INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 11(6):860-876, December 2016.

A New Method for Colour Image Segmentation

G. Schleyer, G. Lefranc, C. Cubillos, G. Millán, R. Osorio-Comparán

Gustavo Schleyer, Gastón Lefranc*, Claudio Cubillos
Pontificia Universidad Católica de Valpaíso
Escuela de Ingeniería Eléctrica
Avda. Brasil #2147
Valparaíso, Chile
*Corresponding author: glefranc@pucv.cl

Ginno Millán
Universidad Católica del Norte
Escuela de Ingeniería
Larrondo #1281 Coquimbo, Chile
E-mail: gmillan@ucn.cl

Román Osorio-Comparán
Universidad Nacional Autónoma de México
Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas
Ciudad Universitaria
Coyoacán, México D.F., México
E-mail: roman@unam.mx

Abstract: This paper presents an unsupervised algorithm of colour image segmentation and is an extension of [1][a] . This method combines the advantages of the approaches based on split and merge and region growing, and the use of the RGB and HSV colour representation models. The effectiveness of the proposed method has been verified by the implementation of the algorithm using three different testing images with homogeneous regions, spatially compact and continuous. It was observed that the proposed algorithm outperforms the other analysed techniques requiring shorter processing time when compared with the other methods.

Keywords: Computer vision, Image processing, Image segmentation.

[a] Reprinted (partial) and extended, with permission based on License Number 3930961102612 [2016] ©IEEE, from "Computers Communications and Control (ICCCC), 2016 6th International Conference on".

1 Introduction

Image segmentation involves the identification of regions of interest, which generally are an object or a part in a digital image. Each region must maximise the homogeneity of its pixels features (colour, texture) and simultaneously maximise the differences with neighbouring regions; moreover, each region must be spatially compact. In general, image segmentation considers an important initial step in most image processing applications, which use the segmentation information in order to perform upper level tasks, such as object tracking or identification, and scenes interpretation.

This paper is an extension of paper presented in ICCCC2016 [1]. Two approaches widely used in colour image segmentation are split and merge [2]-[4] and region growing [5]-[9]. In general, the split and merge method begins with an initial and no homogeneous image partition, then keeps on splitting it until homogeneous partitions are obtained. A common data structure used in the implementation of this procedure is the quad tree representation [10]. After a division step, usually, many small and fragmented regions appear connected in some way during the merging phase. On the other hand, the method based on region growing consists in obtaining a homogeneous image region through a growing process that, beginning from a preselected seed, progressively crowd pixels around it fulfilling a determined homogeneity criterion. The growing process stops when it is not possible to add new pixels to the region. A common post-processing step consists in a union phase that eliminates small regions or neighbouring regions with similar attributes, making larger regions.

In this paper proposes an unsupervised colour image segmentation algorithm that combines the advantages of the split and merge and region growing approach using model features in the RGB and HSV colour representation. The effectiveness of the proposed method is compared with a parameter less quadrilateral-based image segmentation method [10] with a fuzzy segmentation HCI [11], and with a image segmentation using situational descriptors DCT [12]. The proposed algorithm performs the best segmentation and requires the least processing time compared with other methods used in its evaluation. It resulted to be the best approach for real applications that provides regions of homogeneous color, spatially compact and continuous, in a shorter processing time and more robustly against over-segmentation.

2 Colour images models

A digital image is modelled as three monochrome images, where each group is related to different colour. The RGB model is one of the most used methods where the resulting colour is formed by combining red, green and blue. The RGB model representation is a cube, using RGB coordinates instead of (x, y, z). The RGB complex algorithm establishes a different colour in a cube vertex. It is not easy to determine if these coordinates represent a particular tonality. In order to alleviate this problem a cylindrical coordinate model (HSV) is used.

In the HSV model (Hue Saturation Value), the colour represent by three fundamental quantities: tonality, saturation and the intensity value. The tonality identifies the colour, the saturation establishes the colour quantity represented in each pixel and finally its value. When a pixel has low saturation it is grey, and a pixel total saturated is a pure colour. The pixel intensity corresponds to the light quantity of the pixel. HSV model permits colour selection in a better way than RGB model. Colour space in HSV model is represented as a cone, where an angle in the circumference base is the tonality, the saturation is measured outwards from the origin and the vertical axis is the intensity value.

3 Colour image segmentation method

3.1 Technical methods of image segmentation

The Technical methods of image segmentation are classified in image segmentation techniques based colour image features, techniques based colour image, and image segmentation techniques based on physics [13]-[17].

Segmentation techniques based on colour image features can classified in clustering, adaptive clustering and thresholding histogram.

Image segmentation techniques based on colour image are: split and merge, techniques based on growing regions, graph theory, contours based, and neural networks.

- Split & merge technique considers spatial information and it has good results in images with homogeneous regions. However, the definition of uniformity of color can be difficult, and the tree can generate quadrangles not present objects in the image.

- Growing regions technique generates compact and spatially connected regions and some approaches produce very fast algorithms. Nevertheless, they are expensive in computational time and memory, and they have difficulties in establishing the seeds and the appropriate standard of homogeneity, because the algorithms are sequential.

Image segmentation techniques based on physics are ssupposedly robust against excessive lighting, and segmented surfaces depending on the composition of the material. The technique is restricted to one or two types of material. It has difficulties in identifying real images when are excessive lighting and shadows.

The segmentation method proposed in this paper can be classify into image-based approaches and as a mixture between two categories of segmentation techniques: split & merge and growth regions. The proposed approach divides the image in many regions (split), some of which subsequently merges based on a predetermined criterion (merge), and then algorithm can be classified into the first category. However, if the initial regions generated by the proposed algorithm considers the seeds of a region growing process, then the proposed method can be classify in the second category. The fact that the synergy produced between the two approaches can avoid disadvantages that typically find in the independent performance.

3.2 Proposed colour image segmentation method

As the original image I_0 is 24 bits/pixel RGB format and is very large, the segmentation reduces information. Then, the characteristics of the pixels of the image segment is analysed to decide whether to segment them based on their hue, saturation or intensity, using representation in the HSV model, or the segmentation must be done based on the general characteristics of colour using RGB model.

The proposed image segmentation method is in Fig. 1. This method consists on several algorithm to have an efficient and better segmentation of an image that it can use in real time applications.

First step is the RGB in HSV image conversion, reducing to 26 values. It is following with tonality segmentation including tonality label algorithm and extreme value of intensity mask application. After this segmentation, applies a merge of tonality of small and great group algorithm. Then, it continues a pixel segmentation with low saturation algorithm, with mask application low saturation and union of low saturation neighbors groups.

The next two algorithm are pixel segmentation of extreme values of intensity algorithm and remaining pixels segmentation algorithm. To finish, applies the final merge of small groups algorithm.

Because of an excessive amount of information produces in excessive processing times a reduction, gives the possibility of practical application of the algorithm, and it is much simpler to identify a colour from HSV model than RGB, the algorithm is to convert RGB image to a HSV image. The reduction is taking 26 different values for each band, which represents by 5 bits/pixel, and as there are, three bands need 3 x 5 = 15 bits per pixel.

Special pixels those image pixels having extreme values of intensity or low saturation values. The segmentation of pixels with extreme intensity values are white dot (high intensity value) or a black dot (low intensity value) in the digital image. Segmentation pixel Io with low saturation values is more complex, they appear as grey dots on the image and it is not possible segmentation by hue (tonality), so it is necessary to identify them in the first instance and subsequently dedicate a different processing to receive those pixels segmented by colour.

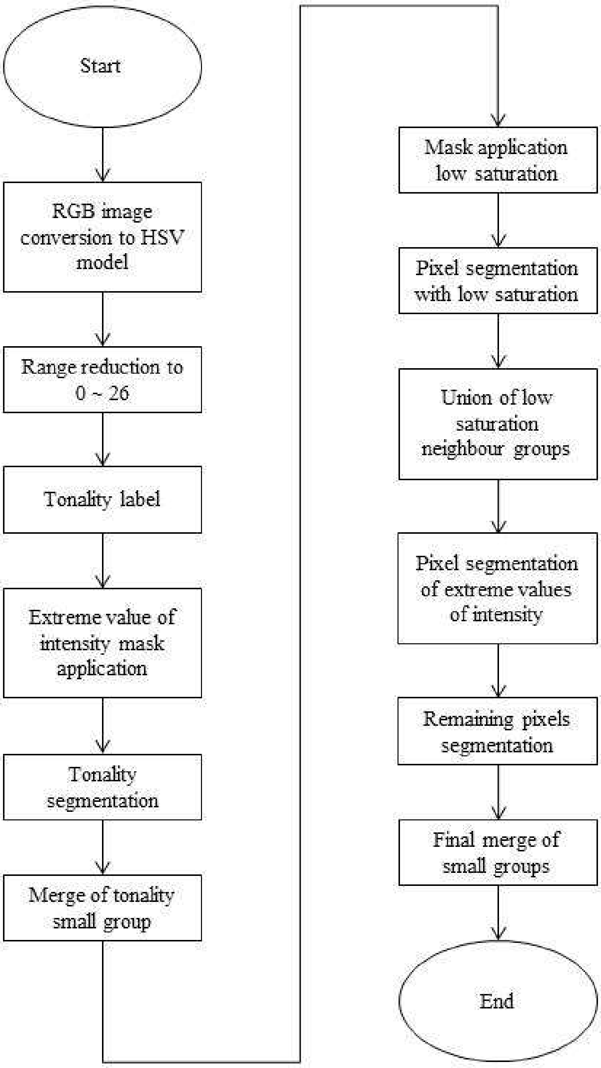

Start
RGB image conversion to HSV model
Range reduction to 0 ~ 26
Tonality label
Extreme value of intensity mask application
Tonality segmentation
Merge of tonality small group
Mask application low saturation
Pixel segmentation with low saturation
Union of low saturation neighbour groups
Pixel segmentation of extreme values of intensity
Remaining pixels segmentation
Final merge of small groups
End

Figure 1: The proposed image segmentation method

Reduction of possible combinations

First step is the RGB in HSV image conversion, reducing to 26 values, to have an acceptable detail level on image segmentation, reducing the information in the image, reducing processing time. It is following with tonality segmentation including tonality label algorithm and extreme value of intensity mask application. After this segmentation, applies a merge of tonality of small and great group algorithm. Then, it continues a pixel segmentation with low saturation algorithm, with mask application low saturation and union of low saturation neighbors groups.

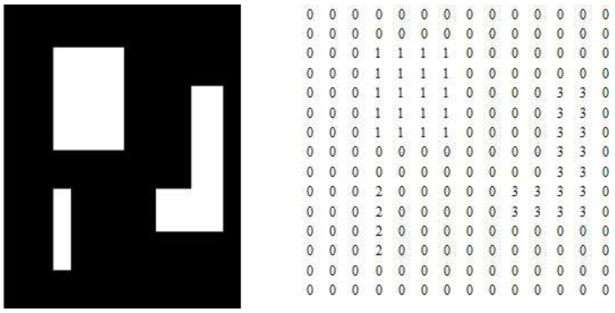
This step diminishes the quantity of information provided by the HSV model, so they can adopt integer values between 1 and 26, reducing the number of possible combinations to the inferior integer.

Segmentation by tonality

In the segmentation by tonality, special pixels (image pixels having extreme values of intensity or low saturation values) are separated from the image.

Once identified the pixels to segment according to their tonality, the neighbouring pixels that have the same tonality are merged and those ones with a different tonality are split. Then, 26 matrixes are created, one for each possible tonality value, which have been called H matrixes.

The segmentation of the no neighbouring pixels of the same tonality is carried out labelling each H matrix. The labelling consists in assigning a number to each component found in the image. So, the image is scanned from left to right and from top to bottom searching for cell sets with value "1", the first set found keeps its original value, but in the next set, the values “1" are replaced by “2", and so on, until the value n is reached, where n is the number of components in the scene. A result of the application of the previous procedure to an image is presented in Fig. 2.



```
0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 1 1 1 0 0 0 0 0 0 0
0 0 0 1 1 1 1 0 0 0 0 0 0 0
0 0 0 1 1 1 1 0 0 0 0 3 3 0
0 0 0 1 1 1 1 0 0 0 0 3 3 0
0 0 0 1 1 1 1 0 0 0 0 3 3 0
0 0 0 0 0 0 0 0 0 0 0 3 3 0
0 0 0 0 0 0 0 0 0 0 0 3 3 0
0 0 0 2 0 0 0 0 0 3 3 3 3 0
0 0 0 2 0 0 0 0 0 3 3 3 3 0
0 0 0 2 0 0 0 0 0 0 0 0 0 0
0 0 0 2 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

Figure 2: Labelling of components

The H matrixes store information of tonality, which is lost when they are merged in order to form H_{seg}.

From this point, the information of tonality of each region is stored in a column vector, called *color_reg*. The label value of each region, designated by the *num_etiqueta* variable, indicates the row of *color_reg* in, which must be, stored the tonality information.

Tonality small groups merging

Analyzing H_{seg} matrix, it shows an over segmentation image, because within the images are small groups of pixels that can correspond to shadows, grooves, reliefs, etc. These details do not represent relevant information, with less than 25 pixels. If H matrices are filtered to remove these small groups, tonality information could lost. Instead of filtering these arrays, it creates an algorithm capable of merging these small groups small with larger groups.

Then, all the groups that have less than a certain quantity of pixels, determined by an algorithm parameter called *par_area* must be merged with some neighbouring group. The *par_area* parameter, establishes a pixel minimum area of each segmented group.

To find the group with which each component has the biggest quantity of neighbouring pixels, it must be counted each kind of elements, inside the vector of the *vecinos* structure associated, with this component and to identify which one appears more times. Then, in order to identify small groups the *areas* vector is analysed searching values less than *par_area*, more formally a small group.

The group to merge with G_p, with its neighbouring group G_{fusion_p} with which exists the least tonality difference. In the case that there are more than one group with such difference, that is, if the set has more than one element, the group choose is a set, which has the bigger quantity of element.

Tonality big groups merging

A big group G_g is defined as all those groups which pixels quantity is bigger than *par_area*.

The big groups G_g are merged with those neighbouring groups with which it have a tonality difference less than or equal to one. The condition of the areas establishes that the prevailing label value is the one belonging to the group that have the biggest quantity of pixels, considering this, the label value of all the pixels belonging to G_g must be replaced by the label value of G_{fusion_g}.

Mask application low saturation. Re-labeling of groups

The merger of groups leads to the elimination of some regions and increased the size of others. The formation of groups is constantly changing when the segmentation algorithm iterates. This information must be recorded in the matrix which contains the location and H_{seg} tag value of all groups. However, update this matrix every time a group merges with another would require much processing time, thus new label values are only updated once known all new groups in H_{seg}.

As the H_{seg} matrix is scanned, each time that a pair of group, which must be merged, is found, the label values of both groups are stored in a table called T_{fusion}.

An algorithm that searches for all the groups of the table T_{fusion}, which are related with each other, is re-labelling of the c matrix. As a result, k groups that cannot be merged are obtained, which are stored in the $E(k)$ variable, finally, the H_{seg} values are refreshed.

With this step, it concludes the first iteration of the algorithm of segmentation by tonality. In the next iterations the definition of *color_reg* cannot be used, in the future it will be considered that *color_reg* for each group; will correspond to the mean of the I_0 values belonging to the respective group. The iterations of the algorithm of segmentation by tonality continue until no new groups can be merged, that is, when $T_{fusion} = \phi$. Once concluded this part of the segmentation process, it continue with that described in the next sections.

Low saturation pixels segmentation. Pixel segmentation of extreme values of intensity

The low saturation pixel segmentation algorithm is in charge for segmenting these pixels that have not been segmented by tonality segmentation algorithm. The first thing to consider is exist pixels with low saturation, but at the same time presenting intensity extreme values. This type of pixels will not be segmented by the segmentation algorithm pixels with low saturation. Therefore, first is to remove in the matrix those pixels with extreme values of intensity.

During the design segmentation algorithm pixels with low saturation various criteria such as segmentation were tested for strength value gray, HSV values and RGB values, obtaining better results with RGB. To identify the RGB value of each group present in the matrix, considers the average RGB values of each band. However, the tonality segmentation algorithm, has not identified pixels with low saturation were, they separated only those with extreme values of intensity, which has been done intentionally. As a result some groups of the matrix may be formed in part by low saturation pixel by pixel and partly medium or high saturation. It obtains two averages of the RGB bands, one for pixels with low saturation called P_{sat}, named and defined for each group and band RGB with other pixels with medium and high saturation called P_{no_sat}.

The Euclidean distance between the vectors that contains the mean values of the RGB bands coordinates, corresponding to the low saturation pixels of a group, and the vector that contains the same information in relation with the medium and high saturation pixels of the same group.

Each element of the vector $D_{sat\_no\_sat}$ is analysed in order to confirm if they are greater than the δ parameter, if the condition is fulfilled the group associated with this element must be divided, if the area of the two resultant groups is bigger than *par_area*. δ and *par_area* are algorithm parameters entered by the user. Each time a group that must be divided is identified, a register stored in the variable *separa*.

Once all the groups that must be divided are identified, the H_{seg} matrix is labelled again in order to register these changes. In this case, the labelling of the H_{seg} matrix is performed using the *separa* vector.

Low saturation neighboring groups merging

Neighbouring groups with low saturation merge. First, the areas of all the groups are calculated, refreshing the *areas* vector, then the areas of the low saturation groups are obtained refreshing the $areas_{sat}$ vector, through the multiplication of t per $M_{S\_Bajo}(fil, col)$. Second, the neighbouring pixels of the low saturation groups are identified, considering that a low saturation group is the one in which more than 50% of its pixels have low saturation values. The neighbouring pixels of each one of these groups are found.

Another important quantity that must be calculated is the mean value of each one of the coordinates of the RGB model, called P_R, P_G, and P_B, for each segmented group, not considering their saturation level.

In order to decide which low saturation groups will be merged, the Euclidean distance between the mean of each RGB band corresponding to each group, and all their neighbouring groups is calculated. If this distance is lesser than a certain value established by a new parameter called δ_2, then the groups are merged. In other words, each group will be merged with all its neighbouring groups with which the Euclidean distance between their mean RGB values is lesser than δ_2. The temporal information of the group that must be merged is stored in the *agrupa* variable. Once again, only when all the groups that will be merged are known, the labelling of the H_{seg} matrix proceeds.

If there are more than a value stored in $agrupa(k)$, the one that has the minimum Euclidean distance with the RGB mean values of the k group is chosen, and the remaining values are eliminated. During the new H_{seg} labelling the merge and labelling algorithm is utilized, but replacing the T_{fusion} variable, by *agrupa*. Once the building of the E matrix is finished, the labelling of the H_{seg} matrix. With this step the low saturation pixels segmentation concludes.

Segmentation of pixels with extreme brightness values

So far they have been segmented groups of pixels with high saturation, which allow good segmentation, then the pixel groups with low saturation have been segmented using averages of the RGB coordinates. It yet to analyze what happens with pixels with intensity extreme values, corresponding to high and low intensity values. The pixels with high intensity values appears as a white dot on the image regardless of the value of hue and saturation, so does the pixels with low intensity values, with the difference that they appear as black dots in the digital image. Pixels with intensity high and low brightness have been identified by M_{V_alto} and M_{V_bajo} matrixes, respectively. The first step of the algorithm is to label these matrices as described by M_{V_alto} and M_{V_bajo} matrixes. But due to the way they were built, in each pixels are black in color which indicates the presence points with extreme brightness intensities, so that are these pixels should re labels. As the labeling was defined to identify color pixel.

To finish the segmentation algorithm with pixels intensity extreme values, each label value of H_{seg} corresponds to a different region in the image. It adds *color_reg*, P_R, P_G, P_G and *areas* variables, and the values for the new groups.

Segmenting remaining pixels

When finish all segmentation procedures, and still exist pixels unsorted, means there are groups consisting of fewer pixels than the established by the *par_area* parameter. During arrays mask filters applied tto M_{V_alto} and M_{V_bajo} matrixes eliminate such groups. Pixels belonging to the edge of the image have not analysed by windows 3x3 and therefore not classified or incorrect. This corresponds to 0.43% of the total number of pixels of the image. To finish segmentation algorithm. To identify all remaining pixels, H_{seg} is traversed looking for pixels with value 0. The information regarding the position of these pixels within is stored in a new array called R. This demands significant processing time compared with the rest of the algorithm.

Final merge of small groups

All pixels in the digital image I_0 are been classified. The next step is merge of small groups or eliminate these groups. As previously, small group has fewer pixels than the parameter set by the *par_area*. Small groups merge with the most appropriate neighbouring group. Merger criterion is the smaller Euclidean distance between RGB averages of two neighbouring groups.

Before deleting or merging small groups identified within the H_{seg} matrix, it is not necessary to traverse the array again. Simply analysing the vector and find the smaller elements that *par_area*, the position vector, of these elements indicates the value of the label of these groups within the matrix H_{seg}, stored in vector *unir*.

The next step is to find neighbouring groups of each small group, with more *par_area* pixels. Using the vecinos expression of all neighbouring groups of small groups of the image, with the exception of those whose pixels are on the edge. In this case, it uses the w 3x3 window to search for group neighbours. To find neighbouring groups of edge pixels of the image must modified according to w window edge where they are looking for these groups.

Thus, the pixel position to which is seeking its neighbouring groups determines the w window. The next task is then to traverse H_{seg} the edges of the matrix and complete the neighbours variable using the w window corresponding pixel according to the position analysed. Once the above process, in the event that any neighbouring row of elements having varying yet, it must filled with a zero.

Then, calculate the Euclidean distance between the RGB averages for each group belonging to the group and all its neighbours groups. For each k element of *unir* vector, it chooses the minimum distance, after the computation of Euclidean distances between the RGB average and all its neighbours. The neighbouring group that has minimum distance is the group that merge k group. This information is stored in a second column of the unir vector created especially for this purpose.

The small groups are identified and what group is merged, the changes is put in the H_{seg} label matrix. Labelling of the H_{seg} matrix is the last step of image segmentation algorithm: The result are image region, identified with a different label value in the HSEG matrix. Segmentation has been used HSV and RGB colour model, identifying areas by tonality, saturation and intensity brightness level, as well as the average of the RGB pixel coordinates of each region.

3.3 Algorithm implementation

It utilizes the filtering images method included in the MATLAB platform. The goal is to designate the minimum image size that can be each targeted group, so that all groups with a smaller size are included in a larger group. The *par_area* parameter of the algorithm implies a minimum group size. The default value of this parameter is set to 25, that is to say, once the image segmentation no groups that contain less than 25 pixels over. This prevents over segmentation, which means that many groups do not constitute an object or a major part, these groups could have a digital area (defined as the number of pixels) as small as a single pixel is created.

The *par_area* parameter, establishes a pixel minimum area of each segmented group. A small value means a high resolution in the segmentation algorithm, but it could have an over segmented image. If the algorithm estimates that *par_area* parameter is greater, the algorithm-processing time is high. A recommendation to adjust this parameter depends on the expected object size in the image.

In an extreme case, the image may be segmented into as many groups as the form pixels, which makes no sense. The filtering process is performed automatically by the "bwareaopen" function of Matlab, which eliminates an original binary image all the components that are under *par_area* pixels with value 0, producing a new binary image without deleted items. To remove pixels with value 1 must be obtained negative image, filtering and then get the negative of the filtered image. The bwareaopen function has three steps: determine the connected components; compute the area of each component and eliminate small objects. It has been used connectivity 4 to set the connected components, considering that a couple of pixels with value 0 are neighbours if they are attached horizontally or vertically, at this level of connectivity if only they are attached diagonally, they are neighbours. See Fig. 3.

When shapes with holes exist, they are removed if less than the minimum number of pixels required to be considered as a region within the segmentation.

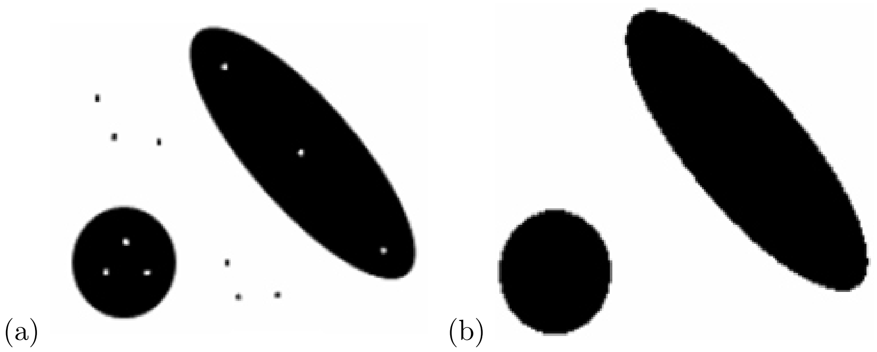


Figure 3: (a) Original image; (b) Filtered image

Once identified the pixels, they are segmented according to their tonality. The neighbouring pixels with the same tonality are merged and those with a different tonality are split. It creates 26 H matrixes, one for each possible tonality value.

The segmentation of the no neighbouring pixels of the same tonality is carried out labelling each H matrix. The labelling consists in assigning a number to each component found in the image. So, the image is scanned from left to right and from top to bottom searching for cell sets with value 1, the first set found keeps its original value, but in the next set, the values “1" are replaced by “2", and so on, until the value “n" is reached, where "n" is the number of components in the scene. A result of the application of the previous procedure to an image is represented in Fig. 4.

The results shows that there is an over segmentation image, because there are small groups of pixels may correspond to shadows, grooves, reliefs, etc. constituting the image details but alone do not represent relevant information. It observes many groups with less than 25 pixels. If they had leaked the H matrices would have removed these small groups, but had lost the information tonality of these pixels. It creates an algorithm capable of merging these small groups with larger groups. This merging algorithm determines the neighbours groups for each component.

The merger of groups lead to the elimination of some regions and increased the size of others, the formation of groups is constantly changing as the segmentation algorithm iterates. Once merging clusters and updating values are done, the first iteration of the tonality segmentation algorithm concludes until no new groups can be fused.

Pixels with low saturation, that have not been segmented by the tonality segmentation algorithm, are segmented by other algorithm using RGB intensity value grey segmentation, which gives better results.

After the segmentation of the groups of pixels with low saturation, there are two possible scenarios for each group. One is that no segmentation is performed and the group remains united and the other is that the pixels with low saturation of the group are separated from their pixels with medium or high saturation, dividing the group into two.

To decide if the group remains united or divided the δ parameter is used, with a default value of this parameter 25. The criteria states that for a group to be divided, the Euclidean distance between two vectors, one containing the coordinates of the mean values of the bands RGB pixels with low saturation of a group, and with other similar pixels with medium and high saturation of the same group must be greater than . In addition the area of the two groups resulting after the division must be greater than the *par_area* division parameter to be performed. The Euclidean distance between two vectors is described by

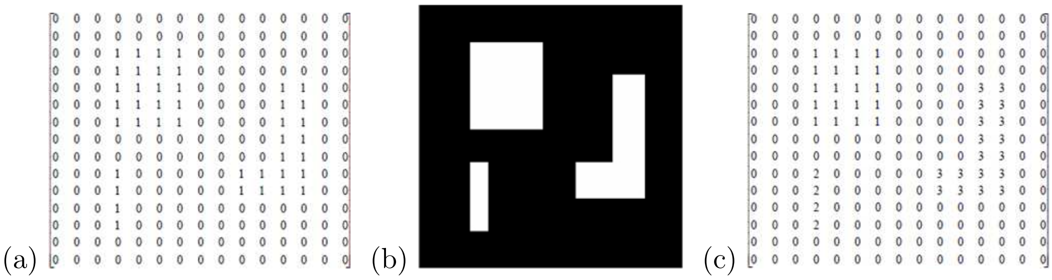
$$D_{sat\_no\_sat}(k) = 0.2264\sqrt{R^2 - G^2 - B^2}, \tag{1}$$

where $R = P_{sat\_R}(k) - P_{no\_sat\_R}(k)$, $G = P_{sat\_G}(k) - P_{no\_sat\_G}(k)$, $B = P_{sat\_B}(k) - P_{no\_sat\_B}(k)$, and k is the label number.

Each element of the $D_{sat\_no\_sat}$ is vector check if greater than δ parameter. If it is greater than δ means that the group associated with this element has to always be divided. When the area of the resulting two groups is greater than *par_area* parameter, the group to divide, a record is stored in the *separa* variable is created, which is defined by the expression

$$separa(k) = \begin{cases} \max(separa) + 1, & \text{if } D_{sat\_no\_sat}(k) > \delta \wedge area_{sat}(k) > par\_area \\ 0 & , \text{ if } D_{sat\_no\_sat}(k) \leq \delta \vee area_{sat}(k) \leq par\_area \end{cases}, \tag{2}$$

where k is the label number an the initial value of *separa* is zero.



(a)

$$\begin{bmatrix} 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&1&1&1&1&0&0&0&0&0&0&0&0 \\ 0&0&0&1&1&1&1&0&0&0&0&0&0&0&0 \\ 0&0&0&1&1&1&1&0&0&0&0&1&1&0&0 \\ 0&0&0&1&1&1&1&0&0&0&0&1&1&0&0 \\ 0&0&0&1&1&1&1&0&0&0&0&1&1&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&1&1&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&1&1&0&0 \\ 0&0&0&1&0&0&0&0&0&1&1&1&1&0&0 \\ 0&0&0&1&0&0&0&0&0&1&1&1&1&0&0 \\ 0&0&0&1&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \end{bmatrix}$$

(b)

(c)

$$\begin{bmatrix} 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&1&1&1&1&0&0&0&0&0&0&0&0 \\ 0&0&0&1&1&1&1&0&0&0&0&0&0&0&0 \\ 0&0&0&1&1&1&1&0&0&0&0&3&3&0&0 \\ 0&0&0&1&1&1&1&0&0&0&0&3&3&0&0 \\ 0&0&0&1&1&1&1&0&0&0&0&3&3&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&3&3&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&3&3&0&0 \\ 0&0&0&2&0&0&0&0&0&3&3&3&3&0&0 \\ 0&0&0&2&0&0&0&0&0&3&3&3&3&0&0 \\ 0&0&0&2&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&2&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \end{bmatrix}$$

Figure 4: (a) Original matrix binary; (b): Binary matrix image; (c): Matrix labeled

Once the labelling, all new groups are identified. The δ parameter is the maximum Euclidean distance between the mean of each low saturation pixels RGB average in each group, and all their neighbouring groups.

To decide which groups with low saturation will be merged, the Euclidean distance between the averages of each RGB band of each group with respect to all its neighbouring groups is calculated, if the distance is less than a certain set δ_2 value, then it proceeds to the merger groups. In other words, each group will be merged with its neighbours groups with which Euclidean difference between the averages of RGB values is less than the parameter δ_2. The newly described Euclidean distance is defined by following expression with its mixture of pixels with different levels of saturation,

$$D_{P\_RGB}(k)_i = 0.2264\sqrt{R_v^2 - G_v^2 - B_v^2}, \tag{3}$$

where $R_v = P_R(k) - vecinos(k)$, $G_v = P_G(k) - vecinos(k)$, $B_v = P_B(k) - vecinos(k)$, $k = 1, 2, ..., \max(H_{seg})$, and $i = 1,2,...,$ quantity of neighbours groups to k group.

The temporal information groups to be merged is stored in the *agrupa* variable, expression

$$agrupa(k) = agrupa(k) \cup v(k), \tag{4}$$

where $v(k) = \left\{vecinos(k)_i | D_{P\_RGB}(k)_i \leq \delta_2\right\}$.

Then, only once all the groups will be merged known proceed to label matrix H_{seg}. In the case, that there is more than one stored in $agrupa(k)$ value, the minimum distance of RGB average with group k is chosen, and the remaining values are removed. A new labelling of H_{seg} and labelling groups, is used the fusion algorithm.

Once all kinds of segmentation described above, it is possible that still exist unclassified pixels. These correspond to those groups formed by a smaller number of pixels than that set by the *par_area* parameter, which is set to 25.

To identify all the remaining pixels, it has to traverse the array in searching of pixels with value 0. The information regarding the position of these pixels within H_{seg} is stored in a new array called R.

When all pixels in the digital image are classified, the next step is to remove all small groups. As previously stated, a small group is one that is composed of a smaller number of pixels than that set by the parameter *par_area*. Small groups merge with the most appropriate neighbouring group.

Once the small groups are identified and known to be merged with that group, you must register the respective label changes in the matrix. For this, we have created an algorithm.

Summarizing, the image segmentation method has three parameters: *par_area*, δ and δ_2.

The *par_area* parameter, establishes a pixel minimum area of each segmented group. A small value means a high resolution in the segmentation algorithm, but it could have an over segmented image. If the algorithm estimates that *par_area* parameter is greater, the algorithm-processing time is high. A recommendation to adjust this parameter depends on the expected object size in the image.

The δ parameter is the maximum Euclidean distance between the mean of each low saturation pixels RGB average in each group, and all their neighbouring groups. If this distance is less than a certain established value, the parameter is called δ_2; in this case, the groups are merged. In other words, each group will be merged with all neighbouring groups with which the Euclidean distance between their mean RGB values is less than δ_2.

The δ and δ_2 work with low saturation pixels, controlling the pixel tendencies to be in a higher saturation group or to merge to neighbouring groups with low saturation pixels. The image segmented groups probably grows when these parameters values diminishes. Their configuration has to consider the different tonalities. For example, in the image with several objects of similar tonalities, it has to separate groups with low saturation and to avoid those neighbouring groups with the same saturation that merge. This is achieved reducing δ and δ_2 values, changing the variable value in the proposed algorithm.

The *par_area*, δ, and δ_2 parameters should have to adjust to allow the proposed segmentation algorithm to adapt to images with different colour characteristic. During evaluations of this proposed point of view, the parameters values remain constant in all image tested, obtaining good segmentation. However, the results estimated can be improved choosing optimal parameters values in each digital image to be segmented. In the algorithm, the initials parameters are $par\_area = 25$, $\delta = 5$ y $\delta_2 = 2$.

The HSV possible values are reduced from 256 to 26, in an arbitrary way, to have good results in images with small tonality, instead of reducing all tonalities spectrum presented.

4 Evaluations of proposed segmentation method

4.1 Proposed segmentation method test

To test the proposed segmentation method, it utilize a House image and Lego image.

Figure 5a shows the original House test image, having neighbouring regions with similar tonal values inside; different objects with complex shapes and different textures; and shadows in some regions of the image. Figures 5b and 5c show a good performance segmentation, despite some shadow areas segmented as a region. Objects with complex shapes are segmented, without an over-segmentation or excessive union of regions.

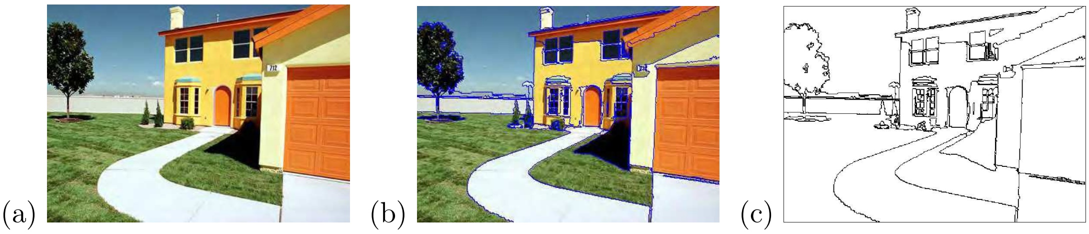


Figure 5: (a) Original matriz binary; (b): Segmented image; (c): Segmented regions

The house image test has a resolution of 449x322. Proposed algorithm obtains a segmented image with 120 groups and it lasts 35.19 seconds. Proposed segmentation method has good performance.

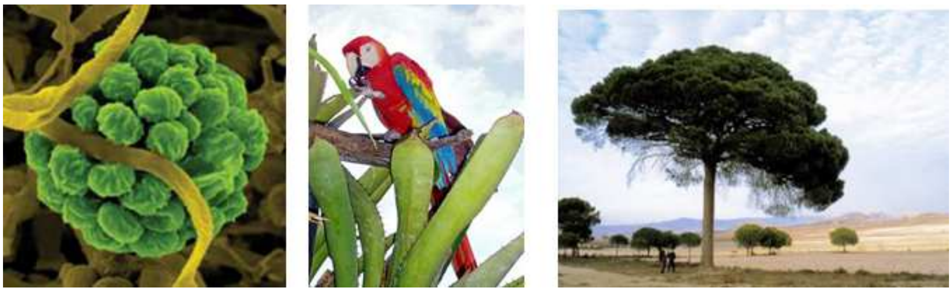


Figure 6: Original images for the evaluation

To calculate the processing time, it is necessary to consider that the algorithm has been programmed in version 7.0 (release 14) MATLAB and has been run on a computer with Intel Centrino processor only 1.86 GHz and 1.5 GB of RAM. The processing time is linked to the number of regions in the image. An image so large and few regions may require less processing time than and image of smaller size and more regions to segment. Furthermore, for images with the same number of regions and different resolution, it requires a longer processing time. In summary, the processing time required by the algorithm depends on both the image size and the number of regions present therein, the latter being unknown amount before execution algorithm.

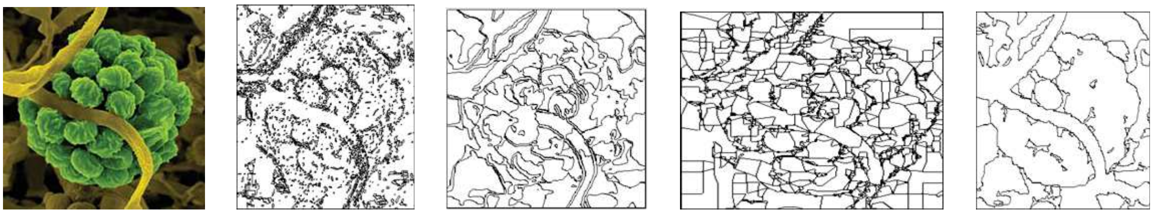
To compare the results with other methods, it is used the objective evaluation method proposed in [18], since it does not require the setting of parameters or a reference image. The $L(I)$ evaluation function is defined through equation (5).

$$L(I) = \sqrt{N_r}[1000(MN)]^{-1} \sum_{i=1}^{N_r} (e_i^2 A_i^{-1/2}), \tag{5}$$

where I is the original image to be segmented N_r is the number of segmented regions in I, A_i is the number of pixels in the region R_i, M and N are the height and the width of I, and e_i^2 is the colour error in the R_i region, which is defined as the sum of the Euclidean distance of the RGB colour vector between I and the corresponding segmented image for each pixel in the region.

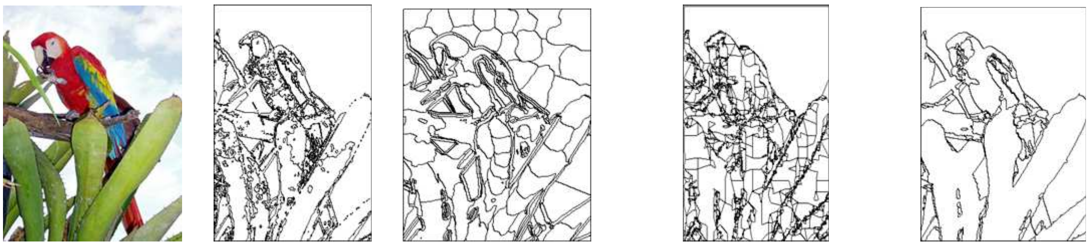
Smaller L value means a better performance. The evaluation is performed comparing the obtained results by means of the proposed algorithm with results obtained by other three methods: HCI fuzzy segmentation [11], segmentation using situational DCT (Discrete Cosine Transform) descriptors [12] and parameterless quadrilateral-based image segmentation method [10].

Figures 7, 8, and 9 show image segmentation using the different methods. Where (a) is the original image, (b) the HCI fuzzy segmentation, (c) the segmentation using situational DCT descriptors, (d) is the parameter less quadrilateral-based image segmentation, and (e) the proposed image segmentation.



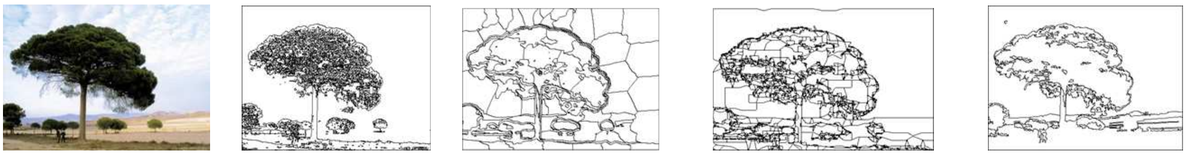
(a) Original (b): HCI (c): DCT (d): Less quad-based (e): Proposed

Figure 7



(a) Original (b): HCI (c): DCT (d): Less quad-based (e): Proposed

Figure 8



(a) Original (b): HCI (c): DCT (d): Less quad-based (e): Proposed

Figure 9

Parameter less quadrilateral-based image segmentation is constructed from a map edge (generated from a segmentation algorithm edges), neighbours quadrangles which have similar characteristics are fused to generate quadrilaterals regions allow local variations eliminating unnecessary details, so each region is full and accurately described by a set of quadrilaterals having neighbouring pixels between them. These areas have been separated into different regions, generating more groups. The amount of resulting groups has been called actual or real number of groups, while the original amount of groups calculated by the respective method is called simply number of groups.

Since (1) penalizes the creation of new groups, which increase the error in those segmentation methods that tend to do over segmentation. The $L(I)$ error is always less than the real $L(I)$ error. When the HCI fuzzy and the situational DCT descriptors segmentation methods are compared with the proposed method, the real $L(I)$ error must be considered. The objective analysis of the results summarized in the Table 1 demonstrates that the proposed method presents a smaller error and it requires less processing time than the methods used during its evaluation when they are applied to the three test images presented in Figs. 6, 7, 8.

Table 1: Comparison of results with other methods

Image	**Stachybotrys Chartarum**	**Parrot**	**Tree**
Resolution	300x300	256x340	450x315
Method 1: HCI Fuzzy Segmentation			
Number of Groups	6	8	5
L(I) Error	0.0731	0.0869	0.0275
Real Number of Groups	1731	1029	1990
Real Error L(I)	1.3967	1.5517	0.8052
Time (sec)	381.47	604.66	641.58
Method 2: Parameterless Quadrilateral-based Image Segmentation			
Number of Groups	62	106	105
L(I) Error	0.3511	0.8985	0.41
Real Number of Groups	252	448	390
Real Error L(I)	0.9105	2.54	1.0628
Time (sec)	1706.20	1622.06	4287.42
Method 3: Segmentation Using Situational DCT Descriptors			
Number of Groups	347	357	446
L(I) Error	4.1938	10.282	12.3995
Time (sec)	4742.25	7551.27	12071.94
Method 4: Proposed Segmentation Method			
Number of Groups	62	106	105
L(I) Error	0.199	0.4654	0.2781
Time (sec)	24.75	30.11	36.08

The region quantity is also smaller than in the other methods. It is compact and contents precise regions. Hence, the proposed method outperforms the other segmentation algorithms especially for real time applications. All experiments are carried out using the same computer.

The HCI Fuzzy segmentation method resulted to have a good segmentation in high value zones and small intensity, but it can over segment in other zones. Its processing times are large. In comparison, the DCT method has good segmentation in some regions, but over segmentation in others. The parameter less quadrilateral-based image segmentation method has the same problem. In this latter case, the processing time was very large.

The obtained results indicate that the proposed method is fast without over segmentation. It has good segmentation capability compared to the other methods. Few regions were not segmented, but with overall better performance delivering more homogeneous regions of colour, spatially compact and continuous.

The processing time required by the proposed algorithm in order to perform the segmentation mainly depends of the image size as much as the initial quantity of regions contained in the image. The evaluation of the proposed method indicates that this requires less than 7% of the processing time required by the approaches involved in the comparison in order to perform the segmentation over the test images.

Therefore, it is estimated that the proposed approach is suitable in order to be implemented in real applications with small processing times. The parameters par_area, δ, and δ_2 must be adjusted to allow the segmentation with different colour features. During the evaluation, the value of these parameters was maintained constant for all testing images, obtaining good results in the segmentation. In the proposed algorithm, the possible values of each HSV channel that the pixel image can take have been arbitrarily reduced from 256 to 26. This reduction has provided good results. It was considered that a scarce variation of tonality was preferable to reduce the complete HSV spectrum. All corresponding algorithms to each method run on the same computer (Intel Centrino processor 1.86 Ghz, 4 GB of RAM). As noted in Table 1, the proposed method requires less than 7% of the processing time involved employing approaches in comparison to the segmentation of the test images. However, the computer processing time required can decrease further proposed implemented in C++ programming language.

Conclusions

A new unsupervised colour image segmentation algorithm is presented. This method combines the advantages of the approaches based on split and merge and region growing, and the use of the RGB and HSV colour representation. The effectiveness of the proposed method has been checked by means of its implementation using an algorithm, which supplies homogeneous regions, spatially compacts and continuous. It has been observed that the proposed algorithm outperforms other segmentation algorithms and requires shorter processing time.

The combinations of both segmentation techniques results in an increment of the rate of the division in the split and merge method. The determining the initial number in the growing region is solved. The proposed algorithm is compared with the results obtained by other three methods: HCI fuzzy segmentation, segmentation using situational DCT (Discrete Cosine Transform) descriptors and parameter less quadrilateral-based image segmentation method.

The obtained result indicates that the proposed method is fast, does not have over segmentation, it has good segmentation and is better than the other methods. A few regions are not segmented. The method has better performance, delivering more homogeneous regions of colour, spatially compact and continuous.

The processing time required by the proposed algorithm in order to perform the segmentation mainly depends on the image size and the amount of regions contained within the image. The evaluation of the presented method indicates that this requires less than 7% of the processing time required by the compared approaches making it a good candidate where fast image processing is required.

The proposed method can make good segmentation without previous information. It is estimated that it can perform better if par_area, δ, and δ_2 parameters are adjusted automatically and to design a method to reduce the HSV space automatically according to the images colour characteristics.

Bibliography

[1] Scheleyer, G.; Cubillos, G.; Millán G., Osorio-Comparán, R.; Lefranc G. (2016); A New Colour Image Segmentation, *Computers Communications and Control (ICCCC), 2016 6th International Conference on*, IEEE Xplore, DOI: 10.1109/ICCCC.2016.7496766, ISBN 978-1-5090-1735-5, 232-239.

[2] Sanmati S. Kamath; Jackson, J.(2006); Color Image Segmentation in RGB Using Vector Angle and Absolute Difference Measures, *Proc. 14th European Signal Processing Conference (EUSIPCO 2006)*, 1-5.

[3] Navkirat Kaur, V. K.; Banga, Avneet Kaur (2013); Image Segmentation Based On Color, *Proc. IJRET: International Journal of Research in Engineering and Technology*, 495-497.

[4] Shamik Sural; Gang Qian; Sakti Pramanik (2002); Segmentation and Histogram Generation Using the HSV Color Space for Image Retrieval, *Proc. 2002 International Conf. on year 2002* 2(2):589-592.

[5] Zhi-Kai Huang; De-Hui Liu (2007); Segmentation of Color Image Using EM algorithm in HSV Color Space, *Information Acquisition Intl Conf.*, 8(11): 316-319.

[6] Kanchan Deshmukh; IAENG, B.; Ganesh Shinde (2006); Adaptive Color Image Segmentation Using Fuzzy Min-Max Clustering, *Engineering Letters*, 13(2): 1-8.

[7] Krishna Kant Singh; Akansha Singh (2010); A Study of Image Segmentation Algorithms For Different Types Of Image, *IJCSI International Journal of Computer Science*, 7: 414-417.

[8] Dibya Jyoti Bora; Anil Kumar Gupta (2014); Clustering Approach towards Image Segmentation: An Analytical Stud, *IJRCAR*, 12(7): 115-124.

[9] A. Jurio, M. Pagola, M. Galar, C. Lopez-Molina (2010). A Comparison Study of Different Color Spaces in Clustering Based Image Segmentation, *Information Processing and Management of Uncertainty in Knowledge-Based Systems. Applications*, 81: 532-541.

[10] Chung, R., Yung N., Cheung P. (2005); An Efficient Parameterless Quadrilateral-Based Image Segmentation Method, *IEEE Trans. Pattern Analysis and Machine Intelligence*, 27(9): 1446-1458.

[11] Carron, T; Lambert, P. (1996); Symbolic Fusion of Hue-Chroma-Intensity Features for Region Segmentation, *Proc.1996 IEEE Intl Conf. on Image Processing*, 1: 971-974.

[12] Jie Wei (2001); Image Segmentation Using Situational DCT Descriptors, *Proc.2001 IEEE Intl Conf. on Image Processing*, 1: 738-741.

[13] Osorio-Comparán, R.; Olivera, M.; Pe na, M.; López-Juárez, I., Savage, J.; Lefranc, G. (2013); Using Background and Segmentation Algorithms Applied in Mobile Robots, *6th IFAC Int. Conf. on Management and Control of Production and Logistics*, 6(1): 135-140.

[14] Rajeshwar Dass; Priyanka, Swapna Devi (2012); Image Segmentation Techniques, *Intl Journal of Electronics & Communication Technology*, 3(1): 66-70.

[15] MD Hedayetul Islam Shovon (2012); *Clustering and Image Segmentation*, LAP Lambert Academic Publishing, 2012.

[16] Nida M. Zaitoun; Musbah J. Aqel (2015); Survey on Image Segmentation Techniques. *Procedia Computer Science*, 65: 797-806.

[17] Szeliski, R. (2010); *Computer Vision: Algorithms and Applications*, Springer.

[18] Liu, J., Yang, Y. (1994); Multiresolution Color Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(7): 689-700.