CSC 449 – Homework 3
Phyo Thiha

Introduction

In this assignment, we implemented k-means clustering algorithm and experimented with different 'k' values to find segments in a color image. In addition, we used different color channels—RGB, HSV and a variant of RGB called 'LST'—as feature vectors to observe which one performs better. We present our findings in this report in the following order: list of files we used/implemented in this assignment under 'Relevant Files'; explanation of our implementation in 'Methods' section; presentation of our results and discussion of our findings in 'Results and Discussion' section and finally, a list of resources we used for this assignment.

Relevant Files

PeppersRGB.tif	- image used for color segmentation
run_kmeans.m	- script that runs k-means assignment as a whole
create_feature_vectors.m	- function to build features from the original image
kmeans.m	- implementation of k-means clustering algorithm

Methods

We relied heavily on the algorithm outline provided by Professor Jiebo and [1,2] in implementing our k-means module in Matlab. Our implementation consists of three main parts—the script that runs the k-means, the function that builds feature vectors from the original image, and the implementation of the k-means algorithm.

The script that runs the k-means loads the image provided ("PeppersRGB.tif"), and calls Matlab's *im2double* to transform RGB values to double (floating point) format. It then prepares feature vectors and reshapes the image matrix into "n x 1" dimension vector so that we can use it as an input for the k-means clustering. Finally, the script displays the result from clustering using *label2rgb*.

We experimented with three different types of feature vectors: RGB, LST and HSV. The RGB simply treats the red, green and blue channels of the original image as separate features. The LST is a variant of RGB defined as:

$$L = (R + G + B)/3$$
$$S = (R - B)/2$$
$$T = (2G - R - B)/4$$

The HSV—also known as HSB for hue, saturation and brightness—feature vector is built from treating hue, saturation and value (brightness) as separate features. We used Matlab's *rgb2hsv* function to convert original image's RGB values into HSV. After performing necessary conversions, we have the feature vector, *f*, as

$$f = \begin{bmatrix} f_{1i} & f_{1j} & f_{1k} \\ f_{2i} & f_{2j} & f_{2k} \\ f_{3i} & f_{3j} & f_{3k} \\ f_{4i} & f_{4j} & f_{4k} \\ \ldots & & \\ \ldots & & \\ f_{ni} & f_{nj} & f_{nk} \end{bmatrix}$$

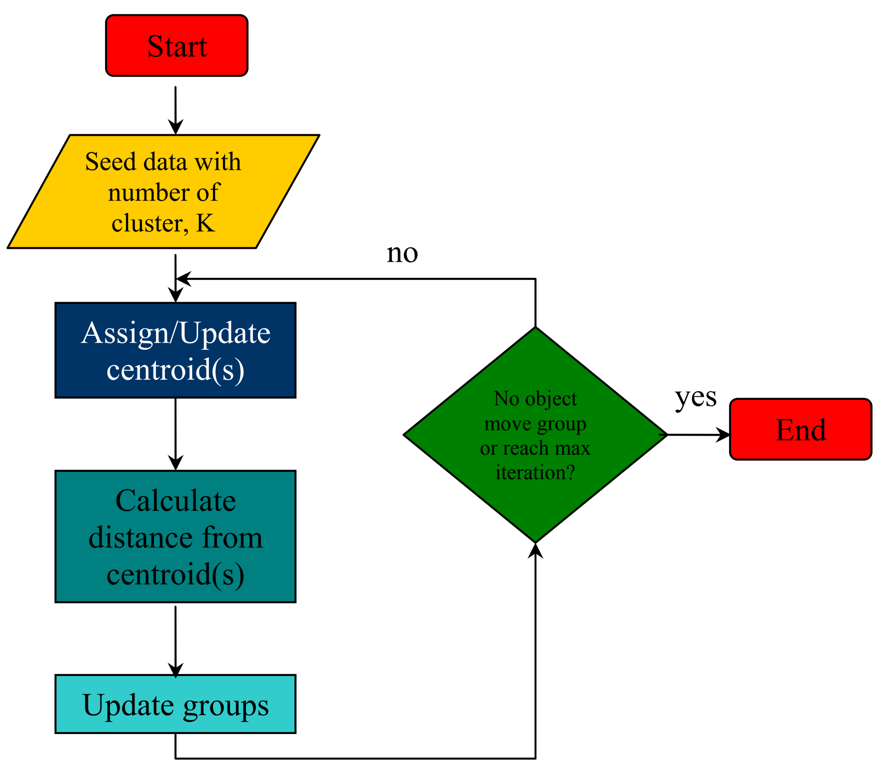
where ‘n’ is the total number of pixels in the original image, and ‘i’, ‘j’ and ‘k’ represents each vector component from RGB, LST or HSV, depending on which feature set we were testing.

The k-mean algorithm can be (crudely) summarized in three steps as below.

Iterate until no centroid moves or when max_iteration_limit is reached:

Step 1. Determine/assign centroids
Step 2. Calculate the distance of each object from the centroids
Step 3. Group the objects based on the group of centroid they are closest

Alternatively, we can represent k-means algorithm in a simple flow chart as follows [1].



Results and Discussion

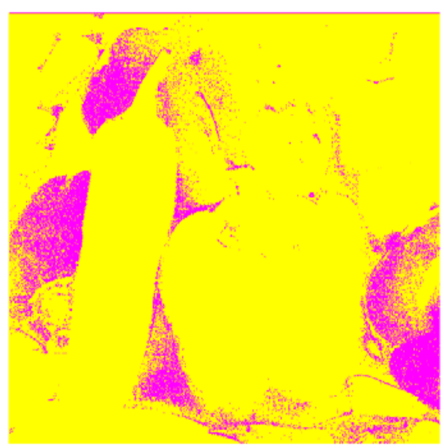
Although our k-means implementation set the default value for ‘k’ to be 1, we never experimented with it because the original image has more than one class of object that we would like to segment. We also noticed that the image has two primary colors—red and green—with some tint of white light from flash reflecting off from the surfaces of the vegetables. Therefore, we decided to try with ‘k’ of 2 and 3, respectively. From this observation, we would suggest that the best ‘k’ for a color image might depend primarily on how many objects of different color exist in the image.

Our implementation of k-means assigns the initial centroids by picking ‘k’ random objects from the input. The results from ‘k=2’ for different color space, along with the original image, are shown below.

a)

b)

c)



d)

Figure 1: Comparison of k=2 clustering. a) Original image b) RGB c) LST and d) HSV

As can be seen in figure 1, the RGB and LST features outperformed HSV in correctly segmenting the objects of two primary colors. In fact, HSV performance was so poor that it cannot even distinguish the red pepper from the green one in the center of the image. We believe that this poor performance of HSV could partly be explained by the fact that the saturation and the brightness features might be similar for the red and green peppers despite their stark contrast along hue dimension[3]. We observed that the results from RGB and LST features are almost the same; this could be explained by the fact that LST features are, in fact, linear combinations of RGB features. In addition, we noticed that the RGB and LST color space successfully separated the dark and bright regions on the peppers.

Figure 2 shows the result from 'k=3' alongside the original image.

a)

b)

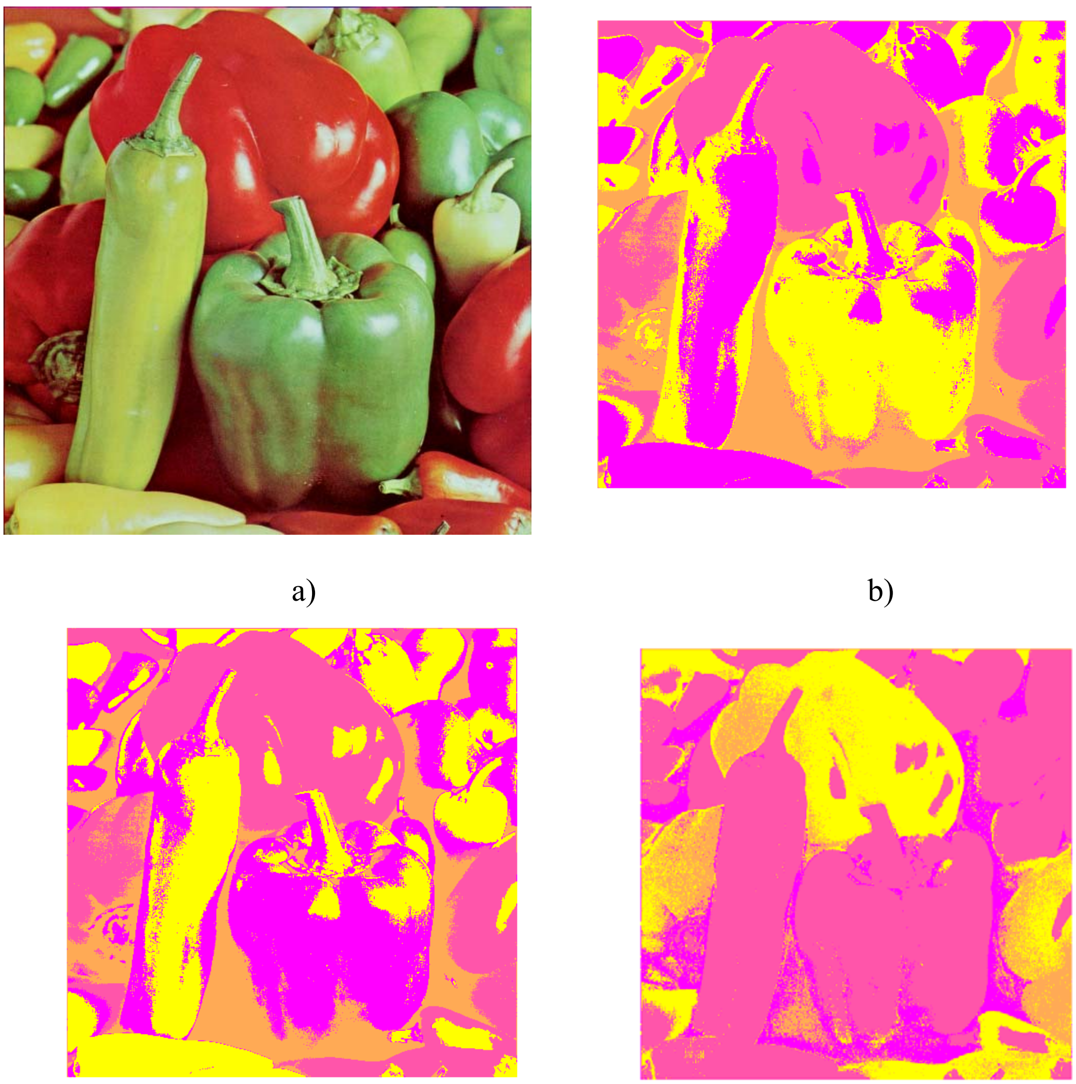
c)

d)

Figure 2: Comparison of k=3 clustering. a) Original image b) RGB c) LST and d) HSV

As seen in figure 2, setting the value of 'k' to 3 introduced undesirable noise in clustering outcome of the RGB color space. It has brought forth the incorrect clustering on the same pepper. The performance was also degraded in the case of LST—for example, it identified the red peppers on the left and right of the image as different from the one in the center. It also has less distinctive boundaries between peppers of different colors. Compared to the RGB case, it fared well probably because its features are weighted transformations of the original RGB channels. On the other hand, the performance of HSV features improved dramatically from that of 'k=2'. The reason could be that HSV channels carry independent information from each other; therefore, when we allowed 'k' to be '3', it was able to utilize those three features fully to correctly segment even the bright and dark regions on the same pepper.

Finally, we present the results from our experiment with 'k=4'.



a) b) c) d)

Figure 3: Comparison of k=4 clustering. a) Original image b) RGB c) LST and d) HSV

Here, we notice that the results from RGB and LST features are similar. Although these segmentation results may not be very helpful if we were to count the number of peppers in the image, they are good enough to capture both the bright and the dark regions in the image, thereby making the segmented images appear three dimensional. On the other hand, the result from HSV features looks less clear than it was when 'k' was set to '3'. In addition, it seems to be have grouped the green and white color into the same cluster group while finding two separate cluster for red (as can be seen on the pepper in the center). Therefore, we determined that in the case of 'k=4', RGB and LST outperformed HSV.

From this assignment, we learned that it is not trivial to cleanly segment a given color image. The segmentation results vary largely based on the type of features we use as well as the original image's complexity (in terms of the number of objects, colors and shades that are present in it). In addition, the correctness of the segmentation result could very well be subjective. For example, in the case of the pepper image above, if we are only interested in segmenting the main objects, we should be satisfied with the results from clustering with 'k=2' and RGB features. But if we are also interested in segmenting the reflecting spots (caused by the camera flash) on the peppers, we may have to use 'k=3' or 'k=4' with HSV features.

Appendix

[1] Teknomo, Kardi. "K Mean Clustering Tutorial". July 2007. 29 February 2012. URL: http://www.croce.ggf.br/dados/K%20mean%20Clustering1.pdf

[2] Moore, Andrew. "K-means and Hierarchical Clustering". 16 November 2001. 29 February 2012. URL: http://www.autonlab.org/tutorials/kmeans.html

[3] "HSL and HSV". Wikipedia®. 29 February 2012. 29 February 2012. URL: http://en.wikipedia.org/wiki/HSL_and_HSV