

# A Novel Histogram-Based Multi-Threshold Searching Algorithm for Multilevel Colour Thresholding

Regular Paper

Chi-Yi Tsai<sup>1,\*</sup>, Tsung-Yen Liu<sup>1</sup> and Wei-Chieh Chen<sup>1</sup>

<sup>1</sup> Department of Electrical Engineering, Tamkang University, Tamsui Dist., New Taipei City, Taiwan ROC

\* Corresponding author E-mail: chiyi\_tsai@mail.tku.edu.tw

Received 20 Jun 2012; Accepted 29 Aug 2012

DOI: 10.5772/52851

© 2012 Tsai et al.; licensee InTech. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

**Abstract** Image segmentation is an important preliminary process required in object tracking applications. This paper addresses the issue of unsupervised multi-colour thresholding design for colour-based multiple objects segmentation. Most of the current unsupervised colour thresholding techniques require adopting a supervised training algorithm or a cluster-number decision algorithm to obtain optimal threshold values of each colour channel for a colour-of-interest. In this paper, a novel unsupervised multi-threshold searching algorithm is proposed to automatically search the optimal threshold values for segmenting multiple colour objects. To achieve this, a novel ratio-map image computation method is proposed to efficiently enhance the contrast between colour and non-colour pixels. The Otsu's method is then applied to the ratio-map image to extract all colour objects from the image. Finally, a new histogram-based multi-threshold searching algorithm is developed to search the optimal upper-bound and lower-bound threshold values of hue, saturation and brightness components for each colour object. Experimental results show that the proposed method not only succeeds in separating all colour objects-of-interest in colour images, but also provides satisfactory

colour thresholding results compared with an existing multilevel thresholding method.

**Keywords** Multi-object Segmentation, Multilevel Thresholding, Colour Thresholding, Multi-threshold Searching, Ratio-map Image

## 1. Introduction

Object tracking in video streams is essential to many intelligent vision systems. In such systems, image segmentation plays a necessary preliminary processing role in a variety of image processing applications, especially in object detection and recognition. Accurately segmenting foreground objects from a background image is a significant issue in many image analysis fields. Several sophisticated image segmentation techniques have been reported in the literature, however, nature image segmentation is still a challenging task due to an inherently ambiguous problem [1]. To handle such an inherent complexity, some researchers have tried to extend the traditional bilevel thresholding techniques to a multilevel process since histogram thresholding is one of

the most widely used techniques in image segmentation. This is commonly known as multilevel thresholding.

Various multilevel thresholding techniques have been proposed for grey images. They roughly can be divided into parametric and nonparametric approaches [2]. In parametric approaches [3, 4], the probability density function (pdf) of each object's grey-level distribution is required. But, to estimate the pdf of the grey-level distribution is typically a nonlinear optimization problem. This usually leads to an inefficient algorithm with high computational complexity and poor performance. By contrast, nonparametric approaches directly determine the threshold values by optimizing some certain cost functions [5-8]. For instance, Otsu proposed a thresholding method that determines the optimal thresholds by maximizing a between-class variance criterion using an exhaustive search [5]. To speed up the maximization process in the conventional Otsu's method, Liao et al. proposed a fast multilevel global thresholding algorithm that maximizes a modified between-class variance with a look-up table acceleration approach [6]. However, this method still takes too much time for multilevel threshold selection since it requires evaluating all possible solutions. The authors in [7] proposed a new criterion for automatic multilevel thresholding with low computational complexity. Recently, Gao et al. used a quantum-behaved particle swarm optimization technique to improve the convergence rate of the Otsu's method [8]. These nonparametric multilevel thresholding approaches produce satisfactory segmentation results for grey images, however, the papers [5-8] considers only grey scale images (i.e., single channel) and cannot be directly applicable to multi channel images.

As to colour image thresholding, some researchers have employed neural-network techniques to achieve unsupervised colour segmentation. For instance, Wu et al. proposed an automatic self-organizing map (SOM) based multilevel thresholding algorithm for colour segmentation and human hand localization [9]. They developed a SOM transduction algorithm to learn the non-stationary colour distribution in HSI colour space that overcomes the issue of dynamic lighting condition. Haghighatdoost et al. proposed an automatic multilevel thresholding algorithm for colour image quantization [10]. They first used a growing time adaptive SOM network to find the 3-dimensional RGB histogram. Then a peak finding algorithm was employed to find the thresholds automatically. Deshmukh et al. proposed an adaptive colour image segmentation approach based on a multilayer neural-network, which detects multiple objects in an input image by independently segmenting saturation and intensity planes of the image [11]. Although the neural-network-based approaches usually work well in colour image segmentation, they have to

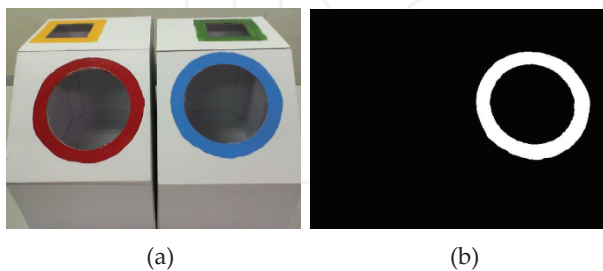
undergo a supervised training process to obtain the best segmentation performance. This requirement restricts the applicability of the neural-network-based segmentation approaches.

Some researchers have attempted to use fuzzy set theory to deal with colour image segmentation [12-15]. The authors in [12] proposed a coarse-to-fine segmentation strategy based on the thresholding and fuzzy c-means (FCM) techniques. This method first coarsely segments the image using a thresholding technique. The fine segmentation process then assigns the unclassified pixels to the closest class using the FCM technique. Chaabane et al. also adopted the concept of the coarse-to-fine strategy to segment colour images, but utilized an automatic histogram thresholding method in the coarse segmentation stage [13]. Kurugollu et al. developed a colour image segmentation technique using fuzzy thresholding and Dempster-Shafer's fusion rule [14]. This method first uses a fuzzy thresholding method to fuzzify each colour band into mass functions of three classes (object class, background class and uncertain class). These mass functions are then merged using the Dempster-Shafer's orthogonal sum rule. Finally, the optimal thresholds are taken by comparing the mid point of belief intervals of the object and background classes. Recently, Yu et al. introduced an adaptive unsupervised clustering algorithm based on the FCM technique and ant colony optimization [15]. This method is able to adaptively detect the cluster centroid distribution and centroid number, which are two critical initialization problems in FCM approach. The fuzzy-logic segmentation approaches usually require the knowledge of cluster number, otherwise they will become quite sensitive to the initialization condition. This requirement increases the complexity of image segmentation, leading to an inefficient solution to the unsupervised colour image segmentation problem.

From the literature survey, how to develop a new unsupervised colour image segmentation method efficiently extracting multiple colours-of-interest from foreground pixels in the image remains a challenge. This problem motivates us to develop a computationally efficient unsupervised multilevel thresholding algorithm for colour image segmentation. The proposed algorithm consists of a colour-pixel extraction process and a histogram-based multi-threshold searching process. In the colour-pixel extraction process, a novel ratio-map image computation method is proposed to enhance the contrast between colour and non-colour pixels. This helps leading to an efficient colour-object extraction algorithm using a simple bilevel thresholding method, such as the Otsu's method. On the other hand, the histogram-based multi-threshold searching process automatically searches the optimal upper-bound and lower-bound threshold values of hue, saturation and brightness components for

each colour object. Experimental results validate the performance of the proposed method by comparing with an existing multilevel thresholding method using nature colour images.

The rest of this paper is organized as follows. Section 2 defines the problem statement of this work. Section 3 introduces an existing method related to the development of the proposed algorithm. Section 4 presents the proposed multi-threshold searching algorithm in details. Experimental results are carried out in Section 5 to evaluate the performance of the proposed colour image segmentation approach. Section 6 concludes the contributions of this paper.



**Figure 1.** An example of image threshold segmentation: (a) the original colour image and (b) the blue-colour segmentation result using expression (1).

## 2. Problem statement

Image segmentation techniques have been widely used in various machine vision applications. To obtain the information of an object-of-interest from images, one of the most important tasks is to accurately extract all foreground objects from the background. Traditionally, a simple but efficient multilevel thresholding method that distinguishes pixels belonging to foreground and background based on colour distribution of a specific object-of-interest is used for this purpose. Let  $H_{in} \in [0, 360]$ ,  $S_{in} \in [0, 1]$  and  $V_{in} \in [0, 1]$  denote the input colour values of each pixel in HSV colour space. Suppose that the objects-of-interest are belonging to monotone colour objects, so that colour distribution in the HSV space is continuous with a bounded range and can be characterized by a rectangle region for each colour channel. Then, it is possible to find three threshold pairs  $(h_l, h_u)$ ,  $(s_l, s_u)$ , and  $(v_l, v_u)$  respectively thresholding each colour channel to create a binary image from an input HSV colour image such that

$$B(x, y) = \begin{cases} 1, & h_l \leq H_{in}(x, y) \leq h_u, \\ & s_l \leq S_{in}(x, y) \leq s_u, \\ & v_l \leq V_{in}(x, y) \leq v_u, \\ 0, & \text{Otherwise.} \end{cases} \quad (1)$$

where  $B(x, y)$  is the output binary image allowing to segment the object-of-interest from the background. Expression (1) means that if the hue, saturation and

brightness values of an input colour pixel are within the range of lower-bound to upper-bound threshold values, then the corresponding output pixel is assigned to the object class labelled 1, otherwise assigned to a null class. Figure 1 shows an example of image threshold segmentation using the image-thresholding method (1). In Fig. 1(a), the original colour image contains several colour objects. In this example, the threshold values are chosen as  $(h_l, h_u) = (190, 235)$ ,  $(s_l, s_u) = (0.21, 1)$ , and  $(v_l, v_u) = (0.1, 1)$  for extracting the blue-colour object (Fig. 1(b)).

Although the expression (1) provides a simple way to deal with colour image segmentation, the main challenge is how to determine the optimal threshold values of each colour channel for segmenting multiple colour objects. That is, for  $N$  object classes,  $N \geq 1$ , we try to find  $3N$  optimal threshold pairs  $(\hat{h}_l^{(n)}, \hat{h}_u^{(n)})$ ,  $(\hat{s}_l^{(n)}, \hat{s}_u^{(n)})$  and  $(\hat{v}_l^{(n)}, \hat{v}_u^{(n)})$ , for  $n=1 \sim N$ , to threshold an input HSV colour image into  $N$  labels such that

$$L_N(x, y) = \begin{cases} n, & \hat{h}_l^{(n)} \leq H_{in}(x, y) \leq \hat{h}_u^{(n)}, \\ & \hat{s}_l^{(n)} \leq S_{in}(x, y) \leq \hat{s}_u^{(n)}, \\ & \hat{v}_l^{(n)} \leq V_{in}(x, y) \leq \hat{v}_u^{(n)}, \\ 0, & \text{Otherwise.} \end{cases} \quad (2)$$

where  $L_N(x, y)$  is the output labelled image separating  $N$ -object and background pixels with minimum error. Expression (2) is the type of multilevel thresholding. According to [16], multilevel thresholding is generally less reliable since it is difficult to establish multiple thresholds effectively isolating each region-of-interest. Therefore, searching for optimal multiple thresholds typically is performed in a supervised manner (i.e., manual adjusting the thresholds to optimize thresholding) in order to improve the reliability of multilevel thresholding. This problem highlights the importance of developing a reliable multi-threshold searching algorithm to assist searching for optimal thresholds. This study thus proposes a novel histogram-based multi-threshold searching algorithm to achieve this purpose efficiently and effectively.

## 3. The existing Otsu's image thresholding method

This section briefly describes the Otsu's method [5] that is a well-known image thresholding technique used in grey image segmentation. Suppose that a grey image consists of one or more foreground objects, each having different grey-level values. The purpose of image thresholding is to separate these objects from the background using one or more threshold values obtained from the histogram of the image. More specifically, if the image histogram contains  $M \geq 2$  dominant modes (including one background mode), the image thresholding algorithm aims to find  $M \geq 2$  threshold values to extract all foreground objects in the image. Let  $N_p$  denote the

number of total pixels in the image,  $L$  the maximum grey-level value of the pixel and  $n_i$  the number of pixels with grey level  $i$ . Given the pixel-level probabilities  $p_i = n_i/N_p$  for  $i = 1 \sim L$ . In the case of  $M=2$  (bilevel thresholding), Otsu first defined a between-class variance of the thresholded image with respect to a threshold value  $t$  such that

$$\sigma_B^2(t) = w_1(t)[\mu_1(t) - \mu_T]^2 + w_2(t)[\mu_2(t) - \mu_T]^2, \quad (3)$$

where  $\mu_T = \sum_{i=1}^L ip_i$  is the mean intensity of the image;  $w_1(t) = \sum_{i=1}^t p_i$  and  $w_2(t) = \sum_{i=t+1}^L p_i$  are the probabilities of class occurrence;  $\mu_1(t) = w_1^{-1}(t) \sum_{i=1}^t ip_i$  and  $\mu_2(t) = w_2^{-1}(t) \sum_{i=t+1}^L ip_i$  are the class mean values. Then, the optimal threshold value  $\hat{t}$  can be determined by maximizing the between-class variance so that

$$\hat{t} = \arg \max_{1 \leq t \leq L} \sigma_B^2(t). \quad (4)$$

In the case of  $M > 2$  (multilevel thresholding), the expression (3) can be easily extended with respect to  $M-1$  threshold values  $\{t_1, t_2, \dots, t_{M-1}\}$  satisfying  $t_1 < t_2 < \dots < t_{M-1}$  such that

$$\sigma_B^2(t_1, t_2, \dots, t_{M-1}) = \sum_{i=1}^M w_i(t_i) [\mu_i(t_i) - \mu_T]^2, \quad (5)$$

where  $w_i(t_i) = \sum_{j=t_{i-1}+1}^{t_i} p_j$  for  $t_0 = 0$  and  $t_M = L$ , and  $\mu_i(t_i) = w_i^{-1}(t_i) \sum_{j=t_{i-1}+1}^{t_i} jp_j$ . Similarly, the optimal threshold values  $\{\hat{t}_1, \hat{t}_2, \dots, \hat{t}_{M-1}\}$  are determined by maximizing the criteria (5) so that

$$\bar{\sigma}_B^2(t_1, t_2, \dots, t_{M-1}) = \sum_{i=1}^M w_i(t_i) \mu_i^2(t_i). \quad (6)$$

The optimal threshold values are then obtained by maximizing the criteria (6) instead of (5) so that

$$\{\hat{t}_1, \hat{t}_2, \dots, \hat{t}_{M-1}\} = \arg \max_{1 \leq t_1 < t_2 < \dots < t_{M-1} \leq L} \bar{\sigma}_B^2(t_1, t_2, \dots, t_{M-1}). \quad (7)$$

The main advantage of using the modified criteria (6) is that the computation of maximization process (7) can be reduced by employing two look-up tables. Let  $M_0(u, v)$  and  $M_1(u, v)$  denote, respectively, the  $u$ - $v$  interval zero-order and first-order moments of a class with grey levels from  $u$  to  $v$  defined as

$$M_0(u, v) = \sum_{i=u}^v p_i, \text{ and } M_1(u, v) = \sum_{i=u}^v ip_i. \quad (8)$$

For index  $u = 1$ , the above expression can be rewritten in a recursive form:

$$M_0(1, v) = M_0(1, v-1) + p_v \text{ for } M_0(1, 0) = 0, \text{ and} \quad (9)$$

$$M_1(1, v) = M_1(1, v-1) + vp_v \text{ for } M_1(1, 0) = 0. \quad (10)$$

From the expressions (8)-(10), it follows that

$$M_0(u, v) = M_0(1, v) - M_0(1, u-1), \text{ and}$$

$$M_1(u, v) = M_1(1, v) - M_1(1, u-1). \quad (11)$$

According to (11),  $w_i(t_i)$  and  $\mu_i(t_i)$  can be rewritten as

$$w_i(t_i) = \sum_{j=t_{i-1}+1}^{t_i} p_j = M_0(1, t_i) - M_0(1, t_{i-1}) \\ = M_0(t_{i-1} + 1, t_i), \quad (12)$$

$$\mu_i(t_i) = \frac{1}{w_i(t_i)} \sum_{j=t_{i-1}+1}^{t_i} jp_j = \frac{M_1(1, t_i) - M_1(1, t_{i-1})}{w_i(t_i)} \\ = \frac{M_1(t_{i-1} + 1, t_i)}{M_0(t_{i-1} + 1, t_i)}. \quad (13)$$

Substituting (12) and (13) into (6) yields

$$\bar{\sigma}_B^2(t_1, t_2, \dots, t_{M-1}) = \sum_{i=1}^M \frac{M_1^2(t_{i-1} + 1, t_i)}{M_0(t_{i-1} + 1, t_i)} \\ = \frac{M_1^2(t_0 + 1, t_1)}{M_0(t_0 + 1, t_1)} + \frac{M_1^2(t_1 + 1, t_2)}{M_0(t_1 + 1, t_2)} + \dots + \frac{M_1^2(t_{M-1} + 1, t_M)}{M_0(t_{M-1} + 1, t_M)}, \quad (14)$$

where  $t_0$  and  $t_M$  are previously defined in (5). Expression (14) shows that the modified between-class variance  $\bar{\sigma}_B^2$  can be simplified as a function of  $M_0(u, v)$  and  $M_1(u, v)$ . To speed up the computation of (14), an efficient way is to pre-compute the two moments  $M_0(u, v)$  and  $M_1(u, v)$  for all possible intensities from  $u$  to  $v$  using the expressions (9)-(11). By doing so, the computations required for (14) are significantly reduced. Based on this observation, one can define the following function.

$$H(t_{i-1} + 1, t_i) = \frac{M_1^2(t_{i-1} + 1, t_i)}{M_0(t_{i-1} + 1, t_i)}, \quad (15)$$

and the expression (14) then becomes

$$\bar{\sigma}_B^2(t_1, t_2, \dots, t_{M-1}) = H(t_0 + 1, t_1) + H(t_1 + 1, t_2) + \dots \\ + H(t_{M-1} + 1, t_M). \quad (16)$$

Since the value of  $H(t_{i-1} + 1, t_i)$  for all threshold combinations satisfying  $t_{i-1} < t_i$  can be pre-computed in an  $L$ -by- $L$  look-up table (LUT), the computations of (16) are reduced to addition and LUT indexing operations that drastically speeds up the maximization process given in (7). In this paper, we term this LUT-accelerated multilevel thresholding method as fast Otsu's method. Although the fast Otsu's method is useful in achieving multilevel



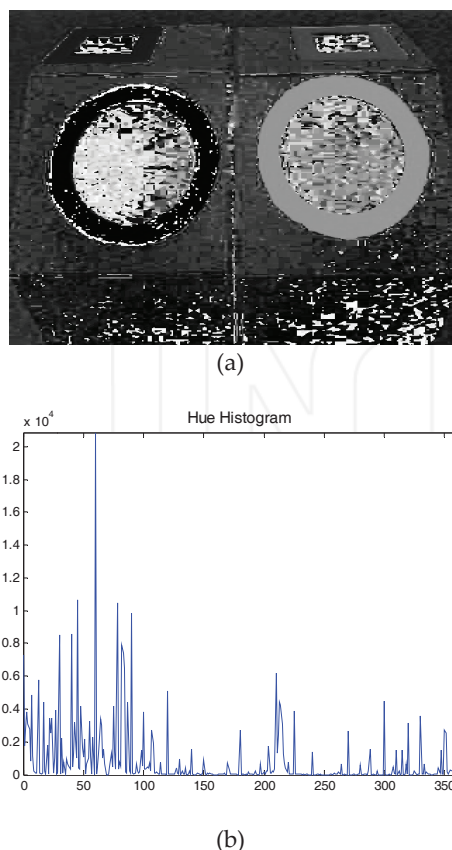
thresholding for grey images, it cannot directly be applied to the multilevel colour thresholding problem defined in the previous section. Therefore, the following section will present a new method to efficiently resolve the multilevel colour thresholding problem.

#### 4. The proposed algorithm

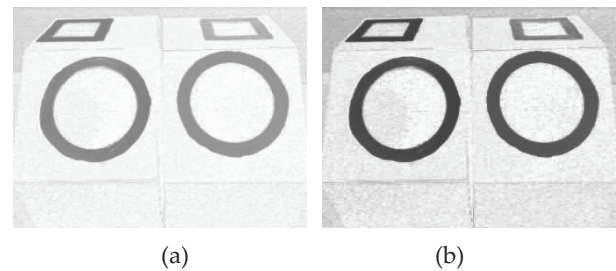
This section presents the proposed multi-threshold searching algorithm, which consists of a colour-pixel extraction process and a histogram-based multi-threshold searching process. The former aims to distinguish colour and non-colour (i.e., white and black colours) regions of an input colour image, and the latter searches the optimal multiple thresholds for thresholding the colour regions of the image with minimum error.

##### 4.1 Colour-pixel extraction process

Colour is an important cue for image segmentation. To extract the colour information from an image, the existing colour-based segmentation approaches usually require representing the image in the HSV colour space since the hue component of the HSV colour model measures the perceived colours of a visual sensing. Figure 2(a), for example, presents the hue component of the image shown in Fig. 1(a).



**Figure 2.** Colour information of the image shown in Fig. 1(a): (a) the hue component of the image and (b) the corresponding hue histogram. One can see that the hue histogram cannot efficiently distinguish the colour and non-colour object classes in the image.



**Figure 3.** Ratio-map images obtained by applying (3) and (4) on Fig. 1(a): the result with (a)  $\beta=2$  (producing less contrast) and (b)  $\beta=4$  (producing more contrast).

The colour and non-colour regions of the image can then possibly be distinguished by finding the optimal multiple thresholds from the corresponding hue histogram. However, this method may lead to unreliable results because the hue histogram generally cannot distinguish the colour and non-colour object classes in the image (Fig. 2(b)). This problem motivates us to develop a new method to efficiently distinguish the colour and non-colour regions of the image.

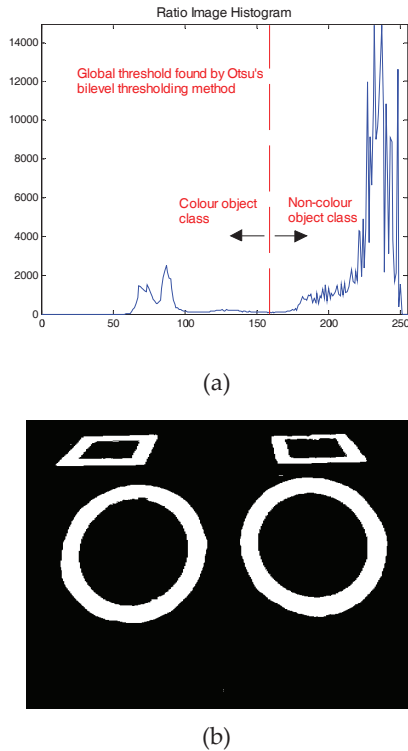
Suppose that the input colour image is represented in HSV colour space. As inspired by the existing shadow detection methods [17, 18], the proposed colour extraction algorithm first highlights all non-colour pixels on the basis of the saturation and brightness components of the image such that

$$\alpha(x, y) = \frac{V_{in}(x, y)}{S_{in}(x, y) + \varepsilon} + \frac{1}{S_{in}(x, y)V_{in}(x, y) + \varepsilon}, \quad (17)$$

where  $\varepsilon$  is a small non-zero positive value to avoid dividing by zero. In expression (17), the first term aims to highlight white pixels and the second term to highlight black pixels of the image. Expression (17) results a high-dynamic range image, therefore, a dynamic range compression process is employed to produce a ratio-map image with respect to  $\alpha(x, y)$  such that

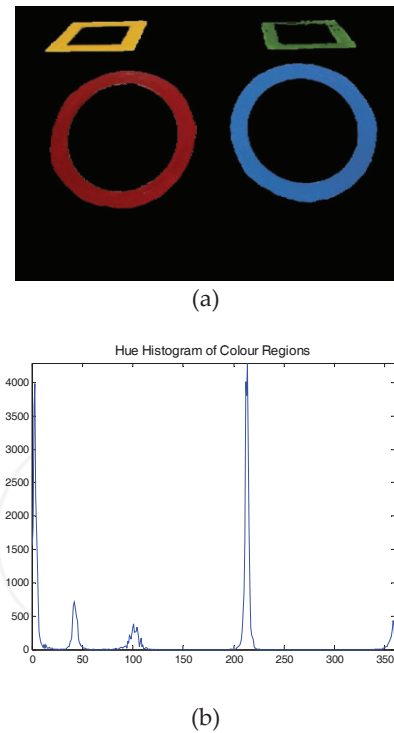
$$R(x, y) = 255 \left( \frac{\alpha(x, y)}{1 + \alpha(x, y)} \right)^\beta, \quad (18)$$

where  $R(x, y) \in [0, 255]$  is a greyscale image highlighting all non-colour pixels in the original image,  $\beta$  is a non-zero positive parameter satisfying  $\beta \geq 1$  to control the contrast of the ratio-map image  $R(x, y)$ . Figure 3(a) and 3(b) present two ratio-map images obtained by applying (17) and (18) on Fig. 1(a) with  $\beta=2$  and  $\beta=4$ , respectively. From Fig. 3, it is clear that the parameter  $\beta$  is able to control the contrast of the ratio-map image. This property helps to distinguish colour and non-colour pixels in the original image by thresholding the ratio-map image via a conventional single-level threshold approach, i.e., the Otsu's method presented in the previous section. Figure 4 is an example to explain this observation. Fig. 4(a) shows the grey histogram of the ratio-map image Fig. 3(b).



**Figure 4.** Bilevel thresholding result of the ratio-map image Fig. 3(b) using the conventional Otsu's method [5]: (a) the grey histogram of the ratio-map image Fig. 3(b) and (b) colour regions of the image Fig. 1(a) distinguished by thresholding the ratio-map image Fig. 3(b) using the Otsu's method.

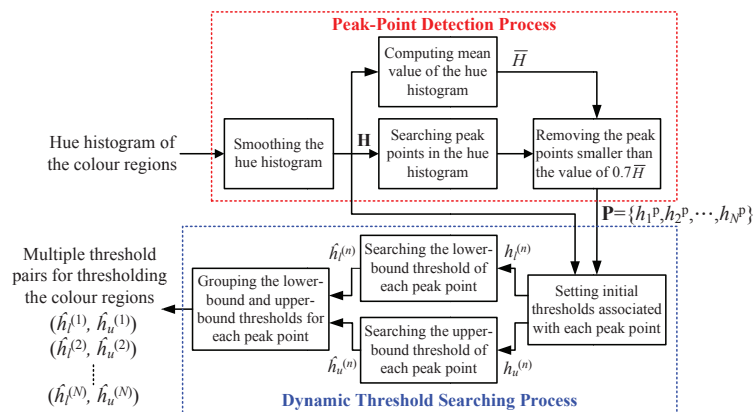
It is clear from Fig. 4(a) that the colour and non-colour object classes can be simply distinguished by a global threshold. Hence, the colour regions of the original image can be efficiently extracted by using the conventional Otsu's bilevel thresholding method to threshold the ratio-map image obtained from (17) and (18) (Fig. 4(b)). After extracting the colour regions of the image, one can obtain all colour pixels in the original image (Fig. 5(a)) and compute the corresponding hue histogram of the colour information (Fig. 5(b)). Visually comparing Fig. 2(b) with Fig. 5(b) finds that Fig. 5(b) contains only hue information of the colour pixels, which is benefit for the following multi-threshold searching process to determine the optimal colour thresholds effectively and efficiently.



**Figure 5.** Hue histogram of the colour regions of the image: (a) colour pixels in the extracted colour regions and (b) the corresponding hue histogram that will be used in the following multi-threshold searching process.

#### 4.2 Histogram-based multi-threshold searching process

Figure 6 shows the flowchart of the proposed histogram-based multi-threshold searching algorithm, which consists of a peak-point detection process and a multi-threshold searching process. Suppose that the hue histogram of the colour regions of the image is obtained from the colour-pixel extraction process described in the previous section. The peak-point detection process aims to detect peak points in the input hue histogram depending on the mean value of the histogram. Since searching the peak points directly on the original hue histogram is very sensitive to image noise, the process first performs a smoothing operation on the input hue histogram to reduce the noise effect during peak-point detection (Fig. 7(a)).



**Figure 6.** Flowchart of the proposed histogram-based multi-threshold searching algorithm.

Let  $\mathbf{H} \in \mathbb{R}^{361}$  denote the smoothed hue histogram and  $\bar{H}$  the mean value of  $\mathbf{H}$ . Then, the peak points in the smoothed hue histogram can be detected such that

$$\mathbf{P} = \{h \in \mathbb{N} \mid \mathbf{H}(h) - \mathbf{H}(h-2) > 0, \mathbf{H}(h) - \mathbf{H}(h-1) > 0, \\ \mathbf{H}(h) - \mathbf{H}(h+1) > 0, \mathbf{H}(h) - \mathbf{H}(h+2) > 0, \\ \mathbf{H}(h) > 0.7\bar{H}, 0 \leq h \leq 360\}, \quad (19)$$

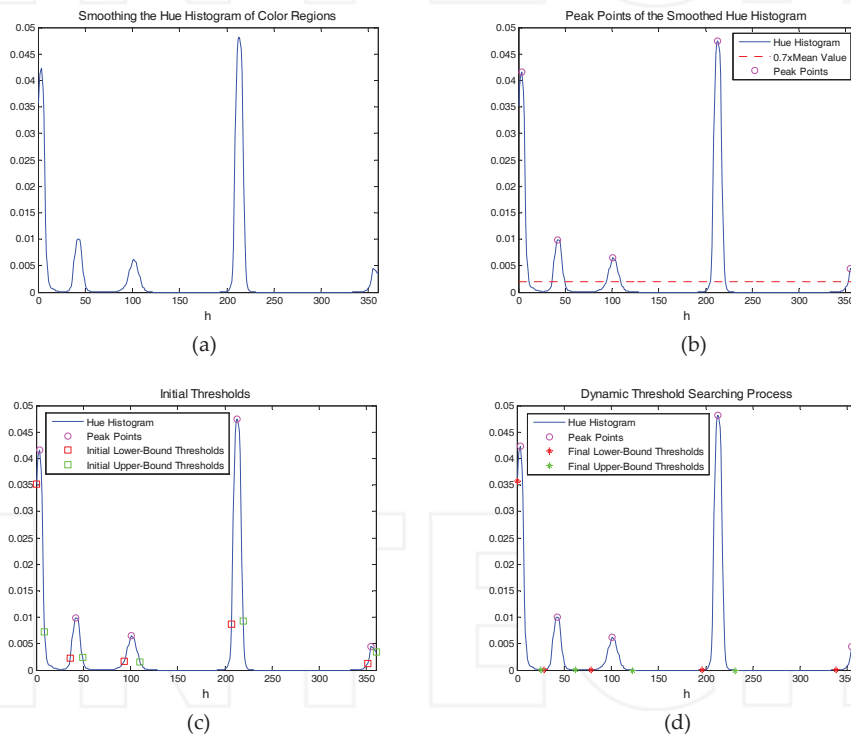
where  $h$  is an integer index ranged from 0 to 360 and  $\mathbf{P}$  is the peak-point set recording the position of each peak-point in the histogram, as shown in Fig. 7(b). Note that only the peak value larger than the value of  $0.7\bar{H}$  will be considered as a peak point empirically. Since a peak point is normally corresponding to one object class, there would be  $N$  peak points within the smoothed hue histogram in the case of  $N$  object classes. Consequently, the detected  $N$  peak-point positions, denoted by  $\mathbf{P} = \{h_1^p, h_2^p, \dots, h_N^p\}$  with  $h_1^p < h_2^p < \dots < h_N^p$ , can be used as the start positions in the next dynamic threshold

searching process to assist finding the optimal lower-bound and upper-bound thresholds for each object class (Fig. 6).

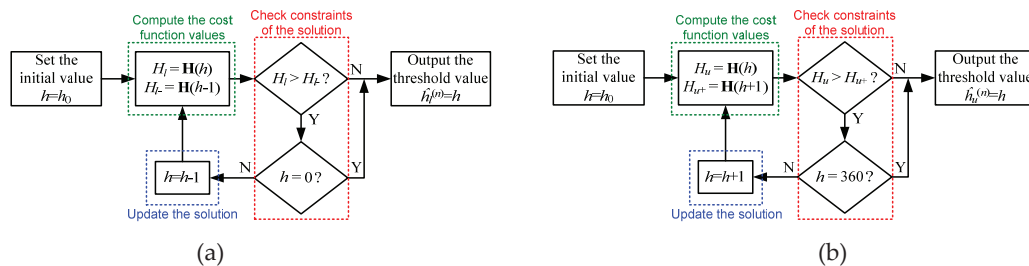
To search the optimal thresholds, a valid assumption is that the peak-point position is located within the range of lower-bound to upper-bound thresholds. Based on this assumption, the multi-threshold searching process can be divided into two unidirectional searching processes, one for searching lower-bound threshold and the other one for upper-bound threshold. Therefore, the first step in the proposed dynamic multi-threshold searching process is setting two initial thresholds around the peak point such that

$$h_l^{(n)} = \max\{h : \mathbf{H}(h) < \lambda \mathbf{H}(h_n^p), h < h_n^p\} \quad \text{and}$$

$$h_u^{(n)} = \min\{h : \mathbf{H}(h) < \lambda \mathbf{H}(h_n^p), h > h_n^p\}, \quad \text{for } n = 1 \sim N, \quad (20)$$



**Figure 7.** Processing steps of the proposed multi-threshold searching algorithm: (a) smoothing the hue histogram, (b) detecting peak points of the hue histogram, (c) setting initial thresholds related to the detected peak points and (d) searching final thresholds via the proposed dynamic threshold searching process.



**Figure 8.** Dynamic threshold searching process: flowchart of dynamic searching the (a) lower-bound threshold and (b) upper-bound threshold associated with the peak point  $h_n^p$ ,  $n=1 \sim N$ .

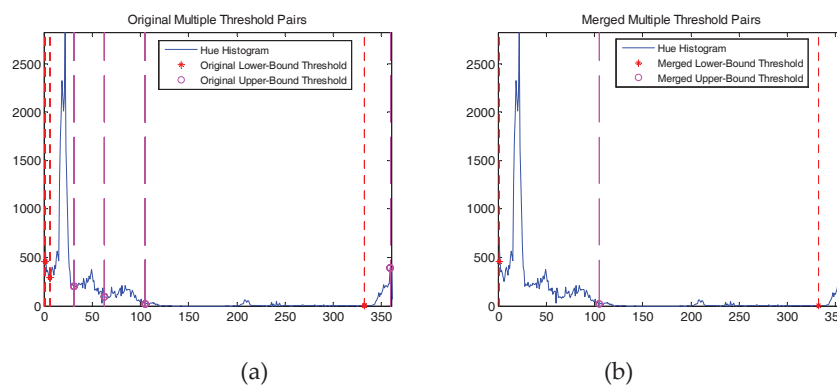
where the operators  $\max\{\Omega\}$  and  $\min\{\Omega\}$  denote respectively the maximum and minimum element of the set  $\Omega$ . The parameter  $\lambda$  in (20) is a non-zero scale factor satisfying  $0 < \lambda < 1$ . Here, the default value of  $\lambda$  is set as 0.3 empirically. Figure 7(a) presents an example to explain the initial-threshold detection via expression (20). Note that if the peak-point position  $h_n^p$  is located on the boundary of hue histogram, i.e.  $h_n^p = 0$  or  $h_n^p = 360$ , the out-of-range initial threshold will be set as the same value as the peak-point position. After setting the initial thresholds, the optimal-threshold searching problem can be modelled as the following two constrained minimization problems such that

$$\begin{aligned} \hat{h}_l^{(n)} &= \arg \min_h \mathbf{H}(h) \Big|_{h_l = h_l^{(n)}} & \hat{h}_u^{(n)} &= \arg \min_h \mathbf{H}(h) \Big|_{h_u = h_u^{(n)}} \\ \text{subject to } \mathbf{H}(h) - \mathbf{H}(h-1) &> 0 & \text{subject to } \mathbf{H}(h) - \mathbf{H}(h+1) &> 0 \\ h &\geq 0 & h &\leq 360 \end{aligned} \quad (21)$$

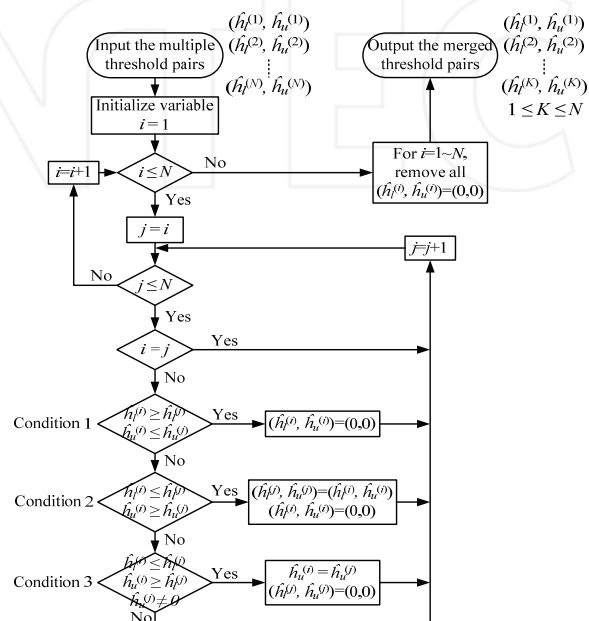
where  $h_0$  denotes the initial value of  $h$  at the begin of minimization process. The constrained minimization

problems in (21) can be solved by employing a simple direct-search method as shown in Fig. 8. Figure 7(d) illustrates the optimal-threshold searching results obtained from resolving the above two constrained minimization problems. It is clear that the final thresholds successfully separate each colour region in the smoothed hue histogram. Note that, as mentioned before, if the threshold value is out-of-boundary, then it will be set as the boundary value.

After the dynamic threshold searching process, a potential problem is that the range bounded by the obtained lower-bound and upper-bound threshold values may overlap or repeat each other. Figure 9(a) shows an example to explain this problem. In Fig. 9(a), there are four threshold pairs obtained from the dynamic threshold searching process:

$$(\hat{h}_l^{(1)}, \hat{h}_u^{(1)}) = (2, 64), \quad (\hat{h}_l^{(2)}, \hat{h}_u^{(2)}) = (7, 32), \quad (\hat{h}_l^{(3)}, \hat{h}_u^{(3)}) = (2, 106), \quad \text{and} \quad (\hat{h}_l^{(4)}, \hat{h}_u^{(4)}) = (333, 360).$$


**Figure 9.** An example of merging the redundant threshold pairs: (a) the threshold pairs obtained from the dynamic threshold searching process and (b) the merged threshold pairs via the proposed threshold pair merging process.



**Figure 10.** Flowchart of the proposed threshold pair merging process.



It is clear from this case that the first and second threshold pairs are redundant since their thresholding ranges are within the range of the third threshold pair. This problem deteriorates the computational performance of the final colour thresholding process due to the redundant thresholding operations. To avoid generating the redundant threshold pairs, a threshold pair merging process is proposed to minimize the number of threshold pairs with the same thresholding ranges (Fig. 9(b)). Figure 10 shows the flowchart of the proposed threshold pair merging process, which aims to detect and reduce all redundancy in the input multiple threshold pairs according to three conditions. The first and second conditions detect the redundant threshold pairs that will be removed from the input threshold pairs. By contrast, the third condition detects the overlapped threshold pairs that will be merged together. Taking Fig. 9(a) as an example, the first two threshold pairs,  $(\hat{h}_l^{(1)}, \hat{h}_u^{(1)}) = (2, 64)$  and  $(\hat{h}_l^{(2)}, \hat{h}_u^{(2)}) = (7, 32)$ , satisfy the second condition of the procedure, leading to remove one redundant threshold pair so that  $(\hat{h}_l^{(2)}, \hat{h}_u^{(2)}) = (2, 64)$  and  $(\hat{h}_l^{(1)}, \hat{h}_u^{(1)}) = (0, 0)$ . Next, one can observe that the threshold pair  $(\hat{h}_l^{(2)}, \hat{h}_u^{(2)}) = (2, 64)$  will be set as  $(\hat{h}_l^{(2)}, \hat{h}_u^{(2)}) = (0, 0)$  due to satisfying the first condition of the procedure. After removing all zero threshold values, the final threshold pairs are simplified as  $(\hat{h}_l^{(1)}, \hat{h}_u^{(1)}) = (2, 106)$  and  $(\hat{h}_l^{(2)}, \hat{h}_u^{(2)}) = (333, 360)$ . Observing Figs. 9(a) and 9(b) we find that the merged result produces the same thresholding range to the original one, but using a smaller number of threshold pairs. Therefore, the computational complexity of the following colour segmentation process can be reduced efficiently.

In the process of searching saturation and brightness thresholds, we only find the lower-bound thresholds  $\hat{s}_l^{(n)}$  and  $\hat{b}_l^{(n)}$  as the first non-zero position with a non-zero value in the saturation and brightness histograms, respectively. The upper-bound thresholds  $\hat{s}_u^{(n)}$  and  $\hat{b}_u^{(n)}$  are fixed as 1. The performance of the proposed algorithm will be evaluated in the experiment section.

## 5. Experimental results

This section contains two parts. The first part presents the experimental results of the proposed method compared with an existing method and the second part presents the extension of the proposed method to skin colour segmentation.

### 5.1 Performance evaluation of the proposed method

In the first experiments, two test images, as shown in Figs. 11(a) and 12(a), were employed to evaluate the performance of the proposed method.

The fast Otsu's multilevel thresholding method described in Section 3 was used as a competing method to evaluate the performance of the proposed method. Table 1

tabulates the parameter settings of the competing and proposed methods used in the experiments. In order to quantify the performance of the proposed method, the PSNR metric (in dB) was adopted in the experiments. The PSNR metric in this paper is defined as

$$PSNR(\mathbf{T}_s, \mathbf{T}_p) = 10 \log_{10} \left[ 255^2 \left( \frac{1}{UV} \sum_{1 \leq y \leq U} \sum_{1 \leq x \leq V} \|\mathbf{T}_s(x, y) - \mathbf{T}_p(x, y)\|^2 \right)^{-1} \right],$$

where  $U$  and  $V$  are respectively the total row and column number of the image,  $\mathbf{T}_s$  is the multilevel thresholding result obtained by a manually supervised thresholding, and  $\mathbf{T}_p$  is the corresponding result by using an unsupervised thresholding method. Figure 11(b) presents the multilevel thresholding result obtained by the supervised method that is manually adjusting each threshold to get the best segmentation result by visual observation.

Figures 11(c) and 11(d) present respectively the multilevel thresholding result and the multiple thresholds of hue channel obtained from the fast Otsu's method. It is clear from these two figures that the fast Otsu's method cannot provide satisfactory segmentation results since the multiple threshold values are not able to threshold the two colour classes in the hue histogram properly. This observation also can be verified in Table 2, which records the multiple threshold values obtained by each compared method and the corresponding PSNR measures. From Table 2, one can see that the PSNR metric between Figs. 11(b) and 11(c) is 17.1552 dB, which means a low-similarity between the results produced by the fast Otsu's and manually supervised methods.

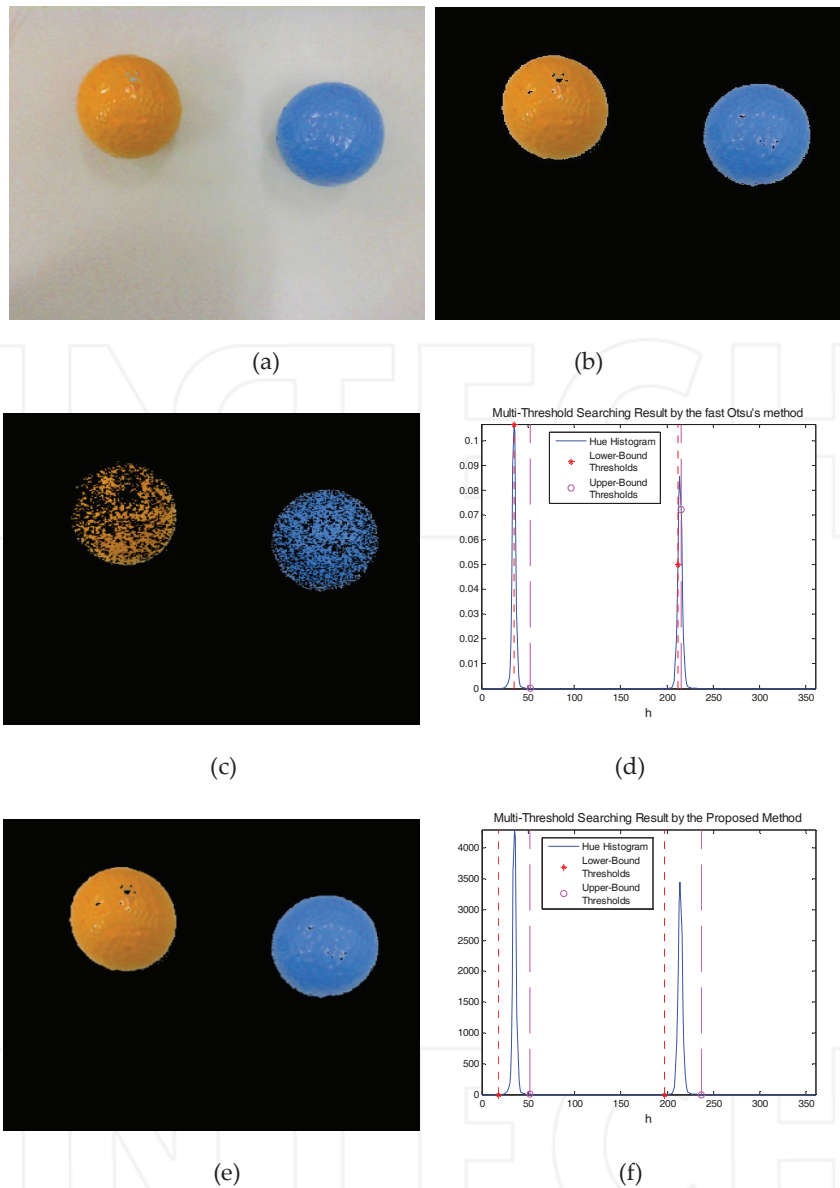
Method	Symbol	Values	Description
Fast Otsu's method	$M$	5	Class number in (7)
Proposed method	$\beta$	1	Contrast control parameter in (18)
	$\lambda$	0.3	Scale factor in (20)

**Table 1.** Parameter settings of the competing and proposed methods.

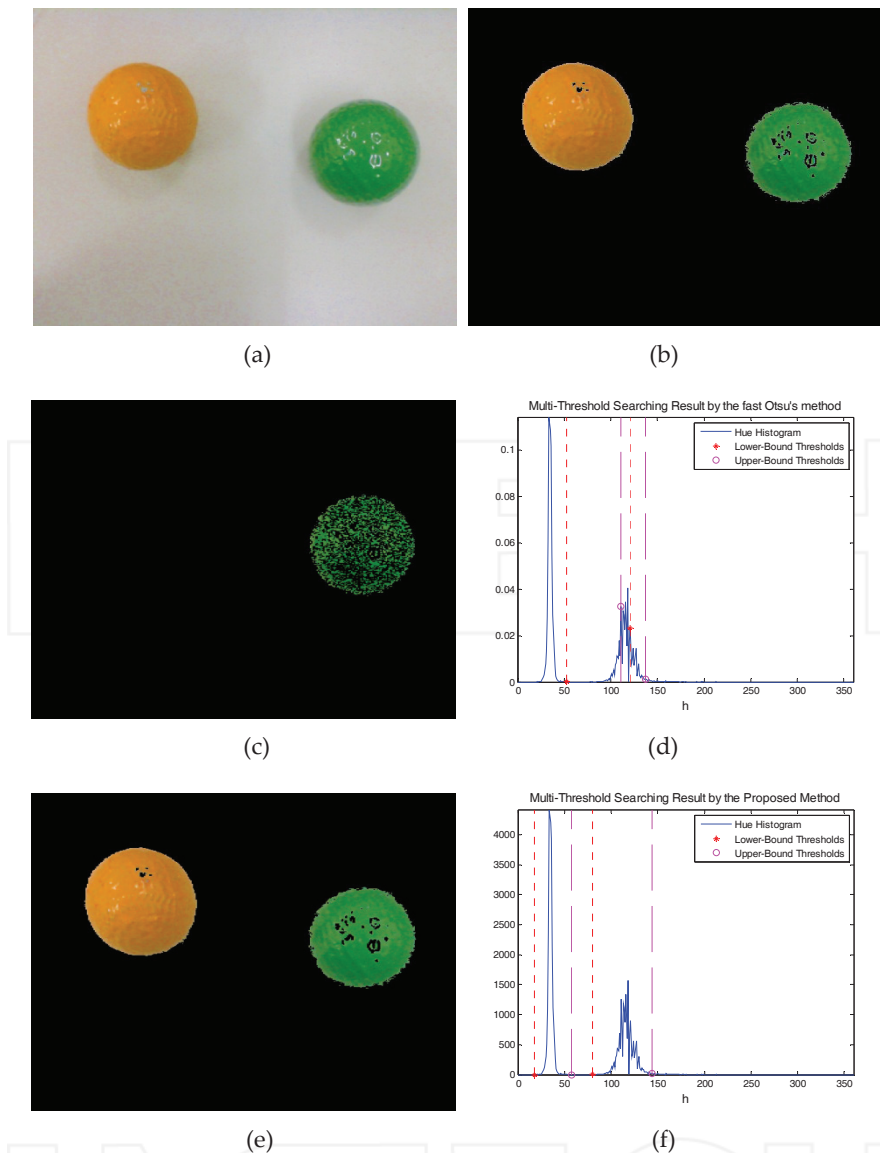
On the other hand, Fig. 11(e) shows the multilevel thresholding result obtained from the proposed method. Visually comparing Figs. 11(b) and 11(e) finds that the proposed method produces similar segmentation results to the manually supervised method. Figure 11(f) illustrates the multiple thresholds of hue channel obtained from the proposed method. It is clear from Fig. 11(f) that the multiple threshold values succeed to threshold the two colour classes in the hue histogram, producing a high PSNR of 38.0751 dB as shown in Table 2. These results explain why the proposed method produces a similar result to the manually supervised

method. Similar observations can also be found in the case shown in Figs. 12(a)-12(f). Therefore, the above

experimental studies validate the performance of the proposed method.



**Figure 11.** Experimental results of multilevel thresholding process: (a) the original image, (b) thresholding by a manually supervised method, (c) multilevel thresholding by the fast Otsu's method, (d) the multiple thresholds of hue channel obtained from the fast Otsu's method, (e) multilevel thresholding by the propose method and (f) the multiple thresholds of hue channel obtained from the proposed method.



**Figure 12.** Experimental results of multilevel thresholding process: (a) the original image, (b) thresholding by a manually supervised method, (c) multilevel thresholding by the fast Otsu's method, (d) the multiple thresholds of hue channel obtained from the fast Otsu's method, (e) multilevel thresholding by the propose method and (f) the multiple thresholds of hue channel obtained from the proposed method.

Method	Figure	$n$	$\hat{h}_l^{(n)}$	$\hat{h}_u^{(n)}$	$\hat{s}_l^{(n)}$	$\hat{s}_u^{(n)}$	$\hat{v}_l^{(n)}$	$\hat{v}_u^{(n)}$	PSNR(dB)
Fast Otsu's method	Fig. 11(c)	1	36	54	0.3451	1.0	0.5765	1.0	17.1552
		2	213	216					
	Fig. 12(c)	1	54	112	0.3412	1.0	0.4431	1.0	16.1645
		2	122	138					
Proposed method	Fig. 11(e)	1	19	53	0.3451	1.0	0.5765	1.0	38.0751
		2	198	238					
	Fig. 12(e)	1	19	59	0.3412	1.0	0.4431	1.0	34.4459
		2	81	145					

**Table 2.** Multiple threshold values obtained by each compared method and the corresponding PSNR measures.



(a)



(b)



(c)

**Figure 13.** Experimental results of skin colour segmentation: (a) the original image, skin colour thresholding by (b) the propose method and (c) the previous work [19].

Method	Figure	$n$	$\hat{h}_l^{(n)}$	$\hat{h}_u^{(n)}$	$\hat{s}_l^{(n)}$	$\hat{s}_u^{(n)}$	$\hat{v}_l^{(n)}$	$\hat{v}_u^{(n)}$
Proposed method	Fig. 13(b)	1	1	41	0.1843	1.0	0.0941	1.0
Previous work [19]	Fig. 13(c)	1	0	50	0.23	0.68	0	1.0

**Table 3.** Multiple threshold values obtained by the proposed method and the previous work [19] for skin colour thresholding.

### 5.2 Extension to skin colour segmentation

The proposed method can be extended to other colour image segmentation applications, such as skin colour segmentation. Figure 13(a) illustrates a skin colour image, which challenges many existing skin colour segmentation methods since it contains a variety of ethnic skin tones. Figure 13(b) shows the colour thresholding result obtained from the proposed method. It can be observed from Fig. 13(b) that the proposed method efficiently extracts most skin pixels in the original image, except some white skin pixels. Table 3 tabulates the multiple threshold values found by the proposed method in this experiment. From Table 3, one can see that the skin colour distribution ranges from 1 to 41 in hue channel. This observation is consistent with the result of a previous work [19] which shows that the skin colour distribution in the Hue-Saturation plane typically ranges from 0 to 50

in hue channel for Asian and Caucasian ethnicities. On the other hand, the reported result of skin colour in saturation channel is characterized by values between 0.23 and 0.68, but our finding is between 0.1843 and 1.0. This is caused by that the test image used here additionally containing the Ethiopian ethnicity, leading to a smaller lower-bound threshold value on saturation. Furthermore, as mentioned in the end of Section 4.2, the upper-bound threshold of saturation is fixed as 1.0 in our design. Figure 13(c) shows the colour thresholding result obtained by the previous work [19]. Visually comparing Figs. 13(b) and 13(c) we can observe that the skin colour thresholding results obtained by the proposed method and in the previous work are very similar to each other. This result validates the performance of the proposed method extended to skin colour segmentation as we expected.



## 6. Conclusions and future work

In this study, a novel histogram-based multi-threshold searching algorithm has been developed to achieve automatic multilevel colour image thresholding for segmenting multiple objects in colour images. The proposed method comprises a new colour-pixel extraction algorithm, which computes a ratio-map image efficiently enhancing the contrast between colour and non-colour pixels. This property helps to extract all foreground colour objects from background image via a conventional single-level image thresholding method. Moreover, a new histogram-based multi-threshold searching algorithm is also proposed to search the optimal upper-bound and lower-bound thresholds of hue, saturation and brightness channels for each colour object. Experimental results validate the performance of the proposed method by comparing with the fast Otsu's multilevel thresholding method, both visually and quantitatively. It is worth noting that the proposed algorithm can be extended to other colour image segmentation applications, such as nature colour image segmentation or skin colour segmentation. This advantage greatly increases the applicability of the proposed method. The extension to multilevel video thresholding is left to our future study.

## 7. Acknowledgments

This work was supported by the National Science Council of Taiwan, ROC under grant NSC 101-2221-E-032-022.

## 8. References

- [1] A. Y. Yang, J. Wright, Y. Ma, and S. S. Sastry (2008), Unsupervised Segmentation of Natural Images via Lossy Data Compression, *Computer Vision and Image Understanding*, Vol. 110, No. 2, pp. 212-225.
- [2] A. S. Abutaleb (1989), Automatic Thresholding of Gray-Level Pictures Using Two-Dimensional Entropy, *Computer Vision, Graphics, and Image Processing*, Vol. 47, No. 1, pp. 22-32.
- [3] A. Rosenfeld and A. C. Kak (1982), *Digital Picture Processing*, 2<sup>nd</sup> edition, New York: Academic.
- [4] W. Snyder (1990), Optimal Thresholding - a New Approach, *Pattern Recognition Letters*, Vol. 11, pp. 803-810.
- [5] N. Otsu (1979), A Threshold Selection Method from Gray-Level Histograms, *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 9, No. 1, pp. 62-66.
- [6] P.-S. Liao, T.-S. Chen, and P.-C. Chung (2001), A Fast Algorithm for Multilevel Thresholding, *Journal of Information Science and Engineering*, Vol. 17, No. 5, pp. 713-727.
- [7] J.-C. Yen, F.-J. Chang, and S. Chang (1995), A New Criterion for Automatic Multilevel Thresholding, *IEEE Transactions on Image Processing*, Vol. 4, No. 3, pp. 370-378.
- [8] H. Gao, W. Xu, J. Sun, and Y. Tang (2010), Multilevel Thresholding for Image Segmentation Through an Improved Quantum-Behaved Particle Swarm Algorithm, *IEEE Transactions on Instrumentation and Measurement*, Vol. 59, No. 4, pp. 934-945.
- [9] Y. Wu, Q. Liu, and T. S. Huang (2000), An Adaptive Self-Organizing Color Segmentation Algorithm with Application to Robust Real-time Human Hand Localization. In *Proc. Asian Conf. on Computer Vision*, Taiwan, pp. 1106-1111.
- [10] V. Haghighatdoost and R. Safabakhsh (2006), Automatic Multilevel Color Image Thresholding by the Growing Time Adaptive Self Organizing Map, 2<sup>nd</sup> *IEEE International Conference on Information and Communication Technologies*, Damascus, Syria, pp. 1768-1772.
- [11] K. S. Deshmukh and G. N. Shinde (2005), An Adaptive Color Image Segmentation, *Electronic Letters on Computer Vision and Image Analysis*, Vol. 5, No. 4, pp. 12-23.
- [12] Y.-W. Lim and S.-U. Lee (1990), On the Color Image Segmentation Algorithm Based on the Thresholding and the Fuzzy C-Means Techniques, *Pattern Recognition*, Vol. 23, No. 9, pp. 935-952.
- [13] S. B. Chaabane, M. Sayadi, F. Fnaiech, and E. Brassart, (2008), Color Image Segmentation using Automatic Thresholding and the Fuzzy C-Means Techniques. 14<sup>th</sup> *IEEE Mediterranean Electrotechnical Conference*, Ajaccio, France, pp. 857-861.
- [14] F. Kurugollu, A. Bouridane, and M. A. Roulia (2007), Fuzzy Thresholding of Color Images Using Dempster-Shafer Theory, 9<sup>th</sup> *International Symposium on Signal Processing and Its Applications*, Sharjah, United Arab Emirates, pp. 1-4.
- [15] Z. Yu, O. C. Au, R. Zou, W. Yu, and J. Tian (2010), An Adaptive Unsupervised Approach toward Pixel Clustering and Color Image Segmentation, *Pattern Recognition*, Vol. 43, No. 5, pp. 1889-1906.
- [16] R. C. Gonzalez and R. E. Woods (2002), *Digital Image Processing*, 2<sup>nd</sup> edition, NJ: Prentice-Hall.
- [17] V. J. D. Tsai (2006), A Comparative Study on Shadow Compensation of Color Aerial Images in Invariant Color Models, *IEEE Transactions on Geoscience and Remote sensing*, Vol. 44, No. 6, pp. 1661-1671.
- [18] K.-L. Chung, Y.-R. Lin, and Y.-H. Huang (2009), Efficient Shadow Detection of Color Aerial Images Based on Successive Thresholding Scheme, *IEEE Transactions on Geoscience and Remote sensing*, Vol. 47, No. 2, pp. 671-682.
- [19] K. Sobottka and I. Pitas (1998), A Novel Method for Automatic Face Segmentation, Facial Feature Extraction and Tracking, *Signal Processing: Image Communication*, Vol. 12, No. 3, pp. 263-281.