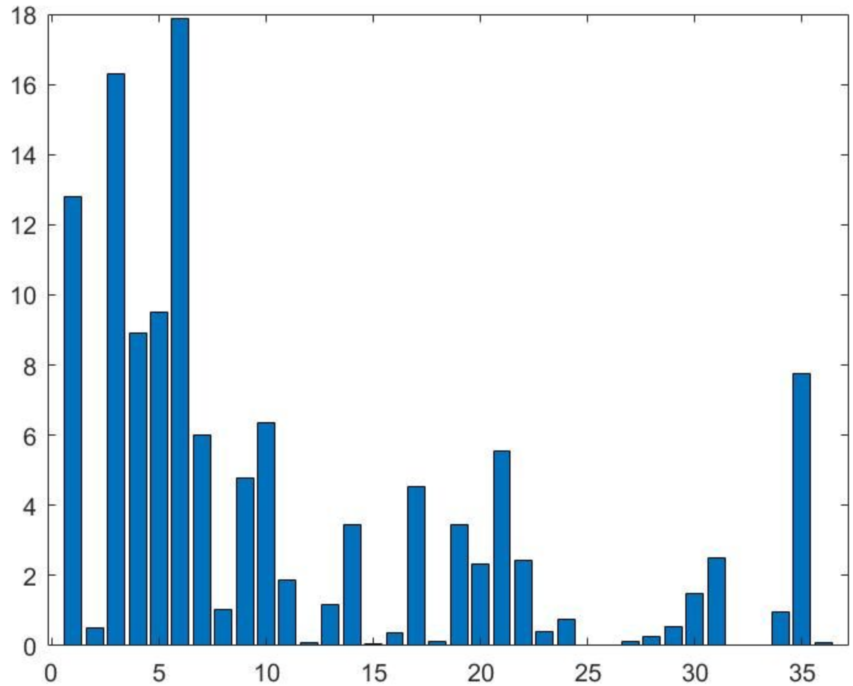


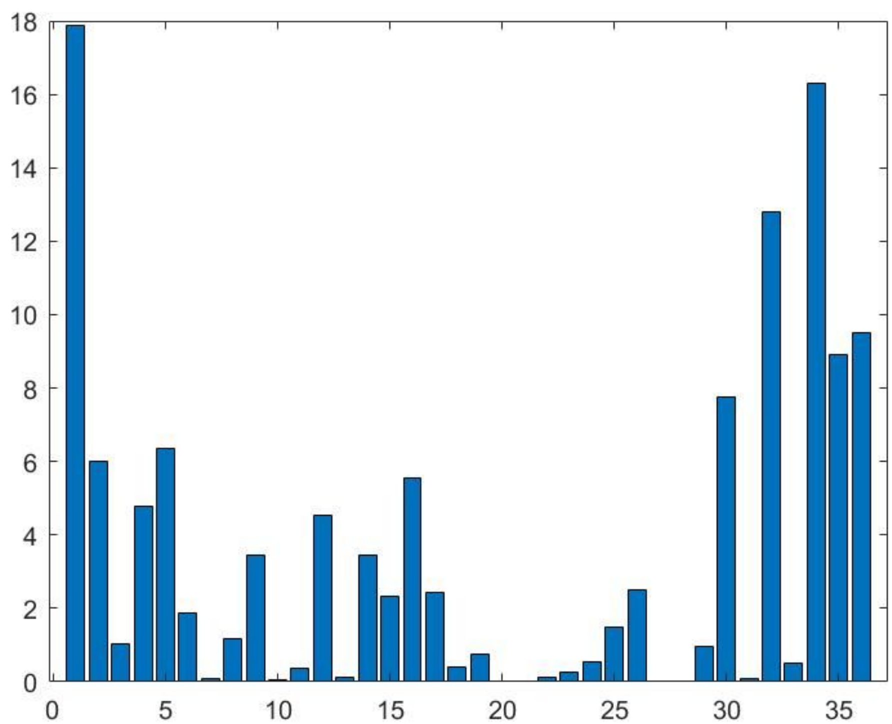


4.

Here, I chose a random point in the image (The 200th index in my points cell array - see question4() function for clearer picture but this data structure holds the sift keypoints) to take a point from my image at a scale. The first image is the image patch, second is the gradient magnitudes of each pixel in that patch, third is the gradient direction of each pixel in that patch (using quiver function in matlab), and finally is the weighted gradient magnitude with a gaussian kernel sigma of 1.5.

5. Histograms


0
5
10
15
20
25
30
35
0
2
4
6
8
10
12
14
16
18

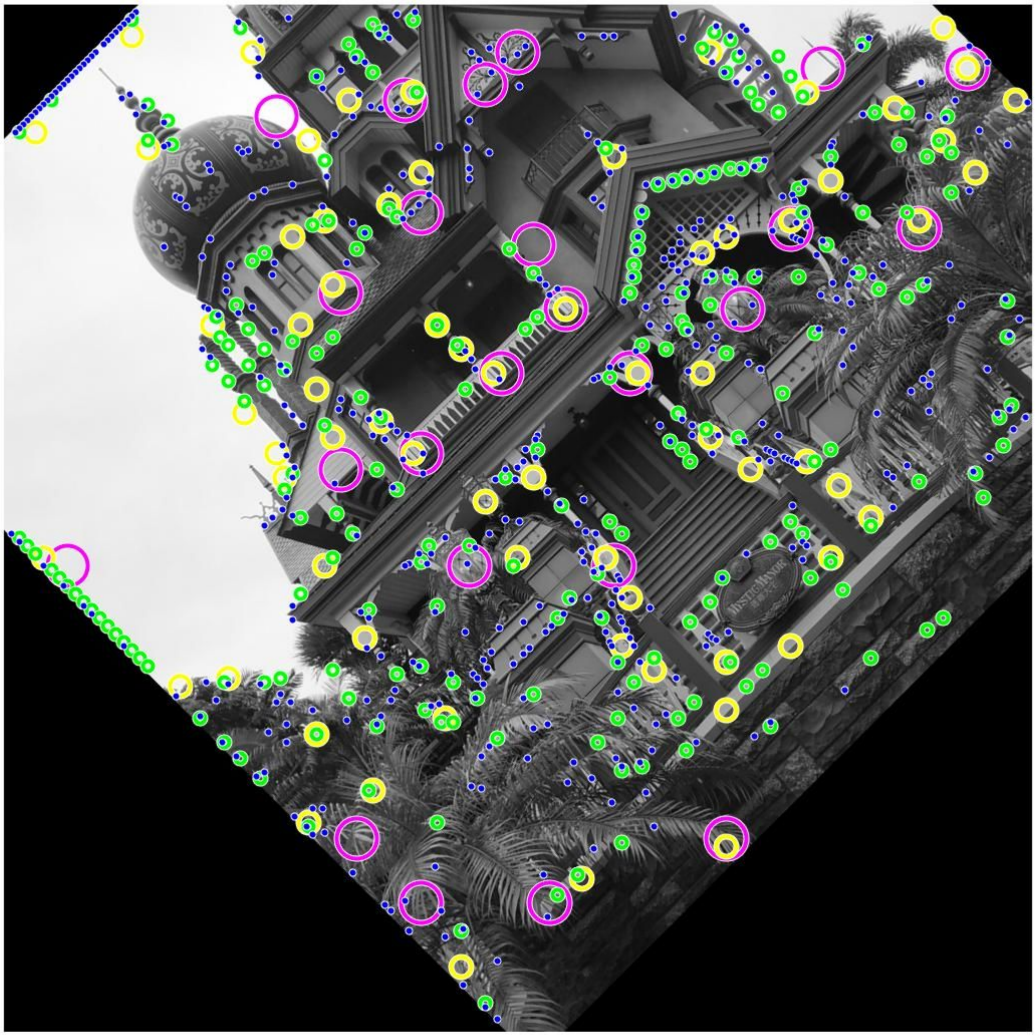

0
5
10
15
20
25
30
35
0
2
4
6
8
10
12
14
16
18

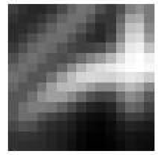
The first histogram is the histogram of weighted magnatudes in each orientation bucket (as specified by the directions). The second histogram is the shifted histogram such that the largest value in the histogram is at the front. Both histograms are again shown on my 200th index of my patches cell array. But these images can be gotten for any keypoint (any index in the points cell array).

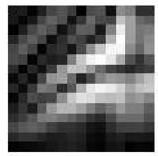
6.

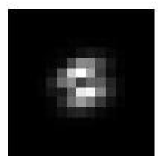
Here is one example of my rotation function call. It rotates the image 45 degrees at the point (300, 400).

Below are some corresponding images that are the same as the earlier questions, but on the rotated image:









The second rotated image, this one is scaled by 0.5, rotated 90 degrees about (200, 200):

7.

As you can see, my matches are not perfect. This can be due to a number of reasons. 1. The number of sift keypoints between the two images is not equal. 2. My filtering of key points could be refined to distinguish outliers. 3. My visualization technique could be improved. My algorithm only picks out points within a given neighborhood, so that could be improved by better choosing a larger/smaller neighborhood, or manipulating the output images to better hide outliers and better define the area being looked at.

Image2 matchings:

8. This matching strategy can be made more robust by keeping track of “objects”. When I say objects, I mean groupings of feature vectors that consistently appear within a certain distance of one another. If feature points were found to be close to one another across a few test trials, it may be able to assume these feature points are connected. From then on, if one of these feature detectors is found, the others can quickly be filled in without heavy computational strain. Some of the new information needed would be the direction and distance in which these points appear in relation to one another. This could also allow SIFT vectors to assume things about obscured objects, if it can see part of the object, based on new data, it can assume the object would continue behind whatever is covering it. This would also allow false matches to be taken out, based on the contextual data.
Also, as seen in my results above, there are many-to-one matchings, meaning one point on one image is being mapped to multiple on another, if these points kept track of whether they have been paired or not, then this would occur less (understanding the points around it and their matchings).