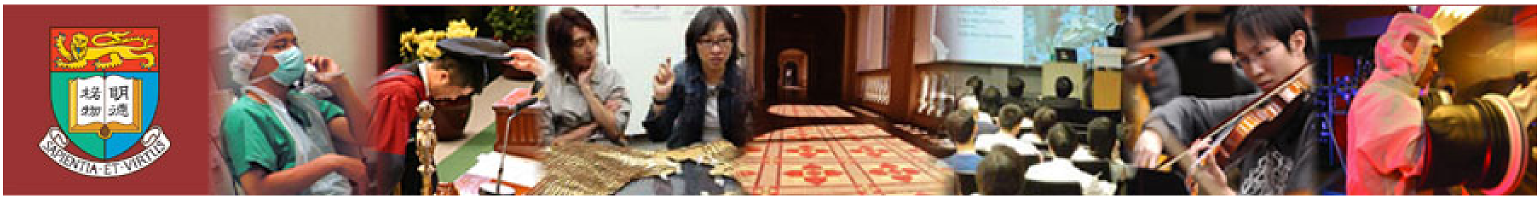


An Efficient Parameterless Quadrilateral-Based Image Segmentation Method

Ronald H.Y. Chung, *Member*, *IEEE*, Nelson H.C. Yung, *Senior Member*, *IEEE*, and Paul Y.S. Cheung, *Senior Member*, *IEEE*

Abstract—This paper proposes a general quadrilateral-based framework for image segmentation, in which quadrilaterals are first constructed from an edge map, where neighboring quadrilaterals with similar features of interest are then merged together to form regions. Under the proposed framework, the quadrilaterals enable the elimination of local variations and unnecessary details for merging from which each segmented region is accurately and completely described by a set of quadrilaterals. To illustrate the effectiveness of the proposed framework, we derived an efficient and high-performance parameterless quadrilateral-based segmentation algorithm from the framework. The proposed algorithm shows that the regions obtained under the framework are segmented into multiple levels of quadrilaterals that accurately represent the regions without severely over or undersegmenting them. When evaluated objectively and subjectively, the proposed algorithm performs better than three other segmentation techniques, namely, seeded region growing, K-means clustering and constrained gravitational clustering, and offers an efficient description of the segmented objects conducive to content-based applications.

Index Terms—Approximate methods, object representations, region growing, quadrilateral-based segmentation.

1 Introduction

The extraction of meaningful regions or objects from visual data has always been central to many content-based applications. This process, which is usually termed as image segmentation, is a fundamental step in most image analysis applications. In essence, it decomposes images, according to specific features of interest, into distinct regions to make high-level tasks such as object tracking, recognition, and scene interpretation possible. As the demand for object-based multimedia services continues to increase [1] and computer vision applications such as robotics, autonomous vehicles, and visual surveillance are in need of more sophisticated analysis techniques, the problem of image segmentation has received considerable attention in the literature [2].

Despite the fact that image segmentation has been intensively studied in the past and considerable research and progress have been made in segmenting objects, the robustness and generality of such algorithms have not been fully exploited [3]. Many algorithms are reasonably good at *analyzing* an image and partitioning it into groups of pixels that have similar or homogenous features. However, to support the aforementioned applications, these groups of pixels must be further interpreted before useful region (object) information, such as approximated boundaries, shape descriptors, etc., can be *synthesized*. It is observed that most content-based applications consider the *analysis step* and *synthesis step* as separate problems which are being optimized independently. For instance, the *analysis step* may be optimized so that a good trade-off is achieved between speed and the details of the image obtained for a particular region descriptor. Likewise, the *synthesis step* may be optimized so that a good trade-off is obtained between accuracy of the descriptors and speed. However, the outcome of these sequential steps does not necessarily imply a good trade-off between accuracy and speed as the *analysis step* might preserve too much details which are not required in the *synthesis step*. For this reason, it has been proposed that image segmentation should no longer remain at just the *analysis step*, but should be closely coupled with the *synthesis step* [4].

In view of this, we are motivated to propose a general image segmentation framework that offers an approach, which integrates the *analysis* and *synthesis steps*, for segmentation that not only extracts regions but also provides sufficient region information. The concept of such framework is built upon a network of quadrilaterals to represent regions. The idea is that quadrilaterals are first constructed from an edge map, where neighboring quadrilaterals with similar features of interest are grouped together to form regions. The grouping is performed through a number of decomposition levels to ensure the representation itself is accurate. This approach has two merits: First, quadrilateral representation offers an efficient data reduction similar to polygonal approximation techniques except that it is far more flexible with much lower approximation error. Second, interframe tracking of regions such as geometric invariant matching [5], [6], which correlates regions under different perspective projections, is possible by matching the quadrilaterals. In other words, tracking of dynamically changing objects is easily achievable using the quadrilateral-based approach as described in [7] and [8]. To illustrate the capability of the proposed framework, a parameterless segmentation algorithm is developed under the framework and compared both objectively and subjectively with quadrilateral-based segmentation (QBS) [7],

- *R.H.Y. Chung is with the Department of Computer Science, The University of Hong Kong, Room 425, Chow Yei Ching Bldg., Pokfulam Road, Hong Kong. E-mail: hychung@cs.hku.hk.*
- *N.H.C. Yung is with the Department of Electrical and Electronics Engineering, The University of Hong Kong, Room 503, Chow Yei Ching Bldg., Pokfulam Road, Hong Kong. E-mail: nyung@eee.hku.hk.*
- *P.Y.S. Cheung is with the Department of Electrical and Electronics Engineering, The University of Hong Kong, Room 601C, Chow Yei Ching Bldg., Pokfulam Road, Hong Kong. E-mail: cheung@eee.hku.hk.*

Manuscript received 13 Feb. 2004; revised 20 Sept. 2004; accepted 22 Nov. 2004; published online 14 July 2005.
Recommended for acceptance by G. Healey.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-0082-0204.

0162-8828/05/$20.00 © 2005 IEEE Published by the IEEE Computer Society

seeded region growing (SRG) [9], K-means clustering (KMC) [10], and constrained gravitational clustering (CGC) [11]. It is found that the proposed algorithm is robust in handling images of vastly different contents and is the best among all the other methods. Moreover, the segmented images generated by the proposed algorithm are potentially suitable for content-based applications, which further demonstrate the capability of the proposed segmentation framework.

This paper is organized as follows: Section 2 gives an overview of existing segmentation methods, Section 3 gives the generalized quadrilateral-based segmentation framework, Section 4 details the parameterless segmentation method developed under this framework, while Section 5 presents the evaluation results, and Section 6 concludes the whole paper.

2 Segmentation Overview

The methods for segmenting objects out of an image sequence can be broadly divided into two categories: Region-Based and Boundary-Based.

2.1 Region-Based Segmentation

Classical segmentation such as Seeded Region Growing (SRG) [9], K-means (KMC) [10], and Constrained Gravitational Clustering (CGC) [11] are region-based segmentation techniques. KMC and CGC explore the similarity between pixels by *clustering*. One or more features such as color, intensity, etc., are first evaluated at each pixel, where the n-dimensional features associated with each pixel are then treated as data points. For KMC, it determines a set of k points called centers so that each data point associates with one of them. The k centers are determined by minimizing the squared-error distortion from each associated data point. The pixels are then merged with its neighbor if their associated centers are the same and finally regions can be obtained. KMC usually produces reasonably good segmentation results for simple images and is particularly simple and efficient if k is small and known a priori. However, KMC may not perform well in the case of complex image in which determination of k is not trivial, in particular, when the image is highly textured with large number of colors.

The CGC proposed by Yung and Lai in [11], on the other hand, clusters data points by using the concept of gravitational clustering [12]. Data points are treated as masses and the gravitational forces among the masses are calculated. Each data point is allowed to move slowly under the influence of the resultant forces and when two points become close to each other, they are merged to form a cluster. They introduced a clustering constraint to limit the attractive forces between clusters so that the termination condition and the number of clusters can be easily defined. It was claimed that by considering the color in the RGB color space under their proposed clustering scheme, good segmentation can be achieved. CGC eases the determination of the number of clusters (corresponds to k value in KMC), however, it requires more computation power in clustering as simulation of the slow movements of data points is required. The merit of this approach is that the computational complexity of clustering is independent of the size of the image. But, the complexity will become high if the number of colors within an image is large.

Unlike KMC and CGC, SRG does not employ the *clustering* concept to explore the similarity among the pixels. Instead, it employs the concept of *Region Growing* which requires the specification of a set of pixels grouped into disjoint sets. Given these disjoint sets of pixels called seeds, SRG then finds a tessellation of the image into regions by successively including the pixels, that border at least one of the regions, into one of the regions such that a homogeneity measure is optimized. This process repeats until each pixel is associated with one region. In [9], only gray-scale images are considered and the homogeneity measure is the difference in gray values. SRG is simple and can be easily extended to color image. However, the performance of SRG depends heavily on the selection of seeds which is not always trivial in many applications.

Clustering and *Region Growing* are the underlying concepts that most of the effective region-based segmentation techniques employed. On the other hand, the research progress made in thresholding techniques [2], [13], [14] also boosts the development of a variety of homogeneity criteria [15], [16] in which *Region Growing* depends. As a result, a number of region-based segmentation techniques [14], [17], [18], [19], [20] have been proposed, based on different or a combination of clustering algorithms and region growing methods.

Region-based approach with appropriate choices of features of interest can generate reasonably good region pixel-wise boundaries in a subjective sense. The boundaries can be easily detected as the location at which two different regions meet, but with only those pixel-wise boundary points, further processing such as polygonal approximations [21], [22] B-spline approximations [23], [24], mesh-based approximations [25], [26], etc., are required before any high level region representation can be generated. Such further processing techniques, however, can be computationally intensive.

2.2 Boundary-Based Segmentation

Segmentation techniques that take the boundary-based approach primarily use edge information to locate region boundaries. Edge information refers to discontinuities in feature space, such as intensity, color, texture, etc. In this approach, boundary points are first located at the edge points, where these points are connected up to form closed curves that enclose regions. For example, an image segmentation technique has been proposed in [3] to segment an image based on texture edges. Texture edge flow vectors are located at the positions where discontinuity of spectral energies and phases occur. From the discontinuity, boundaries can be detected at the points where two opposite direction of flows encounter each other. However, disjoint, instead of closed, boundaries are usually obtained, which do not always imply well-defined regions. As such, a postprocessing technique is required for rectification. In [3], one boundary connection method has been proposed to associate a half elliptical neighborhood with its center located at the unconnected end of an open boundary, a smooth boundary segment is generated to connect another nearest boundary element that is within the half ellipse. The authors claimed that closed boundaries can be obtained by repeating this procedure 2-3 times. This segmentation approach is efficient if the edges in the image can be easily localized, but it may not be the case when most edge points are isolated which require large number of elliptical neighborhood searches for connecting those edge points.

Many boundary-based segmentation techniques [27], [28] use more or less the same approach. They generally differ from each other in terms of the usage of edge detection methods and boundary connection methods. Although edge detection and boundary connection methods are abundantly available, they are generally computational intensive if good segmentation is required. Besides, edge detection may be error-prone due to the sensitivity of operators. If edges have to be precisely determined with a thickness of single pixel (as in the case of Canny edge detector [29]), additional computation power has to be invested. On the other hand, if a less accurate edge detection method is used, then additional instruments such as edge thinning and gap filling (edge linking), which could be computationally intensive as well, might be required in the phase of boundary connection.

Some other boundary-based segmentation techniques, as opposed to the aforementioned ones, take the contour model [30], [31], [32], [33], [34], [35] approach in which it always guarantees the formation of closed boundary without the need of any boundary connection tools. In essence, these segmentation techniques start from an initial curve, which is then deformed and split successively according to some predefined models and rules. The final curves obtained will then hopefully coincide with the edges of the regions. In particular, the curve can be deformed or split to minimize the associated energy derived from both the edge and geometric information, as suggested in [31], [32], [33], [34], and [35]. However, the drawback of this energy minimization approach is its nonlinearity that results in inefficient implementations and instability [34]. As a result, many of these segmentation techniques require a proper initialization of the initial curve, which is usually chosen manually, in the vicinity of the expected boundaries in order to reduce the computation required in guiding the curve to the expected boundaries and to avoid being trapped in local minima. In [30], the points on the initial curve are regarded as independent cells in living tissue, strictly connected at the same time, which can reproduce, move and die according to their local environment (edge information). Under this model, the deformation of the curve follows a very simple set of rules, without the problems encountered in the energy minimization approach. However, the segmentation results depend heavily on the accuracy of the edge information.

By taking into consideration the merits of region-based and boundary-based segmentation, the segmentation framework proposed in this paper uses both the homogeneity and discontinuity measures for segmentation. There are some other works [36], [37] that use this hybrid approach too. But, our work differs from them in the sense that we focus not only on segmentation, but also the generation of region descriptors. In principle, our framework utilizes edge information for boundary approximation and similarity in features for region growing, so that approximated closed boundaries can always be obtained to serve as region descriptors after segmentation.

3 Generalized Quadrilateral-Based Segmentation Framework

The concept of the quadrilateral-based segmentation framework is to represent regions by a network of quadrilaterals. Essentially, each region obtained under this framework will be completely described by a set of quadrilaterals. Fig. 1 presents a block diagram of the framework. As depicted in Fig. 1, edge detection is first performed to obtain the edge map. For region-based segmentation, a simple edge detection method is sufficient whereas for boundary-based segmentation, a more sophisticated edge detection algorithm may be employed. As such, the choice of edge detection method depends on the needs of the segmentation method to bias toward homogeneity measure or discontinuity measure. The other components in the framework are discussed in the following sections.

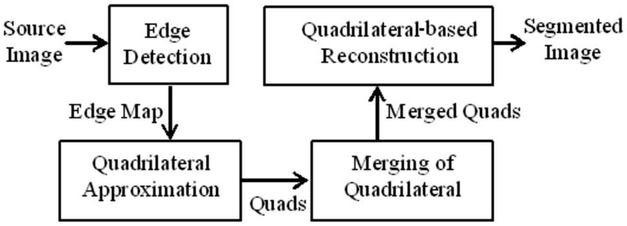


Fig. 1. Block diagram of segmentation framework.

3.1 Quadrilaterial Approximation

Assume that each region can be represented by a group of four-connected (left, right, top, and bottom) quadrilaterals. To approximate the regions by quadrilaterals, we first divide an image frame of size $M \times N$ into square blocks of width W, where $W = 2^r$, $M = 2^r m$, $\mathrm{N} = 2^r n$ for some positive integers m, n, and nonnegative integer r. Then, for each block, a *feature point* is derived based on the edge values within the block. The block may be further divided into four equal subblocks in case a single *feature point* cannot completely characterize the block.

Let $B(i,j,l)$ be a square block in row i and column j at level l in the image frame I, with block width 2^{r-l} (level 0 corresponds to the top most level with block width $W = 2^r$). We define $B(i,j,l)$ as follows:

$$\begin{aligned} B(i,j,l) &= \left\{ \begin{array}{ll} (x,y): & \frac{W}{2^l} i \le x < \frac{W}{2^l}(i+1), \\ & \frac{W}{2^l} j \le y < \frac{W}{2^l}(j+1), \\ & \text{where } x, y, \in Z^+ \cup 0 \end{array} \right\} \\ &= \left\{ \begin{array}{ll} (x,y): & 2^{r-l} \le x < 2^{r-l}(i+1) \\ & 2^{r-l} \le y < 2^{r-1}(j+1), \\ & \text{where } x, y, \in Z^+ \cup \{0\} \end{array} \right\} \\ & \quad \text{for } 0 \le i < 2^l m, 0 \le j < 2^l n. \end{aligned} \qquad (1)$$

We further denote the parent block of $B(i,j,l)$ as $P(B(i,j,l)) = B(\lfloor i/2 \rfloor, \lfloor j/2 \rfloor, l-1)$ for $l > 1$. If we let $P^0(B(i,j,l)) = B(i,j,l)$ and $P^1(B(i,j,l)) = P(B(i,j,l))$, we can obtain the hth predecessors of $B(i,j,l), Ph(B(i,j,l))$ as:

$$P^h(B(i,j,l)) = P(P^{h-1}(B(i,j,l))). \qquad (2)$$

After the *feature point* $FP(B(i,j,l))$ of each block $B(i,j,l)$ is determined, a network of quadrilaterals can be constructed by connecting the *feature points* of adjacent blocks. However, as adjacent blocks may not necessarily be in the same hierarchy, we introduce the terms *terminating block* and *intermediate block* to facilitate later discussion. A *terminating block* is defined as the block at which block subdivision terminates, whereas *intermediate block* is defined as the block that needs further subdivision. In other words, a *terminating block* is the block that has only one *feature point* whereas an *intermediate block* has more than one *feature point*. To precisely

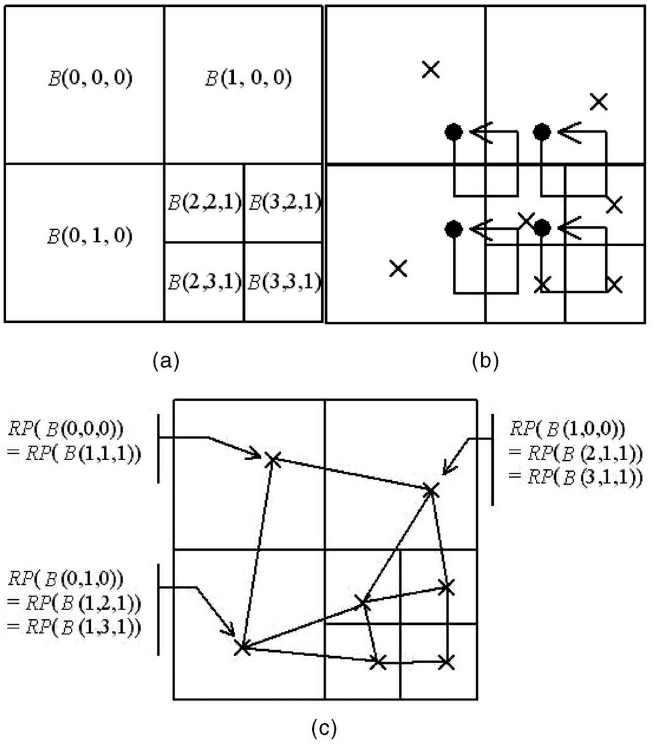

B(0, 0, 0)
B(1, 0, 0)
B(0, 1, 0)
B(2,2,1)
B(3,2,1)
B(2,3,1)
B(3,3,1)
(a)
(b)
RP(B(0,0,0)) = RP(B(1,1,1))
RP(B(1,0,0)) = RP(B (2,1,1)) = RP(B (3,1,1))
RP(B(0,1,0)) = RP(B (1,2,1)) = RP(B (1,3,1))
(c)

Fig. 2. (a) Block labels. (b) Four scan paths for $B(2,2,1)$. (c) Four quadrilaterals constructed.

describe the quadrilateral approximation procedure, the term *representative point* $RP(B(i,j,l))$ is introduced as follows:

$$RP(B(i,j,l)) = \begin{cases} FP(B(i,j,l)) & \text{if } B(i,j,l) \text{ is a } \textit{terminating block} \\ \phi & \text{if } B(i,j,l) \text{ is an } \textit{intermediate block} \\ FP(P^h(B(i,j,l))) & \text{otherwise} \end{cases}$$
$$\text{where } h = \min\{h' : P^{h'}(B(i,j,l)) \text{ is a } \textit{terminating} \text{ block}\}. \quad (3)$$

In essence, the *representative point* of a block is *its feature point* if the block is a *terminating block*. Note that, in the case of *intermediate block*, there is no *representative point*. In case the block is neither a *terminating block* nor an *intermediate block*, the *feature point* of the *terminating block* that encloses the block is used as the *representative point*. Quadrilaterals are constructed by interconnecting these *representative points*. We adopt the four connectivity scheme here, where each *representative point* only connects to the counterparts of its top, left, bottom, and right blocks. By doing so, a set of quadrilaterals, $SOQ(I)$, can be constructed for the given image frame I as follows:

$$SOQ(I) = \left\{ \begin{array}{l} (V_0, V_1, V_2, V_3): \\ V_0 = RP(B(i + h \bmod 2 - 1, j + \lfloor h/2 \rfloor - 1, l)), \\ V_1 = RP(B(i + h \bmod 2 - 1, j + \lfloor h/2 \rfloor, l)), \\ V_2 = RP(B(i + h \bmod 2, j + \lfloor h/2 \rfloor, l)), \\ V_3 = RP(B(i + h \bmod 2, j + \lfloor h/2 \rfloor - 1, l)), \\ h \in \{0,1,2,3\}, \quad V_0, V_1, V_2, V_3 \neq \phi, \\ B(i,j,l) \subset I \wedge B(i,j,l) \text{ is a } \textit{terminating block} \end{array} \right\}. \quad (4)$$

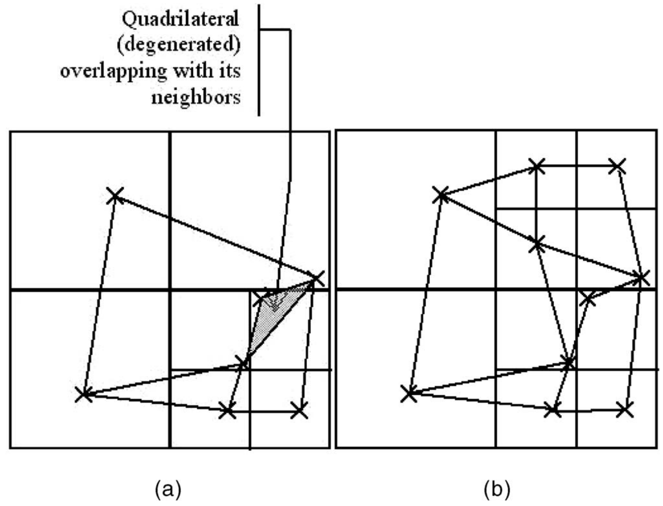


Fig. 3. (a) Overlapped quadrilaterials. (b) Quadrilaterals formed after subdivision of $B^0(1,0,0)$.

As given by (4), each quadrilateral is represented by a four-tuple (V_0, V_1, V_2, V_3) with V_0, V_1, V_2, V_3 representing the associated vertices. Considering a *terminating block*, four scan paths (indexed by h in (4)) are traversed to see if there is any quadrilateral that can be constructed. Fig. 2 shows an example of constructing four quadrilaterals starting from $B(2,2,1)$. It should be noted that, in some cases, a triangle is obtained instead of a quadrilateral. These triangles are treated as degenerated quadrilaterals with two of the vertices merged together.

The above scheme can be used to build a network of quadrilaterals that approximates an edge map. However, it is possible that the quadrilaterals approximated may overlap with their neighbors. As depicted in Fig. 3a, the quadrilateral formed by $B(2,2,1)$, $B(3,2,1)$, and $B(1,0,0)$ overlaps with the one formed by $B(2,2,1)$, $B(1,0,0)$, $B(0,0,0)$, and $B(0,1,0)$. This occurs when quadrilaterals are formed in an inconsistent orientation, i.e., when vertices of the quadrilateral and the corresponding blocks are not in the same orientation. Therefore, whenever a quadrilateral is not formed in a proper orientation, the block associated with that quadrilateral with the highest level of hierarchy is subdivided. The *feature point* of each subblock is recalculated and quadrilaterals associated with the subdivided block will be reconstructed. For example, in Fig. 3a, the block that requires further division is $B(1,0,0)$. Fig. 3b shows the quadrilaterals formed after $B(1,0,0)$ is subdivided.

3.2 Merging of Quadrilaterals

In this segmentation framework, regions are obtained from merging neighboring quadrilaterals with similar features of interest. However, we need to know whether two quadrilaterals are neighbors before merging can proceed. For each quadrilateral Q, there are four (three in the case of degenerated quadrilateral) possible direct neighbors as shown in Fig. 4.

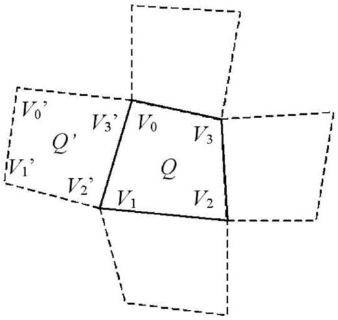


Fig. 4. Four possible direct neighbor quadrilaterals.

Suppose $Q = (V_0, V_1, V_2, V_3)$ and there is a left neighbor of $Q, Q' = (V_0', V_1', V_2', V_3')$. We can deduce that $V_0 = V_3'$ and $V_1 = V_2'$ and the common edge of Q and Q' is (V_0, V_1). If four possible neighbor positions are considered, we can obtain the common edge of Q and Q' as follows:

$$CME(Q, Q') = \begin{cases} (V_0, V_1) & \text{if } V_0 = V_3' \wedge V_1 = V_2' \\ (V_0, V_3) & \text{if } V_0 = V_1' \wedge V_3 = V_2' \\ (V_2, V_3) & \text{if } V_2 = V_1' \wedge V_3 = V_0' \\ (V_1, V_2) & \text{if } V_1 = V_0' \wedge V_2 = V_3' \\ \phi & \text{otherwise.} \end{cases} \tag{5}$$

With that, Q and Q' are neighbors if and only if $CME(Q, Q') \neq \phi$. We can then further define the set of neighbors for a quadrilateral Q for frame I as:

$$NB(Q) = \{Q' \in SOQ(I) : CME(Q, Q') \neq \phi\}. \tag{6}$$

Note that different features of interest can be applied in quadrilateral merging and the proposed framework does not restrict the merging order, which could affect the segmentation results [15].

3.3 Quadrilateral-Based Reconstruction

After the merging, regions defined by a set of quadrilaterals with similar features can be obtained and the final segmented image can be constructed. In some applications, region boundaries are required to be extracted instead. In that case, we can obtain the boundary of a region R easily as a set of edges $\{E : E = CME(Q, Q'), Q \in R, Q' \notin R \text{ for some } Q', E \neq \phi\}$.The segmented image can then be obtained by drawing all the line segments in the set of edges. The ease of boundary extraction is one of the merits of this segmentation framework as no postprocessing is required. This also demonstrates the effectiveness of having the *analysis step* tightly coupled to the *synthesis step*.

3.4 Close Inspection of the Generalized Segmentation Framework

Upon close inspection of the framework, it is observed that region-based segmentation techniques such as KMC, CGC, SRG, and some other boundary-based segmentation techniques are indeed special cases of the proposed framework. To show that, let us consider the degenerated case in the proposed framework with the block width $W = 1$ at level $l = 0$. Then, each block $B(i, j, 0)$ in frame I contains a single pixel (i, j) and the *feature point* is again (i, j). Further, the set of quadrilaterals is:

$$SOQ(I) = \left\{ \begin{array}{l} (V_0, V_1, V_2, V_3) : \\ V_0 = (i, j), \\ V_1 = (i, j+1), \\ V_2 = (i+1, j+1), \\ V_3 = (i+1, j), \\ (i, j) \subset I \end{array} \right\}. \tag{7}$$

For KMC and CGC, they first evaluate the features of interest at each pixel location, and then assign cluster label to each pixel by clustering. The corresponding merging criterion for any two quadrilaterals $Q = (V_0, V_1, V_2, V_3), Q' \in (V_0', V_1', V_2', V_3')$ under our framework is to test whether the cluster label of V_0 equals to that of V_0'. If the test is positive, they are merged together. For SRG, let T be the set of all the unallocated pixels (pixels not belonging to any region) as defined in [9], and R_i be region i, T can be formulated as follows:

$$\begin{aligned} T = \Bigg\{ V_0 \Bigg| \Bigg(Q = (V_0, V_1, V_2, V_3) \in \bigcup_{i=1}^{n} R_i \Bigg) \\ \wedge NB(Q) \cap \bigcup_{i=1}^{n} R_i \neq \phi \Bigg\}, \end{aligned} \tag{8}$$

where the corresponding homogeneity measure and the merging algorithm in [9] can be applied directly.

For boundary-based segmentation techniques, emphasis is placed on edge detection to obtain a well-defined edge map with single-pixel thickness which forms closed contours. Using the set of quadrilaterals obtained from (7), for any two quadrilaterals $Q = (V_0, V_1, V_2, V_3), Q' \in (V_0', V_1', V_2', V_3')$, they are merged if both V_0 and V_0' are not edge points. Actually, edge points are the locations at which discontinuities occur or the locations at which pixels not similar to their neighbors are detected. As a result, the merging condition can be well transformed to "Merge if both V_0 and V_0' are similar to each other." From the above, we observe that the aforementioned segmentation methods can be completely characterized by different specifications of edge detection methods, *feature point* definition for quadrilateral approximation, features of interest for merging, merging criteria and merging algorithms under the proposed segmentation framework. However, these segmentation methods can only generate pixel-wise region boundaries, as they have merely focused primarily on the *analysis step*.

4 Parameterless Quadrilateral-Based Segmentation Algorithm

Based on the framework described in Section 3, we developed a parameterless segmentation algorithm which can also generate approximated region boundaries, through the following specifications of edge detection method, *feature point* definition for quadrilateral approximation, features of interest for merging, merging criteria, and the merging algorithm.

4.1 Edge Detection

There are plenty of edge detection methods that can be used for this purpose. Both interframe and intraframe edge detections may be used. In order to demonstrate the effectiveness of the quadrilateral-based segmentation framework and to note the effect of the merging criteria and the parameterless segmentation algorithm, we choose a simple edge detector, Sobel.

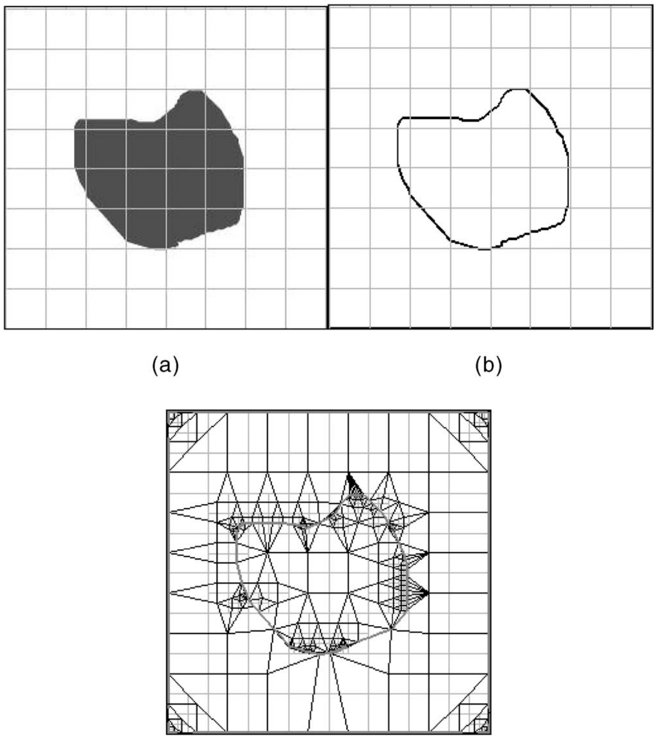
4.2 Feature Point Definition for Quadrilateral Approximation

Let $G(x,y)$ represents the nonthresholded magnitude of the Sobel edge detector output at pixel (x,y), then we choose the *feature point* in a block $B(i,j,l)$ to be the center of mass as follows:

$$FP(B(i,j,l)) = \left(\frac{\sum\limits_{(x,y)\in B(i,j,l)} x \times G(x,y)}{\sum\limits_{(x,y)\in B(i,j,l)} G(x,y)}, \frac{\sum\limits_{(x,y)\in B(i,j,l)} y \times G(x,y)}{\sum\limits_{(x,y)\in B(i,j,l)} G(x,y)} \right). \quad (9)$$

The reason for choosing center of mass is that, if the edge points in a block approximate a straight line, the center of mass is usually sufficiently close to the line. However, that might not be the case when multiple edge lines or corner points exist in a block. In that case, the corresponding block should then be divided into four equal subblocks as discussed in Section 3.1. This process of subblock division and recalculation of center of mass is repeated until the centers of mass obtained are sufficiently close to the edge lines or no further subdivision is possible (zero width and height). To do this, we must know whether a center of mass is able to approximate the edge feature of the block. It is therefore necessary to establish a criterion to test whether the edge points in a block approximate a straight line. To do this, we correlate the x-coordinates with the y-coordinates of the edge points within a block. Let X and Y be random variables representing the x and y-coordinates of the edge points in $B(i,j,l)$, respectively. Let $\mathrm{var}_X = E(X^2) - E(X)^2, \mathrm{var}_Y = E(Y^2) - E(Y)^2$, respectively, denote the variance of X and Y; $\mathrm{var}_{XY} = E(XY) - E(X)E(Y)$ denotes the covariance of X and Y. Then, the correlation of edge points, $Corr(B(i,j,l)) = \mathrm{var}_{XY}/[(\mathrm{var}_X.\mathrm{var}_Y)]^{1/2}$ can be obtained as:

$$\begin{aligned} &Corr(B(i,j,l)) \\ &= \left[\frac{\sum\limits_{(x,y)\in B(i,j,l)} x\times y\times G(x,y)}{\sum\limits_{(x,y)\in B(i,j,l)} G(x,y)} - \frac{\sum\limits_{(x,y)\in B(i,j,l)} x\times G(x,y) \sum\limits_{(x,y)\in B(i,j,l)} y\times G(x,y)}{\left(\sum\limits_{(x,y)\in B(i,j,l)} G(x,y)\right)^2} \right] \\ &\times \left[\frac{\sum\limits_{(x,y)\in B(i,j,l)} x^2\times G(x,y)}{\sum\limits_{(x,y)\in B(i,j,l)} G(x,y)} - \left(\frac{\sum\limits_{(x,y)\in B(i,j,l)} x\times G(x,y)}{\sum\limits_{(x,y)\in B(i,j,l)} G(x,y)} \right)^2 \right]^{-1/2} \quad (10) \\ &\times \left[\frac{\sum\limits_{(x,y)\in B(i,j,l)} y^2\times G(x,y)}{\sum\limits_{(x,y)\in B(i,j,l)} G(x,y)} - \left(\frac{\sum\limits_{(x,y)\in B(i,j,l)} y\times G(x,y)}{\sum\limits_{(x,y)\in B(i,j,l)} G(x,y)} \right)^2 \right]^{-1/2}. \end{aligned}$$


(a) (b)

(c)

Fig. 5. (a) Original image frame divided into square blocks. (b) Edge map overlapped with divided image frame. (c) Quadrilaterals approximated for the image frame.

From (10), the block subdivision rule can be ideally specified as follows: If $|Corr(B(i,j,l))|$ is 1, then the edge points lie on a straight line, otherwise subdivide $B(i,j,l)$.

However, in reality, $|Corr(B(i,j,l))|$ can hardly equal to 1 unless the edge within the block is a perfect straight line without being tampered by noises. We found that, through the test images in this paper and those in [7], $0.9 \leq |Corr(B(i,j,l))| \leq 1$ is a good approximation to the ideal block subdivision rule. As a result, we used the following revised block subdivision rule in our actual implementation: If $0.9 \leq |Corr(B(i,j,l))| \leq 1$, then the edge points lie on a straight line, otherwise subdivide $B(i,j,l)$.

With this block subdivision rule, each block in the image will have its *representative point* defined according to (3). Quadrilateral approximations can then proceed to end up with a set of quadrilaterals as described in (4). Fig. 5 demonstrates the use of this scheme to approximate an object in an image frame. Fig. 5a shows the original image frame divided into square blocks, Fig. 5b shows the edge map overlapped with the divided image frame whereas Fig. 5c shows the extracted quadrilaterals with quadrilaterals drawn in solid lines whereas edge outline drawn in a think solid gray line. From this, we can see that the quadrilateral approximation actually moves the vertices of each quadrilateral to the nearby edge points, which are usually the boundary points of segmented regions. By doing so, the problem of segmentation reduces to merging of quadrilaterals, instead of merging of pixels, which leads to a data size reduction in the merging process.

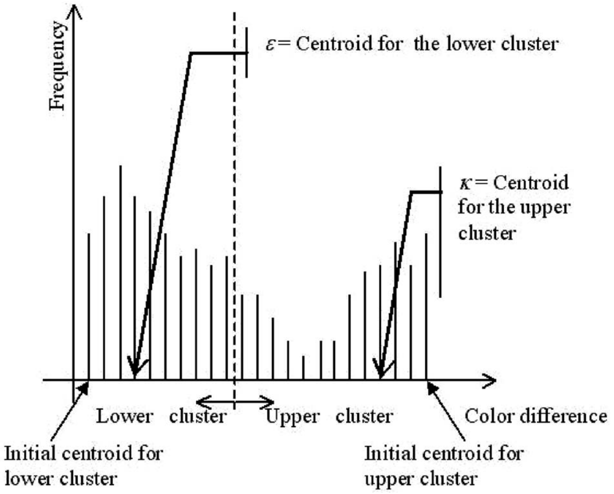


Fig. 6. Determination of ε and κ.

4.3 Features of Interest for Merging

We use color as the features of interest in here, although different features can also be used. In particular, we perform segmentation in the RGB color space, as it is found that RGB produces least noisy segmentation with reasonable number of regions in most cases [38].

4.4 Merging Criteria

For each quadrilateral Q, we first evaluate the mean of each color component of the pixels within Q. Let $M_R(Q), M_G(Q), M_B(Q)$ denote the mean of the Red, Green, and Blue color component of the pixels within Q, respectively. The color difference between two quadrilaterals Q and Q' is defined as:

$$D_C(Q,Q') = (M_R(Q) - M_R(Q'))^2 + (M_G(Q) - M_G(Q'))^2 + (M_B(Q) - M_B(Q'))^2. \tag{11}$$

To determine whether two quadrilaterals should be merged, we check whether the color difference is below a threshold ε, and whether the two quadrilaterals are direct neighbors. That is, if $D_C(Q,Q') < \varepsilon$ and $CME(Q,Q') \neq \phi$, then we merge the quadrilaterals Q and Q' together. Different from our previously proposed method in [7], we introduce a further threshold κ, in addition to ε, such that, if $DC(Q,Q') > \kappa$ and $CME(Q,Q') \neq \phi$, then the two quadrilaterals involved are not merged. The threshold κ relaxes the condition for assigning new region label, which will be described in the next section.

In this paper, thresholds ε and κ are determined algorithmically from the probability density function (histogram) of the color differences. First, we obtain the color difference between each pair of neighbor quadrilaterals, from which we can construct a histogram of the color differences. We then partition the obtained color differences into two clusters by the K-means algorithm. The initial centroids for the two clusters are the minimum and maximum color differences obtained. From the resulting clusters, we take the centroid in the lower cluster as the threshold ε and centroid in the upper cluster as κ, as depicted in Fig. 6.

4.5 Merging Algorithm

We first define three terms: *labeled quadrilateral*, *new-able quadrilateral*, and *merge-able quadrilateral*. A *labeled quadrilateral* is a quadrilateral that has been assigned to an existing region, a *new-able quadrilateral* is the one where a new region label can be assigned whereas a *merge-able quadrilateral* is the one that can be merged into one of the existing regions. The sets of *labeled quadrilaterals* $LQ(I)$, *new-able quadrilaterals* $NQ(I)$, and *merge-able quadrilaterals* $MQ(I)$ for the frame I are defined below, respectively:

$$LQ(I) = \{Q \in SOQ(I) : \exists \text{a region } R \text{ s.t. } Q \in R\}, \tag{12}$$

$$\begin{aligned} NQ(I) = \big\{Q \in SOQ(I) : Q \notin LQ(I) \wedge \\ (\forall Q' \in NB(Q)) \\ (Q' \notin LQ(I) \Longrightarrow D_C(Q,Q') < \varepsilon) \wedge \\ (Q' \in LQ(I) \Longrightarrow D_C(Q,Q') > \kappa))\big\}, \end{aligned} \tag{13}$$

$$\begin{aligned} MQ(I) = \big\{Q \in SOQ(I) : \exists Q' \in NB(Q) \text{ s.t. } Q' \in LQ(I) \\ \wedge D_C(Q,Q') < \varepsilon, \quad Q \notin LQ(I)\big\}. \end{aligned} \tag{14}$$

From (13), a new region label is assigned to a quadrilateral when it can be merged with its non*labeled* direct neighbors and cannot be merged with its labeled direct neighbors. This is different from the region labeling scheme in [7], which always assign new region label to the quadrilateral only when it can be merged with its neighbors. The consequence of this relaxation, through the introduction of threshold κ, is that the birth of a new region next to a labeled region is possible, whereas the previous approach fail to capture these regions appropriately. The merging process starts by including a *new-able quadrilateral* Q into region R_0, such that $R_0 = \{Q\}$ and

$$Q = \arg\min_{Q'} \left\{ \max \left\{ \begin{array}{l} D_c(Q',Q'') : Q' \in NQ(I) \\ \wedge Q'' \in NB(Q') \end{array} \right\} \right\}. \tag{15}$$

The quadrilateral described by (15) is a *new-able* quadrilateral, of which its maximum associated color difference is the minimum among all the *new-able* quadrilaterals. After that, R_0 grows by successively including all the quadrilaterals in $MQ(I)$, i.e., $R_0 = R_0 \cup MQ(I)$. Note that in each iteration, $LQ(I)$, $NQ(I)$, and $MQ(I)$ change accordingly. After a number of steps, $MQ(I)$ becomes empty. This implies that 1) R_0 may be enclosed by some quadrilaterals that should belong to some other regions and 2) the threshold ε is not large enough for R_0 to grow. Therefore, for merging to proceed, we either assign a new region label to a *new-able* quadrilateral or increase ε. Unlike the one presented in [7] in which we always increase ε by a fixed value, we adopt the following testing condition for the decision:

$$\min\{\varepsilon : MQ(I) \neq \phi\} < \min\{\varepsilon : NQ(I) \neq \phi\}. \tag{16}$$

Equation (16) compares the minimum threshold that triggers a merging event with that which triggers the birth of a new region. The condition is such that if (16) holds, then ε is updated to $\min\{\varepsilon : MQ(I) \neq \phi\}$ and the merging continues, otherwise ε is updated to $\min\{\varepsilon : NQ(I) \neq \phi\}$ and a new region R_1 is then formed with a *new-able* quadrilateral determined by (15). By doing so, we give preference to merging rather than assigning new region label. This resolves the oversegmentation problem sometimes encountered by our previous method [7]. By repeating this scheme, non*labeled*

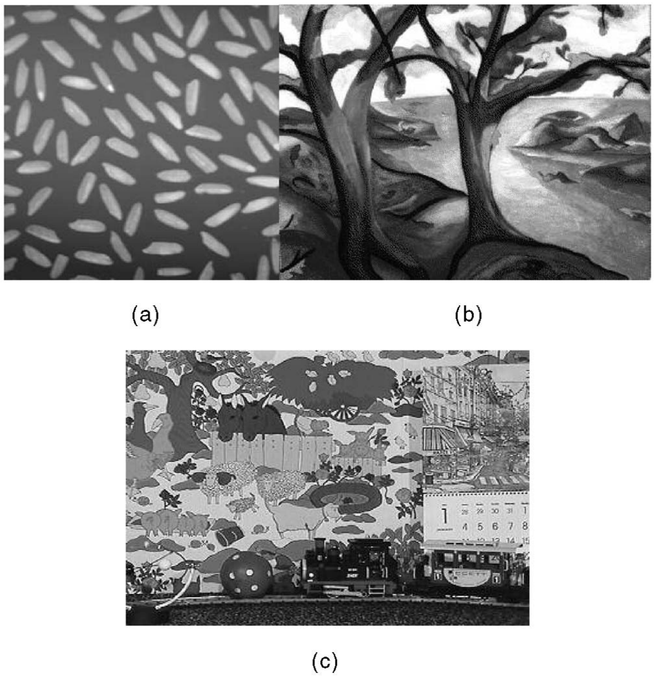


Fig. 7. (a) “Rice” image. (b) “Trees” image. (c) “Mobile and Calendar” image.

quadrilaterals are either merged to existing regions, or included in new regions. The following algorithmic steps summarize the merging process:

Step 1: Set the number of region N_r to 1, pick a quadrilateral Q by (15) and make $R_0 = \{Q\}$.

Step 2: Update $LQ(I)$, $NQ(I)$, and $MQ(I)$ accordingly.

Step 3: If $Q \in LQ(I)$ for all $Q \in SOQ(I)$, go to *Step 7*.

Step 4: If $MQ(I) \neq \phi$, for each region R_i, where $0 \leq i < N_r$, set $R_i = R_i \cup \{Q : Q \in MQ(I) \wedge (\exists Q' \in NB(Q) \text{ s.t. } Q' \in R_i)\}$, and go back to *Step 2* afterwards. Otherwise proceed to *Step 5*.

Step 5: If (16) holds, set ε to $\min\{\epsilon : MQ(I) \neq \phi\}$ and Go back to *Step 2*. Otherwise, set ε to $\min\{\varepsilon : NQ(I) \neq \phi\}$, update $NQ(I)$ accordingly and proceed to *Step 6*.

Step 6: Pick a quadrilateral Q by (15) and make $R_{N_r} = \{Q\}$, increase N_r by 1 and go back to *Step 2*.

Step 7: Stop merging process.

5 Evaluations

To evaluate how well the proposed method performs, we compare its performance with our previous method, QBS [7], and three other methods, SRG [9], KMC [10], and CGC [11]. SRG, KMC, and CGC are chosen because they belong to the special case under our segmentation framework and in particular SRG and KMC are widely used in many applications. Both objective measurement and subjective inspection were considered. For the objective measurement, there are many different evaluation methods and they have been extensively studied by Zhang [39]. These available evaluation methods, however, require some scaling/weighting parameters which often have to be set on the basis of human intuition or judgment, or require a reference image which may not always be available. The one proposed by Liu and Yang [38] is adopted in our work because it is a parameter-free objective evaluation method without requiring a reference image. However, it should be noted that this method gives merely a broad and general indication, where subjective inspection should not be undervalued. The evaluation function L is defined as:

$$L(I) = \frac{\sqrt{N_r}}{1000(MN)} \sum_{i=0}^{N_r-1} \frac{e_i^2}{\sqrt{A_i}}, \tag{17}$$

where I is the image frame, N_r is the number of segmented regions in I, A_i is the number of pixels in the region R_i, M and N are the width and height of I, and e_i^2 is the color error of region R_i, which is defined as the sum of Euclidean distance of the color vector between I and the corresponding segmented image for each pixel in the region. According to [38], a smaller L means better performance.

TABLE 1
Smallest L Obtained and the Corresponding Number of Regions (R) Obtained from Each Method

	KMC	CGC	SRG	QBS	Proposed
“Rice”					
L	0.291	0.264	0.229	0.197	0.178
R	85	151	174	86	74
Rank	5	4	3	2	1
“Trees”					
L	8.651	2.017	1.528	1.164	0.873
R	1422	3672	435	242	126
Rank	5	4	3	2	1
“Mobile and Calendar”					
L	16.071	7.819	4.305	3.860	3.741
R	1834	5240	581	304	302
Rank	5	4	3	2	1

The subjective inspection is based on three criteria: smoothness of the boundaries, boundary correctness, and the reasonable number of resultant regions which closely reflects the number of noticeable regions in the original image. Three test images were used in the evaluation, namely, “Rice,” “Trees,” and “Mobile and Calendar,” as shown in Fig. 7. These three images are chosen because each of them presents different level of difficulties in image segmentation. “Rice” is a gray-scale image consisting of grains of rice on a dark gray background. The contrast of this image decreases from the top to the bottom, which presents some challenges in threshold determination. “Trees” is a color image consisting of two trees located on a riverside with the branches of the trees overlapped with each other. The contrast on the roots is low and the boundaries of the leaves are a little bit fuzzy. When compared with “Rice,” it is a more complex image with higher degree of segmentation difficulty. Finally, “Mobile and Calendar” is a color image consisting of complex objects composed from numerous tiny regions, and the focus of the image is placed on the wallpaper, making the locomotive slightly out of focus. This image is considered to be the most difficult of the three.

For each method, its parameters were varied over a range where their segmented results and L values for the three images were determined. Table 1 shows the best objective performance (L), whereas Figs. 8, 9, and 10 depict the corresponding segmented images with best L. Although the proposed algorithm determines its parameter algorithmically, we first evaluate the best performance of the proposed

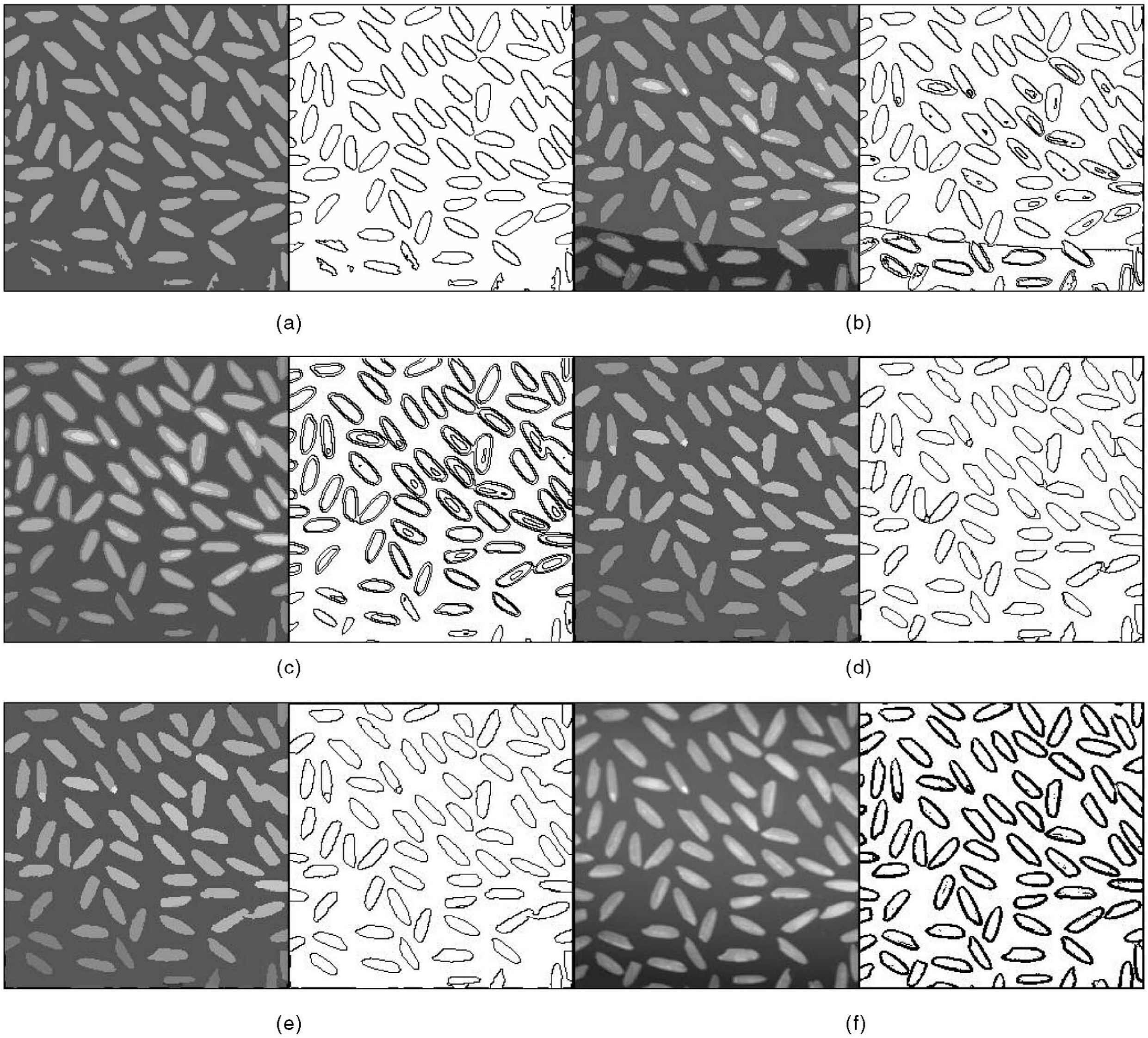


Fig. 8. Segmented images and region maps of "Rice" according to smallest L. (a) KMC. (b) CGC. (c) SRG. (d) QBS. (e) Proposed. (f) Original.

method by adjusting ε manually. Performance of the proposed algorithm for determined algorithmically is given at the end of this section. As depicted in Table 1, it seems that the higher the complexity of an image, the higher is the L value. If we rank the methods with respective L values in ascending order, we can see that the rank of each method is consistent across the images. The proposed algorithm ranks first, QBS follows closely behind, SRG ranks third, CGC ranks fourth, whereas KMC always gives the largest L. In this respect, the proposed algorithm has the best performance among all the others. When the number of regions is concerned, the proposed algorithm also gives the smallest number of regions for all the test images. This shows that our algorithm is more conducive for content-based applications, as smaller number of regions means less computation power is required in correlating regions in different frames. However, a smaller number of regions might not necessarily imply better segmentation results as it might indicate the possibility of overmerging. Therefore, it is necessary to compare the segmented images subjectively.

Figs. 8, 9, and 10 depict the segmented images for each method resulted from the best L. In Fig. 8, we can see that although the brightness of the background of the "Rice" image varies from the top to bottom, each method can segment the background correctly as a whole, except for the CGC, which treats the bottom part of the background as a different region.

For KMC, the rice grains are segmented correctly in the upper part of the image, and generally have smooth boundaries. But, when it comes to the lower part of the image, where the contrast is poorer, some of the rice grains are segmented with incorrect and rugged boundaries. CGC suffers from the same problem as KMC, but the boundaries of the rice grains in the lower part are smoother than KMC. SRG, on the other hand, generates double region edges for most of the rice grains. Although the rice grains in the lower part are segmented with smooth boundaries, they are not correctly segmented when we compare the size of the rice grains with the original. QBS segments the rice grains and background correctly, with reasonably smooth boundaries, but tends to oversegment as shown in the left and near the top-right region of the image. The proposed algorithm does not seem to have similar oversegmented problem, as most of the rice grains are segmented as a whole with smooth boundaries.

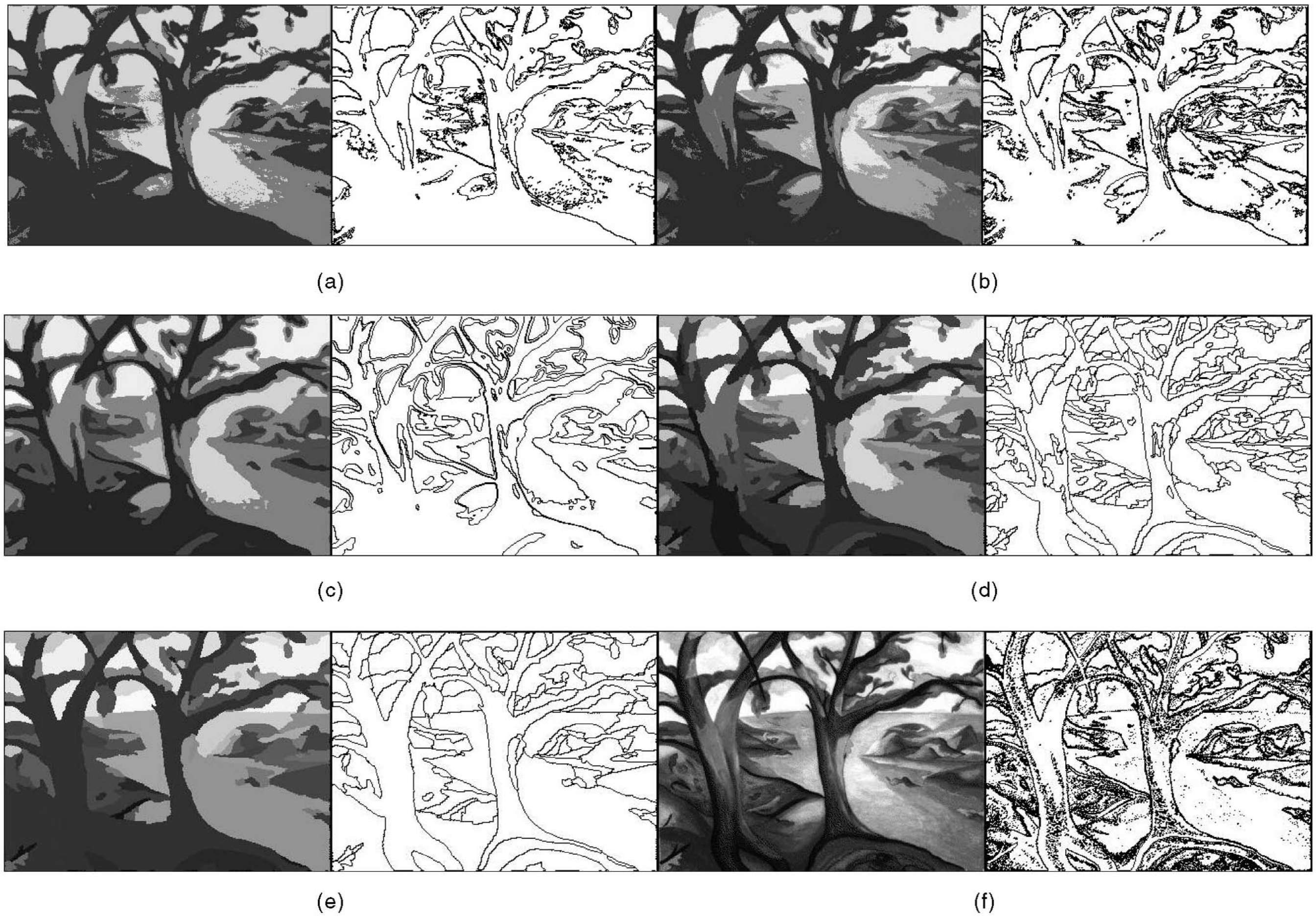


Fig. 9. Segmented images and region maps of "Trees" according to smallest L. (a) KMC. (b) CGC. (c) SRG. (d) QBS. (e) Proposed. (f) Original.

The only problem might be the two rice grains being merged together near to the right edge of the image and with some partial rice grains missing near the edges of the image.

Fig. 9 shows the segmented images of "Trees." KMC, CGC, and SRG generate smooth boundaries, but all have the undersegmentation problem. It can be easily spotted at the lower part of the trees, where they are merged with the riverbank. KMC and CGC even have the branches of the trees merged with the leaves. Besides, KMC and CGC also generate many tiny regions, especially in the river and along the riverbank. On the other hand, SRG encounters the same problem in "Rice" and double region edges can be identified in the boundaries of the trunk of the trees. In contrast, QBS has the roots of the trees correctly segmented, with little problem of undersegmentation. The proposed algorithm is superior to QBS in the way that the number of regions looks more reasonable, while the boundaries of the trees are well preserved.

Fig. 10 shows the segmented images of "Mobile and Calendar," from which we observe that KMC and CGC suffer from severe undersegmentation problem noticeable from the merged regions on the wallpaper. Furthermore, the ball on the track is also merged with the train. SRG suffers from undersegmentation too, but less severely when compared with KMC and CGC, as it correctly segments the ball and train. The boundaries obtained by these three methods are not accurate, but they are smooth. On the other hand, QBS segments noticeable regions from the wallpaper, and the ball and train are correctly segmented. The proposed algorithm has similar performance in terms of visual quality as QBS. Upon closer inspection, it is noted that the proposed method has fewer small regions. Although the boundaries obtained by the proposed algorithm and QBS appear less smooth, it is to be expected as these boundaries are approximated by the boundaries of quadrilaterals instead of pixels as in the case of KMC, CGC, and SRG.

The subjective evaluation is summarized in Table 2, where complexity refers to the number of meaningful regions formed. Note that a segmented image with low complexity is more favorable for content-based applications. To consider the performance of the proposed parameterless algorithm, we determine the value of ε by the method described in Section 4.4 for each image. Then, we segment the images with the corresponding and compare it with the optimal performance. The segmented images obtained look very similar to the best segmented images shown in Figs. 8e, 9e, and 10e. Table 3 shows the L values, the number of regions R obtained for each image and their percentage change with respect to its optimal performance. It is found that the performance of the parameterless method has a graceful degradation, with the slight increase in L and R. In particular, the percentage increase in L ranges from 1.37 percent to 6.84 percent and that in R ranges from 1.59 percent to 6.76 percent.

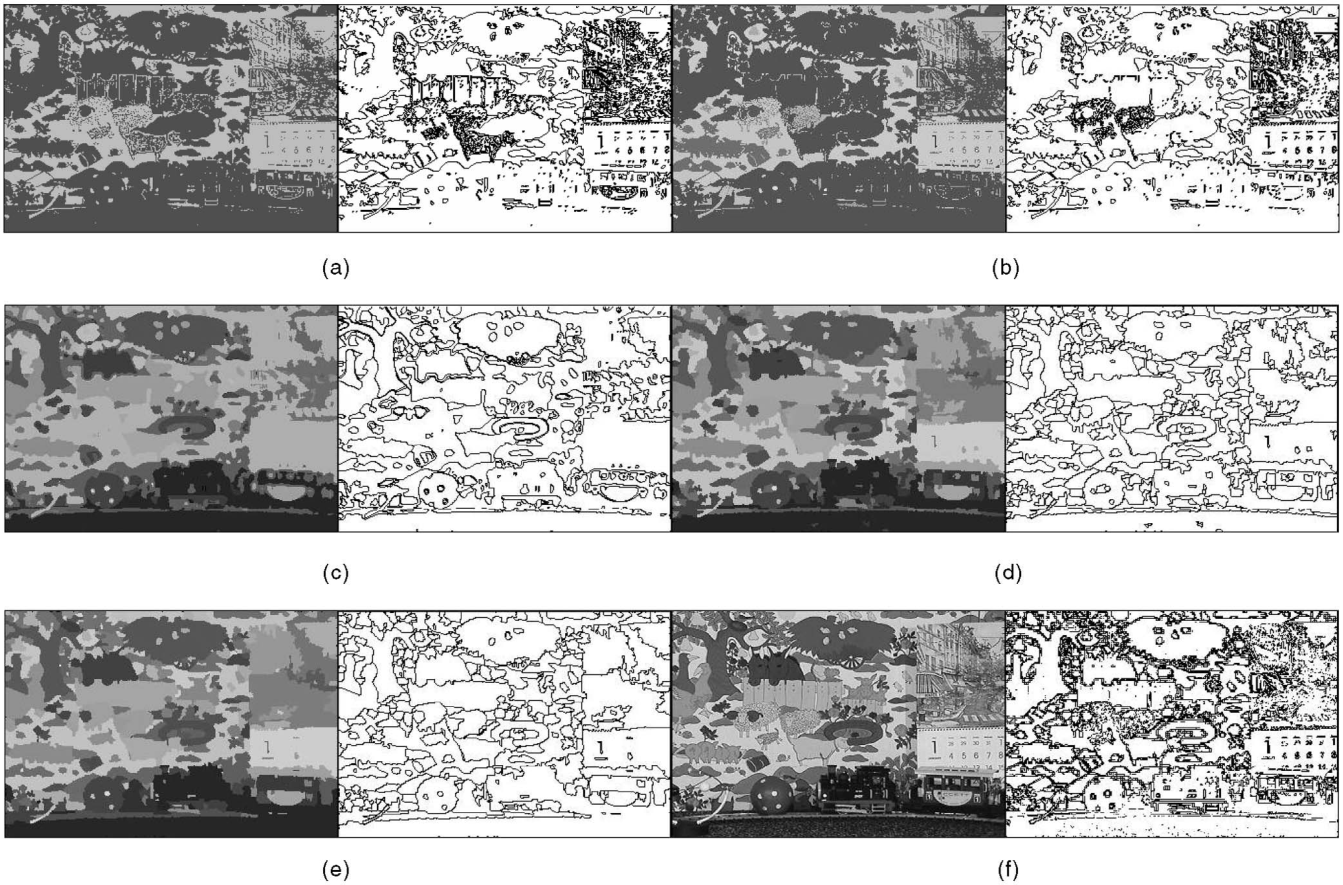


Fig. 10. Segmented images and region maps of "Mobile and Calendar" according to smallest L. (a) KMC. (b) CGC. (c) SRG. (d) QBS. (e) Proposed. (f) Original.

6 CONCLUSIONS

We have presented a new parameterless quadrilateral-based segmentation algorithm, based on a generalized quadrilateral-based segmentation framework. The proposed algorithm was evaluated and compared both objectively and subjectively against four other methods, namely, KMC, SRG, CGC, and QBS. We observed that the proposed algorithm performs best in terms of both objective and subjective measures. The regions obtained by the proposed algorithm are approximated by quadrilaterals, and boundaries are extracted effortlessly without the need of further interpretation or postprocessing. Besides, it also generates a reasonable number of regions with low complexity. By these observations, it can be deduced that the proposed algorithm is more suitable for applications such as tracking and content-based video coding. As described in this paper, a segmentation algorithm, under the generalized quadrilateral-based framework, can be completely characterized by the specification of five items:

1. Edge detection method,
2. Feature point definition,
3. Features of interest,
4. Merging criteria, and
5. Merging algorithm.

TABLE 2
Subjective Evaluation of Each Method for Different Images

	KMC	CGC	SRG	QBS	Proposed
"Rice"					
Smoothness	Fair	Good	Good	Good	Good
Correctness	Fair	Fair	Fair	Fair	Good
Complexity	Low	Moderate	Moderate	Low	Low
"Trees"					
Smoothness	Good	Good	Good	Fair	Good
Correctness	Poor	Poor	Fair	Fair	Good
Complexity	High	High	Moderate	Moderate	Low
"Mobile and Calendar"					
Smoothness	Good	Good	Good	Fair	Fair
Correctness	Poor	Poor	Fair	Fair	Fair
Complexity	High	High	Moderate	Low	Low

TABLE 3
L and R Values Obtained with ε Determined Algorithmically

	Parameter-less	Percentage Change
"Rice"		
L	0.190	6.74%
R	79	6.76%
"Trees"		
L	0.885	1.37%
R	128	1.59%
"Mobile and Calendar"		
L	3.997	6.84%
R	320	5.96%

From the proposed algorithm, we have demonstrated that this framework enables us to develop better segmentation algorithms by modifying the merging criteria and merging algorithm. In this case, a parameterless algorithm proves to be more efficient and accurate. Future effort will be devoted to five areas:

1. to study the effect of different edge detection methods on segmentation result;
2. to refine feature point definition for better quadrilateral approximation;
3. to investigate the impact of different features of interest;
4. to consider different merging criteria; and
5. to quantify the effect of merging algorithm, especially the rules that govern the merging order and birth of regions, on segmentation.

Acknowledgments

The authors would like to thank the reviewers for their valuable and constructive comments for improving this paper.

References

[1] P. Salembier and F. Marques, “Region-Based Representations of Image and Video: Segmentation Tools for Multimedia Services,” *IEEE Trans. Circuits and Systems for Video Technology,* vol. 9, no. 8, pp. 1147-1169, Dec. 1999.

[2] M. Cheriet, J.N. Said, and C.Y. Suen, “A Recursive Thresholding Technique for Image Segmentation,” *IEEE Trans. Image Processing,* vol. 7, no. 6, pp. 918-921, June 1998.

[3] W.Y. Ma and B.S. Manjunath, “Edge Flow: A Technique for Boundary Detection and Image Segmentation,” *IEEE Trans. Image Processing,* vol. 9, no. 8, pp. 1375-1388, Aug. 2000.

[4] R. Castagno, T. Ebrahimi, and M. Kunt, “Video Segmentation Based on Multiple Features for Interactive Multimedia Applications,” *IEEE Trans. Circuits and Systems for Video Technology,* vol. 8, no. 5, pp. 562-571, Sept. 1998.

[5] I. Weiss, “Geometric Invariants and Object Recognition,” *Int'l J. Computer Vision,* vol. 10, no. 3, pp. 207-231, 1993.

[6] J. Munday and A. Zisserman, “Introduction—Towards a New Framework for Vision,” *Geometric Invariance in Machine Vision,* Cambridge, Mass.: MIT Press, 1992.

[7] N.H.C. Yung, H.Y. Chung, and P.Y.S. Cheung, “Quadrilateral-Based Region Segmentation for Tracking,” *Optical Eng.—The J. SPIE,* vol. 41, no. 11, pp. 2844-2855, Nov. 2002.

[8] H.Y. Chung, N.H.C. Yung, and P.Y.S. Cheung, “A Novel Quadrilateral-Based Tracking Method,” *Proc. Int'l Conf. Control, Automation, Robotics, and Vision,* 2002.

[9] R. Adams and L. Bischof, “Seeded Region Growing,” *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 16, no. 6, pp. 641-647, June 1994.

[10] J.A. Hartigan, *Clustering Algorithms.* John Wiley Sons, 1975.

[11] N.H.C. Yung and A.H.S. Lai, “Segmentation of Color Images Based on the Gravitational Clustering Concept,” *Optical Eng.—The J. SPIE,* vol. 37, no. 3, pp. 989-1000, Mar. 1998.

[12] W.E. Wright, “Gravitational Clustering,” *Pattern Recognition,* vol. 9, pp. 151-166, 1977.

[13] W.N. Lie, “Automatic Target Segmentation by Locally Adaptive Image Thresholding,” *IEEE Trans. Image Processing,* vol. 4, no. 7, pp. 1036-1041, July 1995.

[14] R.L. de Queiroz, Z. Fan, and T.D. Tran, “Optimizing Block-Thresholding Segmentation for Multiplayer Compression of Compound Images,” *IEEE Trans. Image Processing,* vol. 9, no. 9, pp. 1461-1471, Sept. 2000.

[15] M. Sonka, V. Hlavac, and R. Boyle, *Image Processing Analysis and Machine Vision,* pp. 176-180. London: Chapman & Hall, 1999.

[16] T. Brox, D. Farin, P.H.N. de With, “Multi-Stage Region Merging for Image Segmentation,” *Proc. 22nd Symp. Information Theory, in the Benelux,* pp. 181-196, May 2001.

[17] X. Hao, C.J. Bruce, C. Pislaru, and J.F. Greenleaf, “Segmenting High-Frequency Intracardic Ultrasound Images of Myocardium into Infracted, Ischemic, and Normal Regions,” *IEEE Trans. Medical Imaging,* vol. 20, no. 12, pp. 1373-1383, Dec. 2001.

[18] O. Lezoray and H. Cardot, “Cooperation of Color Pixel Classification Schemes and Color Watershed: A Study for Microscopic Images,” *IEEE Trans. Image Processing,* vol. 11, no. 7, July 2002.

[19] P. Mendonca and E.A. B. da Silva, “Segmentation Approach Using Local Image Statistics,” *IEE Electronics Letters,* vol. 36, no. 14, pp. 1199-1201, July 2000.

[20] H. Gao, W.-C. Siu, and C.-H. Hou, “Improved Techniques for Automatic Image Segmentation,” *IEEE Trans. Circuits and Systems for Video Technology,* vol. 11, no. 12, pp. 1273-1280, Dec. 2001.

[21] T. Pavlidis and F. Ali, “Computer Recognition of Handwritten Numerals by Polygonal Approximations,” *IEEE Trans. Systems, Man, and Cybernetics,* vol. 6, pp. 610-614, 1975.

[22] P. Gerken, “Object-Based Analysis-Synthesis Coding of Image Sequences at Very Low Bit-Rates,” *IEEE Trans. Circuits and Systems for Video Technology,* vol. 4, pp. 228-235, June 1994.

[23] F.S. Cohen, Z. Huang, and Z. Yang, “Invariant Matching and Identification of Curves Using B-spline Curve Representation,” *IEEE Trans. Image Processing,* vol. 4, pp. 1-10, Jan. 1995.

[24] Y.-H. Gu and T. Tjahjadi, “Efficient Planar Object Tracking and Parameter Estimation Using Compactly Represented Cubic B-Spline Curves,” *IEEE Trans. Systems, Man, and Cybernetics,* vol. 29, no. 4, pp. 358-367, July 1999.

[25] J. Nieweglowski, T.G. Campbell, and P. Haavisto, “A Novel Video Coding Scheme Based on Temporal Prediction Using Digital Image Warping,” *IEEE Trans. Consumer Electronics,* vol. 39, pp. 141-150, Aug. 1993.

[26] Y. Altunbasak and A.M. Tekalp, “Occlusion-Adaptive, Content-Based Mesh Design and Forward Tracking,” *IEEE Trans. Image Processing,* vol. 6, no. 9, pp. 1270-1280, Sept. 1997.

[27] M. Brejl and M. Sonka, “Object Localization and Border Detection Criteria Design in Edge-Based Image Segmentation: Automated Learning From Examples,” *IEEE Trans. Medical Imaging,* vol. 19, no. 10, pp. 973-985, Oct. 2000.

[28] D. Sappa and M. Devy, “Fast Range Image Segmentation by an Edge Detection Strategy,” *Proc. IEEE Conf. 3D Digital Imaging and Modeling,* pp. 292-299, May 2001.

[29] J.F. Canny, “A Computational Approach to Edge Detection,” *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 8, pp. 679-698, Nov. 1986.

[30] G. Iannizzotto and L. Vita, “Fast and Accurate Edge-Based Segmentation with No Contour Smoothing in 2-D Real Images,” *IEEE Trans. Image Processing,* vol. 9, no. 7, pp. 1232-1237, July 2000.

[31] R. Goldenberg, R. Kimmel, E. Rivlin, and M. Rudzsky, “Fast Geodesic Active Contours,” *IEEE Trans. Image Processing,* vol. 10, no. 10, pp. 1467-1475, Oct. 2001.

[32] L.D. Cohen, “On Active Contour Models and Balloons,” *CVGIP: Image Understanding,* vol. 53, pp. 211-218, Mar. 1991.

[33] M. Kass, A. Witkin, and D. Terzopoulos, “Snakes: Active Contour Model,” *Int'l J. Computer Vision,* pp. 321-331, Jan. 1988.

[34] R. Goldenberg, R. Kimmel, E. Rivlin, M. Rudzsky, “Fast Active Object Tracking in Color Video,” *Proc. 21st IEEE Convention of the Electrical and Electronic Eng. in Israel,* pp. 101-105, Apr. 2000.

[35] B. Sumengen, B.S. Manjunath, and C. Kenney, “Image Segmentation Using Multi-Region Stability and Edge Strength,” *Proc. Int'l Conf. Image Processing,* pp. 429-432, Sept. 2003.

[36] K. Haris, S.N. Efstratiadis, N. Maglaveras, and A.K. Katsaggelos, “Hybrid Image Segmentation Using Watersheds and Fast Region Merging,” *IEEE Trans. Image Processing,* vol. 7, no. 12, pp. 1684-1699, Dec. 1998.

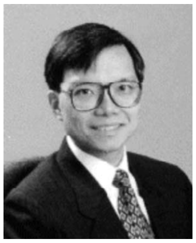
[37] Z. Yu and C. Bajaj, “Image Segmentation Using Gradient Vector Diffusion and Region Merging,” *Proc. Int'l Conf. Pattern Recognition,* pp. 828-831, Sept. 2002.

[38] J. Liu and Y.H. Yang, “Multiresolution Color Image Segmentation,” *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 16, no. 7, pp. 689-700, July 1994.

[39] Y.J. Zhang, “A Survey of Evaluation Methods for Image Segmentation,” *Pattern Recognition,* vol. 29, no. 8, pp. 1335-1346, 1996.

Ronald H.Y. Chung received the BEng degree in computer engineering, the MPhil degree, and the PhD degree in electrical and electronics engineering in 1997, 1999, and 2003, respectively, from the University of Hong Kong. He joined the Department of Computer Science in 2002, where he is now serving as a guest lecturer (also holding the title of honorary assistant professor), and is leading a research team to research into the area of object segmentation and tracking for video surveillance. His research interests include image and video processing, realtime systems, and Internet computing. He is a member of the IEEE.

Nelson H.C. Yung received the BSc and PhD degrees from the University of Newcastle-Upon-Tyne. He was lecturer at the same university from 1985 to 1990. From 1990 to 1993, he worked as senior research scientist at the Department of Defence, Australia. He joined the University of Hong Kong in late 1993 as associate professor. He leads a research team in digital image processing and Intelligent Transportation Systems. He is the founding Director of the Laboratory for Intelligent Transportation Systems Research at HKU and also deputy director of HKU's Institute of Transport Studies. Dr. Yung has coauthored five books and book chapters and has published more than 100 journal and conference papers in the areas of digital image processing, parallel algorithms, visual traffic surveillance, autonomous vehicle navigation, and learning algorithms. He regularly delivers talks/seminars to government units, professional institutions, associations, and commercial companies and gives interviews to local press regarding his research. He serves as reviewer for the *IEEE Transactions on Systems, Man, and Cybernetics*, *IEEE Transactions on Circuits and Systems for Video Technology*, *IEEE Transactions on Vehicular Technology*, *IEEE Transactions on Signal Processing*, *SPIE Optical Engineering*, SPIE *Journal of Electronic Imaging*, *HKIE Proceedings, Microprocessors and Microsystems*, and *Robotics and Autonomous Systems* journal. He was a member of the Advisory Panel of the ITS Strategy Review, Transport Department, HKSAR, regional secretary of IEEE Asia-Pacific region, council member of ITS-HK, and chair of computer division, International Institute for Critical Infrastructures. He was a Croucher Scholar and his team won the Silver Award from the Hong Kong Electronic Industry Association for Outstanding Innovation and Technology Product, 2000, for the MOVER (Mobile and Online Vending EnableR) solution. He is a chartered electrical engineer, member of the HKIE, IEE, and senior member of the IEEE. His biography is published in Who's Who in the World (Marquis, USA) since 1998.



Paul Y.S. Cheung received the BSc (Eng) degree and the PhD degree from the Imperial College of Science and Technology, University of London in 1973 and 1978, respectively. He joined the University of Hong Kong in 1980, and served as dean of engineering from 1994-2000. He was on secondment from the University to the Hong Kong SAR Government as the policy advisor of the Innovation and Technology Commission from 2002-2004. Prior joining the government, he was corporate senior vice-president of technology, at Pacific Century CyberWorks (PCCW) responsible for the strategic development in technology for the group while on leave from the University from 2000-2002. He is currently the managing director of Versitech Ltd of the University, program pirector of the Master of Science in E-Commerce and Internet Computing, and associate professor in the Department of Electrical and Electronic Engineering. He was the Asia Pacific region director of IEEE in 1995-1996, secretary of IEEE in 1997, and a member of IEEE Award Board in 2004-2005. His research interests include parallel computer architecture, Internet computing, VLSI design, signal processing, and pattern recognition. He is a chartered engineer and a senior member of the IEEE.