DEFENCE TERRAIN RESEARCH LABORATORY (DTRL),
DEFENCE RESEARCH AND DEVELOPMENT ORGANISATION (DRDO),
NEW DELHI

**No. : DTRL/THCD/TRNG/2015/S/10W/95**
**Date: July 2015**

**Project Report on**

# IMAGE CONTRAST ENHANCEMENT

## [Using Histogram Equalization & Cubic Spline Interpolation]

**SUBRAT PRASAD PANDA**

B.Tech in ELECTRONICS & INSTRUMENTATION ENGG, 3RD YEAR

National Institute of Technology, Silchar

# Acknowledgement

*I would like to thank* ***Dr MR Bhutiyani****, Sc. G, Director, DTRL for providing me the opportunity of undertaking the project training at DTRL and also for providing all the essential facilities for the completion of my project.*

*I would like to express my profound gratitude and immense regard to Dr. B .G Prusty, Group Head, GSAD and my project guide* ***Miss. Munmun Baisantry****, Scientist-C whose constant inspiration, enthusiasm and encouragement made this project a success. Despite of their busy schedule, they kept track of my progress and have been instrumental in solving problems that I faced from time to time.*

*Subrat Prasad Panda*
*NIT Silchar*

# Abstract

Human vision is more sensitive to find *difference* in the image rather than finding *similarity*. So, we are faster at identifying difference in the image. When we increases the contrast in an image, the difference between its *intensity* values increases, which gives a better visual stimulus to the eye.

A *colored* image contains three or more values depending on the color model used. For example there are 3 values in the *RGB, HSV and LAB*. The *RGB* model are used for display purposes only and *HSV/LAB* are used for the image enhancement purposes.

The contrast enhancement can't be directly applied to the RGB values as it will change the color of the image, resulting to the *color distortion*. So, the RGB image is converted to the color model in which the parameter value have pixel intensity information or the model which *separates* the color information from intensity information. For example *V* in HSV and *L* in LAB contains the intensity information of the image. The contrast enhancement can now applied to intensity information of the image. Can be done in

Contrast Enhancement can be done in spatial domain or in the frequency domain. In spatial domain the pixel values are directly modified, which is easier for implementation. The preferred method in the report are (a) Histogram Equalization: the image will have almost equal numbers of pixel on each intensity level and occupy full range of intensity level *(0-255 in 8 bit image)*. (b) Cubic Spline Interpolation: the pixel intensity *transformation* function will be found out by using the interpolation of user defined data points. The method used is cubic equation spline interpolation.

# Contents

# 1. Introduction

## 1.1 Image:

A digital image is a numeric representation (normally binary) of a two-dimensional image. Depending on whether the image resolution is fixed, it may be of vector or raster type. By itself, the term "digital image" usually refers to raster images or bitmapped images. In imaging science, **image processing** is processing of images using mathematical operations by using any form of signal processing for which the input is an image, such as a photograph or video frame; the output of image processing may be either an image or a set of characteristics or parameters related to the image. Most image-processing techniques involve treating the image as a two-dimensional signal and applying standard signal-processing techniques to it.

## 1.2 Contrast:

Contrast is the difference in luminance or color that makes an object (or its representation in an image or display) distinguishable. In visual perception of the real world, contrast is determined by the difference in the color and brightness of the object and other objects within the same field of view. Because the human visual system is more sensitive to contrast than absolute luminance, we can perceive the world similarly regardless of the huge changes in illumination over the day or from place to place. The maximum *contrast* of an image is the contrast ratio or dynamic range.

*Figure 1 - Example of Image with Enhanced contrast*

## 1.3 Histogram:

Image editors have provisions to create an image histogram of the image being edited. The histogram plots the number of pixels in the image (vertical axis) with a particular brightness value (horizontal axis). Algorithms in the digital editor allow the user to visually adjust the brightness value of each pixel and to dynamically display the results as adjustments are made. Improvements in picture brightness and contrast can thus be obtained.

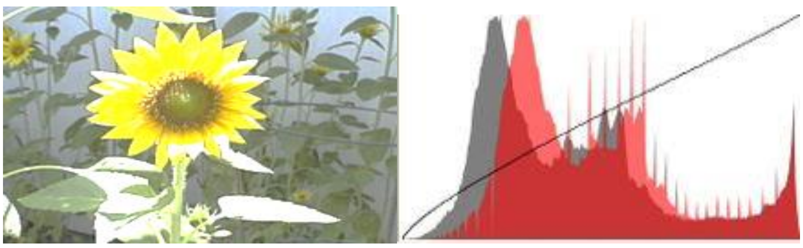


*Figure 2 – An Image with its histogram*

# 2. Color Model & Spaces

## 2.1 Color:

Color is the brains reaction to a *specific visual stimulus*. Although we can precisely describe color by measuring its spectral power distribution (the intensity of the visible electro-magnetic radiation at many discrete wavelengths) this leads to a large degree of redundancy. The reason for this redundancy is that the eye's retina samples color using only three broad bands, roughly corresponding to red, green and blue light. The signals from these color sensitive cells (cones), together with those from the rods (sensitive to intensity only), are combined in the brain to give several different "*sensations*" of the color.

## 2.2 Color Model:

A color model is an abstract mathematical model describing the way colors can be represented as tuples of numbers, typically as three or four values or color components.

When this model is associated with a precise description of how the components are to be interpreted (viewing conditions, etc.), the resulting set of colors is called color space .

*Five types of color models are:*

- **CIE color model** ***(CIE-****International Color Consortium)*
- **RGB color model**
- **CMYK color model**
- **YUV color model**
- **HSL/HSV**

## 2.3 Color space:

A **color space** is a specific organization of colors. In combination with physical device profiling, it allows for reproducible representations of color, in both analog and digital representations.

A color space may be arbitrary, with particular colors assigned to a set of physical color swatches and corresponding assigned names or numbers such as with the Pantone system, or structured mathematically, as with Adobe RGB or sRGB.

List of color spaces:-

1. **Under CIE color model**
   a. CIE 1931 XYZ
   b. CIE LUV
   c. CIE LAB
   d. CIE UVW
2. **Under RGB color model**
   a. sRGB
   b. Adobe RGB
   c. Adobe wide gamut RGB
   d. scRGB
   e. ProPhoto RGB
   f. Rec. 709
3. **Under YUV color model**
   a. YUV,YIQ,YDbDr
   b. YPbPr,YCbCr
   c. xvYCC
4. **HSV/HSL color space**
5. **CMYK color space**

## 2.4 Color Gamut:

A color gamut is the area enclosed by a color space in three dimensions. It is usual to represent the gamut of a color reproduction system graphically as the range of colors available in some device independent color space. Often the gamut will be represented in only two dimensions.

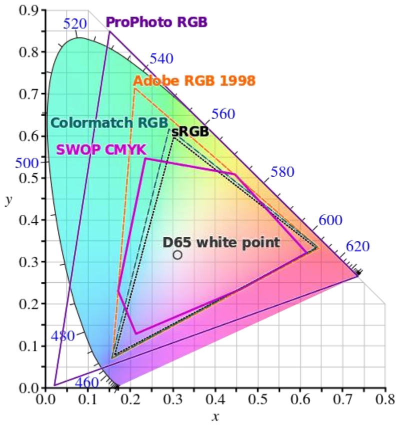



*Figure 3 - Color gamut shown on CIE 1931 XYZ chromaticity diagram*

## 2.5 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. The name of the model comes from the initials of the three additive primary colors, red, green, and blue. The main purpose of the RGB color model is for the sensing, representation, and display of images in electronic systems, such as televisions and computers, though it has also been used in conventional photography. Before the electronic age, the RGB color model already had a solid theory behind it, based in human perception of colors. RGB is a **device-dependent** color model: different devices detect or reproduce a given RGB value differently, since the color elements (such as phosphors or dyes) and their response to the individual R, G, and B levels vary from manufacturer to manufacturer, or even in the same device over time. Thus an RGB value does not define the same color across devices without some kind of management. Zero intensity for each component gives the darkest color (no light, considered the *black*), and full intensity of each gives a white; the *quality* of this white depends on the nature of the primary light sources, but if they are properly balanced, the result is a neutral white matching the system's white point. When the intensities for all the components are the same, the result is a shade of gray, darker or lighter depending on the intensity.

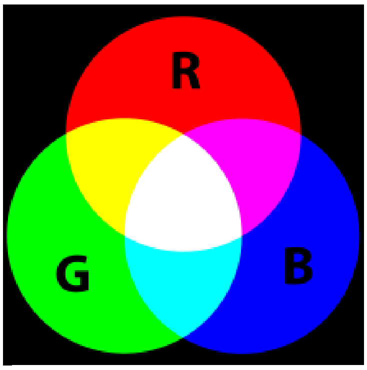



*Figure 4 – Additive color mixing*

## 2.6 HSV/HSL Color Model:

HSL (hue-saturation-lightness) and HSV (hue-saturation-value) are the two most common cylindrical-coordinate representations of points in an RGB color model. Developed in the 1970s for computer graphics applications, HSL and HSV are used today in color pickers, in image editing software, and less commonly in image analysis and computer vision.

The two representations rearrange the geometry of RGB in an attempt to be more intuitive and perceptually relevant than the Cartesian (cube) representation, by mapping the values into a cylinder loosely inspired by a traditional color wheel. The angle around the central vertical axis corresponds to "hue" and the distance from the axis corresponds to "saturation". These first two values give the two schemes the 'H' and 'S' in their names. The height corresponds to a third value, the system's representation of the perceived luminance in relation to the saturation. HSL and HSV are both cylindrical geometries, with hue, their angular dimension, starting at the red primary at 0°, passing through the green primary at 120° and the blue primary at 240°, and then wrapping back to red at 360°. In each geometry, the central vertical axis comprises the *neutral*, *achromatic*, or *gray* colors, ranging from black at lightness 0 or value 0, the bottom, to white at lightness 1 or value 1, the top.

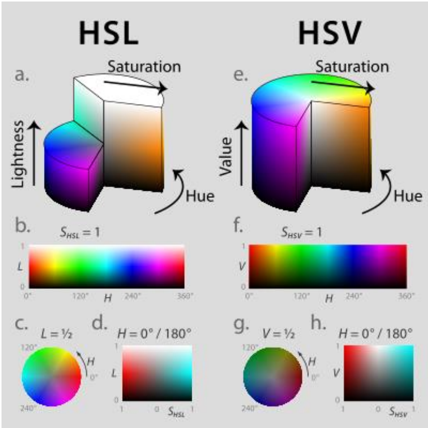


*Figure 5 - HSL (a–d) and HSV (e–h)*

## 2.7 LAB Color Space:

A ***Lab* color space** is a color-opponent space with dimension ***L*** for lightness, ***a*** and ***b*** for the color-opponent dimensions, based on nonlinearly compressed (e.g. CIE XYZ color space) coordinates. The terminology originates from the three dimensions of the **Hunter1948 color space**, which are ***L***, ***a***, and ***b***. However, *Lab* is now more often used as an informal abbreviation for the L-a-b representation of the CIE 1976 color space (or CIELAB).

The L*a*b* color space includes all perceivable colors, which means that its gamut exceeds those of the RGB and CMYK color models (for example, ProPhoto RGB includes about 90% all perceivable colors). One of the most important attributes of the L*a*b*-model is **device independence**. It is used as an interchange format between different devices as for its device independency.

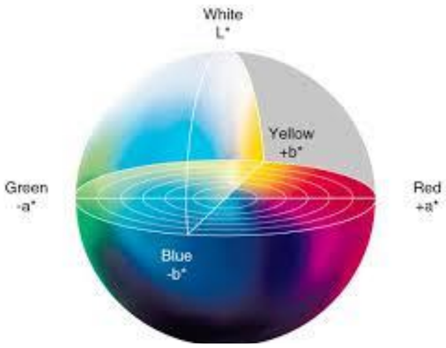


*Figure 6 - LAB Color space in 3D*

The space is usually mapped onto a three-dimensional integer space for device-independent digital representation, and for these reasons, the L*, a*, and b* values are usually absolute, with a pre-defined range. The lightness, L*, represents the darkest black at L* = 0, and the brightest white at L* = 100. The color channels, a* and b*, will represent true neutral gray values at a* = 0 and b* = 0. The red/green opponent colors are represented along the a* axis, with green at negative a* values and red at positive a* values. The yellow/blue opponent colors are represented along the b* axis, with blue at negative b* values and yellow at positive b* values.

# 3. Color Space Conversion:

Color space conversion is what happens when a color management module (CMM) translates color from one device's space to another. Conversion may require approximations in order to preserve the image's most important color qualities. Knowing how these approximations work can help you control how the photo may change — hopefully maintaining the intended look or mood.

## 3.1 RGB to HSV Color Space Conversion:

The dimensions of the HSV and HSL geometries – simple transformations of the not-perceptually-based RGB model – are not directly related to the photometric color-making attributes of the same names, as defined by scientists such as the CIE or ASTM. Nonetheless, it is worth reviewing those definitions before leaping into the derivation of our models.

**Hue:** The "attribute of a visual sensation according to which an area appears to be similar to one of the perceived colors: red, yellow, green, and blue, or to a combination of two of them".

**Lightness, value:** The "brightness relative to the brightness of a similarly illuminated white".

**Saturation:** The "colorfulness of a stimulus relative to its own brightness".

*Brightness* and *colorfulness* are absolute measures, which usually describe the spectral distribution of light entering the eye, while *lightness* and *Chroma* are measured relative to some white point, and are thus often used for descriptions of surface colors, remaining roughly constant even as brightness and colorfulness change with different illumination. Saturation can be defined as either the ratio of colorfulness to brightness or of Chroma to lightness. These are the RGB-HSV conversions given by Travis. To convert from RGB to HSV (assuming normalized RGB values) first find the maximum and minimum values from the RGB triplet.

Saturation "***S***", is then:

$$S = \frac{max - min}{max}$$

Value "***V***", is:

$$\mathbf{V = max}$$

The Hue "***H***", is then calculated as follows. First calculate R'G'B':

$$R' = \frac{max - R}{max - min}$$

$$G' = \frac{max - G}{max - min}$$

$$R' = \frac{max - R}{max - min}$$

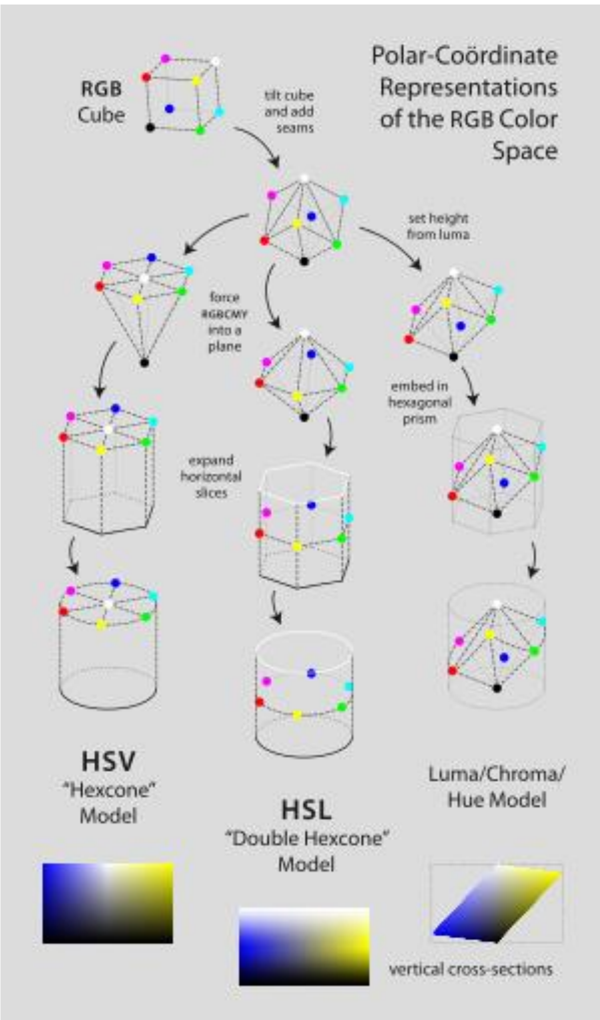



*Figure 7 - The geometric derivation of the cylindrical HSL and HSV representations of an RGB "color cube".*

If saturation “*S*”, is 0 (zero) then hue is undefined (i.e. the color has no hue therefore it is monochrome) otherwise:

Then, if $R = max$ and $G = min$

**H = 5 + B’**

Else if $R = max$ and $G6 \neq min$

**H = 1−G’**

Else if $G = max$ and $B = min$

**H = R’ + 1**

Else if $G = max$ and $B6 \neq min$

**H = 3−B’**

Else if $R = max$

**H = 3 + G’**

Otherwise

**H = 5−R’**

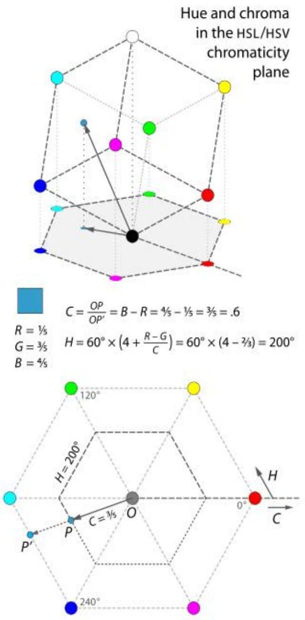



*Figure 8 - Both hue and Chroma are defined based on the projection of the RGB cube onto a hexagon in the*

Hue “***H***”, is then converted to degrees by multiplying by 60 giving HSV with S and V between 0 and 1 and H between 0 and 360.

## 3.2 HSV to RGB Color Space Conversion:

To convert back from HSV to RGB first take Hue, H, in the range 0 to 360 and divide by 60:

**Hex =H÷ 60**

Then the values of primary color, secondary color, a, b and c are calculated.

**Primary color = floor (Hex)**

**Secondary color = Hex - primary color**

**a = (1−S) × V**

**b = (1−(S × secondary color)) ×V**

**c = (1−(S × (1−secondary color))) × V**

Finally we calculate **RGB** as follows:

If primary color = 0 then **R = V, G = c, B = a**

If primary color = 1 then **R = b, G = V, B = a**

If primary color = 2 then **R = a, G = V, B = c**

If primary color = 3 then **R = a, G = b, B = V**

If primary color = 4 then **R = c, G = a, B = V**

If primary color = 5 then **R = V, G = a, B = b**

## 3.3 RGB to LAB Color Space Conversion:

Because our goal is to manipulate RGB images, which are often of unknown phosphor chromaticity, we first show a reasonable method of converting RGB signals to *Ruderman et al.*'s perception-based color space lαβ. Because lαβis a transform of LMS cone space, we first convert the image to LMS space in two steps. The first is a conversion from RGB to XYZ tristimulus values. This conversion depends on the phosphors of the monitor that the image was originally intended to be displayed on. Because that information is rarely available, we use a device-independent conversion that maps white in the chromaticity diagram (CIE xy) to white in RGB space and vice versa. Because we define white in the chromaticity diagram as x= X/(X+ Y+ Z) = 0.333, y = Y/(X + Y + Z) = 0.333, we need a transform that maps X= Y= Z= 1 to R= G= B= 1. To achieve this, we modified the XYZitu601-1 (D65) standard conversion matrix to have rows that add up to 1. The International Telecommunications Union standard matrix is

$$\boldsymbol{M}_{\mathbf{itu}}=\begin{bmatrix} 0.4306 & 0.3415 & 0.1784 \\ 0.2220 & 0.7067 & 0.0713 \\ 0.0202 & 0.1295 & 0.9394 \end{bmatrix}$$

By letting $\mathbf{M}_{\mathbf{itu}}x= (111)^T$ and solving for x, we obtain a vector x that we can use to multiply the columns of matrix $M_{itu}$, yielding the desired RGB to XYZ conversion:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.5141 & 0.3239 & 0.1604 \\ 0.2651 & 0.6702 & 0.0641 \\ 0.0241 & 0.1228 & 0.8444 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

Compared with $\mathbf{M}_{\mathbf{itu}}$, this normalization procedure constitutes a small adjustment to the matrix's values. Once in device-independent XYZ space, we can convert the image to LMS space using the following conversion:

$$\begin{bmatrix} L \\ M \\ S \end{bmatrix} = \begin{bmatrix} 0.3897 & 0.6890 & -0.0787 \\ -0.2298 & 1.1834 & 0.0464 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

**Combining these two matrices gives the following transformation between RGB and LMS cone space:**

$$\begin{bmatrix} L \\ M \\ S \end{bmatrix} = \begin{bmatrix} 0.3811 & 0.5783 & 0.0402 \\ 0.1967 & 0.7244 & 0.0782 \\ 0.0241 & 0.1288 & 0.8444 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

The data in this color space shows a great deal of skew, which we can largely eliminate by converting the data to logarithmic space:

$$\mathbf{L} = \log L$$

$$\mathbf{M} = \log M$$

$$\mathbf{S} = \log S$$

Using an ensemble of spectral images that represents a good cross-section of naturally occurring images, *Ruderman et al*. proceed to decorrelate these axes. Their motivation was to better understand the human visual system, which they assumed would attempt to process input signals similarly. We can compute maximal decorrelation between the three axes using principal components analysis (PCA), which effectively rotates them. The three resulting orthogonal principal axes have simple forms and are close to having integer coefficients. Moving to those nearby integer coefficients, Ruderman et al. suggest the following transform:

$$\begin{bmatrix} l \\ \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{3}} & 0 & 0 \\ 0 & \frac{1}{\sqrt{6}} & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & -2 \\ 1 & -1 & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{L} \\ \boldsymbol{M} \\ \boldsymbol{S} \end{bmatrix}$$

If we think of the L channel as red, the M as green, and the S as blue, we can see that this is a variant of many opponent-color models:

$$\text{Achromatic} \propto r + g + b$$

$$\text{Yellow–blue} \propto r + g - b$$

$$\text{Red–green} \propto r - g$$

Thus the **l** axis represents an achromatic channel, while the **α** and **β** channels are chromatic yellow–blue and red–green opponent channels.

## 3.4 LAB to RGB Color Space Conversion:

After color processing, we must transfer the result back to RGB to display it. For convenience, here are the inverse operations. We convert from lαβ to LMS using this matrix multiplication:

$$\begin{bmatrix} \boldsymbol{L} \\ \boldsymbol{M} \\ \boldsymbol{S} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & -1 \\ 1 & -2 & 0 \end{bmatrix} \begin{bmatrix} \frac{\sqrt{3}}{3} & 0 & 0 \\ 0 & \frac{\sqrt{6}}{6} & 0 \\ 0 & 0 & \frac{\sqrt{2}}{2} \end{bmatrix} \begin{bmatrix} l \\ \alpha \\ \beta \end{bmatrix}$$

Then, after raising the pixel values to the power ten to go back to linear space, we can convert the data from LMS to RGB using:

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 4.4679 & -3.5873 & 0.1193 \\ -1.2186 & 2.3809 & -0.1624 \\ 0.0497 & -0.2439 & 1.2045 \end{bmatrix} \begin{bmatrix} L \\ M \\ S \end{bmatrix}$$

# 4. Histogram equalization:

This method usually increases the global contrast of many images, especially when the usable data of the image is represented by close contrast values. Through this adjustment, the intensities can be better distributed on the histogram. This allows for areas of lower local contrast to gain a higher contrast. Histogram equalization accomplishes this by effectively spreading out the most frequent intensity values.

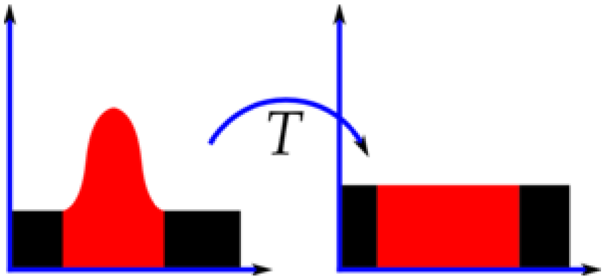



*Figure 9 - Histogram equalization*

The method is useful in images with backgrounds and foregrounds that are both bright or both dark. In particular, the method can lead to better views of bone structure in x-ray images, and to better detail in photographs that are over or under-exposed.

Consider for a moment continuous function ***s***, and let the variable ***r*** represent the intensity levels (V from HSV and L from LAB) of the image to be enhanced .In the initial part of our discussion we assume that ***r*** has been normalized to the interval [0, 1] . Later, we consider a discrete formulation and allow pixel values to be in the interval [0, L-1].

For any ***r*** satisfying the aforementioned conditions, we focus attention on transformations of the form ***s***

$$s = T(r) \qquad 0 \leq r \leq 1$$

That produce a level ***s*** for every pixel value ***r*** in the original image. For reasons that will become obvious shortly, we assume that the transformation function T(r) satisfies the following conditions:

**(a) T(r) is single-valued and monotonically increasing in the interval $0 \leq r \leq 1$; and**

**(b) $0 \leq T(r) \leq 1$ for $0 \leq r \leq 1$.**

The requirement in (a) that T(***r***) be single valued is needed to guarantee that the inverse transformation will exist, and the monotonicity condition preserves the increasing order from black to white in the output image. A transformation function that is not monotonically increasing could result in at least a section of the intensity range being inverted, thus producing some inverted intensity levels in the output image. While this may be a desirable effect in some cases, that is not what we are after in the present discussion.

Finally, condition (b) guarantees that the output intensity levels will be in the same range as the input levels. Figure 3.16 gives an example of a transformation function that satisfies these two conditions. The inverse transformation from s back to r is denoted

$$r = T^{-1}(s) \quad ; 0 \leq s \leq 1.$$

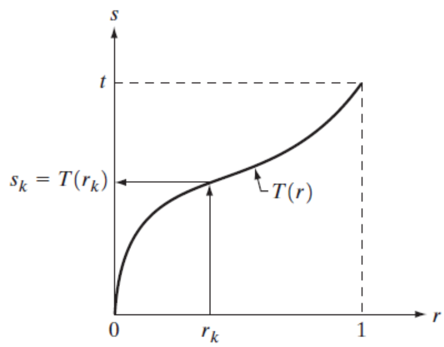



*Figure 10 - Intensity level transformation function that is both single valued and monotonically increasing.*

The Intensity levels in an image may be viewed as random variables in the interval [0, 1].One of the most fundamental descriptors of a random variable is its probability density function (PDF).

Let pr(***r***) and ps(***s***) denote the probability density functions of random variables ***r***and ***s***, respectively. A basic result from an elementary probability theory is that, if pr(***r***) and T(***r***) are known and satisfies condition (a), then the probability density function ps(***s***) of the transformed variable s can be obtained using a rather simple formula:

$$ds \times ps(s) = dr \times pr(r)$$

$$\text{as } \int pr(r).dr = 1 \quad \text{and } \int ps(s).ds=1$$

$$\text{» } ps(s)=\text{constant}=1$$

$$\text{so, } ds = pr(r) \times dr$$

$$\boxed{\text{» } s = \int_0^1 pr(r).dr}$$

Thus, the probability density function of the transformed variable, s, is determined by the intensity-level PDF of the input image and by the chosen transformation function.

In the discrete form the equation 1.1 will be have form:

$$s = \sum_{0}^{r=1} pr(r)$$

Where s is the new intensity level and is replaced to the r for histogram equalized image.

## 4.1 Methodology:

**For histogram Table:**

```
for (scan all rows) {
    for(scan all columns) {
        v= fetch V from HSV with range [0-1] from each pixel of the image
        V=(int)(v*255.0) converting v to the range of 0-255
        histogram[V]= histogram[V] + (1.0/(row*col)) for generating pixel probability density table } }
```

**For Histogram Mapping Table:**

The histogram mapping function table is the cumulative addition of the histogram probability density table

```
sum[0]=histogram[0]
for( count from 1 to 255)
  {sum[i] = histogram[i] + sum[i-1];}
```

After finding the Histogram Mapping table (look up table) the intensity values in the image are assigned to new intensity values from the look up table

# 5. Spline Interpolation:

The intensity levels of the image are mapped to new intensity level by transformation function. The curve of transformation function passes through fixed points defined by the user. The curve between the fixed points are cubic function and is determined by spline interpolation.

In the mathematical field of numerical analysis, **spline interpolation** is a form of interpolation where the interpolant is a special type of piecewise polynomial called a spline. Spline interpolation is often preferred over polynomial interpolation because the interpolation error can be made small even when using low degree polynomials for the spline. Spline interpolation avoids the problem of Runge's phenomenon, in which oscillation can occur between points when interpolating using high degree polynomials.

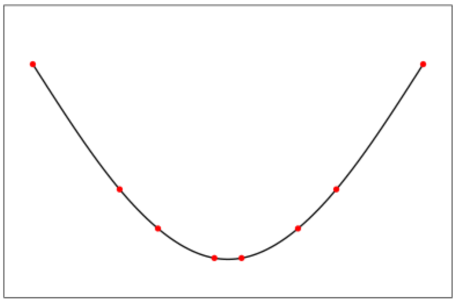


*Figure 11 - Interpolation of points*

## 5.1 Equations for cubic spline interpolation:

The Data Points: We have the ordered x-values $x0 < x1 < \cdots < xn$ and the corresponding y-values y0,yi,...,yn.

**The Spline**: For $xj \le x \le xj+1$, let

$\mathbf{S_j(x) = a_j + b_j(x-x_j) + c_j(x-x_j)^2 + d_j(x-x_j)^3}$, $j = 0,1,\ldots,n-1$ and then the spline is

$$S(x) = \begin{cases} S0(x), & x0 \le x \le x1 \\ S1(x), & x1 \le x \le x2 \\ \vdots & \\ Sj(x), & xj \le x \le xj+1 \\ \vdots & \\ Sn-1(x), & xn-1 \le x \le xn \end{cases}$$

We require 4n conditions to match points and make the spline smooth.

**Interpolation:** Each piece of the spline must interpolate the points. $S_j(xj)=y_j$ and $S_j(x_{j+1})=y_{j+1}$, $j = 0, 1\ldots n-1$ (2n conditions)

**Slope matching:** From one subinterval to the next, the left and right slopes of the spline must match.

$$S'_j(x_{j+1}) = S'_{j+1}(x_{j+1}),\ j = 0, 1\ldots n-2\ (n-1 \text{ conditions})$$

**Curvature matching:** From one subinterval to the next, the left and right curvature (second derivative) of the spline must match.

$$S''_j(x_{j+1}) = S''_{j+1}(x_{j+1}),\ j = 0, 1\ldots n-2\ (n-1 \text{ conditions})$$

**Free or natural spline**: Require no curvature at the ends.

$$S''_0(x_0) = 0 \text{ and } S''_{n-1}(x_n) = 0\ (2 \text{ conditions})$$

### 5.1.1 Equations for the Coefficients (1):

For $x_j \leq x \leq x_{j+1}$ and j = 0, 1... n−1

$$\mathbf{S_j(x) = a_j + b_j(x-x_j) + c_j(x-x_j)^2 + d_j(x-x_j)^3}$$

Let $h_j = x_{j+1}-x_j$ be the length of the j-th subinterval.

**Left interpolation conditions:** $\mathbf{S_j(x_j) = aj = y_j}$, j = 0,1,...,n−1

**Right interpolation conditions:**

$S_j(x_{j+1}) = a_j + b_jh_j + c_jh_j^2 + d_jh_j^3 = y_{j+1} = a_{j+1}$, j = 0,1,...,n−1

$$\Rightarrow \mathbf{b_jh_j + c_jh_j^2 + d_jh_j^3 = a_{j+1}-a_j}, \ j = 0,1,\ldots,n-1$$

### 5.1.2 Equations for the Coefficients (2):

For $xj \leq x \leq xj+1$ and j = 0, 1,...,n−1 ,$\mathbf{S'_j(x) = b_j + 2c_j(x-x_j) + 3d_j(x-x_j)^2}$,$\mathbf{S''_j(x) = 2c_j + 6d_j(x-x_j)}$

**Slope matching conditions:**

$S'_j(x_{j+1}) = S'_{j+1}(x_{j+1})$ hence, $\mathbf{b_j + 2c_jh_j + 3d_jh_j^2 = b_{j+1}}$, j = 0,1,...,n−2,n−1

Add $b_n = S'_{n-1}(x_n)$, so j goes to n−1.

**Curvature matching conditions:**

$S''_j(x_{j+1}) = S''_{j+1}(x_{j+1})$ hence, $\mathbf{2c_j + 6d_jh_j = 2c_{j+1}}$, j = 0, 1... n−2, n−1

Add $c_n = \frac{1}{2}S''_{n-1}(x_n)$, so j goes to n−1.

### 5.1.3 Simplification of the Equations (1):

We know $a_j = y_j$, j = 0,1,...,n−1,n. We need bj, cj, and dj.

**Eliminate $d_j$:**

Curvature matching conditions ⇒

$$\mathbf{d_j = \frac{cj+1-cj}{3hj}}, \ \mathbf{j = 0,1,\ldots,n-1}$$

**Eliminate $b_j$:**

Right interpolation conditions and $d_j$ ⇒

$$b_j = (aj+1-aj)/hj - c_jh_j - d_jh_j^2$$

$$= (aj+1-aj)/hj - c_jh_j - ((c_{j+1}-c_j)/3h_j)h_j^2, \ j = 0,1,\ldots,n-1$$

$$\Rightarrow \mathbf{b_j = (a_{j+1}-a_j)/h_j - h_j(2c_j + c_{j+1})/3}, \ \mathbf{j = 0,1,\ldots,n-1}$$

### 5.1.4 Simplification of the Equations (2):

***System of linear equations for $c_j$:***

**Slope matching conditions⇒**

$$b_{j-1} + 2c_{j-1}h_{j-1} + 3d_{j-1}h^2_{j-1} = b_j, \quad j = 1, 2\ldots, n$$

$$\Rightarrow h_{j-1}c_{j-1} + 2(h_{j-1} + h_j)c_j + h_jcj+1$$

$$= \mathbf{3(a_{j+1}-a_j)/h_j - 3(a_j - a_{j-1})/h_{j+1}}, \ j = 1, 2\ldots n-1$$

*This is a **tridiagonal** linear system with*

$$\mathbf{c_0 = \frac{1}{2}S''_0(x_0) = 0 \text{ and } c_n = \frac{1}{2}S''_{n-1}(x_n) = 0}$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 & 0 \\ h0 & 2(h0+h1) & h1 & 0 & \cdots & 0 & 0 \\ 0 & h1 & 2(h1+h2) & h2 & \cdots & 0 & 0 \\ 0 & 0 & h2 & & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 2(hn-2+hn-1) & hn-1 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 1 \end{bmatrix} \times \begin{bmatrix} c0 \\ c1 \\ c2 \\ c3 \\ \vdots \\ cn-1 \\ cn \end{bmatrix} = \begin{bmatrix} 0 \\ \frac{3(a2-a1)}{h1} - \frac{3(a1-a0)}{h0} \\ \frac{3(a3-a2)}{h2} - \frac{3(a2-a1)}{h1} \\ \frac{3(a4-a3)}{h3} - \frac{3(a3-a2)}{h2} \\ \vdots \\ 0 \end{bmatrix}$$

## 5.2 Thomas algorithm for solving Tridiagonal system:

Let “$\mathbf{g_j}$” in the upper diagonal, $\mathbf{m_j}$ is the diagonal and $\mathbf{n_j}$ is the lower diagonal elements with $\mathbf{o_j}$ as the resultant matrix.

### Method:

The forward sweep consists of modifying the coefficients as follows, denoting the new modified coefficients with primes:

$$g(j)' = \begin{cases} \dfrac{g(j)}{m(j)}, & j = 0 \\ \dfrac{g(j)}{m(j) - n(j)c'(j-1)}, & j = 1, 2, 3 \ldots . n-1 \end{cases}$$

And

$$o(j)' = \begin{cases} \dfrac{o(j)}{m(j)}, & j = 0 \\ \dfrac{o(j) - n(j)o'(n-1)}{m(j) - n(j)g'(j-1)}, & j = 1, 2, 3 \ldots . n-1 \end{cases}$$

The solution is then obtained by back substitution:

$$\mathbf{c_{n-1} = o'_{n-1}}$$

$$\mathbf{c_j = o_j' - g_j' c'_{j+1}} \qquad ; \mathbf{j = n-2, n-3 \ldots\ldots 0}$$

From the values of **c** the values of coefficient **a, b** and **d** can be calculated.

With the 6-point interpolation having points at (0, 0), (50,100), (100,200), (150,230), (200,240) & (255,255) the curve will have shape shown in *figure 12*.

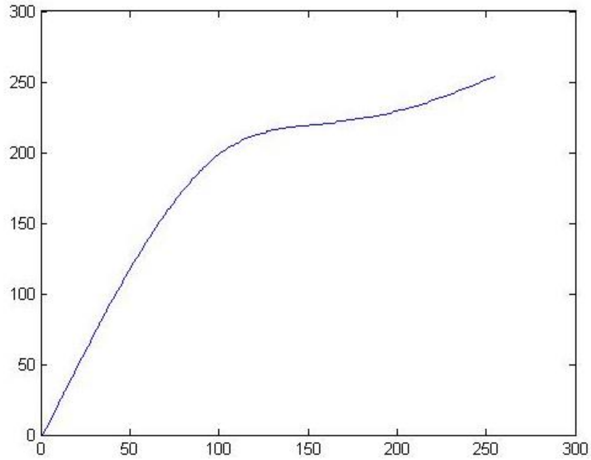
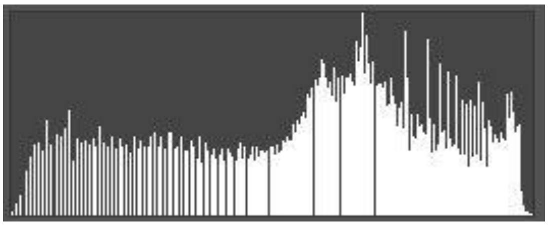
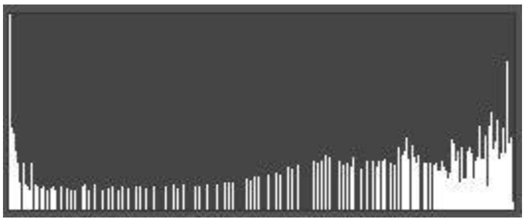
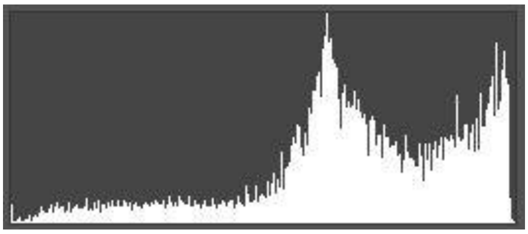
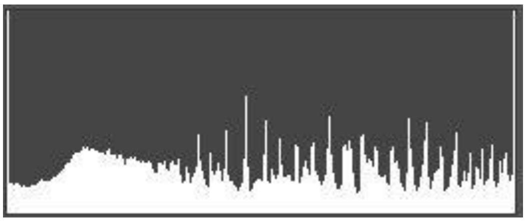
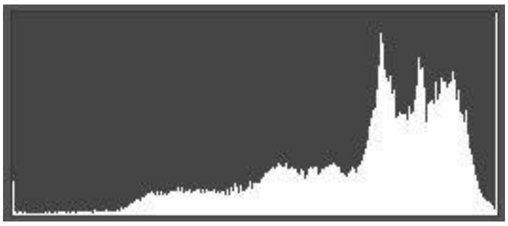



*Figure 12 - Interpolation using six points*

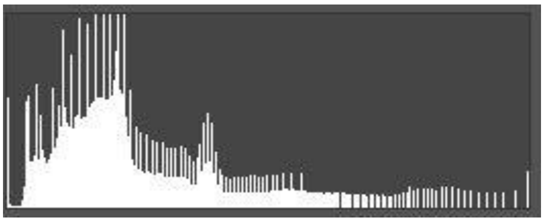
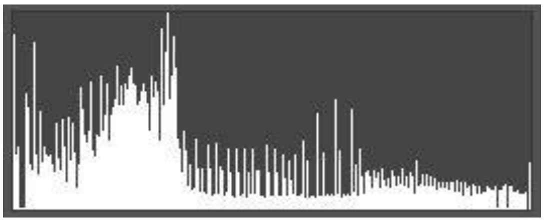
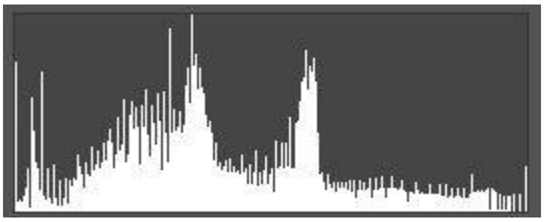
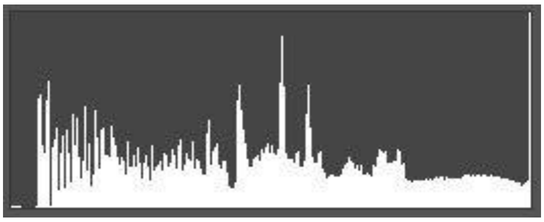
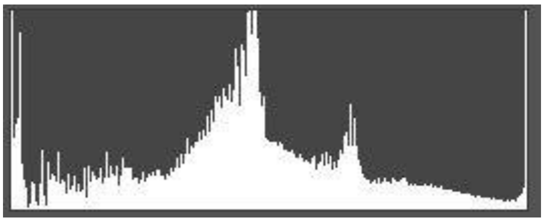
# 7. Results & Conclusion:

The algorithm has implemented on five images and the *output results* are shown below with their histogram. The enhanced images *conclude* the working of color conversion, Histogram equalization and spline interpolation algorithm. These algorithms are simple as well as robust, which can be easily implemented at low-level image enhancement technics.

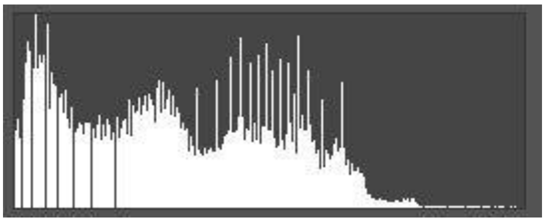
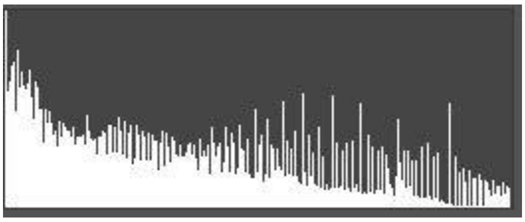
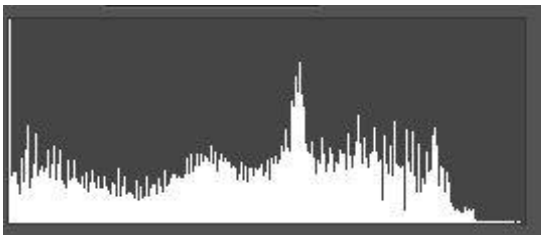
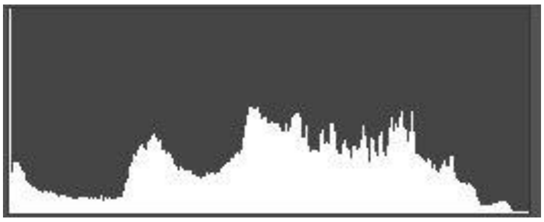
## 7.1 Image (1)

| | | |
|---|---|---|
|  |  | **Original Image**<br>**Resolution:512×480** |
|  |  | **Histogram equalization on HSV color Model** |
|  |  | **Cubic Spline interpolation on HSV color Model** |
|  |  | **Histogram equalization on LAB color Space** |
|  |  | **Cubic Spline Interpolation on LAB color Space** |

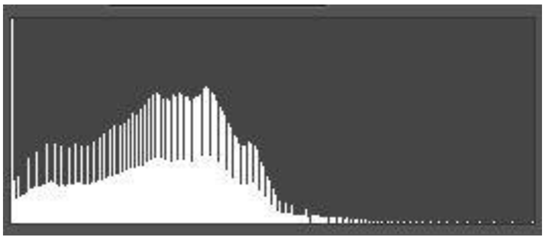
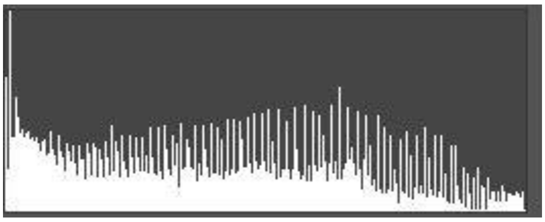
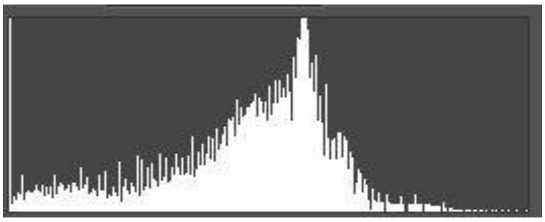
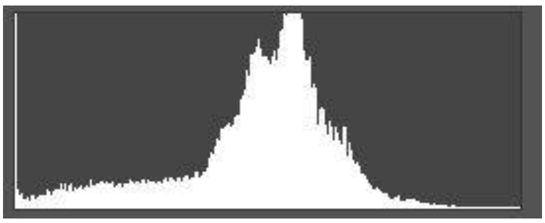
## 7.2 Image (2)

| | | |
|---|---|---|
|  |  | **Original Image**<br>**Resolution:500×362** |
|  |  | **Histogram equalization on HSV color Model** |
|  |  | **Cubic Spline interpolation on HSV color Model** |
|  |  | **Histogram equalization on LAB color Space** |
|  |  | **Cubic Spline Interpolation on LAB color Space** |

## 7.3 Image (3)

| | | |
|---|---|---|
|  |  | **Original Image**<br>**Resolution:512×480** |
|  |  | **Histogram equalization on HSV color Model** |
|  |  | **Cubic Spline interpolation on HSV color Model** |
|  |  | **Histogram equalization on LAB color Space** |
|  |  | **Cubic Spline Interpolation on LAB color Space** |

## 7.4 Image (4)

| | | |
|---|---|---|
|  |  | **Original Image**<br>**Resolution:786×576** |
|  |  | **Histogram equalization on HSV color Model** |
|  |  | **Cubic Spline interpolation on HSV color Model** |
|  |  | **Histogram equalization on LAB color Space** |
|  |  | **Cubic Spline Interpolation on LAB color Space** |

The results conclude the *correctness* and *efficiency* of the algorithm implemented. The poor images are enhanced to better quality of images because of increase in suitable amount of contrast in the image. Hence, the output images gives us a better visual stimulus and these images can be used further for higher level image processing (like edge detection, noise filtering, smoothening).

# 8. Future Scope:

In future the contrast enhancement technic can be integrated to the algorithm used in the camera for real-time contrast enhancement for better display. The histogram equalization can be improved to **adaptive histogram equalization** or **machine learning algorithm** integrated with histogram equalization/spline interpolation to have contrast enhancement by processor itself as of its requirements. Neural Networks, Genetic Algorithm and Machine Learning algorithm can be used to enhance contrast in the images captured by the satellite in real-time.

# 9. References


1. Rafael C. Gonzalez and Richard E. Woods, *Digital Image Processing*, Second Edition, Pearson Education [2002].
2. The CIE XYZ and xyY Color Spaces Douglas A. Kerr Issue 1 March 21, 2010
3. C. A. Bouman: Digital Image Processing - January 12, 2015
4. Charles Poynton : Color Space Conversion (1998)
5. Medical Image Contrast Enhancement Using Spline Concept: Data from the Osteoarthritis Initiative(Hong-Seng Gan1, Tian-Swee Tan1∗, Mohammed Rafiq Bin Abdul Kadir2, Ahmad Helmy Abdul Karim3, Khairil Amir Sayuti3, Liang-Xuan Wong4, and Weng-Kit Tham4)[2014]
6. General Construction of a Cubic Spline:Professor T. Arbogast Department of Mathematics The University of Texas at Austin.
8. SMOOTHING WITH CUBIC SPLINES by D.S.G. Pollock Queen Mary and Westfield College, The University of London.
9. https://en.wikipedia.org/wiki/Color_space
10. https://en.wikipedia.org/wiki/Color_model
11. https://en.wikipedia.org/wiki/RGB_color_model
12. https://en.wikipedia.org/wiki/HSL_and_HSV
13. https://en.wikipediahttps://en.wikipedia.org/wiki/Spline_interpolation
14. https://en.wikipedia.org/wiki/Tridiagonal_matrix_algorithm
15. .org/wiki/Lab_color_space
16. https://en.wikipedia.org/wiki/Histogram_equalization