HW-OpenMP-k-Means

In this project, we are going to apply the k-means algorithms for the image segmentation. The k-means algorithms can be described as following:

Step 1: **Initialization**. Read an image from a JPG file. Choose k pixels from the image. (You can label all the pixels from 1 to N, and choose one pixel every N/k pixel). The colors of the k pixels are called generators. So now we have k generators, labeled from 0 to k-1. (Remember that a generator is a color, not a pixel.)

Step 2: **Grouping the pixels**. Thinking each color as a vector composed by (R, G, B). Then we can define the distance between every two colors. For each pixel, find the generator which has the shortest color distance to the pixel's color. If the index of the generator is i, then put the pixel to group i. After all the pixels are grouped, we get k groups of pixels.

Step 3: **Get new generators**. For each group of the pixels, find the average color of the group. (Average on R, G, B.) And set the average color as the new generator. And finally, we get k new generators.

Step 4: **Jump back to Step 2**. Step 2 and Step 3 form an iteration. Repeat the iteration for a predefined number, for example, 20 times. And then go to Step 5.

Step 5: **Finishing**. For each pixel, replace its color by its group's generator. And write the new image into a JPG file.

Write a code for the k-means algorithm. And parallelize it by the OpenMP method. Optimize your OpenMP method and try to get a good scaling with different number of threads and CPU cores.

The image and the value k and the iteration times are all of your choice. You can test different images and different values and give discussions (optional).

Upload your report and source code to Canvas.