ISSN: 2321-7782 (Online)
Volume 3, Issue 11, November 2015
International Journal of Advance Research in Computer Science and Management Studies
Research Article / Survey Paper / Case Study
Available online at: www.ijarcsms.com

RGB to Lab Transformation Using Image Segmentation

K. Kavi Niranjana[1]
Assistant Professor,
Department of MCA,
Sri Ramakrishna Institute of Technology,
Coimbatore, India.

M. Kalpana Devi[2]
Assistant Professor,
Department of BCA & M.Sc SS,
Sri Krishna Arts and Science College,
Coimbatore, India.

Abstract: This paper contains a new approach for image segmentation. The research presentation is Image Segmentation by applying color transformation method. The segmentation process includes a new mechanism for segmenting the elements of high – resolution images in order to improve the precision and reduce the computation time. Separating objects in an image is one of the most difficult image processing operations. The color transformation methodology is applied in this regard to this problem. This paper evaluates the proposed approach as Lab Color Transformation for image segmentation by comparing with K means clustering [2, 3, 11, 13]. The experimental results clarify the effectiveness of this approach to improve the segmentation quality in aspects of precision and computational time.

Keywords: Image Segmentation, K means, RGB Color space, Lab Color Space, Color Transformation.

I. INTRODUCTION

Image segmentation, the objective of which is to subdivide an image into several meaningful regions, is an important topic in image processing, computer vision and multimedia. The set of segmented regions not only corresponds to the underlying structure of the image, but is also related to the semantic meaning of the image. There are a number of methods to perform image segmentation based on different criteria and different properties of the images. Image segmentation [4, 5] is an effort to classify similar colors of image in the same group. The objective of the image segmentation is to extract the dominant colors. Image segmentation is very important to simplify an information extraction from images, such as color, texture, shape, and structure. The applications of image segmentation are diversely in many fields such as image compression, image retrieval, object detection, image enhancement, and medical image processing.

The goal of segmentation [4, 5] is to simplify and/or change the representation of an image to more meaningful and easier to analyze. It is typically used to locate objects and boundaries (lines, curves, etc.) in the images.

The result of image segmentation is a set of segments that collectively wrap the entire image, or a set of contours extracted from the image [6]. Each of the pixels in a region is similar with respect to some trait or computed property, such as color, intensity, or texture.

The applications of image segmentation are diversely in many fields to locate tumors and other pathologies, measure tissue volumes, Computer-guided surgery, Diagnosis, Treatment planning, image compression, image retrieval, object detection, image enhancement, and medical image processing [4,6], Study of anatomical structure in the Medical Imaging field and many other fields like Locate objects in satellite images (roads, forests, etc.), Face Recognition, Fingerprint Recognition, Traffic control Systems, Brake light detection, Machine vision, Agriculture Imaging-crop disease detection [6].

Several general-purpose algorithms and techniques have been developed for image segmentation [5]. There is no general solution to the image segmentation problem, so these techniques often have to be combined with domain knowledge in order to solve any image segmentation problem effectively [9].

The image segmentation is an effort to classify similar colors of image in the same group. The most popular method is Hierarchical clustering [12, 15]. It is widely applied for image segmentation. Many researchers used Gaussian Mixture Model with its variant Expectation Maximization [10]. This paper proposes a new approach for image segmentation that utilizes Color Space Transformation to produce the optimized result and compared with K means algorithm [3, 11, 13].

II. COLOR SPACES

A color model is an abstract mathematical model that describes the way that the colors can be represented as tuples of numbers, typically three or four values or *color components* (e.g. RGB and CMYK are color models) [8]. We also use "color model" to indicate a model or mechanism of color vision for explaining how color signals are processed from visual cones to ganglion cells. For simplicity, we call these models color mechanism models. The classical color mechanism models are Young - Helmholtz's tri-chromatic model and Hering's opponent-process model [17]. Though these two theories were initially thought to be at odds, it later came to be understood that the mechanisms responsible for color opponent receive signals from the three types of cones and process them at a more complex level.

A wide range of colors can be created by the primary colors of pigment (cyan (C), magenta (M), yellow (Y), and black (K)). Those colors then define a specific color space. To create a three-dimensional representation of a color space, we can assign the amount of magenta color to the representation's X axis, the amount of cyan to its Y axis, and the amount of yellow to its Z axis. The resulting 3-D space provides a unique position for every possible color that can be created by combining those three pigments.

However, this is not the only possible color space. For instance, when colors are displayed on a computer monitor, they are usually defined in the RGB (red, green and blue) color space. This is another way of making nearly the same colors (limited by the reproduction medium, such as the phosphor (CRT) or filters and backlight (LCD)), and red, green and blue can be considered as the X, Y and Z axes. Another way of making the same colors is to use their Hue (X axis), their Saturation (Y axis), and their brightness Value (Z axis). This is called the HSV color space. Many color spaces can be represented as three-dimensional (X, Y, Z) values in this manner, but some have more, or fewer dimensions, and some cannot be represented in this way at all [8].

When formally defining a color space, the usual reference standard is the CIELAB or CIEXYZ color spaces, which were specifically designed to encompass all colors the average human can see.

Adding a certain mapping function between the color model and a certain reference color space results in a definite "footprint" within the reference color space. This "footprint" is known as a gamut, and, in combination with the color model, defines a new color space. For example, Adobe RGB and Standard RGB (sRGB) are two different absolute color spaces, both based on the RGB model.

In the most generic sense of the definition above, color spaces can be defined without the use of a color model. These spaces, such as Pantone, are in effect a given set of names or numbers which are defined by the existence of a corresponding set of physical color swatches. This article focuses on the mathematical model concept.

Since "color space" is a more specific term for a certain combination of a color model plus a mapping function, the term "color space" tends to be used to also identify color models, since identifying a color space automatically identifies the associated color model. Informally, the two terms are often used interchangeably, though this is strictly incorrect. For example, although several specific color spaces are based on the RGB model, there is no such thing as the RGB color space.

Since any color space defines colors as a function of the absolute reference frame, color spaces, along with device profiling, allow reproducible representations of color, in both analogue and digital representations.

a) Conversion

Color space conversion is the translation of the representation of a color from one basis to another. This typically occurs in the context of converting an image that is represented in one color space to another color space, the goal being to make the translated image look as similar as possible to the original.

b) Density

The RGB color model is implemented in different ways, depending on the capabilities of the system used. By far the most common general-used incarnation as of 2006 is the 24-bit implementation, with 8 bits, or 256 discrete levels of color per channel. Any color space based on such a 24-bit RGB model is thus limited to a range of $256 \times 256 \times 256 \approx 16.7$ million colors. Some implementations use 16 bits per component for 48 bits total, resulting in the same gamut with a larger number of distinct colors. This is especially important when working with wide-gamut color spaces (where most of the more common colors are located relatively close together), or when a large number of digital filtering algorithms are used consecutively. The same principle applies for any color space based on the same color model, but implemented in different bit depths [7].

III. LAB COLOR SPACE

A *Lab* color space is a color-opponent space, with dimension *L* for lightness and *a* and *b* for the color-opponent dimensions. The initials *Lab* by themselves are somewhat ambiguous. The color spaces are related in purpose, but differ in implementation. Both spaces are derived from the "master" space CIE 1931 XYZ color space, which can predict which spectral power distributions will be perceived as the same color, but which is not particularly perceptually uniform. Strongly influenced by the Munsell color system, the intention of both “Lab” color spaces is to create a space which can be computed via simple formulas from the *XYZ* space, but is more perceptually uniform than *XYZ*. Perceptually uniform means that a change of the same amount in a color value should produce a change of about the same visual importance. When storing colors in limited precision values, this can improve the reproduction of tones. The lightness correlate in CIELAB is calculated using the cube root of the relative luminance.

The nonlinear relations for L^*, a^*, and b^* are intended to mimic the nonlinear response of the eye. Furthermore, uniform changes of components in the $L^*a^*b^*$ color space aim to correspond to uniform changes in perceived color, so the relative perceptual differences between any two colors in $L^*a^*b^*$ can be approximated by treating each color as a point in a three dimensional space (with three components: L^*, a^*, b^*) and taking the Euclidean distance between them.

Advantages of Lab

Unlike the RGB and CMYK color models, *Lab* color is designed to approximate human vision. It aspires to perceptual uniformity, and its *L* component closely matches human perception of lightness. It can thus be used to make accurate color balance corrections by modifying output curves in the *a* and *b* components, or to adjust the lightness contrast using the *L* component. In RGB or CMYK spaces, which model the output of physical devices rather than human visual perception, these transformations can only be done with the help of appropriate blend modes in the editing application.

Because *Lab* space is much larger than the gamut of computer displays, printers, or even human vision, a bitmap image represented as Lab requires more data per pixel to obtain the same precision as an RGB or CMYK bitmap. In the 1990s, when computer hardware and software was mostly limited to storing and manipulating 8 bit/channel bitmaps, converting an RGB image to Lab and back was a lossy operation. With 16 bit/channel support now common, this is no longer such a problem.

Additionally, many of the “colors” within Lab space fall outside the gamut of human vision, and are therefore purely imaginary; these “colors” cannot be reproduced in the physical world. Though color management software, such as that built in to image editing applications, will pick the closest in-gamut approximation, changing lightness, colorfulness, and sometimes hue in the process, author Dan Margulis claims that this access to imaginary colors is useful, going between several steps in the manipulation of a picture.

IV. RGB Color Space

An **RGB color space** is any additive color space based on the RGB color model. A particular RGB color space is defined by the three chromaticities of the red, green, and blue additive primaries, and can produce any chromaticity that is the triangle defined by those primary colors. The complete specification of an RGB color space also requires a white point chromaticity and a gamma correction curve. **RGB** is initialism for **R**ed, **G**reen, and **B**lue

RGB is a convenient color model for computer graphics because the human visual system works in a way that is similar — though not quite identical — to an RGB color space. The most commonly used RGB color spaces are sRGB and Adobe RGB (which has a significantly larger gamut). Adobe has recently developed another color space called Adobe Wide Gamut RGB, which is even larger, in detriment to gamut density [7].

As of 2007, sRGB is by far the most commonly used RGB color space, particularly in consumer grade digital cameras, HD video cameras, and computer monitors. HDTVs use a similar space, sharing the sRGB primaries, commonly called Rec. 709. sRGB is considered adequate for most consumer applications. Having all devices use the same color space is convenient in that an image does not need to be converted from one color space to another before being displayed. However, sRGB's limited gamut leaves out many highly saturated colors that can be produced by printers or in film, and thus is not ideal for some high quality applications. The wider gamut Adobe RGB is being built into more medium-grade digital cameras, and is favored by many professional graphic artists for its larger gamut.

a) Additive primary colors

To form a color with RGB, three colored light beams (one red, one green, and one blue) must be superimposed (for example by emission from a black screen, or by reflection from a white screen). Each of the three beams is called a component of that color, and each of them can have an arbitrary intensity, from fully off to fully on, in the mixture.

The RGB color model is additive in the sense that the three light beams are added together, and their light spectra add, wavelength for wavelength, to make the final color's spectrum.

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. When the intensities are different, the result is a colorized hue, more or less saturated depending on the difference of the strongest and weakest of the intensities of the primary colors employed.

When one of the components has the strongest intensity, the color is a hue near this primary color (reddish, greenish, or bluish), and when two components have the same strongest intensity, then the color is a hue of a secondary color (a shade of cyan, magenta or yellow). A secondary color is formed by the sum of two primary colors of equal intensity: cyan is green+blue, magenta is red+blue, and yellow is red+green. Every secondary color is the complement of one primary color; when a primary and its complementary secondary color are added together, the result is white: cyan complements red, magenta complements green, and yellow complements blue.

The RGB color model itself does not define what is meant by red, green, and blue colorimetrically, and so the results of mixing them are not specified as absolute, but relative to the primary colors. When the exact chromaticities of the red, green, and blue primaries are defined, the color model then becomes an absolute color space, such as sRGB or Adobe RGB; see RGB color spaces for more details.

b) Physical principles for the choice of red, green, and blue

A set of primary colors, such as the sRGB primaries, define a color triangle; only colors within this triangle can be reproduced by mixing the primary colors. Colors outside the color triangle are therefore shown here as gray. The primaries and the D65 white point of sRGB are shown.

The choice of primary colors is related to the physiology of the human eye; good primaries are stimuli that maximize the difference between the responses of the cone cells of the human retina to light of different wavelengths, and that thereby make a large color triangle.

The normal three kinds of light-sensitive photoreceptor cells in the human eye (cone cells) respond most to yellow (long wavelength or L), green (medium or M), and violet (short or S) light (peak wavelengths near 570 nm, 540 nm and 440 nm, respectively). The difference in the signals received from the three kinds allows the brain to differentiate a wide gamut of different colors, while being most sensitive (overall) to yellowish-green light and to differences between hues in the green-to-orange region.

As an example, suppose that light in the orange range of wavelengths (approximately 577 nm to 597 nm) enters the eye and strikes the retina. Light of these wavelengths would activate both the medium and long wavelength cones of the retina, but not equally the long-wavelength cells will respond more. The difference in the response can be detected by the brain and associated with the concept that the light is orange. In this sense, the orange appearance of objects is simply the result of light from the object entering our eye and stimulating the relevant kinds of cones simultaneously but to different degrees.

Use of the three primary colors is not sufficient to reproduce *all* colors; only colors within the color triangle defined by the chromaticities of the primaries can be reproduced by additive mixing of non-negative amounts of those colors of light.

V. K – MEANS CLUSTERING IN SEGMENTATION

The K-means algorithm is an iterative technique that is used to partition an image into *K* clusters. The basic algorithm is:

1. Pick *K* cluster centers, either randomly or based on some heuristic
2. Assign each pixel in the image to the cluster that minimizes the distance between the pixel and the cluster center
3. Re-compute the cluster centers by averaging all of the pixels in the cluster
4. Repeat steps 2 and 3 until convergence is attained (e.g. no pixels change clusters)

In this case, distance is the squared or absolute difference between a pixel and a cluster center. The difference is typically based on pixel color, intensity, texture, and location, or a weighted combination of these factors. K can be selected manually, randomly, or by a heuristic.

This algorithm is guaranteed to converge, but it may not return the optimal solution. The quality of the solution depends on the initial set of clusters and the value of K.

In statistics and machine learning, the k-means algorithm is a clustering algorithm [12, 15] to partition n objects into k clusters, where $k < n$. It is similar to the expectation-maximization algorithm for mixtures of Gaussians in that they both attempt to find the centers of natural clusters in the data [10]. The model requires that the object attributes correspond to elements of a vector space. The objective it tries to achieve is to minimize total intra-cluster variance, or, the squared error function. The k-means clustering [15] was invented in 1956. The most common form of the algorithm uses an iterative refinement heuristic known as Lloyd's algorithm. Lloyd's algorithm starts by partitioning the input points into k initial sets, either at random or using some heuristic data. It then calculates the mean point, or centroid, of each set. It constructs a new partition by associating each point with the closest centroid. Then the centroids are recalculated for the new clusters, and algorithm repeated by alternate application of these two steps until convergence, which is obtained when the points no longer switch clusters (or alternatively centroids are no longer changed). Lloyd's algorithm and k-means are often used synonymously, but in reality Lloyd's algorithm is a heuristic for solving the k-means problem, as with certain combinations of starting points and centroids, Lloyd's algorithm can in fact converge to the wrong answer. Other variations exist, but Lloyd's algorithm has remained popular, because it converges extremely quickly in practice. In terms of performance the algorithm is not guaranteed to return a global optimum. The quality of the final solution depends largely on the initial set of clusters, and may, in practice, be much poorer than the global optimum. Since the algorithm is extremely fast, a common method is to run the algorithm several times and return the best clustering [12, 15] found. A drawback of the k-means algorithm is that the number of clusters k is an input parameter. An inappropriate choice of k may yield poor results. The algorithm also assumes that the variance is an appropriate measure of cluster scatter.

VI. EXPERIMENTAL RESULTS

a) Experimental Results of K means

The K-means algorithm assigns each point to the cluster whose center (also called Centroid) is nearest. The center is the average of all the points in the cluster. This section briefly explains the basic theory of K-means clustering [3, 15]. Let A = $\{a_i \mid i = 1,\ldots f\}$ be attributes of f – dimensional vectors and X = $\{x_i \mid i = 1,\ldots, N\}$ be each data of A. The K – means clustering separates X into k partitions called clusters S = $\{s_i \mid i = 1,\ldots, k\}$ where MЄX is M_i = $\{m_{ij} \mid j = 1,\ldots, n(s_i)\}$ as members of s_i, where $n(s_i)$ is number of members for s_i. Each cluster has cluster center of $C = \{c_i \mid I = 1,\ldots,k\}$. K – means clustering algorithm can be described as follows:

1. Initiate its algorithm by generating random starting points of initial centroids *C*.
2. Calculate the distance d between *X* to cluster center *C*. Euclidean distance is commonly used to express the distance.
3. Separate x_i for i = 1…N into S in which it has minimum $d(x_i, C)$.
4. Determine the new cluster centers c_i for i = 1…k define as:

$$c_i = \frac{1}{n_i} \sum_{j=1}^{n(s_i)} m_{ij} \in s_i$$

5. Go back to step 2 until all centroids are convergent.

The centroids can be said converged if their positions do not change in the iteration. It also may stop in the t iteration with a threshold ε if those positions have been updated by the distance below ε:

$$\left[\frac{C^t - C^{t-1}}{C^t} \right] < \varepsilon$$

The main advantages of this algorithm are its simplicity and speed which allows it to run on large datasets. Its disadvantage is that it does not yield the same result with each run, since the resulting clusters depend on the initial random assignments. It minimizes intra – cluster variance, but does not ensure that the result has a global minimum of variance. Considering the unstable output as a disadvantage, Color Transformation mechanism is implemented in this image segmentation.

b) Experimental Results of RGB to Lab Color Transformation

Segmentation using the RGB to Lab transforms works well with identifying marks on the foreground locations. The basic procedure is as follows.

1. The color is transformed from RGB to Lab color space for the input image.
2. Read in the gradient magnitude as the segmentation function.
3. Dimensions are split.
4. Reshaping is done with respect to the colors.
5. The colors are clustered and border pixels are fixed in a matrix.
6. Modify the segmented function so that it only has minima at the foreground and background marker locations.
7. Visualize the result.

c) Visual Results of Experiments:

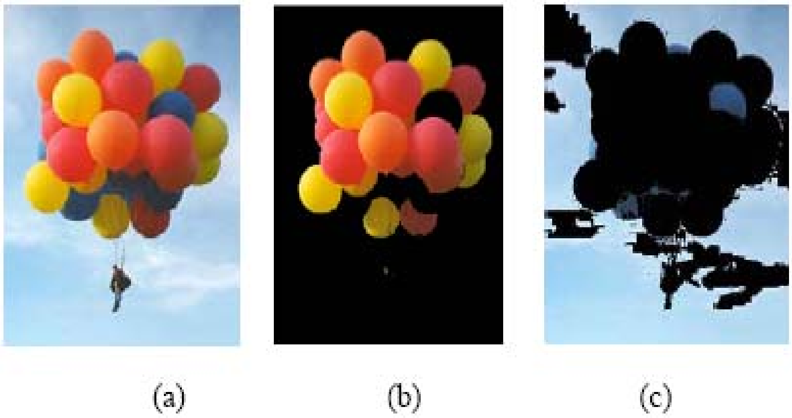

Fig 1: Visual Comparison of K – Means Image Segmentation
(a) Original Image (b) Clustering the objects (c) Clustering the background.

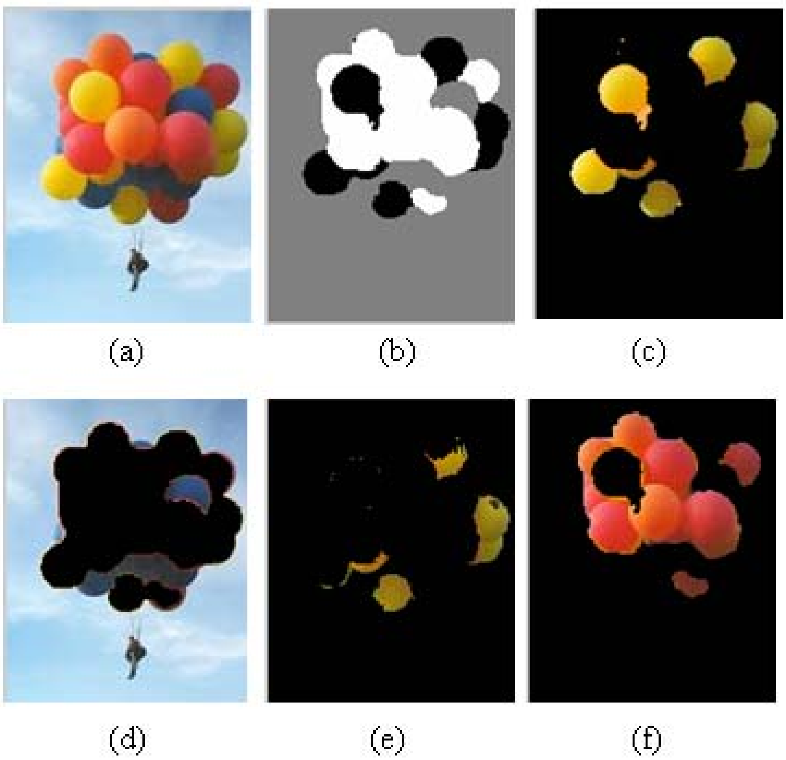

Fig 2: Visual Comparison of RGB to Lab Color Transformation Image Segmentation
(a) Original Image (b) Gray Scale Segmentation
(c) Yellow Color Segmentation (d) Blue Color Segmentation
(e) Pale Yellow Segmentation (f) Red – Orange Color Segmentation

d) Implementation for Other Images

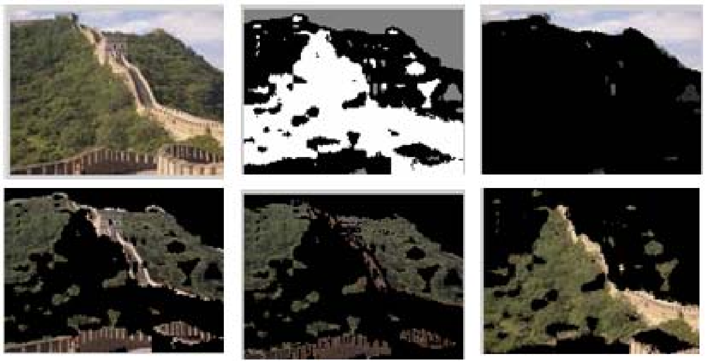

Fig 3: Image Segmentation on Satellite Image of China Wall

Fig 4: Image Segmentation on mixed color flower

e) Comparison Report

1. Colors are segmented individually in Color Transformation Method. But similar color family is segmented together in K – means Algorithm.

2. Blue color is not properly segmented in K – Means Algorithm Fig 1. The same is properly segmented in proposed method Fig 2. Fig 3 and Fig 4 shows the Lab color transformation implementation for difference types of pictures.

3. The CPU Utilization time for K – Means Algorithm execution is 60.7344ms, whereas 19.8281ms is for Color Transformation method as shown in Table 1. The comparison is shown as the graphical representation in Fig 5.

Method	Color Segmentation	CPU Time (ms)
K - Means	Better	60.7344
RGB to Lab	Good	19.8281

Table 1: Comparative Results.

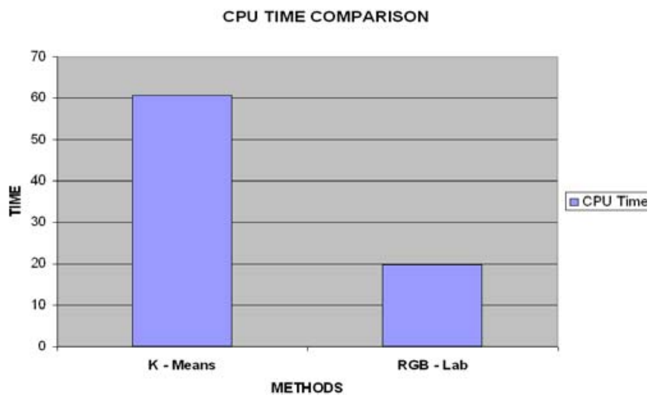

Fig 5: CPU Time Comparison

VII. Conclusion & Future Work

In this paper, we have presented a new approach for image segmentation using RGB to Lab Color Transformation methodology [8, 16]. The K means algorithm [3, 11] does not yield the same result with each run, since the resulting clusters depend on the initial random assignments. It minimizes intra-cluster variance, but does not ensure that the result has a global minimum of variance. The output of the K means algorithm does not generate a clear cluster. Hence, RGB to Lab Color Transformation methodology is implemented for the same problem and the result is shown as an optimized image. This methodology is able to optimize the clustering for image segmentation in aspects of precision and computation time. A series of experiments involving four different color spaces with variance constraint and execution time were conducted. The experimental results show that our proposed approach for image segmentation using RGB to Lab Color Transformation methodology is able to improve the precision and enhance the quality of image segmentation in all color spaces. It also performed the computational time as fast as K-means and kept the high quality of results.

References

1. K. Atsushi, N. Masayuki, "K-Means Algorithm Using Texture Directionality for Natural Image Segmentation", IEICE technical report. Image engineering, 97 (467), pp.17-22, 1998.
2. S. Ray, R.H. Turi, "Determination of number of clusters in K-means clustering and application in colthe image segmentation", Proc. 4th ICAPRDT, pp. 137-143, 1999.
3. B. Kovesi, J.M. Boucher, S. Saoudi, "Stochastic K-means algorithm for vector quantization", Pattern Recognition Letters, Vol. 22, pp. 603-610, 2001.
4. Y. Gdalyahu, D. Weinshall, M. Wermen, "Self-Organizationin Vision: Stochastic clustering for Image Segmentation, Perceptual Grouping, and Image database Organization", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 23, No. 12, pp. 1053-1074, 2001.
5. B. Wei, Y. Liu, Y. Pan, "Using Hybrid Knowledge Engineering and Image Processing in Color Virtual Restoration of Ancient Murals", IEEE Transactions on Knowledge and Data Engineering, Vol. 15, No. 5, 2003.
6. Vincent L: Morphological grayscale reconstruction in image analysis: Applications and efficient algorithms. IEEE Trans Image Proc 2:176– 201, 1993.
7. Qixiang Ye Wen Gao Wei Zeng, "Color Image Segmentation Using Density-based Clustering" IEEE - ICASSP 2003.
8. M. Miyahara and Y. Yoshida, "Mathematical Transform of (R, G, B) Color Data to Munsell (H, V, C) Color Data," SPIE Visual Communications and Image Processing, vol. 1001, pp. 650-65, 1988.
9. Malik Sikandar Hayat Khiyal, Aihab Khan, and Amna Bibi, "Modified Watershed Algorithm for Segmentation of 2D Images", Issues in Informing Science and Information Technology Volume 6, 2009.
10. Y. Xu and E. M.–K. Lai, "Restoration of images contaminated by mixed Gaussian and impulse noise using a recursive minimum–maximum method," IEEE August 1998, Vol. 145 Issue 4 (p.264-270).
11. Ali Ridho Barakbah and Yasushi Kiyoki, "A New Approach for Image Segmentation using Pillar-Kmeans Algorithm", World Academy of Science, Engineering and Technology 59 2009.
12. K. Barnard, P. Duygulu, and D. forsyth, "Clustering Art," Conference on Computer Vision and Pattern Recognition, Hawaii, pp. 434-441, 2001.
13. M. Luo, Y.F. Ma, H.J. Zhang, "A Spatial Constrained K-Means Approach to Image Segmentation", Proc. the 2003 Joint Conference of the Fourth International Conference on Information, Communications and Signal Processing and the Fourth Pacific Rim Conference on Multimedia, Vol. 2, pp. 738-742, 2003.
14. H.M. Lotfy, A.S. Elmaghraby, "CoIRS: Cluster-oriented Image Retrieval System", Proc. 16th IEEE International Conference on Tools with Artificial Intelligence (ICTAI), pp. 224-231, 2004.

Author(s) Profile

Ms. M. Kalpana Devi., M.C.A., M.Phil., M.E., has completed my UG BSc CT in P.S.G College of Technology in the year 2001. I completed my PG degree M.C.A in P.S.G College of Arts and Science in the year 2004.I completed my M.Phil in Bharathiar University in the year 2006. I completed my M.E – CSE in Anna University, Coimbatore in the year 2010. I have totally 11 years of experiences in various esteemed institutions. My Area of interest is Image Processing and Networking.