A Bayesian Approach to Video Object Segmentation via Merging 3D Watershed Volumes

Yu-Pao Tsai[1,3], Chih-Chuan Lai[1,2], †Yi-Ping Hung[1,2], and Zen-Chung Shih[3]

[1]Institute of Information Science, Academia Sinica, Taipei, Taiwan, R.O.C.
[2]Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan, R.O.C.
[3]Department of Computer and Information Science, National Chiao Tung University, Hsinchu, Taiwan, R.O.C.

***Abstract*—In this paper, we propose a Bayesian approach to video object segmentation. Our method consists of two stages. In the first stage, we partition the video data into a set of 3D watershed volumes, where each watershed volume is a series of corresponding 2D image regions. These 2D image regions are obtained by applying to each image frame the marker-controlled watershed segmentation, where the markers are extracted by first generating a set of initial markers via temporal tracking and then refining the markers with two shrinking schemes: the iterative adaptive erosion and the verification against a pre-simplified watershed segmentation. Next, in the second stage, we use a Markov random field to model the spatio-temporal relationship among the 3D watershed volumes that are obtained from the first stage. Then, the desired video objects can be extracted by merging watershed volumes having similar motion characteristics within a Bayesian framework. A major advantage of this method is that it can take into account the global motion information contained in each watershed volume. Our experiments have shown that the proposed method has potential for extracting moving objects from a video sequence.**

***Index Terms*—Video Object Segmentation, Watershed Segmentation, 3D Watershed Volume, Markov Random Field.**

I. Introduction

Video object segmentation plays an important role in many advanced video applications (such as in MPEG-4 [2] or in virtual reality), but still remains a challenging research topic. A popular approach to [21] video object segmentation is to combine a technique for single image segmentation with a temporal tracking procedure. Unfortunately, single image segmentation is itself a very difficult problem (which may not be easier than video object segmentation). Other techniques in [14][17] consider video sequences to be 3D signals and extend 2D methods to them, although the time axis does not play the same role as the spatial axis. The drawback of this technique is that a moving object in one frame must overlap with its corresponding object in the next frame. If the motion distance of the object is large, the object may become disconnected from one frame to the next. Most of the unsupervised segmentation algorithms only utilize low-level features such as color, texture, motion, frame difference and histogram [10][21]. However, without high-order information, semantic video object extraction is hard to achieve. Therefore, many researches have allowed a certain degree of human interaction. For example, the methods introduced in [3][5] require some human interaction for the initial segmentation of the first image in the video. In fact, almost all the automatic algorithms developed for extracting video objects have some limitations. For example, the automatic method proposed in [21] can only extract homogeneous regions, instead of complete objects.

Realizing that there exists no generic automatic algorithm applicable to all kinds of video sequences, we focus on the problem of extracting video objects having similar motion characteristic. The method proposed in this paper consists of two stages: (1) generation of 3D watershed volumes, and (2) Bayesian merging of 3D watershed volumes. Details of the two stages will be

described in Section II and Section III. Experimental results will be shown in Section IV, and the conclusion will be given in Section V.

II. Generation of 3D watershed volumes

Watershed algorithm has been become popular technique for image segmentation [6][17][19]. Given a video clip, $\{I_t, 0 \le t \le T\}$, we can regard the data as one volume image. Our method first partitions the volume image into a set of *3D watershed volumes*, where each 3D watershed volume is a series of corresponding 2D image regions. Fig. 1 shows the flowchart of our method for generating 3D watershed volumes. These 2D image regions are obtained by applying to each image frame the *marker-controlled watershed segmentation* described in Step 2 of Section II-B. The procedure for generating 3D watershed volumes can be divided into two phases: initial segmentation and temporal tracking. Details of these two phases are described below.

A. Initial Segmentation

In the initial phase, the first frame of the video clip, I_0, is partitioned into a set of 2D regions by applying the *watershed segmentation* algorithm to the gradient image of I_0. However, the basic watershed transformation tends to produce over-segmentation due to noise or local irregularities in the gradient image. Since overly segmented regions may not be reliable enough for the next phase of temporal tracking, we adopt a pre-processing method called "topographic simplification" to alleviate the over-segmentation problem. In our current implementation, the topographic surface of the gradient image is simplified by removing the local minima [20]. First, we apply a dilation operation with a structuring element of 2x2 pixels, i.e., let $g_1 = Grad(I_0) \oplus B_{2\times2}$. Next, we apply to $Grad(I_0)$ a "reconstruction by erosion" [18] from g_1+h, i.e., let $g_2 = \varphi^{(rec)}[g_1+h, Grad(I_0)]$. Notice that using a larger h can eliminate more local minima. Finally, we can obtain a reasonable segmentation of I_0 by applying the basic watershed transformation to the simplified gradient image, g_2.

In this paper, the above procedure of "*topographic simplification followed by watershed transformation*" will be referred to as the *pre-simplified watershed segmentation*, and will be applied again to each subsequent frame for the purpose of refining the extracted markers, as described in Step 1.3.

After pre-simplified watershed segmentation, merging of a foreground region and a background region may occasionally occur. That means the volume of parameter h is too large so that watershed regions are over-simplified. The user can select either a smaller h or apply some human intervention supported by our system. Our tool allows the user to draw different markers on some parts of the region to indicate that they should not be merged. Then, the marker-controlled watershed segmentation will be performed so that the merged region will be split automatically. The operation is quit simple for the users, and this operation, if needed, usually is required only for the first frame. Fig. 2 shows an example of our user intervention tool. The edge of the hat and the background are separated by drawing different markers on each of them, as shown in Fig. 2(a). Fig. 2(b) and 2(c) show the results after user intervention.

B. Temporal Tracking

In the second phase, our algorithm repeats the following two steps for each subsequent frame in the video clip: (i) marker extraction, (ii) marker-controlled watershed segmentation. The task of marker extraction is to extract reliable seed regions based on the segmented regions obtained from the previous frame. Given these reliable markers, the marker-controlled watershed segmentation can not only accurately extract the boundaries of the watershed regions, but also can detect newly emerging regions.

Step1: Marker Extraction

Marker extraction is crucial to the success of the temporal tracking phase and deserves some special attention here. Our method

for extracting markers consists of the following three sub-steps:

Step 1.1: Region label propagation by motion-based backward projection

First, initial markers are obtained by using backward pixel projection based on backward motion vectors. That is, for each pixel *p* in the current frame, we assign to the region label of the corresponding pixel in the previous frame to it. The correspondence is determined by using the backward motion vector $\mathbf{m}_p$. Here, we choose to use backward motion to avoid generating empty and conflicting areas in the current frame. The dense field of backward motion vectors is estimated by using a template-matching algorithm that adopts adaptive windows, similar to the one used in [7]. To save the computation time, we first estimate a sparse field of motion vectors at every 4×4 pixel spacing. Then, the dense pixel-wise motion vectors are computed using bilinear interpolation. The approximation error can be dealt with the following process.

Step 1.2: Removing unreliable pixels from initial markers by iterative adaptive erosion

Since motion vectors are usually not very accurate, we must remove unreliable region assignments due to erroneous pixel correspondences. In order to reduce the possibility of generating false boundaries in the next sub-step, the extracted markers should be as large as possible, and completely contained in their *true* corresponding regions - which are unfortunately unknown to the computer.

Consider an initial marker M_i. A pixel $\mathbf{p} \in M_i$, is regarded as an unreliable pixel if it has an unreliable region propagation, that is, if $\bar{\varepsilon}(\mathbf{p})$ is greater than $k \cdot \overline{E_i}$, where $\bar{\varepsilon}(\mathbf{p})$ denotes the local mean of *textural error* centered round pixel $\mathbf{p}$ (that is, the error of texture, including intensity and color, between the corresponding pixels):

$$\bar{\varepsilon}(\mathbf{p}) = \frac{1}{N_{U_\mathbf{p}}} \sum_{\mathbf{p}' \in U_\mathbf{p}} \left| I_t(\mathbf{p}') - I_{t-1}(\mathbf{p}' + \mathbf{m}_{\mathbf{p}'}) \right| \tag{1}$$

where $U_\mathbf{p}$= { $\mathbf{p}$ and its 8-neighbors having the same region label as $\mathbf{p}$} , $N_{U_\mathbf{p}}$ is the number of elements in the set $U_\mathbf{p}$, and $\overline{E_i}$ denotes the global mean of *textural error* for the whole area of marker M_i:

$$\overline{E_i} = \min\left(\max\left(\frac{1}{N_{M_i}} \sum_{\mathbf{p} \in M_i} \left| I_t(\mathbf{p}) - I_{t-1}(\mathbf{p} + \mathbf{m}_\mathbf{p}) \right|, 2 \right), 16 \right) \tag{2}$$

where N_{M_i} is the number of the pixels in marker M_i. The reason for constraining $\overline{E_i}$ to 2 and 16 is to prevent using an unreasonable large or unreasonable small threshold. The two number, 2 and 16, are determined according to our experiments.

In this sub-step, we apply an iterative adaptive erosion to trim off "unreliable border pixels" of the initial markers, as illustrated in Fig. 3. The adaptive erosion (*erode if "unreliable"*) is performed iteratively with a cross-shaped structuring element of 5 pixels, shown in Fig. 3(b), until the result becomes stable. Notice that the adaptively eroded marker shown in Fig. 3(e) is a union of the normally eroded marker (shown in Fig. 3 (d)) and the reliable pixels, coloured in white, are contained in the border portions (shown in Fig. 3(c)).

Note that using a lower *k* can eliminate more marker pixels. In the case of foreground and background objects, which are not distinctive, *k* should be set conservatively. We found that *k* = 1.2 works well for most MPEG-4 test sequences in hand. The resulting markers with different values of *k* using frame 116 of the "foreman" sequence are shown in Fig. 4. Pixels in black represent any undefined areas.

Step 1.3: Removing unreliable pixels by checking with a pre-simplified watershed segmentation

Here, we first generated a reasonably fine segmentation of the current frame by applying the *pre-simplified watershed*

segmentation described in Section II-A, with a small value of parameter *h*. For each generated watershed region, check if it contains only one marker and the sole marker occupies more than half of the watershed region. If so, the sole major marker will be retained for driving the marker-controlled watershed segmentation in the next step. Otherwise, the marker pixel in this watershed region will be considered "unreliable", and will be removed from the markers, as illustrated in Fig. 5. Fig. 6 shows the final markers obtained by applying this sub-step to the markers shown on Fig. 4. We can see that after this step, small and ambiguous pieces of the marker are removed.

Step 2. Marker-controlled watershed segmentation

Based on the reliable markers obtained from the last step, we can then extract more precise region boundaries by using the *marker-controlled watershed segmentation* [9][21]. One problem accompanying marker-controlled segmentation is that no newly exposed regions can be extracted without creating new markers. To solve this problem, we modify the marker-controlled watershed algorithm slightly. For the flooding process of the marker-controlled watershed algorithm used in [21], when the water coming from two different basins is about to meet, the two basins are merged if "both have the same label" or "at least one of them is unlabeled." Our modification for creating new markers is if the dynamics of an unlabeled basin larger than a certain threshold [11][8], the basin will be given a new label (Fig. 7). Fig. 8 shows the result of detecting new regions using frame 26 and 27 of the "coastguard" sequence. The big boat is entering the image from the left, and the background water can be detected as a new region.

III. BAYESIAN MERGING OF WATERSHED VOLUMES

Once the 3D watershed volumes are generated, as described in Section 2, we need to merge them into a set of desired video objects. Here, we propose a Bayesian approach to merging watershed volumes having similar motion characteristics, hoping that more global motion information can be utilized within a formal framework. Here, we use a Markov random field (MRF) to model the spatial and temporal relationships among different watershed volumes. A closely related work is the one done by Gelgon and Bouthemy, which uses region-level MRFs to track a spatial image partition [4]. Another work proposed by Patras *et al.* [14] labels watershed segments by MAP. The labeling criterion is the maximization of the conditional *a posteriori* probability of the labeling field given the motion hypothesis, the estimate of the label field of the previous frame, and image intensities. However, our method is different from theirs, not only in how the MRF is applied (we employ the MRF after tracking while they do it before tracking), but also in how the class-conditional probability is modeled.

A. Extraction of Features from 3D Watershed Volume

Before applying the Bayesian merging to 3D watershed volumes, the representative features for each watershed volume need to be extracted. Motion information is an important cue to produce semantic objects. Hence, for each watershed volume v, we construct a feature vector $\boldsymbol{\theta}_v$ based on motion information. We first decompose each watershed volume v into a set of regions $\left\{R_v^t \mid 0 \le t_b(v) \le t \le t_e(v) \le T\right\}$, where R_v^t denotes a region which can be obtained by intersecting frame t with the watershed volume v, $t_b(v)$ and $t_e(v)$ are the indices of the beginning frame and the ending frame of the watershed volume v, respectively. Note that the indices of the beginning and ending frames of the watershed volumes can vary for the watershed volume v due to the appearance or disappearance of objects in the scene.

In practical situations, image motion of a rigid object can be approximately modeled by a small number of motion parameters. If two regions roughly correspond to the same 3D rigid object, the motion parameter should be about the same. From the above observation, we compute a motion parameter vector $\boldsymbol{\theta}_v^t$ for each region R_v^t by applying the Least-Median Squares (LMedS) robust estimator [15] to the backward dense motion field obtained from Step 1.1 of Section II-B. The motion parameters can be

estimated by

$$\hat{\boldsymbol{\theta}}_v^t = \arg \min_{\boldsymbol{\theta}_v^t} \left\{ \underset{\mathbf{p} \in R_v^t}{median} \left\| \mathbf{m}_{\mathbf{p}}^t - \mathbf{u}(\mathbf{p}; \boldsymbol{\theta}_v^t) \right\| \right\} \tag{3}$$

where $\mathbf{u}(.)$ is a parameterized motion field, $\|\cdot\|$ is defined as two-norm, and $\mathbf{m}_p^t$ is the motion vector of pixel p in frame t. After the parameters for all the regions in the watershed volume v are determined, we can construct a motion feature vector: $\boldsymbol{\theta}_v = [\theta_v^{t_b(v)}, \theta_v^{t_b(v)+1}, \ldots, \theta_v^{t_e(v)}]$. Notice that the dimensionality of $\boldsymbol{\theta}_v$ is $(t_e(v)-t_b(v)+1)\cdot d$, where d is the dimension of $\boldsymbol{\theta}^t$. In our current implementation, the motion characteristics of R_v^t are described by a constant motion field, that is, $\mathbf{u}(\mathbf{p}; \boldsymbol{\theta}_v^t) = \boldsymbol{\theta}_v^t$, where $\boldsymbol{\theta}_v^t = [m_x^t, m_y^t]$ and m_x^t and m_y^t are the coordinates of the mean motion vector. If an object undergoes a complex motion or deformation, a more complex motion model, such as a six-parameter affine model or eight-parameter quadratic model, should be used to enhance discriminative ability [12]. Once a complex motion model, such as a six-parameter affine model, is adopted, the equations presented in next section should be modified slightly.

B. The Proposed Method

In this work, we assume that the number of video objects, N, to be extracted (including the background objects) is known. Given a set of 3D watershed volumes $V = \{v_j, j=1,\ldots,K\}$, where K is the number of 3D watershed volumes, a Volume Adjacency Graph (VAG) can be constructed to express the neighborhood relationship among 3D watershed volumes. Each node in the graph corresponds to a watershed volume, and between two volumes exists an arc if the volumes are spatially connected. Next, we define a label field $L = \{ l_v \mid l_v \in [1..N], v \in \mathrm{V} \}$ on the VAG. Given $M = \{\boldsymbol{\theta}_v \mid v \in V\}$, we estimate the labeling field L by maximizing the *a posteriori* probability (MAP). Using the Bayes rule, the *a posteriori* probability density function can be expressed as:

$$P(L \mid M) \propto P(M \mid L) \cdot P(L) \tag{4}$$

The first term on the right-hand side of (4) is the conditional probability distribution $P(M \mid L)$. It is modeled as a Gaussian distribution, which implies that each object should have minimum motion variance.

$$P(M \mid L) \propto \exp\left(- \sum_{v \in V} \frac{1}{2\sigma_l{}^2} \sum_{t=t_b(v)}^{t_e(v)} \left\| \boldsymbol{\theta}_v^t - \boldsymbol{\mu}^t(l_v) \right\|^2 \right) \tag{5}$$

where $\boldsymbol{\mu}^t(l_v)$ is the mean of the parameter vectors of all watershed volumes in frame t whose corresponding labels are l_v, σ_l is a function of the size of the video object.

The second term on the right-hand side of (4) is the prior probability distribution $P(L)$, which is a regularization term. To take into account the "degree" of adjacency between two watershed volumes, we directly extend a measure of adjacency degree between two regions proposed in [4] to that between two watershed volumes:

$$b(v_j, v_k) = \frac{\ell_{v_j, v_k}}{\ell_{v_j, v_k} + \left\| g_j - g_k \right\|} \tag{6}$$

where ℓ_{v_j, v_k} is the area of the common border between v_i and v_j, and g_j and g_k are the gravity centers of v_j and v_k, respectively. We model the prior as a Gibbs distribution. Before defining a Gibbs distribution, we need to define the cliques. Here, only two-site cliques are considered and straightforwardly obtained from the arc of the VAG. Let $\boldsymbol{C}_v$ be the set of all binary cliques. The Gibbs distribution is given by

$$P(L) = \frac{1}{Z_b} \exp\left(-U_b(L)\right) \tag{7}$$

where Z_b is a normalizing constant and $U_b(L)$, the regularization potential, is defined as

$$U_b(L) = \sum_{(v_j, v_k) \in \mathcal{C}_v} -b(v_j, v_k) \cdot \delta(l_{v_j}, l_{v_k}) \tag{8}$$

where $\delta(.)$ is a Kronecker delta function. The regularization term tends to favor identical labels for two neighboring volume sites.

The maximum a posteriori probability (MAP) estimate of L is obtained by minimizing the following energy function:

$$\hat{L} = \arg\min_{L} \left(\sum_{v \in V} \frac{1}{2\sigma^2} \sum_{t=t_b(v)}^{t_e(v)} \left\| \boldsymbol{\theta}_v^t - \boldsymbol{\mu}^t(l_v) \right\|^2 + \sum_{(v_j, v_k) \in \mathcal{C}_v} -b(v_j, v_k) \cdot \delta(l_{v_j}, l_{v_k}) \right) \tag{9}$$

Energy minimization is performed using an ICM algorithm proposed by Besag [1], sometimes also called the greedy algorithm. At each iteration, each volume sites is visited. The label of each site is either changed to the label that yields maximal decrease of the energy function, or left unchanged if no energy reduction is possible. The process stops when no more changes can be made. The initialization of the label L is estimated by the K-means algorithm. The initial cluster means $\{\boldsymbol{\mu}(l) \mid 1 \le l \le N\}$ for the K-means algorithm are estimated as follows. The first cluster mean is the mean of the total motion parameters. That is,

$$\boldsymbol{\mu}(1) = (\boldsymbol{\mu}^0(1), \boldsymbol{\mu}^1(1), .., \boldsymbol{\mu}^T(1)); \qquad \boldsymbol{\mu}^t(l) = \frac{1}{N_t} \left(\sum_{v \in V} \theta_v^t \right) \tag{10}$$

where N_t is the number of the watershed volume v intersecting with frame t. The c^{th}-cluster mean is the feature vector $\boldsymbol{\theta}_v$ that has the farthest distance from the nearest cluster mean

$$\boldsymbol{\mu}(c) = \arg_{\boldsymbol{\theta}_v} \left\{ \max_{1 \le c' \le c-1} \left[\min_{v \in V} \left(\frac{1}{t_e(v) - t_b(v) + 1} \sum_{t=t_b(v)}^{t_e(v)} \left\| \theta_v^t - \boldsymbol{\mu}^t(c') \right\| \right) \right] \right\} \tag{11}$$

In summary, the algorithm of our method for merging watershed volumes into video objects can be described as follows:

Input: Volume Adjacency Graph (VAG), K

1. Obtain initial cluster means for K-means algorithm using equations (10) and (11).
2. Obtain initial label for each watershed volume by applying K-means algorithm.
3. Update labels for all volumes by applying ICM algorithm based on equation (9).

Output: Labels of all watershed volumes

IV. Experimental results

In this section, we use the "foreman", and "coastguard" sequences, shown in Fig. 9, and Fig. 10, respectively, to demonstrate the performance of our algorithm. In our current implementation, the gradient images are computed on a weighted YUV colour space, i.e. $w_y Y + w_u U + w_v V$. The weighting factors, w_y, w_u, and w_v, are set to one, two, and two, respectively, to stress the color components. The experiments are run in AMD Athlon 1.2GHz PC with 384MB RAM. The sizes of the "foreman" sequence and the "coastguard" sequence are 352x288 and 352x240. The total execution time of the "foreman" sequence (100 frames) is 483sec, and the "coastguard" sequence (50 frames) is 131sec. For the experimental results presented in this paper, no user intervention has been used. However, one can always find some video sequences that contain complex enough scene, such that user intervention may become necessary.

In the "foreman" sequence, the human body has a moderate motion and the camera is moving as well. It can be seen from Fig. 9(b), where cross-sections of watershed volumes are shown, that the results obtained by marker-controlled temporal tracking look pretty good. By setting $N = 2$ (i.e., the number of video objects to be extracted is 2), the watershed volumes depicted in Fig. 9(b) can be correctly merged into two video objects: the foreman and the background, as shown in Fig. 9(c). In this sequence, we have found that the similarity between the motions of the head and the shoulder could be more easily detected when considering a longer sequence. Therefore, our method can obtain better segmentation results than those obtained by Moschni et al. [10].

In the "coastguard" sequence, the horizontal camera drift is present while two boats are moving with different velocities and directions. Notice that the bigger boat is entering the image from the left, and its new emerging regions can be successfully extracted, as shown in Fig. 10(b). If we set $N = 4$, the proposed Bayesian method can partition the video clip into four different objects: the bigger boat, the smaller boat, the water and the shore, as shown in Fig. 10(c). Compared with the results using the method proposed by Patras et al.[14], the segmented boundaries we extracted are much closer to the objects.

V. Conclusion

In this paper, we have proposed a new method for video object segmentation. This method first partitions the video data into a set of 3D watershed volumes, and then extracts video objects by merging motion-coherent watershed volumes within a Bayesian framework. One major contribution of this work is that it models the prior information with a MRF over a Volume Adjacency Graph (VAG), where each node of the VAG is a 3D watershed volume, and hence, is able to take into account the global motion information contained in each watershed volume. This method is appropriate for extracting objects having similar motion because it can merge 3D watershed volumes having similar motion with a Bayesian framework. Another contribution is that this paper proposes an efficient way to extract reliable markers by shrinking with two schemes: the iterative adaptive erosion and the verification against a pre-simplified watershed segmentation. Experimental results have shown that the proposed method has potential for extracting moving objects from a video sequence.

References

[1] J. Besag, "On the statistical analysis of dirty pictures," *J. R. Stat. Soc. B* , 48(3):259-302, 1986.

[2] P. Daras, I. Kompatsiaris, I. Grinias, G. Akrivas, G. Tziritas, S. Kollias and M.G. Strintzis: "MPEG-4 Authoring Tool using Moving Object Segmentation and Tracking in Video Shots", *EURASIP J. on Applied Signal Processing*, vol. 2003, no. 9, pp. 861-877, August 2003.

[3] D. Gatica-Perez, G. Gu and Ming-Ting Sun, "Semantic Video Object Extraction Using Four-Band Watershed and Partition Lattice operators," *IEEE Trans Circuit Systems Video Technology,* 11(5):603-618, 2001.

[4] M. Gelgon and P. Bouthemy, "A Region-Level Motion-Based Graph Representation and Labeling for Tracking a Spatial Image Partition," *Pattern Recognition*, 30(4):725-740, 2000.

[5] C. Gu and M. Lee, "Semiautomatic Segmentation and Tracking of Semantic Video Objects," *IEEE Trans. Circuit Systems Video Technology,* 8(5):572-584, 1998.

[6] K. Haris, S. N. Efstratiadis, N. Maglaveras, and A. K. Katsaggelos, "Hybrid Image Segmentation Using Watersheds and Fast Region Merging," *IEEE Trans. Image Processing,* 7(12):1684-1699, 1998.

[7] T. Kanade and M. Okutomi, "A Stereo Matching Algorithm with an Adaptive Window: Theory and Experiment," *IEEE Trans. Pattern Analysis and Machine Intell.*, 16(9):920-932, 1994.

[8] C. Lemarechal, and Fjortoft, "Comments on "Geodesic Saliency of Watershed Contours and Hierarchical Segmentation," *IEEE Trans. Pattern Analysis and Machine Intell.*, 20(7):762-763, 1998.

[9] F. Meyer and S. Beucher, "Morphological Segmentation," *J. Visual Commu. Image Representation*, 1:21-46, 1990.

[10] F. Moscheni, S. Bhattacharjee, and M. Kunt, "Spatiotemporal Segmentation Based on Region Merging," *IEEE Trans. Pattern Analysis and Machine Intell.*, 20(8):897-915, 1998.

[11] L. Najman, and M. Schmitt, "Geodesic Saliencey of Watershed Contours and hierarchical Segmenation," *IEEE Trans. Pattern Analysis and Machine Intell.*, 18(12):1163-1173, 1996.

[12] H. T. Nguyen, M. Worring and A. Dev, "Detection of Moving Objects in Video Using a Robust Motion Similarity Measure," *IEEE Trans. on Image Processing*, 9(1):137-141, January 2000.

[13] M. Pardas, P. Salembier, "3D Morphological Segmentation and Motion Estimation for Image Sequences," *Signal Processing*, 38, pp. 31-43, 1994.
[14] I. Patras, E. A. Hendriks and R. L. Lagendijk, "Video Segmentation by MAP Labeling of Watershed Segments," *IEEE Trans. on Pattern Analysis and Machine Intell.*, 23(3):326-332, March 2001.
[15] P.J. Rousseeuw, "Least Median of Squares Regression," *J. Amer. Statist. Assoc.*, 79, pp.871-880, 1984.
[16] J. B.T.M. Roerdink, and A. Meijster, "The Watershed Transform: Definitions, Algorithms and Parallelization Strategies," *Fundamenta Informaticae*, 41:187-228, 2000.
[17] P. Salembier and M. Pardas, "Hierarchical Morphological Segmentation for Image Sequence Coding," *IEEE Trans. Image Processing*, 3(5):639-651, September 1994.
[18] L. Vincent, "Morphological Grayscale Reconstruction in Image Analysis: Application and Efficient Algorithms," *IEEE Trans. on Image Processing,* 2:176-201, 1993.
[19] L. Vincent and P. Soille, "Watersheds in Digital Spaces: An Efficient Algorithm Based on Immersion Simulations". *IEEE. Trans. Pattern Analysis and Machine Intell.*, 13(6): 583-598, 1991.
[20] D. Wang, "A Multiscale Gradient Algorithm for Image Segmentation Using Watersheds". *Pattern Recognition*, 30(12):2043-2052, 1997.
[21] D. Wang, "Unsupervised Video Segmentation Based on Watersheds and Temporal Tracking," *IEEE Trans. Circuit Systems Video Technology,* 8(5):539-546, 1998

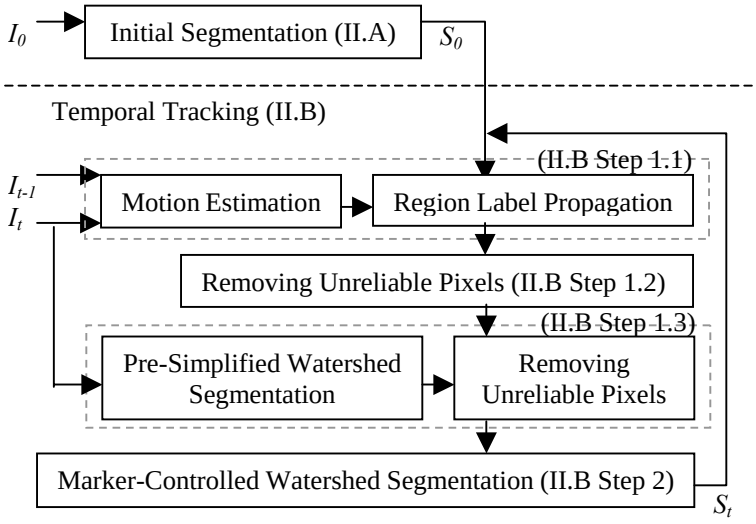
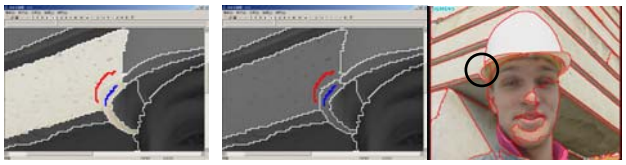
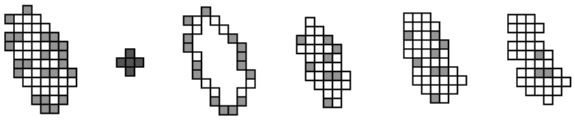


Fig. 1. Follow chat of generating 3D watershed volumes.



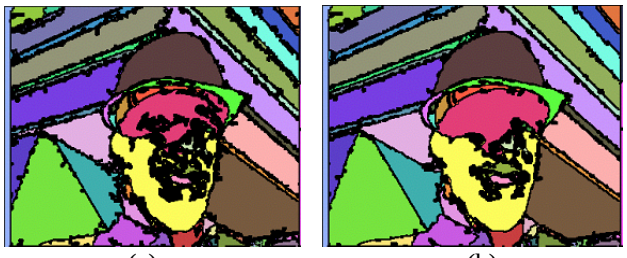
(a) (b) (c)

Fig 2. Example of user assistance

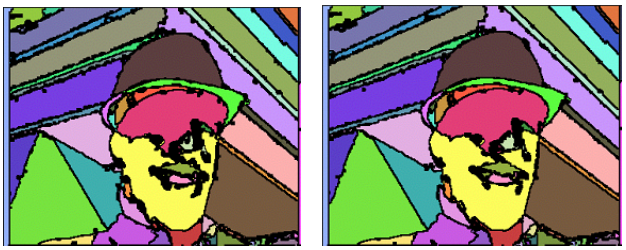


(a) (b) (c) (d) (e) (f)

Fig. 3. An example of Step 1.2 for marker extraction. (a) An initial marker with unreliable pixels coloured in grey. (b) A cross-shaped structuring element of 5 pixels. (c) Border pixels removed with the normal erosion. (d) Interior pixels obtained with the normal erosion. (e) The eroded marker after the first iteration of adaptive erosion. (f) After the second iteration (stable and stopped).

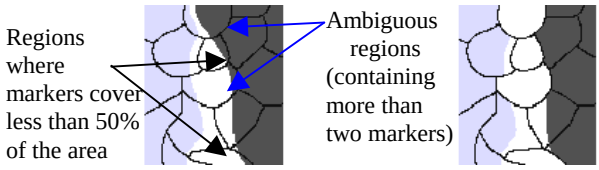


(a) (b)



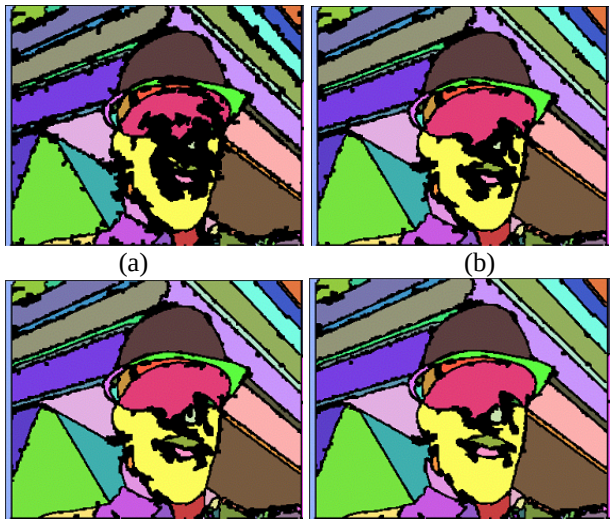
(c) (d)

Fig. 4. Markers extracted from frame 116 of sequence "foreman" with different the value of k after Step 1.2 for marker extraction. (a) k = 0.8. (b) k = 1.0. (b) k = 1.2. (b) k = 1.4.



(a) (b)

Fig. 5. An example of Step 1.3 for marker extraction. (a) Two different markers are overlaid by the watershed lines obtained from pre-simplified watershed segmentation. (b) The shrunk marker after removing the doubtful portions.



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

Fig. 6. Markers extracted from frame 116 of sequence "foreman" with different the value of k after Step 1.3 for marker extraction. (a) k = 0.8. (b) k = 1.0. (b) k = 1.2. (b) k = 1.4.

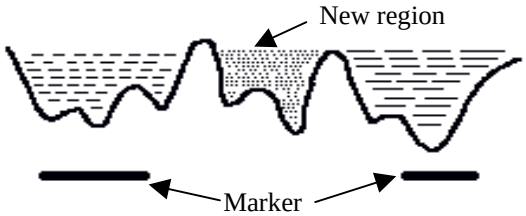
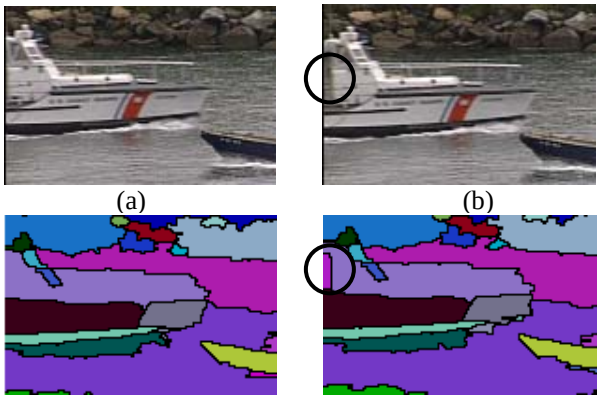


Fig. 7. A new region is labelled if the dynamics of a catchment basin exceeds a certain threshold.



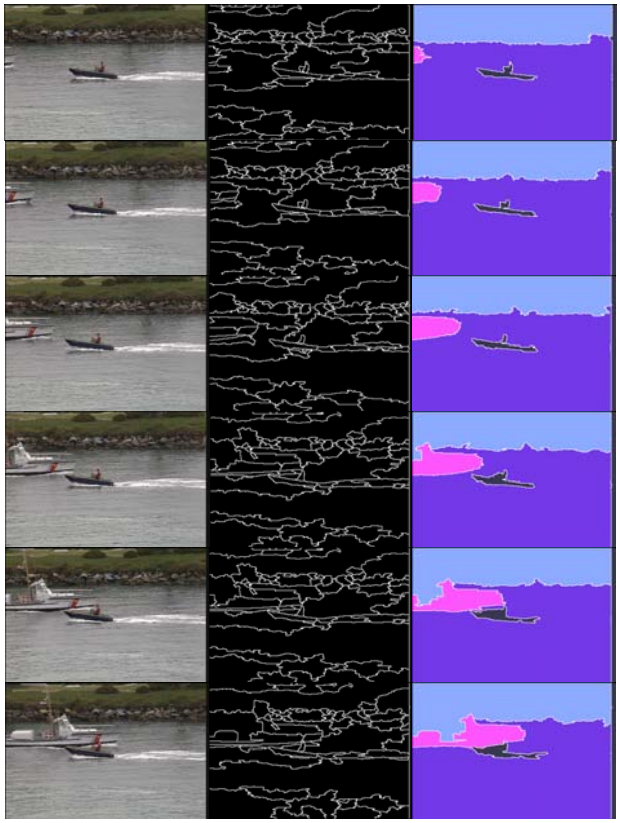
(a) (b)

(c) (d)

Fig. 8. A demonstration of detecting new region by using dynamic thresholding. (a) frame 26. (b) frame 27. (c) segmented result of frame 26. (d) segmented result of frame 27.

(a) (b) (c)

Fig. 9. “Foreman” sequence: Frame 1, 20, 40, 60, 80, 100. (a) Original images. (b) After temporal tracking. (c) After Bayesian merging.



(a) (b) (c)

Fig.10. “Coastguard” sequence: Frame 1, 10, 20, 30, 40, 50. (a) Original Images. (b) After temporal tracking. (c) After Bayesian merging.