Image Analysis - TP4 - K Means

Jander Nascimento, Raquel Oliveira

April 12, 2011

1 Initial concept

The *k-means* split the image in k sets based in some parameters. Those parameters are heuristics to split the set. One common way to split the sets is in a 3d space, in which the domain of the space is [0,255] and each axle is the color space RGB.

The algorithm is composed of the following steps:

1: Given a number of regions k, randomly generate the initial values of the means $m_1, m_2, ..., m_k$, which are called *centroids*.

2: Assign each pixel to the group that has the closest centroid.

3: When all pixels have been assigned, recalculate the m_k values of the k centroids.

4: Repeat Steps 2 and 3 until the centroids no longer move. This produces a separation of the pixels into groups.

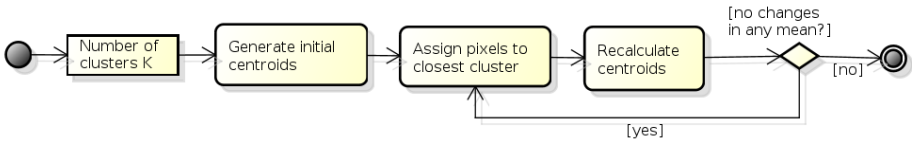


Figure 1: K-means steps

2 Initial *k* influences

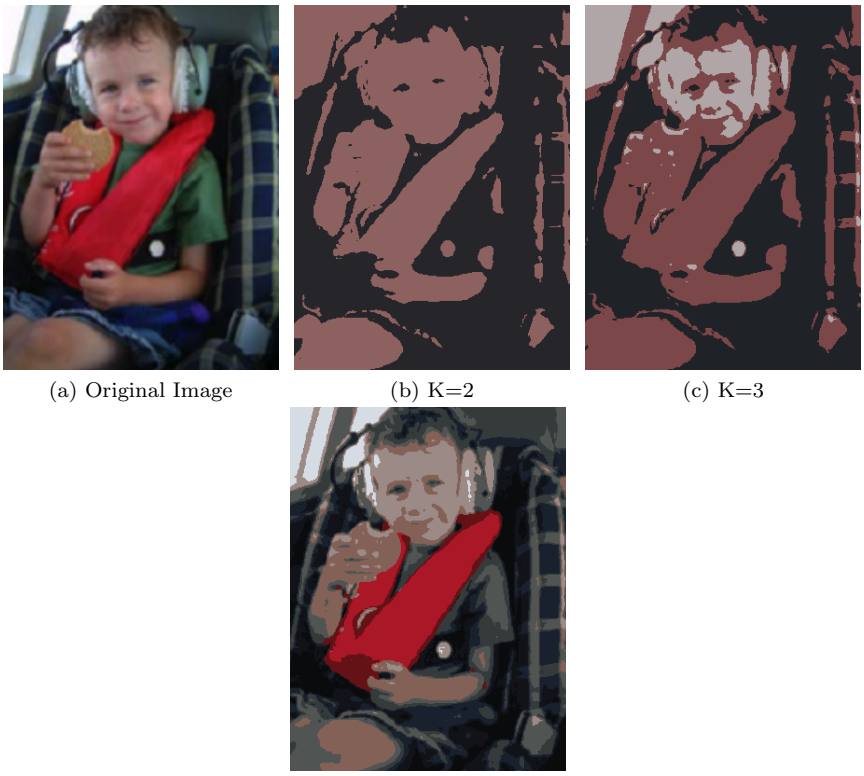
The results produced depend on the initial values for the means (centroids). The simplest way is to select these values randomly, although the better choice is to place them far away as much as possible from each other.

3 Number of regions

K represents the number of clusters that is going to be used in the segmentation process. This information is fundamental to distinguish the number of regions in the image, so as bigger k is, more cluster are going to be used, therefore more detailed the image will be.

In the current implementation we used the color intensity as the orthogonal space, with the axis(x,y,z) been represented by the color red, green and blue of the pixel.

The difference can be seen in the Figure 2. The 2b use only two distinct groups, so the image presents two segments, which does not give too many details about the image. In the 2c with three groups, is possible to see more details than in the previous image. In 2d we can see that the details in the image increases dramatically.



(a) Original Image (b) K=2 (c) K=3

(d) K=10

Figure 2: K-Means

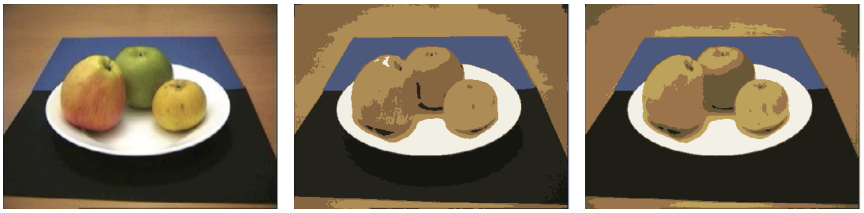
4 Other influences

4.1 Including distance between pixels

Using the distance as a factor to determine the groups allows not only create groups based in the color space distance, but also based in the distance between the group and the pixel.

In the Figure 3 it was used the same number of clusters (k) for the tested image.

It is clear that using the distance as an information, the edges of the objects are better defined. If we do not consider the distance some objects are simply merged based in how close the colors are from each other, which is not always the case. As shown in the Figure 3b the objects were almost completely merged because of the proximity to the green color.



(a) Original Image (b) Not considering distance (c) Considering distance

Figure 3: Using distance between pixels as an information

This result can be achieved simply by adding the Euclidean distance for the distance between the pixels. The Equation 1 should be applied to the intensities red, green and blue and the Euclidean distance between the pixel position.

$$\arg\min \sum_{i=1}^{K} \sum_{x_j \in S_i} \|x_j - \mu_i\| \tag{1}$$

4.2 Balancing distance and colors influence

Since the influence of the distance, as the color, are not the same for every image, it is possible to increase or decrease its influence by using a contribution factor, so this adjustment can be done independently.

So, it's possible to improve the Equation 1 by adding the weight, see Equation 2, where w is the weight. The weight is going to be given for each possible parameter used to calculate the distance, the color intensities(red, green and blue) and the location of the pixel (x and y position).

$$\arg\min \sum_{i=1}^{K} \sum_{x_j \in S_i} w * \|x_j - \mu_i\| \tag{2}$$

4.3 How to run?

Steps to compile the application:

- svn checkout https://jfimageanalysis.googlecode.com/svn/trunk/TP4/ #download source code
- make #compiles the code

As an input image only **ppm plaintext/ansii** files are accepted (P3).
Examples of usage:

- Create 3 clusters:

  ./showregion -i image_in.ppm -g 3 > image_out.ppm

You can always type *./imagetransform –help* to check for more options.