CHAPTER 4
DESIGN OF PREPROCESSING METHODS

The main goal of the last decade is to browse and query multimedia objects, especially images, in large scaled databases. The CBIR system is required to perform automatic searches on these databases based on image content. The area focuses on image processing and machine vision community and is continuing to be an active research area of the 21st century. The problems faced by existing methods has resulted in providing optimization or different variants of existing algorithms as solutions to improve the performance of image retrieval.

The CBIR system, as mentioned in Chapter 1 (Introduction), consists of two major steps which are feature extraction and image retrieval using similarity measures. The main research goal here is to minimize the problem of returning too many false positives, which becomes a major problem when the databases are very large (Costa et al., 2010). In order to provide a solution to minimize this problem, an image preprocessing step is included before feature extraction.

Image pre-processing is an algorithmic challenge that uses image processing algorithms to enhance the representation quality of images in order to improve the quality of image retrieval process. The goal of CBIR image preprocessing is

- to determine the method to store the images in databases
- to determine the filtering method to improve the quality of the database images
- to avoid problems created by different sized images
- to normalize the database images to remove the effect of color variations
• to perform color transformation or color space conversion
• to segment the image into meaningful regions

The preprocessing step of CBIR systems concentrates on improving the interpretation of the image data and to improve the feature extraction and image searching. Rugna et al. (2011), the use of segmentation algorithm in CBIR systems is time consuming and the success of retrieval process depends on the efficiency of the algorithm used for segmentation. The segmentation algorithms are usually sensitive to over- and under- segmentation, both of these conditions affect the performance of the CBIR systems (Rubner and Tomasi, 1999).

Spatial changes in image appearance like textures can cause single textures to be split into smaller segments, leading to over-segmentation. The segmentation algorithm can also combine together small regions of different features (like color or texture), thus leading to the problem of under-segmentation. Problems may also occur when some feature in the image, although significant in size when combined together, are scattered over the image and therefore, are lost during segmentation. An example of this phenomenon is an aerial view of a town in an area with rich vegetation, where both the buildings and the vegetation are made up of numerous but represented as small texture patches. Hence the present research work proposes image retrieval techniques based on contents without using segmentation.

Phase I of the research methodology proposes preprocessing techniques that focus on four tasks which are storage method used, filtering method used, normalization and color space transformation. The research work uses JPEG (Joint Photographic Experts Group) image format for storing database images and a median filter to suppress noise whenever required. The filtering method is provided as an optional command and can be used when required. Thus, Phase I of the study deals with the following preprocessing tasks.
• Step 1 : Image Resizing and Normalization - The images are rescaled to 128 x 128 pixels using enhanced bi-cubic interpolation technique. The aim here is to normalize the image data and not to have any differences related to sizes of images.

• Step 2 : Color Space Transformation - As color distance in the RGB color space is not the same as human perception of color distance, two modified color space transformation techniques are proposed.

This chapter presents details of these techniques used by the proposed CBIR systems.

4.1. IMAGE STANDARDIZATION PROCEDURES

Figure 4.2 presents the steps involved in the image standardization procedure used by the proposed CBIR systems. The procedure combines the advantages of Discrete Wavelet Transformation (DWT), Interpolation and normalization techniques for this purpose.

As human eye is more sensitive to edge areas than smooth areas within an image (Wang et al., 2004), instead of applying the enhancement algorithm to the image as a whole, Discrete Wavelet Transformation (DWT) is used to separate the detail and edges from image. The first stage of the algorithm is the use of DWT to separate the image into detail and edge coefficients. For this purpose, 2 level decomposition using 2-D Daubechies wavelet transformation is used. Two separate algorithms are then used to improve the visual quality of the edge and detail regions separately. The bicubic approach and an edge sensitive approach are then used to enhance the detail and edge coefficients of the image respectively. Thus, only three subbands are used during interpolation. Finally, inverse discrete wavelet transformation is applied to reconstruct the enhanced and rescaled image. In order to correct the intensity
variations of the resized images, a normalization procedure is finally used. The steps involved in this procedure are given in Figure 4.1.

Figure 4.1 : Image Standardization Procedure

4.1.1. Image Interpolation

Image interpolation is a technique that is frequently used by image processing researchers during their quest for creating high quality images from low resolution images during image interpretation. Image interpolation is a process that estimates a set of unknown pixels from a set of known pixels in an image (Tam et al., 2010). According to Thevenaz et al. (2000), interpolation is
defined as a model-based recovery of continuous data from discrete data within a known range of coordinates. It can be considered as an extension to image filters where, given a smaller image as input, the interpolation model produces a larger version as the desired output. Image interpolation is also known as image resizing, image upsampling/downsampling, zooming, magnification and resolution enhancement. This research work uses interpolation for image resizing purposes.

Image interpolation works in two directions, and tries to achieve a best approximation of a pixel's color and intensity based on the values at surrounding pixels. The following example (Figure 4.2) illustrates how resizing / enlargement works and where ‘?’ pixels indicates positions to be interpolated when images are resized.

The applications of image interpolation range from general viewing of online images to sophisticated magnification of satellite images. With the rise of consumer-based digital photography, users expect to have a greater control over their digital images. Digital zooming has a role in picking up clues and details in surveillance images and video. Astronomical images from Rovers and Probes are received at an extremely low transmission rate (about 40 bytes per second), making the transmission of high-resolution data infeasible (Almira and Romero, 2011). Hence the database images are rescaled into a common size of 128 x 128. Traditional interpolation is expressed using Equation (4.1)
where \( f_k \) are the known pixel points, \( \varphi(x-k) \) is a weighting function and \( c_k \) is a coefficient dependent on \( f_k \) but not equal to \( f_k \). According to this equation, interpolation is a linear combination of weighted values of known points.

To obtain a better quality image, high-order interpolation techniques were introduced. Examples include traditional techniques like cubic convolution (Keys, 1981), spline interpolation (Lehmann et al., 1999) and linear interpolation (Blu et al., 2002). The past few decades have also witnessed the usage of transformation techniques like Discrete Cosine Transformation (Ilk and Guler, 2011) and Discrete Wavelet Transformation (Demirel and Anbarjafari, 2011). Edge-based techniques (Lee, 2008) have also been proposed which work on the fact that clear edge contours produce better visual effect. In this research, an interpolation technique that combines Discrete Wavelet Transformation and Bi-cubic Interpolation technique with Edge-Sensitivity Interpolation technique is used. Details regarding each of these techniques are presented in the following sections.

4.1.2. Bi-cubic Interpolation

Bi-cubic interpolation technique, pioneered by Boor (1978), estimates the value at a given point in the destination image by an average of 16 pixels surrounding the closest corresponding pixel in the source image (Figure 4.3). The bi-cubic interpolation uses sixteen \( (4 \times 4) \) neighboring pixels for estimation. It approximates the local intensity values using a bi-cubic polynomial surface.
The general form for a bi-cubic interpolation is as follows:

\[ I_{xy} = \sum_{i=0}^{3} \sum_{j=0}^{3} a_{ij} x^i y^j = a_{00} + a_{10} x + a_{20} x^2 + a_{11} x y + a_{02} y^2 + a_{21} x^2 y + a_{22} x^2 y^2 + a_{30} x^3 + a_{31} x^3 y + a_{13} x y^3 + a_{32} x^3 y^2 + a_{23} x^2 y^3 + a_{33} x^3 y^3 \]

(4.2)

**Figure 4.3 : Bi-cubic Interpolation**

In order to do a bi-cubic interpolation within a grid square one needs to calculate the gradients (the first derivatives) in both the \( x \) and \( y \) directions and the cross derivative at each of the four corners of the square. This gives 16 equations that determine the 16 coefficients (\( a_{ij} \)), as explained by Press *et al.* (2002).

Given a point \((x, y)\) in the destination image and the point \((l, k)\) in the source image, the formulae of bi-cubic interpolation are

\[
f(x, y) = \sum_{m=l-1}^{l+2} \sum_{n=k-1}^{k+2} g(m,n) \cdot r(m-1, dx), (dy - n + k) \]

(4.3)
where \( dx = x - l \) and \( dy = y - k \) and \( l = \lfloor x \rfloor \) and \( k = \lfloor y \rfloor \) are same as the bilinear method. The cubic weighting function \( r(x) \) is defined as

\[
    r(x) = \frac{1}{6} \left[ p(x+2)^3 - 4p(x+1)^3 + 6px^3 - 4p(x-1)^3 \right]
\]

(4.4)

where \( p(x) \) is \( p(x) = \begin{cases} x & x > 0 \\ 0 & x \leq 0 \end{cases} \). Bi-cubic approach can achieve a better performance than the bilinear method because more neighboring points are included to calculate the interpolation value.

### 4.1.3. Edge Sensitive Interpolation

Two major issues of bi-cubic interpolation technique, when applied to the whole image are the edge blurring and jagging artifacts that are introduced. These artifacts are produced by the low-pass filter used to remove the unwanted high-pass replica of the interpolated images in the frequency domain. Because it is not possible to implement the ideal low pass filter in practice, non-ideal filters such as the zero-order hold (nearest neighbor) and the first-order hold (bilinear) operators are often employed to filter out the high frequency replica (Ting and Hang, 1997). These non-ideal low-pass filters suppress low frequency components and bring in high frequency component aliasing. The former, low frequency suppression, reduces the spatial resolution of the interpolated images (blurring) and the latter, undesired high frequency aliasing, produces broken edges (jaggedness). But, an algorithm that tackles both the problems is still under research.

An example of jagging artifacts and edge blurring effect on the interpolated image is shown in Figure 4.4, where applying interpolation has degraded the quality of the output image. Thus, it is imminent to develop interpolation algorithms which can maintain the structural and visual details of an image in an efficient manner. Intuitively, edges are characterized by fast intensity value variation (i.e., large gradient). To avoid edge blurring, the
interpolation coefficients have to be spatially adaptively adjusted (www.csee.wvu.edu/~xinl/courses/ee565/image_interpolation.ppt).

Figure 4.4 : Example of Blurring and Jagging Artifacts

As both the blurring and jagging artifacts are more visible around edges of the image and degrade image quality, a two-step edge-sensitive interpolation technique is introduced. The edge sensitive technique used to interpolate the edges of an image consists of the following two steps.

Step 1: Interpolate the missing pixels along the diagonal

Step 2: Interpolate the other half missing pixels
Consider for example Figure 4.5a, where pixel x (blue colored box) has to be interpolated. Since |a-c| = |b-d| (Figure 4.5b), x has equal probability of being either black or white, and thus, the decision on black or white has to be made. To predict value for x, the algorithm considers neighboring pixels as in Figure 4.5c and the following decision rules (Figure 4.6) are applied to interpolate X. The main advantage of the algorithm is that it avoids interpolating along the direction which has a large gradient.

<table>
<thead>
<tr>
<th>(a)</th>
<th>(b)</th>
<th>(c)</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Figure 4.5" /></td>
<td><img src="image" alt="Figure 4.5" /></td>
<td><img src="image" alt="Figure 4.5" /></td>
</tr>
</tbody>
</table>

**Figure 4.5 : Edge Sensitive Interpolation**

<table>
<thead>
<tr>
<th>Condition</th>
<th>Decision</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>a-c</td>
</tr>
<tr>
<td></td>
<td>a-c</td>
</tr>
<tr>
<td></td>
<td>a-c</td>
</tr>
</tbody>
</table>

**Figure 4.6 : Decision Rules for Interpolating Edges**

4.1.4. Image Normalization

The final task of the Step 1 of preprocessing algorithm is image normalization. Color image normalization is concerned with the distribution of color values in an image. The procedure in Figure 4.7 is used to normalize the rescaled images in this research work.
Figure 4.7 : Image Normalization Procedure

4.2. COLOR SPACE TRANSFORMATION

A color space is a useful conceptual tool for understanding the color capabilities of a particular image. It can be defined as a subset of colors for which a set of numbers used to describe the color can be directly mapped to the manner in which the human eye responds to the color (trichromaticity values).
A color model is a mathematical amalgamation for expressing a color value using a set (usually three) of numbers.

In image databases, color images are generally represented using RGB (Red, Green, Blue) color space model (http://en.wikipedia.org/wiki/RGB_color_model). Frequently, the CBIR systems convert RGB color space to two other frequently used color models, namely, HSV (Hue-Saturation-Value) and YCbCr (Y-Luma, C-Chroma of blue and red components) color models. All the color spaces define a method of uniquely specifying colors via three numbers. This section first presents the conventional color space models followed by the enhanced models.

4.2.1. RGB Color Space

An RGB (Red, Green, Blue) color space is any additive color space based on the RGB color model. The RGB color model is an additive color model in which red, green, and blue light are added together in various ways to reproduce a broad array of colors. A particular RGB color space is defined by the three chromaticities of the red, green, and blue additive primaries, and can produce any chromaticity that is the triangle defined by those primary colors (Figure 4.8) and the representation is shown in Figure 4.9. The complete specification of an RGB color space also requires a white point chromaticity and a gamma correction curve.

Figure 4.8 : RGB Cube
Figure 4.9: RGB color Representation

RGB is a convenient color model for computer graphics because the Human Visual System (HVS) works in a way that is similar to an RGB color space. Some examples of 24-bit representation of RGB colors is given below:

- (255, 255, 255) represents white
- (0, 0, 0) represents black
- (255, 0, 0) represents red
- (0, 0, 255) represents blue
- (0, 255, 0) represents green
- (255, 255, 0) represents yellow
- (255, 0, 255) represents magenta
- (0, 255, 255) represents cyan

4.2.2. HSV Color Space

A three dimensional representation of the HSV color space is a hexacone, where the central vertical axis represents the Intensity (Figure 4.10).

Figure 4.10: HSV Hexacone
Hue is defined as an angle in the range \([0,2\pi]\) relative to the Red axis with red at angle 0, green at \(2\pi/3\), blue at \(4\pi/3\) and red again at \(2\pi\). Saturation is the depth or purity of the color and is measured as a radial distance from the central axis with value between 0 at the center to 1 at the outer surface. For \(S=0\), as one moves higher along the Intensity axis, one goes from Black to White through various shades of gray. On the other hand, for a given Intensity and Hue, if the Saturation is changed from 0 to 1, the perceived color changes from a shade of gray to the most pure form of the color represented by its Hue.

**Hue** - In HSV, hue represents color. In this model, hue is an angle from 0 degrees to 360 degrees. The hue colors are given in Table 4.1.

<table>
<thead>
<tr>
<th>Angle</th>
<th>Color</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-60</td>
<td>Red</td>
</tr>
<tr>
<td>60-120</td>
<td>Yellow</td>
</tr>
<tr>
<td>120-180</td>
<td>Green</td>
</tr>
<tr>
<td>180-240</td>
<td>Cyan</td>
</tr>
<tr>
<td>240-300</td>
<td>Blue</td>
</tr>
<tr>
<td>300-360</td>
<td>Magenta</td>
</tr>
</tbody>
</table>

**Saturation** - Saturation indicates the range of grey in the color space. It ranges from 0 to 100\%. Sometimes the value is calculated from 0 to 1. When the value is '0,' the color is grey and when the value is '1,' the color is a primary color. A faded color is due to a lower saturation level, which means the color contains more grey. In other words, while adding black, the saturation and intensity drops and while adding white, the color becomes lighter. The image in Figure 4.11 demonstrates this; the first box is fully saturated, the second has black added and the third has white added.
Value: Value is the brightness of the color and varies with color saturation. It ranges from 0 to 100%. When the value is '0' the color space will be totally black. With the increase in the value, the color space brightness up and shows various colors. In other words adding black or white to a color affects value. Tints are colors with added white, and shades are colors with added black. The image below shows tints and shades of the hues of the color wheel (Figure 4.12).

4.2.3. YCbCr Color Space

In this model, y represents the luma component, Cb and Cr represents the blue-difference and red-difference chroma components. The YCbCr model uses additional information about image chroma and hue on top of the standard R, G, B components. As this property can effectively differentiate skin and non-skin regions and it is used in this research. In this model, Y is represented in a 8-bit range of 16-235 and Chroma has a range of 16-240. A representation...
model of YCbCr color space model along with an example of CbCr plane at constant luma value of 0.5 is shown in Figure 4.13a and 4.13b.

![YCbCr Example](image)

(a) (b)

Figure 4.13 : YCbCr Example

4.2.4. Enhanced Color Space Models

The HSV color space (Hue, Saturation, Value) is often used by people who are selecting colors (e.g., of paints or inks) from a color wheel or palette, because it corresponds better to how people experience color than the RGB color space does. This goes for digital image users also and hence HSV is considered as a feature in the present research work. But, digital equipments like digital camera often produce image in RGB color model. As most of the images in image databases are photographic images, RGB is also considered. This research work enhances RGB, HSV and YCbCr color space models to extract image features.

The RGB model is the most used color space model throughout computer graphics. This has various advantages like being straightforward and matching well with HVS’s strong response to red, green and blue colors. But they also have some drawbacks like being highly redundant and correlated. For example, all channels hold luminance information, reduces coding efficiency, which make it difficult to use during image processing tasks. Moreover, the color distances in RGB color space do not reflect the actual human perceptual color distance (Wang et al., 1997).
The HSV and YCbCr color space models used in CBIR systems are the most popularly used color spaces apart from RGB color space model. All the three models have been proved to produce efficient results (Deshpande and Borse, 2011; Kaur and Banga, 2013) but its performance degrades with images having multiple objects with different illumination and poses. To solve this problem, an enhanced color model, named as RHC model is used and consists of two steps as explained below.

- **Step 1 : Enhancing RGB Color Space Model**

In step 1, the RGB color model is enhanced to convert and store images in a component color space with intensity and perceived contrasts. This model is termed as ER'G'B' color space model. The ER'G'B' color model converts and stores image in a component color space with intensity and perceived contrasts. The new color values R', G', and B', are obtained from the converted RGB color values (R, G, and B) using the following Equations (Equation 4.5 – 4.7).

\[
R' = \frac{(R + G + B)}{3} \tag{4.5}
\]

\[
G' = \frac{(R + (M - B))}{2} \tag{4.6}
\]

\[
B' = \frac{(R + 2 \times (M - G) + B)}{4} \tag{4.7}
\]

In the above equation, M represents maximum possible values for each color component in the RGB color space. For a standard 24-bit color image, this value is equal to 255 and thus, it is clear that each color component of ER'G'B' ranges from 0-255 as well. This model is similar to the opponent color axes presented in Equations (4.8-4.10) defined by Swain and Ballard (1991).

\[
RG = R - 2 \times G + B \tag{4.8}
\]

\[
BY = -R - G + 2 \times B \tag{4.9}
\]

\[
WB = R + G + B \tag{4.10}
\]

Besides the perception correlation properties (Hurvich and Jameson, 1957) of such an opponent color space, one important advantage of this alternative space
is that the R' axis, or the intensity, can be more coarsely sampled than the other two axes on color correlation. This reduces the sensitivity of color matching to a difference in the global brightness of the image and it reduces the number of bins and subsequent storage in the color histogram indexing.

- **Step 2 : RHC Color Space Model**

  The RHC model enhances the RGB color space model by including important information from both HSV and YCbCr models along with R', G' and B' components of ER'G'B' color space model. According to, Garcia et al. (1998); Rahman et al. (2006) and Ghazali et al. (2012), histogram analysis of RGB color components show that it is uniformly spread across a large spectrum of values, the H in HSV model shows significant discrimination of color regions and high discrimination power is shown by pixels possessing similar chroma (Cb and Cr) values. Thus, the most significant contributor of each color model is R', G', B', H, Cb and Cr and hence the EHRC model combines these components of the ER'G'B', HSV and YCbCr color spaces to form a new enhanced version of color space model.

### 4.2.5. Color Space Conversion

With both the proposed color space models, a conversion of RGB image to HSV and YCbCr color spaces is required. This section presents the method of conversion used for this purpose.

- **RGB to HSV Conversion**

  Conversion of RGB color space to HSV is obtained by first finding the maximum and minimum values from the R'G'B' triplet. Using these values the saturation (S) and Value (V) are calculated using Equations (4.11) and (4.12).

  \[
  S = \frac{(\text{max} - \text{min})}{\text{max}} \quad \text{(4.11)}
  \]

  \[
  V = \text{max} \quad \text{(4.12)}
  \]
The calculation of H (Hue) begins by calculating R", G" and B" using Equation (4.13).

\[
\begin{align*}
R'' &= \frac{\text{max} - R}{\text{max} - \text{min}}; \\
G'' &= \frac{\text{max} - G}{\text{max} - \text{min}}; \\
B'' &= \frac{\text{max} - B}{\text{max} - \text{min}}
\end{align*}
\] (4.13)

The value of H is then calculated as follows:

if \( S = 0 \) then H is undefined (monochrome image)

else if \( R' = \text{max} \)

\[
\begin{align*}
G' &= \text{min} & H &= 5 + B'' \\
G' &\neq \text{min} & H &= 1 - G''
\end{align*}
\]

else if \( G' = \text{max} \)

\[
\begin{align*}
B' &= \text{min} & H &= R'' + 1 \\
B' &\neq \text{min} & H &= 3 - B''
\end{align*}
\]

else if \( R' = \text{max} \)

\[
H = 3 + G''
\]

else

\[
H = 5 - R''
\]

The H value is then multiplying it with 60 degrees, which converts it into degrees giving HSV with S and V values between 0 and 1 and H value between 0 and 360 degrees.

The reverse process, which is converting an image in HSV color space to R'G'B', first the Hue value in the range 0 to 360 degrees is divided by 60 degrees to obtain the Hex value. From this the primary colors (a, b, and c) are calculated using Equations (4.14) – (4.17).

Secondary color = Hex - primary color (4.14)

\[
a = (1 - S)V
\] (4.15)

\[
b = (1 - (S * \text{secondary color}))V
\] (4.16)

\[
c = (1 - (S*(1 - \text{secondary color})))V
\] (4.17)
From the primary color, the R'G'B' values are then calculated using the following procedure.

if primary color = 0 then \( R' = V, G' = c, B' = a \)
else if primary color = 1 then \( R' = b, G' = V, B' = a \)
else if primary color = 2 then \( R' = a, G' = V, B' = c \)
else if primary color = 3 then \( R' = a, G' = b, B' = V \)
else if primary color = 4 then \( R' = c, G' = a, B' = V \)
else if primary color = 5 then \( R' = V, G' = a, B' = b \)

- **RGB to YCbCr Conversion**

The conversion process to convert RGB to YCbCr color format and YCbCr to RGB color format is performed using Equations (4.18) and (4.19) respectively.

\[
\begin{align*}
\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} &= \begin{bmatrix} 16 \\ 128 + \frac{1}{256} \\ 128 \end{bmatrix} \begin{bmatrix} 65.738 & 129.057 & 25.064 \\ -37.945 & -74.494 & 112.439 \\ 112.439 & -94.154 & -18.285 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \\
&= \begin{bmatrix} 290.082 & 0 & 408.583 \\ 298.082 & -100.291 & -208.120 \\ 298.082 & -516.411 & 0 \end{bmatrix} \begin{bmatrix} Y - 16 \\ Cb - 128 \\ Cr - 128 \end{bmatrix}
\end{align*}
\]

4.3. **CHAPTER SUMMARY**

In this chapter, the two preprocessing techniques designed to improve the performance of the proposed CBIR systems were discussed. The next step is to extract the image features and to find a solution to the curse of dimensionality. The techniques used for this purpose is discussed in the next chapter, Chapter 5, Feature Extraction and Reduction.