

CS6640 A3

Jake Bergquist

10/17/2018

1) To begin with for the 5 x 5 Fourier texture feature generation I had to pad the image by 2 on each side so that the window could center on the edge pixels appropriately. The Image used (**Figure 1**) was from my A1 video where a car can be seen driving through the street. It was converted to **Figure 1.1**, a grayscale, for all subsequent use in this assignment. For each pixel the 5 x 5 window Fourier transform was calculated and the power spectrum was generated (by multiplying each Fourier output by its conjugate). The resulting texture vector was saved for every picture. K means was then used to group these texture vectors with different numbers of groups. One thing I learned is that kmeans will group differently every run. By that I mean, the groups will be the same but their group index will change depending on how kmeans converges on the groups. So between runs, a section of grass will always be grouped in the same group (as in with itself) but the group index might change. With that in mind the figures here reference the groups shown. If one were to run my figure generator script the group indices may change, but the group shapes should be the same. Also certain sections of the figure generator would need to be changed to account for this. To run the figure generator properly you must run it in the declared sections rather than all at once and adjust as necessary.





Figure 2.1 shows the color mapping for each group, up to 7 groups, for the K means. I found that generally (not in every case) when using more than 7 groups the K means did not converge under standard parameters and generally using more than 7 groups was not advantageous as far as the result was concerned. **Figure 2** shows the result of grouping with 5 groups. As can be seen groups one and two are the primary textures in the image. Additionally as usual the sky is easily grouped into one texture. It would appear that the bodies of the parked cars and the moving car are mostly comprised of textures from groups 2,3,5 while the windows show as group 1,4 texture. Of interest, one of the textures is only seen on the cars and the sky. This could result in a rudimentary identification of cars (and the sky) which could be useful as a first pass to tell other filters where to look for them. In **Figure 2.2** we see that group highlighted alone.

Figure 2: 5 groups, FFT texture

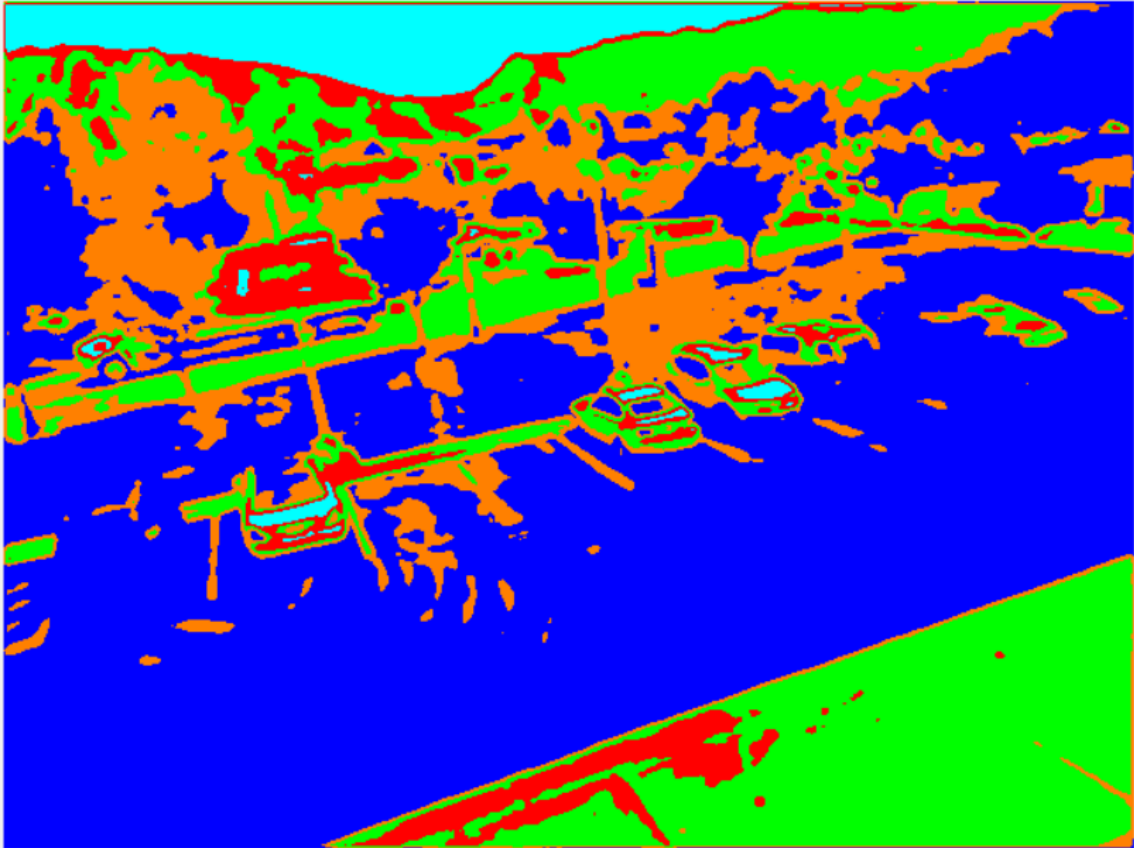
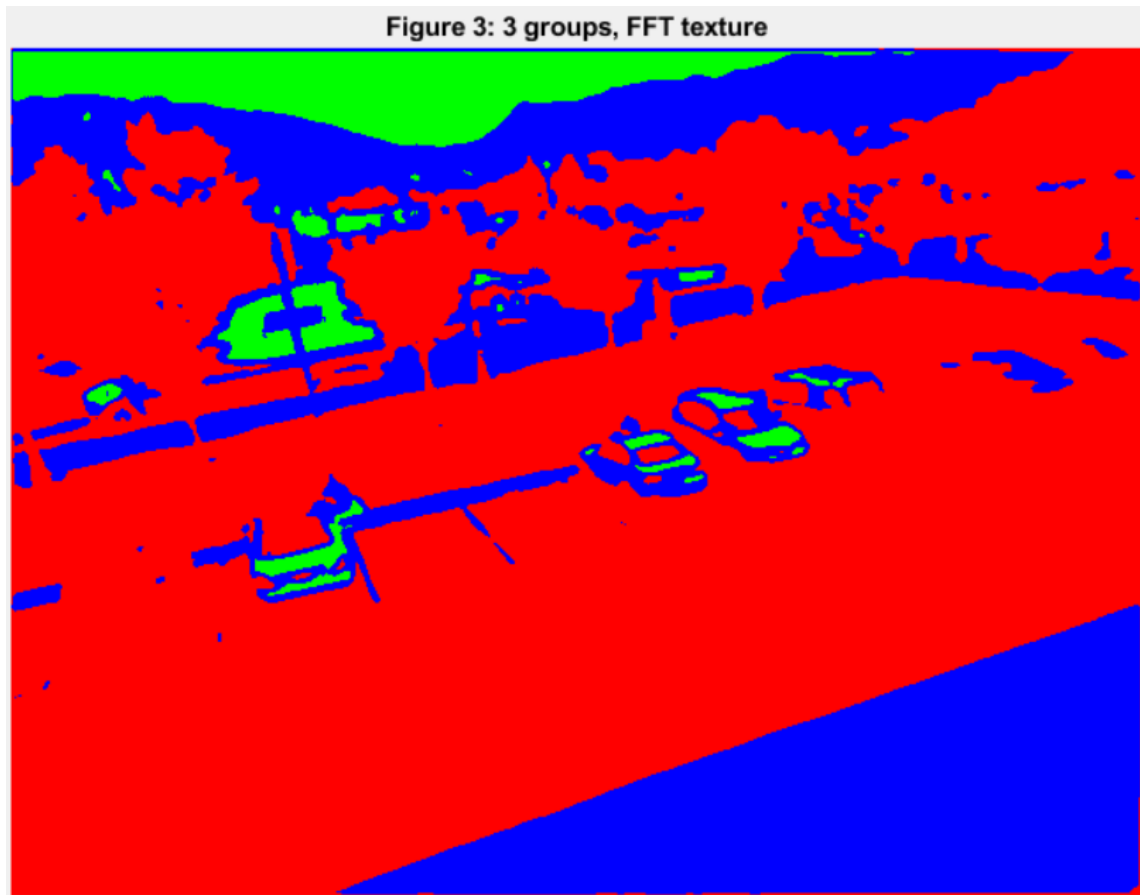


Figure 2.1, Group Coloring

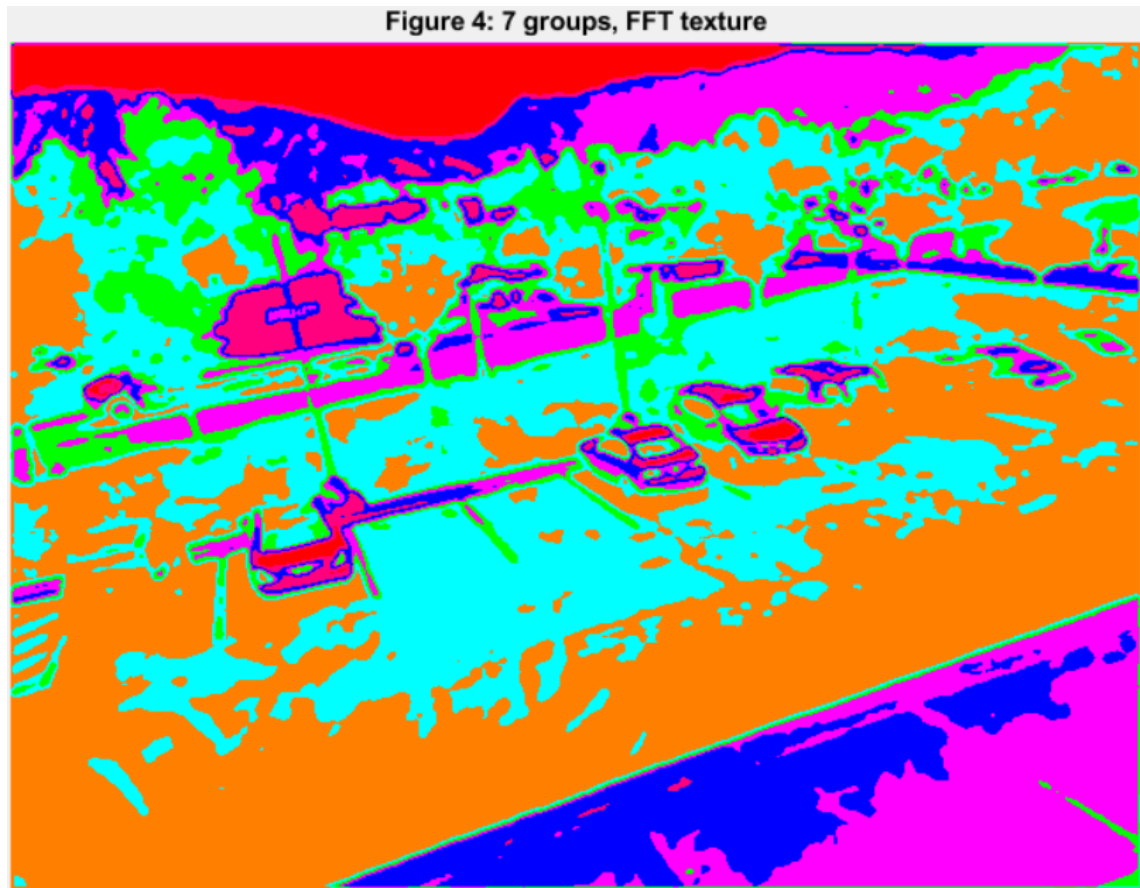




In **Figure 3** we see the use of kmeans with only 3 groups. This could be useful for separating out large areas into semantic sections. For instance group 1 is mostly sidewalk and mountains, group 2 is mostly asphalt/grass/trees (interesting as all of these have a rough texture semantically) and group 3 shows up mostly on shiny smooth textures like parts of cars and the sky.



In the opposite direction, **Figure 4** shows using 7 groups. In this grouping we could see that a rules based approach might be useful. For example, cars are things of a certain size that contain mostly textures 1,3,4,5,6 in close proximity. Overall this is approaching a high detail representation that becomes more difficult to interpret, but allowing for more flexible analysis/complicated combinations of texture groupings. For example saying certain texture interfaces represent car/road boundaries.



2) Now we move on to using the radial power spectrum feature using 10 rings. To do so a 19x19 window is needed, thus the image had to be padded by 18 on all sides. **Figure 5** shows grouping with 5 groups. We see that this method results in a more general segmenting, less detailed. Overall it seems that this method has a hard time distinguishing dark colored cars from dark colored background, seen in the foreground parking lot as the black cars are the same texture group as the black asphalt. Upping the group number in **Figure 6** we can still see lighter cars are better highlighted than darker ones. To illustrate **Figure 7** shows two of the groups from **Figure 6**. We can see that the light colored vehicles are outlined decently but the dark cars are not.

Figure 5: 5 groups, radial FFT

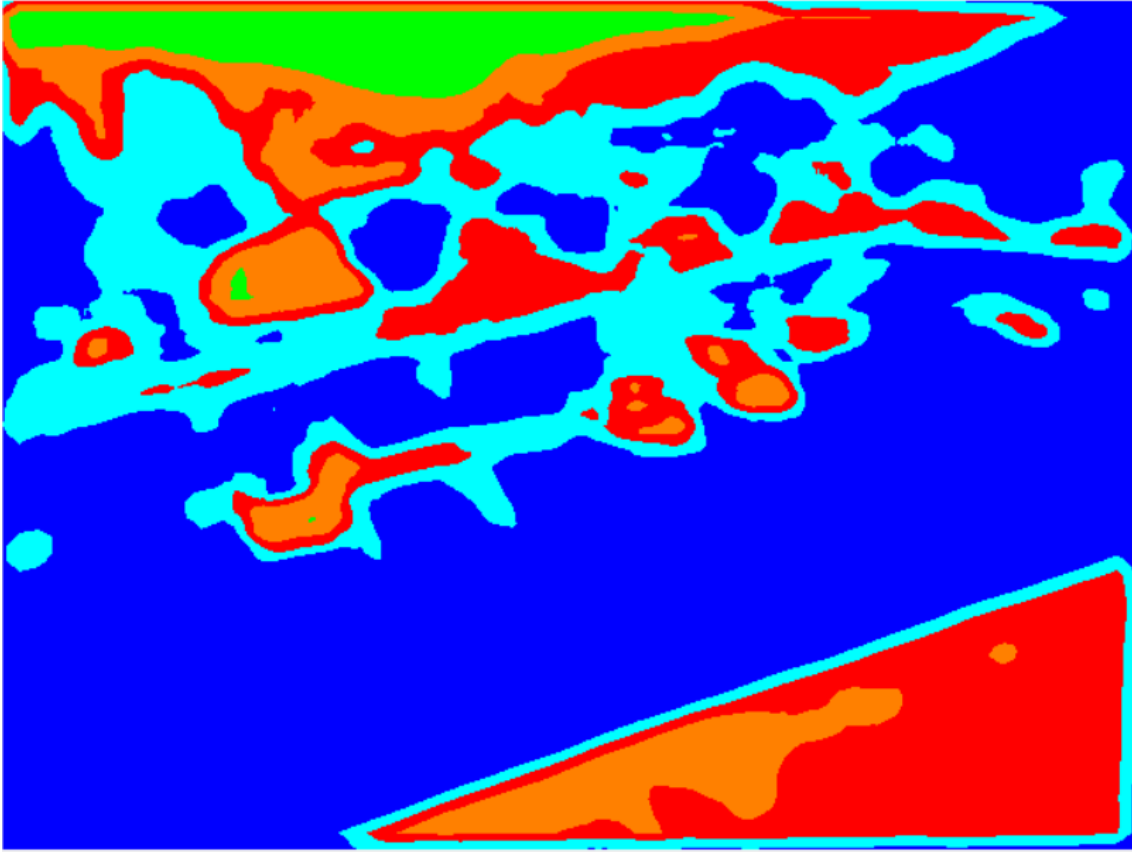
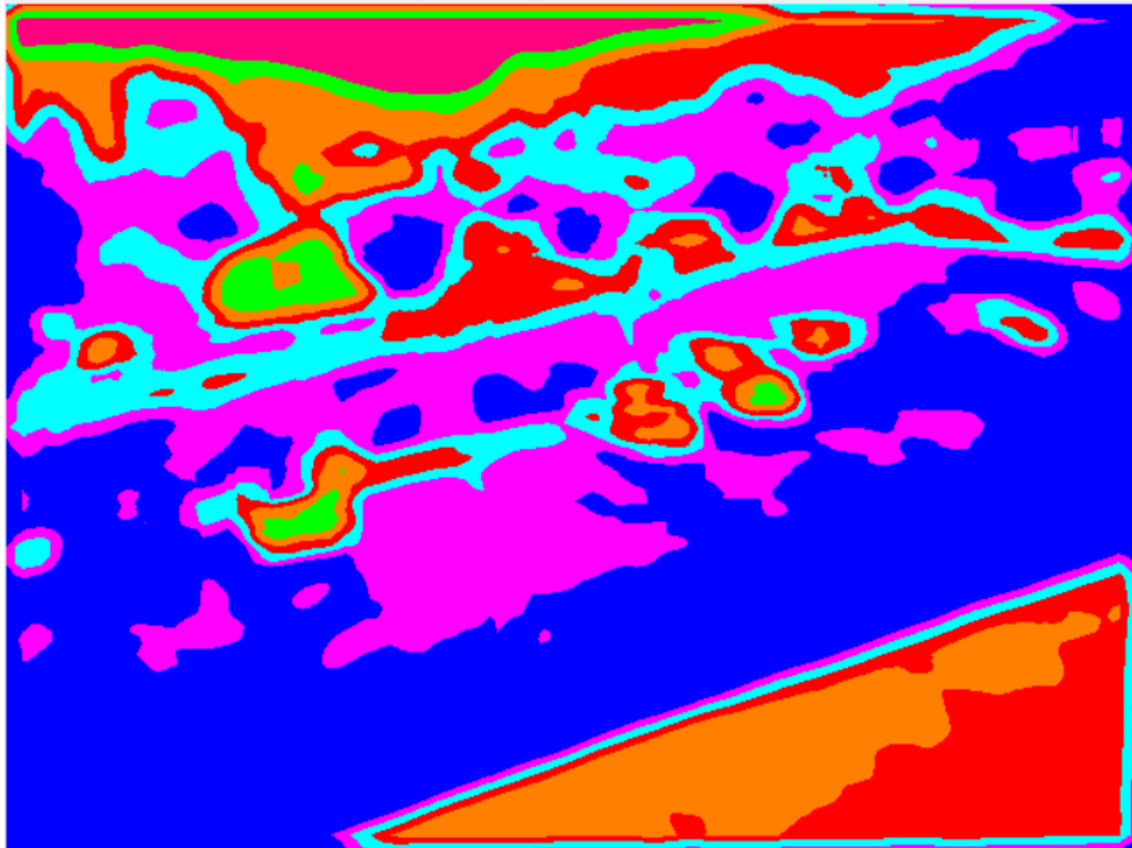
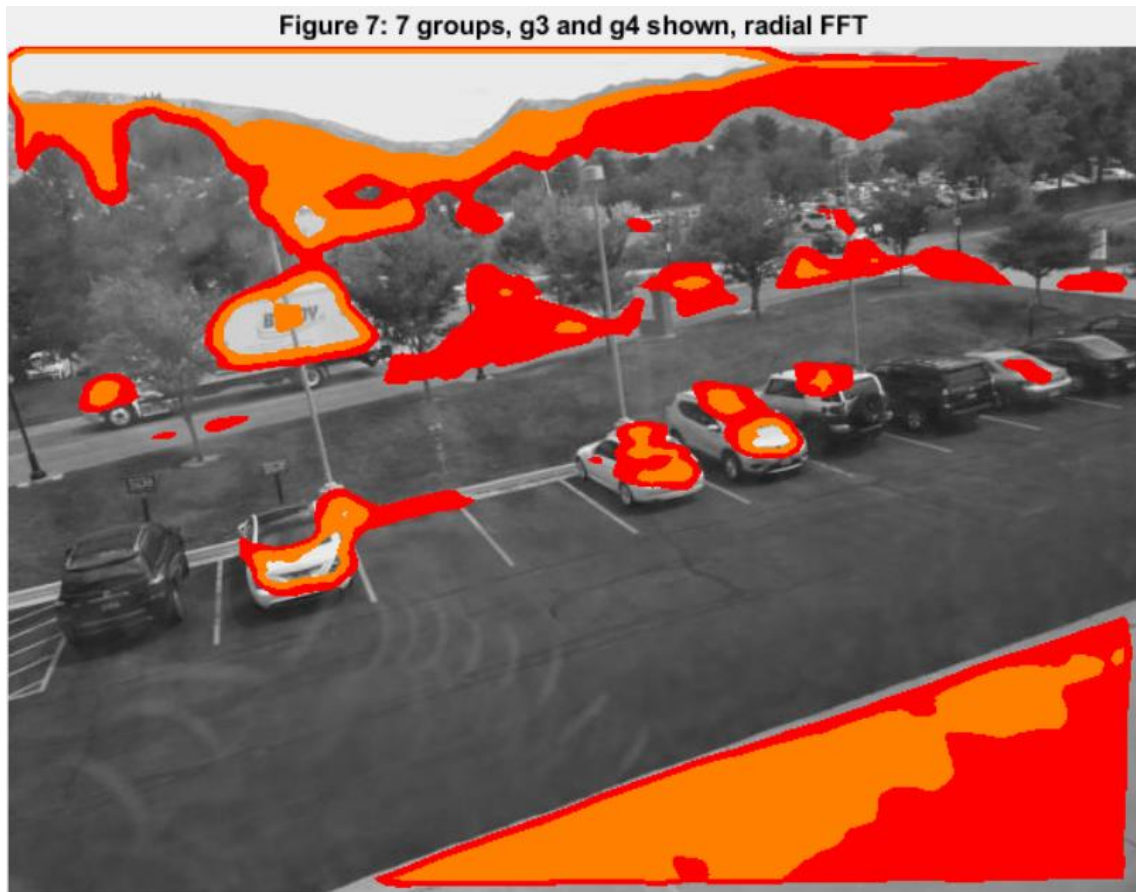


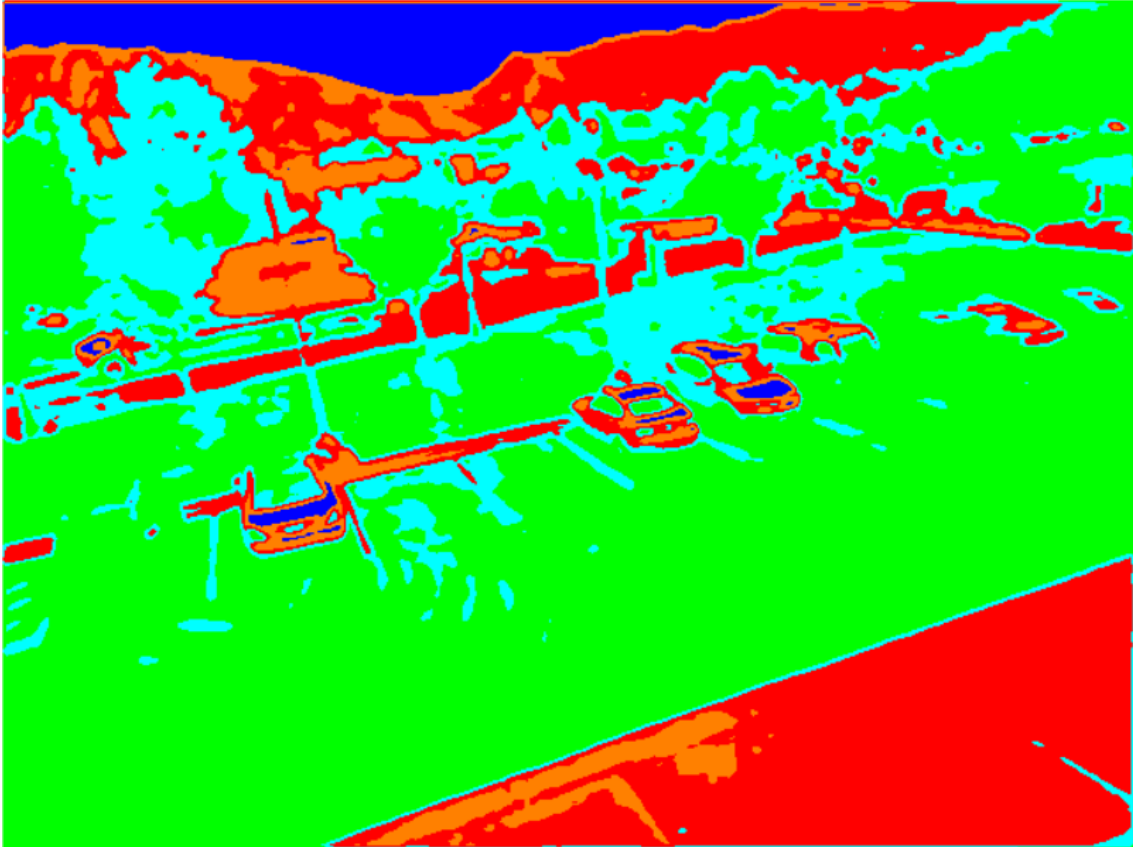
Figure 6: 7 groups, radial FFT

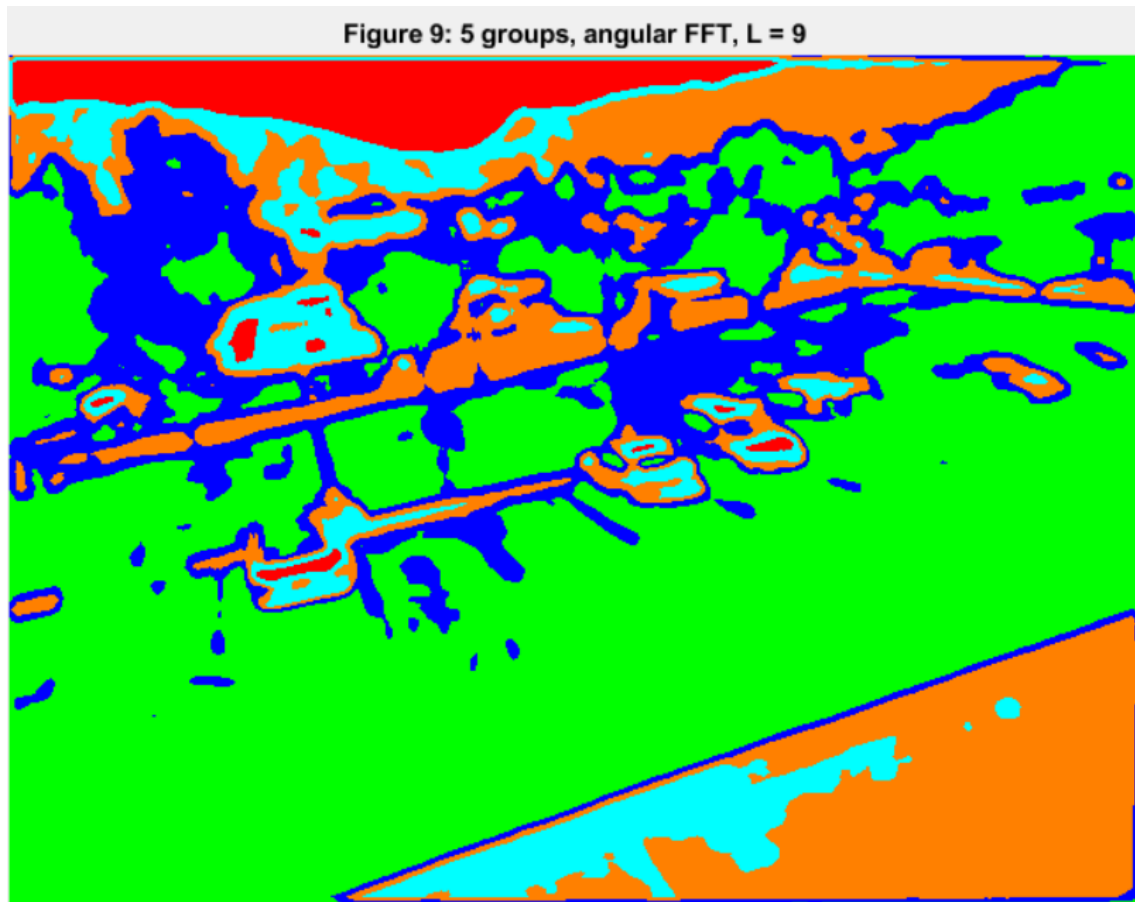




3) For the angular segmenting of the power spectrum I first used a 5×5 window, therefore padding by 2 on all sides. This resulted in **Figure 8** with 5 groups which looks very similar to the standard texture filtering with 5 groups (**Figure 2**). This could be due to the fact that the 5×5 power spectrums are the same so segmenting one vs using each individually will result in a similar grouping. However upping the size of the box to a 9×9 (**Figure 9**) we get a more smoothed result. In either case we can see that the textures of the cars in **Figure 9** are very different from the surrounding textures for the light cars in the parking lot. Again there are issues separating the dark cars with any of the FFT texture filters as their FFT texture seems to be too similar to the surrounding black asphalt.

Figure 8: 5 groups, angular FFT, $L = 5$





4) The FFT curve definer produced a FFT power spectrum of length $p/2-1$ for each shape (an O and an H). These vectors could be used to characterize these characters and distinguish them. **Figure 10** shows the values of the two vectors components with a walking length of 1, that is pixel by pixel. We can see that the profiles of each are very different. When we use a larger step size (of 2) we see that the profiles for each are different than before but still distinct from each other (**Figure 11**). Using a larger step size would be advantageous for smoothing out edge irregularities from noise resulting in a smoother walk over the curves of the object. It is interesting to note that with the bigger step sizes the scales on the y axis become the same, thus there is a disadvantage in that the two shapes start to look too similar as the step size increases. This can clearly be seen when the step size is increased to 5 (**Figure 12**) and the two vectors look almost the same. Thus in order to get the best results a minimal step must be used while still minimizing the effects of uneven or erroneously jagged edges.

