

Predictive Watershed: A Fast Watershed Algorithm for Video Segmentation

Shao-Yi Chien, Yu-Wen Huang, and Liang-Gee Chen

Abstract—Watershed transform is a key operator in video segmentation algorithms. However, the computation load of watershed transform is too large for real-time applications. In this paper, a new fast watershed algorithm, named P-Watershed, for image sequence segmentation is proposed. By utilizing the temporal coherence property of the video signal, this algorithm updates watersheds instead of searching watersheds in every frame, which can avoid a lot of redundant computation. The watershed process can be accelerated, and the segmentation results are almost the same as those of conventional algorithms. Moreover, an intra-inter watershed scheme (IP-Watershed) is also proposed to further improve the results. Experimental results show that this algorithm can save 20%–50% computation without degrading the segmentation results. This algorithm can be combined with any video segmentation algorithm to give more precise segmentation results. An example is also shown by combining a background registration and change-detection-based segmentation algorithm with P-Watershed. This new video segmentation algorithm can give accurate object masks with acceptable computation complexity.

Index Terms—Fast algorithm, image segmentation, predictive watershed, video segmentation, watershed.

I. INTRODUCTION

WATERSHED transform, which can separate an image into many homogeneous nonoverlapped closed regions, has been widely applied in image segmentation algorithms. It is also applied to image sequences as a core operator of video segmentation, which is a key technique in MPEG-4 content-based encoding systems [1]. Video segmentation algorithms with watershed transform [2]–[6] are taken as mainstream since they can generate object masks with accurate boundaries.

Many watershed algorithms have been proposed [7]–[11]. Vincent and Soille proposed a watershed algorithm using immersion simulations [7]. With sorting before the flooding process and with priority queue, this algorithm is dramatically faster than any former ones. Beucher and Meyer's algorithm also uses immersion simulations [8], [9]. Two types of algorithms are included: one creates watershed pixels and the other produces a complete tessellation of an image. An ordered queue is used in this algorithm, whose concept is similar to that of Vincent and Soille's algorithm; however, the minima of the input image need to be detected and labeled first, thus increases the complexity of this algorithm. Dobrin *et al.* proposed a fast watershed algorithm named split-and-merge algorithm [10]. It can solve the isolated area problems of the former two

algorithms when they are employed to create watershed pixels. Although the results obtained are more correct, it is more complex than the other two algorithms. Moreover, watershed transform for video segmentation is often required to produce a tessellation of an image, where the isolated area problem would not occur. Moga *et al.* proposed a watershed algorithm suitable for parallel implementation [11]. With parallel computation, the watershed algorithm can be further accelerated. However, it is also complex and requires a powerful platform, which is impractical for general cases.

Although Vincent and Soille's algorithm provides fast computation, it is still not fast enough for real-time video segmentation. Moreover, video segmentation is different from image segmentation. The temporal information is important in video signal processing. Applying image-signal-processing algorithms on each frame for the video signal will lead to redundant computation.

This paper proposes a predictive watershed algorithm (P-Watershed) for image sequence segmentation on the basis that watersheds are highly related to original frame data. It "updates" watersheds instead of recalculating watershed transform frame by frame. The watershed process can be accelerated, and the results are almost the same as those of conventional watershed algorithms. Furthermore, the segmentation results can be further improved with the intra-inter watershed scheme (IP-Watershed).

The concepts of watershed and Vincent and Soille's algorithm will be introduced in Section II. Section III and Section IV describe the proposed algorithm and the simulation results, respectively. In Section V, predictive watershed is employed with a prior video segmentation algorithm to improve the performance. Finally, Section VI provides a conclusion.

II. WATERSHED

This section introduces the concept and a conventional algorithm of watershed. First, the concept of morphological watershed and immersion simulation are shown. Then the famous Vincent and Soille's algorithm is presented, which is an important benchmark algorithm to be compared with the proposed algorithm.

A. Morphological Watershed

Watershed transform is originally developed in the field of topography, and it is found to be useful in digital image processing. The first step of morphological watershed is morphological gradient. The gradient level of an image is then seen as the altitude level to form a topology surface. Water flooded into this surface will flow to the lower parts of this surface because of gravity. As shown in Fig. 1, water will flow to the minimum of each *catchment basin*. If we give the water in each *catchment basin* a special label and keep flooding water, water from

Manuscript received May 21, 2002; revised August 5, 2002. This work was supported in part by SiS Education Foundation. This paper was recommended by Associate Editor H. Sun.

The authors are with the DSP/IC Design Lab, Graduate Institute of Electronics Engineering, Department of Electrical Engineering, National Taiwan University, Taipei 106, Taiwan, R.O.C. (e-mail: shoayi@video.ee.ntu.edu.tw; yuwen@video.ee.ntu.edu.tw; lgchen@video.ee.ntu.edu.tw).

Digital Object Identifier 10.1109/TCSVT.2003.811605

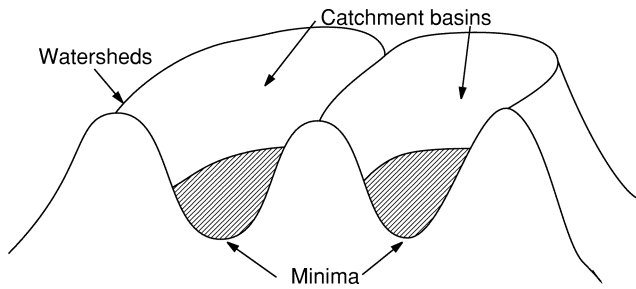


Fig. 1. Illustration of watersheds.

different *catchment basins* will interflow at *watersheds*. *Watersheds*, which correspond to parts of an image with a high gradient level, are then detected, and they can be used to separate an image into many homogeneous closed regions.

The flooding process is often done with two mechanisms: immersion or rainfall. The immersion simulation is more popular and is described below. First, we pierce holes at the minimum of each *catchment basin*, and water is flooded from these holes to form small lakes. When water from different *catchment basins* is going to interflow, a “dam” is built. After the whole surface is flooded, these dams correspond to the *watersheds* of this image.

B. Vincent and Soille’s Algorithm

Vincent and Soille’s algorithm [7] is the most famous and is deemed as the fastest watershed algorithm [12]. This algorithm can be briefly illustrated in Fig. 2 and by the following steps.

- Step 1) Morphological gradient is applied in the whole frame to find the gradient value of every pixel.
- Step 2) All the pixels are first sorted according to their gradient level. Pixels with lower gradient level will be manipulated first.
- Step 3) As shown in Fig. 2(a), pixels with the lowest gradient level of an image are first found, and pixels of each connected region are given a special label.
- Step 4) Pixels are then processed from the lower to higher gradient level. In each level, pixels with labeled neighbors are first added to the priority queue, as the two white balls shown in Fig. 2(b). The label of the pixel at the beginning of the queue is determined by its labeled neighbors, and the pixel is then removed from the queue. After all pixels with labeled neighbors are processed, pixels without labeled neighbors are found and given a new label, as the black ball shown in Fig. 2(b).
- Step 5) After all pixels are labeled, the boundaries between regions with different labels are the watersheds of this image. In Fig. 2(c), the surface is divided into four regions, and three watersheds are detected.

Although this algorithm is efficient since each pixel of the image is accessed only once in the flooding process, the computation load is still large. For video segmentation, large temporal information redundancy exists. Therefore, applying a morphological gradient filter, address sorting, and flooding process on every pixel every frame introduces large redundant computa-

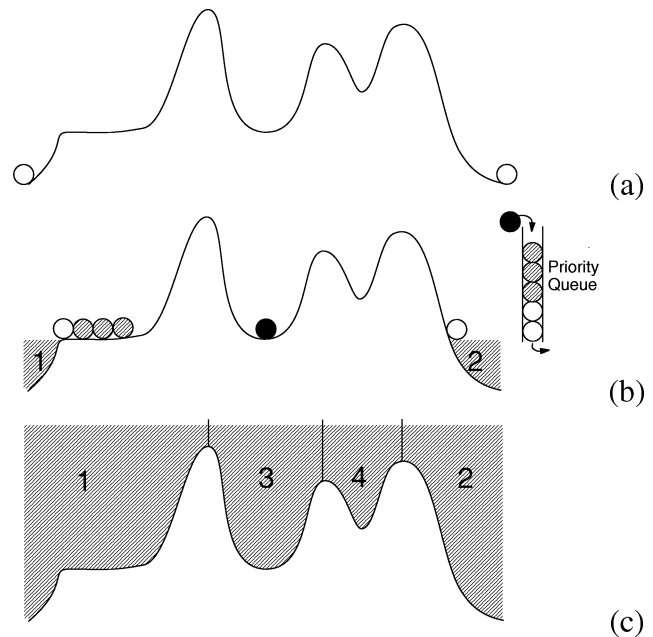


Fig. 2. Illustration of Vincent and Soille’s watershed process.

tion, which makes this algorithm inefficient for video segmentation.

III. PROPOSED WATERSHED ALGORITHM

For image sequences segmentation, Vincent and Soille’s algorithm can be further accelerated when considering temporal coherence of sequences. The core concepts of the proposed algorithm are as follows.

- 1) Watersheds are highly related to the original image sequences.
- 2) Temporal coherence of image sequences leads to redundant computation when watershed transform is applied to the whole image frame by frame, that is, the process can be accelerated with “updating” scheme.
- 3) The computational intensity of sorting and flooding in the watershed algorithm is proportional to the number of processed pixels, that is, reducing the number of pixels can accelerate the process.
- 4) Since watershed is a global operation, the algorithm should be modified in the updating area to keep the globalization.

With these concepts, a new fast algorithm is proposed as follows.

A. Predictive Watershed

The block diagram of the proposed watershed algorithm is shown in Fig. 3. The first frame of an image sequence is processed with the conventional Vincent and Soille’s algorithm, while keeping the region label and gradient information, as shown in Fig. 4(a). The *corresponding frame* stores the original frame data to whom the current watersheds correspond and is set to be the same as the first frame at this time. An *updated area mask (UAM)* is used to indicate the changing parts of the image, where the associated gradient values and watersheds

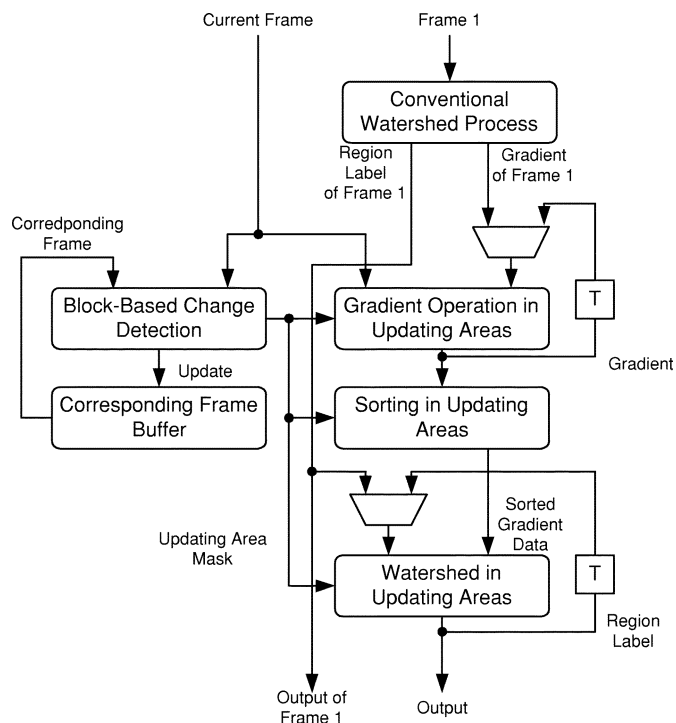


Fig. 3. Block diagram of the proposed watershed algorithm.

need to be updated. Since the UAM is set to be zero in the first frame, *gradient operation in updating areas* and *watershed in updating areas* just pass the gradient values and the region labels of the first frame to the memories (or the delay elements), as shown in Fig. 4(a).

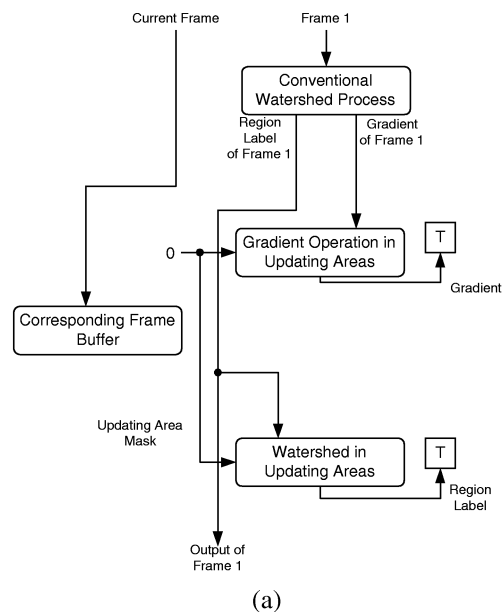
From the second frame, the proposed watershed algorithm named predictive watershed (P-Watershed) takes over, and the conventional watershed process is turned off, as presented in Fig. 4(b). The *current frame* and *corresponding frame* are input into *block-based change detection* to form *updating area mask (UAM)*, and the *corresponding frame* is then updated with the *current frame* and UAM . These two frames are first separated into small blocks, whereas the selection of block size will be discussed in Section IV. The operation executed in *block-based change detection* can be formulated, for each block, as the following equations:

$$SAD_i = \sum_{(x,y) \in \text{block } i} |CFrame_{x,y} - CorFrame_{x,y}| \quad (1)$$

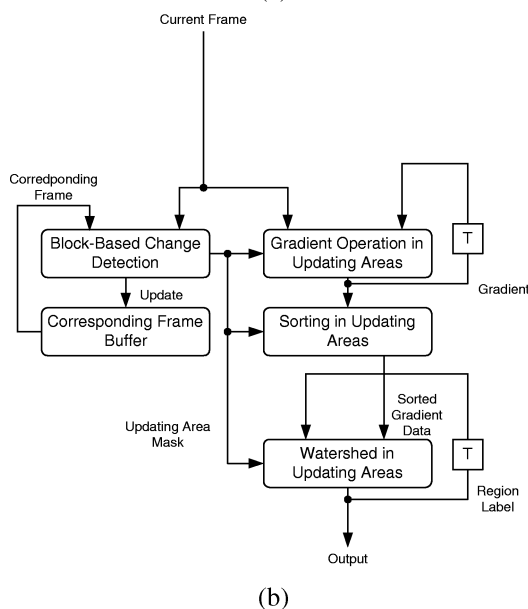
$$UAM_{x,y} = \begin{cases} 1, & \text{if } SAD_i \geq Th \text{ and } (x,y) \in \text{block } i \\ 0, & \text{else} \end{cases} \quad (2)$$

$$CorFrame_{x,y} = \begin{cases} CFrame_{x,y}, & \text{if } UAM_{x,y} = 1 \\ CorFrame_{x,y}, & \text{else} \end{cases} \quad (3)$$

where $CFrame$ is the current frame, $CorFrame$ is the corresponding frame, and SAD_i is the sum of absolute difference of the i th block of the *current frame* and *corresponding frame*. Th is the threshold to decide if the content of a block is “changing.” It can be decided with a significance test [13] where the users can choose the significance level according to the applications, and the threshold can be decided according to the significance level. The test statistic is the sum of absolute value of frame dif-



(a)



(b)

Fig. 4. Procedures of the proposed watershed algorithm for (a) the first frame and (b) the other frames.

ference. The threshold value is decided by required significance level. Their relation is shown in the following equation:

$$\alpha = \text{Prob}(SAD > Th | H_0) \quad (4)$$

where α is the significance level, Th is the threshold value, SAD is the sum of absolute difference of a block, and H_0 denotes the null hypothesis that there is no change at the current block. The selection of the threshold depends on the environments and the applications. Namely, for smaller camera noise, Th can be smaller. For higher correctness requirements, a smaller Th can be chosen; however, the penalty will be the higher false alarm rate. Note that the change detection here is different from that for video segmentation. Unlike video segmentation, where uncovered background should be avoided, in predictive watershed, it is used to indicate the changing parts of the frame, including uncovered regions and occluded regions. The watersheds in the

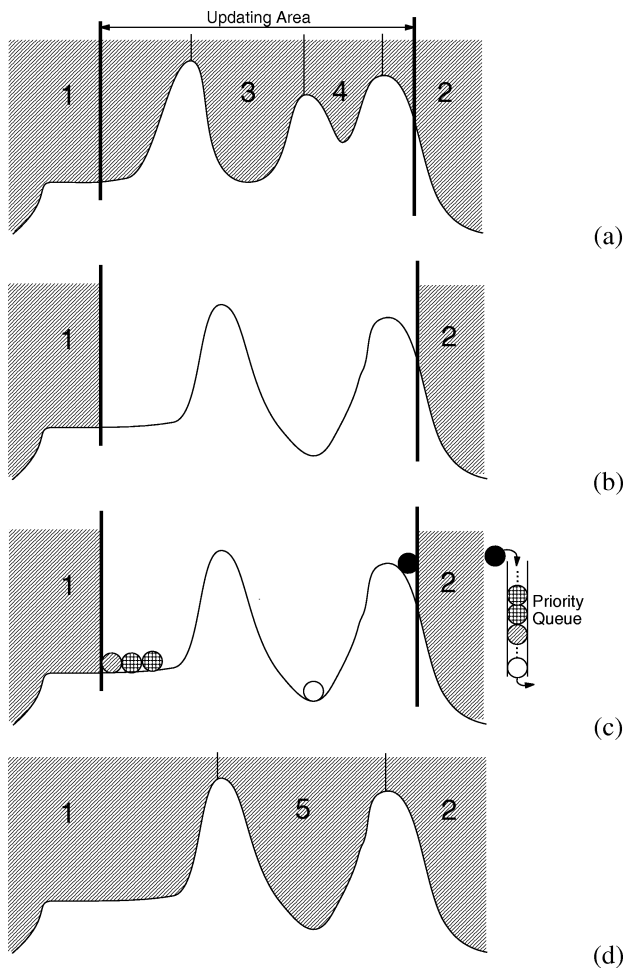


Fig. 5. Illustration of the proposed watershed process.

changing parts are then updated regardless of whether they belong to foreground or background objects.

The generated *UAM* can indicate the changing parts of the image and the corresponding watersheds to be updated. Next, morphological gradient and sorting are applied only to pixels in the updating area (*UA*), which is indicated by *UAM*. The watershed process is then also applied in *UA*. The whole process is shown in Fig. 5 and involves the following steps.

- Step 1) Fig. 5(a) shows the region labels and gradient values of the previous frame, and the updating area is marked.
- Step 2) As shown in Fig. 5(b), region labels and gradient values in *UA* are removed, and morphological gradient operation is then employed in *UA* to generate gradient values of the current frame. *Gradient operation in updating areas* in Fig. 4(b) executes the operations of this step.
- Step 3) Pixels in *UA* are sorted according to the gradient level, which is executed in *sorting in updating areas* in Fig. 4(b).
- Step 4) Pixels in *UA* are then processed from lower to higher gradient level. In each level, the process is the same as Step 4 of Vincent and Soille's algorithm. Note that neighbors outside *UA* are also taken into consideration, as shown in Fig. 5(c). The twilled ball has a

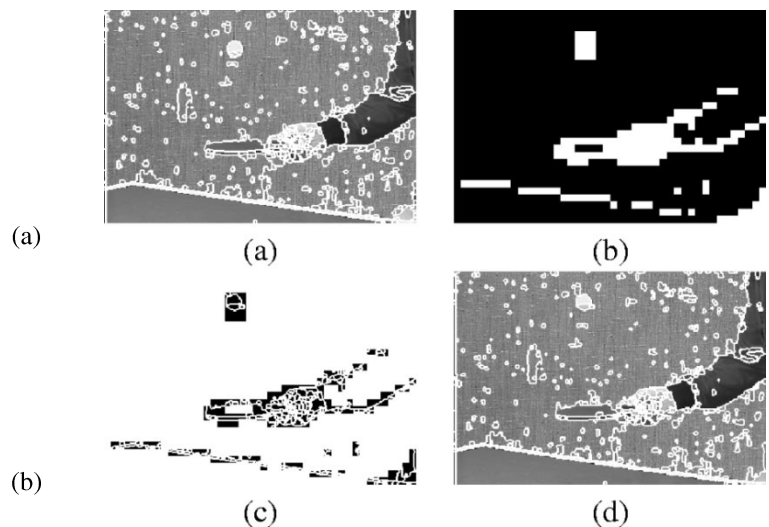


Fig. 6. Example of the proposed watershed algorithm. (a) Watersheds of previous frame. (b) Updating area mask. (c) Watersheds in the updating area, which will then be used to update the watersheds of previous frame. (d) Watersheds of current frame, which is generated by updating watersheds in (a) by watersheds in (c). The difference between (a) and (d) can be found clearly around the ball.

neighbor point outside *UA* in region 1; therefore, it will be also labeled as region 1. Similarly, the black ball will be labeled as region 2. After all pixels with labeled neighbors are processed in each level, pixels without labeled neighbors are found and given a new label, as the case of the white ball shown in Fig. 5(c). In Fig. 4(b), *watershed in updating areas* takes charge of this step.

- Step 5) After all pixels are labeled, the boundaries between regions with different labels are the watersheds of this image. In Fig. 5(d), two watersheds are detected.

An example of P-Watershed is shown in Fig. 6. In Fig. 6(a), the previous frame and its watersheds are shown. The *UAM* is shown in Fig. 6(b), and the watersheds in *UA* are then generated as shown in Fig. 6(c). Finally, in Fig. 6(d), the watersheds of the current frame are formed by updating the watersheds in *UA*. The difference between Fig. 6(a) and (d) can be found clearly around the ball.

Only parts of pixels are processed, hence the proposed algorithm is faster than the conventional watershed algorithm. Pixels inside *UA* are processed with the conventional algorithm, and the region labels and gradient information outside *UA* are also taken into consideration; therefore, the globalization property is also kept with this scheme.

B. Hybrid Scheme

Although the segmentation results are almost the same as those of the conventional watershed algorithms, the proposed algorithm may introduce error in the following conditions. First, change detection is not sensitive enough. Second, the minimum of a catchment basin is included in *UA*, and a part the catchment basin is not included. Third, the scenes change a lot. The error will propagate in the proposed predictive watershed algorithm. The error propagation can be interrupted by inserting a

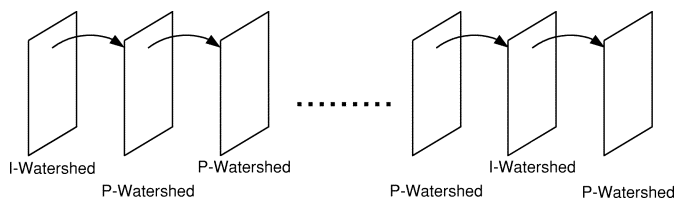


Fig. 7. IP-Watershed. I-Watershed frames are inserted to avoid error propagation of Predictive Watershed.

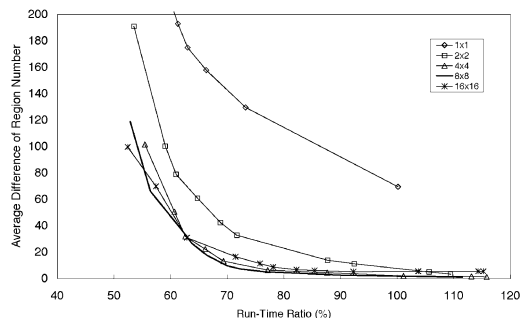


Fig. 8. Block size analysis showing that 8×8 is the optimal block size.

frame where watersheds are generated with the conventional algorithm. These watersheds do not need information from the previous frame, so we have the term “intra watershed” (I-Watershed). The I-Watershed and P-Watershed hybrid scheme, called IP-Watershed, is illustrated in Fig. 7. It can accelerate the watershed process and maintain accuracy at the same time.

The time to insert an I-Watershed can be decided with the following two strategies to interrupt the error propagation. To deal with the first and the second error conditions described above, we can insert an I-Watershed frame after a fixed number of P-Watershed frames. The fixed interval to insert an I-Watershed frame depends on the threshold T_h and the error rate required for the applications. After T_h is decided, the error accumulation behavior, which is the amount of increasing error for each frame, is also decided. An I-Watershed should be inserted when the accumulated error exceeds the required error rate. The less the error accumulated in each frame, the longer the interval to insert an I-Watershed frame; the tighter the error requirement, the shorter the interval. On the other hand, to deal with the third condition, scene change condition, the error rate is monitored in each frame. An I-Watershed frame is inserted dynamically when the monitored error rate exceeds a required value. These two strategies can be employed simultaneously to interrupt error propagation effectively. An example is presented in Section IV.

IV. SIMULATION RESULTS

Block size analysis for determining the optimal block size is shown in Fig. 8. The accuracy of the segmentation results is evaluated by the difference in region number, that is, the less average difference in region number between the results of the conventional algorithm and the proposed one, the higher the accuracy. Since regions in UA and outside UA are both processed with the conventional watershed algorithm, the positions of watersheds must be very close to those of the conventional algorithm, and only the over-segment problem may occur; the region number information is enough to evaluate the accuracy of

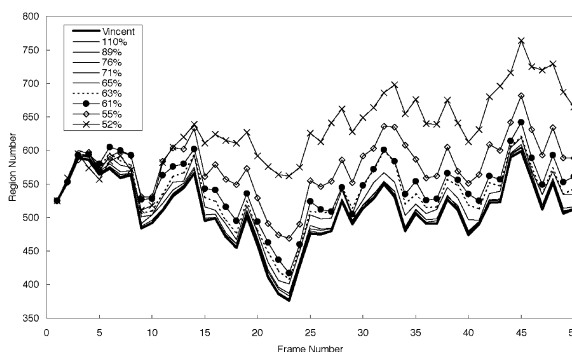


Fig. 9. Region number of different run-time situations.

watersheds, i.e., the closer the region number is between the result of the conventional watershed transform and that of the proposed one, the more accurate the proposed algorithm is. The ratio of the run time of the proposed algorithm to that of the conventional watershed algorithm is used as another criterion in this analysis. For each block size, a different threshold value T_h is tested to decide the runtime-accuracy curve shown in Fig. 8. The lower the threshold, the higher the runtime ratio, and the lower the average difference of region number. The test sequence is *Children*. Fig. 8 shows that 8×8 is the optimal block size because it can provide higher accuracy with less run time than other choices. Several sequences were tested to give a similar conclusion.

The region number corresponding to a different run time is shown in Fig. 9. As can be seen, when the run time increases, the similarity between the results of the conventional watershed and the proposed one also increases. When the run time is reduced to 61%, the region number is still very similar to the reference results.

The segmentation results of Vincent and Soille’s algorithm and the proposed one are shown in Fig. 10. Sequences *Akiyo*, *Hall Monitor*, and *Mother and Daughter* were tested on a PC with a Pentium-III 800-MHz processor. The execution time compared with Vincent and Soille’s algorithm is shown in Table I. Only core operations of watershed are recorded, namely, the gradient operation, sorting process, and flooding process. Note that the average difference in region number is fixed at 5% in these experiments. The simulation results show that the proposed algorithm can save 20%–50% of the computation. Moreover, the results of the conventional and proposed algorithms are similar, as shown in Fig. 10. Consequently, the proposed algorithm can reduce computational intensity while maintaining segmentation quality.

The simulation results of IP-Watershed are shown in Figs. 11 and 12, where *Weather* is used as the test sequence. Both strategies to insert I-Watershed frame described in Section III-B are employed. The threshold T_h is found to be 320 by significance test. If the error rate is evaluated by the region number difference ratio (RNDR), which is the ratio of the region number difference to the region number of current frame derived with conventional watershed algorithm (reference region number), the error rate curve is shown in Fig. 13. If we set $RNDR \leq 5\%$, the fixed interval should be set as about 20 frames, namely, an I-Watershed frame is inserted every 20 frames. On the other hand, I-Water-

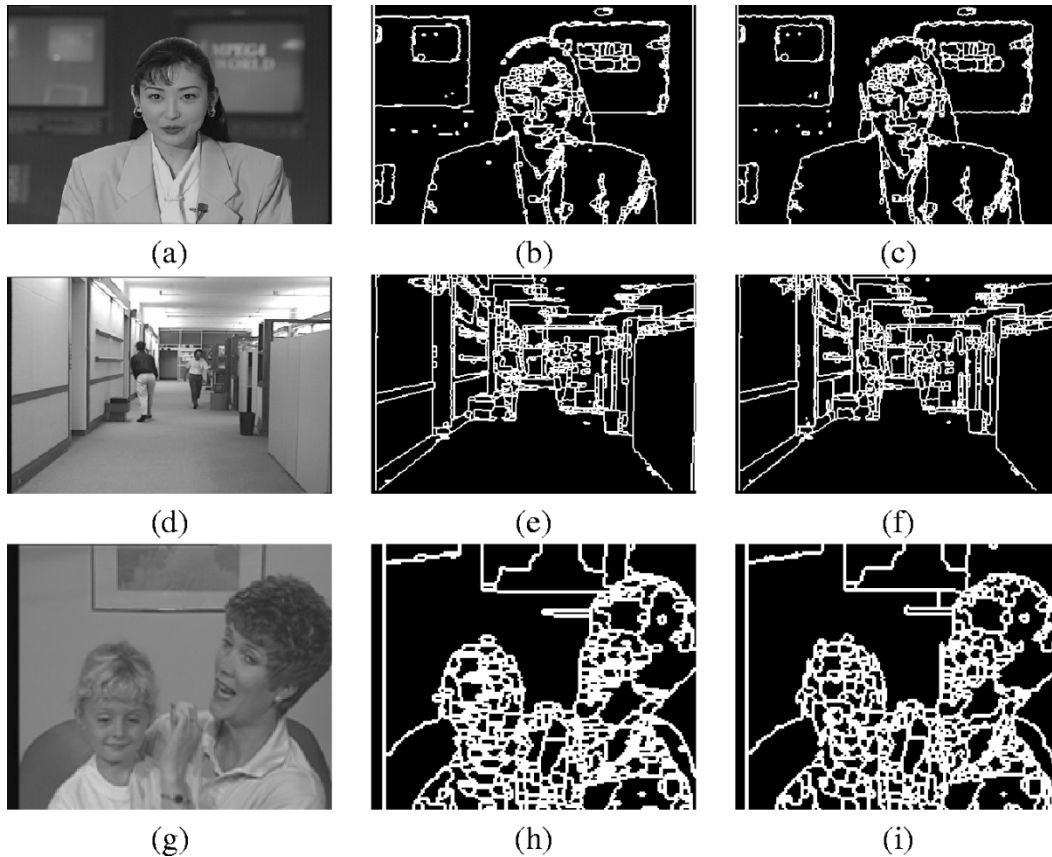


Fig. 10. (a), (d), (g) Original frame of sequence *Akiyo*, *Hall Monitor*, and *Mother and Daughter*. (b), (e), (h) Watersheds generated with the conventional watershed algorithm. (c), (f), (i) Watersheds generated with the proposed algorithm. The 50th frame of these sequences is shown.

TABLE I
RUN-TIME COMPARISON BETWEEN THE PROPOSED ALGORITHM AND VINCENT
AND SOILLE'S ALGORITHM

Sequence	Size	(a) Vincent and Soille's (ms/frame)	(b) Proposed (ms/frame)	Ratio(%) $= (b)/(a)$
Akiyo	360x243	202.16	107.64	53.24
Weather	360x243	203.89	103.15	50.59
Hall	360x240	199.77	160.21	80.20
Children	360x240	200.02	128.61	64.30
Mother and daughter	176x144	55.36	41.87	75.63

shed frames are also inserted dynamically to deal with scene change condition. The values of RNDR cannot be obtained in real cases since the region number of current frame derived with conventional watershed algorithm are not available; therefore, we use an estimated RNDR (ERNDR) value to evaluate the error rate, where the reference region number is substituted by the region number of nearest prior I-Watershed frame. In these experiments, we set $ERNDR \leq 5\%$, that is, if the monitored ERNDR value is higher than 5%, an I-Watershed frame is inserted. Fig. 11 shows that compared with P-Watershed, IP-Watershed can give results more similar to those of conventional algorithms. Note that I-Watershed frames are inserted at frame 1, 21, 41, 61, 81 according to the first strategy and frame 14

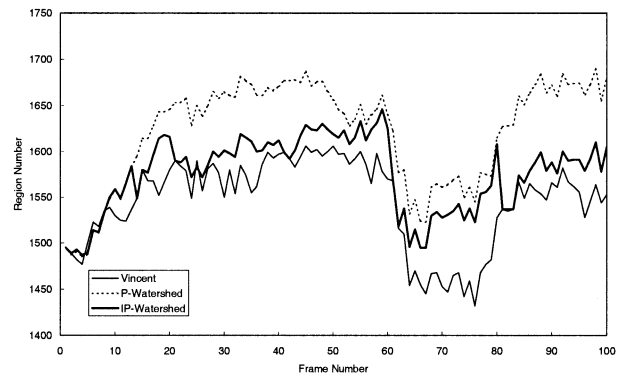


Fig. 11. Simulation results of the IP-Watershed (test sequence: *Weather*).

according to the second strategy. The segmentation quality can be improved with IP-Watershed as shown in Fig. 12, where the result of IP-Watershed [Fig. 12(d)] is more similar to that of the conventional algorithm [Fig. 12(b)] than that of P-Watershed [Fig. 12(c)]. The lower middle regions of Fig. 12(b)–(d) are, respectively, magnified to Fig. 14(a)–(c) to show the results more clearly. The results show that although the segmentation result of IP-Watershed is slightly different than that of the conventional algorithm, it is better than that of the P-Watershed algorithm, where the over-segmentation problem occurs.

V. VIDEO SEGMENTATION BY USE OF PREDICTIVE WATERSHED

The proposed fast algorithm can be combined with video segmentation to improve the segmentation results. An example is

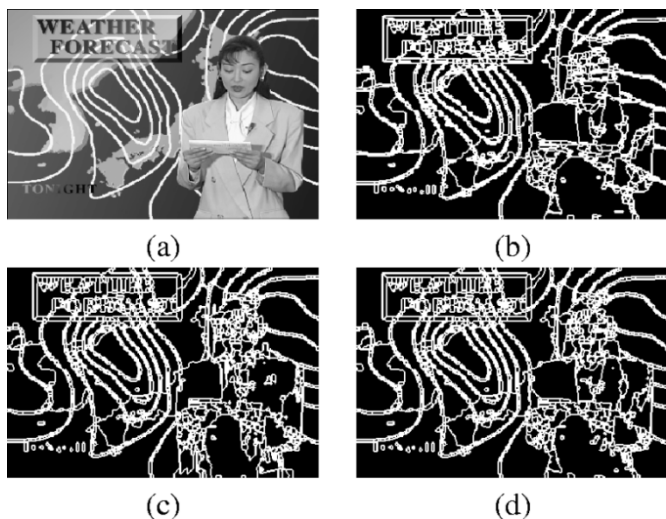


Fig. 12. (a) Original frame of sequence *Weather*. Watersheds generated with (b) the conventional watershed algorithm, (c) the P-Watershed algorithm, and (d) the IP-Watershed. (d) is more similar to (b) than (c).

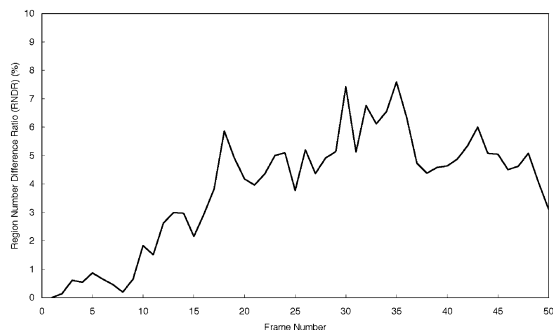


Fig. 13. Region number difference ratio curve of sequence *Weather* under $Th = 320$.

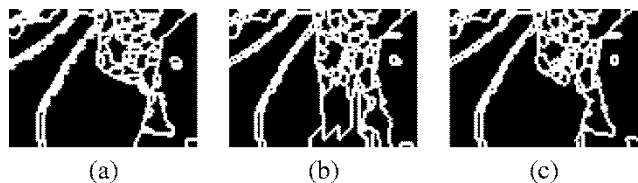


Fig. 14. Detailed segmentation results of: (a) the conventional watershed algorithm; (b) the P-Watershed algorithm; and (c) the IP-Watershed.

presented in this section, where a new video segmentation algorithm is proposed. The block diagram of the proposed algorithm is shown in Fig. 15. It is developed from the real-time change detection and background registration based algorithm proposed in [14], [15], as well as the predictive watershed algorithm.

The algorithm has two modes: a baseline mode for general situations and a shadow-cancellation mode for video sequences influenced by light changing and shadow effects. In baseline mode, *gradient filter* is bypassed. First, the background registration technique can extract background information from accumulated frame difference of successive frames. The frame difference and background difference, which is the frame difference between the current frame and background frame, are then thresholded and combined to form an initial object mask. Next, the *initial object mask* is refined by eliminating the noise re-

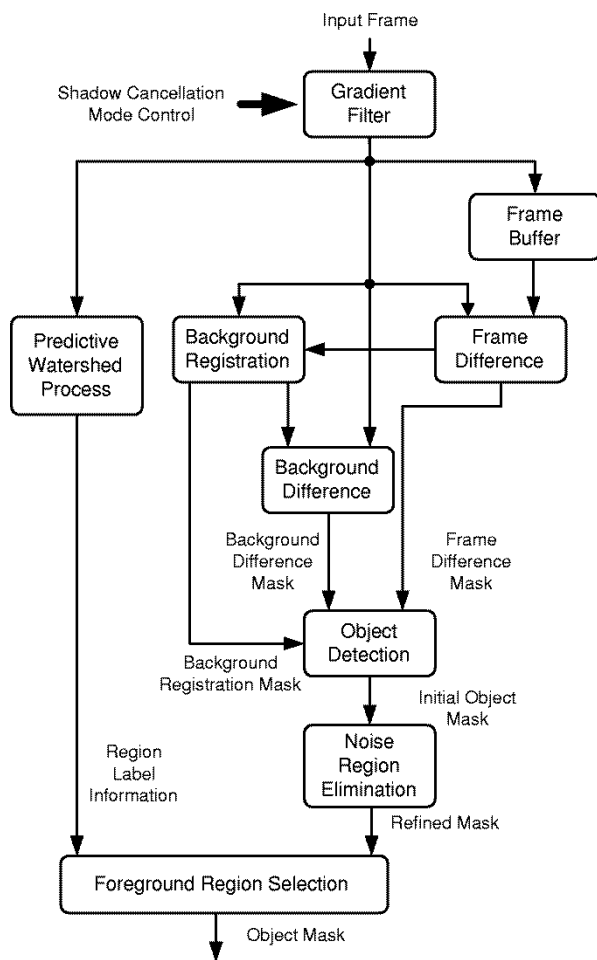


Fig. 15. Block diagram of the proposed video segmentation algorithm.

gions in the mask. Meanwhile, the *predictive watershed process* generates *region label information*; that is, the current frame is divided into many homogeneous regions, and each region is labeled with a unique label. Finally, regions are selected by the *refined mask* to form the *object mask*. If the percentage of pixels in the *refined mask* within a region is beyond a threshold, the whole part of the region is labeled as a foreground object; otherwise, it is regarded as a background region. In the shadow-cancellation mode, the *gradient filter* is used to suppress the effect of shadow and light changing.

A. Simulation Results

The segmentation results of the original real-time change detection and background registration-based algorithm [14], [15] and the proposed algorithm are shown in Fig. 16(b) and (e) and Fig. 16(c) and (f), respectively. The IP-Watershed algorithm is adapted in these experiments. As seen in the comparison, the proposed algorithm provides more precise segmentation results. The difference of the accuracy can be easily observed around the heads of the children in Fig. 16(b) and (c) and around the face of the mother in Fig. 16(e) and (f). It can improve the segmentation results by fitting the object masks to the regions generated with watershed transform.

Fig. 17 shows the simulation results of sequence *Akiyo*, *Mother and Daughter*, *Hall Monitor*, and *Children*. Two frames

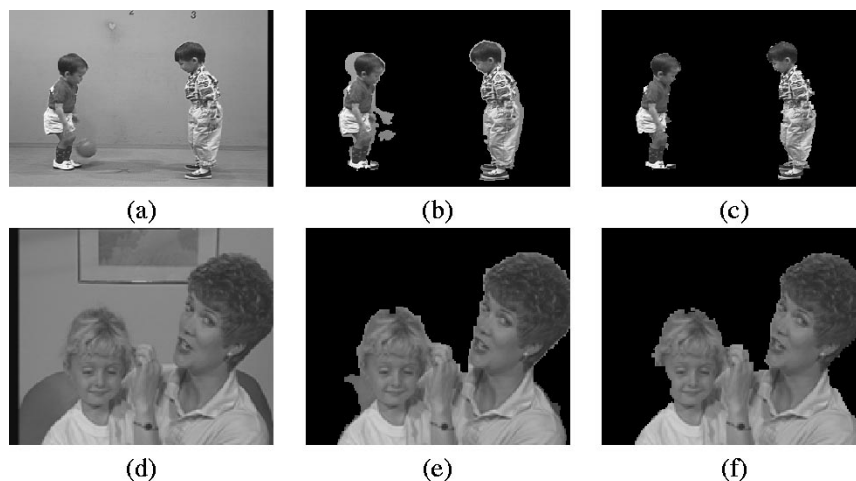


Fig. 16. Comparison between the segmentation results of (b), (e) the original segmentation algorithm and (c), (f) the proposed algorithm. The boundaries of object masks are more accurate in (c), (f) than those of (b), (e). (a), (b) are the original frames.



Fig. 17. Segmentation results of sequence *Akiyo*, *Mother and Daughter*, *Hall Monitor*, and *Children*.

are picked from each sequence. The segmentation results are very precise at boundaries.

The run-time analysis, found in Fig. 18, shows that the run time of the previous segmentation algorithm takes 24% of the whole run time. Only 106.64 ms is required to process a single frame on a PC with a Pentium-III 800-MHz processor, and the

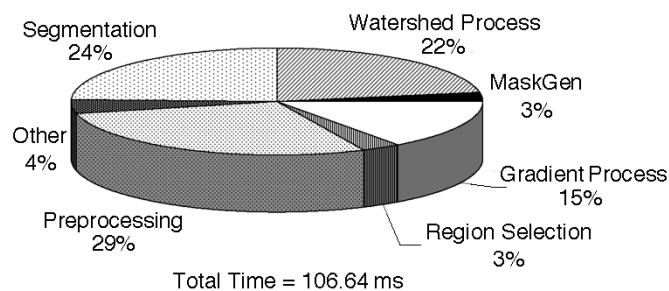


Fig. 18. Run-time analysis.

processing speed is 9.4 QCIF frames per second, which is very close to real-time requirement. The algorithm can be further optimized, and will be included in future work.

VI. CONCLUSION

A new predictive watershed algorithm named P-Watershed for video segmentation is proposed in this paper. Taking into consideration the temporal coherence property of the video signal, the watershed algorithm can be accelerated. It updates watersheds in changing parts while keeping watersheds in other parts of a frame. The watershed process can be accelerated, and the results are almost the same as those of the conventional watershed algorithms. Moreover, the segmentation results can be further improved with the intra-inter watershed scheme, which is named the IP-Watershed. The proposed algorithm can be combined with any video segmentation algorithm to improve the segmentation results.

REFERENCES

- [1] T. Sikora, "The MPEG-4 video standard verification model," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, pp. 19–31, Feb. 1997.
- [2] *Annex f: Preprocessing and Postprocessing*, ISO/IEC JTC 1/SC 29/WG11 N3056.
- [3] D. Wang, "Unsupervised video segmentation based on watersheds and temporal tracking," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, pp. 539–546, Sept. 1998.

- [4] J. G. Choi, S.-W. Lee, and S.-D. Kim, "Spatio-temporal video segmentation using a joint similarity measure," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, pp. 279–286, Apr. 1997.
- [5] M. Kim, J. G. Choi, H. Lee, D. Kim, M. H. Lee, C. Ahn, and Y.-S. Ho, "A VOP generation tool: automatic segmentation of moving objects in image sequences based on spatio-temporal information," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, pp. 1216–1226, Dec. 1999.
- [6] D. K. Park, H. S. Yoon, and C. S. Won, "Fast object tracking in digital video," *IEEE Trans. Consumer Electron.*, vol. 46, pp. 785–790, Aug. 2000.
- [7] L. Vincent and P. Soille, "Watersheds in digital spaces: an efficient algorithm based on immersion simulations," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 13, pp. 583–598, June 1991.
- [8] F. Meyer, "Color image segmentation," in *Proc. Int. Conf. Image Processing and Its Applications*, 1992, pp. 303–306.
- [9] E. R. Dougherty, Ed., *Mathematical Morphology in Image Processing*. New York: Marcel Dekker, 1993, ch. 12, pp. 433–481.
- [10] B. P. Dobrin, T. Viero, and M. Gabbouj, "Fast watershed algorithms: analysis and extensions," in *SPIE Nonlinear Image Processing V*, vol. 2180, 1994, pp. 209–220.
- [11] A. Moga, B. Cramariuc, and M. Gabbouj, "An efficient watershed segmentation algorithm suitable for parallel implementation," in *Proc. Int. Conf. Image Processing*, 1995, pp. 101–104.
- [12] D. Hagyard, M. Razaz, and P. Atkin, "Analysis of watershed algorithms for grayscale images," in *Proc. Int. Conf. Image Processing*, 1996, pp. 41–44.
- [13] T. Aach, A. André Kaup, and R. Rudolf Mester, "Statistical model-based change detection in moving video," *Signal Processing*, vol. 31, no. 22, pp. 165–180, Mar. 1993.
- [14] S.-Y. Chien, S.-Y. Ma, and L.-G. Chen, "An efficient video segmentation algorithm for real-time MPEG-4 camera system," in *Proc. Visual Communication and Image Processing*, 2000, pp. 1087–1098.
- [15] S.-Y. Chien, S.-Y. Ma, and L.-G. Chen, "Efficient moving object segmentation algorithm using background registration technique," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, pp. 577–586, July 2002.