Segmentation of Interest Objects Using the Hierarchical Mesh Structure

Dong-Keun Lim[1] and Yo-Sung Ho[2]

[1] Digital Media R & D Center, Samsung Electronics
416 Maetan-3dong, Youngtong-gu, Suwon, Gyeonggi-do, 442-742, Korea
dk2003.lim@samsung.com
[2] Gwangju Institute of Science and Technology(GIST)
1 Oryong-dong, Buk-gu, Gwangju, 500-712, Korea
hoyo@gist.ac.kr

Abstract. The object boundary of an image plays an important role for image analysis and interpretation. The watershed algorithm and the region growing algorithm are popularly employed for image segmentation. These give reasonable performances, but require a large amount of computation time and sometimes fail to obtain continuous linkage of object boundary. In this paper, we introduce hierarchical mesh-based image segmentation. In each hierarchy, we employ neighborhood searching and boundary tracking methods to refine the initial boundary estimate. The proposed algorithm increases the robustness of linkage of object boundaries by overlooking and estimating connectivity and gives new modified chain coding. Reliable segmentation of objects can be accomplished by the proposed low complexity technique.

1 Introduction

The MPEG-4 international standards try to provide a solution to the challenging task in multimedia environment. It gives the broad range spectrum of requirements and applications through implementing content-based functionalities. One of the prerequisite condition is the ability to encode arbitrarily shaped VOs(Video Object). That skill is generally called image segmentation. In recent years, several algorithms for image segmentation have been proposed for certain applications, such as video conferences, where only one or two speakers exist in the static background [1][2].

Typical image segmentation algorithms include thresholding, region growing, split and merge, watersheds, or edge-based operations. Each operation has its own peculiar features and advantages. Initial segmentation is performed on the first frame of the video sequence by partitioning the frame into homogeneous regions based on image prosperities. The watershed algorithm [3][4] and the region growing algorithm [4-6] are popularly employed for initial segmentation; however, both of them require a large amount of computation time and they sometimes fail to obtain continuous linkage of object boundary. In order to overcome these problems, we propose a hierarchical approach for image segmentation using mesh

K. Aizawa, Y. Nakamura, and S. Satoh (Eds.): PCM 2004, LNCS 3331, pp. 230–238, 2004.

structures. The proposed algorithm also increases the robustness of linkage of object boundaries by overlooking and estimating connectivity at higher hierarchical levels and gives a new modified chain coding method which is applied to natural images, not limited to binary images.

2 New Segmentation Algorithm

2.1 Algorithm Overview

The proposed segmentation algorithm is based on hierarchical mesh classification with a pyramid data structure. The advantage of the hierarchical approach lies in the possibility of making a rough classification at a coarse level and then continuing into finer resolution to improve the segmentation accuracy. Our algorithm focus on interesting parts along object boundaries. The proposed algorithm is computationally efficient since only a fraction of all pyramid nodes is processed during the top-down classification [1][6].

Fig. 1 shows the flow diagram of the hierarchical image segmentation, which is similar to the divide-and-conquer algorithm for boundary detection.

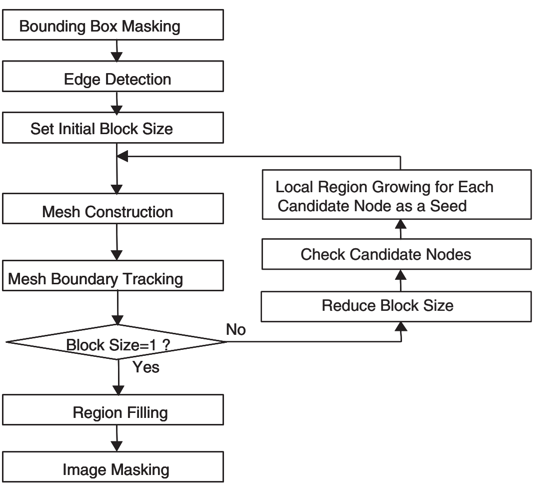
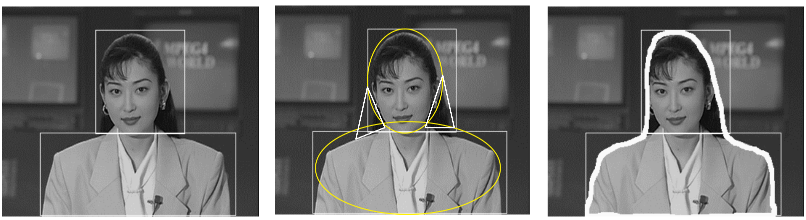


Fig. 1. The flow diagram of the proposed hierarchical mesh-based segmentation

We construct a hierarchical structure of meshes of different sizes using the edge information. Meshes are constructed by connecting the centroid of each candidate block. To follow the mesh boundary, we use a left-hand or a right-hand rule of a maze search based on the previous search direction. The algorithm works counterclockwise for the left-hand rule or clockwise for the right-hand rule along the meshes on the object boundary. The resulting boundary meshes are candidates of the next process in the mesh hierarchy. The candidate meshes are split into smaller meshes. We apply the mesh boundary tracking method and link the mesh boundary. This process continues iteratively until we reach one pixel resolution. In each level, local searches are carried out on the previous approximate boundaries. Since the resulting object boundary may not be accurate, we need to correct the boundary. We employ a local region growing algorithm to refine the boundary starting from the previous boundary as a seed. A segmentation mask is obtained by filling the refined object boundary.

2.2 Bounding Box Masking

Due to semantic ambiguity and content complexity, automatic segmentation algorithms are applied only on specific situation. For general applications, the user usually defines the object at the initial time. Semantic objects in the video frame can be identified.



(a) (b) (c)

Fig. 2. Boundary of Region of Interest: (a) multiple bounding boxes, (b) boxes, circles and polygons, and (c) hand-drawn curve

As shown in Fig. 2, multiple rectangular boxes, circles and hand-drawn lines can be used to define the region of interest (ROI). We can also combine those bounding shapes with some choosing rule such as intersection, union, XOR, etc.

2.3 Boundary Tracking

An image can be represented by a mesh structure [2]. Meshes are generally located on long object boundaries, which are important in the hierarchical structure since they provide the meaning of the objects. After we find locations of candidate boundary points, we refine the object boundaries. Fig. 3 explains a mechanism of hierarchical mesh construction and boundary linking. At Level k, edges are located in three quadrants. Those regions are candidates for Level k+1. In the same way, at Level k+1, edges are located in seven regions, which are candidate locations for Level k+2.

A mesh structure can be generated by linking the centroid (center of mass) position of each block. Since edge information does not change according to the

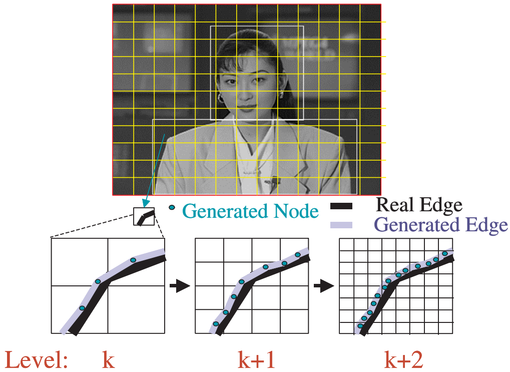
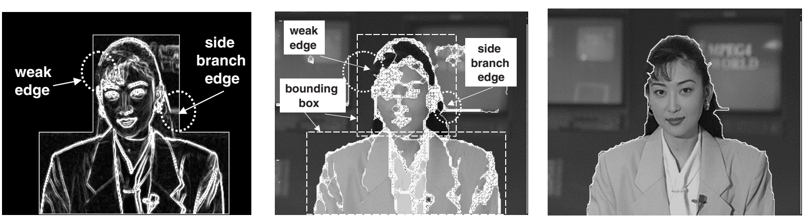


Fig. 3. Object locations at different levels or hierarchies: where k means the level

variation of luminance and has more object boundary information than luminance intensity one, we use edge information to obtain the centroid instead of luminance information.

As shown in Fig. 4, fragmentation problems in boundary tracking can arise when contours and boundaries are not smooth or have broken pieces.



(a) (b) (c)

Fig. 4. Boundary Linkage Problem due to Fragmentation: (a) edge image, (b) mesh overlayed image of (a), and (c) segmented image with several problems that include fragmentation and meaningless side branch

One of the advantages using hierarchical mesh structures can be explained from Fig. 5. From link1 to link3 in level (k+1), we can not estimate the broken link directly without the related link information in level k. Since general edge link and follow algorithms cannot estimate the broken edge within very small region for estimation, they fail to find a continuous linkage. Although they increase the region for estimation to increase the robustness of a continuous linkage, the computational time and ambiguity of linkage may be increased.

In the proposed hierarchical approach, the linkage information in level k gives the estimated path of the broken links in level (k+1). From that information, we

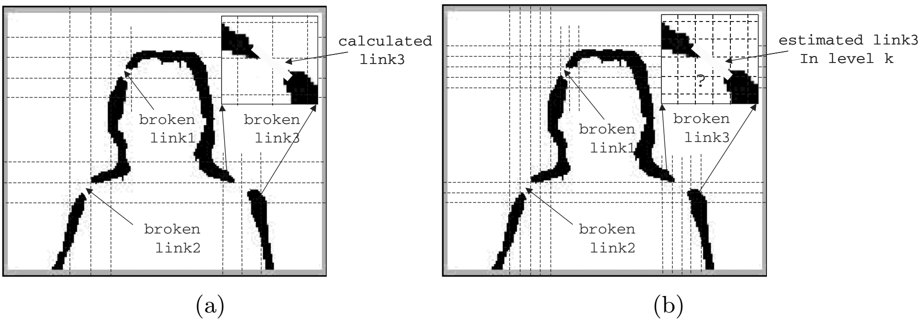


Fig. 5. Continuous Boundary Linkage in Hierarchical Level: (a) at level k, and (b) at level (k+1). The missing edge blocks at level (k+1) can be estimated using the calculated links in level k

reduce the estimation time of a continuous linkage and increase the robustness of the linkage. The following algorithm can effectively find the object boundary.

a) Starting from the upper left corner of the image, we perform raster scanning line by line until we reach the position of an edge block. This position is used as the starting point of boundary tracking. The left-hand rule is initially applied for searching.
b) Check the 3×3 neighborhood of the current position to find an edge block according to the priority of candidates. This procedure is explained in Fig. 6. If we miss a link before we have a complete closed loop, we return to the previous last starting position. We try another search with a different searching rule for the opposite direction of the object boundary.
c) At each edge block position, we define a value, DIR, which stores the direction of the motion for change to the current edge block position along the edge boundary. The next boundary position is searched among candidate edge blocks located in the priority tables, which are determined by DIR.
d) Reduce the edge block size and perform local region growing for the object boundaries. The range of region growing is limited to the previous block size.
e) The process from (b) to (d) is repeated until we obtain the object boundary at pixel accuracy. We can finish it when the result meets a certain condition.

Fig. 7 shows an example of boundary tracking, and summarizes edge positions and directions. We start with the left-hand scan. When we reach the image boundary at No. 18, we switch to the right-hand scan. The next position can be found among the candidates which are located as the priority in Fig. 6, determined by the previous direction(DIR) value. A side branch, which exists from No. 13 to No. 16, should be removed to generate a closed segmentation mask. It sometimes is not required to remove side branches such as antenna, tail, horn and thin-narrow objects.

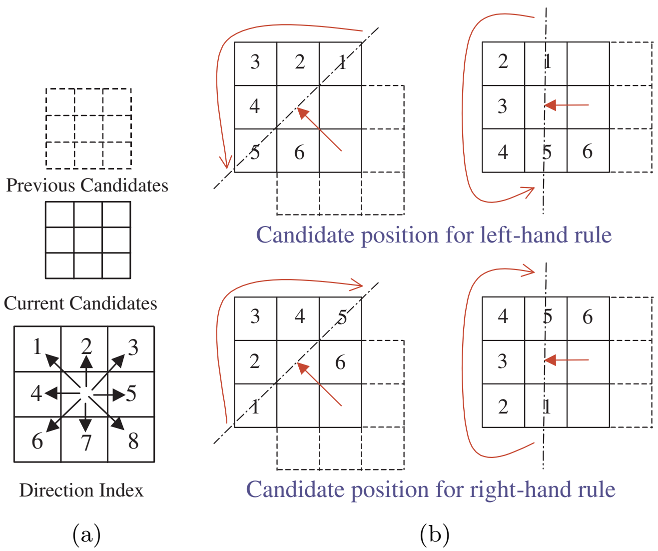
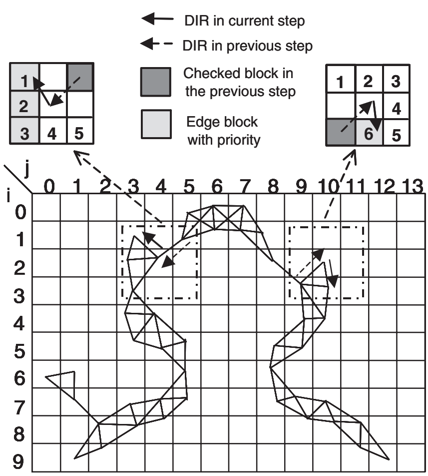


Fig. 6. The Rule of Boundary Tracking: (a) definition of DIR, and (b) two examples for left-hand and right-hand rule



No	Pos	DIR	Scan	No	Pos	DIR	Scan
0	(0,5)	7	left	17	(8,2)	8	left
1	(1,5)	6	left	18	(9,1)	6	left
2	(2,4)	1	left	19	(0,5)	5	right
3	(1,3)	7	left	20	(0,6)	5	right
4	(2,3)	7	left	21	(0,7)	8	right
5	(3,3)	7	left	22	(1,8)	7	right
6	(4,3)	7	left	23	(2,8)	8	right
7	(5,3)	7	left	24	(3,9)	3	right
8	(5,4)	5	left	25	(2,10)	7	right
9	(6,5)	8	left	26	(3,10)	7	right
10	(7,4)	6	left	27	(4,10)	6	right
11	(7,3)	4	left	28	(5,9)	6	right
12	(8,2)	6	left	29	(6,8)	8	right
13	(7,1)	1	left	30	(7,9)	5	right
14	(6,1)	2	left	31	(7,10)	8	right
15	(6,0)	4	left	32	(8,11)	8	right
16	(7,1)	8	left	33	(9,12)		end

Fig. 7. An Example of Boundary Tracking: direction(DIR), position values(Pos), and scan methods(Scan)

2.4 Mesh Construction

It is desirable to have mesh boundaries aligned with object boundaries and edge information of the image to achieve better perceptual quality. There exist several algorithms for constructing mesh structures including uniform triangulation, Delaunay triangulation. In this paper, to construct the mesh structures, we connect two nearest vertices according to the 8-connectivity rule.

2.5 Edge Detection and Region Growing

A contour pixel is defined as a local maximum in the output of an edge detector or the first derivative operator. Since image edges characterize object boundaries, they are important for image segmentation. The goal of image segmentation is to find edge segments whose contours have significant contrast changes.

Region growing is a simple scheme for image segmentation. If the gray-level difference between two adjacent pixels is below a threshold value, those pixels are merged. We can use a statistic of object segments, instead of the pixel difference[4-6].

Both edge detection and region growing are two different aspects of the same process under the assumption of step edges and smooth brightness distribution within regions. Homogeneity for each region may be measured in terms of color, texture, motion, depth, etc. We examine gray-level changes. When we perform local region growing, we use the output of the Sobel edge detector as a seed. The output of the local region growing method is an initial candidate for the object boundary tracking in the next level.

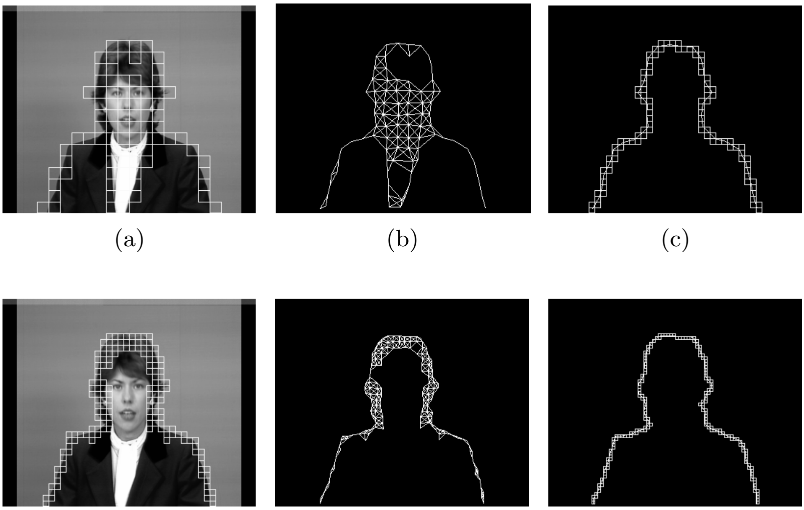
2.6 Region Filling and Image Masking

In order to find the segmentation mask, we fill each region based on object boundary information. To fill a region with a certain gray value, we set each pixel lying on a scan line running from the left edge to the right edge to the same pixel value. The general polygon scan-conversion algorithm [7] handles both convex and concave polygon. In order to obtain image objects, we apply the AND operation with the original image and the segmentation mask.

3 Simulation Results

Computer simulation is performed on video conferencing images of the CIF(352 $\times$ 288) and QCIF(176 $\times$ 144) format. We use several kinds of head and shoulder images such as CLARE, AKIYO, Mother and Daughter.

Fig. 8 shows the hierarchical mesh generation and boundary tracking process for two-level. The object boundary and mesh are refined as proceeding to the next level. In the first hierarchical level, we partition the image into 16 $\times$ 16 blocks and generate meshes in the region of interest. Meshes are mainly distributed around edges. Using the boundary tracking process, we find object boundaries to generate closed contours at each hierarchical level. In the next level, we use 8 $\times$ 8 blocks and smaller meshes to refine the object boundaries. The local region growing method increases robustness of closed contour generation. Since meshes are only distributed around object boundaries, it reduces the computation time in the next level.



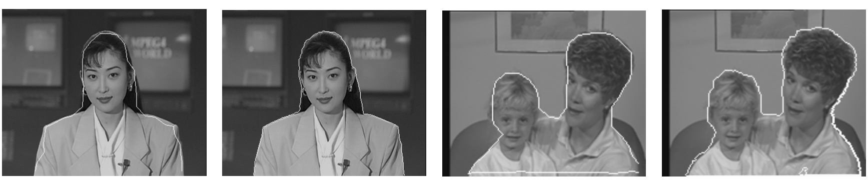
(a) (b) (c)

(d) (e) (f)

Fig. 8. Hierarchical Mesh Generation and Boundary Tracking for Two Level: (a) edge candidate blocks with 16 × 16 block size, (b) generated meshes with (a), (c) boundary meshes after boundary tracking with (b), (d) edge candidate blocks with 8 × 8 block size after local region growing with (c) as local seeds, (e) generated meshed with (d), and (f) boundary meshes after boundary tracking with (e)

(a) (b) (c)

Fig. 9. Segmentation Results for CLAIRE: from 16 ×16 Block Size(a) to 4 × 4 block size(c), where they contain small rectangular block in the beginning of boundary tracking



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

Fig. 10. Segmentation Results for AKIYO and Mother and Daughter: (a) and (c) for 16 × 16 Block Size, (b) and (d) for 1 × 1 block size or one pixel resolution

From Fig. 9 and Fig. 10, we can see the results including two-level meshes of different block sizes for CLAIRE, AKIYO, and Mother and Daughter, respectively. These show the results at each level, where each frame uses a different block size. The square in the Fig. 9 indicates the block size at each level. As proceeding to the next level, we refine object boundaries. We obtain computational reduction by concentrating the meshes on the object boundary.

4 Conclusions

In this paper, we propose a new image segmentation algorithm using hierarchical meshes. Reliable segmentation of objects is obtained by the proposed low complexity method. The proposed algorithm increases the robustness of linkage of object boundaries by overlooking and estimating connectivity at higher hierarchical levels and gives a new modified chain coding method which is applied to natural images, not limited to binary images. Experimental results indicate that reliable segmentation of objects can be accomplished by the proposed low complexity technique, since it reduces the number of processing candidates as proceeding to the next level. We obtain the shape information of the image object from the intermediate results and data that can be used for constructing triangular mesh inside the object. Therefore, the proposed method can be used for continuous sequential processing based on mesh coding on MPEG-4 visual coding standard.

References

1. Westberg, L.: Hierarchical contour-based segmentation of dynamic scenes, IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 14, no. 9 (September 1992) 946-952
2. Altunbasak, Y.: Object-scalable mesh-based coding of synthetic and natural image objects, ICIP'97 (October 1997) 94-97
3. Vincent, L., Soille, P.: Watersheds in digital spaces: An efficient algorithm based on immersion simulations, IEEE Trans. Pattern Analysis Machine Intelligence, vol. 13 (June 1991) 583-598
4. Hojjatoleslami, S.A., Kittler, J.: Region growing: a new approach, IEEE Trans. Im-age Processing, vol. 7, no. 7 (July 1998) 1079-1084
5. Pavlidis, T., Liow, Y.T.: Integrating region growing and edge detection, IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 7, no.3 (September 1992) 225-233
6. Tabb, M., Ahuja, N.: Multiscale image segmentation by integrated edge and region detection, IEEE Trans. Image Processing, vol. 6 (May 1997) 642-655
7. Foley, J.D., (ed.): Introduction to computer graphics, Addison-Wesley (1994)