Pattern Recognition 43 (2010) 3219–3232

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Adaptive image segmentation for region-based object retrieval using generalized Hough transform

Chi-Han Chung, Shyi-Chyi Cheng *, Chin-Chun Chang

Department of Computer Science and Engineering, National Taiwan Ocean University, 2, Peining Rd., Keelung 20224, Taiwan

ARTICLE INFO

Article history:
Received 23 March 2009
Received in revised form
13 February 2010
Accepted 29 April 2010

Keywords:
Object recognition
Hough transform
Image segmentation
Information retrieval

ABSTRACT

Finding an object inside a target image by querying multimedia data is desirable, but remains a challenge. The effectiveness of region-based representation for content-based image retrieval is extensively studied in the literature. One common weakness of region-based approaches is that perform detection using low level visual features within the region and the homogeneous image regions have little correspondence to the semantic objects. Thus, the retrieval results are often far from satisfactory. In addition, the performance is significantly affected by consistency in the segmented regions of the target object from the query and database images. Instead of solving these problems independently, this paper proposes region-based object retrieval using the generalized Hough transform (GHT) and adaptive image segmentation. The proposed approach has two phases. First, a learning phase identifies and stores stable parameters for segmenting each database image. In the retrieval phase, the adaptive image segmentation process is also performed to segment a query image into regions for retrieving visual objects inside database images through the GHT with a modified voting scheme to locate the target visual object under a certain affine transformation. The learned parameters make the segmentation results of query and database images more stable and consistent. Computer simulation results show that the proposed method gives good performance in terms of retrieval accuracy, robustness, and execution speed.

1. Introduction

Humans use high level concepts in everyday life. However, existing computer vision techniques automatically extract only low level features from images. Object segmentation and recognition is the primary step in applying computer vision to image retrieval with higher level image analysis [1,2]. In constrained applications, such as the human face and fingerprint, high level concepts (faces or fingerprints) can be represented using low level features [3]. In a general setting, however, there is a semantic gap between real world objects and their low level features. Automatic segmentation and object recognition via object models is difficult without prior knowledge of the object shapes.

The major bottleneck for a state-of-the-art approach to content-based image retrieval (CBIR) is this gap between low level features and high level semantic concepts. Therefore, an obvious approach to improving a CBIR system is to reduce or, in the best case, a bridge this gap. This paper presents an approach to retrieving visual objects from a target image through region-based image retrieval. Extensive research has been conducted in region-based image retrieval [4–8]. Most existing region-based techniques retrieve images according to the following procedures: (1) segment images into multiple disjointed regions, (2) extract features from image regions, and (3) perform region matching to obtain the similarity between two images. Some approaches focus on segmenting meaningful regions [9,10], while others focus on the design of a configuration-based technique for image matching by exploring the spatial relationships and arrangements of various regions in an image [7,8]. However, little existing work has emphasized enhancing the quality of features in regions.

It is difficult to perform object detection, recognition, or object-based feature extraction without a perceptually coherent grouping of the "raw" regions produced by image segmentation. Automatic segmentation is far from perfect. Perceptual grouping of segmented regions is expected to bridge the semantic gap between image segmentation and high level image understanding. For this purpose, Luo and Guo proposed a non-purposive grouping scheme that merges small regions into larger meaningful regions according to their features and geometric coherency measures, including convexity, completion, symmetry, and occlusion [10]. In [9], Fan et al. proposed a statistical model for conceptualizing natural images based on concept sensitive salient objects, which are defined as the dominant image components

* Corresponding author. Tel.: +886 2 24622192x6653; fax: +886 2 24623249.
E-mail address: csc@mail.ntou.edu.tw (S.-C. Cheng).

0031-3203/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2010.04.022

that are semantic to a human being and are also visually distinguishable. In [11], Zhu provided a good review of statistical modeling and the conceptualization of visual patterns.

Carson et al. [7] proposed the Blobworld system, in which a user is required to select important regions and features. Wang et al. [5] proposed an integrated matching algorithm to retrieve images from picture libraries based on region similarity in terms of a combination of color, shape, and texture information. However, this approach does not provide a general way to measure image similarity using spatial relationships in the region, which are an important cue for middle level image understanding. Hsieh and Grimson [8] proposed an image retrieval framework using spatial templates for region matching. They support matching one-to-many regions in two stages—a similarity comparison followed by a region voting.

Pratikakis et al. [6] used the idea of measuring region weighting, based on a hierarchical watershed driven algorithm that automatically extracts meaningful regions. In this framework, many-to-many region matching, along with region weighting, is used to enhance feature discrimination. In the same spirit, region weighing based on user relevance feedback is proposed in several different region-based image retrieval systems [12,13].

A retrieval system that uses structural information extracted by perceptual grouping has an edge over content-based image retrieval systems that retrieve images containing structural objects based purely on low level features. Various approaches to grouping visual patterns extracted from image blocks, regions, or objects have been proposed to offer a semantic-based representation for image understanding and analysis applications [4]. Grouping visual patterns into image models based on the GHT is one of the most powerful techniques for image analysis [14,15]. However, real time applications using this method have not been practical due to the computational intensity required for similarity searching in a large centralized image collection. In this work, we describe a fast CBIR implementation using region-based GHT to retrieve visual objects under an affine transformation. In addition, content aware image segmentation is proposed to synchronize the image segmentation of query and target images.

This proposed approach has two phases. First, in the learning phase, a training procedure which works as the core of the proposed adaptive image segmentation identifies and stores stable parameters for segmenting each database image. In the retrieval phase, the proposed adaptive image segmentation process segments a query image into regions for retrieving visual objects inside database images. In addition, the region-based GHT is used to locate the target object under the affine transformation.

Fig. 1 shows the block diagram of the proposed algorithm. Although several methods have been proposed for detecting a visually important object in an image, the results are not satisfactory due to the shortage of image understanding models. Instead of proposing an automatic visually important object detector, the proposed visual object retrieval system provides a query-by-example user interface that lets the web based end user

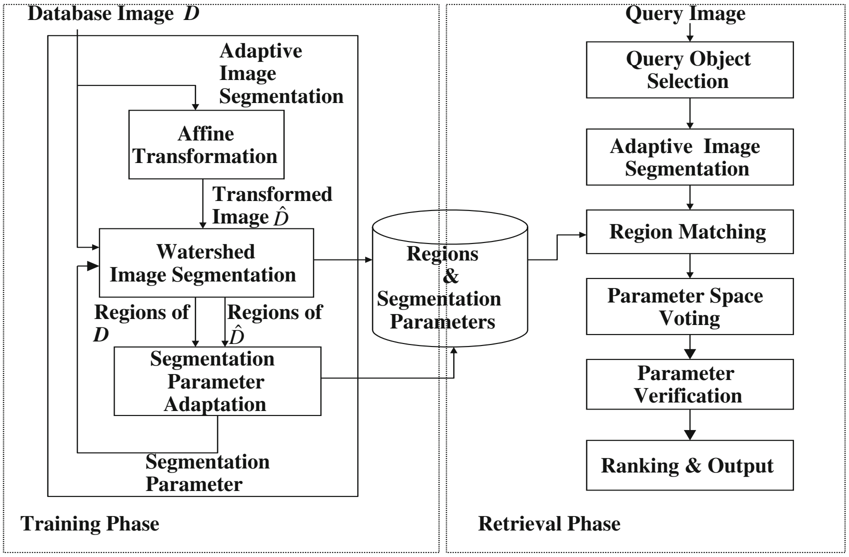


Fig. 1. Block diagram of the proposed image retrieval system.

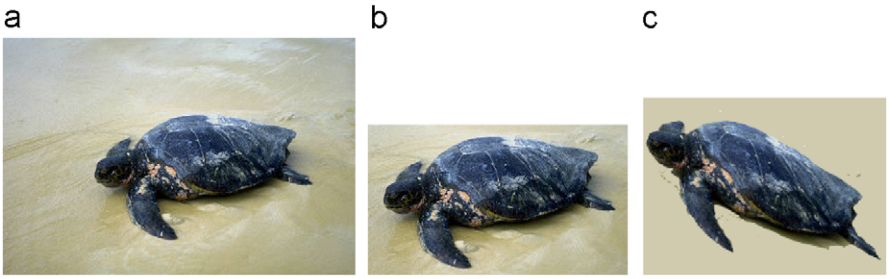


Fig. 2. Object model selection: (a) the example image, (b) the selected object, and (c) the scaled, translated, and rotated version of (b).

crop a sample object from an image and submit it as a search query. The user selects a thumbnail image to represent the full image, and then uses the available selection tools to crop a portion of the image as a sample query object, which can be further scaled, translated, and rotated. The selected object is also called a region-of-interest (ROI) in this work. Fig. 2 shows an example image with a selected object. In this paper, we use the region-based GHT for perceptually grouping segmented image regions of the selected query object. A novel voting scheme for the region-based GHT is proposed to provide an object search method capable of finding a target object of arbitrary position, orientation, and scaling.

The remainder of this paper is organized as follows. Section 2 presents the region-based GHT. Section 3 introduces the proposed segmentation scheme. Section 4 introduces the object search method. Section 5 shows the experimental results. The concluding remarks are given in the last section.

2. Object search using region-based GHT

The GHT [14,15] represents an object with an R-table, which establishes the relationships between every edge point of the object and an object reference point. Based on this representation, however, detecting an object of arbitrary position, orientation, and scaling would give the GHT high computational complexity. In addition, the voting result is generally inaccurate due to noisy edge points and object occlusion. To resolve these two problems, this work proposes a region-based technique for the GHT.

As shown in Fig. 3, a visual object (Fig. 3(a)) consisting of a number of regions may undergo an affine transformation with respect to a target image (Fig. 3(b)). Assume that there exists a subset of regions that share the same affine transformation mapping the visual object into the target image. As shown in Fig. 3(c), the visual object can be described by the geometric relationship between the object centroid X^C and the centers of the regions in the object. Given a region center X, the coordinates of X and X^C have the following relationship:

$$\begin{bmatrix} x^C \\ y^C \end{bmatrix} = \begin{bmatrix} x + r\cos\alpha \\ y + r\sin\alpha \end{bmatrix} \tag{1}$$

where r denotes the Euclidean distance between X and X^C, and α is the angle between the line passing X and X^C, and the x-axis. Note that, if the value of α is determined, the coordinates of X^R can be determined from those of X using Eq. (1). Accordingly, the R-table of the region-based GHT for representing a visual object can be constructed as

$$\begin{array}{ll} c_1 & (r_1^1, \alpha_1^1, |R_1^1|, \phi_1^1), (r_1^2, \alpha_1^2, |R_1^2|, \phi_1^2), \ldots, (r_1^{n_1}, \alpha_1^{n_1}, |R_1^{n_1}|, \phi_1^{n_1}) \\ c_2 & (r_2^1, \alpha_2^1, |R_2^1|, \phi_2^1), (r_2^2, \alpha_2^2, |R_2^2|, \phi_2^2), \ldots, (r_2^{n_2}, \alpha_2^{n_2}, |R_2^{n_2}|, \phi_2^{n_2}) \\ c_3 & (r_3^1, \alpha_3^1, |R_3^1|, \phi_3^1), (r_3^2, \alpha_3^2, |R_3^2|, \phi_3^2), \ldots, (r_3^{n_3}, \alpha_3^{n_3}, |R_3^{n_3}|, \phi_3^{n_3}) \\ \vdots & \\ c_k & (r_k^1, \alpha_k^1, |R_k^1|, \phi_k^1), (r_k^2, \alpha_k^2, |R_k^2|, \phi_k^2), \ldots, (r_k^{n_k}, \alpha_k^{n_k}, |R_k^{n_k}|, \phi_k^{n_k}) \end{array} \tag{2}$$

where c_i, $i = 1, \ldots, k$, denote colors for indexing these regions, and $|R_i^j|$ and ϕ_i^j denote the area and the orientation of the major axis of the jth region of color c_i, respectively. Other kinds of region features may be included as well. The orientation of the major axis of a region can be obtained from the central moments of the region as

$$\phi_A = \frac{1}{2}\tan^{-1}\frac{2\mu_{1,1}}{\mu_{2,0} - \mu_{0,2}} \tag{3}$$

where $\mu_{s,t}$ is the $(s+t)$th central moment of the region [16]. The edge points in a traditional R-table are indexed by the tangent slopes of the edge points, whereas the regions in the proposed R-table are indexed by the colors of the regions. Unlike the exact matching of the original R-table indexing mechanism, a color similarity between two colors is calculated for indexing the proposed R-table.

Based on the user selected object template, we can segment the template object into several regions, construct an R-table for these regions, and perform the object search from the target image using the region-based GHT with the constructed R-table. Considering a region R of color c, which is a part of the target object in the target image, the centroid candidates (x^C, y^C) of the target object in the target image can be located on

$$\begin{bmatrix} x^C \\ y^C \end{bmatrix} = \begin{bmatrix} x_R + r(c)s\cos(\alpha(c) + \tau) \\ y_R + r(c)s\sin(\alpha(c) + \tau) \end{bmatrix} \tag{4}$$

where (x_R, y_R) are the coordinates of the center of R; $r(c)$ and $\alpha(c)$ return the r and α values corresponding to the entry of the R-table of color c; s and τ are the given scaling factor and rotation angle, respectively. Then, votes are cast for the parameter vectors $(s,\ \tau,\ x^C,$ and $y^C)$ in an s–τ–x–y parameter space. In practice, all possible values of s and τ should be evaluated; however, it will be shown later that the parameters s and τ can be roughly estimated from the square root of the area ratio and the angle difference between the major axes of two regions, respectively. Furthermore, a similarity measure (support) between a region of the target image and a region of the template object, which will be defined later, is also calculated. When the support value of a region of the

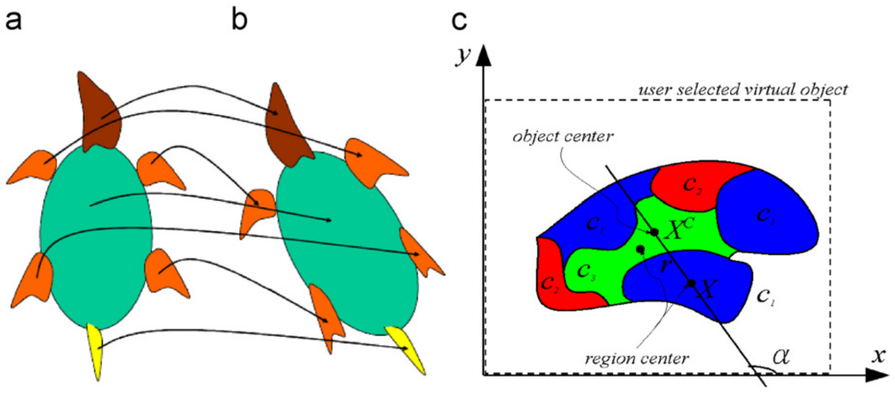


Fig. 3. Using region matching to estimate the affine transform parameters of an arbitrary object: (a) the visual object, (b) the target object, and (c) the region-based model for generalized Hough transform.

target image with respect to a region of the template object is too low, that region is not considered to have a vote. This can dramatically reduce the number of spurious peaks in the resulting parameter space. The details of the voting scheme will be discussed later. After all regions of the target image have been processed, the parameter vector in the 4D parameter space getting a large number of votes describes an affine transformation that makes the regions of the query object coincide significantly with many of the target image regions. Then the problem of visual object extraction is transformed into one of detecting the peaks in the s–τ–x–y parameter space.

3. Segmentation strategy for region-based GHT

The effectiveness of the region-based GHT for visual object retrieval is strongly affected by the accuracy of image segmentation. A similar approach to grouping regions through the GHT for visual object retrieval is found in the work of Chau and Siu [17], which does not achieve good performance if the deficiencies of automatic segmentation are ignored. Consider two images with a common visual object—the segmented regions constituting the visual object in the individual images might be different under differing lighting conditions or when the object is geometrically transformed. Obviously, the segmentation results vary, and depend on the specific segmentation algorithm performed based on the segmentation parameters. For example, for most existing segmentation algorithms, we must determine a threshold to merge two indistinguishably small regions into a larger region. The problem is that it is very difficult to use a single threshold to obtain a perfect segmentation result. The parameter setting is obviously not trivial. Instead of dealing with the problem separately, in this paper, we use a training mechanism to adaptively select stable segmentation parameters for each image. These stable parameters are defined as values that lead to the same set of segmented regions for an image and its transformed versions.

3.1. Problem definition

Given an image D, a set of regions from the initial segmentation using a parameter set Λ are obtained:

$$R_\Lambda = \{R_i : i = 1,\ldots,N\}, \quad R_i = (F_i, \overline{x}_i, \overline{y}_i) \tag{5}$$

where $R_i \cap R_j = \phi$, $\forall i \neq j$, $i,j = 1,2,\ldots,N$; F_i represents the pixel values of region i; $(\overline{x}_i, \overline{y}_i)$ are the coordinates of the centroid of region i. Let $\hat{D}$ be the transformed version of D and segmented into region set $\hat{R}_\Lambda = \{R_i : i = 1,\ldots,\hat{N}\}$ using Λ. Note that N does not necessarily have the same value as $\hat{N}$ because it is impossible to determine a parameter set for a segmentation algorithm that is resilient to any kind of transformation.

The stability of Λ can be defined by the conditional probabilistic model:

$$P(\hat{R}_\Lambda | R_\Lambda) \in [0,1] \tag{6}$$

The problem of optimally determining the segmentation parameters can be formulated as

$$\Lambda^* = \arg\max_{\Lambda} \hat{p}(\hat{R}_\Lambda | R_\Lambda). \tag{7}$$

As mentioned above, the segmented regions are used for matching. The stability of Λ can then be defined as the similarity between D and $\hat{D}$ in terms of region features. More specifically, the conditional probabilistic model $\hat{p}(\hat{R}_\Lambda | R_\Lambda)$ can be estimated as

$$\hat{p}(\hat{R}_\Lambda | R_\Lambda) = 1 - \frac{|\nu(T) - \min(\hat{N},N)|}{\min(\hat{N},N)} \tag{8}$$

where $\hat{N}(N)$ the number of regions in $\hat{R}$ (R), and $\nu(T)$ is the peak of votes for mapping $\hat{R}$ to R using the GHT, taking into account the predefined geometric transformation parameterized by $T=(s,\tau,x,y)$, which is given to transform D to $\hat{D}$. The value of $(\hat{p}(\hat{R}_\Lambda | R_\Lambda))$ is close to one if the parameter space peak s–τ–x–y is significant, and close to zero if the transformed object difference is large.

3.2. Watershed segmentation algorithm

This work uses a modified watershed segmentation algorithm [18] to divide an image into multiple regions, because the algorithm is very simple and involves only a single segmentation parameter.

The watershed algorithm segments regions into catchment basins based on the concept of watersheds in topography. A catchment basin is the set of points constituting the local minimum of a height function that is often defined as the gradient magnitude of the image. More specifically, image data may be interpreted as a topographic surface, with the pixel values representing altitude. Thus, region edges correspond to high watersheds and low gradient region interiors correspond to catchment basins. After locating these minima, the surrounding regions are incrementally flooded to form boundaries of the regions where flooded regions touch. One disadvantage of the process is that it leads to a severely over segmented image, with hundreds or thousands of catchment basins. Marker controlled segmentation and other approaches have been suggested to generate good segmentation. In a marker controlled segmentation approach, markers constrain the flooding process inside their own catchment basins; therefore, the final number of regions is equal to the number of markers.

Developing the watershed segmentation algorithm based on the concept of watersheds and catchment basins is complex, with many of the early methods resulting in either slow or inaccurate execution. In addition, merging regions based on region markers or other approaches can result in instable segmentation, as shown in Fig. 4. Furthermore, it is difficult to determine the appropriate number of markers for merging regions without any prior knowledge. In practice, we do not involve region merging to obtain relatively stable segmentation results for region-based matching.

In this paper, we introduce a threshold value T_l and a simple region growing process to generate initial catchment basins. Given input image I, the gradients of I are obtained with a Gaussian filter and computing partial derivations with respect to x and y on its pixels [18]. Let g_{xy} be the magnitude of the gradient of the pixel at location (x,y). I is converted to a binary image B by

$$b_{xy} = \begin{cases} 0 & \text{if } g_{xy} \leq T_l \\ 1 & \text{otherwise} \end{cases} \tag{9}$$

where b_{xy} is the value of the pixel at location (x, y) in B. Then, the regions that consist of continuous pixels in B with the same value 0 are grown together to form the initial catchment basins of I. Finally, after locating these minima, the surrounding regions are incrementally flooded to form boundaries of the regions where flood regions touch. For the sake of illustration, the watershed segmentation algorithm is summarized as follows.

Algorithm 1. The Watershed Segmentation

```
Input: Image $I$.
Output: A set of segmented regions of $I$.
Method:

(1) Filter with a Gaussian filter with a standard deviation of $\sigma_G$ on $I$.
```

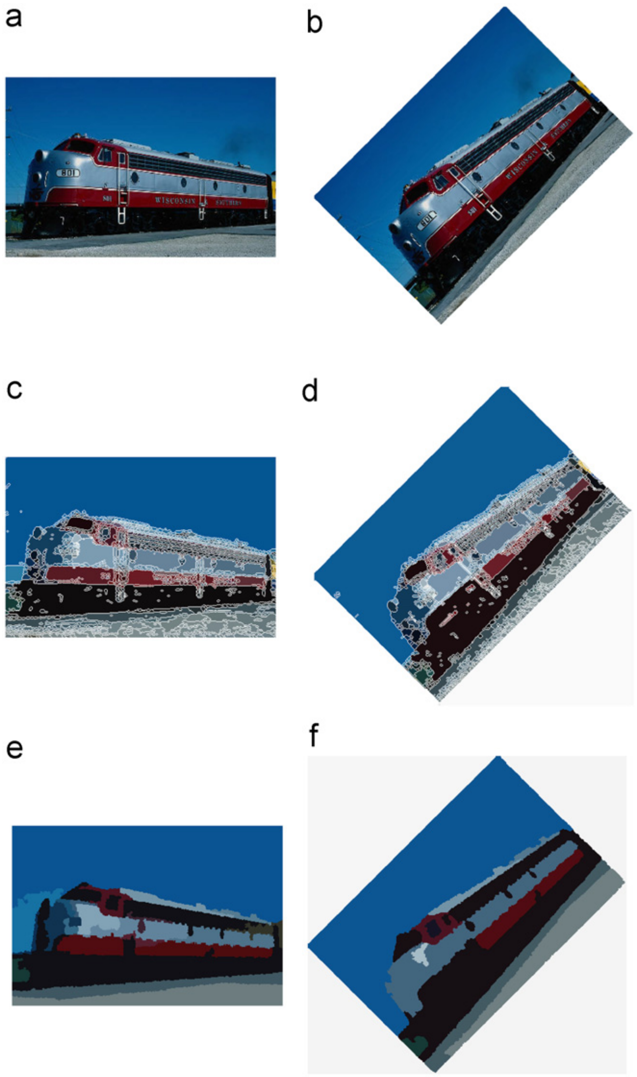


Fig. 4. The problem of instability in merging regions over segmented by the watershed segmentation algorithm: (a) the original image, (b) the transformed image of (a), (c) the segmentation result of (a), (d) the segmentation results of (b), (e) the result of merging small regions in (c) with their adjacent regions, and (f) the result of merging small regions in (d) with their adjacent regions.

(2) For each pixel $I(x, y)$, compute partial derivations $p_x=\partial I/\partial x$ and $p_y=\partial I/\partial y$ with respect to x and y, respectively, using the following two masks:

$$\begin{matrix}-1 & 0 & 1\\ -1 & 0 & 1\\ -1 & 0 & 1\end{matrix}\qquad \begin{matrix}1 & 1 & 1\\ 0 & 0 & 0\\ -1 & -1 & -1\end{matrix} \tag{10}$$

(3) For each pixel $I(x, y)$, compute the magnitude of its gradient from the partial derivatives:

$$g_{xy}^2=p_x^2+p_y^2 \tag{11}$$

(4) Set the gradient threshold T_l and convert I into a binary image B using Eq. (9).

(5) Perform a simple region growing process on B to create a set of initial catchment basins for I.

(6) Set $k=T_l+1$.

(7) Repeat until all pixels in I are labeled:

  (7.1) For each catchment basin L_i, compute its geodesic influence zone, which is defined as the locus of non-labeled image pixels of gradient magnitude k that are contiguous with the catchment basin L_i, for which the distance to L_i is smaller than the distance to any other catchment basin L_j. Label all pixels belonging to the catchment basin L_i influence zone L_i.

  (7.2) $k=k+1$.

Fig. 5 shows an example of watershed segmentation. Note that the value of threshold parameter T_l in Eq. (9) affects the final segmentation result of the algorithm—a larger T_l produces fewer segmented regions.

3.3. The adaptive image segmentation with a training procedure

Determining the optimal segmentation parameters for all affine transformation parameters is computationally complex. As mentioned above, segmentation results from watershed segmentation strongly depends on the gradient magnitude of the input image. Looking at the watershed segmentation presented in Algorithm 1, it is interesting to find that Steps 1–4 implement a simplified version of Canny's edge detector [19] and the binary image B defined by Eq. (9) is the edge map of I. The candidate boundaries of seamless regions consist of edges in B. The choice of T_l is important for boundary detection. Too low a threshold produces too many false edges, which would lead to an over segmented result from the watershed segmentation. On the other hand, too high a threshold would throw away too many true edges and result in under segmentation. To quantify this tradeoff, the noise behavior of the segmentation algorithm should be analyzed in detail.

Any affine transformation on an input image produces noise in the segmentation results of the watershed algorithm. However, it is difficult to eliminate every kind of noise through a training process that determines a robust threshold for segmenting a specific image. Without a loss of generality, we assume that the noise at each image pixel is stationary, white (independent), Gaussian noise $N(x, y)$ with a mean $=0$ and variance $=\sigma_n{}^2$. Following the edge detection analysis presented by Lee and Cok [20], the noise behavior of watershed segmentation is discussed as below.

The partial derivatives $p_x=\partial I/\partial x$ and $p_y=\partial I/\partial y$, obtained from Step 2 of Algorithm 1, can be shown to be independent, and their variances are given by [20]

$$\sigma_d^2=\sigma_{P_x}^2=\sigma_{P_y}^2\approx\frac{\sigma_n^2}{4\pi\sigma_G^2}(6+8c-2c^4-8c^5-4c^8) \tag{12}$$

where $c=\exp[-1/(4\sigma_G^2)]$. When σ_G is large, $c\approx 1-1/(4\sigma_G^2)$, and

$$\sigma_d^2=\sigma_{P_x}^2=\sigma_{P_y}^2\approx\frac{2\sigma_n}{\sqrt{2\pi\sigma_G^2}}. \tag{13}$$

According to Eq. (13), the noise standard deviations of the partial derivatives are reduced approximately by a factor of σ_G^2 when we smooth an input image with a Gaussian filter of size σ_G. The distribution of the gradient magnitude g_{xy}^2, defined by Eq. (11) is a χ^2 distribution, if p_x and p_y are Gaussian random variables. According to Lee and Cok [20], some interesting statistical characteristics of the distribution of g_{xy}^2 can be found: (1) the peak occurs at $g_{xy}^2=\sqrt{2m-1}\sigma_d^2$; (2) the mean is $2m\sigma_d^2$; and, (3) the variance is $4m\sigma_d^4$, where $m=1$ for grey level images and $m=3$ for color images. Also,

$$T_l=2\sqrt{2m-1}\sigma_d \tag{14}$$

was suggested to obtain reasonable boundary points [20].

Although Eq. (14) provides a good initial guess for the value of the threshold T_l, T_l should be fine tuned by the proposed training procedure since the noise model of an input image might not be

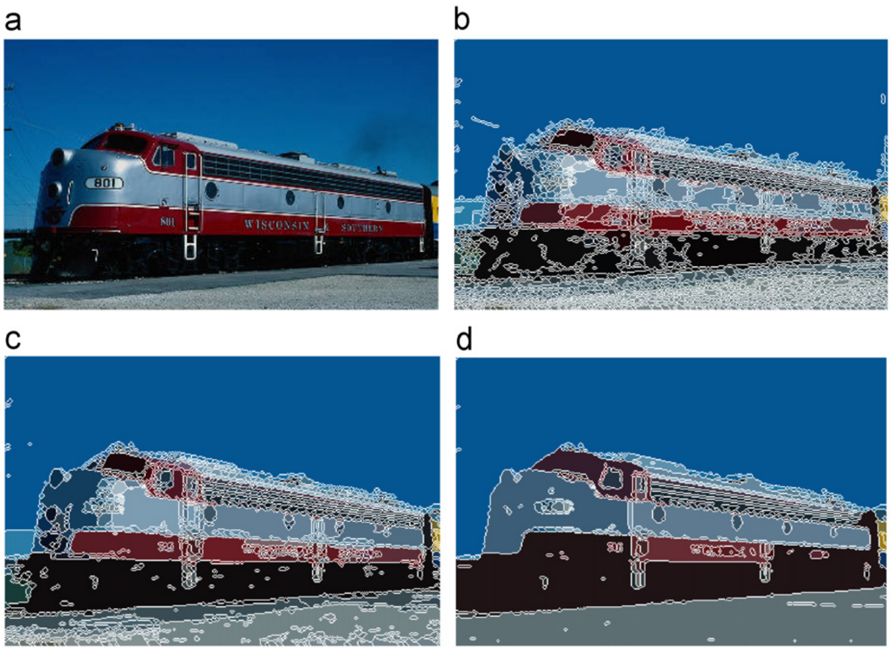


Fig. 5. A segmentation example: (a) the original image, (b)–(d) are the segmentation results of (a) using the threshold parameters 20, 45, and 115, respectively.

exactly Gaussian. Let $\tilde{D}$ be the transformed version of a database image D using affine transformation A, which is known in advance in the training phase. The difference image C, between $\tilde{D}$ and D, can be computed as

$$C = D - A^{-1}\tilde{D} \tag{15}$$

Actually, C is the noise image produced by A. The estimated noise variance is then easy to compute as

$$\sigma_n^2 = \frac{1}{N_x N_y}\sum_{x=1}^{N_x}\sum_{y=1}^{N_y}(n_{xy} - \overline{n})^2, \quad \overline{n} = \frac{1}{N_x N_y}\sum_{x=1}^{N_x}\sum_{y=1}^{N_y} n_{xy} \tag{16}$$

where n_{xy} is the pixel value at the location (x, y) of C. Based on Eq. (16), we can estimate the value of σ_d and the initial threshold $T_l^{(0)}$using Eqs. (13) and (14), respectively. Finally, we search the best threshold from the interval $[T_l^{(0)} - \alpha\sigma_d,\ T_l^{(0)} + \alpha\sigma_d]$ using the criterion function defined in Eq. (8). In practice, we set $\sigma_G = 4$ and $\alpha = 0.5$.

The training procedure presented so far cannot make sure to be affine invariant. Matas et al. [21] proposed an interesting approach to achieve the goal of affine invariant by regulating the most robust image level sets and level lines. The method normalizes all of the six parameters in the affine transform. Following the concept of extreme regions in [21], the proposed training procedure is dedicated to find the segmentation parameter T_l which generates the maximally stable extreme regions (MSERs) for an input image. MSERs are defined as maximally contrasted regions in the following way. Let $R_1, \ldots, R_{i-1}, R_i, \ldots$ be a sequence of nested extreme regions $R_i \subset R_{i+1}$, where R_i is defined by a threshold at level i. Based on the area variation, in [21], an extreme region in the list R_{i*} is said to be maximally stable if $R_{i*} = \arg\min_i |R_{i+1} \backslash R_{i-1}|/|R_i|$, where $|R|$ is the area of a region R. As mentioned above, the segmentation results highly affect the retrieval accuracy for a specific similarity measure. Thus, in this work, the definition of MSERs is reformulated as follows.

$$R_{i*} = \underset{i}{\operatorname{argmax}}[2\hat{p}(R_i|R_{i-1}) - \hat{p}(R_{i-1}|R_{i-2}) - \hat{p}(R_{i+1}|R_i)] \tag{17}$$

where $\hat{p}(R_i|R_{i-1})$ is the conditional probability defined in (8) to measure the similarity between MSERs at levels i and $i-1$.

The following algorithm summarizes the training procedure for determining a relatively stable threshold T_l for image segmentation.

Algorithm 2. The adaptive image segmentation.

Input: An image D and its transformed version $\tilde{D}$ using affine transformation A.

Output: The threshold T_l for segmenting D and the MSERs.

Method:

```
(1) Compute the noise image $C$ using Eq. (15).
(2) Compute the noise variance $\sigma_n$ using Eq. (16).
(3) Compute the estimate of $\sigma_d$ using Eq. (13).
(4) Determine the initial threshold $T_l^{(0)}$ using Eq. (14).
(5) Set $T_l^{(1)} = T_l^{(0)} - \alpha\sigma_d$, $i = 1$, and $p^* = 0$.
(6) while $(T_l^{(i)} T_l^{(0)} + \sigma_d)$do
    (6.1) Perform the watershed segmentation to generate region
          sets $R_i$ for $D$ using the threshold parameter $T_l^{(i)}$.
    (6.2) Compute the value of $\hat{p}(R_i|R_{i-1})$ $(R_i|R_{i-1})$ using Eq. (8).
    (6.3) $T_l^{(k+1)} = T_l^{(k)} + \Delta T$, $k = k+1$.
(7) Perform affine normalization to obtain MSERs $R_{i*}$ and
    $T_l^* = T_l^{(k)}$ using (17).
```

To speed up the segmentation algorithm in finding out the MSERs, an incremental segmentation algorithm which is slightly different than the proposed watershed segmentation algorithm is presented. Basically, the watershed segmentation consists of four parts: computation of the gradients (Steps 1–3); generating the edge map (Step 4); growing the initial catchment basins (Step 5); and the water flooding process (Steps 6 and 7). Considering two thresholds parameters T_l and T_l' $(T_l < T_l')$ for consecutive segmentation of a query image, we find that: (1) the gradients are the same, (2) the edge maps for case $_l$can be derived from those of T_l, and, (3) the initial catchment basins in case T_l' can be derived from for case T_l. Adding suitable memorization functions to these parts of the watershed segmentation can dramatically reduce the

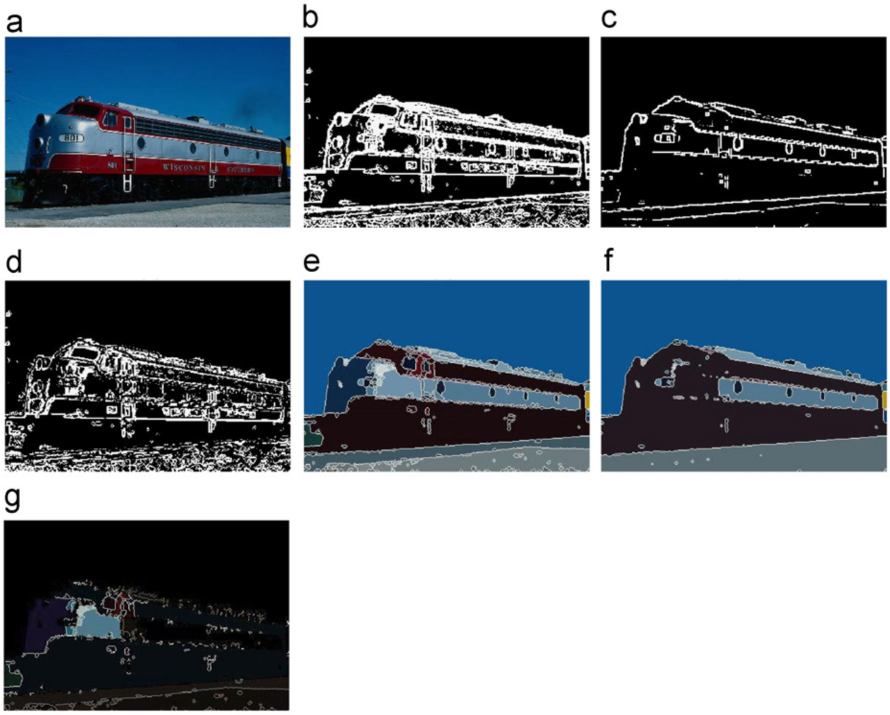


Fig. 6. An example of incremental image segmentation: (a) the original image; (b) and (c) are edge maps of (a) using the thresholds 80 and 150, respectively; (d) is the difference between (b) and (c); (e) and (f) are the segmented regions of (a) using the thresholds 80 and 150, respectively; and (g) is the difference between (e) and (f).

segmentation complexity required to answer a query. The first three parts of the segmentation algorithm are time consuming compared with the final water flooding process. Fig. 6 shows an example of incremental image segmentation, which demonstrates the high correlation between two segmented results using different threshold parameters.

4. The object search method

4.1. Model selection and object matching

Our object retrieval system provides a query-by-example end user interface that allows a user on the web to crop a sample object from an image and submit it as a search query. The user selects a thumbnail image of the full image, and then uses the available selection tools to crop out a portion of the image as the sample query object, which can be further scaled, translated, and rotated.

The user selected object image is represented by an R-table using Eq. (2) and sent to the server for matching against the database images. The R-table of the submitted model object is considered an appropriate structure for the object—image structures must be found according to the region information in the R-table. For each region in the R-table, we need to search for possible matches from the target image by recovering 2D rigid object translation, scale, and rotation. The sequence of steps for object matching is shown in Fig. 7: (a) user model selection, (b) image segmentation and region parameters computing, (c) R-table construction, (d) region matching and testing, (e) object centroid determination, and scaling factor and rotation angle calculation, (f) voting on the s–τ–x–y space, and (g) peak detection and parameter verification. One advantage of the proposed method is that the small number of image segments results in a fast object search process.

4.2. Parameter computation for geometric transformation by voting

To increase the robustness of the estimation process, as well as to reduce computation time, the transformation parameter vector $V(s, \tau, x, y)$ between two similar images can be obtained through the following process. Assume that the R and R' are corresponding regions from two different images; the scaling factor s is determined by the square root of the ratio of area R' to area R:

$$s=\sqrt{\frac{|R'|}{|R|}} \tag{18}$$

To find the rotation angle τ, we can use the degree of misalignment between the major axes of R and R':

$$\tau=\phi_{R'}-\phi_R \tag{19}$$

where $\phi_{R'}$ and ϕ_R are the orientations of the major axes of R' and R, respectively. Given a region, the direction of the major axis can be obtained by Eq. (3). Furthermore, for each point $u=(x, y)$ in R, one can find the corresponding point $u'=(x',y')$ in R', such that

$$\begin{pmatrix} x'-\overline{x'} \\ y'-\overline{y'} \end{pmatrix}=s\begin{pmatrix} \cos\tau & \sin\tau \\ -\sin\tau & \cos\tau \end{pmatrix}\begin{pmatrix} x-\overline{x} \\ y-\overline{y} \end{pmatrix} \tag{20}$$

where $(\overline{x'},\overline{y'})$ and $(\overline{x},\overline{y})$ are the centroids of R' and R, respectively.

Given a region match pair (R, R'), we compute the support value for the match pair in the RGB color space as

$$h(R,R')=\frac{2}{1+\exp^{\rho\times\varepsilon(R,R')}} \tag{21}$$

where parameter ρ controls the speed at which the support h achieves one of its two extremes (0 and 1) according to the value of $\varepsilon(R,R')$, which is computed from

$$\varepsilon(R,R')=\sum_{(x,y)\in R}||\vec{c}_R(x,y)-\vec{c}_{R'}(x',y')|| \tag{22}$$

where the relationship between (x',y') and (x,y) is defined in Eq. (20), $\vec{c}_R(x,y)$ and $\vec{c}_{R'}(x',y')$ are the color vectors of the pixel

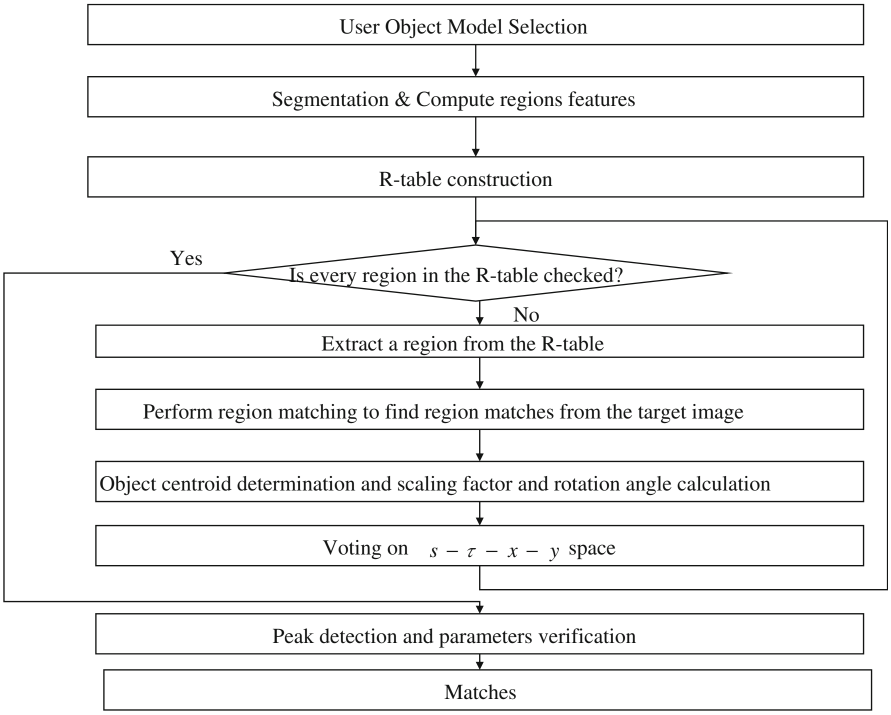

User Object Model Selection
Segmentation & Compute regions features
R-table construction
Yes
Is every region in the R-table checked?
No
Extract a region from the R-table
Perform region matching to find region matches from the target image
Object centroid determination and scaling factor and rotation angle calculation
Voting on $s-\tau-x-y$ space
Peak detection and parameters verification
Matches

Fig. 7. A flowchart of object matching.

at (x, y) in R and the pixel at (x', y') in R', respectively, and $\|.\|$ is the Euclidean distance. The value of h is about 1 if $\varepsilon(R, R')$ nears zero; h is about 0 if the difference between R and R' is large. The value of ρ is set to 0.01 in this study. Once the value of $h(R,R')$ is larger than a pre-defined threshold (i.e., 0.5), the region match pair (R,R') has a vote for a point in the s–τ–x–y parameter space as follows:

$$++(s,\tau,x^{ref},y^{ref}) \tag{23}$$

where (x^{ref}, y^{ref}) is computed using Eq. (4) and $(s, \tau, x^{ref}, y^{ref})$ is a point in the 4D parameter space. On the other hand, a small $h(R,R')$ indicates that the region match pair (R,R') is not good enough to be included in the voting process.

The time to locate the peak corresponding to the query object in the s–τ–x–y parameter space can be reduced by first approximating s^*, τ^*, x^* and y^* and then refining the search in a small area. The approximating values of s^*, τ^*, x^*, and y^* can be determined by the procedure described below. Multiple peaks can be selected to detect the visual object with multiple geometric transformations, where each characterizes a region grouping (sub-object) of the visual object. The proposed region matching for voting is summarized as

Algorithm 3. Approximating geometric transformation parameter estimation (AGTPE).

Input: A selected query object Q, which is partitioned into a set of regions, and an R-table constructed for those regions; a database image D, which is partitioned into a set of regions $R'_j, j=1\ldots m$.

Output: The approximating values of geometric transformation parameters s^*, τ^*, x^*, and y^*.

Method:

```
/* $s$-hist[]: accumulated scores of the scaling factors */
/* $\tau$-hist[]: accumulated scores of the rotation angles */
/* H-hist[]:accumulated scores of the $xy$-plane */
for $j=1$ to $m$ do /* check every region in the target image */
    use the average color of region $R'_j$ to index the R-table, and let regions $R_i$, $i=1\ldots n$, be the regions in the retrieved entry of the R-table.
  for i=1 to n do /* check every region in Q */
    calculate geometric transformation parameters between $R_i$ and $R'_j$ using Eqs. (18) and (19);
    calculate the center coordinates $(\overline{x}_i,\overline{y}_i)$ and $(\overline{x'},\overline{y'})$ of $R_i$ and $R'_j$, respectively;
    calculate the support value $h(R,R')$ using Eq. (21);
  if $(h'(R,R') > \upsilon)$/* $\upsilon$: a pre-defined threshold */
    calculate the object reference point $(x, y)$ using Eq. (4);
    $\tau$ is quantized to an approximate value, $\tilde{\tau}$;
    $s$ is quantized to an approximate value, $\tilde{s}$;
    $\tau$-hist[$\tilde{\tau}$]++;
    $s$-hist[$\tilde{s}$]++;
    H-hist[$x$][$y$]++;
  set the approximating values of geometric transformation parameters $s^*$, $\tau^*$, $x^*$, and $y^*$ by detecting the peaks in the accumulation arrays $s$-hist[], $\tau$-hist[], and H-hist[], respectively.
```

Note that, in this approach, the translation and rotation terms are detected separately, thus, reducing the computational cost of the GHT with respect to the 4D parameter space. The time complexity of the proposed algorithm is $O(4\,mn)$. In general, since most areas in an image are uniform and the number of segmented regions is not large, the execution speed of the proposed approach is fast.

4.3. Geometric transformation parameter verification

After performing the proposed AGTPE method, we know the approximating values of the geometric transformation parameters s, τ, x, and y, and can apply them to mapping visual object in the selected model to the corresponding object in the target image. Let $\Lambda=(s,\tau,x,y)$ denote the parameter set of the geometric transformation corresponding to the peak in the s–τ–x–y space. Applying the geometric transformation with parameters Λ to the query object O_q determines the corresponding visual object O_t in target image. Then, the similarity between O_q and $O_{t,}$ based on the histogram intersection measure [7], is computed:

$$S(\Lambda)=\frac{\sum_{j=1}^{n}\min(H(O_q,j),H(O_t,j))(s)^2}{\sum_{j=1}^{n}H(O_q,j)(s)^2} \tag{24}$$

where $H(O_q)$ and $H(O_t)$ denote the color histogram of O_q and O_t, respectively, and n is the number of bins in the histograms.

We can construct a small search window in the parameter space with the parameter set $\Lambda^*(s^*,\tau^*,x^*,y^*)$ as its center; then we can check the parameter sets one by one within the search window using the same parameter verification process to find the best parameters nearing Λ^*. Although this process introduces additional time to fine tune the geometric transformation parameters, our experimental results show that it significantly improves the retrieval accuracy.

Once we have the verified parameters, the image is reported as a match and its object match measure S is also returned, if S is large enough. After obtaining match measures for all images in the database, the measures are sorted in descending order. The number of matches can further be restricted to the top k if necessary.

5. Experimental results

The proposed system was implemented on an AMD Athlon 64 3000+ PC with 512 MB memory. Three test databases were used to demonstrate the performance of the proposed system. First, an artificially created database containing 600 color images of six types was constructed to test the robustness of the system. A retrieval example based on the test database is shown in Fig. 8. Fig. 9 shows the second database, consisting of 5000 color images, each of which contains a meaningful visual object sorted in 25 classes. The third database consists of 20,000 color scenery images sorted into one hundred classes, which are from Corel's CorelPhoto image collections. Each database image is 384 × 256. Query images of different sizes were extracted from these images.

A retrieval method is classified as accurate if, for a given query image, the perceptually (to a human) most similar image in the database is retrieved as the top selection. Also, a robust system should be stable for all types of queries, i.e., the system must not break down under specific samples. To test the robustness of the proposed system, normal query images were supplemented with translated, rotated, scaled, and noise added query images to test the system.

Six types of query images are presented below.

- *Normal*: Every image in the database was presented as the query image.
- *Translating*: The object in every database image was translated and then presented as the query image.
- *Cropping*: The object in every database image was selected and then presented as the query image.
- *Rotating*: Every image in the database was rotated arbitrarily and then presented as the query image.
- *Scaling*: Every image in the database was scaled and presented as the query image.
- *Noise added testing*: Zero-mean normal noises of 10, 15, and 20 db were added to every image in the database and presented as the query image.

The region-based retrieval technique proposed by Chau and Siu [17] (CGHT for short) was also implemented for performance comparison. Table 1 presents the results of the proposed method and Chau's method tested against 100 color images, where n refers to the position of the correct retrieval. The proposed method was superior in the presence of the geometric

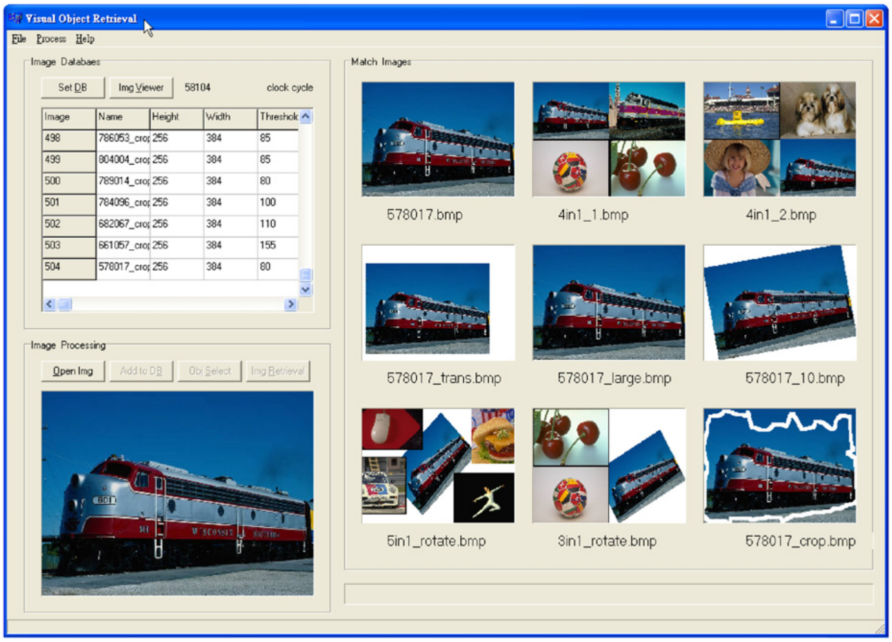


Image	Name	Height	Width	Threshok
498	786053_crop	256	384	85
499	804004_crop	256	384	85
500	789014_crop	256	384	80
501	784096_crop	256	384	100
502	682067_crop	256	384	110
503	661057_crop	256	384	155
504	578017_crop	256	384	80

Fig. 8. Retrieval results of the system using test database 1.

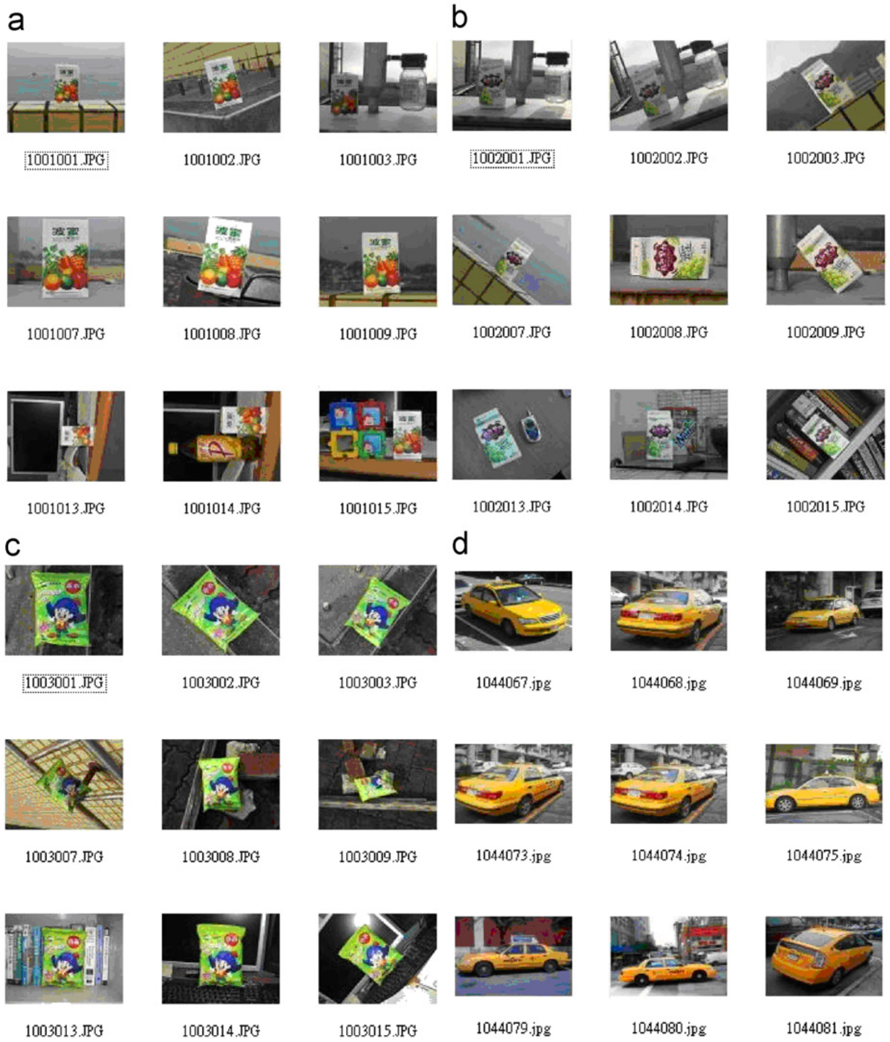


Fig. 9. Samples of test database 2 with image types: (a) juice box, (b) juice box 2, (c) Kuai-Kuai, and (d) taxi.

Table 1
Image retrieval results for the simulated methods based on test database 1: n refers to the position of the correct retrieval; the last column indicates the average retrieval time.

Test mode	Method	$n=1$ (%)	$n\leq 3$ (%)	$n\leq 5$ (%)	$n\leq 20$ (%)	Average retrieval time (s)
Normal	Chau	91	91	92	92	5.3
	Proposed	100	100	100	100	0.115
Scaling	Chau	6	7	8	11	5.3
	Proposed	100	100	100	100	0.115
Cropping 10%	Chau	91	91	91	92	5.3
	Proposed	98	100	100	100	0.115
Rotation	Chau	23	25	25	26	5.3
	Proposed	95	97	97	98	0.115
Translation	Chau	100	100	100	100	5.3
	Proposed	100	100	100	100	0.115
Noise-adding (20 db)	Chau	100	100	100	100	5.3
	Proposed	100	100	100	100	0.115
Noise-adding (15 db)	Chau	91	91	92	92	5.3
	Proposed	100	100	100	100	0.115
Noise-adding (10 db)	Chau	76	77	77	79	5.3
	Proposed	91	92	92	95	0.115

transformation. For the six types of query image, the proposed method seems more sensitive to rotated images than to scaled or translated images. The worst case retrieval accuracy of the system was 91%, which is much better than that of Chau's method, and the average retrieval time of the proposed method is much shorter.

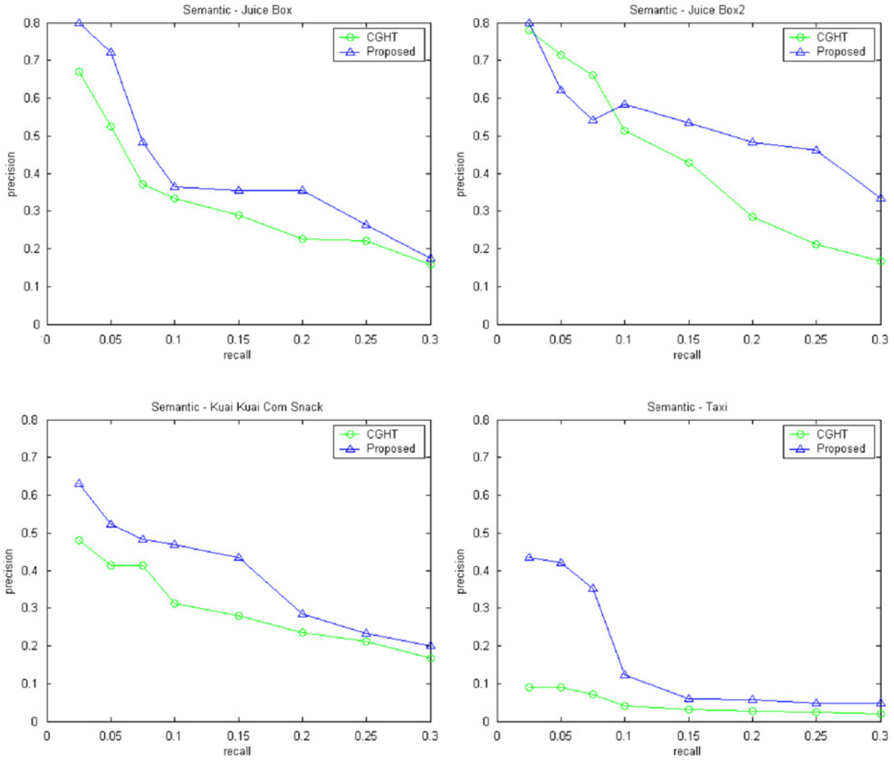


Fig. 10. Average precision-recall plots for the four test cases.

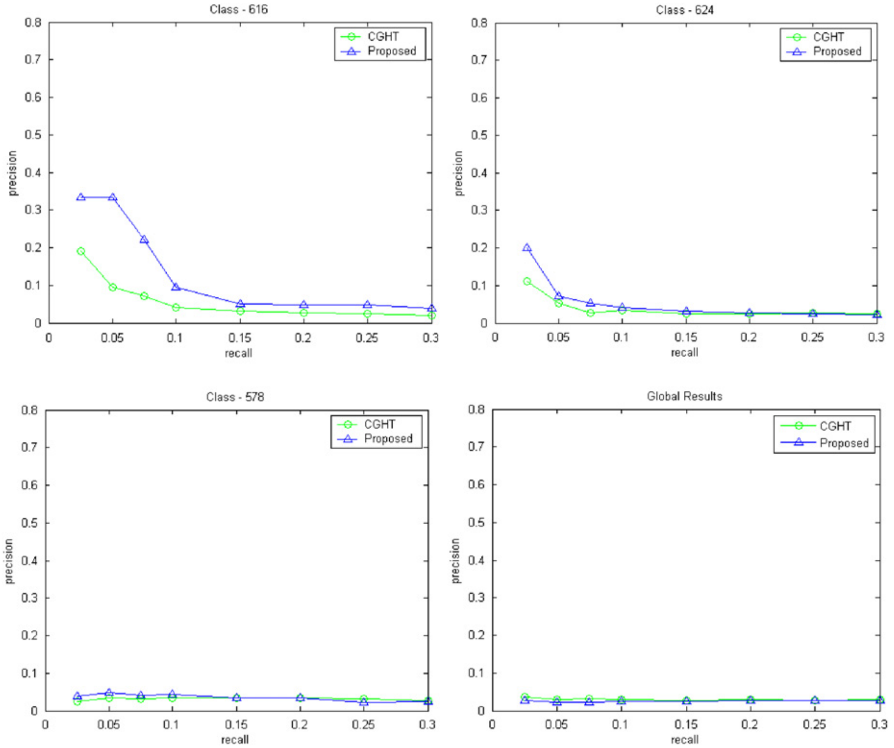


Fig. 11. Precision versus recall of queries using Corel's image collection.

We also evaluated our algorithm by calculating the precision and recall for visual object retrieval. Given ground truth labeling and predicted labeling of the visual objects obtained by annotating the visual objects in the query images using region-based image retrieval, let n_{tp}, n_t, and n_r be the number of true positives (correctly annotated objects), the number of true visual objects in the

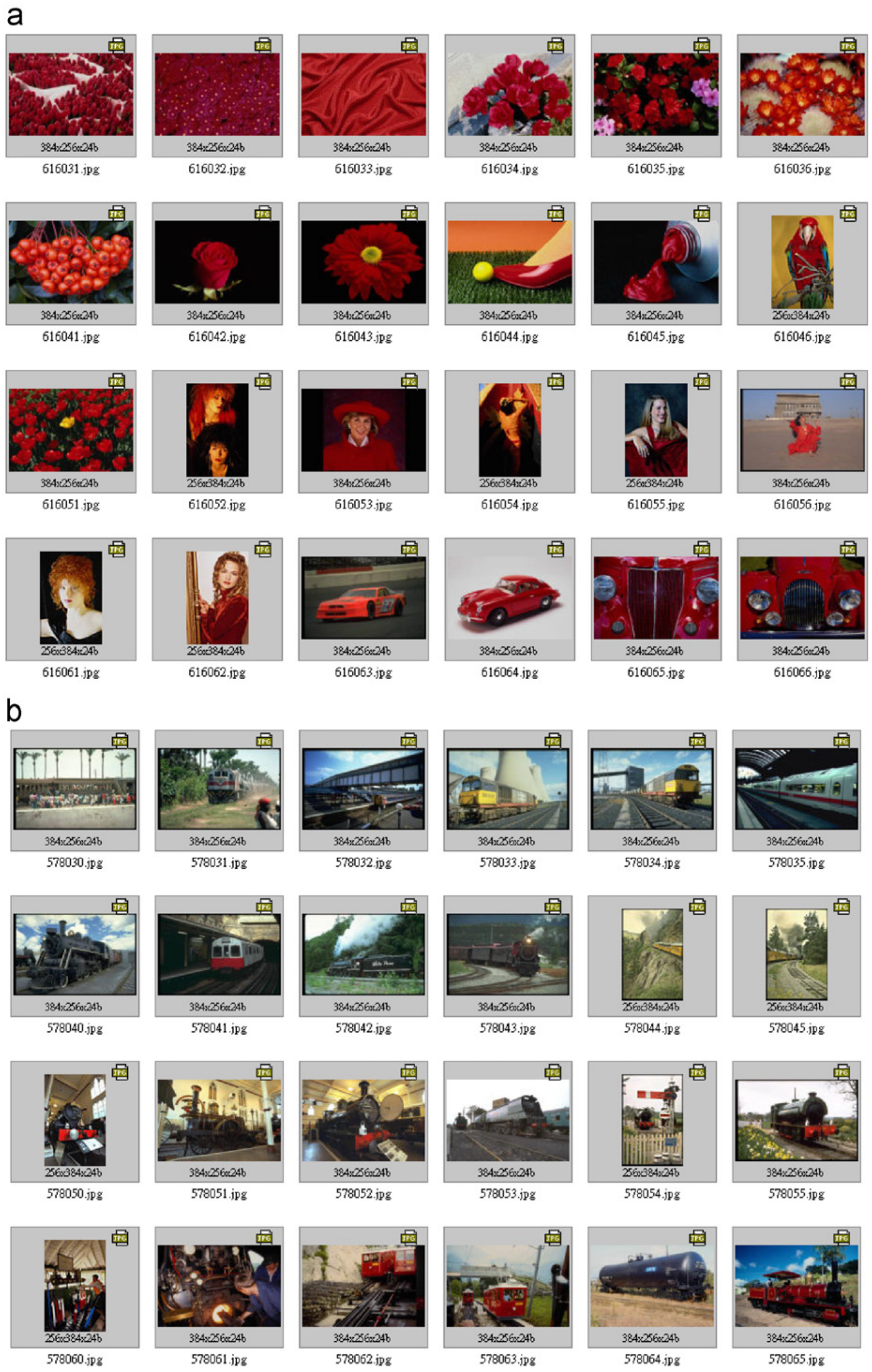


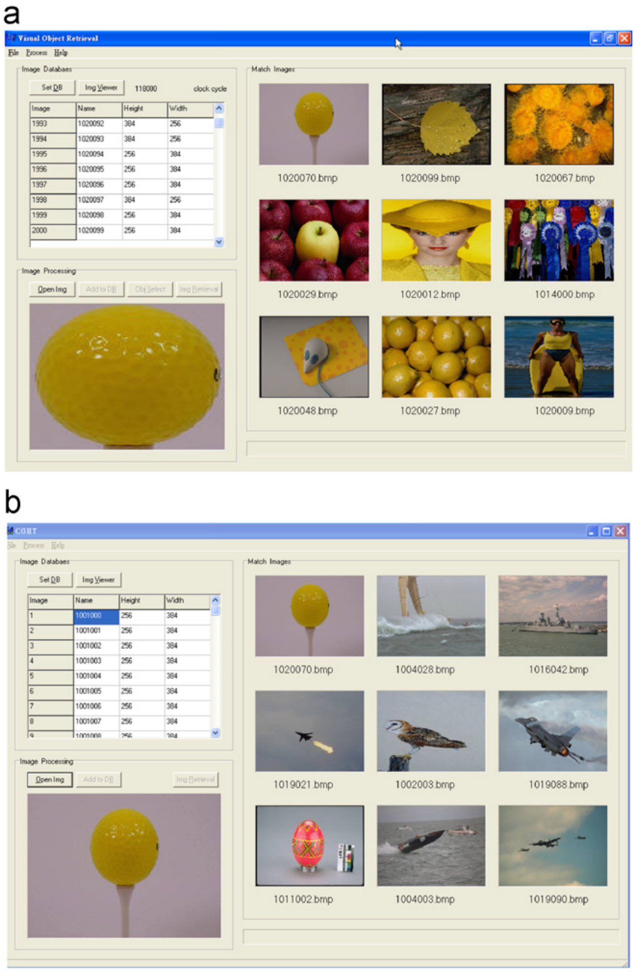
Fig. 12. Samples of Corel data images with image types: (a) Class-616 and (b) Class-578.

database, and the number of retrievals, respectively. Recall is defined as n_{tp}/n_t, meaning the proportion of true labels annotated by the algorithm. Precision is defined as n_{tp}/n_r, meaning the proportion of retrievals that are true. By varying n_r, we can vary the tradeoff between precision and recall. To summarize the precision-recall curve in one number, we can use the *F*-measure, which is the geometric mean [22]:

$$F = 2 \times \frac{precision \times recall}{precision + recall} \tag{25}$$

Recall and precision require a ground truth to assess the relevance of images for a set of significant queries. For each query image, we defined the relevant images based on human assessment, using test database 2. Fig. 9 shows a sample of test images. We compared the retrieval performance of the proposed method to Chau's method [17]. The average precision and recall curves are plotted in Fig. 10.

Finally, we used the Corel database to test our system, as shown in Fig. 11. The figures show that the proposed method achieves good results in terms of its retrieval accuracy compared to Chau's method [17]. However, including the proposed method,



(a)

Image	Name	Height	Width
1993	1020092	384	256
1994	1020093	384	256
1995	1020094	256	384
1996	1020095	256	384
1997	1020096	256	384
1998	1020097	384	256
1999	1020098	256	384
2000	1020099	256	384

(b)

Image	Name	Height	Width
1	1001000	256	384
2	1001001	256	384
3	1001002	256	384
4	1001003	256	384
5	1001004	256	384
6	1001005	256	384
7	1001006	256	384
8	1001007	256	384
9	1001008	256	384

Fig. 13. An example of precision-recall test based on the Corel database using the (a) proposed method and (b) CGHT.

region-based approaches to image matching in general suffer from bad performance when the image is feature sparsity [23]. Because there are few highly contracted level sets in the Corel data images (cf. Fig. 12), in Fig. 11, several of the figures evaluating precision and recall obtain very poor values for precision—this is the limit of shape-color based retrieval systems. However, in class 616 and 624, although they do not have the same object shape, the color characteristics are similar, and our system seems to work. Fig. 13 shows an example of precision-recall test using the Corel database. As shown in Fig. 14, to summarize the accuracy results over many samples, we also compared the performance of the simulated methods using the *F*-measure defined in Eq. (25). Accordingly, the proposed method outperforms Chau's method.

6. Conclusions

This paper presented an object search method using the GHT based on content aware image segmentation. By incorporating the proposed scheme for learning segmentation parameters, the adaptive image segmentation is used to segment maximally stable extreme regions from database and query images. This improves the retrieval effectiveness of the region-based GHT. In other words, the proposed method does not suffer from the problems of object segmentation in conventional object search approaches. The fusion of image segmentation and matching functions stabilizes the segmentation results for region-based image retrieval. Furthermore, the proposed method, using the region-based GHT to find the correct geometric transformation parameters in object searches, does not have the computational complexity of the traditional GHT.

The system retrieval speed is enhanced without affecting its robustness by advance sorting of database images in terms of segmentation parameters for the proposed incremental query segmentation scheme. Future work will deal with linking semantic interpretations into regions and increasing the database size.

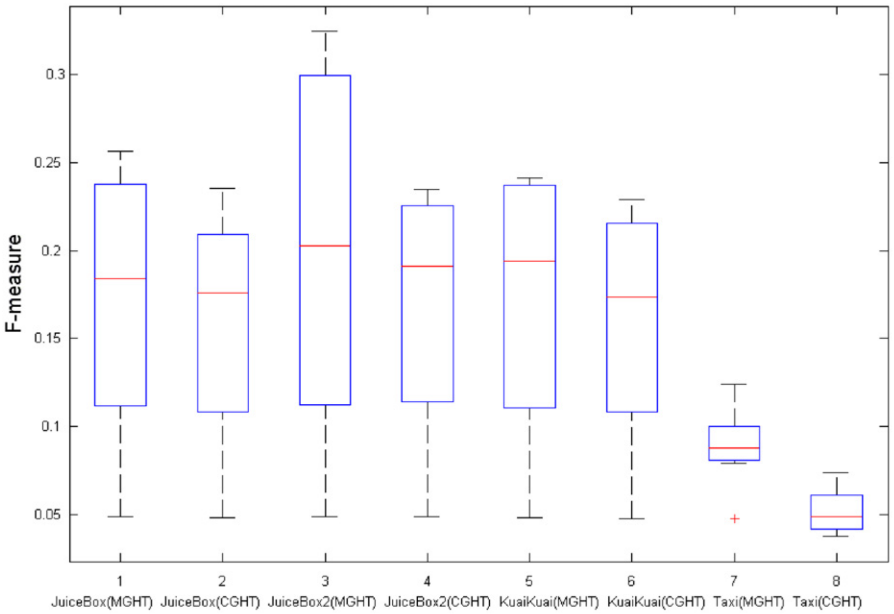


Fig. 14. Average *F*-measure versus number of retrievals.

Acknowledgments

This work was supported in part by National Science Council, Taiwan and the National Taiwan Ocean University under Grant nos. NSC 97-2221-E-019-032 and NTOU-RD961-05-02-01-01, respectively.

References

[1] P. John Eakins, Toward intelligent image retrieval, J. Pattern Recognition 1 (2002) 3–14.

[2] A.W.M. Smeulders, M. Worring, A. Gupta, R. Jain, Content-based image retrieval at the end of the early years, IEEE Trans. Pattern Anal. Mach. Intell. 22 (2000) 1349–1380.

[3] Yong Rui, T.S. Huang, Shih-Fu Chang, Image retrieval: current techniques, promising directions, and open Issues, J. Visual Commun. Image Represent. 10 (1999) 39–62.

[4] Y. Liu, D. Zhang, G. Lu, W.-Y. Ma, A survey of content-based image retrieval with high-level semantics, J. Pattern Recognition 40 (2007) 262–282.

[5] J.Z. Wang, J. Li, G. Wiederhold, SIMPLIcity: semantic-sensitive integrated matching for picture libraries, IEEE Trans. Pattern Anal. Mach. Intell. 23 (2001) 947–963.

[6] I. Pratikakis, I. Vanhamel, H. Sahli, B. Gatos, S.J. Perantonis, Unsupervised watershed-driven region-based image retrieval, IEE Proc. Vision Images Signal Process. 153 (2006) 313–322.

[7] C. Carson, S. Belongie, H. Greenspan, J. Malik, Blobword: image segmentation using E-M and its application to image querying, IEEE Trans. Pattern Anal. Mach. Intell. 24 (2002) 1026–1038.

[8] J.-W. Hsieh, W.E.L. Grimson, Spatial template extraction for image retrieval by region matching, IEEE Trans. Image Process. 12 (2003) 1404–1415.

[9] J. Fan, Y. Gao, H. Luo, G. Xu, Statistical modeling and conceptualization of natural images, J. Pattern Recognition 38 (2005) 865–885.

[10] J. Luo, C.-E. Guo, Perceptual grouping of segmented regions in color images, J. Pattern Recognition 36 (2003) 2781–2792.

[11] S.-C. Zhu, Statistical modeling and conceptualization of visual patterns, IEEE Trans. Pattern Anal. Mach. Intell. 25 (2003) 691–712.

[12] W. Jiang, G. Er, Q. Dai, J. Gu, Similarity-based online feature selection in content-based image retrieval, IEEE Trans. Image Process. 15 (2006) 702–712.

[13] F. Jing, M. Li, H.-J. Zhang, B. Zhang, Relevance feedback in region-based image retrieval, IEEE Trans. Circuits Syst. Video Technol. 14 (2004) 672–681.

[14] D. Ballard, Generalizing the Hough transform to detect arbitrary shapes, J. Pattern Recognition 13 (1981) 111–122.

[15] S.-C. Cheng, C.-T. Kuo, H.-J. Chen, Visual object retrieval via block-based visual pattern matching, J. Pattern Recognition 40 (2007) 1695–1710.

[16] D.H. Ballard, C.M. Brown, Computer Vision, Prentice-Hall, Englewood Cliffs, NJ, 1982.

[17] C.P. Chau, W.C. Siu, Generalized Hough transform using regions with homogeneous color, Int. J. Comput. Vision 59 (2004) 183–199.

[18] S. Beucher, F. Meyer, The morphological approach to segmentation: the watershed transformation, in: E.R. Doughertty (Ed.), Mathematical Morphology in Image Processing, Marcel Dekker, New York, 1993.

[19] J.F. Canny, A computational approach to edge detection, IEEE Trans. Pattern Anal. Mach. Intell. PAMI-8 (1986) 679–698.

[20] H.-C. Lee, D.R. Cok, Detecting boundaries in a vector field, IEEE Trans. Signal Process. 39 (1991) 1181–1194.

[21] J. Matas, O. Chum, M. Urban, T. Pajdla, Robust wide-baseline stereo from maximally stable extremal regions, Image Vision Comput. 20 (2004) 761–767.

[22] S.P. Shan, X. Xuan, R.J. DeLeeuw, M. Khojasten, W.L. Lam, R. Ng, K.P. Murphy, Integrating copy number polymorphisms into array CGH analysis using a robust HMM, J. Bioinformatics 22 (2006) e431–e439.

[23] J.M. Morel, G. Yu, ASIFT: a new framework for fully affine invariant image comparison, SIAM J. Imag. Sci. 2 (2009) 438–469.

About the Author—CHI-HAN CHUANG received the B.S. degree from Chung Hua University, Hsinchu, Taiwan, in 2002, and M.S. degree in National Taiwan Ocean University, Keelung, Taiwan, in 2007, where he is currently pursuing his Ph.D. degree. His research interests are in image retrieval, computer vision, and intelligent information processing.

About the Author—SHYI-CHYI CHENG received the B.S. degree from National Tsing Hua University, Hsinchu, Taiwan, R.O.C., in 1986, and the M.S. and Ph.D. degrees in Electronics Engineering and Computer Science and Information Engineering from National ChiaoTung University, Hsinchu, in 1988 and 1992, respectively.

From 1992 to 1998, he was a Research Staff Member at Chunghwa Telecom Laboratories, Taoyuan, Taiwan. He is currently a Professor and Chairman of computer science and engineering, National Taiwan Ocean University, Keelung, Taiwan. His research interests include multimedia databases, image/video compression and communications, and artificial neural network applications.

About the Author—CHIN-CHUN CHANG received the B.S., M.S., and Ph.D. degrees in computer science from National Chiao Tung University, Hsinchu, Taiwan, R.O.C., in 1989, 1991, and 2000, respectively.

From 2001 to 2002, he was on the faculty at the Department of Computer Science and Engineering, Tatung University, Taipei, Taiwan. In 2002, he joined the Department of Computer Science, National Taiwan Ocean University, Keelung, where he is currently an Assistant Professor. His research interests include computer vision, machine learning, and pattern recognition.