

Pixel-based Extraction of Moving Objects for Sprite Coding

Shigemitsu WATANABE ^{*1}, Kumi JINZENJI ^{*2}, Hiroshi WATANABE ^{*1}

^{*1} Graduate School of Global Info. and Telecom. Studies, Waseda University
29-7 Bldg., 1-3-10 Nishi-Waseda, Shinjuku-ku, Tokyo 169-0051, Japan

^{*2} NTT Cyber Space Labs, NTT Corporation,
1-1 Hikari-no-oka,
Yokosuka-shi, Kanagawa 239-0847, Japan

ABSTRACT

In this paper, a method to extract moving objects on a pixel base for MPEG-4 “sprite coding” is proposed. Sprite coding is a form of object coding: it uses a unified panoramic background image derived from a sequence having a camera motion, and a foreground object as video object planes (VOP’s). The proposed algorithm utilizes background difference, which is the difference between the original image and the background image, and watershed transform to extract the foreground moving objects. We first apply background difference to generate a foreground candidate image. Then, watershed transform is applied to this candidate image to extract the contours of the foreground objects. Furthermore, we utilize the macroblock mask of the foreground objects, which is generated by the conventional method, to delete the unnecessary edges extracted by watershed transformation. Results given by our proposed method are more than twice as better than that of the conventional method.

Keywords: Object Extraction, Watershed Transform, Background Difference, Sprite Coding

1. INTRODUCTION

MPEG-4 provides an efficient coding tool, “sprite coding.” In sprite coding, a “sprite” i.e. a unified panoramic background image derived from a sequence having a camera motion, and a foreground object are utilized as VOP’s. These VOP’s can be coded in different ways to suit their characteristics. For instance, the foreground object can be coded using an arbitrary shape, and the background sprite

can be coded using sprite coding. Sprite coding offers high compression since number of frames can be represented by a single (panoramic) still image. An algorithm that automatically generates the background sprite and the foreground objects, called “Two-layer VOP generation scheme,” has been proposed [1-4]. However, this method approximates the foreground object by using macroblocks. For this reason, unnecessary background regions are extracted as foreground, and the composed image may include visual discrepancy. Thus, a method to extract the foreground objects on a pixel base is necessary. We propose an algorithm that automatically extracts the moving objects on a pixel base using background difference and watershed transform.

The rest of this paper is organized as follows. In Section 2, the proposed object extraction algorithm is described. The overview of the proposed algorithm is introduced first, and the details of the algorithm are described in the following subsections. In Section 3, results are shown and compared with the results of the conventional method. Moreover, the results are evaluated numerically. Finally, Section 4 concludes this paper.

2. OBJECT EXTRACTION ALGORITHM

A method for object extraction that uses background difference and watershed transform has been proposed [5]. However, it can only be applied to a video sequence where background is not moving, and camera motion such as pan, tilt, and zoom is not included. Hence, it cannot be applied to sprite coding. The flow chart of our proposed algorithm is shown in Fig. 1. The input images are the proposed object extraction method consists mainly

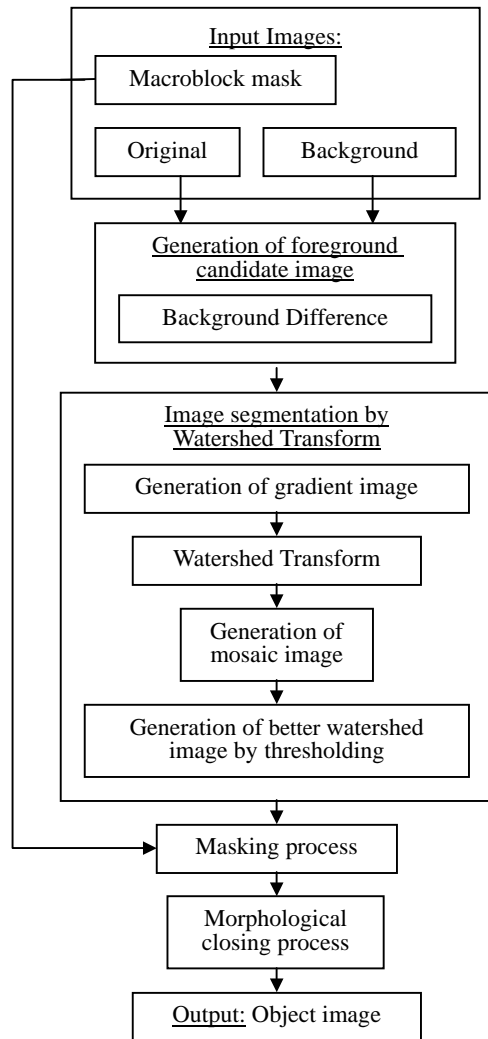


Fig. 1: Flow chart of the proposed algorithm

of four operations. First, we generate a foreground candidate image by background difference using the original and background images. Second operation is the image segmentation by watershed transform. We apply watershed transform to the gradient image of the foreground candidate image. During this second operation, we make a mosaic image and generate the better watershed image by thresholding in order to avoid the over-segmentation problem. Third operation is masking process using the macroblock mask of the foreground object that is generated by the conventional method. By this operation, the unnecessary edges can be deleted, leaving only the

edges around the objects that we want to extract. Fourth operation is the generation of the final object mask by applying the morphological closing process. In the following subsections, each of these four operations is discussed in detail.

2.1. Extraction of Foreground Candidate Image by Background Difference

First, we calculate the difference image between the original image and the background image. The background image is extracted from the background sprite that is generated by the conventional method. However, since the background image is extracted using the global motion, it does not always coincide with the original image if the image contains complicated background. It is rather shifted by a few pixels from the original image. For this reason, we use a window of certain size to scan the background image, and calculate the differences between the target pixel of the original image and the pixels of the background image within the window. Then, we define the minimum of the differences as the target pixel value. The resulting difference image becomes the foreground candidate image.

2.2. Image Segmentation by Watershed Transformation

In order to extract the foreground object accurately from the foreground candidate image, image segmentation such that the boundary of the segmented region coincides with the contour of the object must be needed. To satisfy this condition, we use the well-known image segmentation tool, "watershed transform" [6-7]. The concept of the watershed transform is to consider an image as a topographic surface, and pierce each minimum of the surface and then, sink this surface into the water with a constant vertical speed (Fig. 2). By this operation, the water entering through the holes floods the surface. During this flooding process, we build a dam, which defines the watershed, on the points of the surface where the floods coming from different minima would merge. At the end of the process, only the dams merge, separating the catchment basins that contain one minimum each.

Since the watershed transform considers an image as a topographic surface, we apply the watershed transform to the gradient of the

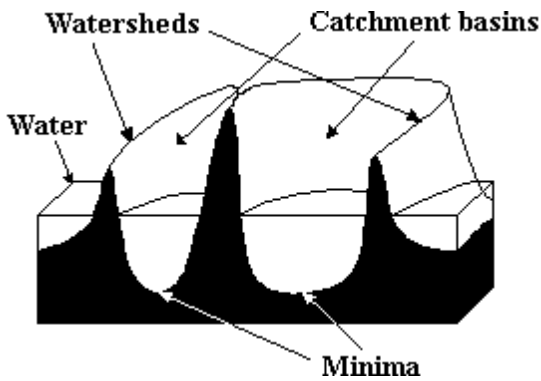
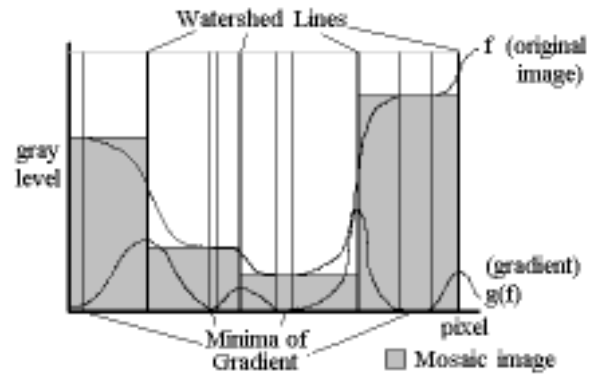


Fig. 2: Concept of watershed transform

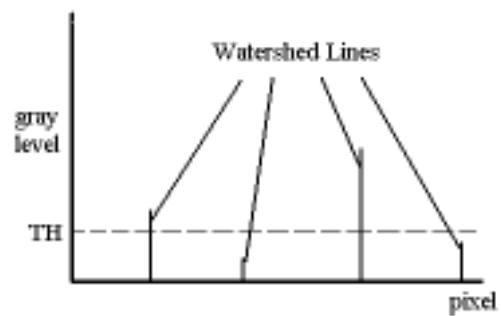
foreground candidate image generated by the background difference. However, watershed transform tends to over-segment because, in an actual image, many unnecessary catchment basins, which are the minima of the gradient image, are produced mainly due to noise. In order to avoid this problem, we generate a simplified mosaic image of the original image. The mosaic image can be generated in the following way. First, we calculate the watershed of the gradient image. Secondly, we fill the region between the watersheds, i.e. label every catchment basin of the watershed, with the gray value in the original image corresponding to the local minima of the gradient image (Fig. 3a). The boundaries between two regions of mosaic image are valued with the gray tone difference between these regions. The generated mosaic image is made of the catchment basins of constant gray levels, where no information regarding the contours has been lost. Then, taking the gradient of the mosaic image and deleting the boundaries less than some threshold gives the better watershed image (Fig. 3b).

2.3. Masking Process Using foreground Macroblock image

The conventional method generates a mask image of the foreground object that is approximated by macroblocks (see Fig.4). This macroblock image is produced by the following method. First, the difference image between the original image and the background image extracted from the background



(a) Graph of the mosaic image



(b) Graph of the gradient of the mosaic image

Fig. 3: Construction of mosaic image

sprite is generated. Then, this difference image is binarized by thresholding, and is split into a foreground candidate image and a background image. From this foreground candidate image, the final foreground image is produced by using two-phase macroblock approximation. First, the macroblock that contains more than threshold $Th1$ foreground pixels is regarded as foreground. All the other macroblocks are regarded as background. Secondly, the background macroblock that is adjacent to the foreground macroblock in the first macroblock approximation phase is considered. If more than threshold $Th2$ ($Th2 < Th1$) of foreground pixels are included in that macroblock, it is regarded as foreground. We use this macroblock mask image to mask the watershed image obtained in Section 2.2. By this operation, the unnecessary edges obtained by watershed transformation can be deleted, leaving only the edges around the objects we want to extract.

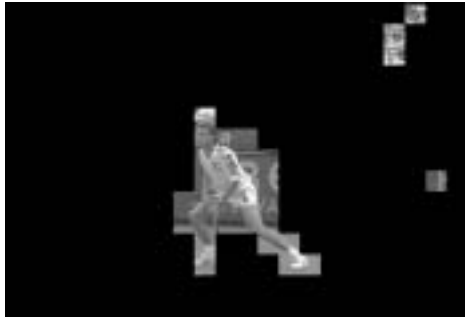
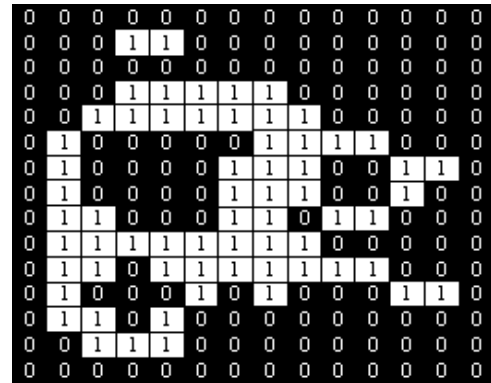
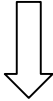


Fig 4: Macrobloc mask of the foreground object generated by the conventional method

2.4. Generation of the Object Mask by the Morphological Closing Process

In order to generate the final object image, we apply morphological closing process to the masked watershed image generated in section 2.3. Mathematical morphology is based on two operations: erosion and dilation. These operators are typically applied to binary images, and processed as follows. First, we scan the binary image using a window of certain size (for example, 3x3), called the structuring element. In the case of erosion, if every pixel in the structuring element is a foreground pixel, then the target pixel is left as it is. If any of the pixels are background, however, the target pixel is set to the background value. In the case of dilation, if at least one pixel in the structuring element is a foreground pixel, then the target pixel is set to the foreground value. If all the pixels are background, however, the target pixel is left at the background value. Closing operator is derived from these fundamental operations of erosion and dilation. It is simply a dilation followed by an erosion using the same structuring element for both operations. It has an effect of filling in particular background regions of the image such as gaps and holes (Fig. 5). By applying this morphological closing operator to the watershed image where foreground is the contour of the object, we fill in the regions inside the contour, and hence, generate the final object mask image.



CLOSING  PROCESS

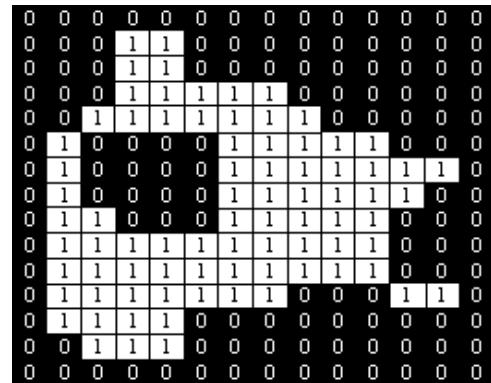


Fig. 5: Effect of morphological closing operator

3. RESULTS AND EVALUATION

The original image and the object image generated by the conventional method and by our proposed method are shown in Fig. 6. As shown in Fig. 6d-f, the conventional method approximates the foreground object by macroblocks. Thus, unnecessary regions that are supposed to be the background are included in the foreground. In comparison to this, Fig. 6g-h shows that the method we are proposing has been successful in improving the object extraction accuracy. Although the object images generated by our method still have some background regions as foreground, they are much more accurate compared to the object images



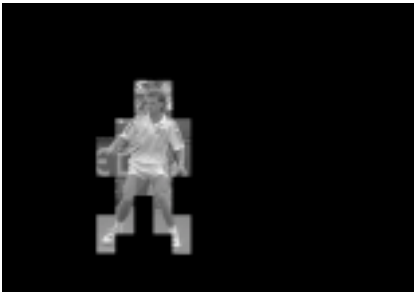
(a) Original image (40th frame)



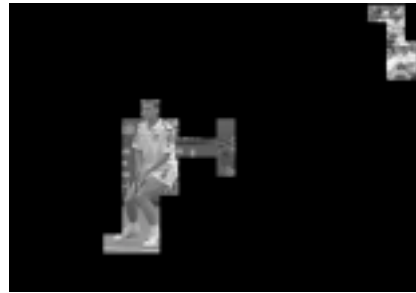
(b) Original image (60th frame)



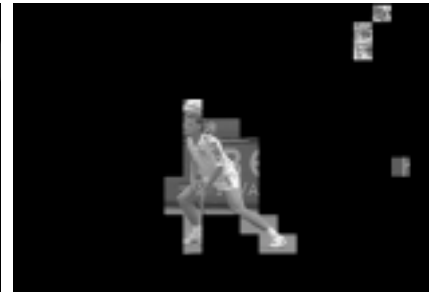
(c) Original image (80th frame)



(d) Object image generated by the conventional method (40th frame)



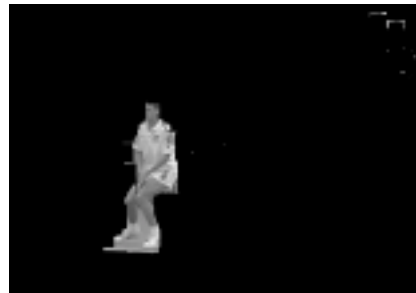
(e) Object image generated by the conventional method (60th frame)



(f) Object image generated by the conventional method (80th frame)



(g) Object image generated by the proposed method (40th frame)



(h) Object image generated by the proposed method (60th frame)



(i) Object image generated by the proposed method (80th frame)

Fig. 6: Object images generated by the conventional method and the proposed method

generated by the conventional method. In order to evaluate the results numerically, we used the video sequence “Stefan” which has the correct object segmentation mask. We evaluated our results by equation (1).

$$e(n) = A_d(n) / A_s(n), \quad (1)$$

where n is the frame number, A_d is the area of regions that is different from the segmentation mask, and A_s is the area of the correct segmentation mask.

Comparison of the proposed method with the conventional macroblock based method is shown in Fig. 7. It shows that the results given by our proposed method are more than twice as better than that of the conventional method.

4. CONCLUSION

We proposed an algorithm to extract the moving objects automatically from the video sequence for

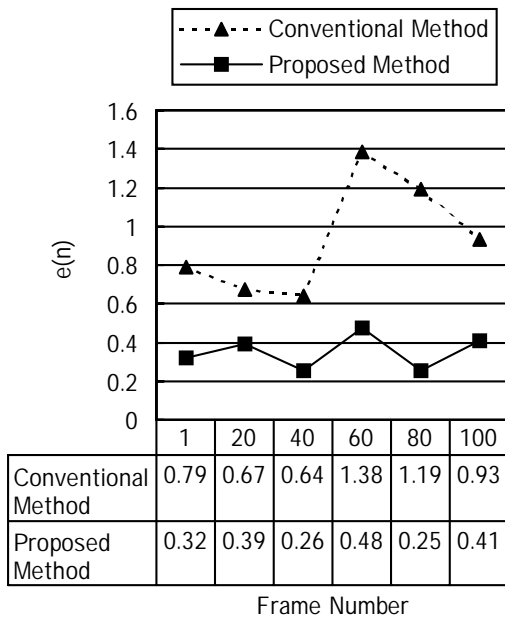


Fig. 7: Comparison with the conventional method

sprite coding. The proposed method utilizes background difference to extract the foreground candidate image first. Secondly, it uses the watershed transform to extract the contour of the object in order to achieve higher accuracy. Then, it utilizes the macroblock image of foreground objects, which is generated by the conventional method, to delete unnecessary edges. Finally, by operating the morphological closing process, final object mask image is generated. The object image generated by our proposed method is more than twice as better as the conventional method.

5. REFERENCES

- [1] K. Jinzenji, S. Ishibashi, N. Kobayashi, "Automatic Sprite Producing Method by Camera Motion Detection," IEICE Transaction, Vol. J82-D-II, No.6, pp.1018-1030, June 1999.
- [2] K. Jinzenji, H. Watanabe, N. Kobayashi, "Global Motion Estimation for Sprite Production and Application to Video Coding," IEICE Transaction, Vol. J83-D-II No.2, pp.535-544, Feb 2000.
- [3] H. Watanabe, and K. Jinzenji, "Sprite Coding in Object-based Video Coding Standard: MPEG-4,"

World Multiconference on SCI 2001 Proc. Vol. XIII, pp.420-425, July 2001.

- [4] K. Jinzenji, H. Watanabe, S. Okada, and N. Koboyashi, "Very Low Bit-Rate Video Compression Using MPEG-4 Sprite Coding," IEICE Transaction, Vol.J84-D-II, No.5, pp.758-768, May 2001.
- [5] S. Sakaida, M. Naemura, and Y. Kanatsugu, "Moving Object Extraction Using Background Difference and Region Growing with Spatio-Temporal Watersheds," IEICE Transaction, Vol.J84-D-II, No.12, pp.2541-2555, December 2001.
- [6] S. Beucher, "The Watershed Transformation Applied to Image Segmentation", 10th Pfefferkorn Conf. on Signal and Image Processing in Microscopy and Microanalysis, 16-19 sept. 1991.
- [7] Beucher S. and Lantuéjoul C., "Use of watersheds in contour detection," International Workshop on Image Processing, Real-Time edge and motion detection/estimation, Rennes, France, pp.17-21, Sept. 1979.