EURASIP Journal on Applied Signal Processing 2004:6, 798–813
© 2004 Hindawi Publishing Corporation

Spatio-Temporal Video Object Segmentation via Scale-Adaptive 3D Structure Tensor

Hai-Yun Wang

School of Electrical and Electronic Engineering, Nanyang Technological University, Nanyang Avenue, Singapore 639798
Email: haiyun@pmail.ntu.edu.sg

Kai-Kuang Ma

School of Electrical and Electronic Engineering, Nanyang Technological University, Nanyang Avenue, Singapore 639798
Email: ekkma@ntu.edu.sg.

Received 29 January 2003; Revised 5 September 2003

To address multiple motions and deformable objects' motions encountered in existing region-based approaches, an automatic video object (VO) segmentation methodology is proposed in this paper by exploiting the duality of image segmentation and motion estimation such that spatial and temporal information could assist each other to jointly yield much improved segmentation results. The key novelties of our method are (1) *scale-adaptive* tensor computation, (2) *spatial-constrained* motion mask generation *without* invoking dense motion-field computation, (3) *rigidity analysis*, (4) motion mask generation and selection, and (5) *motion-constrained* spatial region merging. Experimental results demonstrate that these novelties jointly contribute much more accurate VO segmentation both in spatial and temporal domains.

Keywords and phrases: video object segmentation, 3D structure tensor, rigidity analysis.

1. INTRODUCTION

Due to large amount of data and highly dynamic contents, digital video processing creates many technical challenges for conducting even some basic tasks that we, human beings, have been simply taking for granted in our daily lives. Among these operations, *video object (VO) segmentation* is an emerging signal processing tool, and is gradually becoming indispensable to many digital video applications often encountered in multimedia, virtual reality, computer vision, and machine intelligence. Given a digital video, how to bestow a machine with the capability of automatically (i.e., unsupervisedly) segmenting dominant VOs with reasonable accuracy on objects' boundaries is by no means a small goal.

Various VO segmentation methods [1, 2, 3, 4, 5, 6, 7, 8] are proposed to combine image (or spatial) and motion (or temporal) segmentations together to enhance the accuracy of VO extraction. Typical VO segmentation methodologies can be grouped into three categories: (1) region-based [1, 2]; (2) boundary-based [3, 4, 5]; and (3) probabilistic model-based approaches [6, 7, 8].

Region-based methods were developed by performing the clustering operation [1] or regional splitting and growing [2] on the feature space, which is usually formed by motion vectors and some spatial features, like color, texture, and position. However, accurate region boundary is difficult to achieve. Since human visual system (HVS) is very sensitive to the edge and contour information, *boundary-based* techniques were implemented with this consideration in mind by using *edge detectors* [3], *level set and fast marching* [4], or *active contours* [5], further combined with motion-field information for VO segmentation. Such approaches are very sensitive to noise, and the evolution of the active contour highly depends on the given initial position or convergence parameters imposed by the user. *Probabilistic model*-based methods exploit Bayesian inference [6], minimum description length (MDL) [7], or expectation maximization (EM) [8] to extract moving objects. Although these approaches are theoretically formulated, they suffer from high computational complexity. Some of them also need the number of objects/regions preassumed as an input parameter, which might prohibit their usage in practical applications.

Automatic VO segmentation is intimately affected by the image content and some frequently encountered issues: (1) *multiple motions*, which is encountered when multiple VOs under different moving velocities (i.e., various displacements and directions), and even with various object sizes, are involved in the video sequence. How to appropriately select the local scale size is imperative to achieve more accurate motion mask generation for moving VOs; and

(2) *deformable/nonrigid* motion, which is encountered when the VO moves with various changes in sizes and shapes during a scene of the video sequence. How to perform *rigidity analysis* is important to yield accurate motion models for capturing objects' individual characteristics.

To address the above-mentioned two problems encountered in VO segmentation, a novel VO segmentation methodology is proposed in this paper that integrates spatial and temporal information in a similar way to what are performed in the HVS cortex's pathways. The novelty of our method is that we only use the eigenvalues of the local three-dimensional (3D) structure tensor *without* computing dense motion vectors; thus, our method yields much lower computational load and is less sensitive to noise and global/background motion [9]. Furthermore, in the calculation of the 3D structure tensor, the *scale-adaptive* spatio-temporal Gaussian filter is introduced to handle multiple VOs under different motions in which the *scale* (i.e., the window size) is driven by the *condition number*. To differentiate whether the sequence contains rigid or nonrigid motion, rigidity analysis is performed using *correlation coefficients* over a range of successive video frames. The largest eigenvalue and coherency measurements of the 3D structure tensor are computed to form motion masks (i.e., *eigenmap* and *corner map*, respectively), which are further selected by the *change detection* and refined by graph-based spatial segmentation for rigid and nonrigid motions, respectively. And, for the final spatial VO segmentation, region merging is performed on adjacent over-segmented spatial segments based on the thresholding of the distance computed between the 3D structure tensor and an affine motion model [10]. Such a parametric method results in much more relevant VO segmentation and accurate VO boundaries as compared to energy minimization approach [11].

The paper is outlined as follows. Section 2 highlights the main ideas of our methodology. Section 3 introduces the basics of the 3D structure tensor and provides an overview of existing methods relevant to our work. Section 4 describes our proposed VO segmentation methodology. Experimental results of our scheme and comparison with other approaches are presented in Section 5. Finally, conclusions are drawn in Section 6.

2. FOUNDATION

2.1. The duality of image segmentation and motion estimation

In the previous works, several image segmentation [12] and motion segmentation [10, 13] techniques have been proposed for extracting the moving VOs. *Image segmentation* is to partition an image into nonoverlapping regions so that each one is "homogeneous" in some sense, such as intensity, color or, texture. The most commonly-used segmentation techniques can be classified into two broad categories [12]: (1) region-based segmentation that looks for regions satisfying a given homogeneity criterion; and (2) boundary-based segmentation that looks for boundaries between adjacent regions whose characteristics are different. Generally speaking, image segmentation techniques can produce good results among homogeneous regions with distinct boundaries (e.g., cartoon images), in which the produced segments are assumed to be piecewise constant/smooth. However, region-based techniques often fail to yield the desired region boundaries due to the difficulty of choosing a reasonable starting "seed" for region growing and appropriate growing/stopping rules. Moreover, boundary-based techniques are sensitive to noise and tend to be trapped into local minimum points like small edges.

Two main methods of motion estimation used for *motion segmentation* are optical flow (OF) and block matching. In both approaches, motion information is extracted through detecting the change of pixel intensities between successive frames in the video sequence. However, OF estimation is often chosen for achieving boundary-accurate VO segmentation because it allows motion detection at pixel level and ensures finer objects' boundaries than what block matching approach can accomplish. Furthermore, from the computational or numerical point of view, OF estimation is well-defined in the areas of complex textures/patterns with large gradients. But in piecewise constant regions, it suffers from the ill-posed least-squares constraint that is yielded by very small or zero local gradient; consequently, no motion vector can be estimated.

In summary, motion estimation is well-posed at the locations where image segmentation is ill-posed, such as texture-like areas, while image segmentation succeeds more easily in those areas where OF methods fail, such as homogeneous areas without (sufficient) gradients. That is, image segmentation techniques can more easily identify region boundaries where motion segmentation techniques have a difficulty. On the other hand, motion information is a helpful indicator to merge over-segmented spatial segments into semantic objects. Because of this duality, it is intuitive to construct an algorithm which uses image segmentation to assist the determination of motion field, and vice versa.

2.2. Two pathways involved in human visual perception

VO extraction should be in accordance with the human perception, which involves two cortical pathways: *form perception* pathway (processing spatial information) and *motion perception* pathway (processing temporal information) [14]. They interact with each other in all stages along the visual cortex in the HVS to associate different aspects of visual information and establish the perception of objects.

In order to fill the gap between perceiving processing in human eyes and the information processing in a digital computer, intensive research works for VO segmentation have been carried out (e.g., [15, 16]) by exploiting extracted spatial or temporal features. Since a moving VO usually has different motion features from the background and from other VOs, most proposed automatic VO segmentation approaches use motion information in *temporal* domain as an important cue to generate VOs' motion masks, and the spatial

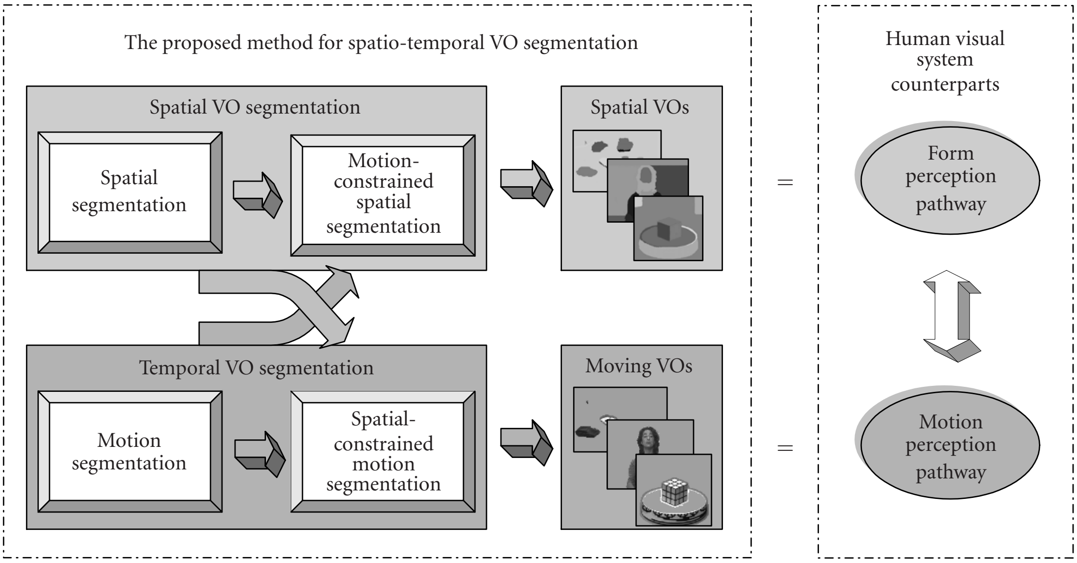

The proposed method for spatio-temporal VO segmentation
Spatial VO segmentation
Spatial segmentation
Motion-constrained spatial segmentation
Spatial VOs
Temporal VO segmentation
Motion segmentation
Spatial-constrained motion segmentation
Moving VOs
=
=
Human visual system counterparts
Form perception pathway
Motion perception pathway

FIGURE 1: Our proposed dual spatio-temporal scheme for automatic video object (VO) segmentation corresponding to the two pathways of the HVS.

information, like color, texture, and edge, is mainly used as an assistant cue to refine the generated motion mask, thus, only yielding the segmentation results for *moving* VOs with distinct motions.

However, little effort has been made for exploiting motion information to assist VO segmentation in the *spatial* domain, for it is quite helpful to extract and track temporally stand-still VOs, for example. Therefore, a new methodology is proposed in this paper by jointly exploiting the duality and synergism of spatial segmentation and motion estimation as illustrated in Figure 1, in which the processes in the four white rectangular boxes mimic the interactions incurred between the two pathways in the HVS. On the one hand, spatial VO segmentation is performed through merging the generated spatial masks driven by parametric motion models. On the other hand, temporal VO segmentation is achieved via refining the yielded motion masks by incorporating spatial information, thus, leading to the effective interaction between spatial segmentation and motion estimation. The detailed description of the processes implemented in each module of our framework as shown in Figure 1 will be presented in Section 4.

3. 3D STRUCTURE TENSOR-BASED VIDEO OBJECT SEGMENTATION

3.1. 3D structure tensor

Image sequence $L(\mathbf{x})$ can be treated as a *volume* data, where $\mathbf{x} = [x \;\; y \;\; t]^T$; x and y are the spatial components, and t is the temporal component. Spatio-temporal representation $I(\mathbf{x})$ is generated by convolving the image sequence $L(\mathbf{x})$ with a spatio-temporal filter $H(\mathbf{x})$. That is,

$$I(\mathbf{x}) = L(\mathbf{x}) * H(\mathbf{x}), \tag{1}$$

as "$*$" denotes convolution, and $H(\mathbf{x})$ is defined as

$$H(\mathbf{x}) = \frac{1}{\sqrt{2\pi\sigma_x^2}\sqrt{2\pi\sigma_y^2}\sqrt{2\pi\sigma_t^2}} \exp\left(-\frac{x^2}{2\sigma_x^2} - \frac{y^2}{2\sigma_y^2} - \frac{t^2}{2\sigma_t^2}\right), \tag{2}$$

where $\Sigma = [\sigma_x \;\; \sigma_y \;\; \sigma_t]$ is called the *spatio-temporal scale.*

The *3D structure tensor* is an effective representation of the local orientation for VO's spatio-temporal motion [17]. It can be generated on $I(\mathbf{x})$ according to

$$\begin{aligned} \mathbf{J} &= \begin{bmatrix} J_{11} & J_{12} & J_{13} \\ J_{21} & J_{22} & J_{23} \\ J_{31} & J_{32} & J_{33} \end{bmatrix} = \nabla I(\mathbf{x}) \cdot \nabla I(\mathbf{x})^T \\ &= \begin{bmatrix} I_x^2 & I_xI_y & I_xI_t \\ I_yI_x & I_y^2 & I_yI_t \\ I_tI_x & I_tI_y & I_t^2 \end{bmatrix}, \end{aligned} \tag{3}$$

where $\nabla := (\partial_x, \partial_y, \partial_t)$ denotes the spatio-temporal gradients. The eigenvalue analysis of the 3D structure tensor corresponds to a total least-squares (TLS) fitting of the local constant displacement of image intensities [17]. After performing eigenvalue decomposition of the 3×3 symmetric positive matrix $\mathbf{J}$, the eigenvectors $\mathbf{e}_k$ (for $k = 1, 2, 3$) of $\mathbf{J}$ can be used to estimate the local orientations. The corresponding eigenvalues λ_k of $\mathbf{e}_k$, which denote the local grayvalue variations along these directions, respectively, are sorted into the descending order $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq 0$ [17] for further analysis on their solution stability. The details will be presented in Section 4.2.4.

3.2. Previous works

In conventional OF estimation [18], only a small number of consecutive video frames are used for computing the motion

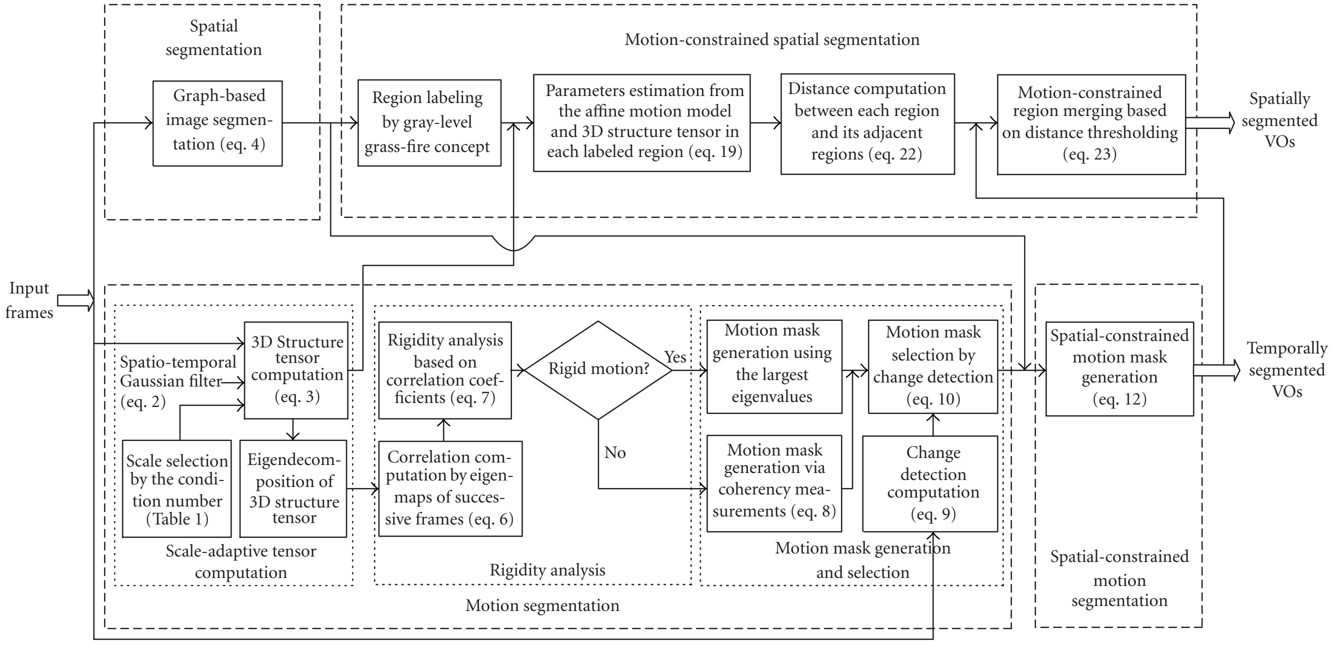

Spatial segmentation
Motion-constrained spatial segmentation
Graph-based image segmentation (eq. 4)
Region labeling by gray-level grass-fire concept
Parameters estimation from the affine motion model and 3D structure tensor in each labeled region (eq. 19)
Distance computation between each region and its adjacent regions (eq. 22)
Motion-constrained region merging based on distance thresholding (eq. 23)
Spatially segmented VOs
Input frames
Spatio-temporal Gaussian filter (eq. 2)
3D Structure tensor computation (eq. 3)
Scale selection by the condition number (Table 1)
Eigendecomposition of 3D structure tensor
Scale-adaptive tensor computation
Rigidity analysis based on correlation coefficients (eq. 7)
Correlation computation by eigenmaps of successive frames (eq. 6)
Rigidity analysis
Rigid motion?
Yes
No
Motion mask generation using the largest eigenvalues
Motion mask generation via coherency measurements (eq. 8)
Motion mask selection by change detection (eq. 10)
Change detection computation (eq. 9)
Motion mask generation and selection
Motion segmentation
Spatial-constrained motion mask generation (eq. 12)
Spatial-constrained motion segmentation
Temporally segmented VOs

Figure 2: Detailed description in each module of our proposed 3D structure tensor-based methodology for automatic VO segmentation in spatial and temporal domains, via exploiting the duality of image segmentation and motion estimation.

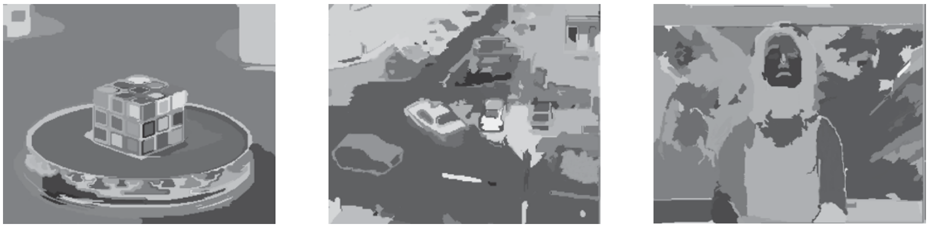
vectors, which might create “holes” within the motion masks and small isolated motion masks in the background. Therefore, a stack of consecutive frames treated as a 3D space-time image cube are used to estimate the motion vectors by analyzing the orientations of local gray-value structures, and this is described as the 3D structure tensor-based OF in [17]. Tensor-based OF field can be integrated with spatial information for improving VO segmentation as proposed in [5, 9, 10]. Such methods can be further classified into contour-based and region-based approaches as follows.

Contour-based VO segmentation lies on interactively refining the contour models based on motion masks generated from motion field. As proposed in [5], the tensor-based motion field is used as the *external force* to converge the geodesic active contour model and aligns the boundaries of the moving VOs. Instead of computing dense OF field for motion detection as described above, the novelty of the technique in [9] is that only the smallest eigenvalues of the 3D structure tensors are chosen and formed as the motion masks. Based on such motion information, the curve evolution driven by narrowband *level-set* technique [19] was implemented to perform VO segmentation. These contour-based techniques use the enclosed contours to match VOs which can reach more smooth and accurate objects’ boundaries than those obtained from the region-based approaches. But the evolution of the contour model is sensitive to the given initial contour, and it can be easily trapped into the local minimum positions like small edges or discontinuities of motion vectors.

Inspired by the region-based moving-layer segmentation scheme as proposed in [1], the 3D structure tensor was exploited as motion information in [10] to replace the conventional gradient-based OF in [1]. The segmentation is performed based on the region growing concept [12] as follows. First, the candidate regions are selected from the initially divided, but possibly overlapping, regions (e.g., with a fixed size of 21 × 21 pixels). Based on the distance computed between an affine motion model and each local 3D structure tensor, the candidate region with the smallest distance is identified, followed by the region-growing process, in which the costs of adjacent pixels of this region are computed and the pixel with the smallest distance will be added to this region. Such a region-growing process is implemented iteratively until the lower limit (200 pixels) or the upper limit (400 pixels) of the generated real region size is reached. However, this iterative region-based VO segmentation scheme is very time consuming, for example, consumes around 45 minutes per frame as mentioned in [10]. Furthermore, it is unable to detect multiple motions due to lacking of scale adaptation on tensor computation.

4. PROPOSED METHODOLOGY FOR SPATIO-TEMPORAL VO SEGMENTATION

To address the problems encountered in the existing 3D structure tensor-based VO segmentation approaches and to handle multiple VOs under various motions as described in Section 3.2, a unified region-based framework for performing spatio-temporal VO segmentation is proposed and illustrated in Figure 2, in which the processes in four dashed-line boxes are the detailed implementations of the corresponding main modules as shown in Figure 1, respectively.



(a) Rubik cube. (b) Taxi. (c) Silent.

FIGURE 3: Spatial segmentation results (the 9th frame) by implementing graph-based image segmentation approach [20].

In our methodology, for spatial segmentation, an efficient graph-based image segmentation approach [20] is implemented in the target frame to generate homogeneous spatial subregions with small intensity variations. These regions are exploited as the spatial constraint to refine the boundaries of motion masks. For motion segmentation, without computing the dense OF field, motion masks are obtained by executing the following three proposed steps: scale-adaptive tensor computation, rigidity analysis, and motion mask generation and selection, as shown in three subboxes belonging to motion segmentation dashed-line box, respectively. Finally, the spatial-constrained motion masks is generated and the motion-constrained spatial region merging is performed to achieve VO segmentation in spatial and temporal domains.

4.1. Spatial segmentation

Graph-based segmentation is based on the graphical representation of the image. The pixels are arranged as a lattice of vertices connected using either a first- or second-order neighborhood system. As proposed in [20], graph-based approach connects vertices with edges which are weighted by the intensity or RGB-space distance between the vertices' pixel values. After sorting the edges in a certain order, pixels are merged together iteratively based on some criteria as follows.

Let $G = (V, E)$ be an undirected graph with vertices $v \in V$, and $e_{\Omega_m,\Omega_n} \in E$ corresponds to the edge connected between each pair of neighboring segments Ω_m and Ω_n. Initially, each pixel $I(i, j)$ in the image is labeled as an unique segment Ω by itself. It is associated to its nearest eight neighboring pixels: $I(i-1, j-1), I(i-1, j), I(i-1, j+1), I(i, j-1), I(i, j+1), I(i+1, j-1), I(i+1, j)$, and $I(i+1, j+1)$ to form an eight-neighbor graph with the vertex of $I(i, j)$. Each edge between $I(i, j)$ and one of its neighbors is given a nonnegative weight computed from the intensity difference $\omega(e_{\Omega_m,\Omega_n}) = |I(\Omega_m) - I(\Omega_n)|$ for example. After all the edges are sorted in nondecreasing order according to their weights, the initial graph $G = (V, E)$ is constructed based on the weighted edges. Further region merging is started from the edge with the minimum weight. If both of the following criteria [11] are matched, two segments Ω_m and Ω_n need to be merged together, and the edges within them should be deleted from the initial graph $G = (V, E)$ to form the updated graph $G' = (V', E')$:

$$\begin{aligned} \omega(e_{\Omega_m,\Omega_n}) &\leq \text{MaxWeight}(\Omega_m) + \frac{\rho}{\text{Size}(\Omega_m)}, \\ \omega(e_{\Omega_m,\Omega_n}) &\leq \text{MaxWeight}(\Omega_n) + \frac{\rho}{\text{Size}(\Omega_n)}, \end{aligned} \tag{4}$$

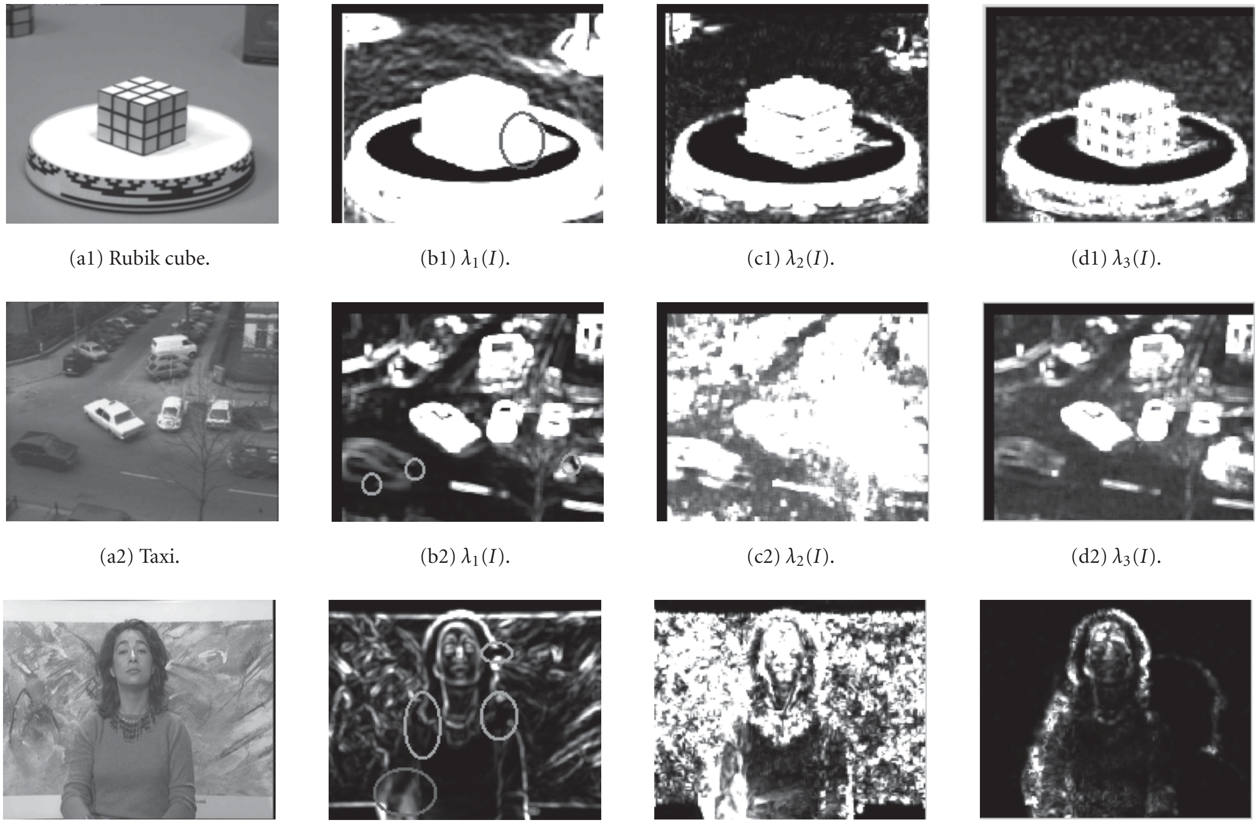
where MaxWeight(Ω_m) and MaxWeight(Ω_n) are the largest weights of the edges included in the segment Ω_m and Ω_n, respectively. Such a graph-based region merging process will be iterated until the edge with the maximum weight in the graph is reached. The factor ρ is used to adjust the segmented image between over-segmentation and under-segmentation. In order to avoid under-segmentation where two separately moving objects are joined into one spatial segment, the value of ρ is set to be 300 in our work.

This graph-based image segmentation algorithm is chosen because it performs the segmentation in $O(n \log n)$ time for n graph edges which takes about one second per frame using Pentium III 800 MHz personal computer. Furthermore, using the same image segmentation approach, our final motion-constrained spatial VO segmentation results can be fairly compared with the results provided in [11]. As suggested in [20], Gaussian filtering is used to remove noise as a preprocessing stage, and the scale-size of the spatial Gaussian filter is set to be 1.0 in our experiments. In the postprocessing stage, some small isolated regions are merged into their neighboring segments. The spatial segmentation results of the three test sequences are illustrated in Figure 3.

4.2. Motion segmentation

4.2.1. Exploiting the eigenvalues of conventional 3D structure tensor

Intuitively, $\nabla I(\mathbf{x}) \cdot \nabla I(\mathbf{x})^T$ in (3) can be viewed as a correlation matrix constituted by the gradient vectors of the space-time image volume. From the perspective of principal component analysis (PCA) [21], if the eigenvectors of the correlation matrix computed from the input data are sorted in the descending order, the first eigenvector which corresponds to the largest eigenvalue indicates the direction that incurs the largest variance of the data. Furthermore, the ratio of each eigenvalue to the total sum of three eigenvalues reveals how much of the data energy is concentrated along the corresponding eigenvector (direction) [21]. Therefore,



(a1) Rubik cube.

(b1) $\lambda_1(I)$.

(c1) $\lambda_2(I)$.

(d1) $\lambda_3(I)$.

(a2) Taxi.

(b2) $\lambda_1(I)$.

(c2) $\lambda_2(I)$.

(d2) $\lambda_3(I)$.

(a3) Silent.

(b3) $\lambda_1(I)$.

(c3) $\lambda_2(I)$.

(d3) $\lambda_3(I)$.

FIGURE 4: Figures 4a1, 4a2, and 4a3 are the 9th frames of the three test sequences; (b), (c), and (d) are the eigenmaps based on the three eigenvalues λ_1, λ_2, and λ_3, respectively, using conventional fixed-scale 3D structure tensor. Note that $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq 0$.

the eigenvalues of local 3D structure tensor can be used to detect the local variances of the input frames.

The *smallest* eigenvalue has been proposed in [9] as the indicator of the frame difference, which was proved to be more robust to noise and low object-background contrast as compared to the simple frame difference. To further investigate, the three *eigenmaps* based on the three eigenvalues $\lambda_1(x, y, t)$, $\lambda_2(x, y, t)$, and $\lambda_3(x, y, t)$ of the local 3D structure tensor are denoted as $\lambda_1(I)$, $\lambda_2(I)$, and $\lambda_3(I)$, respectively, and illustrated in Figure 4. It has been observed that, in fact, eigenmap $\lambda_1(I)$ captures both the moving objects and some of isolated texture-like areas in the background. The information revealed in eigenmap $\lambda_2(I)$ as shown in Figures 4c1, 4c2, and 4c3 is not so informative as that of $\lambda_1(I)$, thus, more difficult to exploit for VO segmentation. Furthermore, eigenmap $\lambda_1(I)$, in general, shows more accurate boundaries around the moving VOs and less number of small holes within the VOs' masks (see Figures 4b1, 4b2, and 4b3) than those generated by $\lambda_3(I)$ (see Figures 4d1, 4d2, and 4d3); thus, $\lambda_1(I)$ is selected to generate the motion mask in our scheme.

Notice that both multiple motions (e.g., "Taxi" sequence) and deformable motions (e.g., "Silent" sequence) cannot be handled accurately by applying the conventional fixed-scale 3D structure tensor. (See Figures 4b2 and 4b3 for demonstration with explanation provided below.) This is due to the fact that there is no scale adaptation for conventional 3D structure tensor computation. That is, the fixed-scale $\Sigma = [\sigma_x \;\; \sigma_y \;\; \sigma_t]$ was used in (2) for the spatio-temporal Gaussian filter $H(\mathbf{x}, \Sigma)$.

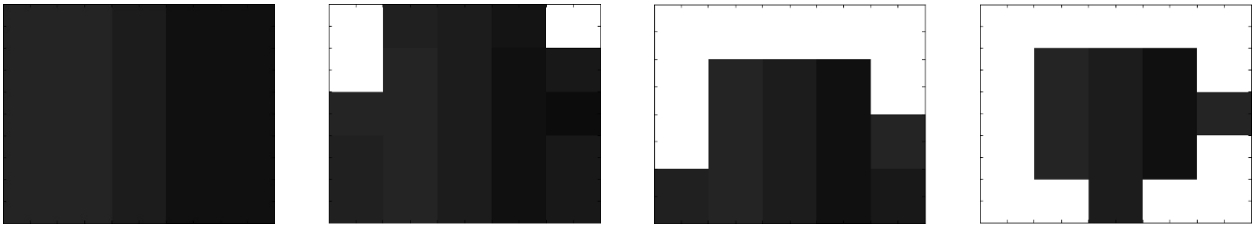
Consequently, exploiting large scale size for slow motion will reduce the effectiveness of localization, causing inaccurate motion boundaries as highlighted by the circle in Figure 4b1. On the other hand, large displacement of a VO cannot be properly matched if a small scale window was exploited, thus, leading to unconnected motion masks as highlighted by the two small circles in Figure 4b2. Such phenomena are also incurred for the deformable moving object as shown in Figure 4b3 which contains multiple motions within one body like rotating and translating. Therefore, it is highly desirable to have adaptive scale for the spatio-temporal filtering rather than using *fixed* scale.

4.2.2. Scale-adaptive 3D structure tensor computation

Due to possible involvement of different velocities in a local region, the small scale size would not be able to

Table 1: Experimental scales and spatial windows for the spatio-temporal Gaussian filter, where the three component values in Σ correspond to the scales on directions x, y, and t, respectively.

Scale $\Sigma = [\sigma_x \quad \sigma_y \quad \sigma_t]$	[0.5 0.5 0.5]	[1 1 1]	[1.5 1.5 1.5]	[2 2 2]	[2.5 2.5 2.5]
Spatio-temporal window	$3 \times 3 \times 3$	$5 \times 5 \times 5$	$7 \times 7 \times 7$	$9 \times 9 \times 9$	$11 \times 11 \times 11$



(a) CN = ∞.

(b) CN = 6.8229×10^3.

(c) CN = 68.0100.

(d) CN = 5.8320.

Figure 5: Some typical spatial subregions and their corresponding condition number (CN) computed from the matrix which is constituted by the pixels' grayvalues: (a) homogeneous region, (b) region with corners, (c) region with edges, and (d) region with corners and edges.

match/capture the motion of a VO with large displacements, thus, leading to unconnected object boundaries. On the other hand, exploiting large scale size for slow motions will reduce the effectiveness of localization and cause blurred motion discontinuities, thus, causing less accurate estimation due to the local minima. Therefore, representing images at multiple scales is a good approximation of the HVS on perceiving images. Several multiscale methods were proposed using *nonlinear filtering* [22, 23], *Gaussian pyramid* [5, 24], *multiwindow* [25] or *scale-space* [26, 27, 28]. For these multiscale approaches, automatic scale selection is an essential problem to be addressed. Since our method for motion detection is based on the 3D structure tensor without dense OF field estimation, we propose an effective automatic scale-selection method with incorporation of the measurement of local image structure.

In the previous works, the spatio-temporal filter with variable scales is introduced in [29] by iterative symmetric Schur decomposition. But its scale adaptation through thresholding is determined experimentally. In the 3D structure tensor-based method, the TLS approach [30] is exploited for OF estimation. Since the numerical stability of the TLS solution can be indicated by singular value decomposition (SVD) [30] of the local grayvalue variations, we exploit the condition number to guide the scale selection of the spatio-temporal Gaussian filter $H(\mathbf{x}, \Sigma)$, which is defined as follows. The condition number of a local area I_Ω can be computed by

$$\mathrm{Cond}\left(I_\Omega\right) = ||I_\Omega|| ||I_\Omega^{-1}|| = \frac{\sigma_{\max}}{\sigma_{\min}}, \tag{5}$$

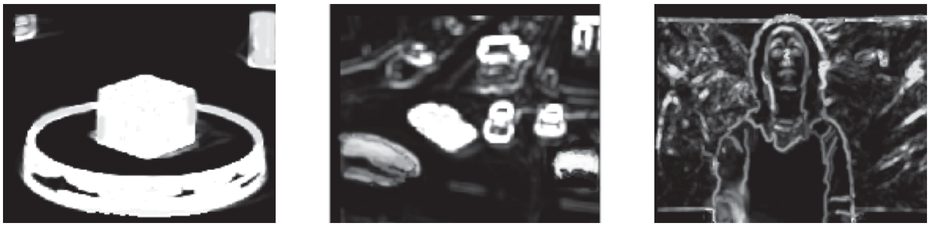
where Ω denotes any area in the input frame whose size is determined by the spatial scales σ_x and σ_y of the spatio-temporal filter, which can be referred to in Table 1. $\sigma_{\max}$ is the maximum singular value and $\sigma_{\min}$ is the minimum singular value, which are obtained by performing SVD on the matrix constituted by the grayvalues of each of the Figures 5a, 5b, 5c, and 5d as illustrated. Note that the condition number of a singular matrix is infinite, and a smaller condition number implies a more stable solution.

It can be further observed from Figure 5 that the more homogeneous the area, the larger value the condition number. The reason for this phenomenon is that coherent grayvalues will cause high correlation in matrix I_Ω; thus, the computed condition number is near to the infinity as shown in Figure 5a. With the presence of corners and edges, the matrix correlation is decreased significantly, and the condition number becomes much smaller (see Figures 4b, 4c, and 4d). Therefore, it is reasonable to use the condition number of the local intensities to steer the scale Σ of the spatio-temporal Gaussian filter. In our experiments, the initial scale Σ is set to be $[0.5 \quad 0.5 \quad 0.5]$ (thus, using $3 \times 3 \times 3$ window as indicated in Table 1), and it will be extended progressively according to Table 1 until either the condition number is below a threshold (e.g., 100) or the scale size reaches the maximum $11 \times 11 \times 11$.

The eigenmaps of the largest eigenvalues computed from the scale-adaptive 3D structure tensor are illustrated in Figure 6. More accurate boundaries and more integrity motion masks can be observed as compared to those in Figure 4 for various test sequences. However, note that the result for the *nonrigid* moving VO (see Figure 6c) fails to yield meaningful motion masks. On the contrary, satisfactory motion masks are generated for *rigid* VOs (see Figures 6a and 6b). Thus, a rigidity analysis is developed in the following to distinguish whether the sequence frame contains rigid or nonrigid VOs, and further facilitating the following motion mask generation processes.

4.2.3. Rigidity analysis

A dynamic region matching is proposed in [31] for conducting rigidity analysis using the residual values computed from


(a) Rubik cube.

(b) Taxi.

(c) Silent.

Figure 6: The eigenmaps $\lambda_1(I)$ (the 9th frame) based on the *largest* eigenvalues of the scale-adaptive 3D structure tensors.

the difference between the motion vectors and the initialized velocity. However, its results are affected by the inaccuracy of VO tracking and motion estimation. Without invoking OF computation, we propose an efficient rigidity analysis method by exploiting the correlation between two successive frames based on their 3D structure tensors. The basic concept is quite intuitive as follows. If the moving VO has rigid motion under a certain speed, then only the *interframe* changes will be observed. On the other hand, for the nonrigid moving VO, besides the interframe changes, the *intraframe* changes can also be observed in the body of VOs. Therefore, the correlation between two successive frames is expected to be high for rigid VOs and low for nonrigid VOs.

As illustrated in Figures 4d1, 4d2, and 4d3, the eigenmap $\lambda_3(I)$ inclines to indicate only the moving parts of VOs and reveals much less textured details of the still background than that in $\lambda_1(I)$ (see Figures 4b1, 4b2, and 4b3). Therefore, the correlation coefficient R [32] is computed based on two successive eigenmaps $\lambda_3(I_t)$ and $\lambda_3(I_{t+1})$ of frames I_t and I_{t+1}, respectively, as follows:

$$R = \frac{\sum_{i=1}^{N}(x_i \cdot y_i) - (1/N)\left(\sum_{i=1}^{N} x_i \cdot \sum_{i=1}^{N} y_i\right)}{\sqrt{\left(\sum_{i=1}^{N} x_i^2 - (1/N)\left(\sum_{i=1}^{N} x_i\right)^2\right)\left(\sum_{i=1}^{N} y_i^2 - (1/N)\left(\sum_{i=1}^{N} y_i\right)^2\right)}}, \tag{6}$$

where $x_i \in \lambda_3(I_t)$ and $y_i \in \lambda_3(I_{t+1})$. N is the total number of pixels in the frame.

It can be seen that the fluctuation of the curve (see Figure 7) for rigid VOs (e.g., "Taxi" and "Rubik cube") is much smoother than that for the nonrigid VO (e.g., "Silent"). Such a fluctuation can be measured by the standard deviation S [32] of the correlation coefficients R_i, for $i = 1, 2, \ldots, n$, as

$$S = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}\left(R_i - \bar{R}\right)^2}, \tag{7}$$

where n is the total number of R_i over a set of frames under consideration and $\bar{R}$ is the average of R_i. The values of S computed from "Rubik cube," "Taxi," and "Silent," are 0.013, 0.0126, and 0.0436, respectively. Based on extensive experiments, the threshold for S is determined to be 0.015. Sequence with the value of S lower than 0.015 is considered having rigid VOs; otherwise, it contains nonrigid VOs.

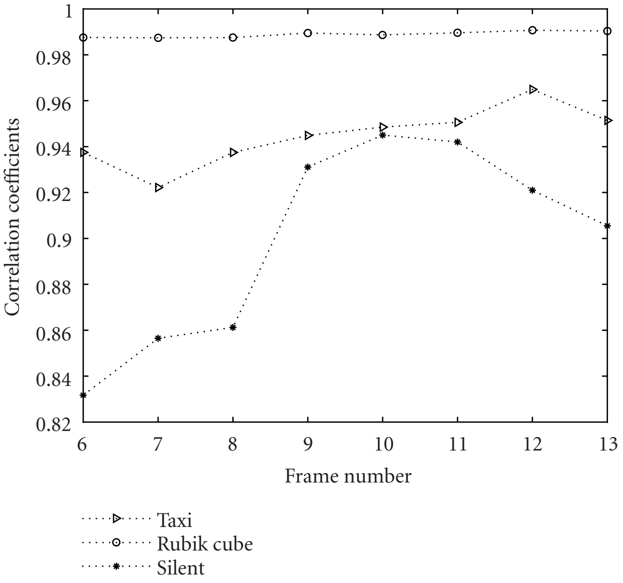


Figure 7: Correlation coefficients computed over a range of successive video frames.

4.2.4. *Motion mask generation and selection*

Basics of the eigenvalue analysis of 3D structure tensor

Since tensor-based OF estimation is based on the TLS approach, its solution can be resolved by using the widely used Jacobi method [30] to perform eigenvalue decomposition of the 3D structure tensor **J**. The generated three eigenvalues λ_k (for $k = 1, 2, 3$), which denote the local grayvalue variations along local dominant directions [17], respectively, can be exploited to derive the *coherency measurements* for motion field classification.

(i) If all the three eigenvalues are equal to zero, that is, rank(**J**) = 0, it means that all its partial derivatives along the principal axes (x, y, and t) vanish. Physically, this indicates that the local area has a constant grayvalue; thus, no motion can be detected.

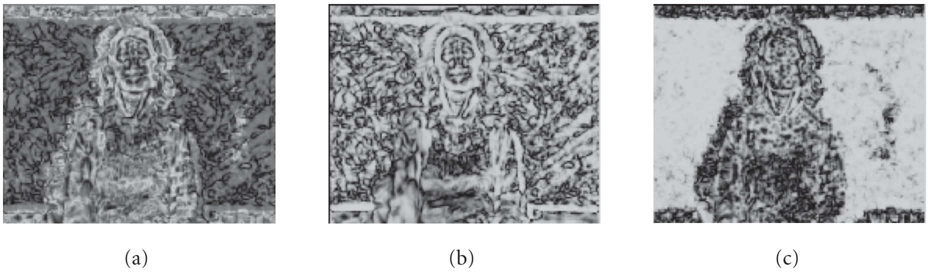


Figure 8: Maps based on the coherency measurements of the scale-adaptive 3D structure tensors: (a) total coherency measure C_t, (b) edge measure C_s, and (c) corner measure C_c.

(ii) If $\lambda_1 > 0$ and $\lambda_2 = \lambda_3 = 0$, that is, $\text{rank}(\mathbf{J}) = 1$, this indicates that the grayvalue changes only happen in the normal direction, indicating that the area contains an edge. This is the well-known *aperture problem* encountered in OF estimation.

(iii) If $\lambda_1 > 0$, $\lambda_2 > 0$, and $\lambda_3 = 0$, that is, $\text{rank}(\mathbf{J}) = 2$, this indicates that a spatio-temporal structure containing grayvalues changes in two directions, and it moves at a constant speed, thus, indicating a corner area. The real motion can be accurately estimated in this case.

(iv) If all the three eigenvalues are greater than zero, that is, $\text{rank}(\mathbf{J}) = 3$, this indicates that the local area is located on the border of two moving fields under different motions; thus, no reliable motion can be estimated due to the presence of motion discontinuity.

Although the rank of $\mathbf{J}$ proves to contain all necessary information to distinguish different types of motion, it cannot be used for practical implementations because it does not constitute a normalized measure of certainty. Therefore, the coherency measurements for motion field classification have been proposed [17], which yield real-valued numbers between zero and one.

Coherency measurements

The purpose of computing coherency measurements [17] in our method is to provide some indicators regarding the motion of nonrigid moving objects. Instead of using the *parametric* approaches for nonrigid VO segmentation as proposed in [7, 10], which need dense motion field and are sensitive to motion estimation errors, a *nonparametric* method is proposed here using coherency measurements to quantify the degree of motion estimation certainty. They were derived from the eigenvalues of the 3D structure tensor and can be used as the indicators of local motion structures, such as edge, corner, homogeneous region, and so on. They are defined [17] as follows:

(i) total coherency measure: $C_t = ((\lambda_1 - \lambda_3)/(\lambda_1 + \lambda_3))^2$,

(ii) edge measure: $C_s = ((\lambda_1 - \lambda_2)/(\lambda_1 + \lambda_2))^2$,

(iii) corner measure:

$$C_c = C_t - C_s = \frac{4\dot{\lambda}_1(\lambda_2 - \lambda_3)(\lambda_1^2 - \lambda_2\lambda_3)}{(\lambda_1 + \lambda_3)^2(\lambda_1 + \lambda_2)^2}. \qquad (8)$$

The masks of C_t, C_s, and C_c computed from the local scale-adaptive 3D structure tensors of "Silent" are illustrated in Figures 8a, 8b, and 8c, respectively. Among them, the map of corner measure C_c (i.e., *corner-map*) reveals the most distinct VO boundary information to yield the motion masks for nonrigid motions; thus, it is exploited in our framework.

Change detection computation

Change detection is used as the indicator for motion mask selection in our scheme because it can be implemented efficiently and enables the detection of appearance motion according to the predetermined thresholds [33]. The purpose of change detection is to locate moving objects through detecting intensity changes between subsequent frames of image sequences. One of the change detection techniques is so-called *frame differencing* $D(N)$ [33], which is defined as

$$D(N) = ||I(t + N) - I(t)||, \qquad (9)$$

where $\| * \|$ is the L_p norm, and $I(t)$ and $I(t + N)$ are the tth frame and the $(t + N)$th frame, respectively. The threshold setting of $D(N)$ depends on the requirement of practical applications. Since the image with noise (e.g., illumination change) may cause false alarms or missing parts of the motion mask, in our method, the threshold for $D(N)$ in (9) is set to be high enough (e.g., 30) in order to avoid the occurrence of false alarm. The missing parts of $D(N)$ within the areas of moving objects will not affect our motion segmentation results because the final motion masks are not generated from $D(N)$, it is only used for motion mask *selection* here.

Motion mask selection

So far, we obtained the eigenvalue mask (based on $\lambda_1(I)$) and corner mask (based on C_c) for rigid and nonrigid motion detection, respectively. Although there is no obvious camera motion in the test sequences we experimented, the obtained motion masks, however, do contain not only the moving areas but also some parts of the still background, as shown in Figures 6 and 8. The undesirable areas from the still background are caused by the computation of the 3D structure tensor on still, but textured, areas, yielding high spatial gradients but low temporal gradients. To exclude the undesirable areas, $D(N)$ is used here because it can identify the position

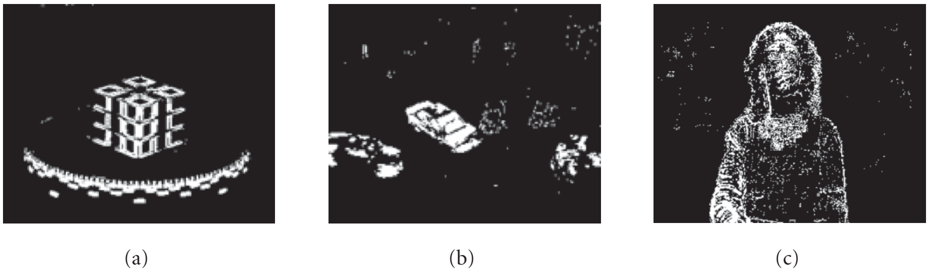
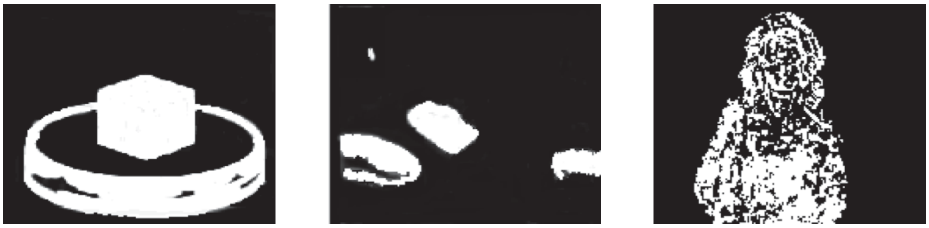


Figure 9: Change detection based on the 5th and the 9th frames via (9): (a) "Rubik cube," rigid rotating motion, (b) "Taxi," rigid moving VOs under different motions, and (c) "Silent," nonrigid moving VO.



(a) $\lambda_1(I)$, Rubik cube. (b) $\lambda_1(I)$, Taxi. (c) C_c, Silent.

Figure 10: Motion mask selection results (the 9th frame) obtained by the proposed percentage thresholding method using the original motion masks (see Figures 6a, 6b, and 8c) and the corresponding change detection maps (see Figures 9a, 9b, and 9c).

of moving objects correctly from the still background as illustrated in Figure 9.

Using a rigid VO as an example, if the size of its motion mask is large enough both in the map $D(N)$ (see Figures 9a and 9b) and in eigenmap $\lambda_1(I)$ (see Figures 6a and 6b), that is, there is distinct motion that occurred within the mask area, thus, the eigenmap mask is considered as part of the moving VO. Otherwise, it is determined to be part of the background. The proposed motion mask selection is performed using our proposed *percentage thresholding* method as follows.

In order to select (i.e., keep or delete) the masks in eigenmap $\lambda_1(I)$ one by one, each area (either in white color or in black color in Figure 6) is labeled by an unique number using *gray image grass-fire labeling* as proposed in [34], which is an extended version of the *grass-fire* concept [35] for gray-level image labeling. The labeled area in $\lambda_1(I)$ is denoted as A_{eigen}. The percentage R_c of change detection mask A_{change} (white pixels in Figure 9) within the labeled area A_{eigen} of eigenmap $\lambda_1(I)$ is computed as

$$R_c = \frac{A_{\text{change}}}{A_{\text{eigen}}} \times 100\%. \tag{10}$$

If the value of R_c is larger than the predetermined threshold (e.g., 40 %), A_{eigen} is kept as the motion mask of a moving VO. Otherwise, area A_{eigen} is considered as part of the background because there is no distinct motion that occured in it.

For nonrigid motions, the motion mask selection process is implemented in the same way as for rigid motions as described above, except that A_{eigen} should be replaced by the mask A_{corner} in the corner map C_c (illustrated by the white color in Figure 8c), and the computation of R_c should be modified as

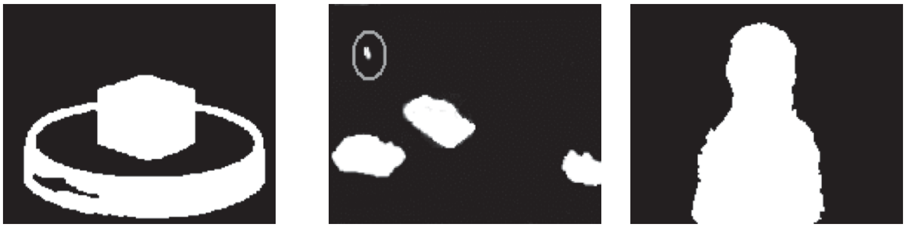
$$R_c = \frac{A_{\text{change}}}{A_{\text{corner}}} \times 100\%. \tag{11}$$

After the motion mask selection process, motion masks for moving VOs are generated in eigenmaps (see Figures 10a and 10b) and in the corner map (see Figure 10c), where the homogeneous background and the selected motion masks are shown in black and white colors, respectively.

4.3. Spatial-constrained motion segmentation

However, the motion masks as shown in Figure 10 still have small holes in the body of VOs and inaccurate boundaries along the borders of VOs. To address this problem, graph-based image segmentation results (see Figure 3) as described in Section 4.1 is used in order to benefit from the advantages of spatial segmentation, such as the integrity of spatial segments and more accurately segmented boundaries.

To refine the boundaries of the selected motion masks (those white-color areas in Figure 10), the shape of each motion mask should be constrained by the shape of its corresponding spatial segment in Figure 3. If the percentage of the motion mask is high enough within a spatial subregion, the shape of the spatial segment will be used to replace the corresponding shape of the motion mask; thus, the boundary of the spatial-constrained motion mask can align the border of the moving VO.



(a) Rubik cube.

(b) Taxi.

(c) Silent.

Figure 11: Spatial-constrained motion masks generated based on our method using three video sequences, respectively.

Each spatial segment in the frame illustrated in Figure 3 is labeled by an unique number according to its color. The percentage R_m of the motion mask A_{mask} within each A_{seg} is computed as

$$R_m = \frac{A_{\text{mask}}}{A_{\text{seg}}} \times 100\%. \tag{12}$$

If the value of R_m is larger than the predetermined threshold (e.g., 50 %), the whole spatial segment A_{seg} should be viewed as the portion of the moving VO (using white color); otherwise, it is treated as part of the background (using black color).

As compared with the motion masks as shown in Figure 10, the spatial-constrained motion masks, as illustrated in Figures 11a, 11b, and 11c, have yielded more accurate boundaries which are aligned with the borders of respective moving VOs. The small holes that occured within some motion masks in Figure 10 are also absent in Figure 11.

4.4. Motion-constrained spatial merging

Similar to the synergism existing between the form and motion perception pathways in the HVS, the proposed methodology is to make spatial segmentation and motion estimation constrained on each other. For that, the boundary information from image segmentation has contributed to the motion mask generation as described in Section 4.3. On the other hand, how to merge the over-segmented spatial regions based on the motion information is the next issue to be addressed in this section as follows.

There are two classes of region merging approaches: nonparametric techniques [12] and parametric models [10, 11]. The nonparametric approach such as boundary melting [12] cannot be integrated with the motion field because the motion field discontinuities are difficult to be identified accurately. A parametric method for region merging using energy minimization was proposed in [11] based on Horn's OF field [36]. The results are sensitive to motion estimation errors, and it is computationally slow through the iterative way. Since there is no dense motion field required in our scheme, a novel parametric motion model [10] for the motion-constrained region merging is exploited by using the 3D structure tensor. The parameters of the affine motion model estimated from each spatial segment are used to compute the distance between two adjacent segments. Two segments will be merged together if the motion model distance between them is short enough, that is, sharing the similar motions.

4.4.1. Affine motion model

The planar motion at each pixel position $I(x, y)$ can be described by an affine model containing six parameters:

$$\begin{aligned} v_x(x, y) &= ax + by + c, \\ v_y(x, y) &= dx + ey + f, \end{aligned} \tag{13}$$

where v_x and v_y are the x and y components of the velocity $\mathbf{v}$, and a, b, c, d, e, and f are the parameters of the model.

The 2D velocity can be extended to 3D directional vector $\mathbf{v}$ in order to include the unit temporal velocity, which is defined as

$$\begin{aligned} \mathbf{v} = \begin{bmatrix} v_x \\ v_y \\ 1 \end{bmatrix} &= \begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} x & y & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & x & y & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \\ 1 \end{bmatrix} \\ &= \mathbf{SP}. \end{aligned} \tag{14}$$

4.4.2. Parameter estimation

The motion-model parameters for a VO are estimated by merging the spatial segmented regions based on a distance function measured between the affine motion model and the 3D structure tensor $\mathbf{J}_i$ (for pixel i), which can be derived [10] as follows:

$$d(\mathbf{v}_i, \mathbf{J}_i) = \mathbf{v}_i^T \mathbf{J}_i \mathbf{v}_i = \mathbf{P}^T \mathbf{S}_i^T \mathbf{J}_i \mathbf{S}_i \mathbf{P} = \mathbf{P}^T \mathbf{Q}_i \mathbf{P}, \tag{15}$$

where $\mathbf{Q}_i = \mathbf{S}_i^T \mathbf{J}_i \mathbf{S}_i$ is a positive quadratic matrix. The sum of the pixel-wise distances within a given spatial segment containing N pixels is as follows:

$$d_{\text{seg}}(\mathbf{P}) = \sum_{i=1}^{N} d(\mathbf{v}_i, \mathbf{J}_i) = \mathbf{P}^T \left(\sum_{i=1}^{N} \mathbf{Q}_i \right) \mathbf{P} = \mathbf{P}^T \mathbf{Q}_{\text{seg}} \mathbf{P}. \tag{16}$$

Make the partitions of $\mathbf{P}$ and $\mathbf{Q}_{\text{seg}}$ as follows:

$$\mathbf{P} = \begin{bmatrix} \mathbf{p}_1 \\ 1 \end{bmatrix}, \qquad \mathbf{Q}_{\text{seg}} = \begin{bmatrix} \mathbf{q}_1 & \mathbf{q}_2 \\ \mathbf{q}_2^T & g \end{bmatrix}, \tag{17}$$

where $\mathbf{P}_1 = [a \;\; b \;\; c \;\; d \;\; e \;\; f]^T$, $\mathbf{q}_1$ is a symmetric 6 × 6 matrix, $\mathbf{q}_2$ is a 6 × 1 vector, and g is a scalar. Hence,

$$d_{\text{seg}}(\mathbf{P}) = \mathbf{p}_1^T \mathbf{q}_1 \mathbf{p}_1 + \mathbf{p}_1^T \mathbf{q}_2 + \mathbf{q}_2^T \mathbf{p}_1 + g. \tag{18}$$

Now the problem boils down to find the parameters in vector $\mathbf{p}_1$ that minimizes $d_{\text{seg}}(\mathbf{P})$, and this can be derived as [10] follows:

$$\hat{\mathbf{p}}_1 = -\mathbf{q}_1^{-1}\mathbf{q}_2. \tag{19}$$

Thus, the residual distance value is computed by

$$d_{\text{seg}}(\hat{\mathbf{P}}) = \mathbf{q}_2^T \hat{\mathbf{p}}_1 + g. \tag{20}$$

Notice that $\mathbf{q}_1$ in (19) is not invertible if the spatial segment is (nearly) homogeneous, that is, the vectors in $\mathbf{q}_1$ are highly correlated. In this case, the estimated parameter is changed to $\hat{\mathbf{p}}_1 = -\mathbf{q}_1^{+}\mathbf{q}_2$, where $\mathbf{q}_1^{+}$ is the pseudo inverse of $\mathbf{q}_1$. In fact, the invertibility of $\mathbf{q}_1$ is not an issue for our application because the obtained spatial segments contain certain small gradients which can highly decrease the correlation among the vectors of $\mathbf{q}_1$.

4.4.3. *Normalized distance computation*

Since the quadratic form of the velocity in (15) is more sensitive to large velocities than to small ones as analyzed in [10], $d(\mathbf{v}_i, \mathbf{J}_i)$ is divided by the norm of $\mathbf{v}$ and the trace of $\mathbf{J}_i$ to avoid this issue. The normalized distance is defined as follows:

$$\bar{d}(\mathbf{v}_i, \mathbf{J}_i) = \frac{d(\mathbf{v}_i, \mathbf{J}_i)}{|\mathbf{v}_i|^2 \operatorname{tr}(\mathbf{J}_i)} = \frac{\mathbf{P}^T \mathbf{Q}_i \mathbf{P}}{|\mathbf{S}_i \mathbf{P}|^2 (\lambda_1 + \lambda_2 + \lambda_3)}. \tag{21}$$

Thus, the normalized residual distance used to measure the motion model distance from each spatial segment to its adjoining segments can be derived from (16) and (20) as follows:

$$\bar{d}_{\text{seg}}(\hat{\mathbf{P}}) = \frac{g + \mathbf{q}_2^T \hat{\mathbf{p}}_1}{\sum_{i=1}^{N} |\mathbf{v}_i|^2 \operatorname{tr}(\mathbf{J}_i)} = \frac{g + \mathbf{q}_2^T \hat{\mathbf{p}}_1}{\sum_{i=1}^{N} |\mathbf{S}_i \hat{\mathbf{P}}|^2 (\lambda_{1i} + \lambda_{2i} + \lambda_{3i})}. \tag{22}$$

4.4.4. *Spatial region merging*

The region merging is implemented based on the parametric motion model described in Section 4.4.2. Firstly, each segment which was obtained from the graph-based image segmentation is labeled by a unique number using gray image grass-fire labeling [34]. In each labeled spatial subregion Ω_i, six affine parameters $\hat{\mathbf{p}}_{1\Omega i}$ estimated from (19) are applied to its M neighbors Ω_j, for $j = 1, 2, \ldots, M$, but, $j \neq i$, to compute the *normalized residue distance* for each Ω_j based on (22):

$$\begin{aligned} \bar{d}_{\Omega_j}(\hat{\mathbf{P}}_{\Omega i}) &= \frac{g_{\Omega j} + \mathbf{q}_{2\Omega j}^T \hat{\mathbf{p}}_{1\Omega i}}{\sum_{k=1}^{N} |\mathbf{v}_k|^2 \operatorname{tr}(\mathbf{T}_k)} \\ &= \frac{g_{\Omega j} + \mathbf{q}_{2\Omega j}^T \hat{\mathbf{p}}_{1\Omega i}}{\sum_{k=1}^{N} |\mathbf{S}_k \hat{\mathbf{P}}_{\Omega i}|^2 (\lambda_{1k} + \lambda_{2k} + \lambda_{3k})}, \\ &\qquad \text{for } j = 1, 2, \ldots, M, \; j \neq i, \end{aligned} \tag{23}$$

where k indicates the sequence number of N pixels in segment Ω_j.

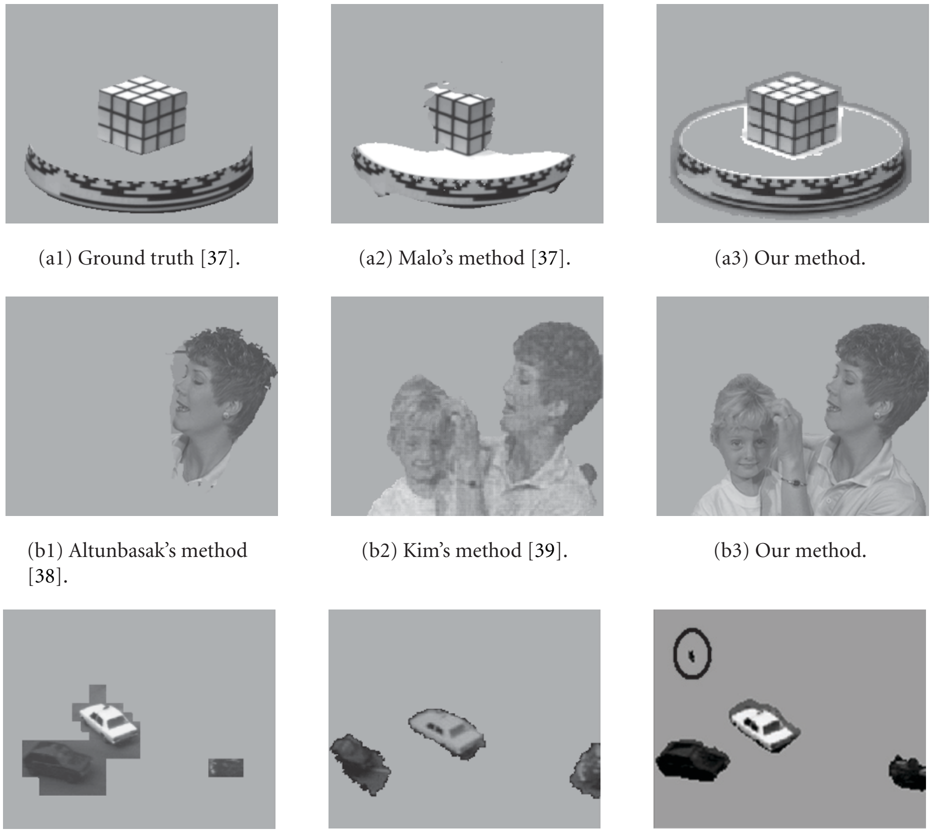
If the distance $\bar{d}_{\Omega_j}(\hat{\mathbf{P}}_{\Omega i})$ is below the predetermined threshold, the segment Ω_j will be merged with the segment Ω_i to generate a new labeled area. In order to group spatial segments into VOs more homogeneously, higher threshold (e.g., 0.5) is used for the region merging within the motion masks than the threshold used for the background (e.g., 0.1) because the motion variance in the moving VOs is much higher than that of the background, which is at most caused by the camera motion or illumination changes.

5. EXPERIMENTAL RESULTS

Experimental results for the temporal and the spatial VO segmentation are illustrated in Figures 11, 12, 13, and 14, respectively. Three kinds of image sequences are chosen for the simulation to show the potential of our method: (1) "Rubik cube," which contains two rigid rotating objects with similar speeds; (2) "Mother and Daughter," which involves two objects, one has obvious rotation movement and the other has much less movement; (3) "Taxi," which has four rigid translational moving objects with different speeds; and (4) "Silent," which is constituted by a nonrigid moving object in front of a standstill background with complicate textures.

Our spatial-constrained motion masks as illustrated in Figure 11 show that the moving objects can be separated accurately from different kinds of background. Both rigid and nonrigid motions can be captured well, using the proposed motion mask generation and selection approaches as described in Section 4.2. Thanks to the scale-adaptive 3D structure tensor computation, multiple motions are matched correctly as shown in Figure 11b, and notice that even a very small size "walking person" (highlighted by the circle) can also be extracted from the background.

Our spatial-constrained motion segmentation results are also compared with other results obtained from some existing methods [37, 38, 39, 40], as illustrated in Figures 12a, 12b, and 12c. For "Rubik cube," it is obvious that our method yields much more accurate VO segmentation as compared with that of Malo's Method [37] as shown in Figure 12a. For "Mother and Daughter", the VO "Daughter" is unable to be detected and segmented as documented in [38] and presented in Figure 12b1. The improved result has been provided by [39], where two VOs ("Mother" and "Daughter") are successfully extracted as shown in Figure 12b2. And the much improved result is obtained from our method with the best segmented boundary of VOs (see Figure 12b3) among the three methods. For "Taxi," our method presents a distinct

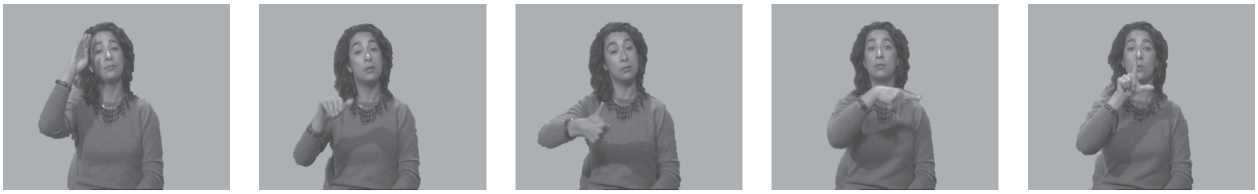


(a1) Ground truth [37]. (a2) Malo's method [37]. (a3) Our method.

(b1) Altunbasak's method [38]. (b2) Kim's method [39]. (b3) Our method.

(c1) Malo's method [37]. (c2) Bors's method [40]. (c3) Our method.

Figure 12: Segmented *rigid* moving VOs obtained by implementing the motion masks over each original frame of the three test sequences, respectively. The motion masks are yielded either from some existing methods [37, 38, 39, 40] or yielded by our proposed spatial-constrained motion segmentation approach, which have been illustrated in Figure 11 as examples.



(a) The 20th frame. (b) The 40th frame. (c) The 60th frame. (d) The 80th frame. (e) The 100th frame.

Figure 13: Segmented *nonrigid* moving VO over several frames of "Silent" sequence.

improved segmentation result on the "walking person" as circled in Figure 12c3, where other approaches failed to capture such a very small-size moving VO. Furthermore, our result yields more accurate boundaries around the taxies as compared with Malo's [37] and Bors's [40] results as illustrated in Figures 12c1 and 12c2, respectively. Notice that the so-claimed ground truth as shown in Figure 12a1 is only given as the suggested good segmentation result.

Our motion segmentation results for nonrigid moving VO are presented in Figure 13 using several frames of "Silent" sequence as an example. Obviously, our method yields good VO's boundary in accordance with human perception, and various motions of the VO have been successfully captured.

Our motion-constrained spatial segmentation results are compared with those proposed in [11] using the same input frames as illustrated in Figure 3. Figures 14a1, 14b1, and 14c1 were obtained by performing region merging through energy minimization on Horn's OF field [36]. Consequently, some small spatial segments could be merged into larger homogeneous region. However, since the global gradient-based OF estimation like Horn's algorithm is sensitive to noise and could yield inaccurate objects' boundaries, the method

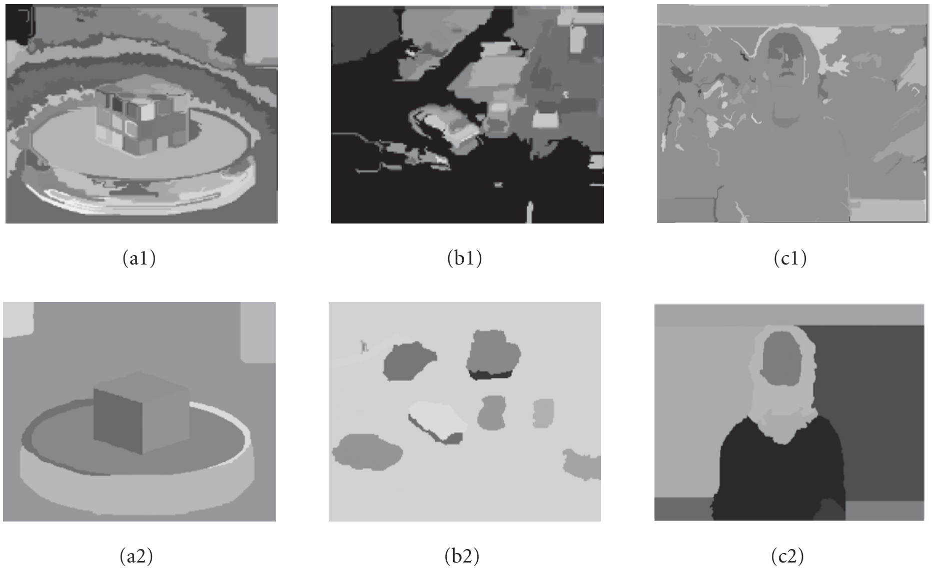


Figure 14: Motion-constrained spatial segmentation results using graph-based image segmentation (see Figure 3) as the inputs. Figures 14a1, 14b1, and 14c1 use Ross's method [11] which is based on Horn's optical flow [36], and Figures 14a2, 14b2, and 14c2 exploit our proposed method which is based on the 3D structure tensor. (a) Rubik Cub. (b) Taxi. (c) Silent.

proposed in [11] was unable to provide the whole background and homogeneous foreground objects. Using our proposed motion-constrained region merging scheme that benefits from the accuracy of spatial-constrained motion mask and the precision of affine motion model-based distance computation, the spatial VOs are segmented correctly from the complex background as illustrated in Figures 14a2, 14b2, and 14c2.

6. CONCLUSIONS

Using the duality of image and motion segmentations, a new region-based methodology using the 3D structure tensor is developed for extracting not only moving VOs constrained by spatial information, but also spatial VOs constrained by motion information; thus, both the VOs with and without motions can be segmented much more accurately in a unified framework. First, to handle the situation when multiple object motions occurred in the input image sequence, adaptive scale selection steered by the condition number is exploited for conducting the spatio-temporal Gaussian filtering. Second, rigidity analysis is performed based on the correlation coefficients of the smallest eigenvalue map computed over a range of successive frames. Third, the largest eigenvalue and the coherency measurements of the 3D structure tensor have been exploited for generating the motion masks of rigid and nonrigid VOs simultaneously. Consequently, the obtained eigenmap and corner map are selected with assistance from the change-detection map, and the boundaries of motion masks are further refined by implementing the spatial constraint. Fourth, the normalized distance measurement between the affine motion model and the 3D structure tensor is utilized in our scheme to perform motion-constrained spatial region merging via thresholding, in which different thresholds are set for various regions (e.g., moving VOs and the background), respectively. Experimental results show that the performance of our scheme is superior to that of the previous work particularly on the aspects of improving the boundary accuracy of VO segmentation as well as simultaneously handling multiple VOs with rigid or nonrigid motion.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their constructive comments.

REFERENCES

[1] J. Y. A. Wang and E. H. Adelson, "Representing moving images with layers," *IEEE Trans. Image Processing*, vol. 3, no. 5, pp. 625–638, 1994.

[2] P. Salembier, F. Marques, M. Pardas, et al., "Segmentation-based video coding system allowing the manipulation of objects," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 7, no. 1, pp. 60–74, 1997.

[3] T. Meier and K. N. Ngan, "Automatic segmentation of moving objects for video object plane generation," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 8, no. 5, pp. 525–538, 1998.

[4] J. A. Sethian, *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*, Cambridge University Press, Cambridge, UK, 1999.

[5] G. Kuhne, J. Weickert, O. Schuster, and S. Richter, "A tensor-driven active contour model for moving object segmenta-

tion," in *Proc. IEEE International Conference on Image Processing*, vol. 2, pp. 73–76, Thessaloniki, Greece, October 2001.

[6] M. M. Chang, A. M. Tekalp, and M. I. Sezan, "Simultaneous motion estimation and segmentation," *IEEE Trans. Image Processing*, vol. 6, no. 9, pp. 1326–1333, 1997.

[7] H. Gu, Y. Shirai, and M. Asada, "MDL-based segmentation and motion modeling in a long image sequence of scene with multiple independently moving objects," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 18, no. 1, pp. 58–64, 1996.

[8] M. J. Black and P. Anandan, "The robust estimation of multiple motions: parametric and piecewise-smooth flow fields," *Computer Vision and Image Understanding*, vol. 63, no. 1, pp. 75–104, 1996.

[9] J. Zhang, J. Gao, and W. Liu, "Image sequence segmentation using 3-D structure tensor and curve evolution," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 11, no. 5, pp. 629–641, 2001.

[10] G. Farneback, "Motion-based segmentation of image sequences," M.S. thesis, Linköping University, Linköping, Sweden, May 1996.

[11] M. G. Ross, "Exploiting texture-motion duality in optical flow and image segmentation," M.S. thesis, Massachusetts Institute of Technology, Cambridge, Mass, USA, August 2000.

[12] M. Sonka, V. Hlavac, and R. Boyle, *Image Processing, Analysis, and Machine Vision*, PWS Publishing, Pacific Grove, Calif, USA, 2nd edition, 1999.

[13] S. Ayer and H. S. Sawhney, "Layered representation of motion video using robust maximum likelihood estimation of mixture models and MDL encoding," in *Proc. 5th International Conference on Computer Vision*, pp. 777–784, Boston, Mass, USA, June 1995.

[14] B. A. Wandell, *Foundations of Vision*, Sinauer Press, Sunderland, Mass, USA, 1995.

[15] R. M. Haralick and L. G. Shapiro, *Computer and Robot Vision*, vol. 2, Addison-Wesley, Reading, Mass, USA, 1992.

[16] A. Mitiche and P. Bouthemy, "Computation and analysis of image motion: a synopsis of current problems and methods," *International Journal of Computer Vision*, vol. 19, no. 1, pp. 29–55, 1996.

[17] B. Jähne, H. Haussecker, and P. Geissler, *Handbook of Computer Vision and Applications*, vol. 2, pp. 309–396, Academic Press, San Diego, Calif, USA, 1999.

[18] J. L. Barron, D. J. Fleet, and S. S. Beauchemin, "Performance of optical flow techniques," *International Journal of Computer Vision*, vol. 12, no. 1, pp. 43–77, 1994.

[19] R. Plankers, "A level set approach to shape recognition," Tech. Rep., Ecole Polytechnique Fédérale de Lausanne, Lausanne, Switzerland, 1997.

[20] P. Felzenszwalb and D. Huttenlocher, "Efficiently computing a good segmentation," in *Proc. DARPA Image Understanding Workshop*, pp. 251–258, Monterey, Calif, USA, November 1998.

[21] I. T. Jolliffe, *Principle Component Analysis*, Springer-Verlag, New York, NY, USA, 1986.

[22] S. K. Mitra and G. L. Sicuranza, Eds., *Nonlinear Image Processing*, Academic Press, San Diego, Calif, USA, 2001.

[23] N. U. Ahmed, *Linear and Nonlinear Filtering for Scientists and Engineers*, World Scientific, River Edge, NJ, USA, 1998.

[24] N. Paragios and G. Tziritas, "Adaptive detection and localization of moving objects in image sequences," *Signal Processing: Image Communication*, vol. 14, no. 4, pp. 277–296, 1999.

[25] F. Bartolini, V. Cappellini, C. Colombo, and A. Mecocci, "A multiwindow least squares approach to the estimation of optical flow with discontinuities," *Optical Engineering*, vol. 32, no. 6, pp. 1250–1256, 1993.

[26] K. S. Pedersen and M. Nielsen, "Computing optic flow by scale-space integration of normal flow," in *Proc. 3rd International Scale-Space Conference*, pp. 14–25, Vancouver, British Columbia, Canada, July 2001.

[27] W. J. Niessen, J. S. Duncan, M. Nielsen, L. M. J. Florack, B. M. ter Haar Romeny, and M. A. Viergever, "Multiscale approach to image sequence analysis," *Computer Vision and Image Understanding*, vol. 65, no. 2, pp. 259–268, 1997.

[28] M. Gutierrez, M. Rebelo, and S. Furuie, "A multiresolution approach for computing myocardial motion," in *Proc. IEEE-SP International Symposium Time-Frequency and Time-Scale Analysis*, pp. 89–92, Pittsburgh, Pa, USA, October 1998.

[29] M. Middendorf and H.-H. Nagel, "Estimation and interpretation of discontinuities in optical flow fields," in *Proc. 8th IEEE International Conference Computer Vision*, vol. 1, pp. 178–183, Vancouver, British Columbia, Canada, July 2001.

[30] G. H. Golub and C. F. Van Loan, *Matrix Computations*, Jason Hopkins University Press, Baltimore, Md, USA, 1983.

[31] A. J. Lipton, "Local application of optic flow to analyze rigid versus non-rigid motion," in *Proc. ICCV Workshop on Frame-Rate Vision*, Corfu, Greece, September 1999.

[32] R. J. Larsen and M. L. Marx, *An Introduction to Mathematical Statistics and Its Applications*, Prentice-Hall, Englewood Cliffs, NJ, USA, 3rd edition, 2001.

[33] T. Aach, A. Kaup, and R. Mester, "Statistical model-based change detection in moving video," *Signal Processing*, vol. 31, no. 2, pp. 165–180, 1993.

[34] K.-K Ma and H.-Y. Wang, "Region-based nonparametric optical flow segmentation with preclustering and postclustering," in *Proc. International Conference on Multimedia and Expo*, vol. 2, pp. 201–204, Lausanne, Switzerland, August 2002.

[35] I. Pitas, *Digital Image Processing Algorithms*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1993.

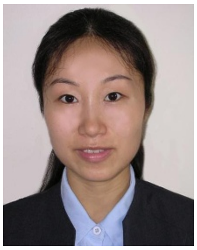
[36] B. K. P. Horn and B. G. Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 17, no. 1–3, pp. 185–203, 1981.

[37] J. Malo, J. Gutierrez, I. Epifanio, and F. J. Ferri, "Perceptually weighted optical flow for motion-based segmentation in MPEG-4 paradigm," *Electronics Letters*, vol. 36, no. 20, pp. 1693–1694, 2000.

[38] Y. Altunbasak, P. E. Eren, and A. M. Tekalp, "Region-based parametric motion segmentation using color information," *Journal of Graphical Models and Image Processing*, vol. 60, no. 1, pp. 13–23, 1998.

[39] M. Kim, J. G. Choi, D. Kim, et al., "A VOP generation tool: automatic segmentation of moving objects in image sequences based on spatio-temporal information," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 9, no. 8, pp. 1216–1226, 1999.

[40] A. G. Bors and I. Pitas, "Optical flow estimation and moving object segmentation based on median radial basis function network," *IEEE Trans. Image Processing*, vol. 7, no. 5, pp. 693–702, 1998.



Hai-Yun Wang received the B. Eng. degree and M. Eng. degree from the College of Marine Engineering, Northwestern Polytechnical University, China. She is currently pursuing the Ph.D. degree in the Division of Information Engineering, School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. Her research interests include image/video segmentation, pattern recognition, and image/video processing.

Kai-Kuang Ma received the Ph.D. degree from North Carolina State University, and the M.S. degree from Duke University, USA, both in electrical engineering, and the B.E. degree from Chung Yuan Christian University, Taiwan, in electronic engineering. He is a tenured Associate Professor in the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. His research interests are in the areas of digital signal, image and video processing. Before that, he was with the Institute of Microelectronics, National University of Singapore, working on digital video coding research. From 1984 to 1992, he was working in IBM Corporation at Kingston, New York, and then Research Triangle Park, North Carolina, engaging on various DSP and VLSI advanced product development. Dr. Ma is an Editor of the IEEE Transactions on Communications, Associate Editor of the IEEE Transactions on Multimedia and International Journal of Image and Graphics. He is an elected Technical Committee Member of the IEEE Signal Processing Society Image and Multidimensional Signal Processing (IMDSP) and the IEEE Communications Society Multimedia Communications Committee. He is the Singapore MPEG Chairman and Head of Delegation, the Chairman of IEEE Signal Processing Singapore Chapter, and Technical Program Cochair, IEEE International Conference on Image Processing (ICIP) 2004.