A new Colour Image Segmentation

Gustavo Scheleyer*, Claudio Cubillos**, Gastón Lefranc*
Pontificia Universidad Católica de Valparaíso
*Escuela de Ingeniería Eléctrica
**Escuela de Ingeniería Informática
Avda. Brasil #2147.
Valparaíso – Chile,
glefranc@ucv.cl

Román Osorio-Comparán
IIMAS, Universidad Nacional Autónoma de México,
roman@unam.mx

Ginno Millán
Universidad Católica del Norte, Chile
Departamento de Ingeniería Informática
Coquimbo – Chile
gmillan@ucn.cl

***Abstract*— In this paper an unsupervised colour image segmentation algorithm is presented. This method combines the advantages of the approaches based on split&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 analysed methods.**

Index Terms - Image Processing, Image Segmentation, Computer Vision.

I. 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 is considered 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.

Two approaches widely used in colour image segmentation are split&merge [1]-[3] and region growing [4]-[8]. In general, the split&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 [11]. After a division step, usually, many small and fragmented regions appear which must be 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, it is proposed an unsupervised colour image segmentation algorithm that combines the advantages of the split&merge and region growing approach using model features in the RGB and HSV colour representation. The effectiveness of the proposed method has been compared with fuzzy segmentation HCI [9], image segmentation using situational descriptors DCT [10], and a parameterless quadrilateral-based image segmentation method [11]. The proposed algorithm performs the best segmentation and requires the least processing time when it is compared with the methods used in its evaluation. It resulted to be the best approach for real applications.

II. MODELS FOR COLOUR IMAGES

A digital image can be modelled as three monochrome images, where each group is related to different colour. The RGB model (Red, Green, Blue) 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 is represented 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.

The image is in RGB format, nevertheless, as it was previously established, is very simpler to identify a colour from the HSV model, and therefore is necessary to perform a conversion between the RGB and the HSV model. The conversion must map the RGB cube coordinates to the HSV cone ones. Having this in mind, the HSV cone is inscribed in the RGB cube, so the neutral axis of the cone correspond with the straight line passing the coordinates (R,G,B) = (0,0,0) y (R,G,B) = (255,255,255) represented in Fig. 1.

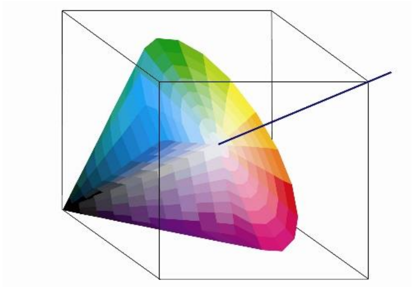


Fig. 1. HSV and RGB model conversion.

III. COLOUR IMAGE SEGMENTATION METHOD

The original image I_o is 24 bits/pixel RGB format. As the image is very large, the information for segmentation is reduced. 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 color, in which case the RGB model is used.

The proposed image segmentation method is in Fig. 2. 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 neigbourgh 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 color 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 3x5 = 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 I_o 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 color.

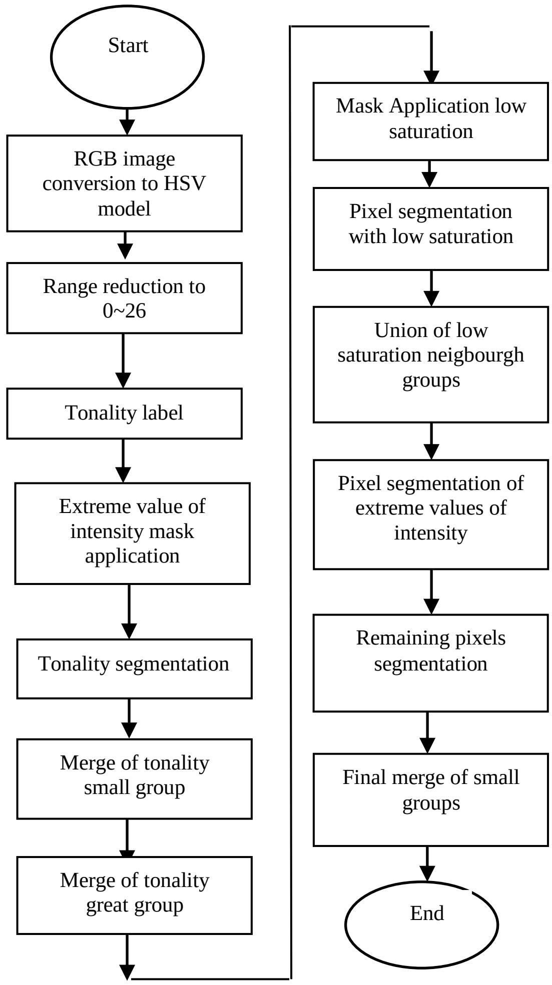


Fig. 2. The proposed image segmentation method

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 figure 3.

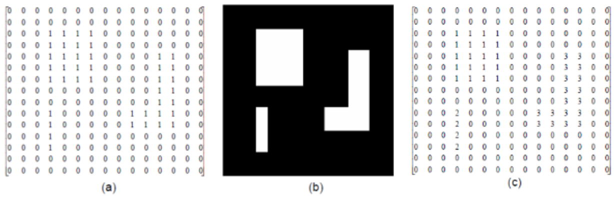
Fig. 3. a) Original image b) Filtered image

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. In the case of having enough pixel, the hole is a region within the segmentation and is not removed by the filter.

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.



a) Original Matrix Binary. (b) binary matrix image. (c) Matrix (a) Labeled.

Fig. 4. Label components

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 δ. 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 the expression (1).

$$D_{sat\_no\_sat}(k) = 0.2264\sqrt{\left(P_{sat\_R}(k)-P_{no\_sat\_R}(k)\right)^2 - \left(P_{sat\_G}(k)-P_{no\_sat\_G}(k)\right)^2 - \left(P_{sat\_B}(k)-P_{no\_sat\_B}(k)\right)^2} \quad (1)$$

Where k = 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 (2).

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

Where k = label number and the initial value of *separa* is zero.

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 δ2 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 δ2. The newly described Euclidean distance is defined by expression (3) with its mixture of pixels with different levels of saturation.

$$D_{P\_RGB}(k)_i = 0.2264\sqrt{(P_R(k)-vecinos_i(k))^2-(P_G(k)-vecinos_i(k))^2-(P_B(k)-vecinos_i(k))^2} \tag{3}$$

Where k = 1,2,……, max(H_{seg})

i = 1,2,……, quantity of neighbours groups to k group

The temporal information groups to be merged is stored in the *agrupa* variable, expression (4).

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

Where: $v(k)=\{vecinos(k)_i \mid D_{P\_RGB}(k)_i\leq\delta_2\}$

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_{xeg} 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 y 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.

Par_area, δ y 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, δ = 5 y δ_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 in the image.

IV. EVALUATIONS OF SEGMENTATION ALGORITHM

To evaluate the segmentation algorithm three-color test images are used: a plant (stachybotrys chartarum), a parrot and a tree, as shown in Figure 5.

Fig. 5 Original images for the evaluations.

Los resultados obtenidos en la segmentación de las siete imágenes de prueba presentadas hasta aquí han sido sintetizados en la Tabla I, donde se aprecia la resolución de cada imagen de prueba, el número de regiones obtenidas producto de la segmentación, el error L(I) calculado utilizando la ecuación (5), y el tiempo en segundos que tarda el método propuesto en segmentar la imagen. En el cálculo del tiempo de procesamiento es necesario considerar que el algoritmo ha sido programado en la versión 7.0 (release 14) de Matlab y ha sido ejecutado en un computador con procesador Intel Centrino solo de 1.86 GHz y 1.5 GB de RAM. El tiempo de procesamiento va ligado a la cantidad de regiones presentes en la imagen, de este modo una imagen de gran tamaño y pocas regiones puede requerir menos tiempo de procesamiento que una imagen de tamaño más pequeño y más regiones a segmentar. Por otro lado, para imágenes con la misma cantidad de regiones y diferente resolución, aquella que tenga mayor tamaño requerirá de un mayor tiempo de procesamiento. En resumen, el tiempo de procesamiento requerido por el algoritmo depende tanto del tamaño de la imagen como de la cantidad de regiones presentes en ésta, siendo esta última cantidad una incógnita antes de la ejecución de algoritmo.

To compare the results with other methods, it is used the objective evaluation method proposed by Liu and Yang in [11], since it does not require the setting of parameters or a reference image. The $L(I)$ evaluation function is defined through equation (5).

$$L(I)=\frac{\sqrt{N_r}}{1000\cdot(M\cdot N)}\sum_{i=1}^{N_r}\frac{e_i^2}{\sqrt{A_i}} \qquad (5)$$

Where I is the original image to be segmented.

N_r is the number of segmented regions in I,

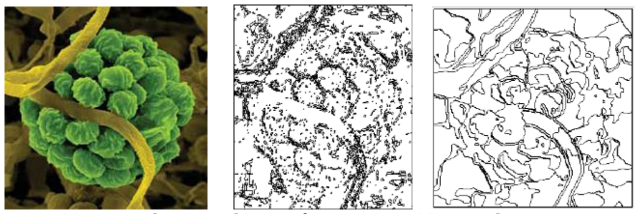
A_i is the number of pixels in the region R_i,

M and N are the height and the width of I, and

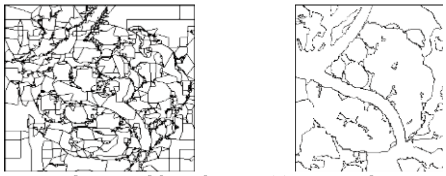
e_i^2 is the colour error in the R_i region, which is defined as the sum of the Euclidean distance of the RGB colour vector between I and the corresponding segmented image for each pixel in the region.

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 [9], segmentation using situational DCT (Discrete Cosine Transform) descriptors [10] and parameterless quadrilateral-based image segmentation method [11].

Figures 6, 7, 8 show image segmentation using the different methods. Where (a) is the original image; (b) is HCI fuzzy segmentation; (c) is segmentation using situational DCT descriptors; (d) is parameterless quadrilateral-based image segmentation; and (e) is the proposed image segmentation.

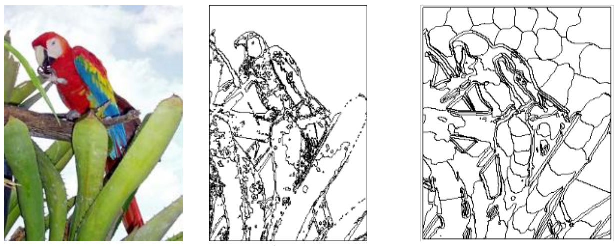


(a) Original (b) HCI fuzzy segm. (c) DCT descriptor

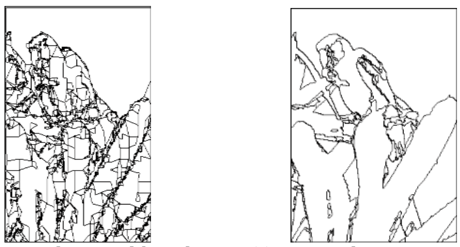


(d) Parameterless quad-based (e) proposed segmentation.

Fig. 6 Plant segmentation with different methods

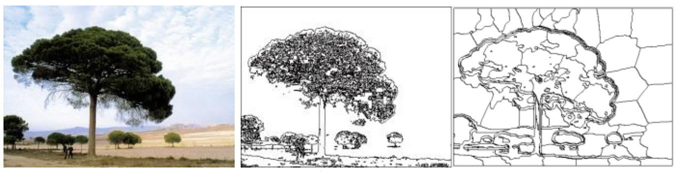


(a) Original (b) HCI fuzzy segm. (c) DCT descriptor

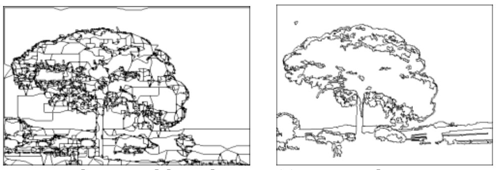


(d) parameterless quad-based (e) proposed segmentation.

Fig. 7 Parrot segmentation with different methods



(a) Original (b) HCI fuzzy segm. (c) DCT descriptor



(d) parameterless quad-based (e) proposed segmentation.

Fig. 8 Tree segmentation with different methods

In the HCI fuzzy and situational DCT descriptors a distinction between number of groups and real number of groups is done. This is necessary because it was observed that these methods can create regions, which present discontinuities, that is, it can be areas in the image belonging to the same region, but without any neighbouring pixel between them. These zones have been separated in different regions, generating more groups. The quantity of resulting groups, is called number of groups.

Parameterless 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 the equation (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 Fig. 6, 7, 8.

In comparison with the Situational DCT Descriptors method and despite the No of groups obtained with respect to the proposed method, it is observed that, as the L (I) error is bigger than the L (I) error of the proposed method for the three test images. In general, the quantity of obtained regions with the proposed method is less than in the approaches used for the evaluation, which evidences the robustness of the presented algorithm in front of the over segmentation. The proposed algorithm obtains more compact and precise regions in comparison with the remaining approaches analysed, requiring less processing time. This means that is better in real time applications.

Symbol (*) in method 2 processing time indicates the processing time that DCT method lasts in creating situational descriptors (it is not the total time): This time does not consider the time required to analyse the principal component and the adaptive k-mean classification algorithm iteration. Method 2 takes more time than the proposed algorithm.

Table I shows that the proposed method has smaller error and requires less processing time than the other compared methods. 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.

Table I Comparison of results with other methods

Image	stachybotrys chartarum	parrot	tree
Resolution	300 x 300	256 x 340	450 x 315
Method 1: HCI Fuzzy Segmentation			
No. 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 (s)	381.47	604.66	641.58
Method 2: Parameterless quadrilateral-based image segmentation			
No. 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 (s)	1706.20 (*)	1622.06 (*)	4287.42 (*)
Method 3: Segmentation using situational DCT (Discrete Cosine Transform) descriptors			
No. of Groups	347	357	446
L(I) error	4.1938	10.282	12.3995
Time (s)	4742.25	7551.27	12071.94
Method 4: Proposed Segmentation Method			
No. of Groups	62	106	105
L(I) error	0.199	0.4654	0.2781
Time (s)	24.75	30.11	36.08

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 parameterless 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 only 1.86 GHz and 4 GB of RAM). As noted in Table I, 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 be decreased further proposed implemented in a programming language that allows greater speed of execution algorithm, such as C ++.

The algorithm has been tested in a mobile robots (used in reference [12]), to detect, to track and to recognize objects.

CONCLUSIONS

A new unsupervised colour image segmentation algorithm is presented. This method combines the advantages of the approaches based on split&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&merge method. The problem of determining the initial number in the growing region is solved and the speed increased.

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 parameterless 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 color, 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 image’s colour characteristics.

REFERENCES

[1] Sanmati S. Kamath and Joel R. Jackson (2006). Color image segmentation in RGB using vector angle and Absolute difference measures. Proceedings of 14th European Signal Processing Conference (EUSIPCO 2006), Florence, Italy

[2] Navkirat Kaur, V. K. Banga, Avneet Kaur (2013). Image Segmentation Based On Color. Proceedings of IJRET: International Journal of Research in Engineering and Technology

[3] Shamik Sural, Gang Qian and Sakti Pramanik (2002). Segmentation and histogram generation using the HSV color space for Image retrieval

[4] Zhi-Kai Huang, De-Hui Liu (2007). "Segmentation of Color Image Using EM algorithm in HSV Color Space," in Information Acquisition, 2007. ICIA '07. International Conference on, vol., no., pp.316-319, 8-11 July 2007.

[5] Kanchan Deshmukh, IAENG, B. Ganesh Shinde (2006). Adaptive Color Image Segmentation Using Fuzzy Min-Max Clustering. Engineering Letters, 13:2, EL_13_2_2

[6] Krishna Kant Singh,Akansha Singh(2010), A Study of Image Segmentation Algorithms For Different Types Of Images‖, IJCSI International Journal of Computer Science Issues 2010

[7] Dibya Jyoti Bora, Anil Kumar Gupta (2014), Clustering Approach towards Image Segmentation: An Analytical Study‖, IJRCAR, Vol2, Issue 7, July 2014, pp. 115-124.

[8] A. Jurio (2010), A comparison study of different color spaces in clustering based image segmentation, IPMU (2010)

[9] Carron, T; Lambert, P. (1996) Symbolic Fusion of Hue-Chroma-Intensity Features for Region Segmentation. Proceedings of the 1996 IEEE International Conference on Image Processing, Vol. 1, pp. 971 - 974 vol. 2.

[10] Jie Wei (2001). Image Segmentation Using Situational DCT Descriptors. Proceedings of the 2001 IEEE International Conference on Image Processing, Vol. 1, pp. 738 - 741.

[11] Chung, R.; Yung N. and Cheung P. (2005). An Efficient Parameterless Quadrilateral-Based Image Segmentation Method, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 27, Nº9, pp. 1446 -1458.

[12] Roman Osorio, Waldo Olivera, M. Peña, I. López-Juárez, Jesus Savage, Gastón Lefranc., Using Background and Segmentation Algorithms Applied in Mobile Robots, MCPL 2013, 6th IFAC International Conference on Management and Control of Production and Logistics, (Volume #6, Part#1), Fortaleza - Ceará, Brasil.

Roman Osorio obtained a BEng from UNAM, Mexico. He has acted as consultant in several companies as well as published over 100 papers in related fields. His main research interests are in mobile robotics, automation, machine vision and software development. He is currently a researcher within the Computing Systems and Automation department in the Research Institute on Applied Mathematics and Systems (IIMAS-UNAM) and lecturer on Electronics.

Claudio Cubillos es Ph.D. in Information and System Engineering, Politecnico di Torino, Italia. Magíster en Ingeniería Industrial (PUCV), Ingeniero Civil Industrial (PUCV). Académico adjunto, Escuela de Ingeniería Informática, Pontificia Universidad Católica de Valparaíso. Sus líneas de interés son: Agentes inteligentes y sistemas multiagentes; ontologías y Web semántica; investigación de operaciones, con énfasis en problemas de routing and scheduling.

Ginno Millán (M' 2015) es Ingeniero Electrónico (2001) y Magister en Ciencias de la Ingeniería (2009), de la Pontificia Universidad Católica de Valparaíso, Chile, y Doctor en Ciencias de la Ingeniería (2013) de la Universidad de Santiago de Chile. En la actualidad es profesor de la Escuela de Ingeniería de la Universidad Católica del Norte. Su investigación se centra en el modelado de tráfico en redes de computadoras.

Gaston Lefranc received a BEng form the Universidad Técnica Federico Santa María in Chile and an MSc degree in Electrical and Computer Science at the Northwestern University, Evanston, Illinois, USA. Currently is Professor in Electrical Engineering at the Pontificia Universidad Católica de Valparaíso, Chile. His main research areas are Artificial Intelligence, Robotics, Flexible Manufacturing Systems and Biomedical Engineering. He has been co-author of 10 books and published over 160 papers in related fields. Currently is an IEEE Senior Member.