Unsupervised Segmentation of Stereoscopic Video Objects: Proposal and Comparison of Two Depth-Based Approaches

Klimis S. Ntalianis and Athanasios S.Drigas

Net Media Lab, NCSR Demokritos, Athens, Greece
E-mail: kntal@image.ntua.gr
dr@iit.demokritos.gr

Abstract

In this paper two efficient unsupervised video object segmentation approaches are proposed and thoroughly compared in terms of computational cost and quality of segmentation results. Both methods are based on the exploitation of depth information, estimated for stereoscopic pairs of frames. In particular, in both schemes an occlusion compensated disparity field is initially computed and a depth map is generated. Then a depth segments map is produced by incorporating a modified version of the multiresolution Recursive Shortest Spanning Tree segmentation algorithm (*M-RSST*)*. Next considering the first "Constrained Fusion of Color Segments"* (*CFCS*) *approach, a color segments map is created, by applying the conventional M-RSST to one of the stereoscopic channels. In this case video objects are extracted by fusing color segments according to depth similarity criteria. In the second method an active contour is automatically initialized onto the boundary of each depth segment, according to a fitness function that considers different color areas and preserves the shapes of depth segments' boundaries. Then each point of the active contour is associated to an "attractive edge" point and a greedy approach is incorporated so that the active contour converges to its final position to extract the video object. Several experiments on real life stereoscopic sequences indicate the promising performance of the proposed methods as efficient unsupervised video object segmentation tools. Extensive comparisons are also performed concerning speed and accuracy issues of the proposed schemes.*

Keywords: unsupervised video object segmentation, disparity field, depth map, M-RSST, active contour, attractive edge point, greedy approach, performance evaluation.

1. Introduction

Although block-based video coding standards such as MPEG-1/2 and H.261/3 have served as corner stones to many conventional applications such as video compression/transmission, they meet several limitations when adverting to current needs. During the last decade bursting increase in popularity of multimedia applications based on content interaction, has led to the emergence of new standards as MPEG-4/7. The MPEG-4 standard introduced the concepts of video objects (VOs) and video object planes (VOPs). A VO corresponds to a semantically meaningful object such as a human, a house etc., while a VOP is an arbitrarily shaped region originating from the projection of a VO onto an image plane. These ideas would benefit the implementation of systems for content-based indexing, directive video browsing/summarization, object-based coding, transmission and rate control, or multimedia synthesis and manipulation.

Additionally, although 2-D sequences currently prevail video archives, the demand for stereoscopic sequences is lately increasing since they enhance communications, offering spectacular shows that highly approach real experiences. This is due to the fact that they provide the sensation of depth, which is an inherent characteristic of stereoscopic sequences and its exploitation can lead to the implementation of more effective content segmentation algorithms. However the main disadvantage of stereoscopic video is its increased bandwidth and capacity demands for transmission and storage. For these reasons in this paper a 3-D to 2-D scheme is proposed, where the sequence is captured by a stereoscopic camera, undergoes high-level analysis towards extraction of VOPs and finally an MPEG-4 compatible 2-D sequence is produced.

On the other hand, the visual part standardization phase of MPEG-4 was oriented on developing unsupervised VOP segmentation algorithms [1], but did not result into a globally applicable scheme. This is due to the fact that semantic objects usually consist of multiple regions with different color, texture and motion and although humans can connect the suitable regions almost effortlessly, a machine cannot effectively perform the same task, which still remains a challenging problem. Exceptions can be generally classified into two main categories: (A) The first category includes cases where conditions are known such as graphics, where content description is a-priori available, or the case of video sequences produced using the chroma-key technology. (B) The second category includes cases where VOP segmentation should not or cannot be performed, as in cases of frames with a large number of similar video objects e.g. a crowd, or when the covering region of a video object occupies most of the frame area. In 2D-video, low-level features, such as color, texture and motion information are used as content descriptors [2]. In stereoscopic video however, the problem of content-based segmentation can be addressed more effectively as depth information can be reliably estimated and since each video object usually occupies a distinct depth plane [3].

Several mainly supervised techniques have been proposed in literature for VOP segmentation. Some characteristic examples include: (A) the approach in [4] where multidimensional analysis of several image features is performed by a spatially constrained fuzzy C-means algorithm. (B) The work described in [5], where initially the VOP is detected using a combination of human assistance and a morphological watershed-based segmentation technique. Experimental results are obtained using a low-end Pentium 133-MHz machine. The average time for tracking is 2 s (“Mobi”), 3 s (“Akiyo”), and 5 s (“Foreman”). (C) The works in [6], [7], where an active contour is manually initialized around the object of interest and moves towards the direction of energy minimization. In case of [6] the 2-D GVF computations were implemented using MATLAB code. For an 256 × 256-pixel image on an SGI Indigo-2, typical computation time is 420 s for the GVF forces, which can be reduced to 53 s, when written in C. On the other hand in case of [7], in all demonstrated cases the deformation process took a fraction of a second on a SUN-IPX workstation, for less than 100 vertices. However, in the aforementioned supervised methods the time of user interaction (to group regions or properly initialize an active contour) is generally not estimated, and can be from several seconds to minutes. For these reasons several video applications, cannot include supervised techniques.

On the other hand some unsupervised schemes have been presented, based on information fusion techniques. In [8] a multiscale gradient technique followed by the watershed segmentation algorithm is used for color segmentation, while motion parameters of each region are estimated and regions with coherent motion are merged to form the object. Experimental results show that segmentation and tracking of 12 frames (352×288 pixels) takes about 30 s of CPU time on a Sun workstation. Additionally, as stated in the conclusions, an object can be split into several objects (oversegmentation) according to motion compensation error. In [9] a binary model of the object of interest is derived, the points of which consist of edge pixels detected by the Canny operator. The model is initialized and updated according to motion information, derived by incorporation of *Change Detection Masks* (CDMs) or morphological filtering. Another scheme is presented in [10] where temporal and spatial segmentation are performed. Temporal segmentation is based on intensity differences between successive frames including a statistical hypothesis test, while spatial segmentation is carried out by the watershed algorithm. Information about computational complexity is not provided, while accuracy is heavily influenced by shadow effects, reflections, and noise. Furthermore if there is insufficient motion, parts of objects might be missing in the corresponding VOPs. The work presented in [11] is based on localizing moving VOPs over the MPEG compressed domain. Initially macroblock MVs are utilized to identify the locations of moving VOPs and then DC coefficients are considered so as to achieve finer boundary segmentation. All aforementioned techniques are based on motion information and can produce satisfactory results mainly in cases where background and foreground have different motion parameters and color segments belonging to the foreground VOP have coherent motion. However in real life sequences there are cases where motion information is not adequate for VOP segmentation. These cases include: (A) Insufficient motion between successive frames (slowly progressing scenes). (B) Scenes consisting of more than two objects (foreground/background) having different motion parameters. (C) Cases where color segments of the region of interest do not have coherent motion (non-rigid object) (D) Scenes where the background and the video object of interest have similar motion.

In this paper, two efficient unsupervised video object segmentation techniques are proposed, which take as input stereoscopic video sequences and provide at the output MPEG-4 compatible 2-D sequences. Both approaches share in common the exploitation of depth information, as in many cases most points of a video object have the same depth. According to this consideration, initially a video sequence is analyzed and the disparity field, occluded areas and a depth map are estimated. In a second step, a segmentation algorithm is incorporated to partition the occlusion compensated depth map into homogeneous regions. In this paper a modified version of the multiresolution Recursive Shortest Spanning Tree (M-RSST) segmentation algorithm is adopted. This scheme, apart from accelerating the segmentation process compared to the conventional M-RSST it also prevents oversegmentation, which is not desirable in cases where efficient visual content description should be accomplished. Then, for the first *"Constrained Fusion of Color Segments"* (*CFCS*) scheme, one of the image channels (left or right) is partitioned into color segments by the conventional M-RSST. The *CFCS* scheme is based on the idea that color segments provide accurate object contours, while each depth segment roughly approximates a video object. In the final step of the *CFCS* method, the color map is projected onto the depth map and color segments with similar depth are fused, so that video objects with reliable boundaries are extracted.

The second method is based on active contours, which are automatically initialized on the boundary of each depth segment, in contrast to most existing methods where initialization is manually performed. Firstly the initial points of the active contour are unsupervisedly selected by means of a hybrid polygonal shape approximation technique, which also takes into consideration regional color variations. Secondly each active contour point is associated to an "attractive edge" point that exercises an attraction force to the active contour point. Finally a greedy algorithm is incorporated and the active contour evolves from its initial position to the final minimal energy position to extract the video object. The presented experimental results on real life stereoscopic video sequences indicate the prominent performance of both proposed schemes as content segmentation tools. Furthermore comprehensive comparison tables exhibit the advantages and disadvantages of each scheme regarding computational cost and video object segmentation accuracy.

This paper is organized as follows: In Section 2 depth map estimation and segmentation are compactly presented. Description of the *CFCS* video object segmentation method together with implementation issues are discussed in Section 3 while the active contour-based scheme is investigated in Section 4. Experimental results are presented and analyzed in Section 5 together with a detailed discussion of the advantages and disadvantages of each method. Finally, section 6 concludes this paper.

2. Unsupervised Estimation of the Depth Segments Map

A block diagram describing the modules for producing the proposed depth segments map is illustrated in Figure 1. As it can be observed, the process consists of three main modules: disparity field estimation, occlusion detection/compensation and depth field segmentation.

2.1 Disparity Field Estimation

Let us assume that a point **w** with world coordinates (X, Y, Z) is perspectively projected onto the two image planes of a stereoscopic capturing system, providing point (x_l, y_l) onto the left plane I_l and point (x_r, y_r) onto the right plane I_r. Then the following relations between the two points (x_l, y_l), (x_r, y_r) and depth Z can be obtained [12]:

$$x_r = \lambda \frac{(\lambda s + x_l c)Z - \lambda b c'}{(\lambda c - x_l s)Z + \lambda b s'}, \quad y_r = \lambda \frac{y_l Z}{(\lambda c - x_l s)Z + \lambda b s'} \ , \tag{1}$$

with $s = sin\theta$, $c = cos\theta$, $s' = sin(\theta/2)$, $c' = cos(\theta/2)$

where λ is the focal length and θ is the angle formed by the converging optical axes of the two cameras (fixed camera parameters).

Equation (1) indicates that, if the correspondence between pixel (x_l, y_l) belonging to the left channel and pixel (x_r, y_r) belonging to the right channel is known, then depth Z of point **w** can be straightforwardly estimated. In this direction let us define as $d_x(x_l, y_l) = x_r - x_l$ ($d_y(x_l, y_l) = y_r - y_l$) the horizontal (vertical) left to right disparity for a given point (x_l, y_l) on image plane I_l, where (x_r, y_r) is the corresponding point of (x_l, y_l) on image plane I_r. Using equation (1) the horizontal and vertical disparity fields, d_x and d_y can be expressed as:

$$d_x = d_x(x_l, y_l) = x_r - x_l = \frac{[\lambda(\lambda s + x_l c) - x_l(\lambda c - x_l s)]Z - \lambda b(\lambda c' + x_l s')}{(\lambda c - x_l s)Z + \lambda b s'} \tag{2a}$$

$$d_y = d_y(x_l, y_l) = y_r - y_l = \frac{[\lambda - (\lambda c - x_l s)]y_l Z - \lambda b s' y_l}{(\lambda c - x_l s)Z + \lambda b s'} \tag{2b}$$

If disparity is known, equations (2a) and (2b) reduce to an overdetermined linear system of two equations with a single unknown (Z) and a least-squares method can be applied for estimation of depth Z [13]. In this paper disparity field estimation is performed by minimizing a cost function consisting of a displaced frame difference (DFD) related term, and a spatial consistency criterion for smoothing the disparity field as proposed in [14].

2.2 Occlusion Detection/Compensation and Depth Field Segmentation

In the previous analysis it was assumed that every point (x_l,y_l) of I_l has a corresponding point (x_r,y_r) on I_r. However, due to the different viewpoints of the two cameras, there may be areas of I_l that are occluded in I_r. *Occlusion detection* is accomplished by locating regions of I_l where horizontal disparity continuously decreases with a slope approximately equal to –1 [15]. Then, occluded areas are compensated by keeping disparity field constant and equal to the maximum disparity value of the occluded region [14].

Once an occlusion compensated depth field is estimated the depth segmentation module of Figure 1 is activated. In this paper a modified version of the multiresolution *Recursive Shortest Spanning Tree* segmentation algorithm (*M-RSST*) [16], is incorporated for depth segmentation. The M-RSST recursively applies the RSST to images of increasing resolution and the RSST iteration phase terminates when either the total number of segments or the minimum link weight reaches a target threshold. The minimum link weight is preferable since it results in different number of segments according to the image content. In the proposed modified scheme, the RSST phase of the M-RSST algorithm is applied only to the lowest resolution level of the depth map, and the results are just propagated to the highest level without any adjustment, since exact boundaries of video objects cannot be accurately detected even if higher levels are processed.

3. The “Constrained Fusion of Color Segments” (*CFCS*) approach

The *“Constrained Fusion of Color Segments”* (*CFCS*) approach utilizes color and depth information in order to perform video object extraction. The first step of the proposed scheme includes partitioning of one of the image channels (left or right) into color segments. Now the conventional M-RSST algorithm is incorporated, where initial segmentation at the lowest resolution level is not just propagated but it is also adjusted at higher levels [16]. After producing a color segments map, color segments can be fused according to depth similarity criteria to provide the video objects. The idea of information fusion is justified in our case, since usually video objects are composed of regions located at the same depth plane. However, object boundaries (contours) cannot be identified with high accuracy by a depth segmentation algorithm, mainly due to erroneous estimation of the disparity field, even after it has been improved by the occlusion detection and compensation algorithm. On the contrary, segmentation based on color homogeneity criteria provides reliable object boundaries, but in most cases oversegments a video object into multiple regions. For this reason, VOP segmentation is accomplished by merging together color segments according to depth similarity criteria as described next.

Let us assume that K^c color segments and K^d depth segments, denoted as S_i^c, $i=1,2,\ldots,K^c$ and S_i^d, $i=1,2,\ldots,K^d$ respectively, have been extracted by applying the M-RSST and the modified M-RSST segmentation algorithms on the color and depth intensity planes respectively. Segments S_i^c and S_i^d satisfy the mutual exclusiveness condition, i.e., $S_i^c \cap S_k^c = \varnothing$ for any $i,k=1,2,\ldots,K^c$, $i \neq k$ and, similarly, $S_i^d \cap S_k^d = \varnothing$ for any $i,k=1,2,\ldots,K^d$, $i \neq k$. Let us also denote by G^c and G^d the color and depth segmentation masks respectively, which are given as the union of all color (depth) segments as

$$G^c = \bigcup_{i=1}^{K^c} S_i^c, \quad G^d = \bigcup_{i=1}^{K^d} S_i^d \tag{3}$$

Assuming two different segmentation masks over the same lattice, say G^c and G^d, an operator $p(\cdot)$ can be defined, which projects a segment of the first segmentation mask, i.e., S_i^c, on the second mask, i.e., G^d,

$$p(S_i^c, G^d) = S_k^d \;\text{ such that }\; S_k^d = \arg\max_{j=1,\ldots,K^d} a(S_j^d \cap S_i^c), \tag{4}$$

where $a(\cdot)$ is the area, i.e., the number of pixels, of a segment. Equation (4) indicates that the projection of a color segment, say the i-th (S_i^c), onto the depth segmentation mask, G^d, returns that depth segment, S_k^d, among all K^d available, with which the respective color segment S_i^c has the maximum area of intersection. As a result, the projection operator actually maps a color segment to a depth segment, i.e., $S_i^c \to S_k^d$.

Using the projection operator, each color segment of G^c is associated to one depth segment and all color segments belonging to the same depth segment compose a group. More specifically, K^d classes are created, say C_i, $i=1,2,\ldots,K^d$, each of which corresponds to a depth segment S_i^d and contain all color segments that are projected on S_i^d, i.e.,

$$C_i = \{S_j^c : p(S_j^c, G^d) = S_i^d\}, \quad i=1,2,\ldots,K^d \tag{5}$$

Using classes C_i, the final video object segmentation mask is obtained by merging all color segments, which belong to the same class C_i. Let us denote as *G*, this final segmentation mask consisting of *K*=*K*^{d} segments, say S_i, $i=1,2,\ldots,K$. These segments are generated as:

$$S_i = \bigcup S_g^c, \quad i=1,2,\ldots,K \;\text{ such that }\; S_g^c \in C_i \tag{6}$$

while the final mask *G*, is given as the union over all segments S_i, $i = 1,2,\ldots,K$, that is

$$G = \bigcup_{i=1}^{K} S_i \tag{7}$$

Equation (7) denotes that color segments are merged into $K=K^d$ new segments according to depth similarity. A graphical example of the proposed *CFCS* approach is presented in Figure 2, which describes the case of a shuttle. In Figure 2(a) the color segments mask can be viewed while in Figure 2(b) the respective depth segments map is presented. In Figure 2(c) color segments are projected onto the depth map and each color segment is associated to one depth segment according to the projection operator $p(\cdot)$. The final video objects segmentation mask can be observed in Figure 2(d).

4. The active contour scheme

Two-dimensional deformable models, also known as "active contours" or "snakes", were originally proposed by Kass et. al. [17]. The energy functional of an active contour can be formed as:

$$E^*_{ac} = \int_0^1 E_{ac}(v(s)) = \int_0^1 [E_{\text{int}}(v(s)) + E_{image}(v(s)) + E_{con}(v(s))]ds \tag{8}$$

where E_{int} represents the internal deformation energy of the active contour due to bending or discontinuities, E_{image} expresses the image forces and E_{con} allows for external constraint forces.

In the proposed scheme the energy to be minimized can be expressed as:

$$E^*_{ac} = w_1 E_{\text{int}} + w_2 E_{edge} \tag{9}$$

where E_{int} is the internal energy and E_{edge} is a kind of constraints energy originating from the edge map of each depth area. Furthermore w_1 and w_2 weight the two energy factors of equation (9) and different weights' adjustment can produce a wide range of active contour behaviors.

4.1 Calculation of Energy Factors

The internal spline energy can be written as:

$$E_{\text{int}}=(\alpha(s)|v_s(s)|^2 + \beta(s)|v_{ss}(s)|^2)/2 \tag{10}$$

The above equation contains a first-order term that will have larger values at curves' gaps and a second-order continuity term that will be larger at curves' bendings. The values of α and β control the stretching and bending potentials at each point of the active contour.

In this paper the energy due to stretching ($v_s(s)$) for two neighboring active contour points $P_{i-1}=(x_{i-1},y_{i-1})$ and $P_i=(x_i,y_i)$ is computed using the conventional formula:

$$|v_s(s)|^2=|dP_i/ds|^2 \approx |P_i-P_{i-1}|^2=(x_i-x_{i-1})^2+(y_i-y_{i-1})^2 \tag{11}$$

On the other hand a new method for estimating bending energy (term $v_{ss}(s)$ of Equation (10)) is proposed, which simplifies the handling and usage of the bending term. More specifically let us suppose that the bending energy of a random active contour point P_i should be calculated. For this reason the previously processed point P_{i-1} and the next point to be processed P_{i+1} are also considered (Figure 3). Using these three points two lines are formed, line ε_1 defined by points P_{i-1} and P_i and line ε_2 defined by points P_i and P_{i+1}. Then the angle ω between ε_1 and ε_2 can be calculated by:

$$\omega = \arctan\left(\frac{\lambda_2 - \lambda_1}{1 + \lambda_1 \lambda_2}\right) \tag{12}$$

where $\omega \in (0 - \pi)$ and λ_1, λ_2 are the direction coefficients of lines ε_1 and ε_2 respectively. As illustrated in Figure 3 two cases exist, in case 1 the estimated angle ω is inside the triangle $P_{i-1}\hat{P_i}P_{i+1}$ ("in"-angle) while in case 2 the estimated angle ω is outside ("out"-angle). Thus in case 1 the angle θ to be used is the computed angle ($\theta=\omega$) while in case 2, the angle θ is given by:

$$\theta=|180^0-\omega| \tag{13}$$

Then the bending energy is estimated using the following equation:

$$|v_{ss}(s)|^2 = \left(\frac{1}{R\theta}\right)^2 \tag{14}$$

where R is a normalization parameter that maps angle θ to the interval [0 255].

On the other hand, considering energy factor E_{edge}, the idea of attractive fields/forces for active contour models has been presented in literature [6], [18]. Attractive fields/forces usually exercise forces onto active contours so as to evolve faster to the minimum energy positions, while avoiding getting trapped to local minima. However the generation of a field is highly time consuming and thus it is difficult to be included into fast video object segmentation schemes. For example, it takes approximately 53 seconds for the GVF field to be generated, considering an image of size N=256×256 pixels, for code written in C and assuming about 250 iterations (estimation for SGI Indigo-2 machine) [6]. For this reason and considering the special conditions of our problem (that depth segment contour is near to video object contour), in our scheme an "attractive edge" points idea is proposed, which can be implemented much faster than estimating an attraction field and can provide satisfactory results. An "attractive edge" is a pixel of the edge map estimated inside the visual content defined by a depth segment, which is associated to an active contour point and attracts it. Then the constraints energy E_{edge} is estimated according to the distance of each active contour point from its "attractive edge". Both edge map generation and detection of "attractive edge" points are described next.

Let ds_i be the i-th depth segment of a depth segments' map, the area of which defines a specific visual content Vds_i onto the respective image channel it has been derived from. Then edge detection is performed inside Vds_i and the estimated edge map is filtered so that isolated edges are discarded, producing a filtered edge map e_{init_f}. This edge map is used for selection of the "attractive edge" points according to the following iterative algorithm:

- Step 1: Selection of an active contour point P_i.
- Step 2: Detection of the specific edge point of e_{init_f} that has minimum distance from point P_i.
- Step 3: If minimum distance is greater than a threshold d_{thr} then proceed with the next active contour point and repeat process from Step 1.
- Step 4: Else, mask adaptation (3×3) centered at the minimum distance edge point.
- Step 5: Computation of the number of edge points inside the mask (NF).

- Step 6: If $NF < T_{num}$, where T_{num} controls the density of edges inside each region, then detection of the second minimum distance edge point and iteration from Step 3, else current minimum distance point becomes "attractive edge" for the processed active contour point.

If none of the edge points or only edge points located farther than the distance threshold d_{thr} satisfy the "$NF \geq T_{num}$" condition then the active contour point is not associated to an "attractive edge" and moves only according to E_{int} during convergence. The process is terminated after all active contour points are considered. Finally the energy E_{edge} for an active contour point (x_i , y_i) is estimated by:

$$E_{edge}(x_i , y_i) = B*d_i(x_i , y_i) \tag{15}$$

where $d_i(x_i , y_i)$ is the Euclidean distance of the active contour point from its "attractive edge" and B is a constant that maps d_i in the interval [0 255].

4.2 Unsupervised Active Contour Initialization

In the proposed scheme an active contour is unsupervisedly initialized onto the boundary of each depth segment, in contrast to most existing schemes where active contour initialization is manually performed. The method can be divided into three main modules: (a) Detection of the boundary points for each depth segment, (b) Determination of a specific ordering for the detected points and (c) Selection of the initial active contour points from the set of ordered boundary points.

Boundary points of a depth segment are those points located at the border of the depth segment and can be directly found. However, as the set of these points does not satisfy any ordering, they cannot be straightforwardly used for active contour initialization, since energy calculation requires the formation of ordering for the points comprising the active contour. For this reason ordering is established by finding the neighbors of each boundary point, so as to efficiently describe the shape of the depth segment under consideration.

Towards this direction let us denote by P_i=(x_i,y_i) an initially selected random boundary point named *"current point"*. Then starting from this point and moving in a clockwise manner, an order of the boundary points is created. In particular the process is iterative and for every iteration the following actions are performed:

i) Mask adaptation of size 3×3, centered at *"current point"*.

ii) Detection of those points inside the mask that belong to the depth boundary.

iii) Selection of the nearest neighbor point, which becomes *"current point"*.

The process is terminated when starting from an initial point (x_i,y_i) it results to the same point (closed loop). Finally the set of ordered boundary depth segment points is denoted as SD= $\{P_1, P_2,, \ldots, P_N\}$, where P_1 is the initial point and P_N the final point ($P_{N+1} \equiv P_1$).

4.2.1 Initial Points Selection

After finding set SD, links are generated between the points of this set according to the estimated order, i.e. point P_1 is linked to point P_2, point P_2 is linked to point P_3, … and point P_N is linked to point P_1, resulting in the $P_1P_2 \ldots P_NP_1$ polygon. However, this polygonal line cannot be used as initial active contour since a large computational cost would be induced during convergence (due to the large number of nodes). For this reason and since in the specific application there is high redundancy in the set of the depth boundary points, only a small part of these points should be selected to form the initial active contour. Towards this direction linear piecewise approximation techniques can be adopted to produce a polygon resembling the original depth boundary, while drastically reducing the number of depth boundary points. More particularly the target of the proposed scheme is to select the minimum number of points that best describe the shape of the depth contour, according to a threshold. The threshold affects convergence time since the number of active contour points is strongly related to this threshold. However as the final polygon (initial active contour) depends on the order that points are discarded and in order to keep points belonging to different areas, the mean color difference *MC* between triplets of adjacent points guides the process. Considering that P_{i-1}, P_i and P_{i+1} are three neighboring points and the image is in RGB format, the mean color difference MC_i for point P_i is defined as:

$$MC_i = \frac{\left(\left|R(P_i) - R(P_{i-1})\right| + \left|R(P_i) - R(P_{i+1})\right|\right) + \left(\left|G(P_i) - G(P_{i-1})\right| + \left|G(P_i) - G(P_{i+1})\right|\right) + \left(\left|B(P_i) - B(P_{i-1})\right| + \left|B(P_i) - B(P_{i+1})\right|\right)}{2} \tag{16}$$

In the proposed scheme an iterative merging technique is incorporated, where starting from polygon $P_1P_2\ldots P_NP_1$ and by discarding points according to color homogeneity and shape distance criteria, the initial active contour is produced. More specifically let $SD = \{P_1, P_2, \ldots, P_N\}$ denote the ordered set of points comprising the nodes of the constructed polygon, like the one depicted in Figure 4. In order to effectively discard nodes of the polygon it is essential to define error criteria that measure the quality of fitness of the polygon. In this direction and without loss of generality, if point P_2 is discarded then distance $df_2=|P_2-K_2|$ can be used as the approximation error of the curve. In the case under study a maximum approximation error df_{max} is adopted as fitness criterion. Then the iterative process described below is activated:

Step 1. For each point P_i of set *SD*, find MC_i (Eq. 16) and calculate the mean distance D_i from its two adjacent points P_{i-1}, P_{i+1}.

Step 2. Select point of set *SD*, say P_i, with the minimum MC_i value and calculate df_i.

Step 3. If $df_i > df_{max}$, or $D_i > D_{max}$, where D_{max} is a constant regulating the gaps between adjacent points, select the point of set *SD* that possesses the next minimum *MC* value and repeat Step 3.

Step 4. If $df_i \leq df_{max}$ AND $D_i < D_{max}$, discard P_i, recalculate the *MC* and *D* values for the rest of the points in the neighborhood of P_i, update set *SD* and repeat process from Step 2.

Step 5. If no point satisfies the "$df_i \leq df_{max}$ AND $D_i < D_{max}$" condition then terminate process.

After termination of the process the remaining points constitute the initial points of the active contour. In our experiments values of df_{max} and D_{max} are properly selected so that a small number of points is kept (about 2-2.5% of the points of initial set *SD*) to reduce the computational cost, while providing a well-positioned initial active contour for video object segmentation.

4.3 Greedy Algorithm: A Fast Approach Towards Active Contour Convergence

After initialization, convergence of the active contour to the minimum energy position is investigated. Many solutions to this problem have been proposed in literature. The solution proposed in [17] can be derived by incorporating variational calculus techniques. However it has been shown in [19] that these techniques present several problems such as: (A) the solution can reveal numerical instabilities and (B) points show a tendency to pile up on strong portions at an edge contour. Amini et al. [19] proposed an algorithm for the active contour model using dynamic programming, an approach that is more stable and allows the inclusion of hard constraints in addition to the soft constraints inherent in the formulation of the functional. This method though is slow having complexity $O(nm^3)$, where n is the number of points of the active contour while m is the size of the region, inside which a point can move during a single iteration.

In this paper a greedy algorithm approach is adopted [20], which allows the active contour to quickly converge to its final position. Performance of this approach is comparable to dynamic programming, retains stability improvements and flexibility and allows the inclusion of hard constraints, while simultaneously reduces execution time to an order of $O(nm)$. The quantity being minimized during the greedy algorithm phase is given by Equation (9).

In order to minimize this overall energy, each point of the active contour starts moving on a grid. The grid is centered at the processed point and covers the 3×3 neighborhood. The energy functional is computed for every point in this area and the active contour point moves to the position of the grid that: (a) has the least energy compared to the rest seven pixels of the grid and (b) has less energy than the current position of the active contour point. If these conditions are not satisfied then the active contour point stays at its current position and the algorithm continues with the next point. The algorithm terminates if no position changes occur during an iteration.

5. Experimental Results

In this section, performance of the proposed schemes is investigated while a detailed comparison of the video object segmentation results is also provided. Results have been obtained using stereoscopic pairs taken from the 3-D stereoscopic television program “Eye to Eye” [21], of total duration 25 minutes (12,739 frames at 10 frames/sec). The sequence was produced in the framework of the ACTS MIRAGE project [22] in collaboration with AEA Technology and ITC. Studio shots were executed using Europe’s stereoscopic studio unit, which was developed jointly by AEA Technology and Thomson Multimedia within the earlier RACE DISTIMA project [23], while location action shots were captured using a special lightweight and rugged stereo camera built for the ITC by AEA Technology.

5.1 Depth Segments Map Generation

Initially a depth segments map is produced for each examined stereo pair. After examining several stereo pairs and for presentation reasons, three characteristic stereo pairs are selected, to illustrate the performance of the proposed algorithms. In particular the left channels are depicted in Figures 5(a), (c) and (e) while the respective right channels are shown in Figures 5(b), (d) and (f). In the first step an occlusion compensated disparity field is estimated for each stereo pair and a depth field is straightforwardly generated as described in section 2. The occlusion compensated disparity field and the corresponding depth map of the first stereo pair are shown in Figures 6(a) and (b) respectively. Similar results for the other two stereo pairs are illustrated in Figures 6(c)-(d) and 6(e)-(f).

Next the modified M-RSST segmentation algorithm is applied to each depth field to provide a depth segments map. More particularly multiresolution decomposition is performed, so that a hierarchy of depth fields $I(0)$, $I(1)$,…, $I(L_0)$ is constructed. $I(1)$ is of size $M_0/2$ x $N_0/2$, while $I(L_0)$ of size $M_0/2^{L_0}$ x $N_0/2^{L_0}$, where $M_0 \times N_0$ is the size of the full resolution depth field. In our experiments the lowest resolution level has been selected to be L_0=3 (i.e., block resolution of 8×8 pixels). Afterwards, according to the modified M-RSST the RSST phase of the algorithm is applied only to the lowest resolution level and the results are just propagated to the highest level, without performing any adjustment at the intermediate levels. This modification led to a more than 5 times acceleration in the performed experiments, compared to the conventional M-RSST algorithm and more than 300 times compared to the RSST, providing at the same time a good quality of depth segmentation results. Depth segmentation results for depth fields of Figures 6(b), (d) and (f) are presented in Figures 7(a), 7(c) and 7(e) respectively. As it can be observed, boundaries present blocky artifacts, something that was expected. Here it should be mentioned that boundaries of video objects cannot be accurately detected even if higher levels are processed, which justifies the complexity-reducing modification of the conventional M-RSST algorithm. To better elucidate this idea, results provided by the conventional M-RSST algorithm are also depicted in Figures 7(b), 7(d) and 7(f). Furthermore, from the results of Figure 7 it becomes clear that each depth segment roughly approximates a video object.

5.2 Segmentation Results of the CFCS Approach

Considering the *CFCS* approach color segments are fused based on depth homogeneity criteria. For this reason, after producing a depth segments map, a color segments mask is also produced for each stereo pair. Towards this direction the conventional M-RSST segmentation algorithm is incorporated, where results of the lowest resolution level are not just propagated but also adjusted at higher levels [16]. This is essential as color segments accurately detect the boundaries of video objects. The multiresolution color segmentation process is illustrated in Figure 8 for frame of Figure 5(a). More specifically again an initial lowest resolution level of L_0=3 has been adopted. In this figure for better apprehension and evaluation reasons, each color segment is an area surrounded by a white contour and not a fragment possessing the average color intensity of the area. In particular, Figure 8(a) presents the results of the lowest resolution level (L_0=3), where block resolution around the object boundaries is evident. Then, initial segmentation is propagated to the following resolution level by splitting each boundary pixel (or 8×8 block) into four new segments (of size 4×4) according to the M-RSST algorithm. This is presented in Figure 8(b). As shown, the total number of segments for the next iteration of M-RSST is considerably reduced compared to the initial number of segments of the conventional RSST at the same resolution level. Thus, a significant reduction of the computational complexity is achieved. Color segmentation at resolution level 2 (L_0=2) is depicted in Figure 8(c), where, as observed, object boundaries have been refined. Similarly, Figure 8(d) illustrates the boundary pixels (or 4×4 blocks) that are split into four new segments of size 2×2, while Figure 8(e) shows the final color segments mask. Similarly color segmentation results for Figures 5(c) and (e) are presented in Figures 10(a) and 11(a) respectively.

As observed in all cases, color segments accurately detect the boundaries of video objects but oversegment them into multiple regions. On the other hand each depth segment roughly approximates a video object providing an incorrect contour. For this reason color segments are projected onto the depth segments map and they are fused according to depth similarity criteria as described in section 3. More specifically in Figure 9(a) projection of the color segments mask of Figure 8(e) onto the depth map of Figure 7(a) can be observed. White lines correspond to the contours of color segments, while each region with different color represents a different depth segment. Then in Figure 9(b) fusion of color segments belonging to the same depth segment is performed. Finally in Figures 9(c) and 9(d) the two video objects are extracted (foreground and background). Extraction of the video objects of Figures 5(c) and (e) are also presented in Figures 10 and 11 respectively, where in 10(b), 11(b) projection of the color segments masks onto the depth maps is performed and in Figures 10(c), 11(c) fusion of color segments belonging to the same depth is accomplished. Finally the extracted foreground and background objects for the two cases can be seen in Figures 10(d), 10(e) and 11(d), 11(e) respectively. As it can be observed, the results are very promising even in cases of complicated backgrounds or of different numbers of video objects. Furthermore it should be mentioned that in case of Figure 5(e) extracted video objects have not been very accurately detected due to inaccurate depth segmentation results, an issue which is also evident at the upper left part of the man-object (Figure 10(d)). As we will see next, this issue has greater impact in the segmentation fusion approach, since final segmentation heavily depends on the estimated depth segments.

5.3 Segmentation Results of the Active Contour Approach

According to the second approach an active contour is initialized on the boundary of each depth segment to perform VOP segmentation. Initialization is accomplished using the proposed unsupervised technique described in subsection 4.1, where firstly an order of the detected boundary points of a depth segment is estimated and then a fitness function is incorporated so that only a small portion of these points is kept. In the polygonal approximation scheme and after several experiments df_{max} was set equal to 5 and D_{max} equal to 170 in order to avoid large gaps in the initial active contour. Then active contour segmentation results in case of Figure 5(a) are presented in Figure 12. In particular the depth contour of the foreground video object was comprised of 794 points, but only 14 points were used to form the initial active contour, which is depicted in Figure 12(a). Next edges were estimated for the visual content Vds_1 defined by the foreground depth segment and the resulting edge map was filtered (so that isolated edges were discarded) to produce e_{init_f}. Afterwards each point of the initial active contour was associated to an “attractive edge” of e_{init_f} as described in subsection 4.1, where T_{num} and d_{thr} were experimentally set equal to 4 and 11 respectively. The filtered edge map e_{init_f} is depicted in Figure 12(b), where initial points are sketched as green squares while “attractive edges” are marked as red circles. As it can be seen only 13 out of the 14 points are associated to an attractive edge and the remaining point moves only according to the internal energy E_{int}. Afterwards the greedy approach is activated so that the initial active contour converges to its final position. In case of multiple video objects, convergence of the multiple active contours is fully paralleled, since each active contour moves independently. In Figure 12(c) video object detection is accomplished after 57 iterations. In this case and since the existing depth segments are two (foreground - background), one active contour is enough. Finally the two extracted video objects are shown in Figures 12(d) and 12(e) respectively.

Similar results for Figures 5(c) and 5(e) can be seen in Figures 13 and 14 respectively. In particular in case of Figure 5(c) and for the left video object (man) 918 points where detected 15 of which were kept to comprise the initial active contour, while in case of the right video object (woman) 867 points were found 21 of which were kept. On the other hand concerning video object of Figure 5(e) 823 points were found, 20 of which where finally kept, after the polygonal approximation method, to form the initial active contour. Initial active contours for Figures 5(c) and 5(e) can be seen in Figures 13(a) (both video objects) and 14(a) respectively, while the edge maps e_{init_f} containing initial points and attractive edges are shown in Figures 13(b) and 14(b). As it can be observed in Figure 13(b) only 10 out of the 15 points are associated to an "attractive edge" for the left video object, while in case of the right video object 20 out of 21 points have an "attractive edge". In case of Figure 14(b) all 20 points have their respective "attractive edge", two points of which are associated to the same "attractive edge". Finally active contours' convergence after 77 iterations for the left object and 64 iterations for the right object is presented in Figure 13(c). Furthermore video objects' extraction can be seen in Figures 13(d) and 13(e). Similarly in case of Figure 5(e) and after 59 iterations, the active contour converges to its minimal energy position depicted in Figure 14(c). Finally the foreground and background video objects are depicted in Figures 14(d) and 14(e) respectively.

As observed only a small percentage of points (~2% on average) is considered in all cases, for construction of the initial active contours, leading to a substantial reduction of the computational complexity. Furthermore the active contour scheme generally provides a good segmentation quality even in cases of multiple video objects or complex visual content. On the other hand the problem of inaccurate depth segmentation is better addressed by this technique, compared to the segmentation fusion approach.

5.4 Comparison of the Two Unsupervised Video Object Segmentation Schemes

In this subsection comparison of the proposed video object segmentation schemes is performed in terms of computational complexity and accuracy. Considering accuracy, the first approaches for objective evaluation of video object segmentation results can be found during standardization of the MPEG-4 [24], within the core-experiment on automatic segmentation of moving objects. The objective evaluation scheme in [25] uses an a-priori known 2-D shape in order to appraise the estimation result, while in [26] additional geometric features like size and position of a video object, as well as the average color within a VOP area are also considered. The aforementioned approaches do not distinguish between a lot of small deviations between the estimated and original mask and a few but larger deviations. Both cases can lead to the same value of spatial accuracy, although they are visually different. In order to confront this problem Mech and Marques have proposed a spatial accuracy and temporal coherence evaluation scheme based on the mean and standard deviation of 2-D shape estimation errors [27]. According to this scheme for each pixel i of the original contour, distance d_i to the estimated video object contour is measured and the mean and standard deviation of these distances is calculated. Assuredly, attention should be paid in two important issues: (a) The technique to find the correspondence between pixels of the original contour and the estimated contour is position sensitive, thus different distances can be found for different parsing direction of the original contour and (b) Wrong correspondences and thus wrong distances can be found, due to wrong orientation of the assignment vectors.

In our approach the scheme presented in [28] is adopted, where the quality of segmentation masks is evaluated by means of algorithmically computed figures of merit, weighted to take into account the visual relevance of segmentation errors. This scheme assumes the existence of a perfect mask so that evaluation and ranking of the performance of segmentation algorithms can be accomplished. For this reason a spatial quality measure (SQM) with higher perceptive meaning is build, by considering that some errors are visually more important than others. According to this scheme the following definitions and assumptions are made:

- Two types of error exist: (a) missing foreground points (MF) and (b) added background points (AB), with different visual importance.
- Wrong pixels have bigger visual relevance when located further away from the border of the reference mask.
- MF errors are usually more tolerable than AB errors at distances very near to the reference mask, since in the latter case background noise is added to the video object (halo effect), while in the former case a slightly thinned but usually acceptable VOP is extracted.
- Moving away from the border, MF errors attain greater relevance, since a bigger part of the object is being missed.
- Although AB errors also increase their relevance at far locations, this increment tends to stabilize with the distance.

Then the absolute spatial quality measure is defined as:

$$\textbf{SQM}= \sum_{d=1}^{D_{FG\max}} \mathrm{w}_{\mathbf{MF}}(d)\cdot Card(\mathrm{R}_d \cap E^c) + \sum_{d=1}^{D_{BG\max}} \mathrm{w}_{\mathbf{AB}}(d)\cdot Card(R_d^c \cap E) \tag{17}$$

where $\boldsymbol{E}$ is the estimated mask, $\boldsymbol{R}$ the reference mask, $(^c)$ indicates the complement of a set, $\mathrm{D}_{\mathrm{FGmax}}$ and $\mathrm{D}_{\mathrm{BGmax}}$ are the maximum distances for the MF and AB pixels respectively and sets $\boldsymbol{R_d}$ are defined as:

$$R_i \;= \{\mathrm{x}|\mathrm{x}\in R,\ d(x,R^c)= i\} \tag{18a}$$

$$R_i^c = \{\mathrm{x}|\mathrm{x}\in R^c,\ d(x,R)= i\} \tag{18b}$$

Furthermore a diagram of the weighting functions $\mathrm{w}_{\mathbf{MF}}(d)$ and $\mathrm{w}_{\mathbf{AB}}(d)$ is depicted in Figure 16. Finally for normalization purposes the SQM value is divided with the number of pixels of mask $\boldsymbol{R}$. For the 4 foreground video objects of Figures 5(a), 5(c) and 5(e) the manually created reference masks are shown in Figures 15(a), 15(b), 15(c) and 15(d) respectively.

Comparison of the two proposed techniques for the extracted foreground video objects is provided in Tables I, II, III and IV. The first column is occupied by the perfect mask and the two estimated masks to be compared (fusion mask, active contour mask). The second column refers to the number of pixels that each mask contains. In the third column, the number of pixels of each mask that belong to the VOP area (based on the perfect mask) is presented. The fourth column contains factor *CP*, which expresses the percentage of the perfect video object mask covered by the estimated mask and is calculated by:

$$CP = \frac{\text{Number of estimated pixels of video object}}{\text{Number of pixels of video object}} \tag{19}$$

The fifth column contains factor *AP*, which expresses the percentage of the estimated mask that covers area of the video object and is calculated by:

$$AP = \frac{\text{Number of estimated pixels of video object}}{\text{Number of pixels of the estimated mask}} \tag{20}$$

The sixth column contains the value of SQM while in the seventh column the SQM′ (logarithmic scale of SQM) is provided that is estimated by:

$$SQM' = 10\log\left(\frac{1}{1+SQM}\right) \tag{21}$$

Finally in the eighth column execution times are also provided. Execution times were estimated for a Pentium 2.6 GHz processor, 512 MB RAM and ANSI C implementation of the various schemes.

As it can be observed from Table I, which refers to the foreground video object of Figure 5(a) (woman), both techniques provide very good results with the best results given by the fusion technique. Here it should be mentioned that for SQM′ values between 0 and -2.5 dB the extracted video objects have highly accurate contours. For SQM′ in the range of [-5 –2.5) dB video objects are generally very well detected but there are also small visually observable areas that are covered and do not belong to the video object or small regions not covered and belonging to the video object. Furthermore for values in the range [-7.5 –5) dB video objects are well detected but now the visually observable missing foreground or added background areas are larger. Finally for values less than –7.5 dB segmentation accuracy decreases, providing in many cases low quality results. Returning to Table I, most of the video object area is covered by the estimated masks (~ 95.2 % on average) while the greatest part is covered by the mask of the fusion scheme (95.5 %). On the other hand each of the estimated masks mainly covers video object area (~ 97.5 % on average). Finally SQM′ is worse for the mask estimated by the active contour scheme. This is due to the fact that there are no edges at the lower border of the video object under consideration. On the other hand in terms of time the active contour approach is faster achieving segmentation time of 0.55 sec against 1.06 sec of the fusion scheme, leading in about 48% reduction of segmentation time compared to the fusion scheme.

In Table II results for the left foreground video object of Figure 5(c) (man), are presented. In particular in this case both the active contour and *CFCS* approaches provide similar SQM′ values. This is due to inaccurate depth segmentation, as the upper left part of the image is contained in the same depth segment with the video object. In terms of time, the active contour approach is faster giving an improvement of about 61 % compared to the fusion scheme.

In Table III results for the right foreground video object of Figure 5(c) (woman) can be observed. More specifically in this case the segmentation fusion method gives more accurate results (-2.33 SQM′ / 99.5% coverage). However the active contour scheme is again faster with execution time 0.69 sec against 1.9 sec performed by the fusion scheme (about 64 % improvement).

Finally in Table IV results for the foreground video object of Figure 5(e) (car) are shown. In particular the active contour technique achieves lower SQM′, providing however worse coverage of the video object area (93.7 %). On the other hand the fusion scheme provides a very high SQM′ value (-15.23 dB) and an *AP* equal to 68.9 %. Again it is evident that inaccurate depth segmentation affects more the *CFCS* approach.

For completeness reasons the overall performance of the *CFCS* and active contour approaches in terms of segmentation accuracy (SQM′) and complexity (execution time) are illustrated in Figures 17(a) and (b) respectively, for more than 500 video objects. As it can be observed the *CFCS* approach achieves better accuracy in most cases, where accurate depth segmentation is accomplished, providing mean SQM′=-3.92

while in case of the active contour approach the mean SQM´=-5.47. On the other hand the active contour scheme is faster than the *CFCS* scheme (Figure 17(b)), providing an average execution time equal to 0.94 sec (1.93 sec for the CFCS approach), which leads to a complexity improvement of about 51 %.

6. Conclusions

In this paper two depth-based unsupervised schemes for VOP segmentation have been proposed that can complement several existing methods based on motion information. Both schemes receive as input stereoscopic sequences, providing MPEG-4 compatible 2-D sequences in their output. In particular in both schemes stereoscopic pairs are initially analyzed and depth segments maps are produced by applying a modified version of the M-RSST segmentation algorithm. Afterwards the first scheme (*CFCS*) performs VOP segmentation by appropriately combining color and depth descriptors in a hierarchical framework. Towards this direction one of the stereoscopic channels is partitioned into color segments by utilizing the conventional M-RSST algorithm. Color segments provide very accurate boundaries but oversegment a VOP into multiple regions. On the other hand depth segments efficiently describe semantic visual content, but usually blur VOPs' boundaries. For this reason, the proposed *CFCS* algorithm merges color segments based on depth constraints. The main issue of *CFCS* is its computational cost, which is worse compared to the active contour technique, while its main advantage is its very good performance in terms of segmentation quality. On the other hand the active contour scheme is based on the idea of energy minimization. An active contour is initialized onto the boundary of each depth segment by utilizing information provided by the depth contour. Initialization is unsupervisedly performed, in contrast to most existing supervised schemes. In the next phase, and since usually depth segments’ boundaries are near to the VOPs boundaries, each point of the active contour is associated to an “attractive edge” point according to distance and color criteria, so that the active contour evolves faster and reliably towards the boundary of the VOP. Finally an efficient and cost-effective greedy approach is incorporated during convergence.

Both schemes provide promising results and can support a new range of capabilities in terms of access, creation, manipulation or editing of visual content. In particular, content segmentation enables for more effective image/video coding, permits sophisticated content-based queries on multimedia databases and enables coders to perform rate control according to semantic information.

Furthermore, although the proposed algorithms are designed for unsupervised content segmentation, they can also be included in interactive applications by making minor modifications, resulting in semi-automatic schemes that would provide more accurate segmentation results. Finally, both approaches are not limited to exploiting depth information but can also use any kind of information that roughly describes a semantic object, based each time on the specific application.

7. Acknowledgments

The authors wish to thank very much Dr. Chas Girdwood, the project manager of the ITC (Winchester), for providing the 3D video sequence “Eye to Eye”, which was produced in the framework of ACTS MIRAGE project. Furthermore the authors want to thank very much Dr. Siegmund Pastoor of the HHI

(Berlin), for providing the video sequences of the DISTIMA project. This work is supported by the EU project MESH "Multimedia Semantic Syndication for Enhanced News Services" (FP6-027685).

8. References

[1] ISO/IEC JTC1/SC29/WG11, MPEG-4 Visual Final Committee Draft, Dublin, Ireland, July 1998.

[2] B. Furht, S.W. Smoliar and H. Zhang, *Video and Image Processing in Multimedia Systems.* Kluwer Academic Publishers, 1995.

[3] N. Doulamis, A. Doulamis, Y. Avrithis, K. Ntalianis and S. Kollias, "Efficient Summarization of Stereoscopic Video Sequences," *IEEE Trans. Circuits and Systems for Video Technology*, Vol. 10, No. 4, pp. 501-517, June 2000.

[4] R. Castagno, T.Ebrahimi, and M. Kunt, "Video Segmentation Based on Multiple Features for Interactive Multimedia Applications," *IEEE Trans. Circuits and Systems for Video Technology*, Vol. 8, No. 5, pp. 562-571, 1998.

[5] C. Gu and M.-C.Lee, "Semiautomatic Segmentation and Tracking of Semantic Video Objects," *IEEE Tran. Circuits and Systems for Video Technology*, Vol. 8, No. 5, pp. 572-584, 1998.

[6] C. Xu and J. L. Prince, "Snakes, Shapes and Gradient Vector Flow," *IEEE Trans. Image Processing*, Vol. 7, No. 3, pp. 359-369, March 1998.

[7] S. Lobregt and M. A. Viergever, "A discrete dynamic contour model," *IEEE Trans. Medical Imaging*, Vol. 14, pp. 12-24, March 1995.

[8] D. Wang, "Unsupervised Video Segmentation Based on Watersheds and Temporal Tracking," *IEEE Tran. Circuits and Systems for Video Technology*, Vol. 8, No. 5, pp. 539-546, 1998.

[9] T. Meier, and K. Ngan, "Video Segmentation for Content-Based Coding," *IEEE Tran. Cicuits and Systems for Video Technology*, Vol. 9, No. 8, pp. 1190-1203, 1999.

[10] M. Kim, J.G. Choi, D. Kim, H. Lee, 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 Tran. Circuits and Systems for Video Technology*, Vol. 9, No. 8, pp. 1216-1226, 1999.

[11] H-L. Eng, and K-K. Ma, "Spatio-Temporal Segmentation of Moving Video Objects over MPEG Compressed Domain," in *Proc. of the International Conference on Multimedia and Expo (ICME),* New York, USA, August 2000.

[12] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Addison-Wesley, 1992.

[13] D. J. Luenberger, *Linear and non-Linear Programming.* Addison-Wesley 1984.

[14] A. D. Doulamis, N. D. Doulamis, K. S. Ntalianis, and S. D. Kollias, "Unsupervised Semantic Object Segmentation of Stereoscopic Video Sequences," in *Proc. of the IEEE International Conference on Information, Intelligence and Systems* (*ICIIS*), Washington D.C., U.S.A, November 1999.

[15] N. Grammalidis and M. G. Strintzis, "Disparity and Occlusion Estimation in Multiocular Systems and Their Coding for the Communication of Multiview Image Sequences," *IEEE Trans. Circuits and Systems for Video Technology*, Vol. 8, No. 3, pp. 328-344, June 1998.

[16] Y. Avrithis, A. Doulamis, N. Doulamis and S. Kollias, “A Stochastic Framework for Optimal Key Frame Extraction from MPEG Video Databases,” *Computer Vision and Image Understanding,* Academic Press, Vol. 75, Nos 1/2, pp. 3-24, July/August 1999.

[17] M. Kass, A. Witkin, and D. Terzopoulos, “Snakes: Active contour models,” *International Journal of Computer Vision,* vol. 1, pp. 321-331,1987.

[18] L. D. Cohen and I. Cohen, “Finite-element methods for active contour models and balloons for 2-D and 3-D images,” *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 15, pp. 1131-1147, Nov. 1993.

[19] A. A. Amini, S. Tehrani, and T. E. Weymouth, “Using Dynamic Programming for Minimizing the Energy of Active Contours in the Presence of Hard Constraints,” in *Proc. of the Second International Conference on Computer Vision (ICCV),* 1988, pp. 95-99.

[20] D. J. Williams and M. Shah, “A fast algorithm for active contours and curvature estimation, ” *GVGIP: Image Understanding,* vol. 55, no.1, pp. 14-26, January 1992.

[21] J. Slater, “Eye to Eye with Stereoscopic TV,” *Image Technology,* p. 23, Nov./Dec. 1996.

[22] C. Girdwood and P. Chiwy, “MIRAGE: An ACTS Project in Virtual Production and Stereoscopy,” *IBC Conference Publication,* No. 428, pp. 155-160, Sept. 1996.

[23] M. Ziegler, “Digital Stereoscopic Imaging and Applications: A Way Towards New Dimensions, the RACE II Project DISTIMA,” *Proc. of IEE Colloquium on Stereoscopic Television,* London, UK, 1992.

[24] MPEG-4: Doc. ISO/IEC JTC1/SC29/WG11 N2502, “Information Technology – Generic Coding of Audiovisual Objects, Part 2: Visual, Final Draft of International Standard”, October 1998.

[25] M. Wollborn, R. Mech, “Procedure for Objective Evaluation of VOP Generation Algorithms”, Doc. ISO/IEC JTC1/SC29/WG11 MPEG97/2704, Fribourg, Switzerland, October 1997.

[26] P. Correia and F. Pereira, “Objective Evaluation of Relative Segmentation Quality”, *in Proc. International Conference on Image Processing (ICIP),* Vancouver, Canada, September 2000.

[27] R. Mech and F. Marques, “Objective Evaluation Criteria for 2D-Shape Estimation Results of Moving Objects”, *in Proc. of Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS),* Tampere, Finland, May 2001.

[28] P. Villegas, X. Marichal and A. Salcedo, “Objective Evaluation of Segmentation Masks in Video Sequences”, *in Proc. Of Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS),* Berlin, Germany, May-June 1999.

List of Figures

Figure 4 Graphical representation of the active contour polygonal approximation method.

Figure 5 Stereoscopic pairs of frames used for experimental evaluation. (a), (c), (e) Left channels and (b), (d), (f) Right channels.

Figure 6 Disparity and depth estimation for the three stereoscopic pairs. (a), (c), (e) Occlusion compensated disparity fields and (b), (d), (f) The respective depth maps.

Figure 7 Estimation of depth segments maps. (a), (c), (e) Depth segments maps produced by the modified M-RSST algorithm (b), (d), (f) Depth segments maps produced by the conventional M-RSST algorithm.

Figure 8 Demonstration of M-RSST algorithm for color segmentation in case of Figure 5(a). (a) Segmentation at resolution level 3, (b) Segment splitting at level 3, (c) Segmentation at resolution level 2, (d) Segment splitting at level 2 and (e) Segmentation at resolution level 1 (final segmentation).

Figure 9 Video objects extraction results by CFCS approach in case of stereoscopic pair of Figures 5(a)-(b). (a) Depth segmentation overlaid with color segment contours (in white), (b) Fusion of color segments belonging to same depth segment, (c) Foreground video object and (d) Background video object.

Figure 10 Video objects extraction results by CFCS approach in case of stereoscopic pair of Figures 5(c)-(d). (a) Color segments mask, (b) Depth segmentation overlaid with color segment contours (in white), (c) Fusion of color segments belonging to same depth segment, (d) Foreground video objects and (e) Background video object.

Figure 11 Video objects extraction results by CFCS approach in case of stereoscopic pair of Figures 5(e)-(f). (a) Color segments mask, (b) Depth segmentation overlaid with color segment contours (in white), (c) Fusion of color segments belonging to same depth segment, (d) Foreground video object and (e) Background video object.

Figure 12 Video objects extraction results by active contour approach in case of stereoscopic pair of Figures 5(a)-(b). (a) Active contour initialized onto depth boundary, (b) Initial active contour points (green squares) and respective “attractive edge” points (red circles) depicted inside e_{init_f}, (c) Convergence of active contour to minimum energy position, (d) Foreground video object and (e) Background video object.

Figure 13 Video objects extraction results by active contour approach in case of stereoscopic pair of Figures 5(c)-(d). (a) Active contours initialized onto depth boundaries, (b) Initial active contours points (green squares) and respective “attractive edge” points (red circles) depicted inside e_{init_f}, (c) Convergence of active contours to minimum energy positions, (d) Foreground video objects and (e) Background video object.

Figure 14 Video objects extraction results by active contour approach in case of stereoscopic pair of Figures 5(e)-(f). (a) Active contour initialized onto depth boundary, (b) Initial active contour points (green squares) and respective “attractive edge” points (red circles) depicted inside

e_{init_f}, (c) Convergence of active contour to minimum energy position, (d) Foreground video object and (e) Background video object.

Figure 15 Manually created reference masks used for quality evaluation of the proposed unsupervised segmentation schemes. (a) Reference mask for video object of Figure 5(a), (b) Reference mask for left video object of Figure 5(c) (man), (c) Reference mask for right video object of Figure 5(c) (woman) and (d) Reference mask for video object of Figure 5(e).

Figure 16 Diagram of the weighting functions $w_{\mathbf{MF}}(d)$ and $w_{\mathbf{AB}}(d)$ of Equation (17).

Figure 17 (a) SQM′ for more than 500 video objects and for both approaches and (b) The respective complexity in seconds.

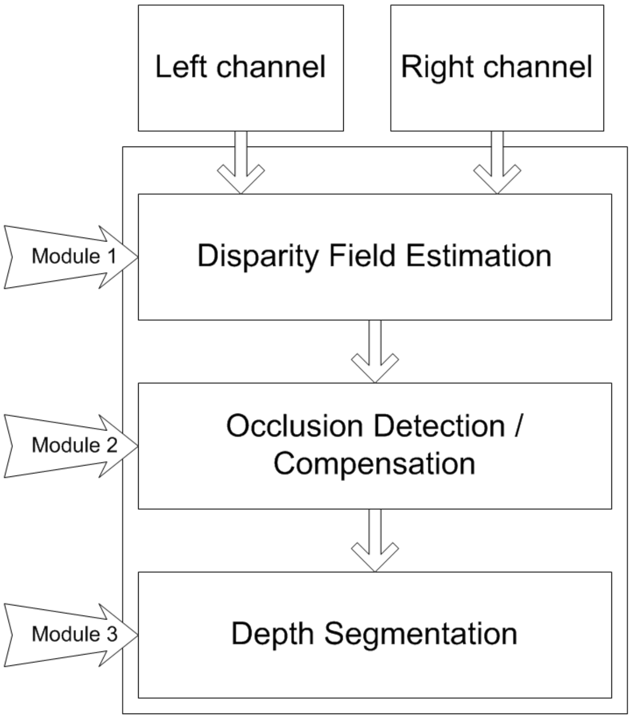
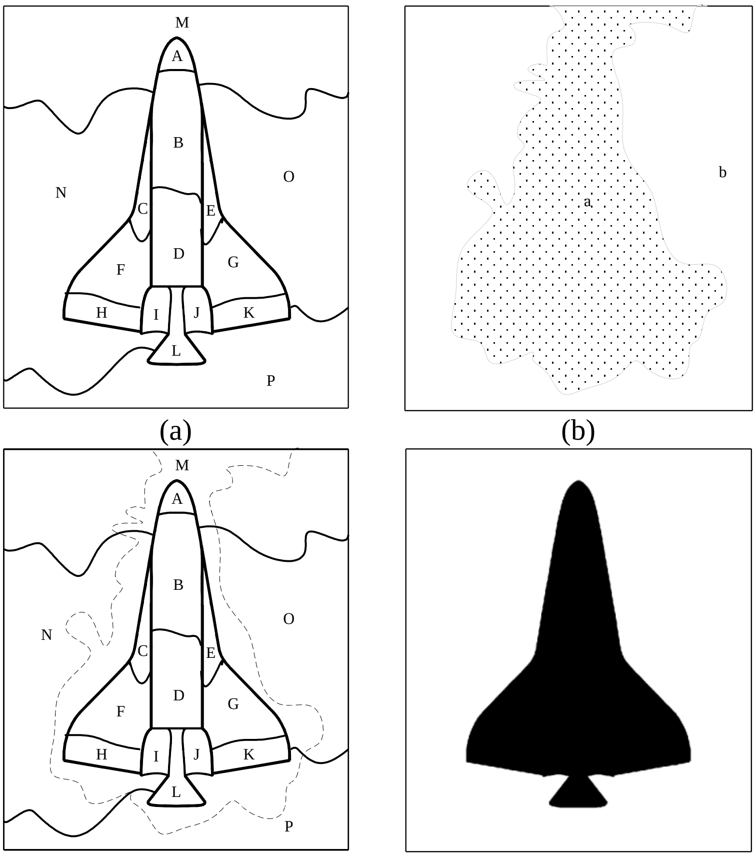


Figure 1



(a)

(b)

(c)

(d)

Figure 2

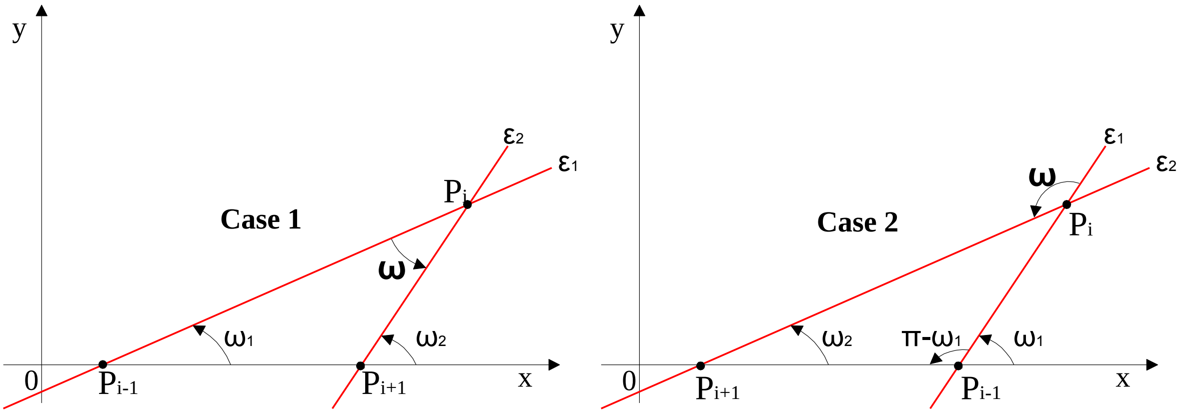


Figure 3

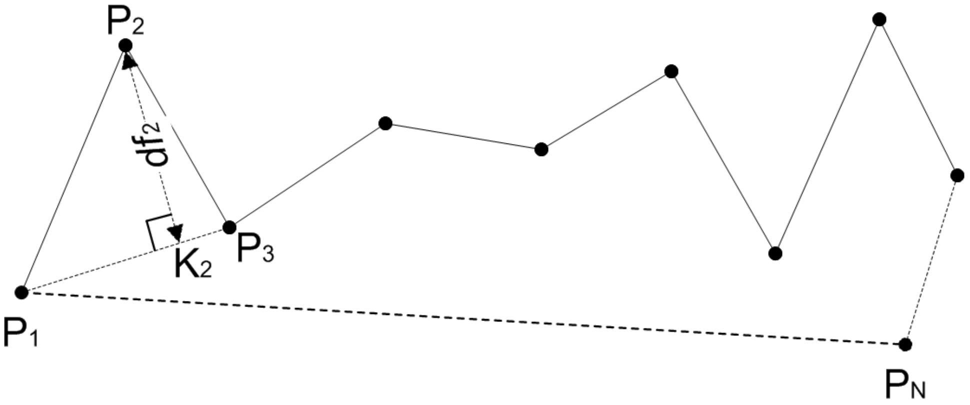
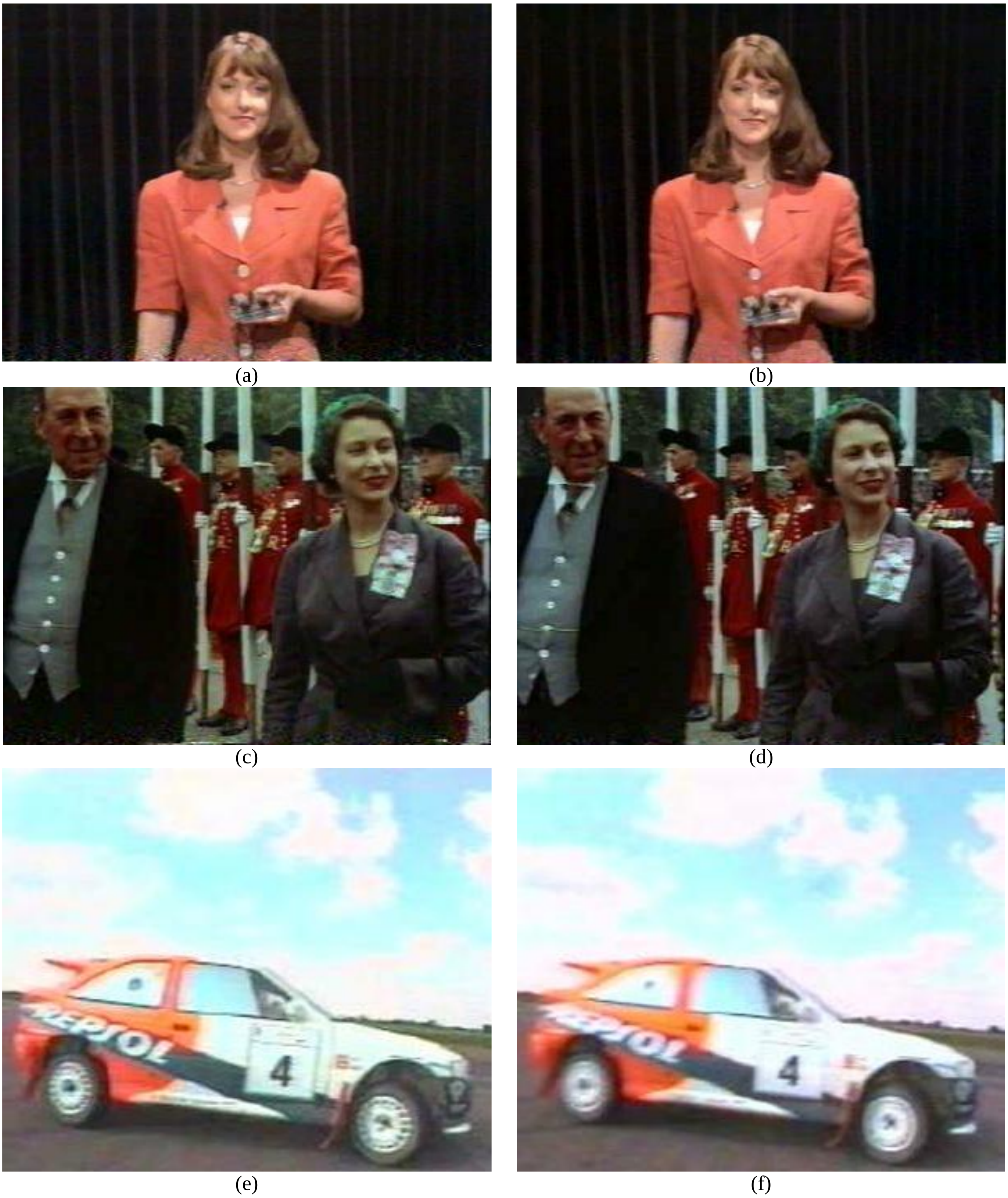
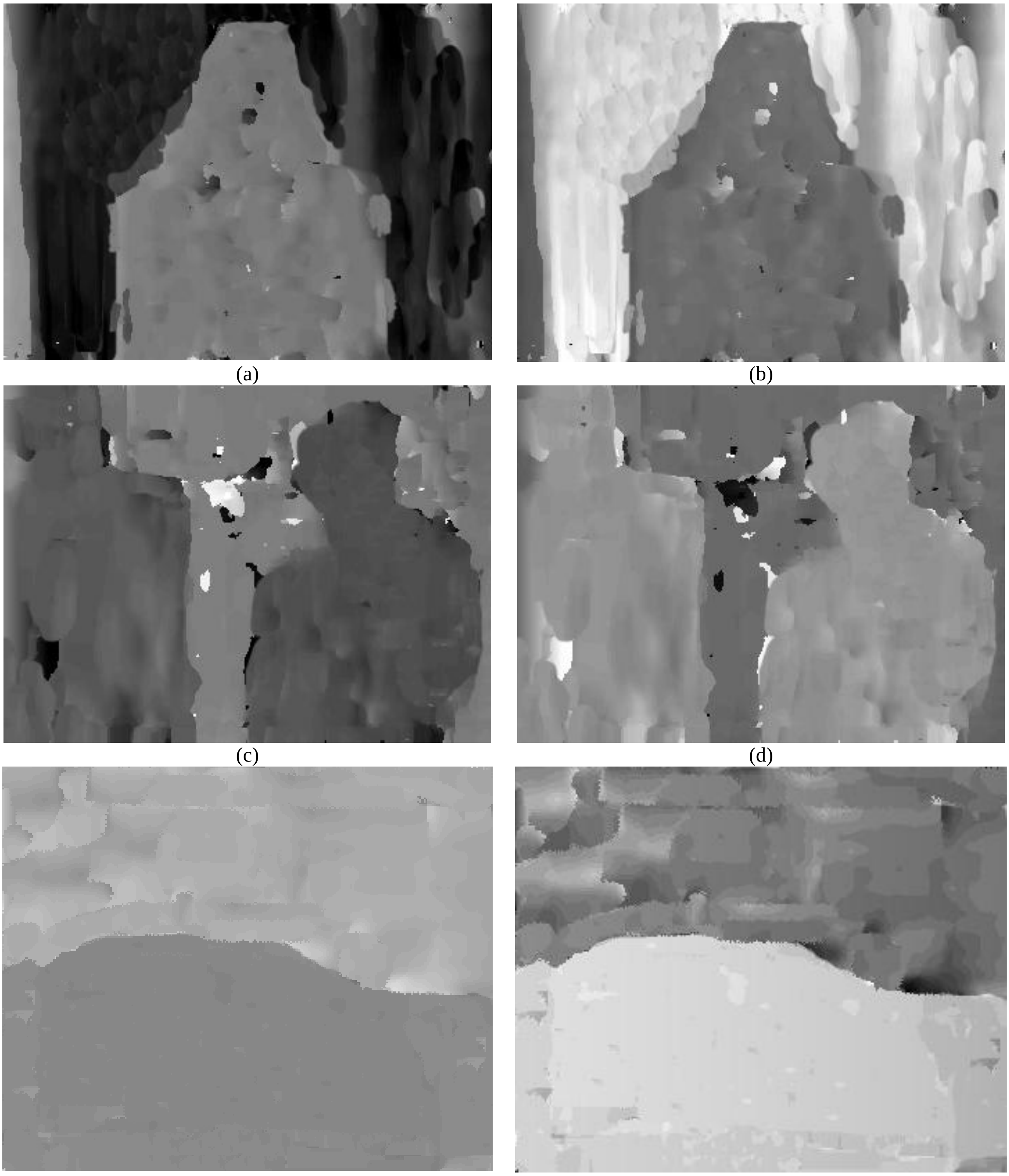


Figure 4



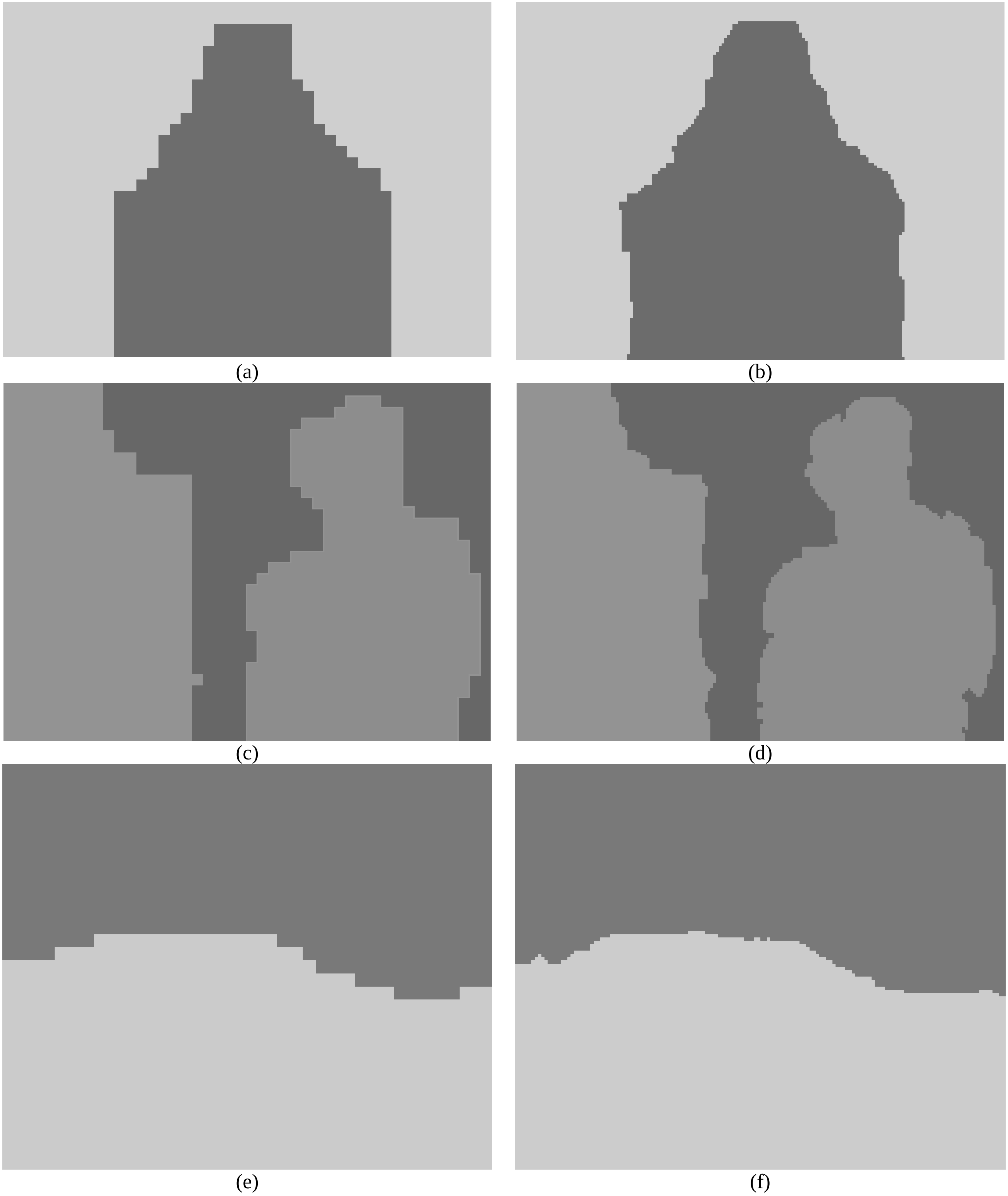
(a) (b) (c) (d) (e) (f)

Figure 5



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

Figure 6



(a)

(b)

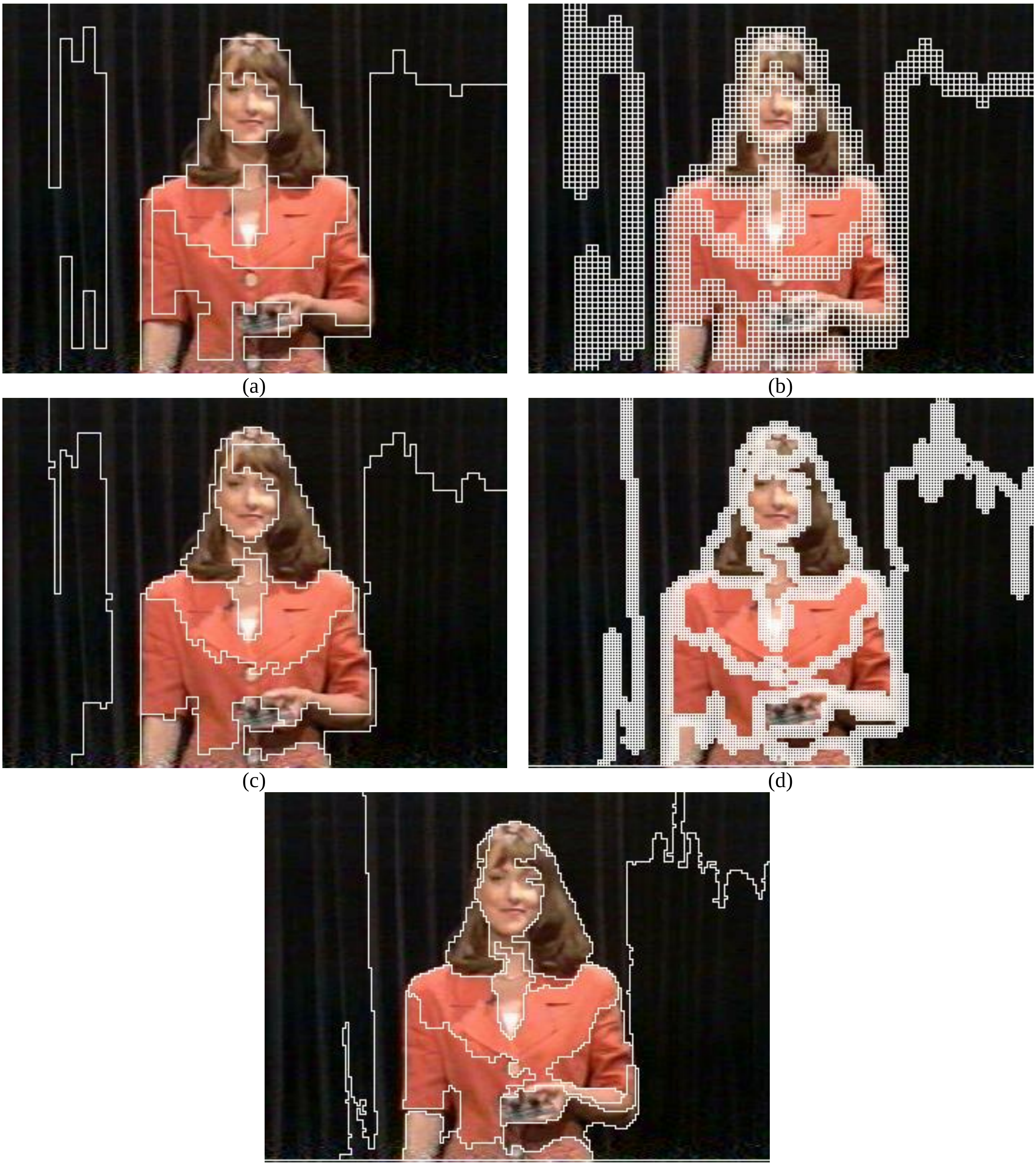
(c)

(d)

(e)

(f)

Figure 7



(a)

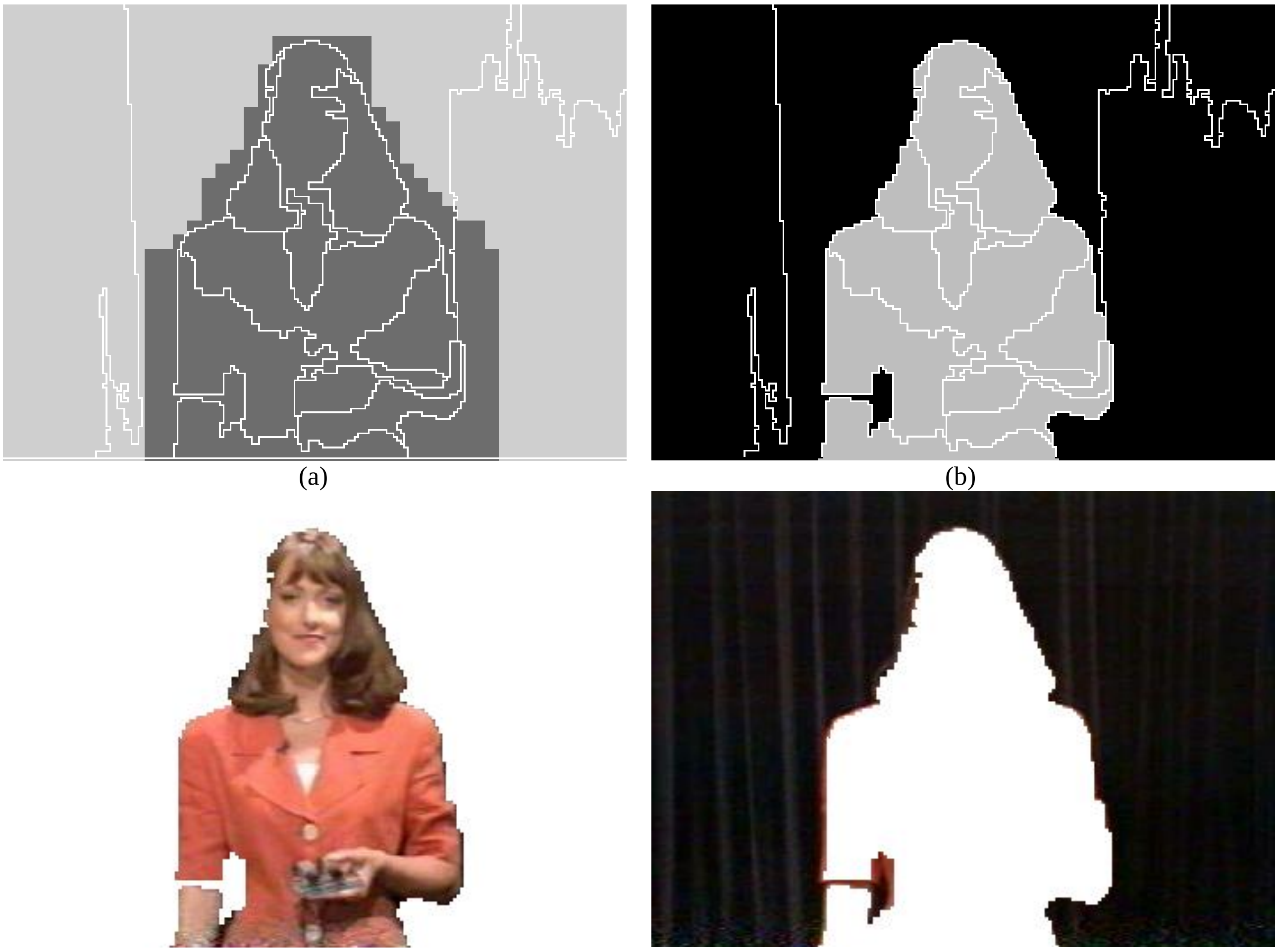
(b)

(c)

(d)

(e)

Figure 8



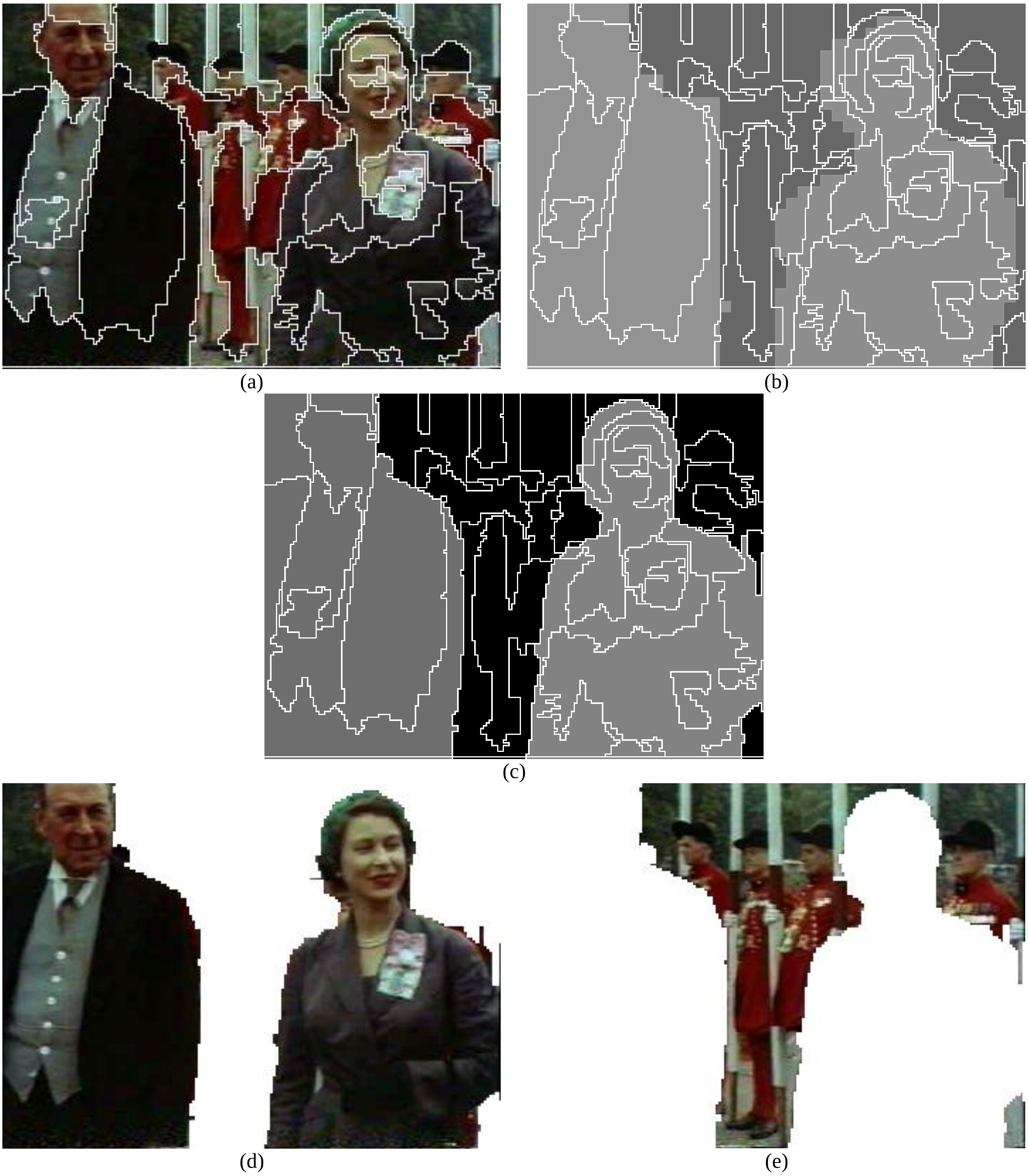
(a)

(b)

(c)

(d)

Figure 9



(a)

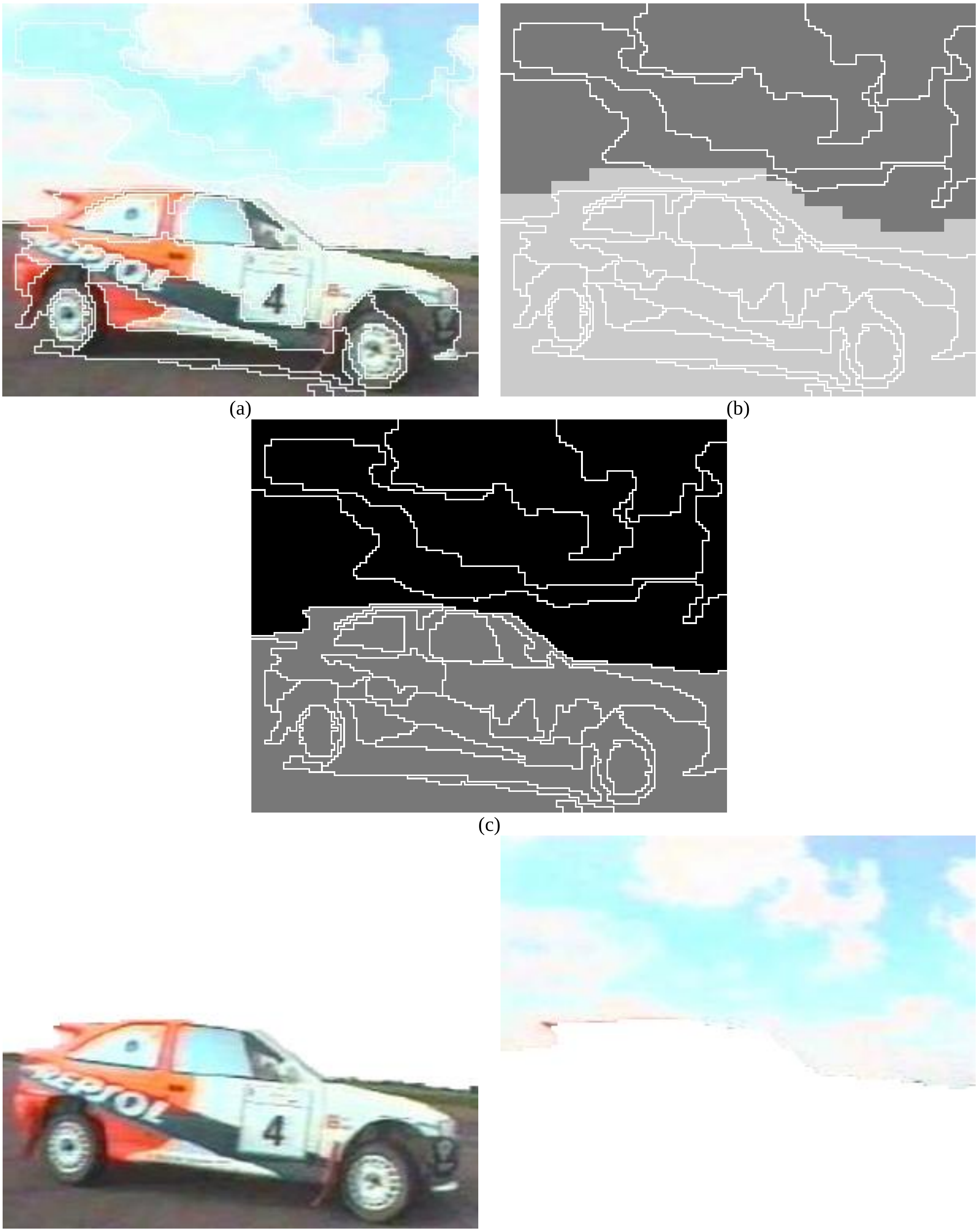
(b)

(c)

(d)

(e)

Figure 10

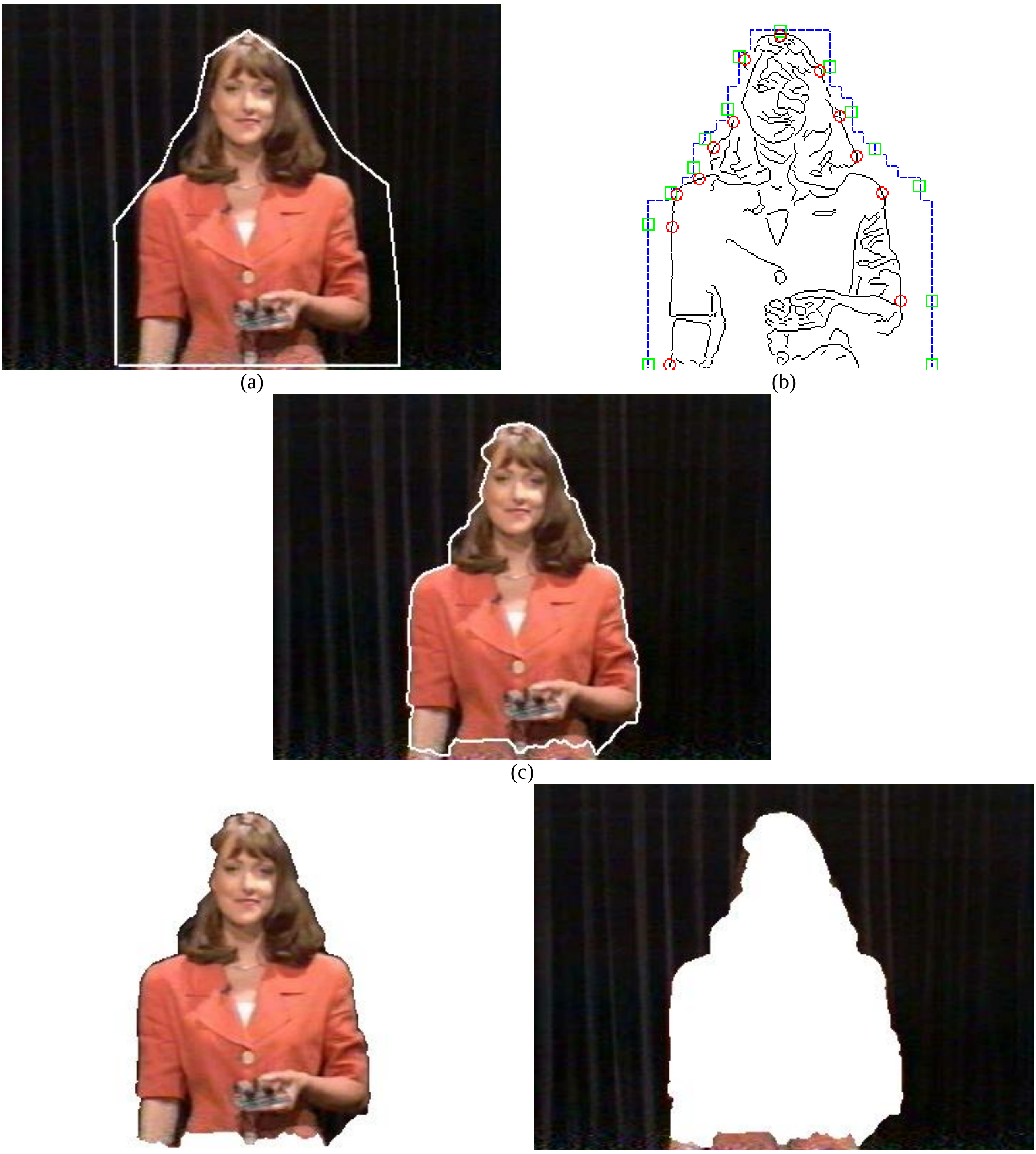


(a) (b)

(c)

(d) (e)

Figure 11



(a)

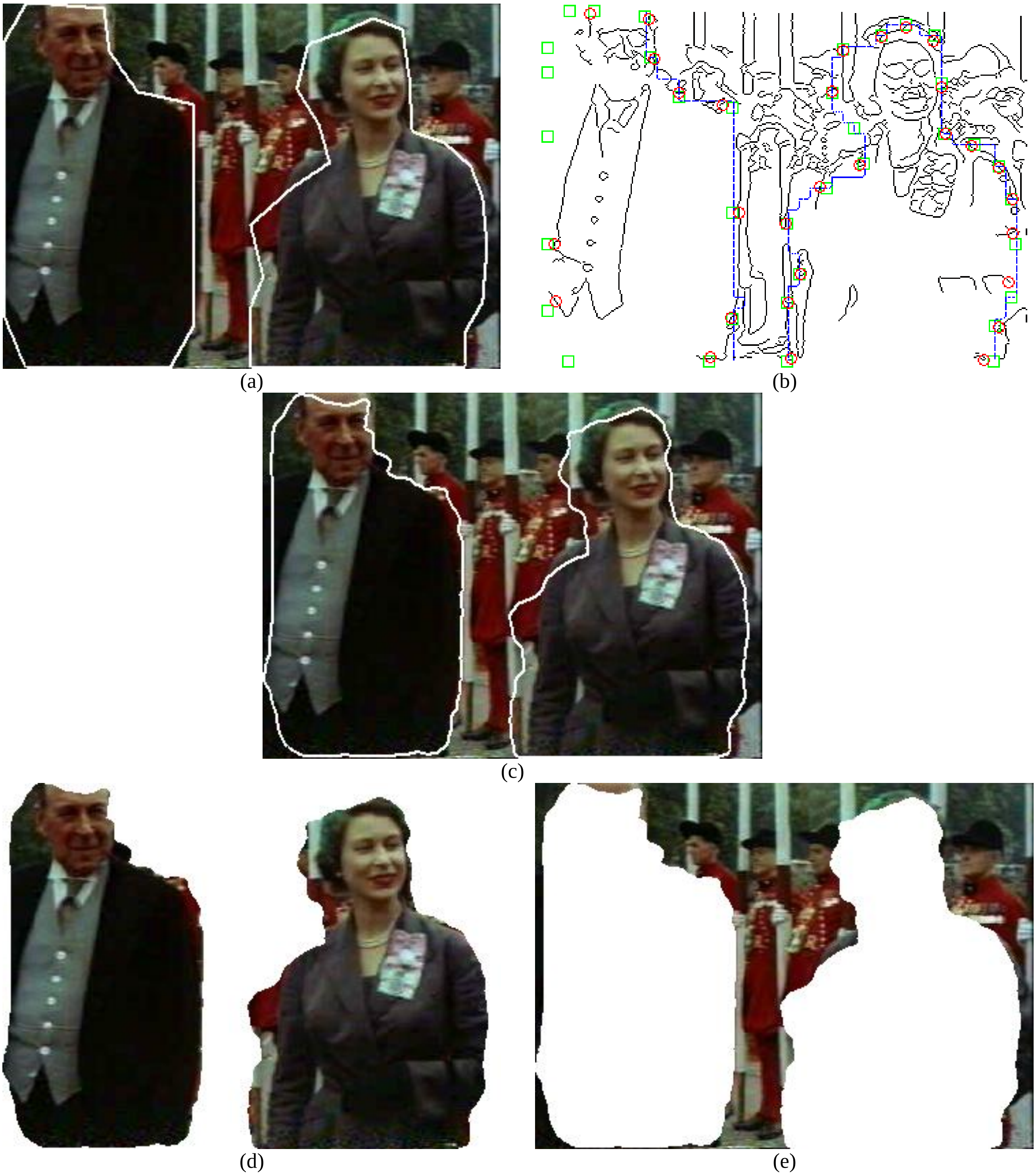
(b)

(c)

(d)

(e)

Figure 12

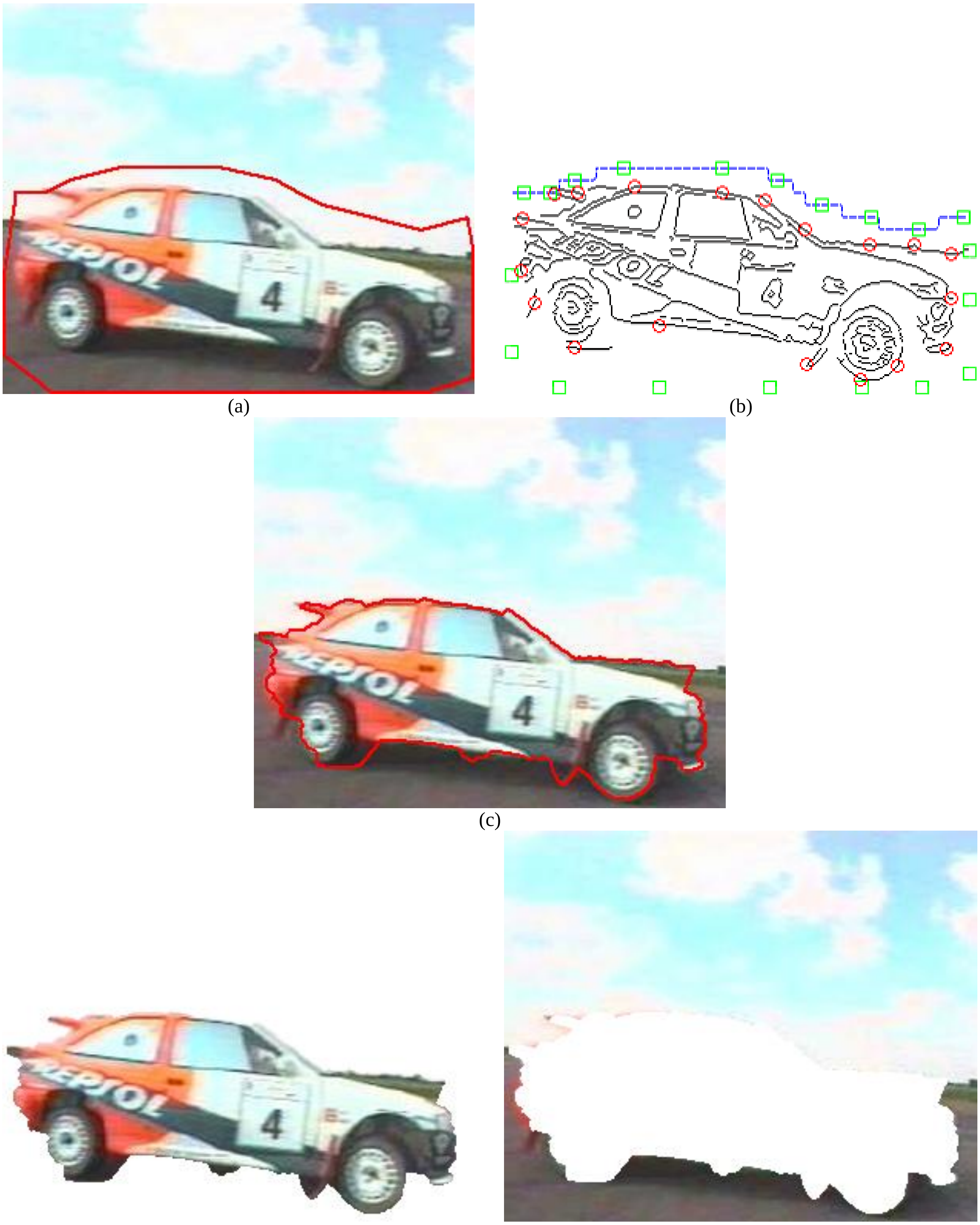


(a) (b)

(c)

(d) (e)

Figure 13

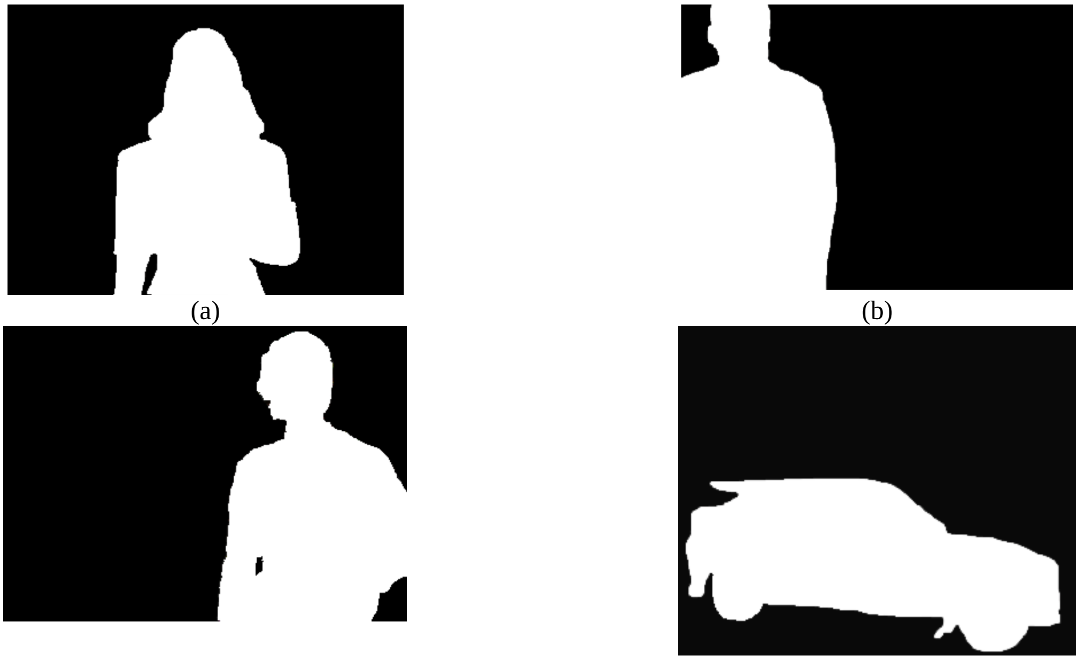


(a) (b)

(c)

(d) (e)

Figure 14



(a)

(b)

(c)

(d)

Figure 15

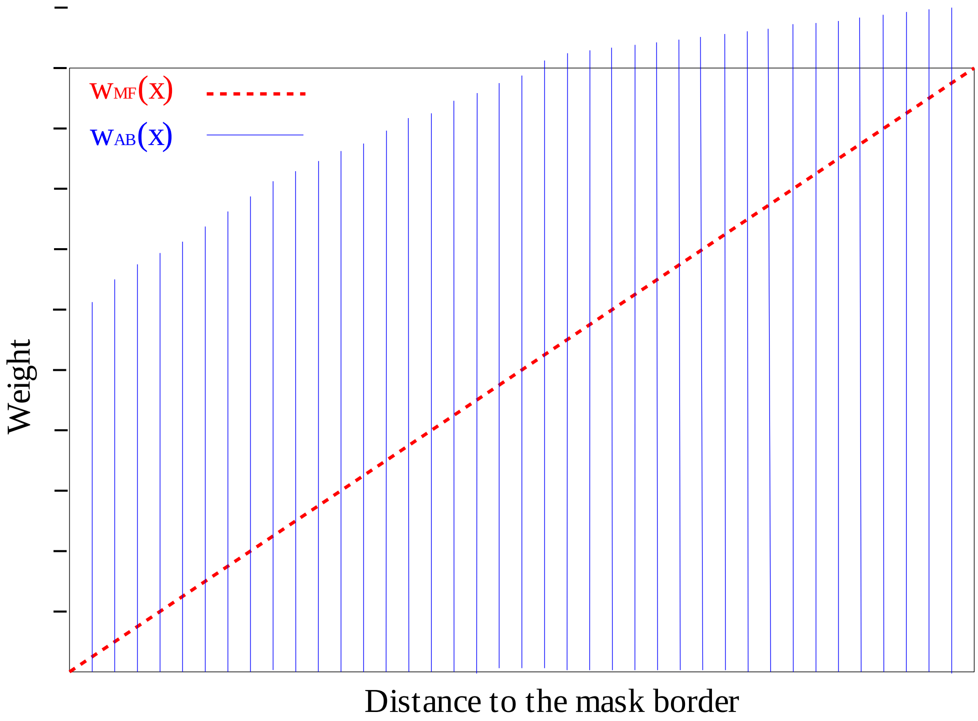


Figure 16

TABLE I

1st CASE (WOMAN)	# of Pixels	# of VOP Pixels	*CP* %	*AP* %	SQM	SQM'	Execution Time (sec)
Perfect Mask	27.171	27.171	100	100	0	0	-
Fusion Mask	26.234	25.950	95.5	98.9	0.08	-0.77	1.06
Active Contour Mask	26.829	25.758	94.8	96.0	0.45	-3.70	0.55

TABLE II

2nd CASE (MAN)	# of Pixels	# of VOP Pixels	*CP* %	*AP* %	SQM	SQM'	Execution Time (sec)
Perfect Mask	29.409	29.409	100	100	0	0	-
Fusion Mask	31.496	29.325	99.7	93.1	0.55	-4.37	2.02
Active Contour Mask	29.803	27.597	93.8	92.6	0.54	-4.29	0.79

TABLE III

2nd CASE (WOMAN)	# of Pixels	# of VOP Pixels	*CP* %	*AP* %	SQM	SQM'	Execution Time (sec)
Perfect Mask	27.867	27.867	100	100	0	0	-
Fusion Mask	28.975	27.724	99.5	95.7	0.26	-2.33	1.90
Active Contour Mask	29.764	26.287	94.3	88.3	1.07	-7.30	0.69

TABLE IV

3rd CASE (CAR)	# of Pixels	# of VOP Pixels	*CP* %	*AP* %	SQM	SQM'	Execution Time (sec)
Perfect Mask	23.804	23.804	100	100	0	0	-
Fusion Mask	34.036	23.440	98.5	68.9	3.5878	-15.23	1.56
Active Contour Mask	23.474	22.313	93.7	95.1	0.3433	-2.95	0.98

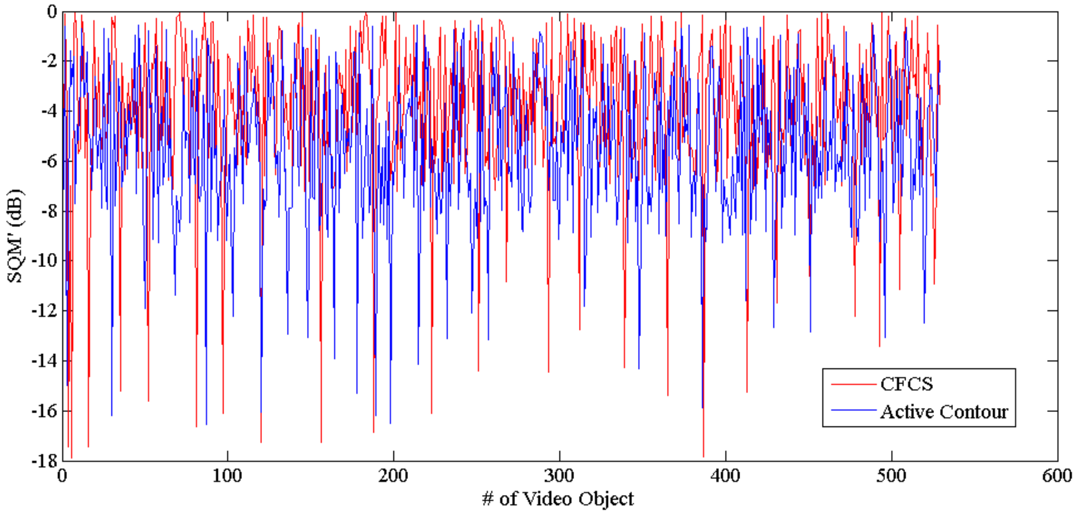


Figure 17(a)

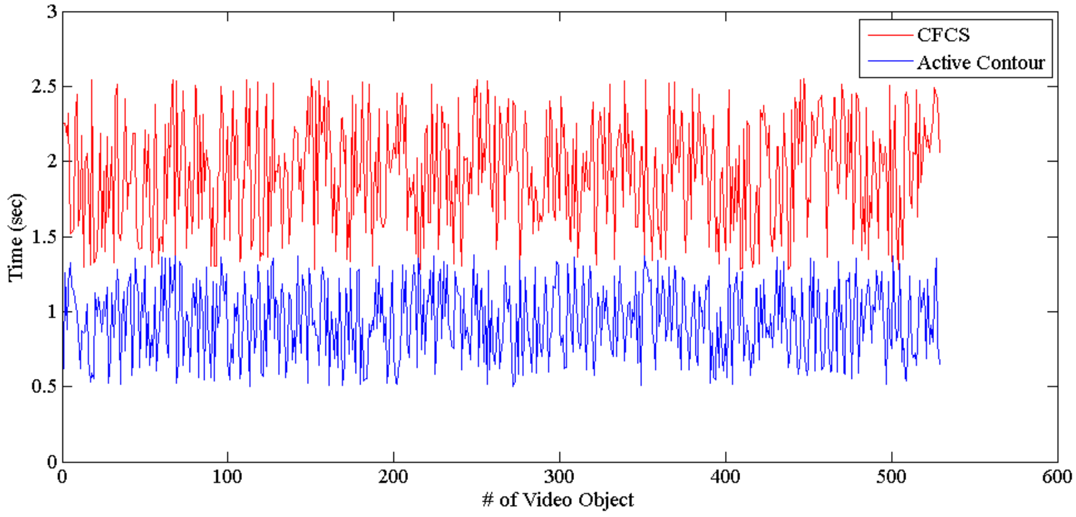


Figure 17(b)