

Image clustering

Practice work #1 - Arnaud QUIRIN - May 2008

The goal of this practice is to implement the K-Means algorithm (try the first one, as it looks like the easier to implement, and it is well suited for this practice, even if it is recommended to try another one).

K-Means groups similar objects. In this practice, we will consider that an object and a pixel are the same concept. A pixel is an object with 3 attributes, the red, the green and the blue. The goal is to transform an image into another image so that the pixels belonging to the same cluster have to be in the same color. This result is called categorization.

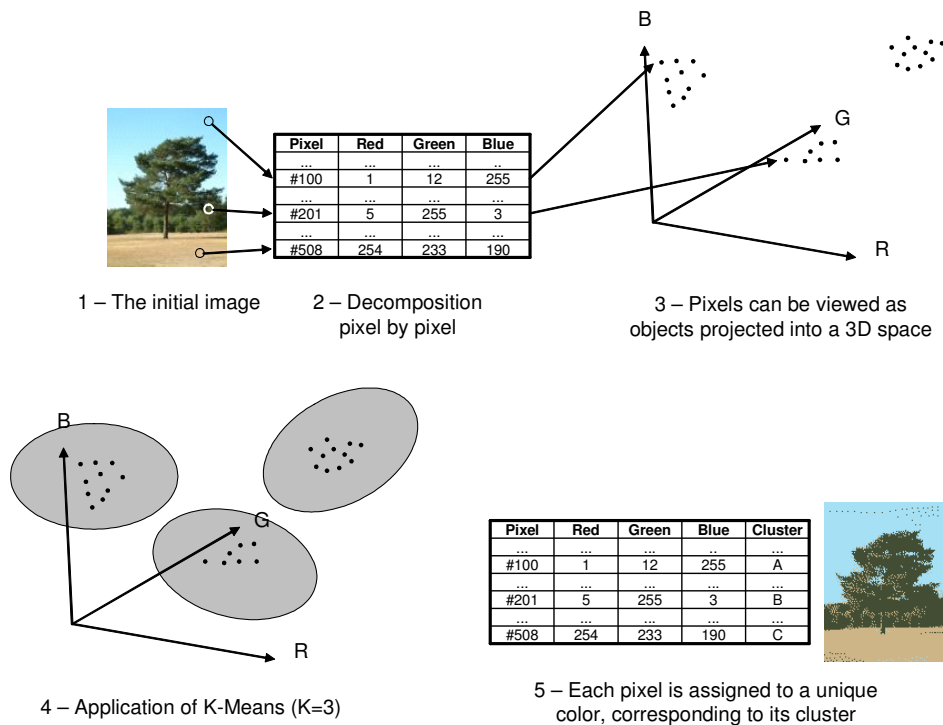


FIG. 1 –

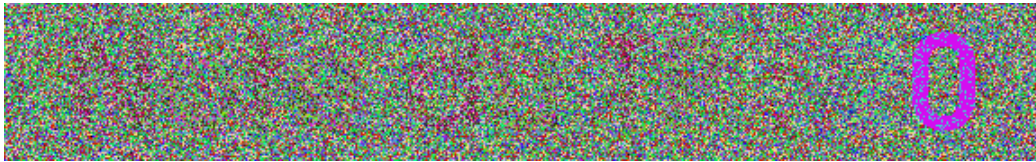
Look carefully the picture in the figure 1. A color version is available here : <http://aquirin.ovh.org/datamining/>

On this site, there is also sources to read and to save an image in C or in Java, and some other images. ImagePPM is the library you have to use to manipulate images, and TestImagePPM is an example of source using the main functions. To view the obtained images, you can use in Linux the tools xvview or eog (Eye-Of-Gnome).

To produce a categorization using K-Means, you have to perform the 5 following steps. Only the step 4 is not implemented yet.

1. First, you have to load an image, using the `PPMload` function (in C) or the constructor (in Java).
2. The pixels are loaded into the object `image`.
3. If you need the individual values of the pixels, you can use the functions `getRed`, `getGreen` and `getBlue`. The size of the image can be obtained using `getWidth` and `getHeight`.
4. The categorization algorithm has to be implemented by yourself.
5. Once you have found the cluster of a pixel, you can assign it a dedicated color (using `setRGB()`), then save the image using `write()`.

Good work !



text.ppm : a hidden text



esplanade.ppm : a satellite image



tree.ppm

FIG. 2 –