

Optimized True-Color Image Processing

Majed O. Al-Dwairi, Ziad A. Alqadi, Amjad A. AbuJazar and Rushdi Abu Zneit

Faculty of Engineering technology, Al-Balqa Applied University

Abstract: Color image was defined as a 3 dimensional array and consisted of 3 two dimensional arrays, one for the red component, the second for the blue component and the third for green component. So it was needed to separate red, blue and green components in order to get the gray equivalent of the color image. The gray image then will be processed using one or more image processing techniques, after that the processed color image will be converted back to enhanced color image. Converting color image to gray and converting the processed gray image back to color image were the tasks that needed a lot of calculation and so a lot of processing time. Decreasing this time to the minimum is a focal task in order to achieve an optimized processing. HSI model of conversion was studied and a new R'G'I model also was studied and the experimental results shows that using R'G'I model speed up the conversion and thus minimize the time at least 8 times, thus R'G'I model can replace the HSI model in order to achieve efficient manipulation of true color image.

Key words: Color image • Direct conversion • Inverse conversion • Speedup • HSI model • R'G'I model • RGB color model • CMY color model • YUV color model • YIQ color model • HSI color model

INTRODUCTION

Most raw pictures are stored in a gray scale format. A gray scale is a color scale that ranges from black to white, with varying intermediate shades of gray. A commonly used gray scale for remote sensing image processing is a 256 shade gray scale, where a value of 0 represents a pure black color, the value of 255 represents pure white and each value in between represents a progressively darker shade of gray [1-3].

Objects in a gray tone display have a brightness value (or digital number), which represents the measured energy level of the item. Contrast refers to the difference in relative brightness between an item and its surroundings as seen in the image. A particular feature is easily detected in an image when contrast between an item and its background are high. However, when the contrast is low, an item might go undetected in an image.

A true color image is one for which the colors have been assigned to digital number values that represent the actual spectral range of the colors used in the image (blue features appear blue, green features appear green, red features appear red). A photograph is an example of a true color image [1-3].

Mathematically we can represent the color image by mean of using 3 dimensional matrixes, one dimension for the red color and the second for the blue color and the

third one for the green color. So we can take the color image as a 3 matrixes one for each color.

An RGB image sometimes referred to as a "true color" image and it can be stored as an m-by-n-by-3 data array that defines red, green and blue color components for each individual pixel. RGB images do not use a palette. The color of each pixel is determined by the combination of the red, green and blue intensities stored in each color plane at the pixel's location. Graphics file formats store RGB images as 24 bit images, where the red, green and blue components are 8 bits each. This yields a potential of 16 million colors. The precision with which a real-life image can be replicated has led to the nickname "true color image" [6-8].

We will notice that RGB color space is used for color representation on monitors, where (traditionally) three color guns (red, green and blue) are used to "bombard" the phosphorus pixels. Variation in intensities of RGB guns produces all the colors possible to be shown on a monitor. Likewise CMYK standard is meant for printing devices. HSI color space is more near to how humans perceive the colors. In image processing, using RGB space, we can tell what is the value of R, G and B not how much R is there in mixture of a color, e.g., (255, 255, 255) means color of the pixel is pure white and (0, 0, 0) means color of the pixel is pure black. When we want to detect magenta (with its variations) it is hard to do so.

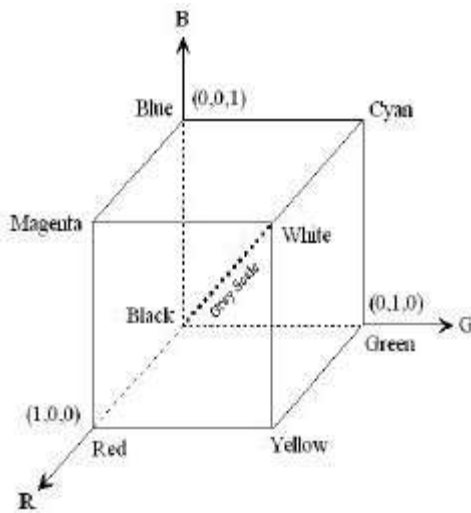


Fig. 1: RGB color cube

In HSI, Hue, Saturation and Intensity are better used to detect the color. Say if we want to deal in the brightness (which is more subjective) it is in "I". Moreover, 85% of the noise tends to be in "I" component. So it is better to process "I" for noise-removal rather than the three R = G = B components. Hue tells us purity of color while Saturation tells us brightness + admixture of color with other ones. That's why color detection operates on H + S components and the amount of "tolerance" one wants to have in color detection.

Usually each color image needs some preprocessing such as image enhancement, image sharpening, image stretching, edge detection, segmentation and so far operations.

To perform each of these operations using color image we need much efforts because we are dealing with an m-by-n-by-3 matrix and to minimize the efforts we have to convert the color image to gray image, process the gray image, then convert the processed gray image back to color image.

Color Models: RGB color model: The colors of the RGB model can be described as a triple (R, G and B). The RGB color space can be considered as a three-dimensional unit cube, in which each axis represents one of the primary colors, Fig. 1 colors are points inside the cube defined by its coordinates. The primary colors thus are red = (1,0,0), green = (0,1,0) and blue = (0,0,1). The secondary colors of RGB are cyan = (0,1,1), magenta = (1,0,1) and yellow = (1,1,0). The nature of the RGB color system is additive in the sense how adding colors makes the image brighter. Black is at the origin and white is at the corner where

R = G = B = 1. The gray scale extends from black to white along the line joining these two points. Thus a shade of gray can be described by (x,x,x) starting from black = (0,0,0) to white = (1,1,1).

The colors are often normalized as given in (1). This normalization guarantees that r + g + b = 1 [9-12]:

$$r = \frac{R}{(R+B+G)}, g = \frac{G}{(R+B+G)}, b = \frac{B}{(R+B+G)} \quad (1)$$

CMY Color Model: The CMY color model is closely related to the RGB model. Its primary colors are C (cyan), M (magenta) and Y (yellow). I.e., the secondary colors of RGB are the primary colors of CMY and vice versa. The RGB to CMY conversion can be performed by:

$$\begin{aligned} C &= 1-R \\ M &= 1-G \\ Y &= 1-B \end{aligned} \quad (2)$$

The scale of C, M and Y also equals to unit: C, M, Y [0, 1]. The CMY color system is used in offset printing in a subtractive way, contrary to the additive nature of RGB. A pixel of color Cyan, for example, reflects all the RGB other colors but red. A pixel with the color of magenta, on the other hand, reflects all other RGB colors but green. Now, if we mix cyan and magenta, we get blue, rather than white like in the additive color system [7,13-15].

YUV Color Model: The basic idea in the YUV color model is to separate the color information apart from the brightness information. The components of YUV are:

$$\begin{aligned} Y &= 0.3 \cdot R + 0.6 \cdot G + 0.1 \cdot B \\ U &= B-Y \\ V &= R-Y \end{aligned} \quad (3)$$

Y represents the luminance of the image, while U,V consists of the color information, i.e., chrominance. The luminance component can be considered as a gray-scale version of the RGB image. The advantages of YUV compared to RGB are:

- The brightness information is separated from the color information
- The correlations between the color components are reduced
- Most of the information is collected to the Y component, while the information content in the U and V is less.

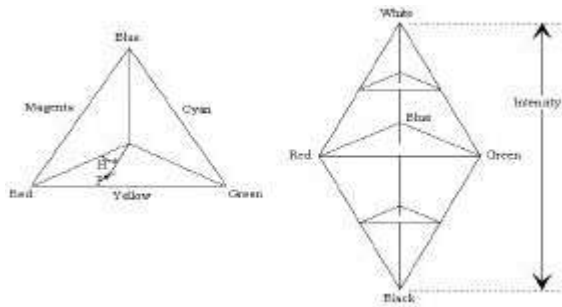


Fig. 2: HSI color triangle (left); HSI color solid (right)

The latter two properties are beneficial in image compression. This is because the correlations between the different color components are reduced, thus each of the components can be compressed separately. Besides that, more bits can be allocated to the Y component than to U and V. The YUV color system is adopted in the JPEG image compression standard [7, 13-15].

YIQ Color Model: YIQ is a slightly different version of YUV. It is mainly used in North American television systems. Here Y is the luminance component, just as in YUV. I and Q correspond to U and V of YUV color systems. The RGB to YIQ conversion can be calculated by [7, 13-15]:

$$\begin{aligned} Y &= 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B \\ I &= 0.596 \cdot R - 0.275 \cdot G - 0.321 \cdot B \\ Q &= 0.212 \cdot R - 0.523 \cdot G + 0.311 \cdot B \end{aligned} \quad (4)$$

The YIQ color model can also be described corresponding to YUV:

$$\begin{aligned} Y &= 0.3 \cdot R + 0.6 \cdot G + 0.1 \cdot B \\ I &= 0.74 \cdot V - 0.27 \cdot U \\ Q &= 0.48 \cdot V + 0.41 \cdot U \end{aligned} \quad (5)$$

HSI Color Model: The HSI model consists of hue (H), saturation (S) and intensity (I). Intensity corresponds to the luminance component (Y) of the YUV and YIQ models. Hue is an attribute associated with the dominant wavelength in a mixture of light waves, i.e., the dominant color as perceived by an observer. Saturation refers to relative purity of the amount of white light mixed with hue. The advantages of HSI are:

- The intensity is separated from the color information (the same holds for the YUV and YIQ models though)
- The hue and saturation components are intimately related to the way in which human beings perceive color [7].

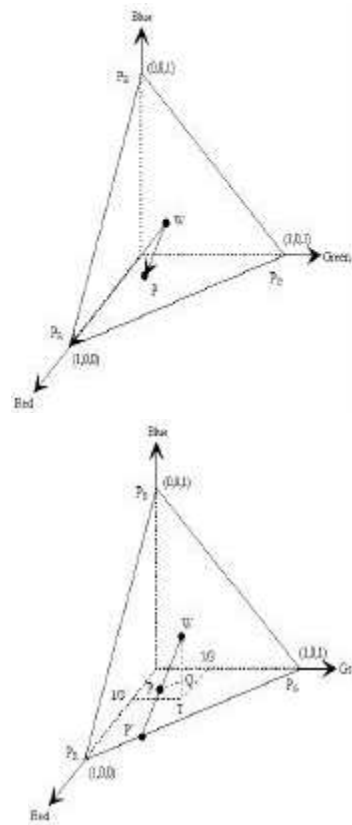


Fig. 3: Details of the HSI color triangle needed to obtain expressions for hue and saturation

Whereas RGB can be described by a three-dimensional cube, the HSI model can be described by the color triangle shown in Fig. 2. All colors lie inside the triangle whose vertices are defined by the three initial colors. Let us draw a vector from the central point of the triangle to the color point P. Then Hue (H) is the angle of the vector with respect to the red axis. For example 0° indicates red color, 60° yellow, 120° green and so on. Saturation (S) is the degree to which the color is undiluted by white and is proportional to the distance to the center of the triangle.

HSI can be obtained from the RGB colors as follows. Intensity of HSI model is defined:

$$I = \frac{1}{3} \cdot (R + G + B) \quad (6)$$

Let $W = (1/3, 1/3, 1/3)$ be the central point of the triangle, Hue is then defined as the angle between the line segments WP_R and WP . In vector form as shown in Fig. 3, this is the angle between the vectors $(p_R - w)$ and $(p - w)$. Here $(p - w) = (r - 1/3, g - 1/3, b - 1/3)$ and $(p_R - w) = (2/3, -1/3, -1/3)$. The inner product of two vectors are

calculated, $(x) \cdot (y) = \|x\| \|y\| \cos \alpha$ where α denotes the angle between the vectors. The following equations thus hold for $0^\circ \leq H \leq 180^\circ$:

$$(p-w) \cdot (p_R-w) = \|p-w\| \|p_R-w\| \cos H \quad (7)$$

The left part of the equation is:

$$\begin{aligned} (p-w) \cdot (p_R-w) &= \frac{2}{3} \left(r - \frac{1}{3} \right) - \frac{1}{3} \left(g - \frac{1}{3} \right) - \frac{1}{3} \left(b - \frac{1}{3} \right) \\ &= \frac{2r-g-b}{3} \\ &= \frac{2R-G-B}{3(R+G+B)} \end{aligned} \quad (8)$$

The right part evaluates as follows:

$$\begin{aligned} \|p-w\| &= \sqrt{\left(r - \frac{1}{3}\right)^2 + \left(g - \frac{1}{3}\right)^2 + \left(b - \frac{1}{3}\right)^2} \\ &= \sqrt{\frac{3(R^2+G^2+B^2) - (R+G+B)^2}{3(R+G+B)^2}} \end{aligned} \quad (9)$$

$$\|p_R-w\| = \sqrt{\left(\frac{2}{3}\right)^2 + \left(-\frac{1}{3}\right)^2 + \left(-\frac{1}{3}\right)^2} = \sqrt{\frac{2}{3}} \quad (10)$$

Now we can determine H:

$$\begin{aligned} H &= \cos^{-1} \frac{(p-w) \cdot (p_R-w)}{\|p-w\| \|p_R-w\|} \\ &= \cos^{-1} \left[\frac{2(R-G-B)}{3(R+B+G)} \sqrt{\frac{9(R+G+B)^2}{6(R+G+B) - 2(R+G+B)^2}} \right] \\ &= \cos^{-1} \frac{\frac{1}{2}(R-G) + \frac{1}{2}(R-B)}{\sqrt{(R-G)^2 + (R-B)(G-B)}} \end{aligned} \quad (11)$$

Equation 11 yields values of H in the interval $0^\circ \leq H \leq 180^\circ$. If $B > G$, then H has to be greater than 180° . So, whenever $B > G$, we simply let $H = 360^\circ - H$. Finally, H is normalized to the scale [0, 1]:

$$H' = \frac{H}{360^\circ} \quad (12)$$

The next step is to derive an expression for saturation S in terms of a set of RGB values. It is defined as the relative distance between the points P and W in respect to the distance between P' and W:

$$S = \frac{|WP|}{|WP'|} \quad (13)$$

Here P' is obtained by extending the line WP until it intersects the nearest side of the triangle, Fig. 3. Let T be the projection of P onto the rg plane, parallel to the b axis and let Q be the projection of P onto WT, parallel to the rg plane [16-17]. The equation then evaluates to:

$$\begin{aligned} S &= \frac{|WP|}{|WP'|} = \frac{|WQ|}{|WT|} = \frac{|WT| - |QT|}{|WT|} = 1 \\ &\quad - \frac{|QT|}{|WT|} = 1 - \frac{b}{\frac{1}{3}} = 1 - 3b \end{aligned} \quad (14)$$

The same examination can be made for whatever side of the triangle that is closest to the point P. Thus S can be obtained for any point lying on the HSI triangle as follows:

$$\begin{aligned} S &= 1 - \min(3r, 3g, 3b) \\ &= 1 - \min \left[\frac{3R}{(R+G+B)}, \frac{3G}{(R+G+B)}, \frac{3B}{(R+G+B)} \right] \\ &= 1 - \frac{3}{(R+G+B)} \min(R, G, B) \end{aligned} \quad (15)$$

The RGB to HSI conversion can be summarized as follows:

$$\begin{aligned} H &= \frac{1}{360^\circ} \cdot \cos^{-1} \left[\frac{\frac{1}{2} \cdot [(R-G) + (R-B)]}{\sqrt{(R-G)^2 + (R-B)(G-B)}} \right], \\ &\quad \text{if } B > G \\ H &= 1 - \frac{1}{360^\circ} \cdot \cos^{-1} \left[\frac{\frac{1}{2} \cdot [(R-G) + (R-B)]}{\sqrt{(R-G)^2 + (R-B)(G-B)}} \right], \\ &\quad \text{otherwise} \\ S &= 1 - \frac{3}{(R+G+B)} \cdot \min(R, G, B) \\ I &= \frac{1}{3} \cdot (R+G+B) \end{aligned} \quad (16)$$

RGB to HSI Conversion and Vice Versa: To convert an RGB image to HSI image model and back to RGB model we apply the following sequence:

- Perform direct conversion by applying the following steps:
- Get the red, blue and green components from the RGB image:

$$r = \frac{R}{R+G+B}, g = \frac{G}{R+G+B}, b = \frac{B}{R+G+B}$$

- Calculate the H, S and I components by applying the following:

$$h = \cos^{-1} \left\{ \frac{0.5 \cdot [(r-g) + (r-b)]}{[(r-g)^2 + (r-b)(g-b)]^{1/2}} \right\}$$

$$h \in [0, \pi] \text{ for } b \leq g$$

$$h = 2\pi - \cos^{-1} \left\{ \frac{0.5 \cdot [(r-g) + (r-b)]}{[(r-g)^2 + (r-b)(g-b)]^{1/2}} \right\}$$

$$h \in [\pi, 2\pi] \text{ for } b > g$$

$$S = 1 - 3 \cdot \min(r, g, b); s \in [0, 1]$$

$$i = (R + G + B) / (3 \cdot 255); i \in [0, 1]$$

- Process the I component:
- Apply inverse conversion using the following:

$$h = H \cdot \pi / 180; s = S / 100; i = I / 255$$

$$x = i \cdot (1 - s)$$

$$y = i \cdot \left[1 + \frac{s \cdot \cos(h)}{\cos(\pi/3 - h)} \right]$$

$$z = 3i - (x + y)$$

when $h < 2\pi/3$, $b = x$; $r = y$ and $g = z$

when $2\pi/3 \leq h < 4\pi/3$, $h = h - 2\pi/3$ and $r = x$; $g = y$ and $b = z$

when $4\pi/3 \leq h < 2\pi$, $h = h - 4\pi/3$ and $g = x$; $b = y$ and $r = z$

Calculated Example: Suppose we have a true-color pixel with the values 100 for the red, 150 for the green and 200 for the blue color:

- Find the components:

$$r = R / (R + G + B) = 0.222; g = G / (R + G + B) = 0.333; b = B / (R + G + B) = 0.444$$

- Here $b > g$ so:

$$h = 2\pi - \cos^{-1} \left\{ \frac{0.5 \cdot [(r-g) + (r-b)]}{[(r-g)^2 + (r-b)(g-b)]^{1/2}} \right\}$$

$$= 1.167\pi$$

$$S = 1 - 3 \cdot \min(r, g, b) = 0.333$$

- Find H, S and I:

$$H = h \times 180 / \pi = 210$$

$$S = s \times 100 = 33.3$$

$$I = (R + G + B) / 3 = 150$$

- No processing (I does not change)
- Apply inverse conversion

$$h = H \cdot \pi / 180 = 7\pi/6; s = S / 100 = 0.333;$$

$$I = I / 225 = 0.588$$

When $2\pi/3 \leq h < 4\pi/3$, we use following formulae:

$$h = h - 2\pi/3 = \pi/2$$

$$r = s = i \cdot (1 - s) = 0.392$$

$$g = y = i \cdot \left[1 + \frac{s \cdot \cos(h)}{\cos(\pi/3 - h)} \right] = 0.588$$

$$b = z = 3i - (x + Y)$$

$$R = 255 \cdot r = 100$$

$$G = 255 \cdot g = 150$$

$$B = 255 \cdot b = 200$$

Proposed R'G'I model: The proposed R'G'I model [4, 5] can be performed using the following sequence of operations:

- Get the R, G, I components from the true-color image
- Apply direct conversion to obtain R', G' and I using the following formulas:

$$R' = R \cdot 256 / (R + G + B); G' = G \cdot 256 / (R + G + B);$$

$$i = (R + G + B) / 3$$

- Process I
- Apply inverse conversion to get the new processed true-color image using the following formulas:

$$R_{new} = 3 \cdot R' \cdot I / 256; G_{new} = 3 \cdot G' \cdot I / 256;$$

$$B_{new} = 3 \cdot (256 - R' - G') \cdot I / 25$$

To prove the correctness of this model let us take a true color pixel (100,150,200):

$$\bullet R = 100; G = 150; B = 200$$

$$\bullet R' = 100 \cdot 256 / 450 = 56.88$$

$$G' = 150 \cdot 256 / 450 = 85.33$$

$$I = 450 / 3 = 150$$

- No processing (I does not change).
- Apply inverse conversion:

Rnew = 3*56.88*150/256 = 100
 Gnew = 3*85.33*150/256 = 150
 Bnew = 3*(256-56.88-85.33)*150/256 = 200

The following mat lab code was implemented and we get c = 0, which means that the original true color image and the converted one were identical:

clear all;

```
J = imread('football.jpg'); J = double(J);
R = J(:,:,1); G = J(:,:,2); B = J(:,:,3)
```

Direct conversion (%)

```
Rd = (R*256)/(R+G+B)
Gd = (G*256)/(R+G+B)
I = (R+G+B)/3
```

Inverse conversion (%)

```
R1 = (3*Rd.*I)/256; G1 = (3*Gd.*I)/256
B1 = (3*(256-Rd-Gd).*I)/256
J1 = uint8(I); J1(:,:,1) = R1; J1(:,:,2) = G1;
J1(:,:,3) = B1
c = 0
```

```
R = uint8(R); G = uint8(G); B = uint8(B); R1 =
uint8(R1); G1 = uint8(G1); B1 = uint8(B1);
```

for rr = 1:256

for j = 1:320

if R(rr,j) ~ = R1(rr,j)

c = c+1;

end

if G(rr,j) ~ = G1(rr,j)

c = c+1;

end

if B(rr,j) ~ = B1(rr,j)

c = c+1;

end end end

c

MATERIALS AND METHODS

True color image was implemented using mat lab. A mat lab code was written, this code was used to find the processing time needed for both direct and inverse conversion using HSI model using different color images with different sizes. The same thing was done for R'G'I model.

RESULTS

The following mat lab code was implemented using RGB-HSI-RGB model, the code was executed several times using different true color images with different resolutions.

For each execution the time needed for direct and inverse conversion was calculated as shown in Table 1:

```
F = imread('saturn.png'); F = im2double(F)
r = F(:,:,1); g = F(:,:,2); b = F(:,:,3)
th = acos((0.5*((r-g)+(r-b)))/((sqrt((r-g).^2+(r-b).*(g-
b)))+eps))
H = th; H(b>g) = 2*pi-H(b>g); H = H/(2*pi);
S = 1-3.*(min(min(r,g),b))./(r+g+b+eps);
I = (r+g+b)/3
His = cat(3,H,S,I)
HE = H*2*pi; HE = histeq(HE); HE = HE/(2*pi); SE =
histeq(S)
IE = histeq(I)
```

The same true color images were used to implement RGB-R'G'I-RGB using the following mat lab code and for each execution the time needed for direct and inverse conversion was calculated as shown in Table 1:

```
J = imread('saturn.png')
J = double(J); R = J(:,:,1); G = J(:,:,2); B = J(:,:,3)
Rd = (R*256)/(R+G+B)
Gd = (G*256)/(R+G+B)
I = (R+G+B)/3
Rd = double(Rd)
Gd = Double(Gd)
R1 = (3*Rd.*I)/256
G1 = (3*Gd.*I)/256
B1 = (3*(256-Rd-Gd).*I)/256
J1 = uint8(I); J1(:,:,1) = R1; J1(:,:,2) = G1; J1(:,:,3) = B1
```

From Table 1 we can obtain the time needed to manipulate each pixel using HSI and R'G'I models as show in Table 2.

Table 1: Comparisons between HSI and R'G'I models

Color Image	Size in pixels	Total time (sec)						Speedup in times
		HSI dir. Conv.	HSI inv. Conv.	HSI total	R'G'I dir. Conv	R'G'I inv. Conv.	R'G'I	
Football.jpg	256×320×3	0.094	0.031	0.125	0.015	0.015	0.030	4.1600
Peppers.png	384×512×3	0.188	0.094	0.282	0.016	0.016	0.032	8.8125
Hestain.png	227×303×3	0.078	0.032	0.110	0.015	0.015	0.030	3.6660
Greens.jpg	300 ×500×3	0.078	0.079	0.157	0.015	0.015	0.030	5.2300
Fabric.png	480×640×3	0.157	0.125	0.282	0.031	0.032	0.063	4.4700
Saturn.png	1500×1200×3	0.829	0.671	1.500	0.313	0.125	0.438	3.4200

Table 2: Manipulation time needed for each true color pixel

Color Image	Size in pixels	HSI total time (sec)	HSI time for each pixel (µ sec)	R'G'I total time (µ sec)	R'G'I time for each pixel (µ sec)	Speedup in times
football.jpg	245760	0.125	0.508	0.030	0.122	4.1600
peppers.png	589824	0.282	0.478	0.032	0.054	8.8125
hestain.png	206343	0.110	0.533	0.030	0.145	3.6660
Greens.jpg	300000	0.157	0.523	0.030	0.100	5.2300
Fabric.png	921600	0.282	0.305	0.063	0.068	4.4700
Saturn.png	5400000	1.500	0.277	0.438	0.081	3.4200

DISCUSSION

From the results shown in Table 1 and 2 we can see that using R'G'I model leads to better true color image processing by decreasing the time needed to handle direct and inverse conversions by at least 3 times and for some images for 8 times.

From the results also we can see that time needed to perform direct conversion and times needed to perform inverse conversion using R'G'I model is more less than the times needed for HSI model, which means getting an optimized true color image processing and the R'G'I model can replace HSI model which is widely used in color image processing.

CONCLUSION

The proposed R'G'I model was implemented and tested for accuracy. The experimental results shows that using R'G'I model leads to optimized true color image processing, thus the HSI model can be replaced by R'G'I model for better utilization.

REFERENCES

- Burkhardt, H. and B. Neumann, 1998. ECCV '98. Lecture Notes Comput. Sci., 1406: 610-622.
- Rojo, M.G., G.B. García, C.P. Mateos, J.G. García and M.C. Vicente, 2006. Critical comparison of 31 commercially available digital slide systems in pathology. Int. J. Surg. Pathol., 14: 285-305. <http://www.ncbi.nlm.nih.gov/pubmed/17041192>

- Kimmel, R., 1999. Demosaicing: Image reconstruction from color CCD samples image processing. IEEE Trans., 8: 1221-1228. DOI: 10.1109/83.784434.
- Akram Mustafa and Ziad AlQadi, 2009. Color image reconstruction using a new model. J. Comput. Sci., 5: 250-159.
- Waheeb, A. and Ziad AlQadi, 2009. Gray image reconstruction. Eur. J. Sci. Res., 27: 167-173.
- Li, Y., D. Lu, X. Lu and J. Liu, 2004. Interactive color image segmentation by region growing combines with image enhancement based on Bezier model. Proceeding of the 3rd International Conference on Image and Graphics, Dec. 18-20, IEEE Xplore Press, USA., pp: 96-99
- Foley, J.D., A. van Dam, S.K. Feiner and J.F. Hughes, 1990. Computer Graphics, Principles and Practice. 2nd Edn., Addison-Wesley, Reading, ISBN: 0-201- 12110-7: 1174.
- Kang, H.R., 1993. Resolution conversion of bitmap, images. US Patent 5270836. <http://www.patentstorm.us/patents/5270836.html>.
- Waheeb, A., A. Moustafa and Z. Alqadi, 2009. Gray image reconstruction. Eur. J. Sci. Res., 27: 167-173. http://www.eurojournals.com/ejsr_27_2_01.pdf
- Gao, M., P. Bridgman and S. Kumar, 2003. Computer aided prostate cancer diagnosis using image enhancement and JPEG2000. Proceeding of the SPIE International Conference on Applications of Digital Image Processing, Aug. 5-8, San Diego CA., pp: 323-334. <http://cat.inist.fr/?aModele=afficheN&cpsidt=15883017>.

11. Li, Y., D. Lu, X. Lu and J. Liu, 2004. Interactive color image segmentation by region growing combines with image enhancement based on Bezier model. Proceedings of the 3rd International Conference on Image and Graphics, Dec. 18-20, IEEE Xplore Press, USA., pp: 96-99. DOI: 10.1109/ICIG.2004.89.
12. Begelrnan, G., E. Gur, E. Rivlin, M. Rudzsky and Z. Zalevsky, 2004. Cell nuclei segmentation using fuzzy logic engine. Proceeding of the IEEE Intern Conference on Image Processing, Oct. 24-27, IEEE Xplore Press, USA., pp: 2937-2940. DOI:10.1109/ICIG.2004.89.
13. Hu, M.P. and X.Y. Ding, 2004. Automated cell nucleus segmentation using improved snake. Proceeding of the Intern Conference on Image Processing, Oct. 24-27, IEEE Xplore Press, USA., pp: 2737-2740. DOI: 10.1109/ICIP.2004.1421670.
14. Wyszecki, G. and W.S. Stiles, 2000. Color Science: Concepts and Methods. Quantitive Data and Formulae. 2nd Edn., Wiley, ISBN: 10: 0471399183, pp: 968.
15. Turk, M. and A. Pentland, 1991. Eigenfaces for recognition. J. Cognit. Neurosci., 3: 71-86.
16. Sanjay Kr. Singh, D.S. Chauhan, Mayank Vatsa and Richa Singh, 2003. A robust skin color based face detection algorithm, department of computer science and engineering institute of engineering and technology Jaipur -222002, India. Tamkang J. Sci. Eng., 6: 227-234.
17. Liang Wang, Tieniu Tanand and Weiming Hu, 2002. Face tracking using motion-guided dynamic template matching, national laboratory of pattern recognition. Proceeding of the 5th Asian Conference on Computer Vision, Jan. 23-25, Institute of Automation, Melbourne, Australia, pp: 1-6.