Solution for Problem #1

The results are shown in the following two figures. Note that the specific value of each bin may vary a little bit due to the different approximation approaches employed in quantization.

Pixel distribution: $n_1=n_2=n_3=4$

Pixel distribution: $n_1=n_2=n_3=5$
The following Matlab codes can be viewed as an example.

**MATLAB code for colorhist**

```matlab
function [ img_hist ] = colorhist ( img, n1, n2, n3 )
% find the quantization region for each pixel in R/G/B channels
% note: each entry in index_R/index_G/index_B is within [0,n1-1]/[0,n2-1]/[0,n3-1], respectively
index_R = floor( double(img(:,:,1)) / floor(256/n1) );
index_G = floor( double(img(:,:,2)) / floor(256/n2) );
index_B = floor( double(img(:,:,3)) / floor(256/n3) );
% find the region for each pixel after scanning in the order of R->G->B
% note: each entry in index is an interger within [1,n1*n2*n3]
index = index_R + index_G * n1 + index_B * n1 * n2 + 1;
% do the statistics
img_hist = hist( reshape(index,1,:),[1:n1*n2*n3] );
```

**Solution for Problem #2**

**(A)** We can get the results by applying Equations 6.2-2 through 6.2-4 and verify those with Figure 6.14 (b). Note that $\theta$ here is in degrees rather than radians. The following Table shows the results of HSI computed using RGB.

<table>
<thead>
<tr>
<th></th>
<th>Red</th>
<th>Yellow</th>
<th>Green</th>
<th>Cyan</th>
<th>Blue</th>
<th>Magenta</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>G</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>B</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>H</td>
<td>0.00</td>
<td>60.00</td>
<td>120.00</td>
<td>180.00</td>
<td>240.00</td>
<td>300.00</td>
</tr>
<tr>
<td>S</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>I</td>
<td>0.33</td>
<td>0.67</td>
<td>0.33</td>
<td>0.67</td>
<td>0.33</td>
<td>0.67</td>
</tr>
</tbody>
</table>

**MATLAB code for RGB2HSI**

```matlab
function [ H, S, I ] = rgb2hsi( R, G, B )
theta = acos( 0.5 * ((R-G)+(R-B)) / sqrt((R-G)^2+(R-B)*(G-B)) ); % radian unit
theta = theta * 180 / pi; % degree unit
if( B <= G ) H = theta;
else H = 360 - theta; end
S = 1 - (3/(R+G+B)) * min( min(R,G), B );
I = (R+G+B) / 3;
```

**(B)** In uniform color space, such as CIE-L*a*b or U*V*W*, the perceptual distance between two colors is approximately proportional to the distance between the two points in the color space. However, as we discuss in the class, such property may not hold in other spaces such as RGB. Computing color distance in the HSI space is not straightforward. Some work in the research literature has derived special distance function in the HSI space.