Project 1: Implement a Simple Content-Based Image Retrieval System
Due Date: February 26, 2003 (Wednesday)

Requirements:

This project is to implement a simple Content-Based Image Retrieval system based on two different color histogram comparison methods (Intensity method and Color-code method) by using Java. You will be asked to demonstrate your programs and to submit well-commented codes and report via soft (floppy) and hard copies.

1. Test Image Database
   This test image database include 100 true-color images in .jpg format. You can download them from http://www.cs.fiu.edu/~chens/courses/cis6931/2003/Images/images.zip

2. Histogram Comparison
   You need to implement the distance metrics for histogram comparison. Let $H_i(j)$ denote the histogram value for the $i^{th}$ image, where $j$ is one of the $G$ possible grey levels. Then the difference between the $i^{th}$ image and $k^{th}$ image can be given by the following distance metric:

   ♦ Manhattan Distance
   $$D_{i,k} = \sum_{j=1}^{G} \left| \frac{H_i(j)}{M_i*N_i} - \frac{H_k(j)}{M_k*N_k} \right|$$

   where $M_i*N_i$ is the number of pixels in image $i$, and $M_k*N_k$ is the number of pixels in image $k$. $M$ is the number of rows and $N$ is the number of columns.

3. Color Histogram
   Color histogram comparison is a simple but effective approach in CBIR systems. There are two ways to combine the information from 3 color channels (R, G, B):

   A. Intensity Method
   $$I = 0.299R + 0.587G + 0.114B$$
   By this way, the 24-bit of RGB (8 bits for each color channel) color intensities can be transformed into a single 8-bit value. The histogram bins selected for this case are listed below:

   $H_1$: $I \in [0,10)$;  $H_2$: $I \in [10,20)$;  $H_3$: $I \in [20,30)$;
   $H_4$: $I \in [30,40)$;  $H_5$: $I \in [40,50)$;  $H_6$: $I \in [50,60)$;
   $H_7$: $I \in [60,70)$;  $H_8$: $I \in [70,80)$;  $H_9$: $I \in [80,90)$;
   $H_{10}$: $I \in [90,100)$;  $H_{11}$: $I \in [100,110)$;  $H_{12}$: $I \in [110,120)$;
   $H_{13}$: $I \in [120,130)$;  $H_{14}$: $I \in [130,140)$;  $H_{15}$: $I \in [140,150)$;
   $H_{16}$: $I \in [150,160)$;  $H_{17}$: $I \in [160,170)$;  $H_{18}$: $I \in [170,180)$;
   $H_{19}$: $I \in [180,190)$;  $H_{20}$: $I \in [190,200)$;  $H_{21}$: $I \in [200,210)$;
   $H_{22}$: $I \in [210,220)$;  $H_{23}$: $I \in [220,230)$;  $H_{24}$: $I \in [230,240)$;
   $H_{25}$: $I \in [240,255]$.  


**B. Color-Code Method**

The 24-bit of RGB color intensities can be transformed into a 6-bit color code, composed from the most significant 2 bits of each of the three color components, as illustrated in the following figure.

![Color Code Diagram](image)

- Red
- Green
- Blue

The 6-bit color code will provide 64 histogram bins.

**What to demo:**

1. **User query interface**
   The graphic user interface should allow users to browse the image database, select the query image, and to view the retrieved images. Given a query image, the retrieved images should be displayed to user according to their similarity ranks to the query image.

2. **Implementation of two color histogram comparison methods**
   You need to demo the query results using the two different color histogram comparison methods described above. Users are allowed to switch between two different query methods within the same application.