Efficient Summarization of Stereoscopic Video Sequences

Nikolaos D. Doulamis, *Student Member, IEEE,* Anastasios D. Doulamis, *Student Member, IEEE,* Yannis S. Avrithis, *Student Member, IEEE,* Klimis S. Ntalianis, *Student Member, IEEE,* and Stefanos D. Kollias, *Member, IEEE*

***Abstract*—An efficient technique for summarization of stereoscopic video sequences is presented in this paper, which extracts a small but meaningful set of video frames using a content-based sampling algorithm. The proposed video-content representation provides the capability of browsing digital stereoscopic video sequences and performing more efficient content-based queries and indexing. Each stereoscopic video sequence is first partitioned into shots by applying a shot-cut detection algorithm so that frames (or stereo pairs) of similar visual characteristics are gathered together. Each shot is then analyzed using stereo-imaging techniques, and the disparity field, occluded areas, and depth map are estimated. A multiresolution implementation of the Recursive Shortest Spanning Tree (RSST) algorithm is applied for color and depth segmentation, while fusion of color and depth segments is employed for reliable video object extraction. In particular, color segments are projected onto depth segments so that video objects on the same depth plane are retained, while at the same time accurate object boundaries are extracted. Feature vectors are then constructed using multidimensional fuzzy classification of segment features including size, location, color, and depth. Shot selection is accomplished by clustering similar shots based on the generalized Lloyd–Max algorithm, while for a given shot, key frames are extracted using an optimization method for locating frames of minimally correlated feature vectors. For efficient implementation of the latter method, a genetic algorithm is used. Experimental results are presented, which indicate the reliable performance of the proposed scheme on real-life stereoscopic video sequences.**

***Index Terms*—Content-based indexing and retrieval, stereoscopic image analysis, video summarization.**

I. Introduction

RECENT progress in the field of video analysis and processing has led to an explosion in the amount of visual information being stored, accessed and transmitted. This has stimulated new technologies for efficient searching, indexing, content-based retrieving and managing multimedia databases [1]–[3]. The key for this rapid growth was urged by the development of various video-compression standards, such as MPEG-1/2 [4], [5] or H.261/3 [6], [7], each of which is associated with different applications and different bit rates. A new dimension to visual communication is expected to be provided by the MPEG-4 standard [8], which allows content-based video coding and representation, giving users new capabilities of accessing, manipulating, and editing visual content [9], [10]. Moreover, the MPEG group has recently begun a new standardization phase (MPEG-7) for a multimedia content description interface [11]. The MPEG-7 standard will specify a set of content descriptors for any multimedia information.

Although most video archives mainly consist of 2-D video sequences, the use of 3-D video, obtained by stereoscopic or multiview camera systems, has recently increased since it provides more efficient visual representation and enhances multimedia communication. 3-D video enables users to handle and manipulate video objects more efficiently by exploiting, for example, depth information provided by stereo-image analysis. Furthermore, the problem of content-based segmentation is addressed more precisely since video objects are usually composed of regions belonging to the same depth plane [12]. Various applications, such as video surveillance, image/video indexing and retrieval, or editing of video content, can gain from such 3-D representation. For this reason, 3-D data acquisition and display systems have attracted a great interest recently and consequently archives of 3-D video information are expected to rapidly increase in the forthcoming years.

Traditionally, 3-D video sequences are represented by numerous consecutive frame sets, such as stereo pairs in the case of stereoscopic video, each of which corresponds to a constant time interval. The images of each set are recorded using slightly different viewpoints of the same scene. However, this image-sequence representation, which stems from the analog tape storage process, results in a linear (sequential) access of video content [13]. While this approach is adequate for viewing a video in a movie mode [14], it has a number of limitations for the new emerging multimedia applications, such as video browsing, content-based indexing, and retrieval. Currently, the only way to browse a video sequence is to sequentially scan video frames, a process that is both time consuming and tedious. Furthermore, video queries on entire video sequences are insufficient, due to significant temporal redundancy of video content [15]. This linear video representation is also not adequate for efficient organization of large video archives, since storage requirements of digitized video information, even in compressed domain, are very large and present challenges to most multimedia servers [16]. To make things worse, most video archives are expected to be located on distributed platforms [3], [13], and thus, access of video data imposes a great deal of bandwidth requirements. For this reason, apart from developing appropriate congestion control schemes or proposing algorithms for effective

Manuscript received March 15, 1999; revised September 30, 1999. This paper was recommended by Guest Editor M. G. Strintzis.

The authors are with the Electrical and Computer Engineering Department, National Technical University of Athens, Zografou 15773, Athens, Greece (e-mail: ndoulam@image.ntua.gr; iavr@image.ece.ntua.gr; stefanos@cs.ntua.gr).

Publisher Item Identifier S 1051-8215(00)04889-8.

network design and management, based, for example, on traffic modeling of video sources [17], new methods for efficient (nonlinear) video-content representation and summarization should also be implemented [18].

Recently, some approaches have been proposed in the literature for visual summarization, mainly in the framework of the MPEG-7 standardization phase. In particular, shot-cut detection has been presented in [19], which can be seen as the first stage of video-content summarization. Extraction of frames at regular time instances has been proposed in [20]. However, this work does not exploit shot information and frame similarity, and therefore, shots of small duration but of significant content may be discarded, whereas at the same time, multiple frames with similar content may be retained from shots of longer duration. Selection of a single key frame for each shot has been presented in [21], [18], which cannot provide sufficient information about the video content, especially for long shots with a lot of activity. Construction of compact image maps or image mosaics has been described in [14], [22]. Although such approaches can be very efficient for specific applications, such as sports programs or studio productions, they cannot provide satisfactory results in real world complex shots, where background/foreground changes or complicated camera effects are encountered. A method for analyzing video and building a pictorial summary for visual representation has been proposed in [13]. This work is concentrated on dividing a video sequence into consecutive meaningful segments (story units) and then constructing a video poster for each story unit based on shot dominance, instead of extracting key frames. Moreover, all the aforementioned works are dealing with 2-D video sequences and cannot be directly applied to 3-D video archives, since 3-D information is not taken into consideration.

In the context of this paper, a generalized framework for nonlinear representation of 3-D video sequences is proposed, regardless of the scene complexity. A content-based sampling algorithm [23] is used which segments the sequences into shots, clusters shots with similar video content together, selects a representative shot from each cluster, and finally, extracts multiple representative frames (key frames) for each selected shot. This approach provides summarization of visual information similarly to that used in current document search engines [3]. Thus, it is possible to automatically generate low resolution video clip previews (trailers) or still image mosaics, which play exactly the same role for stereo video sequences as "thumbnails" for still images. Fast browsing of stereo video content, efficient performance of video queries and easy access to 3-D video databases, located on distributed platforms, can benefit from such content-based representation.

For this purpose, high-level image processing and analysis techniques should be applied to stereo–video sequences in order to obtain an efficient description of video content. This can be accomplished through segmentation into semantically meaningful objects, which, with the exception of some specific applications, is in general a very difficult task [24], [25]. However, in cases of 3-D video sequences, where depth information can be estimated reliably, high-level video processing can be performed more efficiently, since video objects are usually located on the same depth plane [12]. Several algorithms have been proposed in the literature for stereo–video sequence analysis [26], [27]. In this paper, a more reliable representation of video content is proposed by combining the results obtained from color and depth segmentation, so that video objects on the same depth plane are retained, while accurate object boundaries are extracted. A multiresolution implementation of the Recursive Shortest Spanning Tree (RSST) algorithm is employed for both color and depth segmentation. This hierarchical approach, apart from reducing computational cost, also prevents from possible oversegmentation, which is not desirable in the context of video summarization. Then, appropriate features are extracted, including segment size, location, average color components and depth, and gathered together using fuzzy classification to increase the robustness of the proposed summarization scheme. Finally, shots of similar content are grouped using the generalized Lloyd-Max algorithm [28], while key frames within each selected shot are extracted by minimizing a cross correlation criterion by means of a genetic algorithm.

This paper is organized as follows. Section II introduces the stereoscopic image analysis that is necessary for extracting 3-D information from a pair of left and right channel images and constructing disparity, occlusion and depth maps. Section III presents the multiresolution implementation of the RSST algorithm (M-RSST) that is used for segmenting the left channel image and the corresponding depth map. A segmentation fusion algorithm for combining color and depth segments is also presented in this section, while the fuzzy feature vector formulation is introduced in Section IV. Then, all the above frame analysis techniques are used in Section V for stereo video sequence analysis and summarization by means of shot-cut detection, shot clustering and key-frame selection. Finally, experimental results for a real-life stereo sequence are given in Section VI and conclusions are drawn in Section VII.

II. Stereoscopic Image Analysis

Depth information is estimated more reliably in stereo-image sequences in contrast to monocular 2-D sequences [12], since more than one separate image views are available in the former case [29], [30]. The analysis below concentrates on depth estimation from a binocular camera system.

A. Disparity and Depth Estimation

Consider a stereoscopic system with two cameras of *focal length* λ and *baseline distance* b, as shown in Fig. 1. The optical axes of the two cameras are converging with angle θ. The origins of the two camera coordinate systems are located at the focal points (lens centers), at distance λ from the corresponding image planes I_1 (left channel) and I_2 (right channel), respectively. It is assumed, without loss of generality, that the world coordinate system coincides with the coordinate system of camera 1 (left camera), while the coordinate system of camera 2 (right camera) is obtained from the former through appropriate rotations and translations.

A point **w** with world coordinates (X, Y, Z) is projected on image plane I_1 as point (x_1, y_1) and on image plane I_2 as point (x_2, y_2), as illustrated in Fig. 1. Then, assuming a perspective

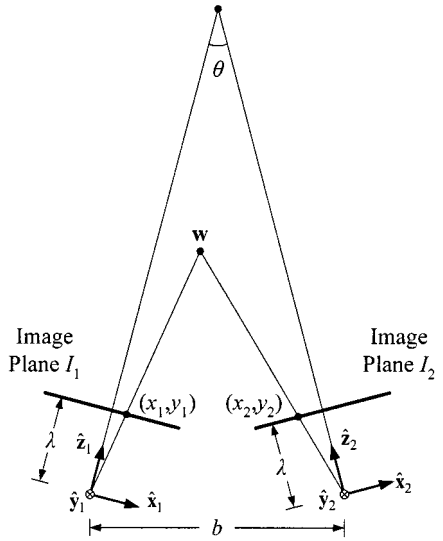


Fig. 1. Geometry of a stereoscopic camera system with convergent optical axes and perspective projection of a 3-D point on the corresponding image planes.

projection scheme, a simple relation between the camera coordinates (x_1, y_1), (x_2, y_2) and depth Z can be obtained [29], [30]

$$x_2 = \lambda \frac{(\lambda s + x_1 c)Z - \lambda bc'}{(\lambda c - x_1 s)Z + \lambda bs'}, \quad y_2 = \lambda \frac{y_1 Z}{(\lambda c - x_1 s)Z + \lambda bs'} \tag{1}$$

where $s = \sin\theta$, $c = \cos\theta$, $s' = \sin\theta/2$, and $c' = \cos\theta/2$. As is observed from (1), the depth Z of $\boldsymbol{w}$ can be estimated if its projections (x_1, y_1) and (x_2, y_2) on image planes I_1 and I_2 respectively are known. Consequently, for a given point (x_1, y_1) on I_1, its correspondent (x_2, y_2) on I_2 should be found. This is accomplished by computing the disparity vector $\boldsymbol{d}(x_1, y_1) = [d_x(x_1, y_1)\, d_y(x_1, y_1)]^T$ at location (x_1, y_1) of camera 1 with respect to camera 2

$$\begin{aligned} d_x &= d_x(x_1, y_1) \\ &= x_2 - x_1 \\ &= \frac{[\lambda(\lambda s + x_1 c) - x_1(\lambda c - x_1 s)]Z - \lambda b(\lambda c' + x_1 s')}{(\lambda c - x_1 s)Z + \lambda bs'} \end{aligned} \tag{2}$$

$$\begin{aligned} d_y &= d_y(x_1, y_1) \\ &= y_2 - y_1 \\ &= \frac{[\lambda - (\lambda c - x_1 s)]y_1 Z - \lambda bs' y_1}{(\lambda c - x_1 s)Z + \lambda bs'}. \end{aligned} \tag{3}$$

If the disparity vector is known, (2) and (3) reduce to an overdetermined linear system of two equations with a single unknown, Z (the depth) and a least-squares solution can be obtained [27].

Disparity estimation is accomplished by means of a block matching algorithm, similar to that proposed in [31]. Let $I_1(x, y)$ and $I_2(x, y)$, $(x, y) \in F = \{1, \cdots, M_0\} \times \{1, \cdots, N_0\}$, denote the gray-level intensities of images projected on planes I_1 and I_2, at (x, y) location, where M_0, N_0 are the image dimensions. The disparity vector $d(x_1, y_1) = (x_2 - x_1, y_2 - y_1) \in B = \{-s_x, \cdots, s_x\} \times \{-s_y, \cdots, s_y\}$ is estimated within a search area of $s_x \times s_y$ pixels, where $s_x \ll s_y$ due to the small converging angle θ between the two cameras. This is achieved by minimizing the following *cost function:*

$$\begin{aligned} \boldsymbol{d}(x_1, y_1) &= \arg\min_{\boldsymbol{u} \in B} J(\boldsymbol{u}, x_1, y_1) \\ &= \arg\min_{\boldsymbol{u} \in B} \{D(\boldsymbol{u}, x_1, y_1) + S(\boldsymbol{u}, x_1, y_1)\}, \\ &\qquad \forall (x_1, y_1) \in F \end{aligned} \tag{4}$$

where $\boldsymbol{u} = [u_x \;\; u_y]^T$ is a displacement of point (x_1, y_1) on image plane I_2. The first term of the right hand of (4), $D(\boldsymbol{u}, x_1, y_1)$, corresponds to a *block error function,* defined as

$$\begin{aligned} D(\boldsymbol{u}, x_1, y_1) = \sum_{x, y \in W} &(I_2(x_1 + u_x + x, y_1 + u_y + y) \\ &- I_1(x_1 + x, y_1 + y))^2 \end{aligned} \tag{5}$$

where $W = \{-w, \cdots, w\} \times \{-w, \cdots, w\}$ is a rectangular window or block. The second term of (4), $S(\boldsymbol{u}, x_1, y_1)$, is a smoothness error function used to reduce possible noise in estimating $\boldsymbol{d}(x_1, y_1)$ and is defined as

$$S(\boldsymbol{u}, x_1, y_1) = R(x_1, y_1) \sum_{\boldsymbol{v} \in N(x_1, y_1)} \|\boldsymbol{u} - \boldsymbol{v}\|^2 \tag{6}$$

where $N(x_1, y_1) = \{\boldsymbol{d}(x_1 - 1, y_1), \boldsymbol{d}(x_1 - 1, y_1 - 1), \boldsymbol{d}(x_1, y_1 - 1), \boldsymbol{d}(x_1 + 1, y_1 - 1)\}$ is the set of all disparity vectors of pixels neighboring to (x_1, y_1) that have already been calculated from (4), and $\|\cdot\|$ is the Euclidean norm. The *smoothing weight function* $R(x_1, y_1)$, whose estimation is based on the local variance of image I_1, takes low values in regions where matching is reliable, such as edges or highly textured regions, and high values in regions where matching is not reliable, such as regions of uniform intensity. This function is also used for determining the exact size of the search area B. Since in regions with a high value of $R(x_1, y_1)$ the disparity field is smooth, a small search area is adequate, while a larger search area is necessary for regions with a low value of $R(x_1, y_1)$. This means that s_x and s_y are varied according to $R(x_1, y_1)$, resulting in a faster implementation of the minimization procedure.

Depth and disparity estimation results are illustrated in Fig. 2 for the Claude sequence. In particular, Fig. 2(a) and (b) show the original left and right channel images, respectively. The vertical disparity is negligible for the given sequence, since the two cameras are located at the same vertical level. Thus, only the horizontal disparity field $d_x(x_1, y_1)$ is presented in Fig. 2(c), where areas in white correspond to positive disparity and areas in gray to approximately zero disparity. Finally, the depth map is illustrated in Fig. 2(d), where areas in black correspond to background and areas in gray to foreground. Note that in both the disparity field and depth map, the shaded areas of gradual intensity change at the left of the person and at the right edge of the image are due to occlusion, which is discussed in the following.

B. Occlusion Detection

The above analysis of disparity estimation assumes that a corresponding point of image I_2 can always be found for all points

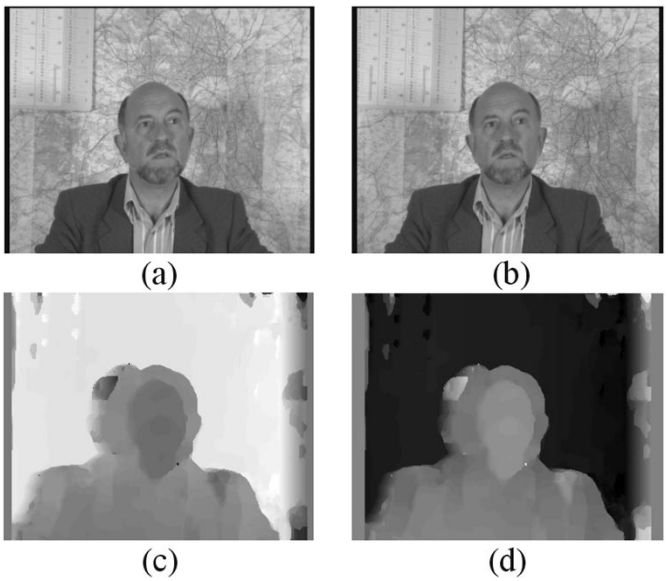


Fig. 2. Disparity and depth estimation for the Claude sequence. (a) Left channel image. (b) Right channel image. (c) Horizontal disparity field. (d) Depth map.

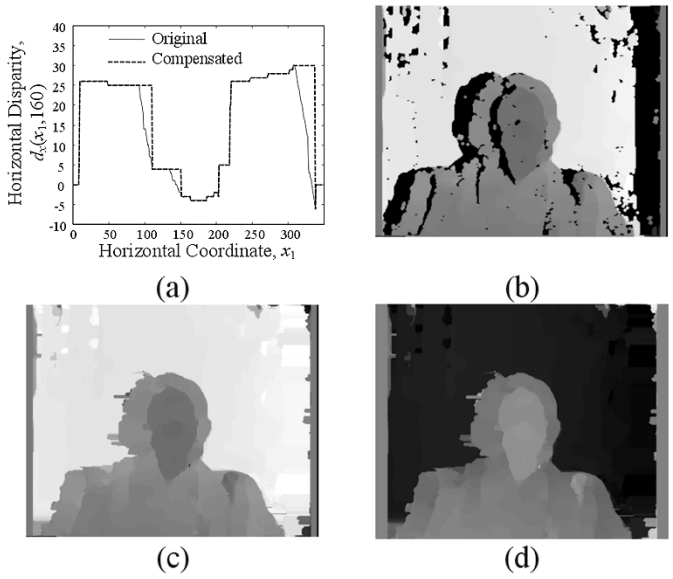


Fig. 3. Occlusion detection and compensation for the Claude sequence. (a) 1-D detection and compensation for line 160 of horizontal disparity field. (b) Occluded areas (in black). (c) Compensated horizontal disparity field. (d) Compensated depth map.

of image I_1. However, due to the different camera viewpoints, there may be areas of I_1 that are occluded in I_2 [32]. All disparity values for occluded areas are not reliable and may result in incorrect depth segmentation. Therefore, it is clear that: i) these areas should be detected and ii) occlusion should be compensated by assigning appropriate disparity values to occluded areas. The former task, *occlusion detection*, is accomplished by locating regions of I_1 where the horizontal disparity decreases continuously with respect to the horizontal coordinate x_1 with a slope approximately equal to -1 [32]. Vertical disparity is not taken into account for this purpose, since all disparities are mostly horizontal, as explained above. The latter task, *occlusion compensation*, is tackled by keeping disparity constant in

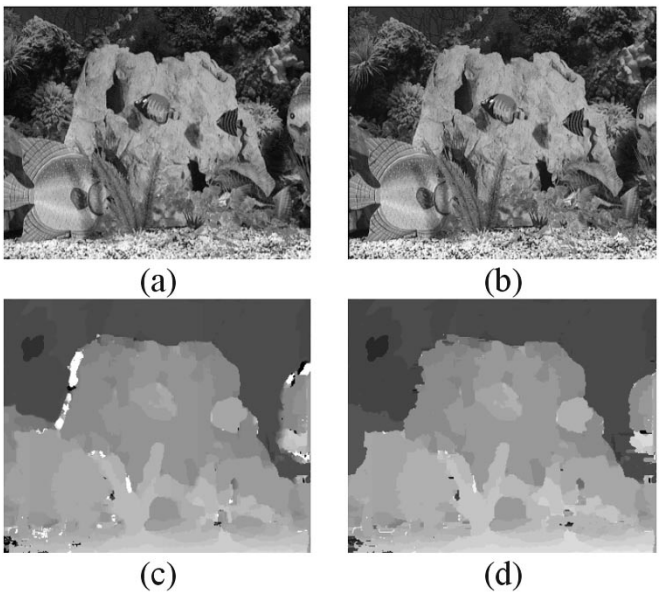


Fig. 4. Depth estimation and occlusion compensation for the Aqua sequence. (a) Left channel image. (b) Right channel image. (c) Depth map without compensation. (d) Compensated depth map.

each occluded area, and equal to the maximum disparity value of that area. This way, each occluded area is effectively merged with the neighboring area of maximum depth, which is consistent with the fact that an object is occluded by another only if it is located farther away from the camera.

The occlusion detection and compensation technique is illustrated in Fig. 3 for the left channel (image plane I_1) of the Claude sequence. The 1-D case is first presented in Fig. 3(a), where the horizontal disparity $d_x(x_1, 160)$ of image line $y_1 = 160$ is plotted versus x_1, with (dotted line) and without (solid line) occlusion compensation. It is evident that in intervals where d_x is nondecreasing, the disparity is left unchanged. On the contrary, intervals of decreasing disparity are detected as occlusion intervals and disparity is compensated. This is accomplished by assigning constant disparity value, equal to the value of the neighboring nonoccluded interval located to the left of each occluded interval. The occluded areas of the entire 2-D horizontal disparity field are shown as black in Fig. 3(b), while the compensated disparity field and corresponding depth map are presented in Fig. 3(c) and (d), respectively. The results for the Aqua sequence are presented in Fig. 4. In particular, Fig. 4(a) and (b) depict the original left and right frame of Aqua, while Fig. 4(c) and (d) the depth maps before and after occlusion compensation. In both cases, the compensated depth map is more reliable.

III. Object Extraction

Video summarization can be performed more efficiently if the visual content of a sequence is described through its semantic video objects. Semantic segmentation has attracted much attention recently, especially in the framework of the emerging MPEG-4 and MPEG-7 standards [9], [24], [25], [33]. Although some solutions exist for specific applications (e.g., videophone systems, news bulletins, etc.) [12], [34], [35], [36] semantic object extraction still remains an unsolved problem [37]. In stereo

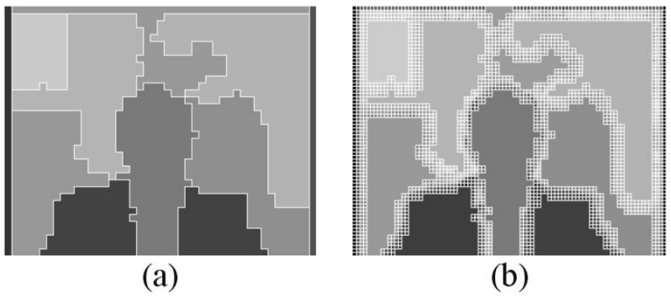


Fig. 5. Demonstration of M-RSST algorithm for color segmentation on Claude sequence. (a) Segmentation at resolution level 3. (b) Segment splitting at level 3.

Fig. 6. Final color segmentation results. (a) Claude sequence. (b) Aqua sequence.

Fig. 7. Final depth segmentation results. (a) Claude sequence. (b) Aqua sequence.

video sequences, however, where depth information can be estimated more reliably, semantic video objects can be identified since usually a video object is located on the same depth plane. In order to retain semantic object extraction, and at the same time obtain accurate object boundaries (contours), color and depth segmentation is first employed and then both segmentation maps are fused together.

A. Color and Depth Segmentation

A multiresolution implementation of the RSST [38] algorithm, called M-RSST, is used both for color and depth segmentation. In this implementation, the RSST algorithm, which is considered one of the most powerful tools for image segmentation compared to other techniques [39], is recursively applied to images of increasing resolution. This approach, apart from accelerating the segmentation procedure, also reduces the number of small objects, which is a useful property in the context of video summarization.

Consider an image I of size $M_0 \times N_0$ pixels. Initially, a multiresolution decomposition of image I is performed until a lowest resolution level, say L_0, so that a hierarchy of images $I(0) = I$, $I(1)$, $\cdots$, $I(L_0)$ is constructed. Consequently, a truncated image pyramid is created, each layer of which contains a quarter of the pixels of the layer below. The conventional RSST algorithm is first applied to the image of the lowest resolution, $I(L_0)$, to provide an initial image segmentation. In the following steps, an iteration begins so that the images of higher resolution are taken into consideration. Particularly, the following tasks are repeated in each iteration of the proposed M-RSST algorithm, until the highest resolution image $I(0)$ is reached.

1) Each boundary pixel of all resulting segments of the current resolution level, corresponding to four pixels of the next higher resolution level, is split into four new segments.
2) New link weights are calculated and sorted.
3) Segments are recursively merged using the conventional RSST iteration phase.

Fig. 5 illustrates the results of color segmentation for the left channel of the Claude sequence, the original frame of which is depicted in Fig. 2(a). A minimum link weight (distance) threshold is selected to terminate the segmentation process similarly to that used in the conventional RSST algorithm [38]. A lowest resolution level of $L_0 = 3$ (i.e., block resolution of 8×8 pixels) is adopted. Fig. 5(a) shows the segmentation results for the image of the lowest resolution. Then, each boundary pixel (or 8×8 block) is split into four new segments (of size 4×4 pixels) according to step 1 of the M-RSST algorithm, as shown in Fig. 5(b). These segments are merged at resolution level 2 and the process is repeated in an iterative way to produce the final segmentation mask, illustrated in Fig. 6(a) for the Claude sequence. Similarly, Fig. 6(b) presents the final color segmentation results for the left channel of the Aqua sequence. Depth segmentation results are depicted in Fig. 7. For the Claude sequence, two segments are extracted as presented in Fig. 7(a), corresponding to the foreground and the background object. Similarly, nine segments are extracted for the Aqua sequence [Fig. 7(b)].

The computational complexity of the M-RSST algorithm is considerably lower than that of the conventional RSST. This is due to the fact that the initial number of segments at each resolution level is significantly reduced; only the boundary pixels are further segmented. However, the computational improvement is not straightforward to calculate, since the speed of the M-RSST algorithm heavily depends on the initial number of segments and the image complexity. Experimental results have indicated an average speed improvement ratio in the order of 400 for an image size of 720×576 and initial resolution level $L_0 = 3$ [40]. Furthermore, the M-RSST algorithm also eliminates very small segments, which is desirable in the framework of video summarization since oversegmentation is avoided.

B. Segmentation Fusion

Although depth segmentation provides a more meaningful frame content representation than color segmentation, i.e., closer to semantic objects, it cannot accurately identify object

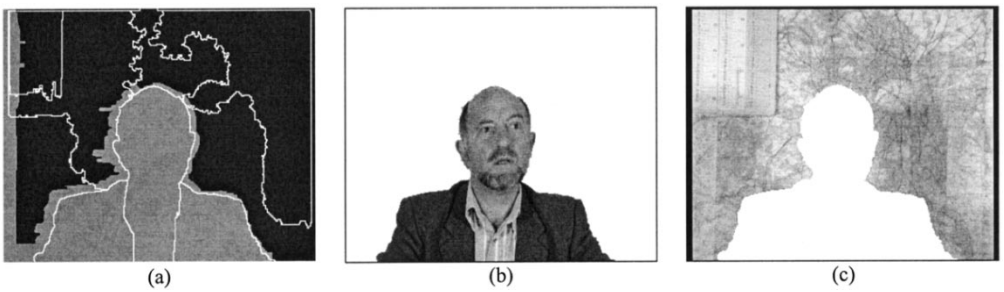


Fig. 8. Segmentation fusion results for the Claude sequence. (a) Depth segmentation overlaid with color segment contours (in white). (b) Foreground object. (c) Background object.

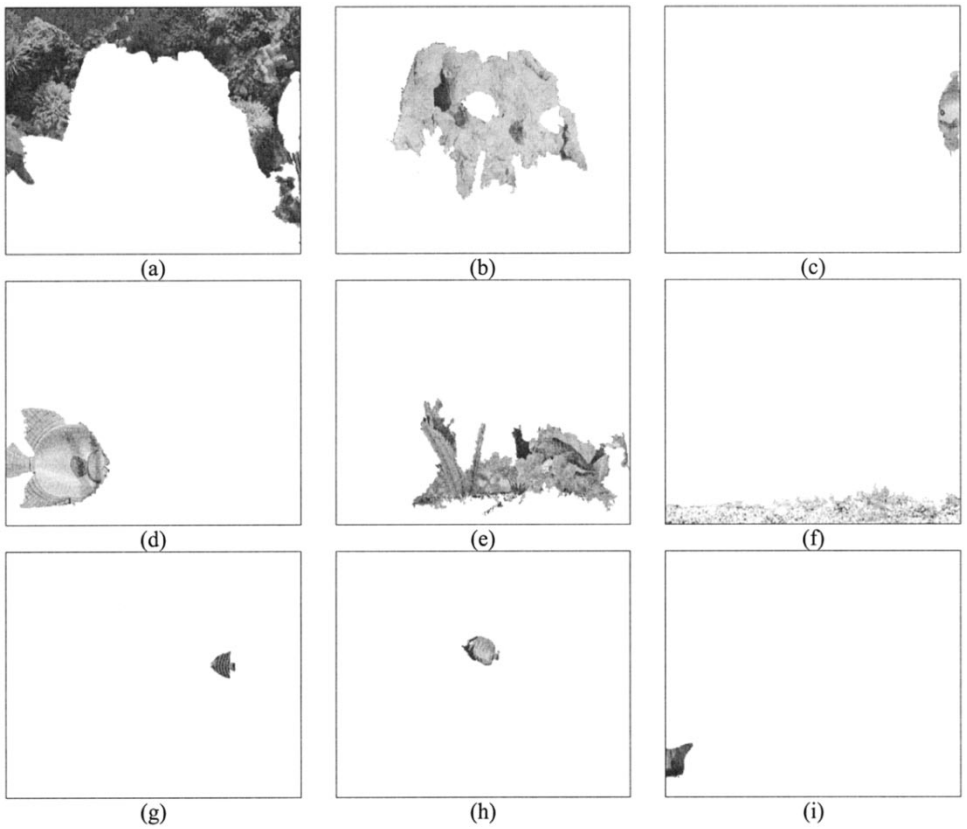


Fig. 9. Object extraction after segmentation fusion for the Aqua sequence.

boundaries (contours), due to erroneous estimation of disparity field and occlusion issues. On the contrary, color segmentation contains the most reliable object boundaries, but usually oversegments a video object into multiple regions [25]. For this reason, a video object is extracted by fusing several color segments using the depth information.

Let us assume that K^c color and K^d depth segments have been extracted using the M-RSST algorithm, denoted as $S_i^c, i = 1, 2, \cdots, K^c$ and $S_i^d, i = 1, 2, \cdots, K^d$, respectively. The S_i^c and S_i^d are mutually exclusive. Let us also denote by G^c and G^d the output masks of color and depth segmentation

$$G^c = \{S_i^c, i = 1, 2, \cdots, K^c\}$$
$$G^d = \{S_i^d, i = 1, 2, \cdots, K^d\}. \tag{7}$$

Each color segment S_i^c is associated with that depth segment whose area of intersection is maximized. This is accomplished by means of a *projection function*

$$p(S_i^c, G^d) = \{\arg\max_{g \in G^d}\{a(g \cap S_i^c)\}, \qquad i = 1, 2, \cdots, K^c \tag{8}$$

where $a(\cdot)$ is the area, i.e., the number of pixels, of a segment. Based on the previous equation, K^d sets of color segments are defined, say $C_i, i = 1, 2, \cdots, K^d$, each of which contains all color segments that are projected onto the same depth segment S_i^d:

$$C_i = \{g \in G^c : p(g, G^d) = S_i^d\}, \qquad i = 1, 2, \cdots, K^d. \tag{9}$$

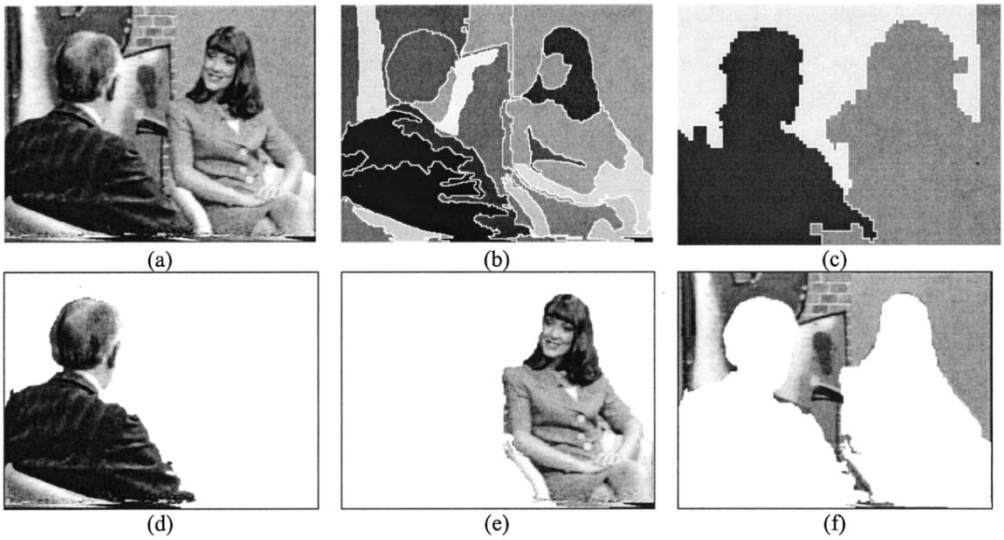


Fig. 10. Object extraction for the "Eye to Eye" sequence. (a) Original left channel frame. (b) Color segmentation. (c) Depth segmentation. (d) Foreground object #1. (e) Foreground object #2. (f) Background object.

Then, the final segmentation mask G consists of $K = K^d$ segments, say, S_i, $i = 1, 2, \cdots, K$, each of which is generated as the union of all elements of the corresponding set C_i

$$S_i = \bigcup_{g \in C_i} g, \qquad i = 1, 2, \cdots, K \tag{10}$$

$$G = \{S_i, i = 1, 2, \cdots, K\}. \tag{11}$$

Segmentation fusion results are presented in Fig. 8 for the Claude sequence. Depth segmentation, shown with two different gray levels as in Fig. 7(a), is overlaid in Fig. 8(a) with the white contours of the color segments, as obtained from Fig. 6(a). It is apparent that the person in the foreground corresponds to one depth segment and to three color segments, while the background to one depth and six color segments. It is also apparent that only depth segmentation contains both objects in their entirety, while only color segmentation contains the exact object contours. One segment for each semantic object with correct boundaries can be provided by fusing color and depth segmentation results. The extracted foreground/background objects for the Claude sequence are illustrated in Fig. 8(b) and (c) respectively. Similarly, it is observed that the nine extracted objects for the Aqua sequence (Fig. 9) all correspond to semantic entities of Aqua.

Another example of segmentation fusion is illustrated in Fig. 10 for a frame of the "Eye to Eye" sequence, which is also used for summarization in Section VI. The original left channel frame is depicted in Fig. 10(a) and presents two people talking in a conference room. Color and depth segmentation is shown in Fig. 10(b) and (c) respectively. The results of segmentation fusion are illustrated in Fig. 10(d)–(f), where it is again verified that the three semantic objects are accurately obtained.

In all previous cases, semantic object identification cannot be achieved by using color segmentation only, since usually an object consists of multiple regions with different color characteristics. In order to compare the two techniques, the number of color segments should be reduced by appropriately regulating the distance threshold. Fig. 11 shows the color segmentation results for the previously described frames of the Claude, Aqua, and "Eye to Eye" sequences using a distance threshold such that the total number of color segments is the same as the number of semantic objects presented in Figs. 8(b), (c), 9, and 10(d)–(f), respectively. The results obtained are not satisfactory since regions corresponding to different objects have been merged together. Instead, combining color and depth information a more meaningful visual content representation is provided, justifying the additional computational cost for depth segmentation. In some cases, however, especially for long shots, disparity differences are small and depth cannot be accurately estimated. These cases are detected since they usually result in only one depth segment, and subsequently, depth information is discarded and segmentation is based on color only.

IV. Fuzzy Feature Vector Formulation

The visual content of a frame is described by extracting several features from each segment (object). All these features are gathered to form a frame feature vector. However, since the number of segments varies from frame to frame, the feature vector length also varies. Thus, any comparison between feature vectors of different frames is practically unfeasible. To overcome this problem, we classify frame segments into pre-determined classes, forming a multidimensional histogram. In this framework, each element of a feature vector corresponds to a specific class, or equivalently to a histogram bin. In order to reduce the possibility of classifying two similar segments to different classes, causing erroneous comparisons, a degree of membership is allocated to each class, resulting in a fuzzy classification formulation [41], [42]. In this case, each sample is allowed to belong to several (or all) classes with different degrees of membership. Therefore, two similar samples are not classified to different bins as in conventional histograms.

In particular, for each segment S_i, $i = 1, \cdots, K$, we form an $L \times 1$ vector $\boldsymbol{s}_i = [\boldsymbol{c}^T(S_i) d(S_i) \boldsymbol{l}^T(S_i) a(S_i)]^T$, where the

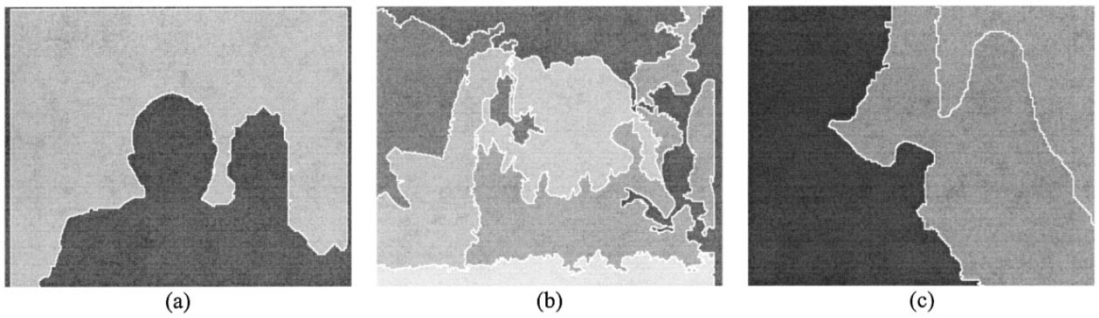


Fig. 11. Color segmentation with a limited target number of segments. (a) Claude sequence. (b) Aqua sequence. (c) "Eye to Eye" sequence.

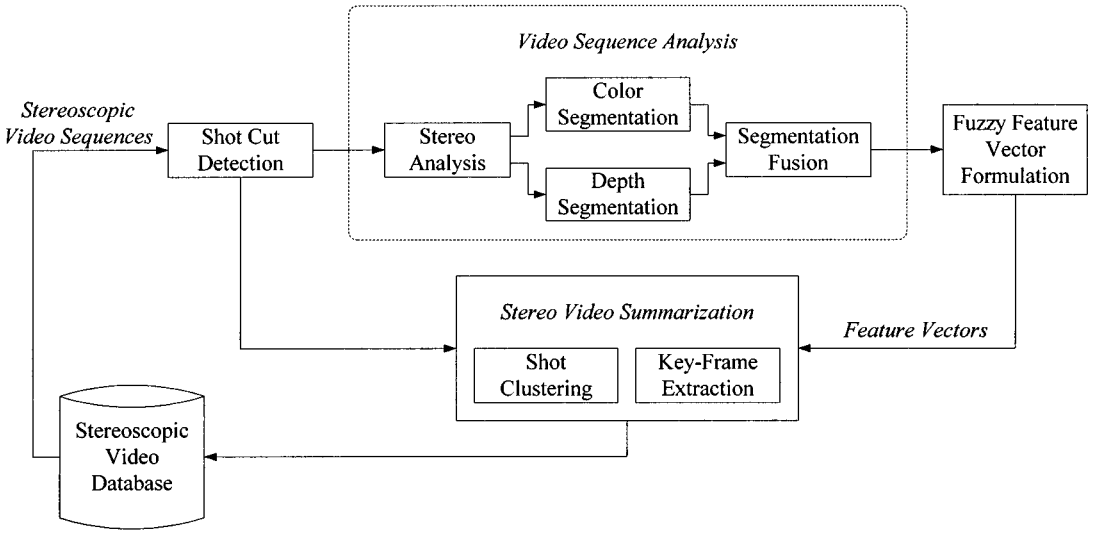


Fig. 12. Block diagram of the proposed scheme for summarization of stereoscopic video sequences.

3×1 vector $\boldsymbol{c}$ includes the average color components of segment S_i, d is its depth, a is its size, and $\boldsymbol{l}$ is a 2×1 vector containing the horizontal and vertical location of the segment center. The domain of jth element $s_{i,j}$, $j = 1, 2, \cdots, L$ of $\boldsymbol{s}_i$ is partitioned into Q regions using the membership functions $\mu_{n_j}(s_{i,j})$, $n_j = 1, 2, \cdots, Q$. The $\mu_{n_j}(s_{i,j})$ denotes the degree of membership of $s_{i,j}$ to the class with index n_j. Gathering class indices n_j for all elements $j = 1, 2, \cdots, L$, an L-dimensional class $\boldsymbol{n} = [n_1\ n_2\ \cdots\ n_L]^T$ is defined. Then, the degree of membership of each vector $\boldsymbol{s}_i$ to class $\boldsymbol{n}$ can be performed through a product of membership functions $\mu_{n_j}(s_{i,j})$ of all elements $s_{i,j}$ of $\boldsymbol{s}_i$ to the respective elements n_j of $\boldsymbol{n}$

$$\mu_{\boldsymbol{n}}(\boldsymbol{s}_i) = \prod_{j=1}^{L} \mu_{n_j}(s_{i,j}). \tag{12}$$

A multidimensional fuzzy histogram is constructed, gathering all feature samples $\boldsymbol{s}_i$, $i = 1, \cdots, K$. The value of the fuzzy histogram $H(\boldsymbol{n})$ is defined as the sum, over all segments, of the corresponding degrees of membership $\mu_{\boldsymbol{n}}(\boldsymbol{s}_i)$

$$H(\boldsymbol{n}) = \frac{1}{K}\sum_{i=1}^{K} \mu_{\boldsymbol{n}}(\boldsymbol{s}_i) = \frac{1}{K}\sum_{i=1}^{K}\prod_{j=1}^{L} \mu_{n_j}(s_{i,j}). \tag{13}$$

Thus, $H(\boldsymbol{n})$ can be viewed as a degree of membership of a whole frame to class $\boldsymbol{n}$. A frame feature vector $\boldsymbol{f}$ is then formed by gathering values of $H(\boldsymbol{n})$ for all classes $\boldsymbol{n}$, i.e., for all combinations of indices, resulting in a total of $M = Q^L$ feature elements: $\boldsymbol{f} = [f_1\ f_2\ \cdots\ f_M]^T$. Note that a large number of partitions does not necessarily improve key-frame extraction effectiveness, as it results in "noisy" classification. Based on experiments, a reasonable choice with respect to complexity and effectiveness is $Q = 3$.

V. Stereo–Video Summarization

In order to analyze an entire set of stereoscopic video sequences in a database and summarize their visual content, several tasks are required, as illustrated in the block diagram of Fig. 12. First, since a video sequence is a collection of different shots, each of which corresponds to a continuous action of a single camera operation [19], a *shot-cut detection* algorithm is applied. Several algorithms have been reported in the literature for shot-change detection of 2-D video sequences which deal with the detection of cut, fading, or dissolve changes either in compressed or uncompressed domain [19], [43]. Since, in case of 3-D video sequences, a shot change occurs at the same frame instance for all multiview channels, the aforementioned algorithms can be applied to one channel, e.g., the left. In our approach the algorithm proposed in [19] has been adopted due to its efficiency and low computational complexity.

Then, the aforementioned video sequence analysis is applied to every frame (stereo pair) for the construction of feature vec-

tors, and a content-based sampling algorithm is used for summarizing the stereo information by discarding shots or frames of similar visual content. In particular, a shot feature vector is constructed based on the feature vectors of all frames in each shot, and shot selection is accomplished by clustering similar shots together and selecting a limited number of shot cluster representatives.

A. Shot Selection

Let $\boldsymbol{h}_i \in \Re^M$, $i = 1, 2, \cdots, N_S$ be the shot feature vector for the ith shot, calculated as the average of all frame feature vectors within the respective shot; N_S is the total number of shots in a sequence and $M = Q^L$ is the feature vector length. Then, $E = \{\boldsymbol{h}_i, i = 1, 2, \cdots, N_S\}$ is the set of all shot feature vectors. Let K_S be the number of shots to be selected and $\boldsymbol{q}_i, i = 1, 2, \cdots, K_S$ the feature vectors that best represent these shots (shot representatives). For each $\boldsymbol{q}_i$, an influence set is formed, say Z_i which contains all shot feature vectors $\boldsymbol{h} \in E$ that are closest to $\boldsymbol{q}_i$

$$Z_i = \{\boldsymbol{h} \in E\colon \delta_S(\boldsymbol{h}, \boldsymbol{q}_i) < \delta_S(\boldsymbol{h}, \boldsymbol{q}_j) \forall j \neq i\}, \quad i = 1, 2, \cdots, K_S \tag{14}$$

where $\delta_S(\cdot)$ denotes the distance between two vectors, e.g., the Euclidean norm. Then the average distortion, defined as

$$R_S(\boldsymbol{q}_1, \boldsymbol{q}_2, \cdots, \boldsymbol{q}_{K_S}) = \sum_{i=1}^{K_s} \sum_{\boldsymbol{h} \in Z_i} \delta_S(\boldsymbol{h}, \boldsymbol{q}_i) \tag{15}$$

is a performance measure of the representation of shot feature vectors by the cluster centers $\boldsymbol{q}_i$. The optimal vectors $\hat{\boldsymbol{q}}_i$ are thus calculated by minimizing R_S. However, direct minimization of R_S is a tedious task since the unknown parameters are involved both in distances $\delta_S(\cdot)$ and influence zones. For this reason, minimization is performed in an iterative way using the generalized Lloyd-Max or *K-means* algorithm [28]. Starting from arbitrary initial values $\boldsymbol{q}_i(0)$, $i = 1, 2, \cdots, K_S$, the new centers are calculated through the following equations for $n \geq 0$:

$$Z_i(n) = \{\boldsymbol{h} \in E\colon \delta_S(\boldsymbol{h}, \boldsymbol{q}_i(n)) < \delta_S(\boldsymbol{h}, \boldsymbol{q}_j(n)) \forall j \neq i\}, \quad i = 1, 2, \cdots, K_S \tag{16a}$$

$$\boldsymbol{q}_i(n+1) = cent(Z_i(n)), \qquad i = 1, 2, \cdots, K_S \tag{16b}$$

where $\boldsymbol{q}_i(n)$ denotes the ith center at the nth iteration, and $Z_i(n)$ its influence set. The center of $Z_i(n)$ is estimated by the function

$$cent(Z_i(n)) = \frac{1}{|Z_i(n)|} \sum_{\boldsymbol{h} \in Z_i(n)} \boldsymbol{h} \tag{17}$$

where $|Z_i(n)|$ is the cardinality of $Z_i(n)$. The algorithm converges to the solution $(\hat{\boldsymbol{q}}_1, \hat{\boldsymbol{q}}_2, \cdots, \hat{\boldsymbol{q}}_{K_S})$ after a small number of iterations. Finally, the K_S most representative shots are extracted as the ones whose feature vectors are closest to $(\hat{\boldsymbol{q}}_1, \hat{\boldsymbol{q}}_2, \cdots, \hat{\boldsymbol{q}}_{K_S})$, since there is no guarantee that the cluster centers correspond to actual shot feature vectors.

Note that as the number of selected shots K_S increases, the average distortion decreases, since the shot feature vectors are closer to the centers of the shot clusters. In a case where $K_S = N_S$, each shot represents a cluster and thus the average distortion R_S reaches zero. On the other hand, small values of K_S are usually desired in order to reduce storage requirements and achieve efficient 3-D video summarization. The optimal value of K_S is estimated using an information-theoretic criterion, namely the *maximum description length* (MDL) [44].

B. Key-Frame Extraction

After extracting the most representative shots, the next step is to select the key frames within each one of the selected shots. This is achieved by minimizing a cross correlation criterion, so that the selected frames are not similar to each other. Let us denote by $\boldsymbol{f}_i \in \Re^M$, $i \in V = \{1, \cdots, N_F\}$ the feature vector of the ith frame of a given shot, where N_F is the total number of frames of the shot. Let us assume that the K_F most characteristic frames should be selected. In order to define a measure of correlation between K_F feature vectors, we first define the *index* vector $\boldsymbol{a} = (a_1, \cdots, a_{K_F}) \in U \subset V^{K_F}$ where

$$U = \{(a_1, \cdots, a_{K_F}) \in V^{K_F}\colon a_1 < \cdots < a_{K_F}\} \tag{18}$$

is the subset of V^{K_F} which contains all sorted index vectors $\boldsymbol{a}$. Each index vector $\boldsymbol{a}$ corresponds to a set of frame numbers. The *correlation measure* of K_F feature vectors $\boldsymbol{f}_i$, $i = a_1, \cdots, a_{K_F}$ is then defined as

$$R_F(\boldsymbol{a}) = R_F(a_1, \cdots, a_{K_F}) = \frac{2}{K_F(K_F-1)} \sum_{i=1}^{K_F-1} \sum_{j=i+1}^{K_F} (\rho_{a_i, a_j})^2 \tag{19}$$

where ρ_{a_i, a_j} is the correlation coefficient of feature vectors $\boldsymbol{f}_{a_i}$ and $\boldsymbol{f}_{a_j}$.

Hence, searching for a set of K_F minimally correlated feature vectors is equivalent to searching for an index vector $\boldsymbol{a}$ that minimizes $R_F(\boldsymbol{a})$. Searching is limited in subset U. Therefore any permutations of the elements of $\boldsymbol{a}$ results in the same key frames. The set of the K_F least correlated feature vectors, corresponding to the K_F key frames, is thus represented by

$$\hat{\boldsymbol{a}} = (\hat{a}_1, \cdots, \hat{a}_{K_F}) = \arg\min_{\boldsymbol{a} \in U} R_F(\boldsymbol{a}). \tag{20}$$

Unfortunately, the complexity of an exhaustive search for the minimum value of $R_F(\boldsymbol{a})$ is such that a direct implementation would be practically unfeasible, since the multidimensional space U includes all possible sets (combinations) of frames. A dramatic reduction in complexity can be achieved through *logarithmic search*, which has been introduced in [45] and provides a very fast convergence to a sub-optimal solution. However, since the search procedure is by definition confined to a very small, pre-defined subset of the search space U, there is always a significant possibility of converging to a local minimum of $R_F(\boldsymbol{a})$, resulting in poor performance. For this reason, a *genetic algorithm* (GA) [40] approach is used in this paper. Both search algorithms are based on the assumption that frames which are close to each other (in time) have similar properties, therefore indices which are close to each other (in U) have similar correlation measures.

Fig. 13. "Eye to Eye" sequence with $N_S = 76$ shots, shown with one frame per shot.

VI. Experimental Results

The 3-D stereoscopic television program "Eye to Eye" [46], of total duration 25 minutes (12 739 frames at 10 frames/s), has been used in our experiments for the evaluation of the proposed summarization scheme. The sequence was produced in the framework of the ACTS MIRAGE project [47] 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 [48], while location action shots were captured using a special lightweight and rugged stereo cam built for the ITC by AEA Technology.

The stereo video sequence is first analyzed and for each stereo pair, a depth map is estimated. The M-RSST algorithm is then applied on both the depth map and the left channel image, and the resulting depth and color segments are fused together. Proper features are derived from the final segmentation, including segment size, location, color, and depth. Each feature domain is partitioned in $Q = 3$ classes and fuzzy classification with triangular membership functions of 50% overlap is used for the construction of feature vectors, so that the total feature vector length is $Q^L = 2\,187$ since, in our case, $L = 7$. The shot detection and feature extraction algorithms are applied offline to the sequence, so that all information regarding shot change instances, as well as the feature vector representation of all video frames, is stored in a database and readily available. Hence, algorithms for shot clustering and key-frame extraction are separately applied using feature vectors of all frames and shots.

The problem of shot clustering and selection is addressed first. The entire sequence under examination consists of $N_S = 76$ shots, which are depicted in Fig. 13. For presentation purposes, each shot is depicted by one frame, whose feature vector is closest to the respective shot feature vector, i.e., the average feature vector over all frames of the shot. In this experiment, the number of shot clusters has been selected to be $K_S = 10$. This number of clusters is estimated using the MDL criterion [44] as described in Section V-A. The results of the shot clustering mechanism are illustrated in Figs. 14 (clusters 1–5) and 15 (clusters 6–10). As is observed, most of the shots containing similar visual content, in terms of the number and complexity of objects, are assigned to the same shot cluster. Fig. 16 depicts the ten shot cluster representatives, which are selected as the shots whose feature vector is closest to the corresponding cluster centers. It is clear that the visual content of the 25-min sequence is efficiently summarized by the ten extracted shots of this figure. Thus, it is possible to automatically generate low-resolution video clip previews (trailers) or still image mosaics of stereo –video sequences.

In order to evaluate the added benefit of including depth map estimation and segmentation fusion in the proposed stereo-summarization system, a comparison between single channel and stereo–video summarization is also accomplished. In particular, color segmentation, without depth estimation and fusion, is applied to the left channel only, and the respective summarization

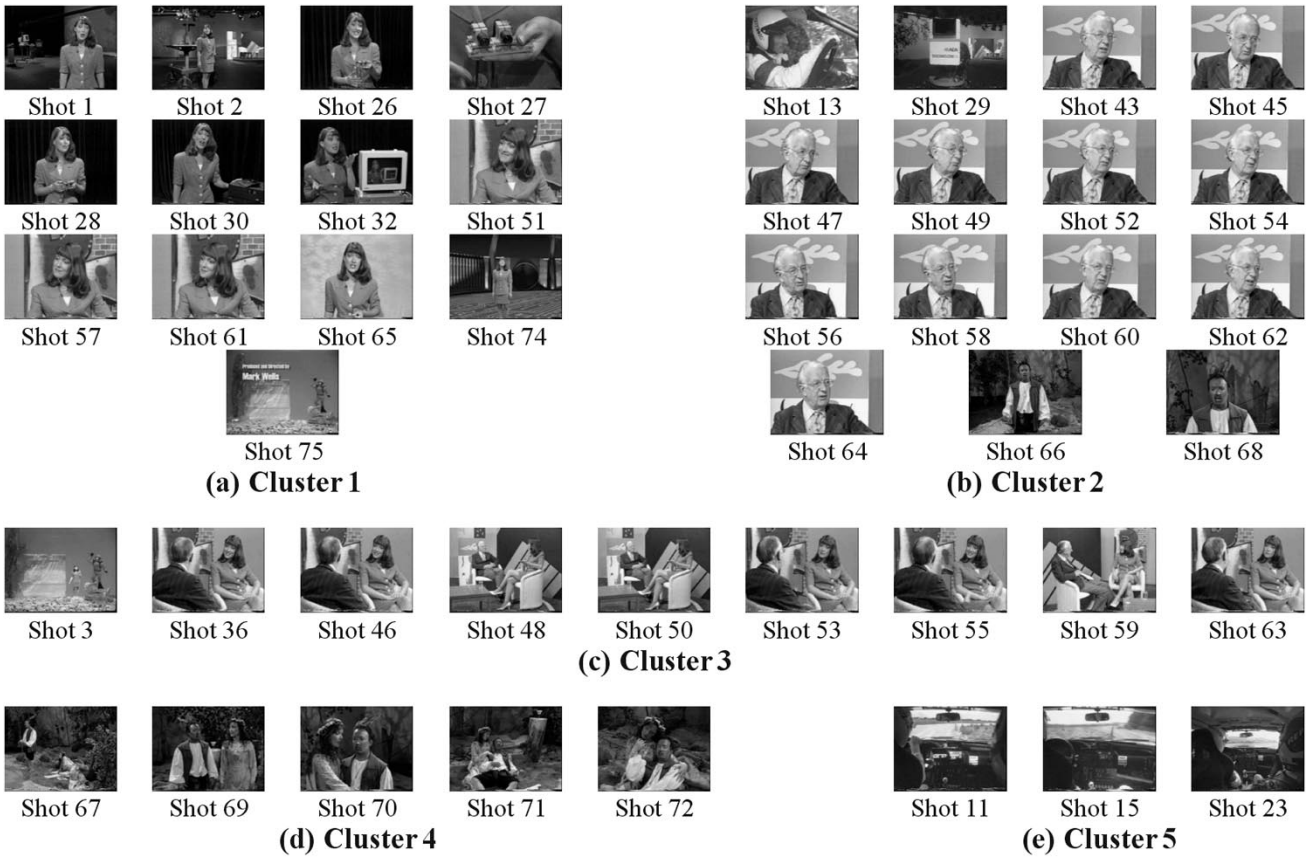


Fig. 14. Shot clusters 1–5 from "Eye to Eye" sequence (with color and depth segment fusion).

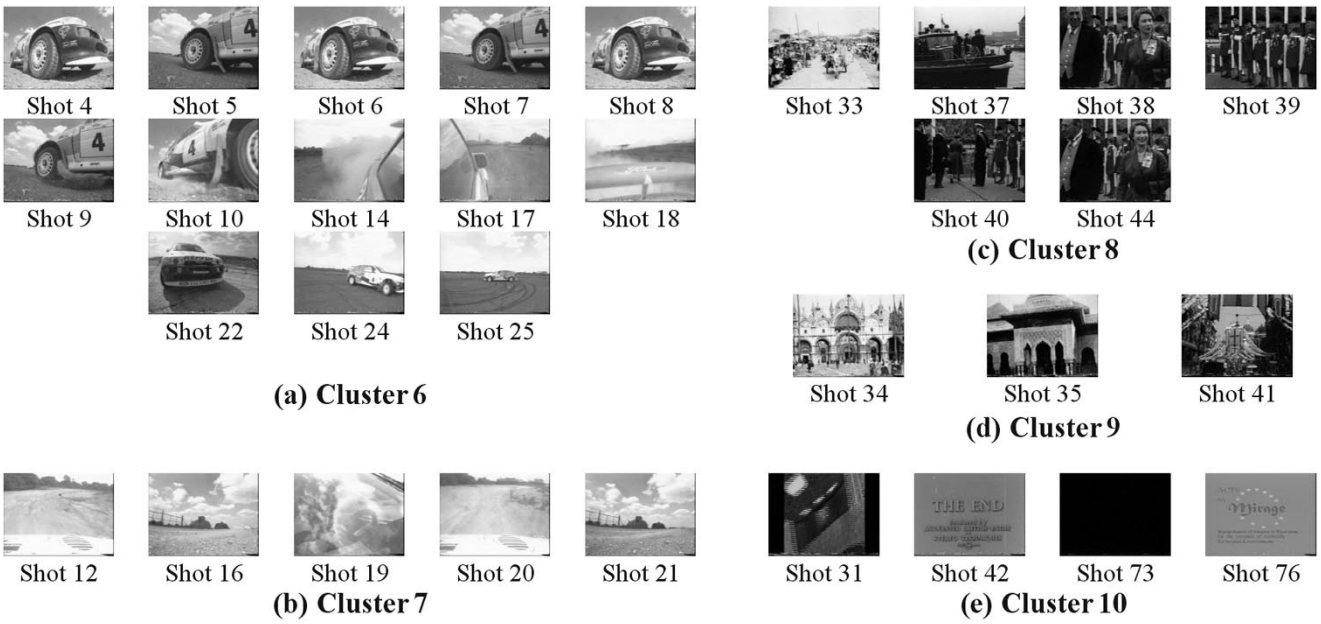


Fig. 15. Shot clusters 6–10 from "Eye to Eye" sequence (with color and depth segment fusion).

results are shown in Figs. 17 (clusters 1–5) and 18 (clusters 6–10). It is clear that shot separation according to visual content is not as successful as in the case of the stereo sequence, where the disparity field is used. For example, using color and depth information, the shots illustrating the hostess are classified to the same group (cluster 1). Using color information only, those shots are assigned to more than one clusters (mainly clusters 2 and 3). Similarly, the shot cluster representatives using color segmentation only are depicted in Fig. 19. It can be observed that the selected key shots contain multiple instances of the same visual content, while certain distinctive shots of long duration do not have an associated cluster representative.

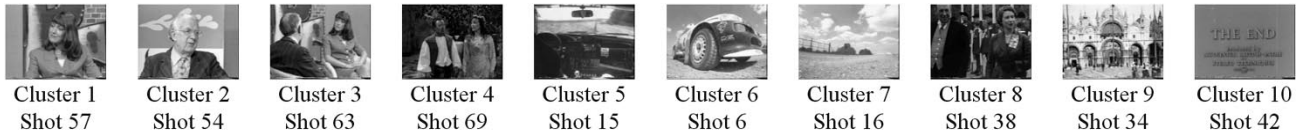


Fig. 16. The $K_S = 10$ selected shot cluster representatives from "Eye to Eye" sequence with color and depth segment fusion.

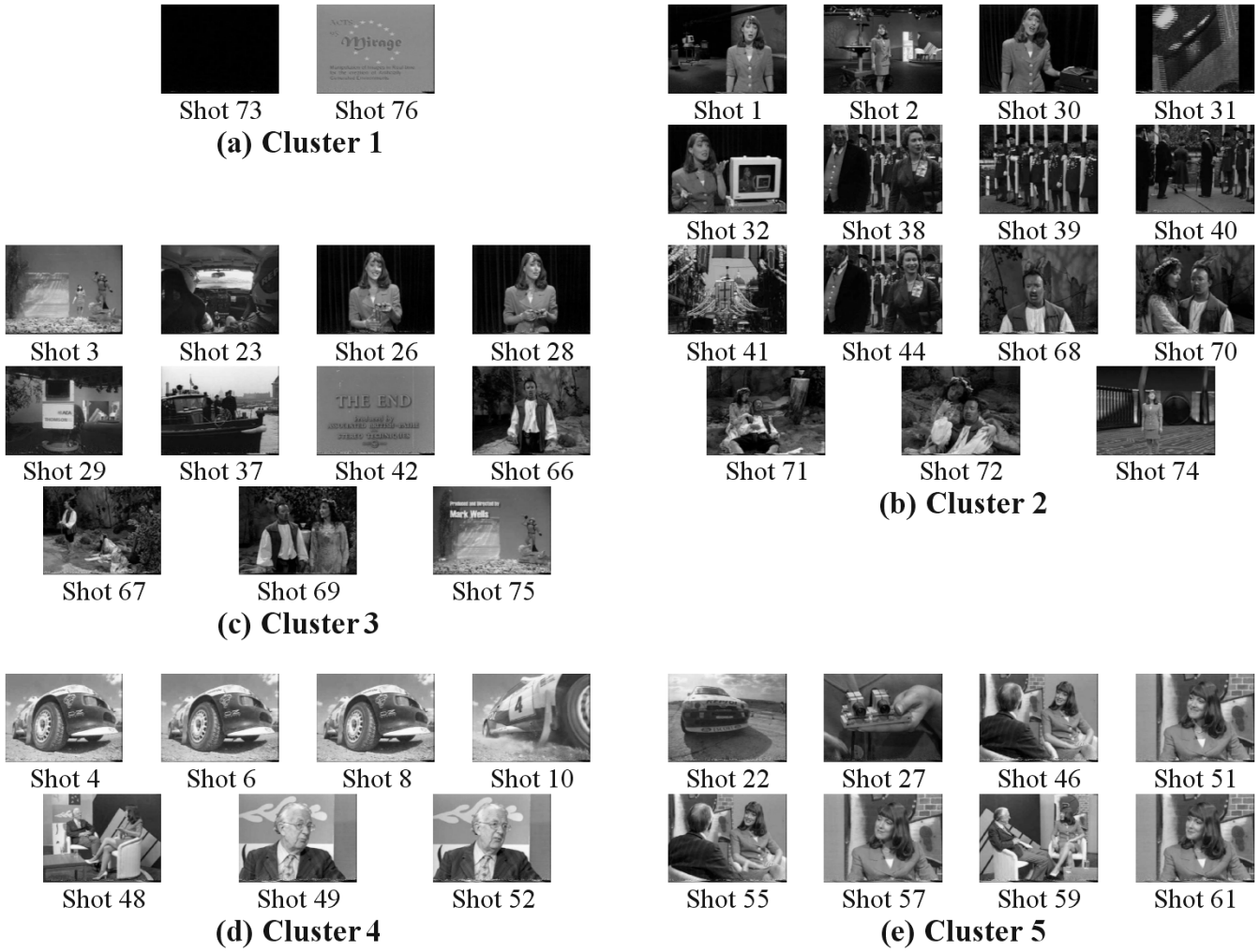


Fig. 17. "Shot clusters 1–5 from "Eye to Eye" sequence (left channel with color segmentation only).

Note also that the additional computational cost for disparity estimation and depth segmentation is not very restrictive. For example, single channel processing using color segmentation only takes about 4.37 s/frame on a Sun Ultra 10 (333 MHz) workstation for a frame size of 352×264 pixels. Instead, two-channel processing with disparity estimation and fusion requires about 9.91 s/frame (including color segmentation). The above processing times are averaged over all frames of the whole sequence. In order to accelerate processing of stereo sequences, a sub-sampled version of the two channel images is used (blocks of 2×2 pixels) for depth estimation and segmentation. Note that with further sub-sampling (blocks of 4×4 pixels) the total corresponding processing time is 5.85 s/frame, which is comparable to single channel processing. Such sub-sampling does not significantly affect object extraction performance, since accurate object contours are obtained through color and depth segment fusion. Further reduction of the computational complexity for estimating the disparity field can be achieved using the algorithm proposed recently in [26]. Thus, it can be claimed that the added benefit of using depth information justifies the additional computational cost.

In some cases, however, the disparity differences among objects (discretized to image resolution) are typically small, resulting in an erroneous estimation of the depth. This happens especially for long shots where different depths of objects cannot be accurately detected. However, this is not the most common case, especially for stereoscopic video sequences, which are usually produced so that depth information is of primary importance. In particular, in the "Eye to Eye" sequence used in our experiments, 532 frames out of 12 739 are detected as frames of no significant depth information, i.e., 4.18% of the total sequence. These shots are detected since only one depth segment is extracted, and then processed using color information only.

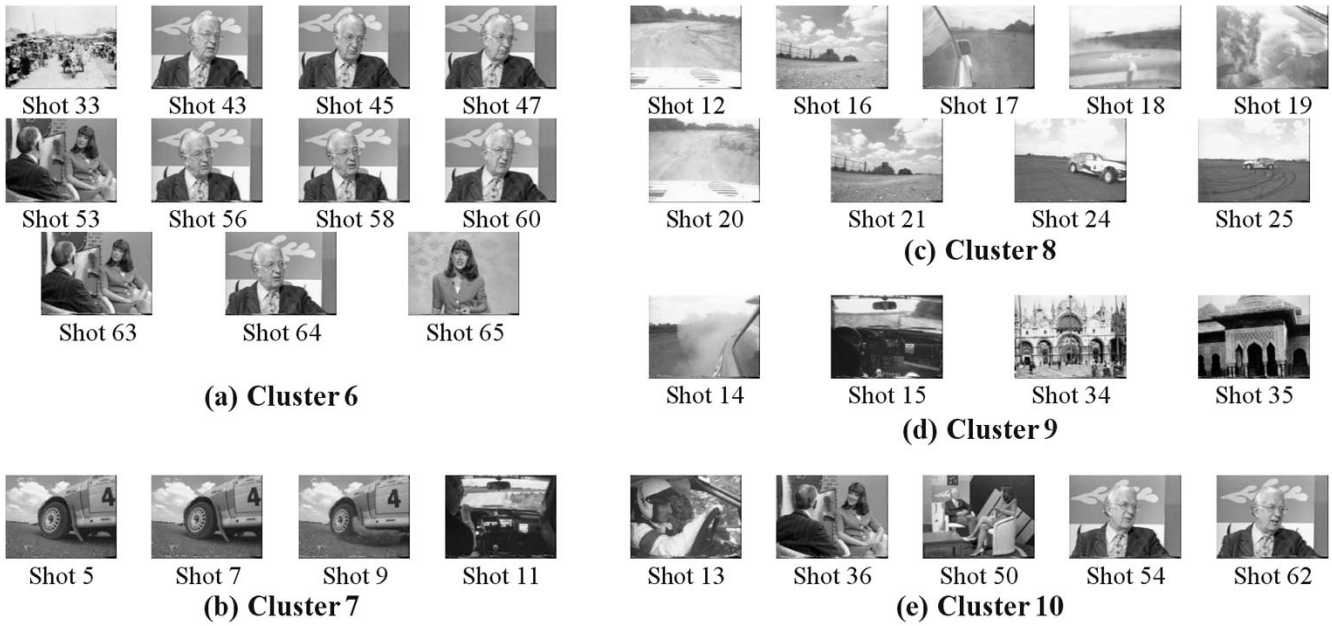


Fig. 18. Shot clusters 6–10 "Eye to Eye" sequence (left channel with color segmentation only).

Fig. 19. Shot cluster representatives (left channel with color segmentation only).

Fig. 20. Shot 38 from "Eye to Eye" sequence with $N_F = 188$ frames, shown with one frame every seven.

The proposed key-frame selection mechanism is evaluated using two different shots. The first, shot 38, consists of $N_F = 188$ frames (stereo pairs) and represents an outdoor crowded scene with considerable camera motion, while the second, shot 69, consists of $N_F = 726$ frames and represents a studio scene with two people and very limited motion. For presentation purposes, one frame every seven is shown for the first shot in Fig. 20, while one frame every 20 is shown for the second in Fig. 21, since it is much longer and involves less action. The results of the cross-correlation approach using the genetic implementation are shown in Fig. 22(a) and Fig. 23(a) respectively. For the first shot, $K_F = 4$ key frames are extracted, while for the second, $K_F = 5$. These numbers are estimated by examining the temporal variation of the frame feature vectors of each shot, as described in [40]. Although a very small percentage of frames is retained, it is clear that, in both cases, one can visualize the content of the shots by just examining the selected key frames.

Key-frame selection is also compared with the single channel case, applying color segmentation only. The respective results are presented in Figs. 22(b) and 23(b) for the two shots. It is observed that the extracted key frames cannot efficiently describe the visual content of the shot. In particular, the first two extracted key frames of shot 38 are of similar content, while the first two and the last two frames of shot 69 also present similar visual characteristics. Furthermore, there is no key frame with visual content similar to that of the frame 9 598 of shot 69 as it happens in the two-channel case where key frame 9 599 has been extracted.

Fig. 21. Shot 69 from "Eye to eye" sequence with $N_F = 726$ frames, shown with one frame every 20.

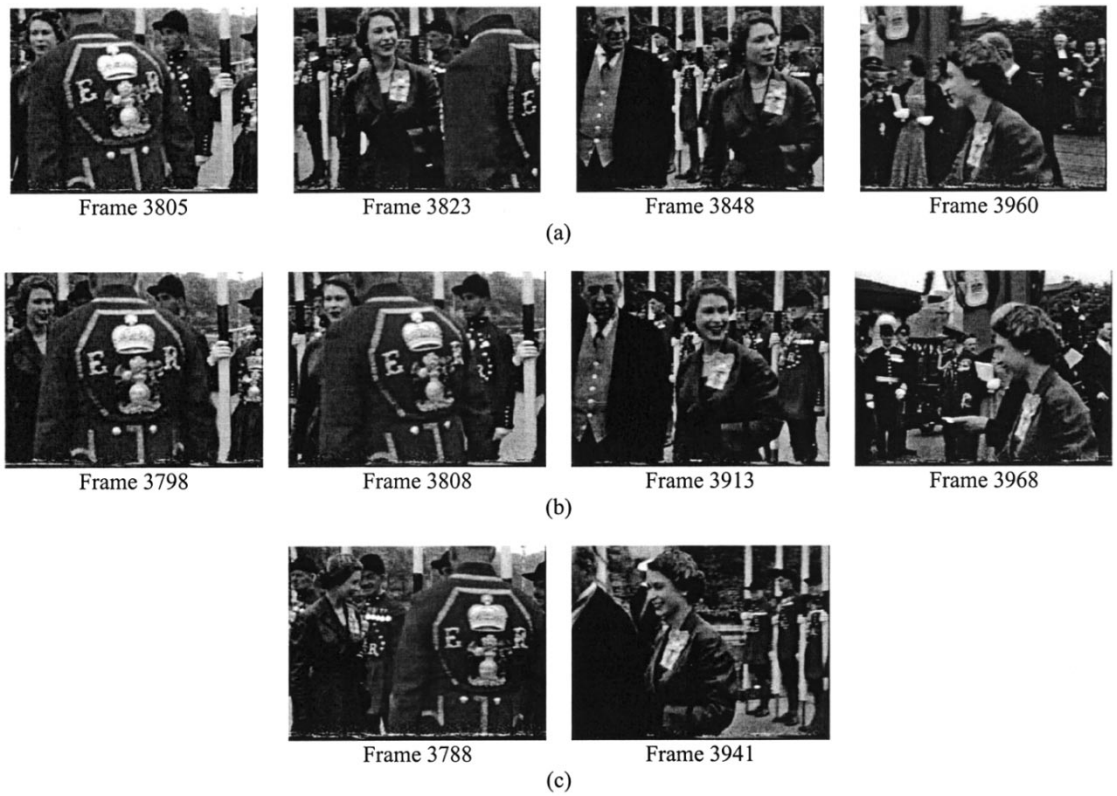


Fig. 22. Key frames from shot 38 selected by: (a) the genetic algorithm with both channels; (b) genetic algorithm with left channel with color segmentation only; and (c) the method of [19].

Finally, the proposed key-frame extraction method is compared with the one presented in [19]. Key frames are extracted at time instances when the accumulated differences of the dc images exceed a pre-determined threshold. The key frames extracted using this method for the shots of Figs. 20 and 21 are presented in Figs. 22(c) and 23(c), respectively. The selection of the threshold value is an *ad hoc* process and in our case it has been tuned so that the average number of key frames extracted for the whole stereo sequence is the same as that of the proposed method. However, using this threshold, two key frames are only extracted from shot 38, which are not adequate for visual content description of the shot. Furthermore, for shot 69, although the same number of key frames is extracted, the second, third, and fourth key frames have similar visual content. It is clear that the summarization performance of this technique is not so satisfactory, especially for shots where complicated camera effects are encountered.

VII. Conclusion

Stereo–video archives are anticipated to rapidly increase in the forthcoming years. However, traditionally stereo–image sequences are represented by numerous consecutive image pairs (frames), each of which corresponds to a constant time interval. Such a linear, or sequential, video representation has a number of limitations for the new emerging multimedia applications, such as video browsing, content-based indexing and retrieval.

Fig. 23. Key frames from shot 69 selected by: (a) the genetic algorithm with both channels; (b) genetic algorithm with left channel with color segmentation only; and (c) the method of [19].

Furthermore, the bandwidth and storage requirements of digitized stereo information, even in compressed form, present challenges to the most multimedia servers. For this reason, new methods for nonlinear video-content representation should be implemented. In this paper, an efficient video-content representation algorithm for stereo-image sequences has been presented. In particular, stereo video is partitioned into shots by applying a shot-cut detection algorithm, and then a content-based sampling algorithm is employed for discarding shots or frames of similar visual properties. This approach provides summarization of visual information similarly to that used in current document search engines.

Stereo-visual content is extracted by a novel segmentation algorithm, which combines both color and depth information. The adopted approach projects color segments onto depth segments so that video objects identified by depth segmentation are retained, while at the same time accurate object boundaries are extracted. A multiresolution implementation of the RSST algorithm (M-RSST) is presented to perform both color and depth segmentation. Apart from accelerating the segmentation procedure, this algorithm also prevents oversegmentation, which is not desirable in the framework of stereo video summarization. Depth is estimated from the disparity field between the left and right channel images, while occluded areas are efficiently detected and compensated with appropriate disparity values. Better performance of the visual description can be achieved by integrating tracking functionalities, thus resulting in a selection mechanism that is less susceptible to noise [49], [50]. It is illustrated in the experiments that both object extraction and video summarization perform significantly better when depth estimation and segmentation fusion are incorporated, compared to single-channel results with color segmentation only, justifying the extra computational load for disparity estimation.

Based on the obtained video-object segmentation, segment features including size, location, color, and depth are used for the construction of feature vectors based on fuzzy classification, reducing the influence of segmentation discontinuities. Consequently, a feature-based video representation is achieved instead of the traditional frame-based one, which is more suitable for stereo content description. The generalized Lloyd–Max algorithm is used for clustering shots of similar visual content due to its efficiency and computational simplicity. For a given shot, key frames are extracted by minimizing a cross-correlation criterion so that frames of minimally correlated feature vectors are located. Since an exhaustive search for the optimal solution is practically unfeasible, a genetic algorithm has been employed. Experimental results indicate reliable performance on real-life stereoscopic video recordings. Finally, the proposed stereo video summarization technique is compared to single channel summarization as well as to other techniques, with quite promising results.

ACKNOWLEDGMENT

The authors wish to thank C. Girdwood, the project manager of the ITC (Winchester), for providing the 3-D video sequence "Eye to Eye," which was produced in the framework of ACTS MIRAGE project. They also want to express their gratitude to Dr. S. Pastoor of the HHI (Berlin), for providing the video sequences of the DISTIMA project. Finally, the authors would like

to thank their colleague, G. Akrivas, for providing them with an efficient implementation of the key-frame selection technique presented in [19].

REFERENCES

[1] "Special issue on content-based image retrieval systems," *IEEE Comput. Mag.*, vol. 28, 1995.

[2] "Special issue on segmentation, description and retrieval of video content," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, Sept. 1998.

[3] S.-F. Chang, A. Eleftheriadis, and R. McClintock, "Next- generation content representation creation, and searching for new-media applications in education," *Proc. IEEE*, vol. 86, pp. 884–904, May 1998.

[4] *Coding of Moving Pictures and Associates Audio for Digital Storage Media at up to about 1.5 Mbps*, ISO/CD 11 172-2, March 1991.

[5] *Generic Coding of Moving Pictures and Associated Audio*, ISO/IEC 13818-2/H.262 Committee Draft, May 1994.

[6] *Video Codec for Audiovisual Data at $p \times 64$ kb/s*, CCITT Recommendation H.261, 1990.

[7] *Experts Group for Very Low Bitrate Visual Telephony*, ITU-T SG 15/Draft Recommendation H.263, Feb. 1995.

[8] *MPEG-4 Video Verification Model*, Version 11.0, ISO/IEC JTC1/SC29/WG11/Doc. N2172, Mar. 1998.

[9] T. Sikora, "The MPEG-4 video standard verification model," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, pp. 19–31, Feb. 1997.

[10] L. Chiariglione, "MPEG and multimedia communications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, pp. 5–18, Feb. 1997.

[11] *MPEG-7: Context and Objectives* , (v.5), ISO/IEC JTC1/SC29/WG11/Doc. N1920, Oct. 1997.

[12] L. Garrido, F. Marques, M. Pardas, P. Salembier, and V. Vilaplana, "A hierarchical technique for image sequence analysis," in *Proc. Workshop Image Analysis for Multimedia Interactive Services (WIAMIS)*, Louvain-la-Neuve, Belgium, June 1997, pp. 13–20.

[13] M. M. Yeung and B.-L. Yeo, "Video visualization for compact presentation and fast browsing of pictorial content," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, pp. 771–785, Oct. 1997.

[14] M. Irani and P. Anandan, "Video indexing based on mosaic representation," *Proc. IEEE*, vol. 86, pp. 805–921, May 1998.

[15] Y. Avrithis, N. Doulamis, A. Doulamis, and S. Kollias, "Efficient content representation in MPEG video databases," in *Proc. of IEEE Int. Conf. Computer Vision and Pattern Recognition (CVPR)*, Santa Barbara, CA, June 1998.

[16] R. V. Cox, B. G. Haskel, Y. LeCun, B. Shahraray, and L. Rabiner, "On the applications of multimedia processing to communications," *Proc. IEEE*, vol. 86, pp. 755–824, May 1998.

[17] N. Doulamis, A. Doulamis, G. Konstantoulakis, and G. Stassinopoulos, "Efficient modeling of VBR MPEG-1 video sources," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, pp. 93–113, Feb. 2000.

[18] S. W. Smoliar and H. J. Zhang, "Content-based video indexing and retrieval," *IEEE Multimedia*, pp. 62–72, Summer 1994.

[19] B. L. Yeo and B. Liu, "Rapid scene analysis on compressed videos," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 5, pp. 533–544, Dec. 1995.

[20] M. Mills, J. Cohen, and Y. Y. Wong, "A magnifier tool for video data," in *Proc. ACM Computer Human Interface (CHI)*, May 1992, pp. 93–98.

[21] F. Arman, R. Depommier, A. Hsu, and M. Y. Chiu, "Content-based browsing of video sequences," *ACM Multimedia*, pp. 77–103, Aug. 1994.

[22] N. Vasconcelos and A. Lippman, "A spatiotemporal motion model for video summarization," in *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition (CVPR)*, Santa Barbara, CA, June 1998, pp. 361–366.

[23] B. Shahraray, "Scene change detection and content-based sampling of video sequences," in *Proc. SPIE 2419: Digital Video Compression: Algorithms and Technologies*, Feb 1995, pp. 2–13.

[24] C. Gu and M.-C. Lee, "Semiautomatic segmentation and tracking of semantic video objects," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, pp. 572–584, Sept. 1998.

[25] A. Alatan, L. Onural, M. Wollborn, R. Mech, E. Tuncel, and T. Sikora, "Image sequence analysis for emerging interactive multimedia services—The European cost 211 framework," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, pp. 802–813, Nov. 1998.

[26] C.-J. Tsai and A. K. Katsaggelos, "Dense disparity estimation with a divide-and conquer disparity space image technique," *IEEE Trans. Multimedia*, vol. 1, no. 1, pp. 18–29, Mar. 1999.

[27] D. Tzovaras, N. Grammalidis, and M. G. Strintzis, "Object-based coding of stereo image sequences using joint 3-D motion/disparity compensation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, pp. 312–327, April 1997.

[28] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Norwell, MA: Kluwer Academic Publishers, 1993.

[29] S. Barnard and W. Thompson, "Disparity analysis of images," *IEEE Trans. Pattern Anal. Machine Intell.*, pp. 333–340, 1980.

[30] I. Kompatsiaris, D. Tzovaras, and M. G. Strintzis, "3-D model based segmentation of videoconference image sequences," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, pp. 547–561, Sept. 1998.

[31] D. Tzovaras, N. Grammalidis, and M. G. Strintzis, "Disparity field and depth map coding for multiview 3D image generation," *Signal Processing: Image Commun.*, no. 11, pp. 205–230, 1998.

[32] 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 Syst. Video Technol.*, vol. 8, pp. 328–344, June 1998.

[33] T. Meier and K. N. Ngan, "Automatic segmentation of moving objects for video object plane generation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, pp. 525–538, Sept. 1998.

[34] N. Doulamis, A. Doulamis, D. Kalogeras, and S. Kollias, "Very low bit-rate coding of image sequences using adaptive regions of interest," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, pp. 928–934, Dec. 1998.

[35] R. Castagno, T. Ebrahimi, and M. Kunt, "Video segmentation based on multiple features for interactive multimedia applications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, pp. 562–571, Sept. 1998.

[36] A. Doulamis, N. Doulamis, and S. Kollias, "On line retrainable neural networks: Improving the performance of neural networks in image analysis problems," *IEEE Trans. Neural Networks*, vol. 11, pp. 137–155, Jan. 2000.

[37] K. N. Ngan, S. Panchanathan, T. Sikora, and M.-T. Sun, "Guest editorial," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, Sept 1998.

[38] O. J. Morris, M. J. Lee, and A. G. Constantinides, "Graph theory for image analysis: An approach based on the shortest spanning tree," *Inst. Elect. Eng. Proc.*, vol. 133, pp. 146–152, April 1986.

[39] P. J. Mulroy, "Video content extraction: Review of current automatic segmentation algorithms," in *Proc. Workshop Image Analysis and Multimedia Interactive Systems (WIAMIS)*, Louvain-la-Neuve, Belgium, June 1997.

[40] Y. Avrithis, A. Doulamis, N. Doulamis, and S. Kollias, "A stochastic framework for optimal key frame extraction from MPEG video databases," *Comput. Vis. Image Understanding*, vol. 75, no. 1/2, pp. 3–24, July/August 1999.

[41] A. Doulamis, Y. Avrithis, N. Doulamis, and S. Kollias, "Interactive content-based retrieval in video databases using fuzzy classification and relevance feedback," in *Proc. IEEE Int. Conf. Multimedia Computing and Systems (ICMCS)*, Florence, Italy, June 1999.

[42] B. Kosko, *Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence*. Englewood Cliffs, NJ: Prentice-Hall, 1992.

[43] N. V. Patel and I. K. Sethi, "Video shot detection and characterization for video databases," *Pattern Recognit.*, vol. 30, no. 4, pp. 583–592, April 1997.

[44] S. Haykin, *Adaptive Filter Theory*. Englewood Cliffs, NJ: Prentice-Hall, 1996.

[45] N. Doulamis, A. Doulamis, Y. Avrithis, and S. Kollias, "Video content representation using optimal extraction of frames and scenes," in *Proc. IEEE Int. Conf. Image Processing (ICIP)*, Chicago, IL, Oct. 1998.

[46] J. Slater, "Eye to eye with stereoscopic TV," *Image Technol.*, p. 23, Nov./Dec. 1996.

[47] C. Girdwood and P. Chiwy, "MIRAGE: An ACTS project in virtual production and stereoscopy," *IBC Conf. Pub.*, no. 428, pp. 155–160, Sept. 1996.

[48] M. Ziegler, "Digital stereoscopic imaging and applications: A way toward new dimensions, the RACE II project DISTIMA," in *Proc. IEE Colloquium Stereoscopic Television*, London, U.K., 1992.

[49] S. Malassiotis and M. Strintzis, "Tracking textured deformable objects using a finite-element mesh," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, pp. 756–774, Oct. 1998.

[50] Y. Wang and O. Lee, "Active mesh—A feature seeking and tracking image sequence representation scheme," *IEEE Trans. Image Processing*, vol. 3, pp. 610–624, Sept. 1994.

Nikolaos D. Doulamis (S'96) was born in Athens, Greece, in 1972. He received the Diploma degree in electrical and computer engineering from the National Technical University of Athens (NTUA), Athens, Greece, in 1995, with highest honors. He is currently working toward the Ph.D. degree in the Electrical and Computer Engineering Department, NTUA, supported by the Bodosakis Foundation Scholarship.

His research interests include 2-D and 3-D image/video analysis, processing and coding, multimedia systems, and content-based indexing and retrieval.

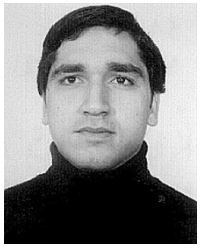
Mr. Doulamis is the recipient of several awards and prizes from the Technical Chamber of Greece, the NTUA, and the National Scholarship Foundation, including the Best New Engineer Award at national level, the Best Diploma Thesis Award, and the NTUA medal. He has also received the Eugenidiou Foundation Prize as the best newly accepted Ph.D. candidate of the NTUA in 1997.

Anastasios D. Doulamis (S'96) was born in Athens, Greece, in 1972. He received the Diploma degree in electrical and computer engineering from the National Technical University of Athens (NTUA), Athens, Greece, in 1995, with the highest distinction, and is currently working toward the Ph.D. degree.

He joined the Image, Video, and Multimedia Lab of the NTUA in 1996. His research interests include neural networks to image/signal processing, multimedia systems and video coding based on neural networks systems.

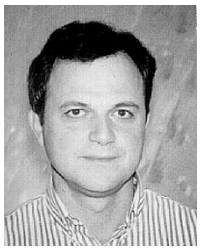
Mr. Doulamis was awarded the Best New Engineer at national level and received the Best Diploma Thesis Award by the Technical Chamber of Greece. He was also awarded the NTUA medal, and has received several awards and prizes from the National Scholarship Foundation.

Yannis S. Avrithis (S'96) received the Diploma degree in electrical and computer engineering from the National Technical University of Athens, Greece, in 1993, and the M.Sc. degree in electrical and electronic engineering (communications and signal processing) from the Imperial College of Science, Technology and Medicine, London, U.K., in 1994. He is currently working toward the Ph.D. degree in the Department of Electrical and Computer Engineering, National Technical University of Athens. His research interests include digital image and video processing, image segmentation, affine-invariant image representation, content-based indexing and retrieval, and video summarization.



Klimis S. Ntalianis (S'99) was born in Athens, Greece, in 1975. He received the Diploma degree in electrical and computer engineering from the National Technical University of Athens (NTUA), Athens, Greece, in 1998. He is currently working toward his Ph.D. degree with support from the National Scholarship Foundation. He has also received a scholarship from the Institute of Communications and Computers Systems of the NTUA as one of the best newly accepted Ph.D. students.

His research interests include 3-D image processing, video summarization, content-based indexing and retrieval as well as video source modeling and transmission.



Stefanos D. Kollias (S'84–M'85) was born in Athens, Greece, in 1956. He received the Diploma degree in electrical engineering from the National Technical University of Athens (NTUA) in 1979, the M.Sc degree in communication engineering from the University of Manchester (UMIST), Manchester, U.K., in 1980, and the Ph.D. degree in signal processing from the Computer Science Division of NTUA in 1984. In 1982, he received a ComSoc Scholarship from the IEEE Communications Society.

Since 1986, he has been with the NTUA, where he is currently a Professor. From 1987 to 1988, he was a Visiting Research Scientist in the Department of Electrical Engineering and the Center for Telecommunications Research, Columbia University, NY. His current research interests include image processing and analysis, neural networks, image and video coding, and multimedia systems, and he is the author of more than 140 articles in these areas.