Code Report for K-Means Image Compression

Running results:

```
USAGE: KMEANS <INPUT-IMAGE> <K> <OUTPUT-IMAGE>
NOW RUNNING UNDER DEFAULT SETTING...
COMPRESSION RATIOS FOR KOALA.JPG ARE:
K=2 => 0.031252543131510414
K=5 => 0.09375635782877605
K=10 => 0.1250127156575521
K=15 => 0.12501907348632812
K=20 => 0.15627543131510416
COMPRESSION RATIOS FOR PENGUINS.JPG ARE:
K=2 => 0.031252543131510414
K=5 => 0.09375635782877605
K=10 => 0.1250127156575521
K=15 => 0.12501907348632812
K=20 => 0.15627543131510416
```

Compression Ratios:

The image segmentation problem discussed above also provides an illustration of the use of clustering for data compression. Suppose the original image has N pixels comprising {alpha, R, G, B} values each of which is stored with 8 bits of precision. Then to transmit the whole image directly would cost 32N bits. Now suppose we first run K-means on the image data, and then instead of transmitting the original pixel intensity vectors we transmit the identity of the nearest vector μ_k. Because there are K such vectors, this requires $\log_2 K$ bits per pixel. We must also transmit the K code book vectors μ_k, which requires 32K bits, and so the total number of bits required to transmit the image is $32K + N \log_2 K$ (rounding up to the nearest integer).

Q: Is there a tradeoff between image quality and degree of compression. What would be a good value of K for each of the two images?

A: Yes, the more you compress the worse the image quality will be. For the example, it seems that K=10 would be a good choice.