International Journal of Knowledge Discovery in Bioinformatics
Volume 8 • Issue 1 • January-June 2018

Estimation of Fractal Dimension in Different Color Model

Sumitra Kisan, Veer Surendra Sai University of Technology, Burla, India
Sarojananda Mishra, Indira Gandhi Institute of Technology, Sarang, Dhenkanal, India
Ajay Chawda, Veer Surendra Sai University of Technology, Burla, India
Sanjay Nayak, Veer Surendra Sai University of Technology, Burla, India

ABSTRACT

This article describes how the term fractal dimension (FD) plays a vital role in fractal geometry. It is a degree that distinguishes the complexity and the irregularity of fractals, denoting the amount of space filled up. There are many procedures to evaluate the dimension for fractal surfaces, like box count, differential box count, and the improved differential box count method. These methods are basically used for grey scale images. The authors' objective in this article is to estimate the fractal dimension of color images using different color models. The authors have proposed a novel method for the estimation in CMY and HSV color spaces. In order to achieve the result, they performed test operation by taking number of color images in RGB color space. The authors have presented their experimental results and discussed the issues that characterize the approach. At the end, the authors have concluded the article with the analysis of calculated FDs for images with different color space.

KEYWORDS

Box Count, CMY, DBC, Fractal Dimension, Fractal Geometry, HSV, Improved DBC, RGB

INTRODUCTION

Mandelbrot in 1982 has invented fractal theory that provides both mathematical and descriptive model for several apparently complex structures present in nature (B. B. Mandelbrot, 1982). Irregular shapes for example; clouds, mountains and coastlines are not simply defined by traditional geometry (Euclidean geometry). They often possess a significant invariance with the changes of magnification. This nature of self-similarity is a crucial eminence of fractal in nature. Usually it is used to estimate fractal dimension (FD). It provides several mathematical models for complex real-world objects. Fractal analysis has excessive impact in digital image analyses. It is extended by many more concepts applicable to a broader class of fractals. Fractal geometry is one of the best areas for doing research. As this area is vast, many researchers have worked on different problems and given their contribution for estimating fractal dimension. After the term fractal geometry has been introduced, several researchers shared their knowledge on this field and because of their effortless job various methodologies have been introduced to calculate fractal dimension. In 1986 Gangepain and Roques

DOI: 10.4018/IJKDB.2018010106

Carmes stated a method called reticular cell counting technique (Gangepain, & Roques-Carmes, 1986) and this method is enhanced by Voss in 1986 (Ford & Roberts, 1998) discussed about the method of probability. Pentland suggested the Fourier power spectrum of image intensity surface for fractal dimension estimation (Pentland, 1984). Keller et al. also proposed Reticular cell counting method and this one is the improved version of the previous reticular cell counting (1989). In 1986, Sarkar and Chaudhuri suggested a well-organized methodology to estimate the fractal dimension known as the Differential Box Counting (DBC) (Sarker & Chaudhuri, 1994). Instead of using the process of cell counting method to count the number of boxes, they used minimum and maximum grays levels of the corresponding grid in the image for calculation known as Differential Box Count Method (DBC). Jian, Qia and Caixin jointly enounced the improved Differential Box Counting or improved DBC (2009). In the improved DBC the drawbacks of DBC, Box height calculation, Box number calculation, partition of image intensity surface was analyzed and improved. The mean and standard deviation of intensity of a block was used for calculation of box height. The minimum gray level was used in the box height calculation rather than gray level 0. The image intensity partition occurs resulting in zero distance of neighboring blocks.

Terminologies

- **Fractal:** A fractal is infinitely complex shape that is self-similar through the different scales. Self- Similar structures such as fern leaf, in nature are examples of fractals. In mathematics, complex equations and formulas form fractals perfectly blending mathematics with art. Fractals that depend on complex numbers are Mandelbrot and Julia sets. Similarly, Iterated Function system fractals are Koch snowflake and Sierpinski triangle. Mathematically, fractals describe many geographical features as mountains and coastlines in the best possible way. Fractal uses are also extended to study of human organs like brain. When cartographers were measuring Britain's coastline, they faced an anomaly. The more detailed map they used resulted in a larger coastline. Accidentally, the self-similar property of fractal was discovered. Another important feature of fractals is non-integer dimension;
- **Fractal Dimension:** Fractals are objects with non- integer dimensions. Some natural occurrences are described using a non-integer dimension or fractional dimension in a better way. A fractal has a dimension between one and two, or two and three depending on the area it fills with its convolutions. A flatter topology results in Fractal dimension closer to 2. Similarly, an area with high hills and mountains would approach to a fractal dimension of 3. The Euclidean geometry deals with dimensions in whole numbers while Fractal geometry is constituted in terms of non-integer dimension or fractal dimension. Fractal dimension (FD) can be demarcated as the ratio that provides a statistical index of roughness relating how the form varies with respect to scale by which it is measured. It can also be considered as the filling capacity of space for a pattern. The fractal dimension may not be same as the topological dimension and does not have to be an integer as well. The fractal dimension can be estimated using the theory of self-similarity. FD for a bounded set S in Euclidian n-shape can be defined as the equation:

$$D = \frac{\log(Nr)}{\log(1/r)}$$

Here N_r represents the least number of discrete copies of S with the scale *r*:

- **Box- Counting Method:** There are several methods to analyse fractals and calculate fractal dimension such as Box-Counting method, Lacunarity analysis, Mass method, Multi-fractal analysis. In comparison to the above listed methods, Box Counting is the most widely used method. As it is useful in fractal analysis of linear and non-linear images and irregularities are easily detected with the use of Box Counting method.

The FD calculation using Box-Count, or The Box-Counting Dimension is:

$$\mathrm{D} = \frac{\log\left(Nr\right)}{\log r} \tag{1}$$

where:

D = Fractal Dimension
Nr = Number of Boxes of side length r
r = Scale of the Box

The Box Counting Dimension is also known as Minkowski Dimension. To calculate this dimension for a fractal, a fractal is placed on a grid that is evenly spaced, and number of boxes required to cover the set are calculated. The box-counting dimension is calculated by seeing how the number changes as we make the grid finer by applying a box-counting algorithm. The slope of best fit line of the log vs log graph with N_r and r as variables gives the Fractal Dimension of the image:

- **Self-Similarity:** In mathematical point of view, a self-similar entity is exactly or almost similar to a chunk of itself (i.e. if we divide the entire object with different scale then each smaller individual is same as whole). A quotidian example of self-similarity in nature is fern-leaf. The magnified visualization of a fern-leaf, resembles similar structure viewed as a whole. The generation of fractals mathematically is because of self-similar attribute, leading to formation of fractal by increasing the scale. The human organs like kidney, pancreas, liver is constructed on fractal rules of self-similarity. The self-similarity feature of human body helps in medical analysis;
- **Grayscale:** The grayscale images are also known as black and white images. Grayscale images store only intensity information of a single value. The intensity value 0 corresponds to black color, while 1 represents white. Less memory is required to store image information of grayscale images. The intensity is stored in the range of 0-255 in a 8 bit integer. Color images can be converted to grayscale as the color images also contain gray level information. Grayscale images are better suited to environments where memory storage is an issue. Less information in each pixel of a gray image helps in achieving the requirement. If the intensity values in a grayscale image are well spaced, they can be comparable to human grayscale perception. Grayscale images play an important role in image processing industry;
- **Color:** A color image includes color information for each pixel. Each color image consists of three color channels represented as co-ordinates in a color space. The extra information contained in color image requires more memory for storage. An image in color model is stored in various spaces such as RGB, HSV and CMY. All these spaces have different methods for storing the intensity information. The dissimilarity from grayscale storage is that a color image has three to four bytes of information regarding 3 co-ordinates and an opacity level, but in grayscale only one byte is required for storage of intensity value. The requirement of color images increases in fields of face recognition, medical analysis where minute changes in image information produces outputs with vast differences. In such cases, color images provide more accurate information in comparison to grayscale images. But more information adds to memory storage overhead and slow computation leading to an increased run time of algorithms for analysis of color images.

The various color spaces are appropriate for different conditions depending on the data required in a particular algorithm as input. Though the information stored is different in respective color models, the image viewed will always be the same in any color space.

In this research work we are focusing on the different color spaces like CMY and HSV. Our proposed methodologies are applied to these models for FD estimation.It is structured as follow: in the next section of our paper we introduce about the color models; Section 3 discusses the related works. The proposed methodologies are mentioned in Section 4. The Section 5 gives the simulation and result. Finally, we are concluding in the section 6.

Different Color Spaces

The color spaces provide a standard approach to specify a certain color, through 3D coordinate system definition and a subspace that holds all constructible colors within a particular model (Ford & Roberts, 1998). The color space contains the description of range of colours that are available for display. Any color which is specified by a model will resemble to a particular point within the subspace it defines. It is also known that each color space is concerned with specific hardware. It is important to note that different color spaces have their own importance for different applications. We can take example as certain equipment, which have some limiting factors that indicate color space type and the size that can be used. A number of color spaces possess linearity, i.e. if we change 10 units in stimulus, then that will yield the same change in the sensitivity wherever it is used. Generally, in computer graphics, many color spaces are not linear. Some color spaces are to circumnavigate within themselves and the required colors are created comparatively easily.

Following are the common computer related color spaces.

RGB (Red, Green, Blue)

RGB is tri-chromatic theory-based color model. This color model is additive as the three light beams (R, G, B) are added together. The intensity of the color depends on the intensity of each color components. Intensity value of zero gives black color and the maximum intensity gives white color. RGB is often found in systems which use CRT to display images. It is devise dependent and easy to implement. RGB is very common to all computer system as well as television, video etc. Figure 1 and Figure 2 shows the schematic of RGB color cube with primary and secondary colors.

Figure 1. Schematic of RGB color cube showing primary and secondary colors

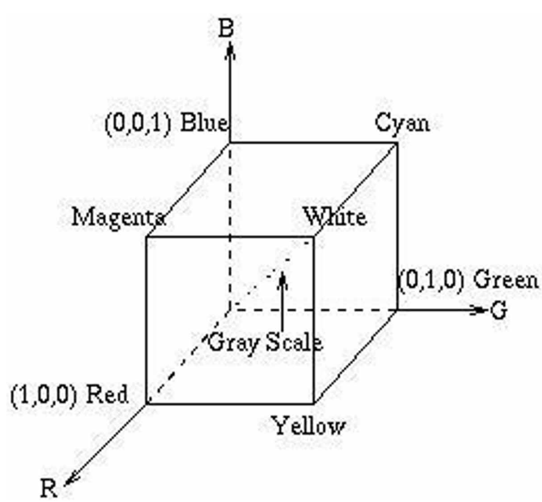
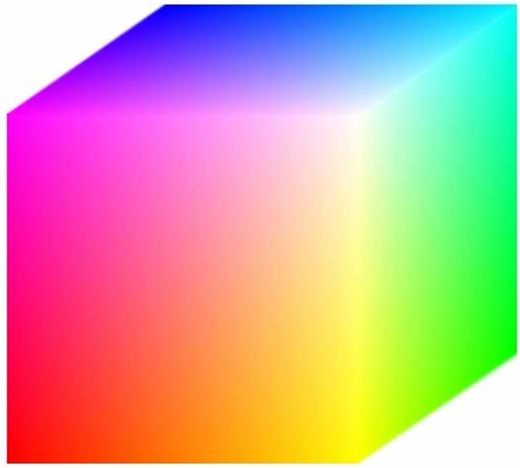


Figure 2. The RGB color cube



The specific intensity levels of red, green and blue in the RGB model describe the color. The maximum intensity value for each color is 255 which is based on a binary number of 32 bits, divided into four parts of 8 bits each. As the color information is divided into four parts of 8 bits, each part corresponds to a byte of information. The first, second and third byte represent the R, G, B intensity levels while the fourth byte is used for storing the opacity of color. The representation of hue in RGB model is specified by one color with full intensity, the second color with a variable intensity, and the remaining color with a zero intensity. For example, Orange is represented with intensities of Red (255), Green (128), Blue (0). The various combinations of RGB produce the additive colors depending on the composition of mixture of intensity values. Television and computer monitors with cathode ray tubes produce different colors by firing electrons of red, green and blue colors with variable intensity is a common example of mixing of spectral light. The significance of RGB color model lies in the fact that it closely relates to the human perception of colors. RGB model is device dependent that is the colors displayed on screen depend on the hardware for display.

CMY(K) (Cyan Magenta Yellow [Black])

This color model is opposite to the RGB, which is additive one. CMY is subtractive based color model where white is the natural color and full combination of all components is the black color. This model is basically used in hard copy output and printing purpose. The objective for using the fourth component, black is to improve the density range and color gamut availability. CMY(K) is easily implemented but proper conversion from RGB to CMY(K) is difficult. CMY(K) is device dependent in nature and nonlinear by means of visual perception (see Figures 3 and 4).

We can derive CMY components from the RGB values as the following matrix:

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{2}$$

The RGB model constitutes the additive colors, while the CMY depicts the subtractive colors. The formula clearly represents that cyan(C) does not contain the red color space as C = 1 - R. Similarly,

Figure 3. CMY color cube

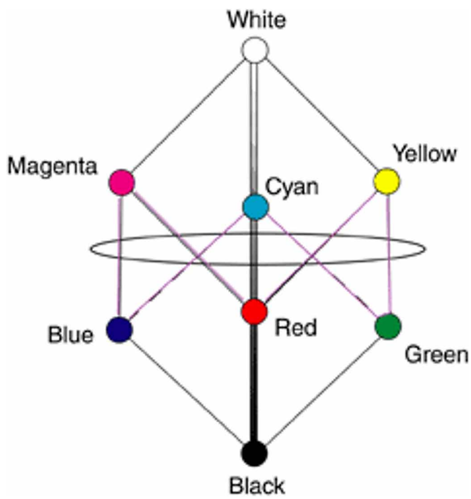
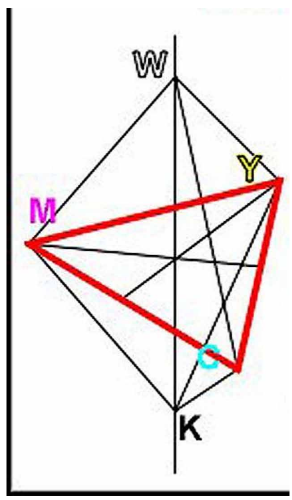


Figure 4. Schematic of CMY color cube

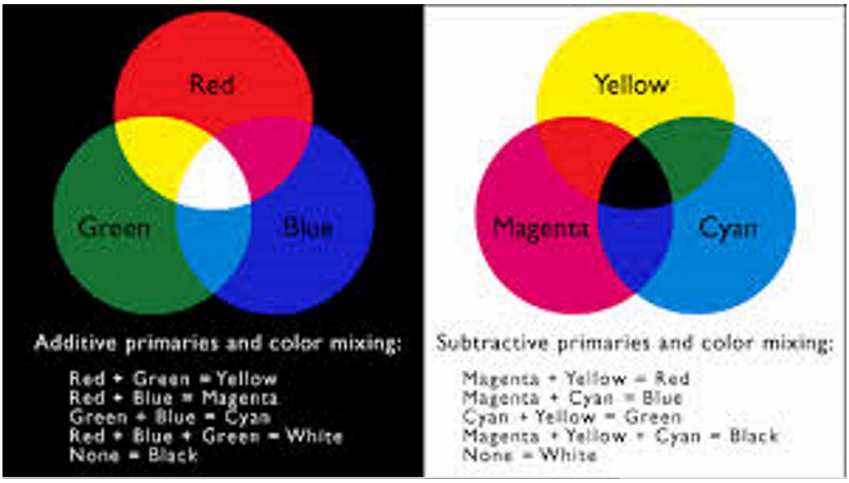


the Magenta is devoid of green color, M =1 – G. The Yellow space does not contain the blue color space, Y = 1 – B. Black color is created by combining the primary colors cyan, magenta and yellow in equal proportions, but the resultant black is not comparable to true black. The reason for this is that colored inks contain minor impurities. Graphic designers prefer the CMY model in comparison to RGB model, the reason being that a photo designed in the RGB space is not same when it is printed on paper and less number of available colors in RGB. For improving the print quality and reducing patterns, different angles are set for each color screen. The hardware devices like monitors and scanner work on RGB principle, while printer is based on the CMY model (see Figure 5).

HSV (Hue Saturation Value)

Hue is considered as an attribute that defines a pure color like pure red, orange or yellow. The term saturation provides a measure to which a pure color is diluted with the white light. The third

Figure 5. Conversion of RGB to CMY color space



component, value or we can say the brightness, is a descriptive one which is practically can cot be measured. There are many other names for HSV, for example, HSL (lightness), HIS (intensity), HCI (colorfulness) (see Figures 6 and 7).

The HSVs are obtained from RGB using the transformation:

$$H = \cos^{-1} \frac{(0.5*(R-G)+(R-B))}{(((R-G)^2+(R-B)(G-B))^{0.5})} \tag{3}$$

$$S = 1 - \left(\frac{3}{(R+G+B)}\right)*a \tag{4}$$

where:

$$a = \min(R, G, B)$$

$$V = \frac{R+G+B}{3} \tag{5}$$

Hue refers to the resemblance of its pure color. The shades, tones, and tints of a particular color have same hue. Hue is described by number that specifies position of corresponding color on color wheel in a fraction lying between 0 and 1. For Red, value is 0. Yellow and Green have 1/6 and 1/3 respectively. Saturation describes the white content of the color. An example of fully saturated color is pure red with a saturation of 1. Value represents the darkness of a color. A color with value 0 is totally black. The advantages of using hue are easy identification of tone relationships around the color circle and easy generation of shades, tints. HSV model is well suited to description of color images. The disadvantage of HSV color space is that there is always increase in total intensity of desaturated colors.

Figure 6. Schematic of HSV color hexagon

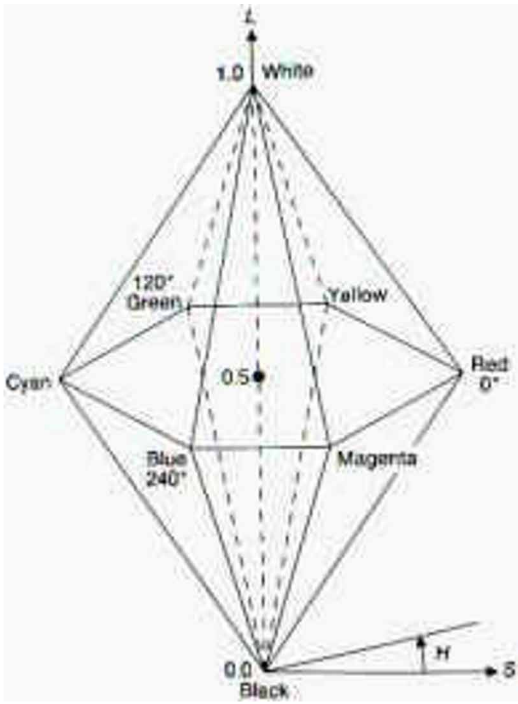
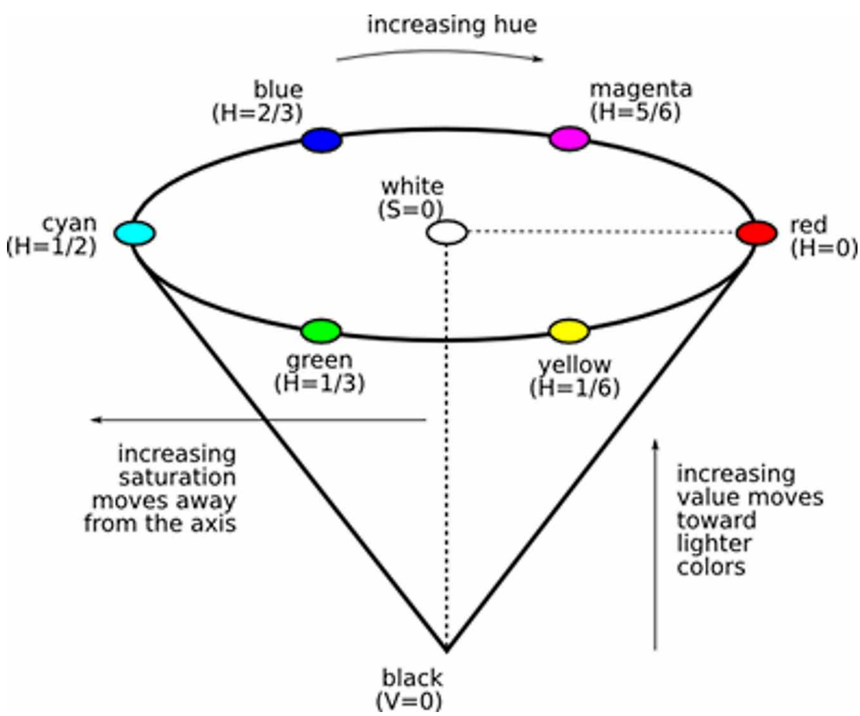


Figure 7. Relation between RGB and HSV color space



The three-color spaces discussed above are different representations of a color image. In RGB model the information is stored in intensity values of Red, Green and Blue. In CMY model the information consists of cyan, magenta, and yellow values that are subtractive of red, green and blue intensities respectively. Similarly, in HSV model the Hue, Saturation and Value levels of an image are stored as information to process the image.

RELATED WORKS

There is no such work has been done in other color spaces. In this section, we are discussing some recent works, done on RGB model.

M Ivanovici and N Richard introduced a probabilistic method for FD estimation in color images (Ivanovici & Richard, 2011). They considered a probability matrix P (m, L) which is nothing but the probability of having m points in a hyper-cube of size L, centered in an arbitrary point of S. then the normalized matrix is defined as:

$$\sum_{m=1}^{N} P(m,L) = 1, \forall L$$

where N is the number of pixels included in a box of size L.

Accordingly, the total number of boxes that contain *m* points is given by:

$$\frac{M}{m} P(m,L)$$

Therefore:

$$N(L) = \sum_{m=1}^{N} \left(\frac{1}{m}\right) P(m,L) \alpha L^{-D} \tag{6}$$

where *D* is the fractal dimension.

Ivanovici and Richard not only proposed an efficient method for estimation of FD but also provided method for generation of color fractal images. The various responses of different weighing functions for line fitting method and all the sizes of a hyper-cube are not well suited for analysis of an image were the issues mainly concentrated upon. An approach for lacunarity computation was proposed to extend the classical probabilistic method for estimation of fractal dimension of color images. The complexity of color fractal images in RGB space was reflected in the computed lacunarity. The most efficient color model for analysis is the RGB space because a large number of images are not stored with adequate information and are uncalibrated. The RGB space is not a constraint and other color models like CMY, HSV can also be used for fractal analysis.

The major drawback in their methodology is that the regression lines are no longer linear, which makes the estimation of the regression line slope a more complicate issue.

There was lack of application of box counting method for estimation of fractal dimension to large and color images due to large time complexities of algorithms. Nikolaidis proposed the Box Merging Method which is fast, easy to implement, and easily expandable to variable dimensions (2011).

The image is partitioned into s partitions. An E-dimensional grid is created:

$$s_{max} = L_{min} = min\left(L_x, L_y, L_y\right) \tag{7}$$

The iterations are recorded for the values of $s = s_{max} / 2^v$, $v = v_{max}, v_{max-1}, \ldots, 1$ where:

$$v_{max} = \log_2\left(L_{min}\right) \tag{8}$$

is the maximum number of iterations if the edges ϵ of the box are halved in each iteration:

$$\epsilon = \frac{L}{s} \tag{9}$$

The partitioning in x axis is determined by:

$$t_x = \frac{x}{\epsilon_x} + x\frac{s}{L_x} \tag{10}$$

where x is coordinate of pixel in box and t_x is coordinate of the box.

The same normalization is applied for all the axes. There are various types of normalization as described by (Panda, & Jana, 2015; Panda, Gupta, & Jana, 2015; Panda, & Jana, 2016; Panda, & Jana, 2016a; Mishra, & Panda, 2017; Meher, Pande, & Panda, 2017). If we know the (x, y) coordinates of the pixel we can find the box to which the pixel belongs.

The box merging algorithm, based on BCM, is fast and helps in easy implementation from one to variable dimensions. With decreased time complexity, the presented algorithm emulates the characteristics of the box counting method producing the same log n − log s table. The tested color images measured fractal dimension between 2 and 5, validating the algorithm. The major drawback of this method is that it requires extra memory for storage of partition table.

Another novel method was proposed by X Zhao and X Wang in 2016 well known as maximum color distance (MCD) method (2016). In their procedure, they had calculated the Euclidian Distance (ED) between all possible pixels in the image. If two points are P and Q for example, then the ED for this pair of point is defined as:

$$\left|PQ\right| = \sqrt{\left(r1 - r2\right)^2 + \left(g1 - g2\right)^2 + \left(b1 - b2\right)^2} \tag{11}$$

In the same way, the fit line of log(*Nr*) versus log(1/*r*) is plotted. The estimated fractal dimension is equal to the slope of the fitted straight line. The MCD algorithm finds the Euclidean Distance between all the two points in a block and finds the maximum among them to find the n_r and N_r values. So apart from finding a block the distance is calculated between every two points in a block. This increases the complexity of the algorithm and makes it slow.

In this procedure, the main problem is the calculation overhead. We need much more calculation for computing the distance among all pair of points.

A method was proposed by Nayak and Mishra (2016) as an improvement to DBC method for estimation of fractal dimension in color space experimented only on RGB model. Therefore, to estimate roughness of color image, first roughness of each individual RGB component is estimated and corresponding smoothness is subtracted. After the smoothness is subtracted the fractional part is added to its corresponding RGB component. In the end, the smoothness is added to get the accurate roughness of the image.

The method was applied to 24-bit representations of color images in RGB space for extraction of roughness. The IDBC method detected the sharp variations in intensity values while estimating roughness in the RGB space.

All the above stated methods have been implemented only in the RGB color space. CMY and HSV though not a constraint to above methods, have never been implemented. Thus, their accuracy in different color models is not validated yet. The methods (N.N. Nikolaidis, I.N. Nikolaidis, & Tsouros, 2011; Zhao & Wang, 2016; Nayak & Mishra, 2016) are similar to box counting method as the image is first divided into blocks for estimation of FD. Method (Ivanovici & Richard, 2011) is probabilistic method principled on mass-dimension.

PROPOSED METHODOLOGIES

M x M sized RGB color image is taken for the experiment. Image is then divided by boxes of size l *l* l'. Fractal Dimension for this image is calculated using box-count method. In our proposed method first the R components, the G components and the B components are separately retrieved. Then R, G, B components are converted to C component, M component and Y component using Equation (2). Or H component, S component, V component using Equation (3), (4) and (5).

Box Height

We divide the image into boxes of size l *l *l' where height of each box is calculated above as:

$$l' = l * \left(Max / M \right)$$

Max is the maximum intensity of the image.

Box Number

The number of boxes present in the vertical direction is found out as:

$$n = ceil\left(\frac{Max}{l'} \right)$$

where Max = Maximum Intensity of particular block. The value of n differs for each block in the image as the intensity varies in each block.

The numbers of boxes in each partitioned block of image are calculated using the given formula:

$$n_r\left(i, j\right) = \begin{Bmatrix} 1 & Max = Min \\ ceil\left(\frac{Max}{l'} \right) & Max \neq Min \end{Bmatrix} \tag{12}$$

where Min = Minimum Intensity of particular block.

Total number of boxes:

$$N_r = \sum n_r \tag{13}$$

and Fractal Dimension is calculated using the best fit line of:

$$\log(N_r) \text{ vs } \log(r)$$

ALGORITHM SUMMARY OF CMY COLOR SPACE

1. Let an image size of M * M and the block size as $l * l'$
2. R_c, G_c, B_c are for storing the extracted values where, $R_c = [R_1, R_2, \ldots, R_n]$, $G_c = [G_1, G_2, \ldots, G_n]$, $B_c = [B_1, B_2, \ldots, B_n]$.
3. Combine all the RGB values in $RGB = [R_1G_1B_1, \ldots, R_nG_nB_n]$.
4. Convert RGB to CMY:

$$CMY = 1 - RGB$$

5. The whole image size is enclosed by the boxes of size l' where:

$$l' = \frac{l * CMY}{M}$$

6. Box Number is calculated using:

$$n_r(i,j) = \begin{Bmatrix} 1 & MI = MN \\ ceil\left(\frac{MI}{l'}\right) & MI \neq MN \end{Bmatrix}$$

7. $N_r = \sum n_r$
8. The slope of best fit line of $\log(N_r)$ vs log (r) is used for calculation of fractal dimension.

ALGORITHM SUMMARY OF HSV COLOR SPACE

1. Let an image size of M * M and the block size as $l * l'$
2. R_c, G_c, B_c for extracting the RGB values separately where R_c=[R_1,R_2,……R_n], G_c=[G_1,G_2,……G_n], B_c=[B_1,B_2,……B_n]
3. By combining all the RGB values RGB=[$R_1G_1B_1$, $R_2G_2B_2$,……RnGnBn]
4. Convert RGB to HSV:

$$H = \cos^{-1}\frac{(0.5*(R-G)+(R-B))}{(((R-G)^2+(R-B)(G-B))^{0.5})}$$

$$S = 1-\left(\frac{3}{\left(R+G+B\right)}\right)*a$$

where:

$$a = \min\left(R,G,B\right)$$

$$V = \frac{R+G+B}{3}$$

5. The whole image size is enclosed by the boxes of size $l^{'}$ where:

$$l^{'} = \frac{l*HSV}{M}$$

6. Box Number is calculated using:

$$n_r\left(i,j\right) = \begin{Bmatrix} 1 & MI = MN \\ ceil\left(\frac{MI}{l'}\right) & MI \neq MN \end{Bmatrix}$$

7. $N_r = \sum n_r$
8. The slope of best fit line of $\log\left(N_r\right)$ vs log (*r*) is used for calculation of fractal dimension.

SIMULATION AND RESULTS

Five RGB color sample images of size 256 x 256 are taken as input to our proposed procedure (see Figure 8). All RGB images are converted to CMY and HSV color spaces respectively and FD for each image has been estimated according to our methodology. F1 shows one sample RGB color image and corresponding CMY and HSV color images in an order. Similarly, F2, F3, F4 and F5 contain RGB image with their corresponding CMY and HSV converted images. Tables 1-3 and Figures 9-11 display the log(r) and log(N_r) values for five images in three different color spaces. Finally, Table 4 and Figure 12 show the FDs for RGB, CMY and HSV color images.

CONCLUSION

Box counting method is a prominent technique to calculate fractal dimension of images like color, gray scale, 1-D, 2-D and 3-D as well. We proposed novel method for estimating FD in CMY and HSV

Figure 8. Five color images (F1, F2, F3, F4, F5) in different color spaces

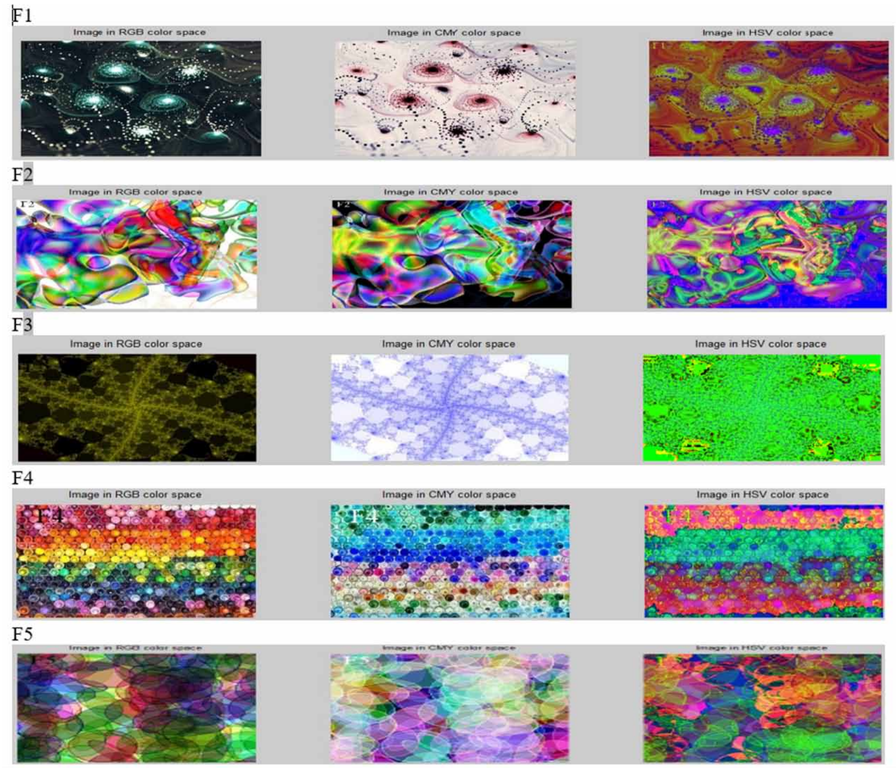


Table 1. log(r), log(N_r) in RGB color space

Figures	(log r) 1	(log N_r) 1	(log r) 2	(log N_r) 2	(log r) 3	(log N_r) 3	(log r) 4	(log N_r) 4	(log r) 5	(log N_r) 5
F1	0.602	1.505	0.903	2.283	1.204	3.103	1.505	3.868	1.806	4.591
F2	0.602	1.505	0.903	2.281	1.204	3.157	1.505	3.991	1.806	4.835
F3	0.602	1.462	0.903	2.199	1.204	2.987	1.505	3.732	1.806	4.492
F4	0.602	1.505	0.903	2.283	1.204	3.160	1.505	4.001	1.806	4.849
F5	0.602	1.491	0.903	2.243	1.204	3.023	1.505	3.827	1.806	4.654

Table 2. log(r), log(N_r) in CMY color space

Figures	(log r) 1	(log N_r) 1	(log r) 2	(log N_r) 2	(log r) 3	(log N_r) 3	(log r) 4	(log N_r) 4	(log r) 5	(log N_r) 5
F1	0.602	3.736	0.903	4.638	1.204	5.538	1.505	6.433	1.806	7.322
F2	0.602	3.732	0.903	4.621	1.204	5.463	1.505	6.278	1.806	7.088
F3	0.602	2.667	0.903	3.561	1.204	4.451	1.505	5.340	1.806	6.224
F4	0.602	3.724	0.903	4.599	1.204	5.473	1.505	6.335	1.806	7.155
F5	0.602	3.735	0.903	4.628	1.204	5.505	1.505	6.374	1.806	7.234

Table 3. log(r), log(N_r) in HSV color space

Figures	(log r) 1	(log N_r) 1	(log r) 2	(log N_r) 2	(log r) 3	(log N_r) 3	(log r) 4	(log N_r) 4	(log r) 5	(log N_r) 5
F1	0.602	4.882	0.903	5.742	1.204	6.595	1.505	7.448	1.806	8.304
F2	0.602	4.884	0.903	5.766	1.204	6.615	1.505	7.445	1.806	8.273
F3	0.602	4.775	0.903	5.588	1.204	6.403	1.505	7.223	1.806	8.043
F4	0.602	4.871	0.903	5.734	1.204	6.602	1.505	7.450	1.806	8.289
F5	0.602	4.878	0.903	5.759	1.204	6.611	1.505	7.452	1.806	8.285

Figure 9. log(r) versus log(N_r) in RGB color space

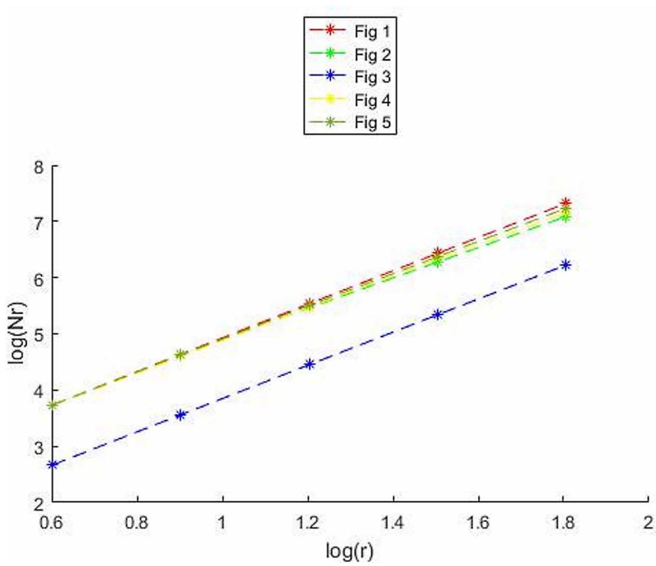


Figure 10. log(r) versus log(N_r) in CMY color space

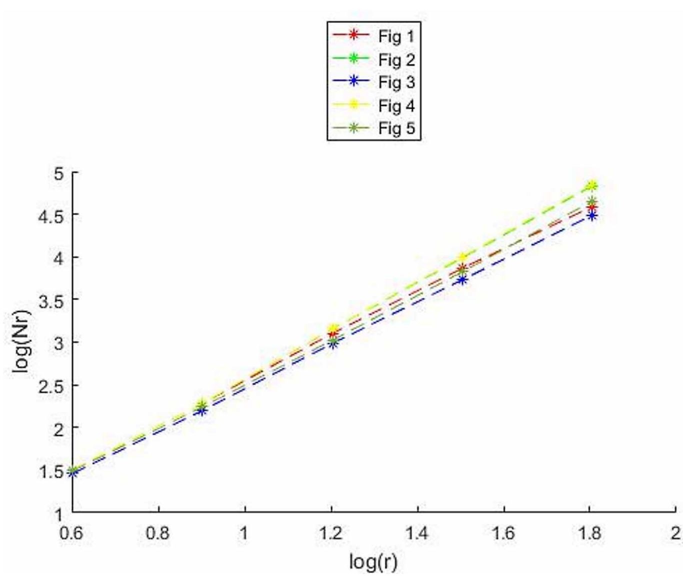


Figure 11. log(r) versus log(N_r) in HSV color space

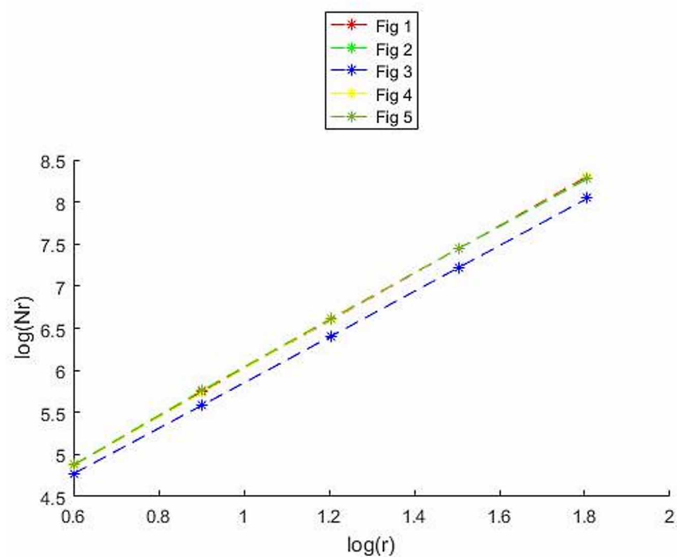
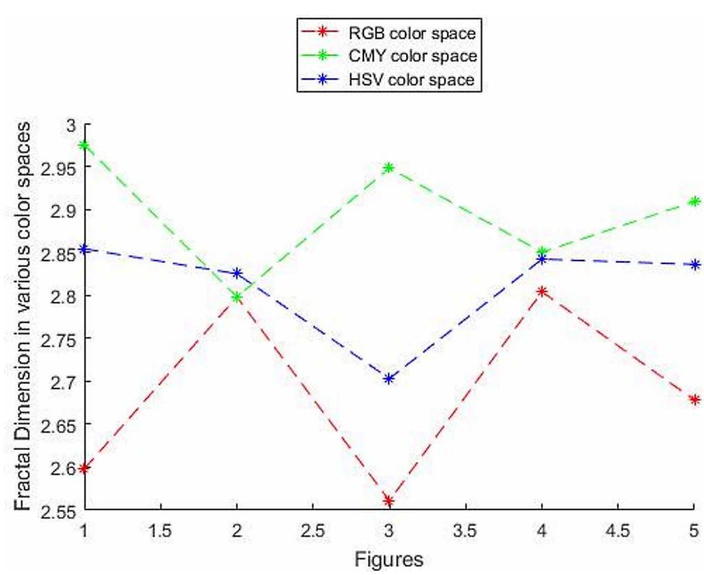


Table 4. Fractal dimension in various color spaces

Figures	FD in RGB Color Space	FD in CMY Color Space	FD in HSV Color Space
F1	2.598	2.975	2.854
F2	2.798	2.798	2.825
F3	2.561	2.948	2.703
F4	2.804	2.850	2.842
F5	2.678	2.909	2.836

Figure 12. FD estimation in various color spaces



color spaces. The proposed algorithm is extensively tested with number of sample images. It estimated FD for the images with less complexity that recently proposed MCD method. It is concluded from the experiments that there is slight difference in FDs of RGB color space images, CMY color space images and HSV color space images. This approach can also be extended with blocks of overlapping boxes.

ACKNOWLEDGMENT

This work is supported in part through are research grant from Veer Surendra Sai University of Technology (VSSUT), Burla, India.

REFERENCES

Fisher, Y., McGuire, M., Voss, R. F., Barnsley, M. F., Devaney, R. L., & Mandelbrot, B. B. (2012). *The science of fractal images*. Springer Science & Business Media.

Ford, A., & Roberts, A. (1998, August 11). *Colour Space Conversions*. London: Westminster University.

Gangepain, J., & Roques-Carmes, C. (1986). Fractal approach to two dimensional and three dimensional surface roughness. *Wear*, *109*(1-4), 119–126. doi:10.1016/0043-1648(86)90257-7

Hu, L., Zhang, Z., Dong, H., & Lin, K. (2014). Outlier Detection Using Neighborhood Radius Based on Fractal Dimension. In *Proceedings of the 9th International Conference on Computer Science & Education* (ICCSE 22-24), Vancouver, Canada.

Ivanovici, M., & Richard, N. (2011). Fractal Dimension of Color Fractal Images. *IEEE Transactions on Image Processing*, *20*(1), 227–235. doi:10.1109/TIP.2010.2059032 PMID:20643608

Keller, J., Crownover, R., & Chen, S. (1989, February). (1098). Texture description and segmentation through fractal geometry. *Computer Vision Graphics and Image Processing*, *45*(2), 150–160. doi:10.1016/0734-189X(89)90130-8

Khan, M. I., Jia, W., & Marsh, R. (2003). Fractal analysis of tumor in brain MR images. *Machine Vision and Applications Springer-Verlag*, *13*(5-6), 352–362. doi:10.1007/s00138-002-0087-9

Kisan, S., & Priyadarshini, M. (2015). Relative improved differential box-counting approach to compute fractal dimension of gray-scale images. *International Journal of Engineering Research and General Science, 3*(1).

Li, J., Du, Q., & Sun, C. (2009). An improved box-counting method for image fractal dimension estimation. *Journal on Pattern Recognition*, *42*(11), 2460-2469.

Liu, S. (2008). An improved DBC approach to compute fractal dimension of gray level image. In *Proceedings of the International Symposium on Information Science & Engineering*.

Mandelbrot, B. B. (1982). *Fractal Geometry of nature*. San Francisco, CA: Freeman.

Meher, S., Pande, S., & Panda, S. (2017). Time-aware task allocation for cloud computing environment. *International Journal of Knowledge Discovery in Bioinformatics*, *7*(1). doi:10.4018/IJKDB.2017010101

Mishra, S., & Panda, S. (2017). Efficient fault tolerant algorithms for internet distributed systems. *International Journal of Knowledge Discovery in Bioinformatics*, *7*(1), 71–90. doi:10.4018/IJKDB.2017010106

Nayak, S. R., & Mishra, J. (2016). *An improved method to estimate the fractal dimension of colour images*. Elsevier.

Nikolaidis, N. N., Nikolaidis, I. N., & Tsouros, C. C. (2011). A Variation of the Box-Counting Algorithm Applied to Colour Images. *Applied Mathematics and Computation*.

Nikolaoua, N., & Papamarkosb, N. (2002). Color image retrieval using a fractal signature extraction technique. *Engineering Applications of Artificial Intelligence*, *15*(1), 81–96. doi:10.1016/S0952-1976(02)00028-3

Panda, S., Gupta, I., & Jana, P. (2015). Allocation-aware task scheduling for heterogeneous multi-cloud systems. In *Proceedings of the 2nd International Symposium on Big Data and Cloud Computing, Procedia Computer Science* (pp. 176-184). Elsevier.

Panda, S., & Jana, P. (2015). Efficient task scheduling algorithms for heterogeneous multi-cloud environment. *The Journal of Supercomputing, Springer*, *71*(4), 1505–1533. doi:10.1007/s11227-014-1376-6

Panda, S., & Jana, P. (2016). Normalization-based task scheduling algorithms for heterogeneous multi-cloud environment. *Information Systems Frontiers.*

Panda, S., & Jana, P. (2016a). Uncertainty-based qos min-min algorithm for heterogeneous multi-cloud environment. *Arabian Journal for Science and Engineering*, *41*(8), 3003–3025. doi:10.1007/s13369-016-2069-7

Pentland, P. (1984). Fractal-based description of natural scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *PAMI-6*(6), 661–674. doi:10.1109/TPAMI.1984.4767591 PMID:22499648

Sarker, N., & Chaudhuri, B. B. (1994). An efficient differential box-counting approach to compute fractal dimension of image. *IEEE Transactions on Systems, Man, and Cybernetics*, *24*(1), 115–120. doi:10.1109/21.259692

Venkata Ramana Chary, R., Rajya Lakshmi, V., & Sunitha, K. V. N. (2012). Feature method extraction for color image similarity. Advanced Computing: An International Journal, 3(2).

Voss, R. (1986). Random fractals: Characterization and measurement. *Scaling Phenom. Disordered Syst*, *10*(1), 51–61.

Xie, W. (1997). Fractal-based analysis of time series data and features extraction. *Chinese Signal Processing Journal*, *13*, 98–104.

Yu, L., Zhang, D., Wang, K., & Yang, W. (2005). Coarse iris classification using box-counting to estimate fractal dimensions. *Pattern Recognition*, *38*(11), 1791–1798. doi:10.1016/j.patcog.2005.03.015

Zhao, X., & Wang, X. (2016). *Fractal Dimension estimation of RGB color images using Maximum Color Distance (MCD).*

Sumitra Kisan is working as Assistant Professor in the department of Computer Science & Engineering, Veer Surendra Sai University of Technology (VSSUT), Burla. She has completed her B.Tech in Computer Science & Engineering from Veer Surendra Sai University of Technology, Burla (formerly University College of Engineering, Burla) in 2007 and M.Tech from Indian School of Mines, Dhanbad in 2011. She has six years of experience in teaching and research. Her areas of research are Image Processing (Fractal Analysis, Image Segmentation, and steganography), Network Security and cryptography. She has published her research in many International and national journals as well as conferences.

Sarojananda Mishra is working as the Prof. & HOD in the Dept of Computer Science Engineering & Application of IGIT Sarang, Odisha, India. He has completed MCA from Sambalpur University, Odisha & M.Tech. from IIT Delhi. Subsequently He has completed his Ph.D. from Utkal University, Odisha. He has a lot of National & International Journals with having more than 20 years of teaching experience.

Ajay Chawda is a final year student at Veer Surendra Sai University of Technology pursuing his Bachelors in the field of Computer Science Engineering. He is currently working on Fractal analysis. His research interests include Data Mining, Machine Learning and Image processing.

Sanjay Nayak completed his Bachelor's Degree in 2017 in Computer Science and Engineering from Veer Surendra Sai University of Technology. His research interests include Artificial Intelligence, Computer Vision, and Image Processing. Currently, he is working in the area of fractal analysis.