Color Objects Extracting from a Color Image Using Adaptive Filters

Chen-Chung Liu	Pei-Chung Chung	I-Li Chiu
National Chin-Yi University of Technology	National Chin-Yi University of Technology	National Chin-Yi University of Technology
Associate Professor	Student	Graduate
ccl@ncut.edu.tw	iam.abaw@msa.hinet.net	easycig@hotmail.com

摘要

本文中，我們只要透過物件的種子的指定（縱使是只有一個像素），本系統將全自動的使用 8 連通鄰域原則及當時當地之編碼像素的排列情況來產生一個接一個的適應性預測濾波器，這些濾波器將依循其自動自適應規則在 HSI 空間中，將組成物件的像素逐一檢測出來。將這些檢測出來的像素在對應的影像的 RGB 空間中還原出我們要擷取的物件。

為驗證本系統的功能，許多各式各樣的彩色影像被我們拿來做試驗，而試驗的結果更證明了縱使我們只給它物件的一個像素，
本系統能更有效的從 RGB 彩色影像中快速無誤地將所要的物件擷取出來。

關鍵字：RGB 域, HSI 域, 可適性, 物件擷取

Abstract

Image segmentation can be put for a great variety of uses; good examples would be pattern recognition and medical image process. In this article, by appointing seeds for the desired object (even if the seed consists of only one pixel), our system can automatically generate a series of adaptive filters based on “8 connected neighborhood rule” and the object’s pixel arrangement. The filters will then be able to identify the pixels which resemble the desired object from the HSI plane. The identified pixels can then be used to reconstruct the original desired object on the RGB plane.

In order to test our system’s practical function, we have experimented with various color images. The experimental results proved that our system can efficiently extract the desired object from a color image even if only one single seeding pixel was used.

Keyword: RGB (Red, Green, and Blue), HSI (Hue, Saturation, and Intensity) domain, adaptive, object extraction.

1.Introduction

Image segmentation has a variety of purposes. For example, segmentation plays an important role in the field of video object extraction [1]. In order to extract the moving object and a homogeneous region always corresponds to a meaningful object, many video object extraction algorithms partition the image into homogeneous regions at first, and then, the regions are merged according to temporal information of the sequence. In image compression [2], the input image is divided into regions that should be separately compressed since better compression is achieved as long as the regions are more homogeneous. Tracking systems that are region-based [3] utilize the information of the entire object's regions. They track the homogenous regions of the object by their color, luminance or texture. Then, a merging technique that is based on motion estimation is used in order to obtain the complete object in the next frame. Image segmentation is also used in object recognition systems [4]. Many of these systems partition the object to be recognized into sub-regions and try to characterize each separately in order to simplify the matching process.

The segmentation of an image is a process of dividing an image into non-overlapping regions which are homogeneous group of connected pixels consistent with some special criteria. There are lots of ways to define the homogeneity of a region in the segmentation process. For example, it may be calculated by color, depth of layers, gray levels, motion, texture, etc. Each pixel belongs to a single region only owing to overlaps among regions is inhibited. The neighboring regions should be merged while the new combined region is homogeneous. Accordingly, each region should be as large as possible under its certain characterization. And, the total number of regions is reduced.

Pal and Pal [5] provided a review on various segmentation techniques; they say that segmentation has no standard approach. Many different types of scene parts can serve as the segments on which descriptions are based, and there are many different ways in which one can attempt to extract these parts from the image. The most appropriate segmentation technique depends on the characteristic of an image and its applications. The most commonly used approaches in still image can be divided into the following main approaches: histogram threshold [6], feature/color space clustering [7], edge detection approaches [8], neural network based approaches [9], region-based approaches [10], Markov random field [11] and mixture-of-Gaussians modeling [12], physics-based approaches [13], and combinations of above [14, 15]; Where Histogram-based methods, boundary-based methods, region-based methods, hybrid-based methods and graph-based techniques are primarily five types of segmentation techniques. The histogram-based algorithms consider a gray level image as a one dimensional histogram; it is treated as a Gaussian probability density function and the segmentation problem become a parameter estimation followed by pixel classification. Most of the histogram-based segmentation methods were used to specific images; it achieved reasonable performance when the input is characterized without noise and with small number of regions. It does not consider spatial details and does not work well for images without obvious peaks and valleys. Boundary-based segmentation methods search for pixels that lie on edges. The boundary-based methods' drawback is the over-segmentation result, which does not always correctly reflect the image nature. It is quite sensitive to noise and do not work well for images containing ill-defined edges. Region-based segmentation methods assume that pixels belong to the same homogeneous region is more alike than pixels from different homogeneous regions. And gather similar pixels according to some homogeneity criteria. The Region-based segmentation methods' drawbacks are strongly dependent on global pre-defined homogeneous criteria thresholds. And it is quite expensive in computation time and sensitive to the examination order of regions and pixels. The hybrid technique for segmentation combines the region-based method and the boundary-based method or the region-based method and the histogram-based methods. It become very common due to it relies on wide information as global (histogram) and local (regions and boundaries). The graph-based segmentation typically constructs a graph in which the nodes represent the pixels in the image and arcs represent affinities between nearby pixels. It segment an image by minimizing the weight, which is associated with cutting the graph into sub-graphs, the weight is the sum of the affinities across the cut.

Many papers about segmentation of images using color have been proposed. Mena and Malpica use the statistical characteristics of color texture in RGB space to do the color image segmentation [16]. Park et al. [17] proposes an algorithm based on mathematical morphology which performs a clustering in 3D color space. They all get a good segmentation for color image. In this paper, we are going to focus on color image segmentation, since it is very important and convenient in objects image recognition and in wide range of applications such as: automated surveillance, images retrieval from a large database and classification of species.

In this paper, we combine a modified SLCCA algorithm which utilizes eight-connectivity, assigned seeds, and compare neighbor pixels to construct a system of color objects extracting from a RGB image. The remainder of this paper is organized as follows: Section 2 is the simple discussion of color image segmentation and object extraction. The proposed color objection algorithm and its detail description are presented in Section 3. Empirical tests are presented in Section 4. Finally, we conclude this paper in Section 5.

2. Color image segmentation

Due to the reason that the pixels constituting the meaningful regions of an image share the same feature characteristics, we consider color uniformity as a criterion for segmenting the image into disjoint regions corresponding to different objects in the original color image.

The color image segmentation techniques can be classified into two classes; analysis in the image domain and analysis in the color space [18-19]. The analysis in the image domain is developed under the assumption that adjacent regions representing distinguishable objects have detectable discontinuities in their local properties or that regions are more or less uniform in their local properties. This can be achieved with edge detection and region reconstruction. Where, edge finding is the process of detecting the parts of an image where there are abrupt changes in local color properties, and region reconstruction a region growing process or a split and merge process. The analysis in the color space is developed on the analysis of the pixel properties in the color space with decomposing each pixel by its three components (the red, the green and the blue), and assumed that (a) Any color is possible to match by mixing appropriate amounts of the three primary colors. (b) Homogeneous regions in the image plane are collected to clusters in the color space, each cluster corresponding to a class of pixels which have the same color properties.

Many authors have proved the RGB color space is the more appropriate for segmentation problems. Brunner shows that detecting specific surface defects on Douglas-fir veneer is well detected when the clustering analysis is performed in the RGB space. Littman also concludes that the RGB space is the better than other space in objects classification. In this paper, we propose a modified SLCCA algorithm which uses generated eight-connectivity, assigned seeds and compare neighbor pixel to construct a system of color objects extracting from a RGB image. Fig. 1 is the algorithm flow chart of our system.

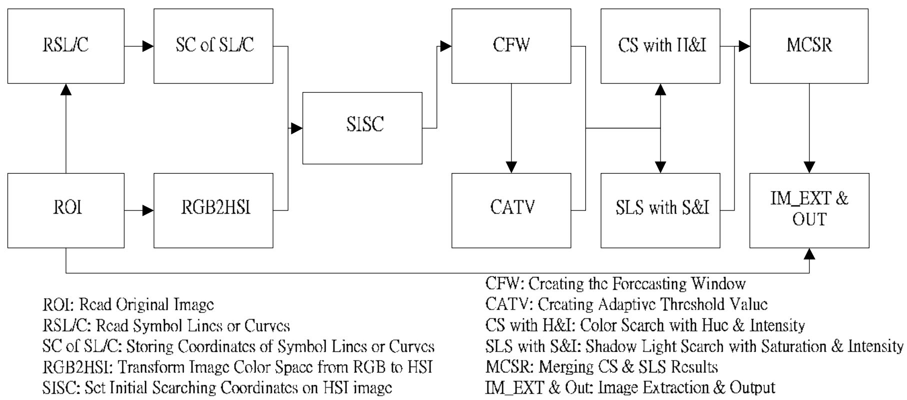


Fig. 1 The algorithm flow chart of our Color objects

3. Color Objects Extraction Algorithm

The overall segmentation procession of our proposed scheme is shown in Fig. 2. The input RGB color image is transformed into HSI color model and decomposed into H plane, S plane and I plane firstly. Owing to the H, S, I planes are measured with different units individually, they all need different methods for distinguishing the objects and the background. In order to create a precision color objects extraction scheme, seeds assigning, RGB to HSI transformation, comparing neighbor criterion, and a modified SLCCA algorithm that utilizes eight-connectivity are use to enhance the correctness of the extraction. Finally, a merge mechanism and planes union mechanism are used to get the color objects extracted from a RGB color image. The details of our segmentation algorithm are described in the following.

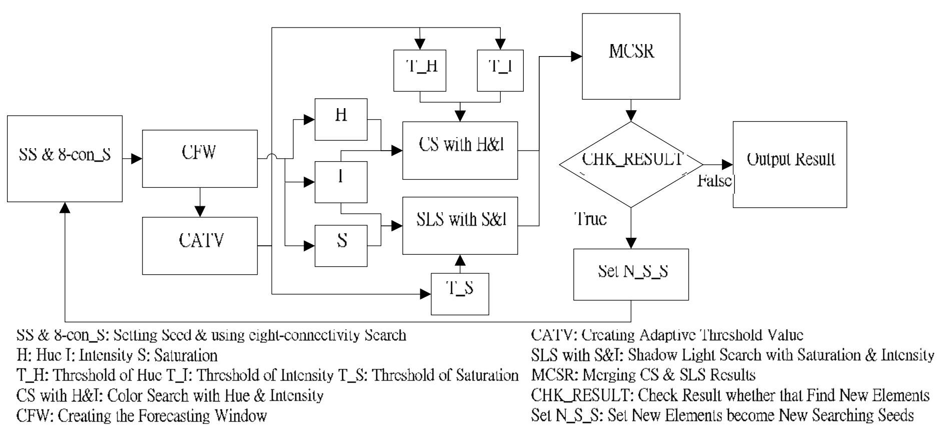


Fig. 2 The flow chart of the object extraction algorithm

3.1 Color Model Transformation from RGB to HIS

The basic understanding of color perception of the human visual system shows the fact, which human eyes pick out the color of an object illuminated by white light is the reflection of selected wavelengths of light by that object [20-21]. The appearance color of an object can be considered as the object absorbing all colors except the colors that are reflected. Although there are an unlimited number of colors, colors of the range of human visual sensations can be produced by the mixtures of finite visual lights of various wavelengths. Human eyes have three cones shaped light detectors [22]. Each cone is sensitive to a special range of colors. One is sensitive to primarily red, a second to primarily green, and a third to primarily blue. The three sensitivity ranges overlap in fact. Each cone transmits a nerve pulse to the brain at a rate proportional to the light's intensity of that cone detects. By combining the signal rates transmitted from three cones, the brain attempts to conclude what color and brightness of the light must be. Thus, to combine various quantities of three colors (red, green and blue) of light can turn out all colors that can be perceived for human.

In televisions, computer monitors, and colored image projection systems, only the three colors are enough to adequately represent any of the unlimited visible colors [23]. For measuring or reproducing color, a number of three dimensional color models have been defined, among which most popularly used is RGB (Red, Green, and Blue) model. The RGB model is a physical system and the image in RGB model is the most suitable for the color representation of a color image. Most imaging devices (CCD cameras, CRT monitors, etc) mimic the retina and utilize three separate channels calibrated for the detection of red, green and blue phosphors to render colors at every pixel on the screen [24]. But it is not suitable for image processing applications, because its *R*, *G*, and *B* components are highly correlated. Besides, the distance in RGB color space does not stand for the perceptual difference in a uniform scale. In image processing and analysis, these *R*, *G*, and *B* components are often transformed into other color models. The HSI (Hue, Saturation, and Intensity) color model is the most representative of the perceptual systems, which is widely used in image processing. The advantages of HSI model are: hue and saturation are good correlation with the human perception of colors, and its separability of chromatic values from achromatic values. The HSI is the reducing redundancy models of the RGB model, its components H(hue), I(intensity), and S(saturation) are given by some color transform from RGB color space[25- 27]

$$H=\begin{cases}\theta & if \quad B\le G\\ 360-\theta & if \quad B>G\end{cases} \tag{1}$$

$$\theta=\cos^{-1}\left\{\frac{\frac{1}{2}[(R-G)+(R-B)]}{[(R-G)^2+(R-B)(G-B)]^{1/2}}\right\} \tag{2}$$

$$S=1-\frac{3}{(R+G+B)}[\min(R,G,B)] \tag{3}$$

$$I=\frac{1}{3}(R+G+B) \tag{4}$$

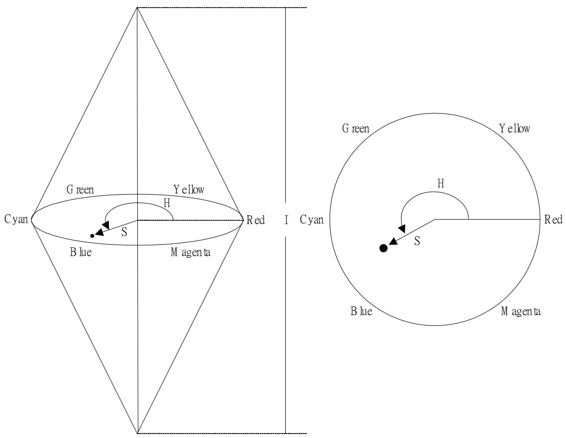


Fig.3 The HSI color model used polar coordinates

The HSI color model uses polar coordinates, as shown in Fig. 3. In the double-cone shape figure, the saturation is proportional to radial distance (the center of the HSI disk corresponds to zero saturated colors) and it corresponds to the relative purity of a color. The hue is defined as an angle measured from the reference line drawn from the center of the HSI disk to the red vertex; it stands for the dominant wavelength in a mixed light and indicates a dominant color as perceived by human. Finally, the intensity or perceived lightness is the distance along the axis perpendicular to the HSI disk plane, and it indicates the brightness of a color.

3.2 Object Extraction on HSI Planes

How to determine whether the neighbor pixels belong to the object or not? Owing to each pixel has its H, S, and I values, and they have their value ranges individually. It's not easily to identify the pixel is a element of the object or not. In this paper, we develop a comparing neighbor pixels criterion (CNPC) and generated eight-connectivity to distinguish the object and the background. We start from a seed and set thresholds for the H component, S component, and I component. Then we compare the seed pixel with its neighbor pixels in each plane, if the difference of the seed pixel and the neighbor pixels is less than the threshold then the neighbor pixels belong to the object; otherwise they belong to the background. The algorithm repeats these steps for each seeds. According this rule and using the eight-connectivity modified SLCCA method to check each pixel of an image on each plane, and then the object extraction on each plane is achieved correctly.

We adopt the HSI color model for color objects extraction, due to the following reason: (A) to collect pixels that are the same hue in HSI color model is easier in other color model. (B) The Intensity component of HSI is high at bright pixel and low at dark pixel. The saturation component of HSI is low at bright pixel and high at dark pixel. So, we combine the properties of the Intensity component and the saturation component to measure the variety of light and shadows in image, and easily find out the regions of light's reflection and shadow regions from an image.

In HSI color model, we can precisely and quickly determine whether two pixels are the same color or not just by their hue component only, but you have to measure R, G, and B components in the same time in RGB color model. In HSI color model, two adjacent primary colors are spaced at 120°, two adjacent sub-colors are spaced at 120° also. You may give the hue distance lager than 3° and less than 10° to check whether two pixels are monochrome or not. You also can use the various shades of color caused by intensity component's effect to determine the region of an object correctly.

The threshold value of the intensity

component shall be set no less than 0.03 and no larger than 0.1. The intensity component is just use to assist the hue component for more precision extraction, so we give the intensity threshold more flexibly.

Extracting objects from saturation plane and the intensity plane uses the reversing property between the saturation component and the Intensity component in the pure white region or in the pure dark region. We use the difference of saturation and intensity to determine whether two pixels belongs to a similar luminosity or not, this method is very suitable in the white regions with Intensity larger than 0.8, or in the dark regions with Intensity less than 0.2. The hue distortion is very large for these regions that very bright or very dark in RGB color model, this distortion should make intensity can't work well in object extraction for over bright or over dark objects. So, we extract over bright or over dark objects from the saturation plane and the intensity plane.

3.3 The adaptive forecasting window

With the exception of some special cases about pattern recognition, most of pattern extractions from a color image distinguish different objects with their own colors. In order to extracting objects quickly and simply from a color image, one has to overcome the influence created by the light on the image. There are some shadows and the luminescent spots along with the light illumination on objects. The searching effect of searching objects from an image is going to be extremely limited while a person only adopts the color characteristic of an object as the measuring standard. In the recent time, the majority searching methods need to make the manual parameter hypothesis for the different image and different object. Therefore, we use the permutation of 8- connect neighborhood pixels to forecast the HSI values and to calculate the corresponding threshold values for correlation pixels automatically without any manual effort.

The construction of the forecasting window is shown in the following figures, where black blocks denote the encoded pixels, the white blocks represent the no-encoded pixels, and the numbers on the each block represent the order (priority) of construction respectively; "0 "must be a encoded point because it is the point that we paint a mark on it and we want to find it from the image first. On the other hand, the 8- connect neighborhood pixels of pixel 0 may be an encoded pixel or not respectively. We calculate the HSI values of these 8- connect neighborhood pixels of a no-encoded pixel one by one according to the permutation of those encoded pixels in the 3*3 block with centering at the encoded pixel. We adopt these new values to form a forecasting window and use it as a filter to filter the original 3*3 block, the way carries on the searching till whole pixels of the window are calculated.

The forecasting window construction is dependent on the permutation of encoded pixels of the original window. There are 7 fundamental types of the permutation of encoded pixels of the 3*3 window; each forecasting window's construction is the combination of the 7 fundamental types. The 7 fundamental types' HSI forecasting of the window's pixels are shown in Fig.4 ,

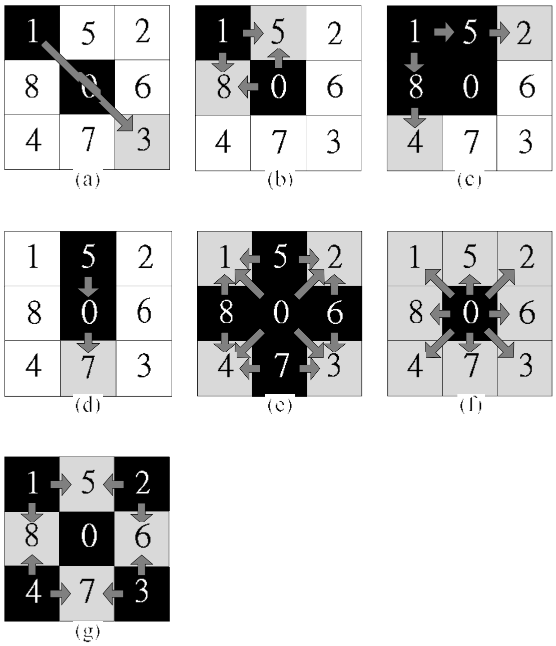


Fig.4 The 7 fundamental types' HIS forecasting of the window's pixels.
(a) opposite pixel on the window's diagonal
(b) pixel adjacent to a corner pixel
(c) last pixel of a side with the other pixels on the same column(row) pixels element
(d) opposite side's middle pixel
(e) corner pixel
(f) Periphery does not have any symbol (it is always the starting seed pixel)
(g)between's pixel with the two side's corner pixels

the calculation methods are described as the following formulas: where H', S', and I' are the hue, the saturation, and the intensity of the adaptive window, H, S, and I are the hue, the saturation, and the intensity of the original window.

a. The forecasting of the opposite pixel on the window's diagonal

$$\begin{cases} H'(3) = \{ H(0) + H(1) \}/2 \\ S'(3) = 2 * S(0) - S(1) \\ I'(3) = 2 * I(0) - I(1) \end{cases} \quad \textbf{(5)}$$

b. The forecasting of the pixel adjacent to a corner pixel

$$\begin{cases} H'(5) = \{ H(1) + H(0) \}/2 \\ S'(5) = \{ S(1) + S(0) \}/2 \\ I'(5) = \{ I(1) + I(0) \}/2 \end{cases} \quad \textbf{(6)}$$

c. The forecasting of the last pixel of a side with the other pixels on the same column(row) pixels element

$$\begin{cases} H'(2) = \{ H(1) + H(5) \}/2 \\ S'(2) = 2 * S(5) - S(1) \\ I'(2) = 2 * I(5) - I(1) \end{cases} \quad \textbf{(7)}$$

d. The forecasting of the opposite side's middle pixel

$$\begin{cases} H'(7) = \{ H(0) + H(5) \}/2 \\ S'(7) = 2 * S(0) - S(5) \\ I'(7) = 2 * I(0) - I(5) \end{cases} \quad \textbf{(8)}$$

e. The forecasting of the corner pixel

$$\begin{cases} H'(1) = \{ H(0) + H(5) + H(8) \}/3 \\ S'(1) = 2 * S(0) - S(5) \\ I'(1) = 2 * I(0) - I(5) \end{cases} \quad \textbf{(9)}$$

f. Periphery does not have any symbol (it is always the starting seed pixel)

$$\begin{cases} H'(i) = H(0), i = 1, \ldots, 8 \\ S'(i) = S(0), i = 1, \ldots, 8 \\ I'(i) = I(0), i = 1, \ldots, 8 \end{cases} \quad \textbf{(10)}$$

g. The forecasting of the between's pixel with the two side's corner pixels

$$\begin{cases} H'(5) = \{ H(1) + H(2) \}/2 \\ S'(5) = \{ S(1) + S(2) \}/2 \\ I'(5) = \{ I(1) + I(2) \}/2 \end{cases} \quad \textbf{(11)}$$

When the merging process is started, it has to be decided when the process must be ended. In other words, it does not know how to determine which pixels should not be merged and when [4, 28- 30]. We introduce an automatic procedure that derives local thresholds according to the changes of each region during the merging process. The merging machine is going to decide whether or not a certain region should be merged by using these thresholds.

Since we regard the segmentation process as a local operation, we can presume that not all the local merges will be done at the same time. Although there are some special emerging cases where one global threshold is sufficient; for example, the image only contains one uniform colored texture object and so is its surrounding background. To use a global threshold on entire image is not suitable because different regions are separated from their surroundings while go on the emerging process with different thresholds. In most of the cases the image contains more than one homogenous regions, it is obvious that it is difficult to segment accurately the input image with one global threshold. In order to prevent the over merged and over-segmented. The calculation of local thresholds has to be based on local characteristics, which is correlated to the regions and their surroundings. We propose a method that calculates adaptive local thresholds automatically. Our system constructs the forecasting window according to the permutation of encoded pixels of the original window first. The forecasting window then uses its each direction's gradient of its cent to calculate the local threshold of the window.

The forecasting window uses its each direction's gradient of its center to calculate the window's threshold of hue H_t , threshold of saturation S_t , and threshold of intensity I_t . They are calculated by following formulas:

$$H_t = \min\{\max(abs[H'(i) - H(0)]_{i=1}^{8}), \varepsilon_H\} \quad \textbf{(12)}$$

$$S_t = \min\{\max(abs[S'(i) - S(0)]_{i=1}^{8}), \varepsilon_S\} \quad \textbf{(13)}$$

$$I_t = \min\{\max(abs[I'(i) - I(0)]_{i=1}^{8}), \varepsilon_I\} \quad \textbf{(14)}$$

Where ε_H is the empirical constant of hue, ε_S is the empirical constant of saturation, and ε_I is the empirical constant of Intensity.

The hue of a color is not a linear function of R, G, or B, it is not also sensitive with respect to R, G, or B. along with the progression of emerging, due to the accumulating of each filtering window threshold value, the hue difference between the starting seed pixel and the final encoded significant pixel may exceed our tolerance. To avoid over encoding, we take double checks for hue on each filtering, we check the hue of the new encoded pixel with the hue of the starting seed pixel to guarantee the precession of segmentation. Fig. 5 shows an example to demonstrate that our adaptive forecasting filter how to extract the objects from a color image on HIS domain.

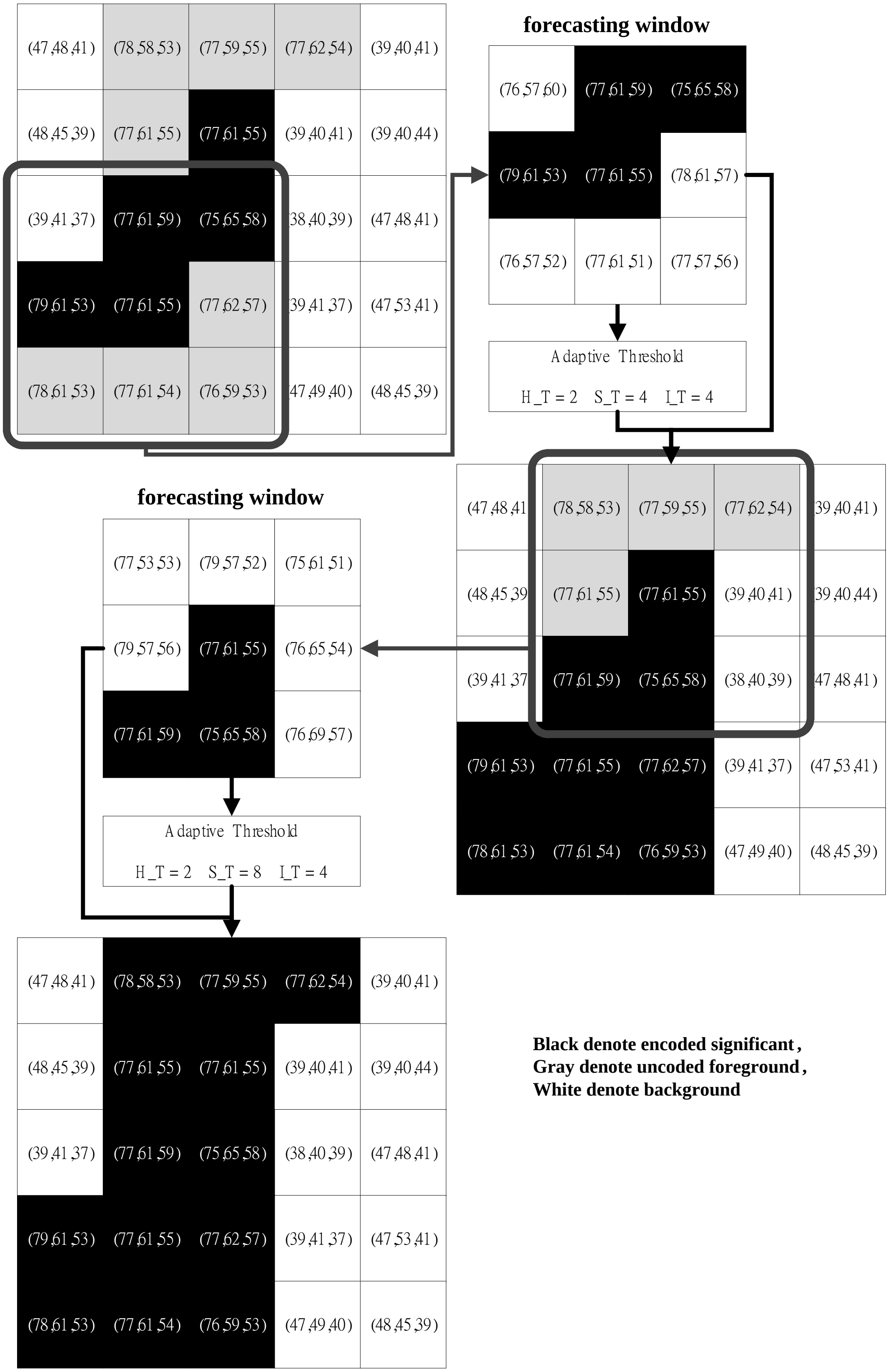


Fig. 5 an example of our adaptive forecasting filter 's working

3.4 Region Merging and Planes Union

The SLCCA structure in Chai et al. [19] is used to improve the coding efficiency. But in this paper, it is hired to construct a fast and effective region segmentation scheme. However, we modify the representation of the SLCCA as follow: (a) if the neighbor pixel is similar to the seed pixel, then they are in the same cluster denoted "same" (S), otherwise denoted "different" (D), and N represents that it has been processed already. (b) Each seed pixel has 8 neighboring pixels, therefore has 8 values S, D or N regularly. (c) According to (a) and (b) we start from a seed to examine the seed and record its neighboring pixels, and then repeat the same steps to examine all pixels of an image, finally we connect these (S) that are grouped together continuously to form a region. In SLCCA structure, it connects the same cluster in generated 8-connectivity. But in this paper, all neighbor pixels are 256 grayscale. The condition of the same cluster is that their grayscale level difference is less than the threshold value V_{th}. Beside, we proposed two situations for improving the region based segmentation performance. One; while the pixel is located in the edge then we decrease the threshold value from V_{th} to V_{th1}. Because the edge is used to distinguish the different regions, so we must decrease the decision value to depart the cluster correctly. Two; we record the minimum value and maximum value of the class at a new class beginning. While the maximum value is bigger than the minimum value with a threshold V_{th3}, then we decreased the threshold value from V_{th} to V_{th2}. This condition can guarantee the grayscale difference of the cluster in a safe range.

A HSI color image is constructed by the H plane, the S plane, and the I plane. For the performance, we extract the color object from the H plane, the S plane, and the I plane respectively. Once all the H plane, the S plane, and the I plane are segmented into object-region and background-region, and then we need to combine the three parts to form the extracted color objects on original RGB color image. The relationship between the color image and the H plane, the S plane, and the I plane segmentation is described in Fig. 6. In Fig. 6(a), the H plane segments the image into object Ho(colored in blue) and background Hb(colored in white). In Fig. 6(b) the S plane divided the image into object So(colored in red) and background Sb(colored in white). In Fig. 6(c), the I plane is separated into object Io(colored in green) and background Ib(colored in white). After the union operation, the really extracted object region is described in Fig. 6(d).

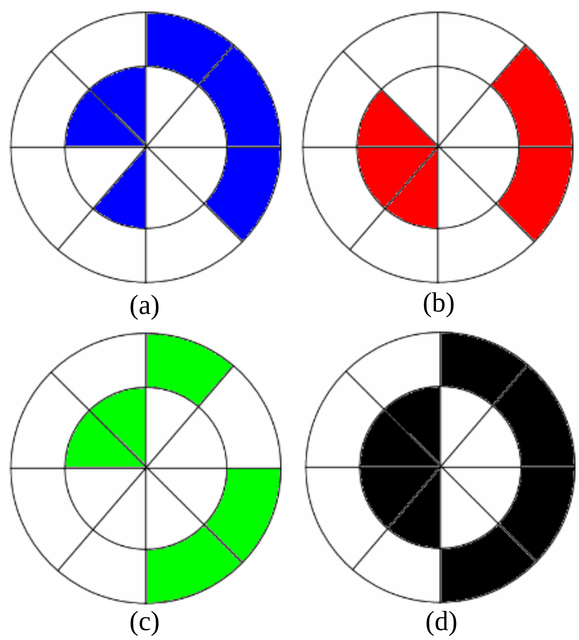


Fig. 6 The schematic of the union mechanism in a HSI color image: (a) extracted objects on the H-plane, (b) extracted objects on the S-plane, (c) extracted objects on the I-plane, (d) the result object after union mechanism.

4. Empirical tests

Several images of the input images with different sizes are used in simulation for demonstrating the performance of the proposed scheme. We take 4 examples to describe the empirical tests of our proposed algorithm, and conclude them with a table.

a. hat(325×415) :

Here we will use this picture as a successful demonstration of our method. By using a single tracking point, we were able to extract the desired object from the picture – the pink hat. The pink hat in this case was easily extracted by a simple search based on the hue component only. It was a feasible mechanism because of the pink hat's constant and narrow distribution of hue, and also no great diversion of lightness and darkness. In addition of the hue component, we also used saturation and intensity components to help extracting a full desired object where the hue distribution became more diversed and less continuous.

b. pottery (350x251)

As shown on the figure here, the pottery has a wide range of colour changes. However, the range of colours can be sorted under a general kind of colour group so it is not too difficult to identify the desired object via the hue component. In the meantime, we also used more tracking points to improve the extracting quality in this case – i.e. when there is a greater variety of hues between the pixels of the desired object.

Also worth of noting is that : the darker coloured drawings on the pottery can be isolated from the object by utilising the adaptive threshold. However, because this particular picture has undergone compression before we extracted the object, several colour noises have appeared on the upper portion of the picture and made some pixels unable to be extracted.

c. girl(256x256)

The tricky part of this particular picture was the dark shadowing between the girl's arm and body. Because of the greater variety of the shadow in colour, if we manually set the extracting thershold of I_T greater, it will make the image extraction cover the unwanted background as well as the desired object. Here, by using the adaptive forecasting filter to automatically adjust for the extracting thershold, we were able to extract a full object including both the body and the arms precisely even when the tracking points did not cover across the shadow between the arms.

d. flowers(172x222)

This picture showed 3 marked flowers. Two of them consisted of simple colors, but with numerous shadows between the petals. One difficulty with extracting object based on the hue component is that the mechanism usually doesn't perform as well on areas which are predominantly black or white. However, by the use of adaptive forecasting window (which anticipates the changes of color); it makes a great improvement in its extracting performance of the white flower in this case. Finally, by using various different extracting thresholds corresponding to different portions of the image, this enables us to extract the red flower (which has got a great shadow variety) via only a single straight tracking line.

e. purple flower (800x600)

This is a picture with higer resolution, and therefore greater variety of color changes of the flower shown. By using the adaptive forecasting window (which anticipates the surrounding pixel colors), we were still able to extract the object rapidly because of our straight forward calculating mechanism.

Several performance metrics for measuring the efficiency of a segmentation algorithm have been proposed. They are misclassification error (ME), relative foreground area error (RFAE), edge mismatch (EM), and shape distortion penalty (SDP) [31]. The misclassification error evaluates the inaccuracy of an algorithm; it is defined as in the following formula:

$$ME = 1 - \frac{TP + TN}{TP + FN + TN + FP} = \frac{FN + FP}{TP + FN + TN + FP} \quad \textbf{(15)}$$

, where TP Indicates the areas of true positive, TN represents the areas of true negative, FP indicates the areas of false positive, and FN represents the areas of false negative respectively. **Fig.7** shows the relations of TP, TN, FP, and FN areas in prediction and target area.

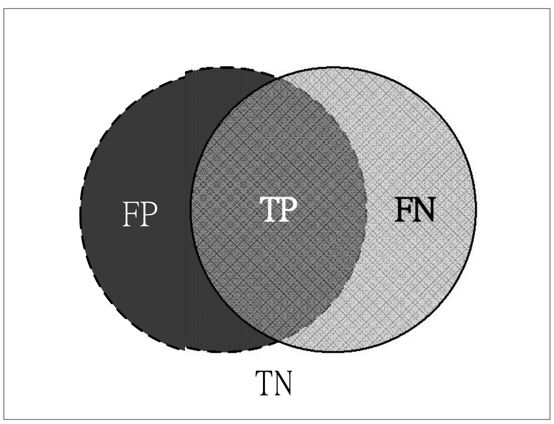


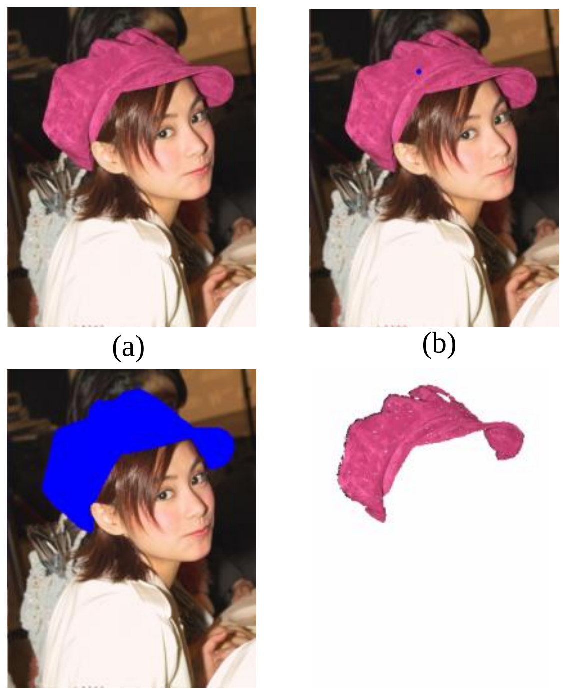
Fig.7 The relations of TP, TN, FP, and FN areas in prediction and target area.

The relative foreground area error (RFAE) evaluates the quality of an algorithm; it is defined as in the following formula:

$$RFAE = \begin{cases} \dfrac{TP+FN-(FP+TP)}{TP+FN} = \dfrac{FN-FP}{TP+FN} & if\ (FP+TP) < (TP+FN) \\ \dfrac{FP+FN-(TP+FN)}{FP+TP} = \dfrac{FP-FN}{FP+TP} & if\ (FP+TP) \geq (TP+FN) \end{cases} \quad \textbf{(16)}$$

Where FP+TP indicates the segmented object while TP+FN denotes the ground-truth object. It can be observed that RFAE is 1 for the worst case of segmentation while it is zero for a perfect segmentation.

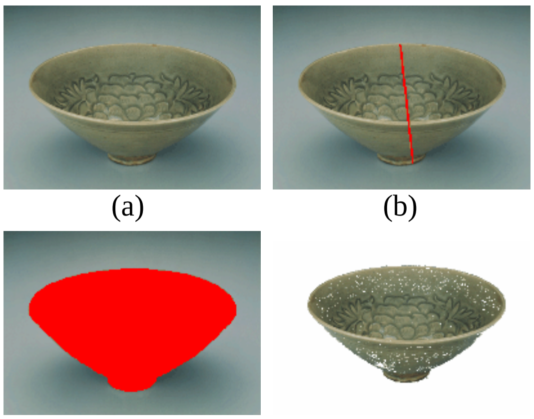
The results images were obtained by through several processes and described in the following. Fig. 8~ Fig. 12 are the performances of the proposed scheme, and table 1 is the table of the statistical results of object extractions at different conditions. The items in Fig. 8 ~ Fig. 12 are: (a) The original input color image. (b) The symbol lines (or points) that we sketched on the original test image. (c) The real object that we want to extract. (d) The extraction object using our algorithm.



(a) (b)

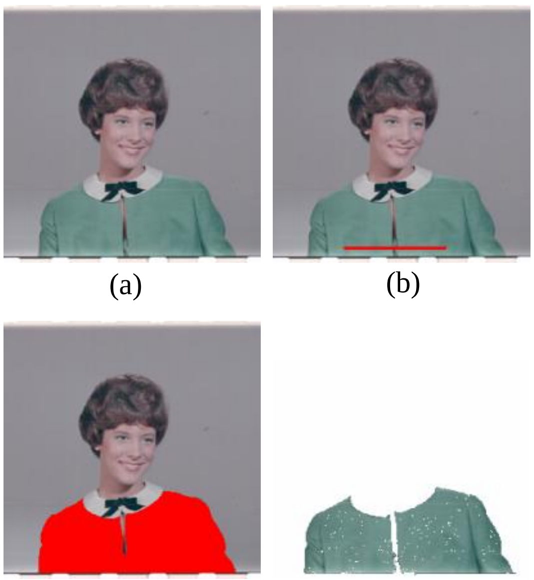
(c) (d)

Fig. 8 test image 1(hat)



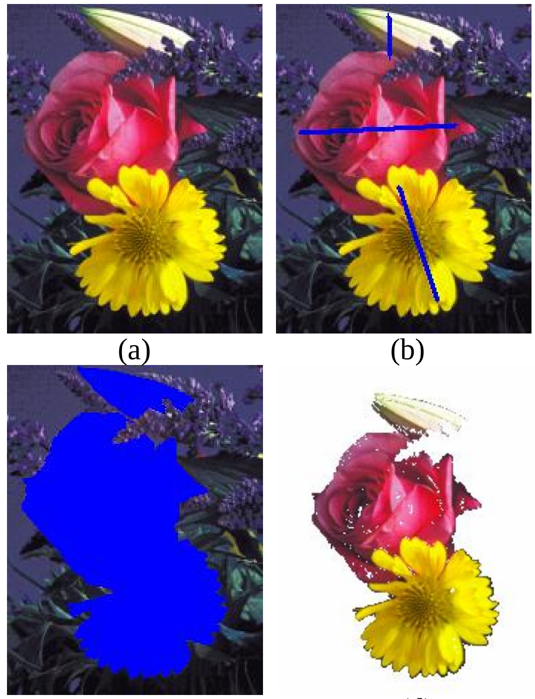
(a) (b) (c) (d)

Fig. 9 test image 2(pottery**)**



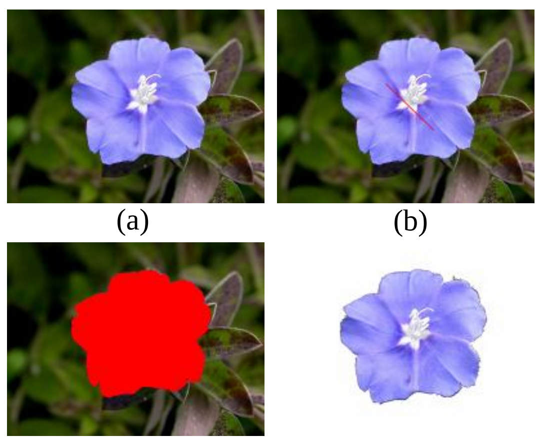
(a) (b) (c) (d)

Fig. 10 test image 3(girl)



(a) (b) (c) (d)

Fig. 11 test image 5(flowers)



(a) (b) (c) (d)

Fig. 12 test image 4(purple flower)

Table 1: The empirical results

Test image	Image size	No. of seeds	Object size	No. of extraction	ME	RFAE
Hat	325x415	1	18787	18042	0.0026	0.0397
pottery	350x251	162(1line)	31959	30824	0.0054	0.0355
Girl	256x256	100(1line)	11665	11490	0.0015	0.0150
Flowers	172x222	214(3lines)	16503	16362	0.0062	0.0085
purple flower	800x600	146(1line)	128584	127962	0.0026	0.0048

5. Conclusion.

Several images were used to demonstrate the efficiency of the proposed algorithm, and the experiment results showed that our proposed algorithm can precisely find the real sub-regions from color images. In this article, by appointing seeds for the desired object (even if the seed consists of only one pixel), our system can automatically generate a series of adaptive filters based on "8 connected neighborhood rule" and the object's pixels arrangement. The filters will then be able to identify the pixels which resemble the desired object from the HSI plane. The identified pixels can then be used to reconstruct the original desired object.

Reference

[1] Gao H., Siu W.-C., Hou C.-H., "Improved techniques for automatic image segmentation", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 11, Issue: 12, 2001

[2] Belloulata K., Konrad J., "Fractal Image compression with region-based functionality", *IEEE Transactions on Image Processing*, Vol. 11, Issue: 4, pp. 351-358, 2002

[3] Ozyildiz E., Krahnst-over N., Sharm R., "Adaptive texture and color segmentation for tracking moving objects", *Pattern Recognition*, Vol. 35, pp. 2013-2029, 2002

[4] E. Navon, O. Miller, A. Averbuch, "Color image segmentation based on adaptive local thresholds', Image and Vision Computing", Vol: 23 Issue: 1, January 1, pp: 69-85, 2005

[5] Pal N.R., Pal S.K., "A review on image segmentation techniques", *Pattern Recognition*, Vol. 26, Issue: 9, pp. 1277-1294, 1993

[6] Kurugollu F., Sankur B., Harmanci A., "Color image segmentation using histogram multithresholding and fusion", *Image and Vision Computing*, Vol. 19, pp. 915-928, 2001

[7] Li C.H., Yuen P.C., "Regularized color clustering in medical image database", *IEEE Transactions on Medical Imaging*, Vol. 19, Issue: 11, pp. 1150-1155, 2000

[8] Xu L., Jackowski M., Goshtasby A., Roseman D., Bines S., Yu C., Dhawan A., Huntley A., "Segmentation of skin cancer images", *Image and Vision Computing*, Vol. 17, pp. 65-74, 1999

[9] Huang H., Chen Y., Hsu W., "Color image segmentation using a self-organizing map algorithm", *Journal of Electronic Imaging*, Vol. 11, Issue: 2, pp. 136-148, 2002

[10] Cheng H.D., Jiang X.H., Wang J., "Color image segmentation based on hamogram thresholding and region merging", *Pattern Recognition*, Vol. 35, pp. 373-393, 2002

[11] Gao J., Zhang J., Fleming M., "Novel technique for multiresolution color image segmentation", *Optical Engineering*, Vol. 41, Issue: 3, pp. 608-614, 2002

[12] Greenspan H., Goldberger J., Eshet I., "Mixture model for face-color modeling and segmentation", *Pattern Recognition Letters*, Vol. 22, pp. 1525-1536, 2001

[13] Onyango C.M., Marchant J.A., "Physics -based colour image segmentation for scenes containing vegetation an d soil", *Image and Vision Computing*, Vol. 19, pp. 523-538, 2001

[14] Chen T.Q., Lu Y., "Color image segmentation—an innovative approach, *Pattern Recognition*", Vol. 35, pp. 395-405, 2002

[15] Mirmehdi M., Petrou M., "Segmentation of color textures", *IEEE Transactions on Pattern Analysis Machine Intelligence*, Vol. 22, Issue: 2, pp. 142-159, 2000

[16] J.B. Mena, J.A. Malpica, "Color image segmentation based on three levels of texture statistical evaluation", Applied Mathematics and Computation, Vol: 161 Issue: 1, February 4, pp: 1-17, 2005

[17] Park, Jaehyun; Kurz, Ludwik, "Unsupervised Segmentation of Textured Images" Information Sciences Vol. 92, Issue: 1-4, July, pp. 255-276, 1996

[18] Cheng H.-D., Jiang X.-H., Sun Y., Wang J., "Color image segmentation: advances and prospects", *Pattern Recognition*, Volume: 34, Issue: 12, pp. 2259-2281, 2001

[19] B. B. Chai, J. Vass and X. Zhuang, "Significance-Link Connected Component Analysis for Wavelet Image Coding," IEEE Trans. on Image Processing, Vol. 8, No. 6, pp.774-784, 1999.

[20] S. S. Deeb, "Genetics of variation in human color vision and the retinal cone mosaic", Current Opinion in Genetics & Development, Vol: 16 Issue: 3, pp: 301-307, 2006

[21] Lennie, Peter, "Color vision: Putting it together", Current Biology, Vol. 10, No.16, pp. R589-R591, 2000

[22] S. D.Buluswar, B. A. Draper, "Color Models for Outdoor Machine Vision", Computer Vision and Image Understanding, Vol.85, No.2, pp. 71-99 , 2002

[23] C. Clausen and H. Wechsler, "Color Image Compression Using PCA and Backpropagation Learning, " pattern recognition, Vol. 33, pp1555-1560, 2000

[24] T. Bose, "Digital signal and image processing", 2004, J. Wiley & sons,New York

[25] F. Ortiz and F. Torres, "Vectorial morphological reconstruction for brightness elimination in colour images", Real-Time Imaging, Vol. 10,No. 6, pp. 379-387, 2004

[26] Arthur, R. Weeks "Fundamentals of Electronic Image Processing", Spie Optical Engineering Press, Washington USA, 1998

[27] F. Ortiz and F. Torres, "Identification of citrus disease using color texture features and discriminant analysis," Computers and Electronics in Agriculture, Vol. 52, No. 1-2, pp.49- 59, 2006

[28]Jo˜ao P.V. Madeiro , Paulo C. Cortez, Francisco I. Oliveira, Robson S. Siqueira, "A new approach to QRS segmentation based on wavelet bases and adaptive threshold technique", Medical Engineering & Physics 29 (2007) 26–37

[29]K.V. Asaria,, T. Srikanthana, S. Kumarb, D. Radhakrishnana, "A pipelined architecture for image segmentation by adaptive progressive thresholding", Microprocessors and Microsystems 23 (1999) 493–499

[30] Y. K. Do, W. P. Jong, "Connectivity-based local adaptive thresholding for carotid artery segmentation using MRA images", Image and Vision Computing 23 (2005) 1277–1287

[31]M. Sezgin, B. Sankur. "Survey over image thresholding techniques and quantitative performance evaluation", Journal of Electronic Imaging, Vol. 13, No. 1, pp. 146-165, 2004.3