

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 => 0.25000254313151044
```

```
K=5 => 0.25000635782877606
```

```
K=10 => 0.25001271565755206
```

```
K=15 => 0.2500190734863281
```

```
K=20 => 0.2500254313151042
```

```
COMPRESSION RATIOS FOR PENGUINS.JPG ARE:
```

```
K=2 => 0.25000254313151044
```

```
K=5 => 0.25000635782877606
```

```
K=10 => 0.25001271565755206
```

```
K=15 => 0.2500190734863281
```

```
K=20 => 0.2500254313151042
```

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 $\{R, G, B\}$ values each of which is stored with 8 bits of precision. Then to transmit the whole image directly would cost $24N$ 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 $24K$ bits, and so the total number of

bits required to transmit the image is $24K + N \log_2 K$ (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.