Fuzzy Data Fusion For Multiple Cue Image And Video Segmentation

Spyros Ioannou, Yannis Avrithis, Giorgos Stamou and Stefanos Kollias

National Technical University of Athens
Department of Electrical and Computer Engineering
Image, Video and Multimedia Systems Laboratory
9 Iroon Polytechniou St., 157 73 Zographou, Athens, Greece
email: sivann,iavr@image.ntua.gr, gstam@softlab.ntua.gr, stefanos@cs.ntua.gr

Summary. Fusion of multiple cue image partitions is described as an indispensable tool towards the goal of automatic object-based image and video segmentation, interpretation and coding. Since these tasks involve human cognition and knowledge of image semantics, which are absent in most cases, fusion of all available cues is crucial for effective segmentation of generic video sequences. This chapter investigates fuzzy data fusion techniques which are capable of integrating the results of multiple cue segmentation and provide time consistent spatiotemporal image partitions corresponding to moving objects.

1 Introduction

Fusion of multiple cue segmentations has proved to be an indispensable tool towards the goal of automatic object-based video segmentation and coding, mainly in the framework of the MPEG-4 standard [20]. State of the art video analysis/coding systems such as the SESAME system and the European COST 211 Analysis Model [10] achieve content-based spatiotemporal segmentation of video sequences employing fusion of multiple cue image partitions. Individual cues usually include color, motion as well as motion-compensated partitions of previous video frames, mainly for the purpose of object tracking [1,9]. The fusion process itself is accomplished by generation of hierarchical image partition structures, construction of decision trees and heuristic rule-based partition processing. In many cases, current approaches mainly focus on video coding, usually through rate-distortion criteria [17]. Other approaches perform simultaneous fusion and segmentation, making it hard to extend the fusion process to multiple cues [16]. Although such systems offer robust distinction between objects with different motion patterns, such as moving and stationary objects, many limitations exist.

First, it has become clear that additional cues, including depth, texture and shape apart from color and motion, are generally necessary for accurate object detection. Each additional cue can offer enhanced segmentation performance in specific cases where other cues fail. Moreover, each segmentation is given as an image partition, or, in a more general approach, is represented by

a set of image partitions at different levels of detail, forming a partition tree derived from a multiscale coarse-to-fine strategy [17]. Unified handling and processing of more complex data representations becomes difficult, especially in the presence of uncertainty.

Moreover, the fusion process should interact with object identification, categorization and recognition; cognitive vision systems require use of a priori knowledge about specific objects for robust segmentation fusion [4]. Strict associations between image partitions make existing systems susceptible to noise, hence fuzziness should be introduced in all levels of partition representation. Adaptation mechanisms are needed for adjustment of the fusion modules behavior in real life dynamic environments. Integration of large cue sets, a priori knowledge, fuzzy representation, intelligence and adaptation mechanisms into fusion algorithms will allow their robust operation in generic cognitive vision systems and in real-life conditions [18].

The above issues give rise to fuzzy rule-based fusion systems [15] that are able to handle complex partition tree representations and accommodate for integration of numerous cues including intensity, color, texture, motion and depth. Fuzzy rule based fusion techniques with built-in complex knowledge representation will enable robust image / video segmentation in contrast to conventional fusion techniques based on rate-distortion criteria or elementary heuristic rules. They may also permit reliable multiple object tracking, effectively dealing with occlusion / disocclusion, appearance variation, articulated motion and distraction problems.

In this chapter, we focus on integration of the results from multiple cue segmentation and on derivation of fuzzy decisions for handling inconsistencies and improving segmentation accuracy. The proposed approach takes into account a priori knowledge related to the expected characteristics of the different cue based segments. For instance, motion information is related to moving parts of objects, depth information defines the main object planes/views, but rather imprecisely, and color information provides precise but oversegmented objects. The target of our approach is to fuse individual single cue partitions, group image regions into objects and derive a representation of candidate objects by projecting and combining the generated cue segments.

In more detail, the generic properties of each candidate cue segmentation is given below, illustrating that segment partitions generated by single cues cannot be directly exploited for semantic object extraction:

- color / intensity segmentation, derived on the basis of spatial homogeneity criteria, allows very accurate definition of region boundaries, but fails to identify objects composed of regions with varying color characteristics
- texture segmentation permits extraction of areas with specific color / intensity patterns but still generates a large number of segments for each object

- motion segmentation, usually based on parametric motion models, produces a more limited number of regions, but produces coarse region boundaries due to matching errors and occlusions, while it segments objects with articulated motion
- depth segmentation, either though stereoscopic analysis or relative motion / occlusion between neighboring regions, achieves reliable approximation of real objects; however it fails to detect distant, stationary objects while depth regions have inaccurate boundaries too
- motion-compensated partitions of previous video frames provide a clear clue of the expected object positions and are necessary for temporal object tracking and detection of newly exposed objects, but need refinement using partitions derived from other cues

In general, image partitions obtained from color, texture, motion and depth segmentation form a hierarchical tree structure, in the sense that a depth region is composed by one or more motion region, motion regions contain several texture regions and so on. Fusion can then be accomplished by means of projection and hierarchical grouping of low-level partitions based on high-level ones. However, this is not always the case, since an object might be decomposed into independent non-overlapping partitions, according to color or motion. For this reason the proposed fusion approach performs intelligent region grouping based on a priori knowledge.

Specifically, based on existing efforts on fusion of color and motion segmentations, an initial set of heuristic rules is constructed, for instance:

1. projection of a color region onto motion regions identifies its degree of membership in each motion region
2. grouping of color regions that belong to the same motion region provides a single region with accurate boundaries
3. projection of such regions onto motion-compensated regions of previous frames permits object matching and tracking
4. discrepancies between motion regions and motion-compensated regions of previous frames identify appearance of new objects or object splitting due to articulated motion.

This set is enriched by additional rules to accommodate for depth segmentation. The system presented in this chapter focuses on color, motion, depth and motion-compensated segments of previous frames; however, the formulation of the proposed fusion approach permits integration of an arbitrary number of cues in a uniform and consistent way. Artificial intelligence techniques, and in particular fuzzy rule based systems are employed for the implementation of the fusion process [14]. Adaptation is possible, allowing addition, removal and updating of rules. Experiments on natural video sequences are presented in this chapter to illustrate the performance of the proposed technique. In particular, the output of the intelligent segmentation

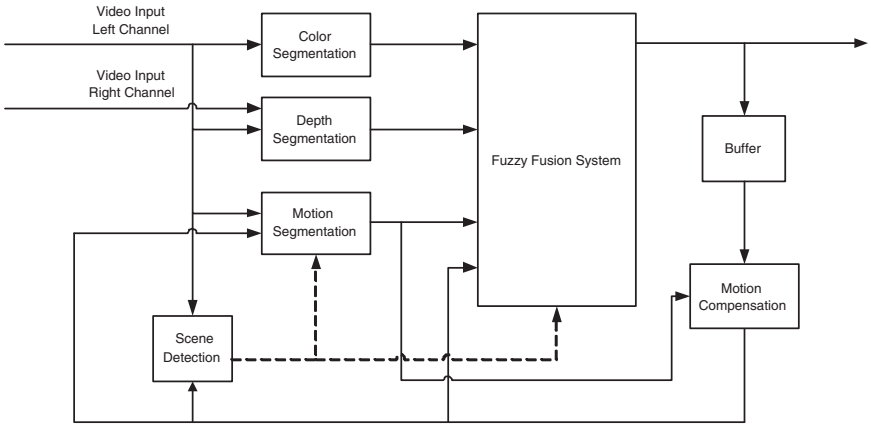


Fig. 1. General System Architecture

fusion and grouping procedure is shown to be of better quality than the one provided by each separate segmentation process.

The remaining of this chapter is organized as follows. In Section 2 the proposed integrated system architecture is described and details on the extraction of color, motion and depth segments are given. Section 3 deals with the formulation of the fusion process and provides a detailed description of the proposed fuzzy segmentation fusion approach. In Section 4, experimental results on natural video sequences are presented and compared to single cue segmentation approaches. Finally, conclusions are drawn and a brief description of further work is given in Section 5.

2 The proposed system architecture

In order to take advantage of depth segmentation in the fusion process, depth estimation is performed on stereoscopic video sequences consisting of two channels. Each stereoscopic sequence is first separated into two sequences, one for the left and one for the right channel. The left-channel sequence is used by the color, depth and motion segmentation modules. The right-channel sequence is only used by the depth segmentation module for the creation of the disparity map. For each frame, when all the modules produce their final segmentation, the segmentation maps are passed to the fuzzy fusion system which produces the final segmentation map. This final segmentation is fed back to the system to be used by the scene detection [11] and the motion estimation / compensation [12] modules. All segmentation modules use the M-RSST algorithm to create the image partitions. The block diagram of the proposed system along with its modules is shown in Fig. 1. A description of the M-RSST algorithm and its application to each module follows.

2.1 The M-RSST Algorithm

The Recursive Shortest Spanning Tree (RSST) algorithm [5] is our basis for cue segmentation of each frame in a given video shot. Despite its relative computational complexity, it is considered as one of the most powerful tools for image segmentation, compared to other techniques (including cue clustering, pyramidal region growing and morphological watershed). Initially an image I of size $M \times N$ pixels, is partitioned into $M \times N$ regions (segments) of size 1 pixel and links are generated for all 4-connected region pairs. Each link is assigned a weight equal to the distance between the two respective regions, which for example in the case of color segmentation could be defined as the Euclidean distance between the average color components of the two regions, using a bias for merging small regions. All link weights are then sorted in ascending order, so that the least weighed link corresponds to the two closest regions. The iteration phase of the RSST is then initiated, where neighboring regions are recursively merged by applying the following actions in each iteration:

1. The two closest regions are merged and the new region cue components and size are calculated.
2. The new region link weights from all neighboring regions are recalculated and sorted.
3. Any duplicated links are removed.

The iteration terminates when either the total number of regions or the minimum link weight (distance) reaches a target value (threshold). A distance threshold is in general preferable since it provides a result that is independent of the image content. The execution time of the RSST is heavily dependent upon the choice of the sorting algorithm, which is certainly a bottleneck of the algorithm. For this reason, a new approach has been proposed, the Multiresolution RSST [6] which recursively applies the RSST algorithm on images of increasing resolution. Initially a multiresolution decomposition of image I is performed with a lowest resolution level of L_0 so that a hierarchy of frames $I(0) = I, I(1), \ldots, I(L_0)$ is constructed, forming a truncated image pyramid, with each layer having a quarter of the pixels of the layer below.

The RSST initialization takes place for the lowest resolution image $I(L_0)$ and then an iteration begins, involving the following steps:

1. Regions are recursively merged using the RSST iteration phase.
2. Each boundary pixel of all resulting regions is split into four new regions, whose cue components are obtained from the image of the next higher resolution level.
3. The new link weights are calculated and sorted.

This 'split-merge' procedure is repeated until the highest resolution image $I(0)$ is reached. It is shown in [6] that effectively only the segment contour

shapes are affected at each iteration, since no segments are created or destroyed, hence the multiresolution segmentation approach yields much faster execution compared to RSST. Although its computational complexity is not straightforward to calculate, since it depends on the number, shape and size of segments, it is shown by experiments that it is at least 400 times faster than RSST for a typical image size of 768x576 pixels.

2.2 Color Segmentation

The M-RSST algorithm is directly applied on the left-channel image for color segmentation. Using the RGB color space, a distance measure between two adjacent regions (segments) S_1 and S_2 is defined as

$$d^C(S_1,S_2) = \left[(R_{S_1}-R_{S_2})^2 + (G_{S_1}-G_{S_2})^2 + (B_{S_1}-B_{S_2})^2\right]^{1/2} \frac{A_{S_1}A_{S_2}}{A_{S_1}+A_{S_2}} \tag{1}$$

where R_{S_i}, G_{S_i} and B_{S_i} respectively represent the average R, G and B values of all pixels inside region S_i and A_{S_i} is the number of pixels within this region.

2.3 Motion Segmentation

In order to solve the motion segmentation problem, numerous different techniques can be employed, such as direct intensity based methods, optical flow based methods, or simultaneous motion estimation and segmentation methods [3,21]. Each method has its own advantages and disadvantages, restricting its use to specific applications. For example, simultaneous motion estimation and segmentation methods are unattractive due to their high computational complexity, while direct intensity based methods cannot handle camera noise and illumination changes. Optical flow methods are quite popular and widely used both for video coding and image analysis and understanding. In this system a block matching algorithm is used to create the motion field. In low-texture areas, the produced motion field can be noisy, thus, a post-processing step for motion field smoothing is indispensable. Median filtering is selected for this purpose due to its speed and its ability to preserve object contours.

Motion field segmentation is performed by dividing each frame into regions of homogeneous motion. The M-RSST algorithm described above is now applied on the motion field with the following distance:

$$d^M(S_1,S_2) = \left[(X_{S_1}-X_{S_2})^2 + (Y_{S_1}-Y_{S_2})^2\right]^{1/2} \frac{A_{S_1}A_{S_2}}{A_{S_1}+A_{S_2}} \tag{2}$$

where X_{S_i}, Y_{S_i} represent the average x and y coordinates of the motion vectors on segment S_i.

2.4 Depth Segmentation

The use of three-dimensional (3-D) video, obtained by stereoscopic or multi-view camera systems provides very efficient visual representation. The problem of content-based segmentation is addressed more precisely since video objects are usually composed of regions belonging to the same depth plane. Therefore, 3-D video enables efficient handling and manipulation of video objects by exploiting depth information provided by stereoscopic image analysis [2,7].

Stereoscopic video sequences consist of two channel images obtained by two cameras and provide the perspective projection of 3-D points onto the two 2-D image planes of the cameras. In the framework of this chapter, the technique proposed in [8] is employed to generate a disparity map from the two images. As described in [19], a depth map is then obtained from the disparity map, occluded areas are detected and compensated for, and finally depth segmentation is performed by applying the M-RSST algorithm on the final depth map. The following distance is used for depth segmentation:

$$d^D(S_1, S_2) = \left| D_{S_1} - D_{S_2} \right| \frac{A_{S_1} A_{S_2}}{A_{S_1} + A_{S_2}} \tag{3}$$

where D_{S_i} represents the average depth of segment S_i.

3 Fuzzy Segmentation Fusion

We have now extracted the color, motion and depth segments, using the M-RSST algorithm. The next step is to fuse the color and motion segments as these segments identify the object boundaries very accurately. Additionally depth segments can provide us with a coarse separation of the semantic objects because usually objects consist of segments that are located at about the same depth. So by using depth segments as a constraining mask we can project color and motion segments onto this mask. Then for every color segment we find in which depth it belongs and finally we fuse the color segments of the same depth into one new segment.

The Fuzzy Segmentation Fusion (FSF) takes as input the color, motion, depth and motion-compensated previous segmentations and gives to the output the final fused segmentation.

Let us first provide the reader with the formal mathematical description of our work. We denote with

$$S^C = \{s_1^C, s_2^C, \cdots, s_N^C\}$$
$$S^M = \{s_1^M, s_2^M, \cdots, s_K^M\}$$
$$S^D = \{s_1^D, s_2^D, \cdots, s_L^D\}$$
$$S^P = \{s_1^P, s_2^P, \cdots, s_Q^P\}$$

the color, motion, depth and motion-compensated previous segmentations (the inputs of the FSF module), while

$$S^F = \{s_1^F, s_2^F, \cdots, s_U^F\}$$

stands for the fused segmentation (the output of the FSF module). For simplicity reasons, we did not involve the time in the above notation. Thus, we assume that time t is implied, unless something else is clearly stated. The above mentioned segmentations are actually *crisp* image partitions.

The FSF module tries to take advantage of the information provided by S^C, S^M, S^D and S^P in order to improve the precision and the semantic framework of the segmentation. Since the color semgnetation provides the most precise boundaries, it is reasonable to use S^C as the basis on which S^F will be constructed. Furthermore, we assume that S^C provides an over-segmented image partition. Thus, FSF actually produces a more "unified" partition S^F in comparison with S^C. This means that:

$$\forall i \in \mathbb{N}_N \quad \text{and} \quad j \in \mathbb{N}_U \quad \text{we have} \quad s_i^C \subseteq s_j^F$$

where

$$N_a = 1, 2, \cdots, a.$$

In order to compute S^F out of S^C, S^M, S^D and S^P we use the M-RSST algorithm. Roughly speaking, we take all the possible color segment pairs provided by S^C and compute the distances between them. Then, the two closest segments (smallest distance) are united (if their distance is less than a threshold) and the whole process is repeated until no segments are united. Obviously, the intelligence of the method lies on the way in which the above distances are computed. Let us now describe this process.

We first introduce some useful operators. For each pair of segments s_1, s_2 the *projection operator* $R(s1, s2)$ is defined by:

$$R(s1, s2) = \frac{|s1 \cap s2|}{|s1|} \tag{4}$$

where $|\cdot|$ denotes the cardinality of the set (in this case the number of the pixels of the segment).

Given a segment s and a set of segments S, the *θ-neighborhood operator* $I_\theta(s, S)$ is defined by:

$$I_\theta(s, S) = \{i \in N_{|S|} : |R(s, si) - \max_{j \in \mathbb{N}_{|S|}} R(s, si)| \leq \theta\}$$

where *theta* is a threshold and $R(\cdot, \cdot)$ the projection operator defined by 4.

The *cue-based distance* of two segments is defined (for any cue like motion, depth, motion-compensated previous) with the aid of the θ-neighborhood operator. For example, the *motion-based distance* $D^M(s_i^C, s_j^C)$ of two color segments $s_i^C, s_j^C, (i, j \in \mathbb{N}_N)$ is defined by:

$$D^M(s_i^C, s_j^C) = a \cdot b \cdot \delta_1 - a \cdot b \cdot \delta_2$$

where

$$\begin{aligned}
a &= \max_{k \in \mathbb{N}_K} R(s_i^C, s_k^M) \\
b &= \max_{k \in \mathbb{N}_K} R(s_j^C, s_k^M) \\
\delta_1 &= \delta(|I_\theta(s_i^C, S^M) \cap I_\theta(s_j^C, S^M)|) \\
\delta_2 &= \delta\Big(\big|[I_\theta(s_i^C, S^M) \cup I_\theta(s_j^C, S^M)] - [I_\theta(s_i^C, S^M) \cap I_\theta(s_j^C, S^M)]\big|\Big) \\
\delta &= \begin{cases} 1, \text{if } x \neq 0 \\ 0, \text{if } x = 0 \end{cases}
\end{aligned}$$

The distance $D(s_i^C, s_j^C)$ between two color segments s_i^C and s_j^C, used in the M-RSST segmentation process of the FSF module, is derived from the

$$\begin{aligned}
&\text{motion-based } (D^M(s_i^C, s_j^C)) \\
&\text{depth-based } (D^D(s_i^C, s_j^C)) \\
&\text{and previous-based } (D^P(s_i^C, s_j^C))
\end{aligned}$$

distances computed with the aid of the above process. This is actually the intelligent part of the FSF module that provides the system with the ability to fuse the various cues using a priori knowledge in the form of fuzzy linguistic rules [13]. Figure 2 shows the structure of the fuzzy inference system that implements the above idea. The system takes three inputs: the motion-based distance, the depth-based distance and the previous-based distance. Obviously, they are fuzzy values (between -1 and 1) representing the degree in which the specific cue "advises" the fusing fuzzy inference system to provide or not a high value of "unity" between the two color segments (its output).

The operation of the fusing fuzzy inference system is based on a set of linguistic rules provided by experts. The rules map the input linguistic variable to the output one, i.e. the fuzzy partitions defined on

$$D^M(s_i^C, s_j^C), D^D(s_i^C, s_j^C) \quad \text{and} \quad D^P(s_i^C, s_j^C)$$

to the fuzzy partition defined on $D(s_i^C, s_j^C)$ (Figure 3). Table 1 summarises this set of rules. Finally, in Figure 4 the surface view of the system input-output mapping is shown.

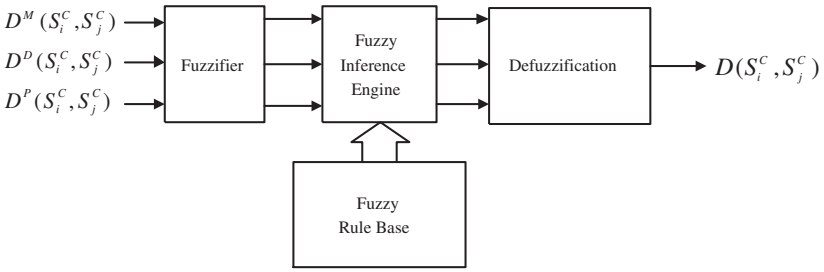


Fig. 2. Fusing Fuzzy Inference System

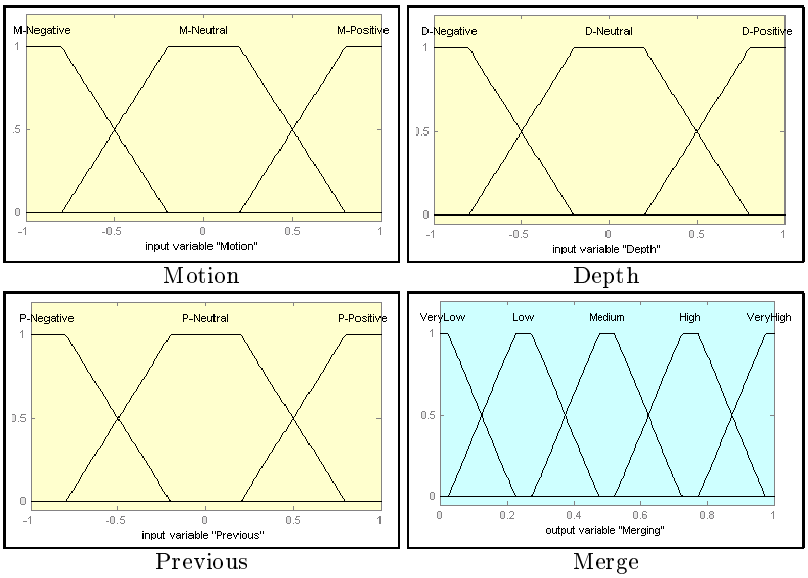


Fig. 3. Definition of the fuzzy partitions

4 Simulation Results

4.1 Equipment

We tested our algorithm in a real 3-D video sequence. This sequence was obtained with the aid of the "SX2000 Stereo-Optix" lens from NuView connected to a normal MiniDV video camera. The SX2000 lens projects two images on the camera's focus with the aid of a mirror; those two images are not recorded simultaneously but sequentially by taking advantage of the interlaced nature of the CCD video signal. Therefore the even lines of the resultant video sequence represent the left view and the odd lines represent the

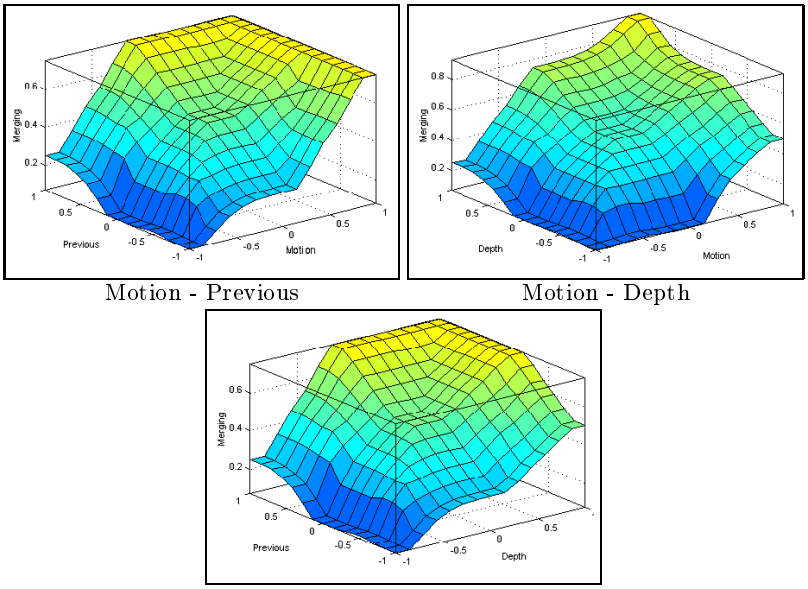


Fig. 4. Schematic representation of the input output associations

right view. The focal distance can be adjusted through mechanical means. Apparently this method diminishes the vertical resolution by half (720×288) but it is still satisfactory for our use.

The obtained stereo video sequence is first analyzed and for each pair of stereo frames a depth map is estimated. The application of the M-RSST algorithm on the depth map results on the depth segmentation. The color and motion segmentation cues are then constructed from the left channel image.

4.2 Color

Since the fusion process merges color segments we want those segments to have precise boundaries. Thus, for the color segmentation we use a level 1 resolution (of size 2×2 pixels) on the M-RSST. Since we also want an over-segmented image, we choose a very low distance threshold for the region merging process. For this experiment we used a distance threshold of 0.05% in the RGB space, but any low threshold would produce similar results. An enlarged segmentation result is illustrated in Fig. 5 where the segments are clearly shown. An interesting feature of the RSST algorithm is that it can stop at a predifined number of segments and this would be an alternative way of achieving an oversegmented color image.

If Motion is	and Depth is	and Previous is	then Merging is
M-Negative	D-Negative	P-none	VeryLow
M-Negative	D-Neutral	P-Negative	VeryLow
M-Negative	D-Neutral	P-Neutral	VeryLow
M-Negative	D-Neutral	P-Positive	Low
M-Negative	D-Positive	P-Negative	VeryLow
M-Negative	D-Positive	P-Neutral	Low
M-Negative	D-Positive	P-Positive	Medium
M-Neutral	D-Negative	P-Negative	VeryLow
M-Neutral	D-Negative	P-Neutral	VeryLow
M-Neutral	D-Negative	P-Positive	Low
M-Neutral	D-Neutral	P-Negative	Low
M-Neutral	D-Neutral	P-Neutral	M-Positive
M-Neutral	D-Neutral	P-Positive	High
M-Neutral	D-Positive	P-Negative	Medium
M-Neutral	D-Positive	P-Neutral	High
M-Neutral	D-Positive	P-Positive	High
M-Positive	D-Negative	P-Negative	Low
M-Positive	D-Negative	P-Neutral	Medium
M-Positive	D-Negative	P-Positive	Medium
M-Positive	D-Neutral	P-none	High
M-Positive	D-Positive	P-none	VeryHigh

Table 1. The set of linguistic rules of the fusing fuzzy inference system

4.3 Motion

A block-matching algorithm is used to construct the motion field. The search area of the block matching in our test was set at 10 pixels in both directions and the block size is 3×3 pixels. Despite the small blocksize, the motion field is quite accurate. During the block matching process for the motion field creation, if a block's minimum distance from the test block is less than 20% greater than the distance of the respective zero-motion block, then we consider the corresponding motion vector as zero. This step reduces the motion noise in low texture areas. A median filter is then applied at the motion field before the segmentation process. For the motion segmentation we apply the M-RSST with a highest resolution level of $L_0 = 3$ (block resolution of 8×8 pixels) to speed up the segmentation process.

4.4 Depth

A disparity map is estimated from the left and right channel images. A depth map is then obtained from the disparity map, occluded areas are detected and compensated for, and finally depth segmentation is performed by applying the M-RSST algorithm on the final depth map. The search area for the disparity

estimation is 10 pixels with a block size of 3×3 pixels. The depth segmentation is created with a block resolution of 4×4 pixels.

4.5 Fusion

The fusion subsystem is then started with the color, motion and depth segmentation maps as inputs along with the cue fusion result of the previous frame. In the case of the first frame where no previous fusion results exist, a blank map is used instead. The fusion subsystem continues with the color segment merging by considering now their fuzzy distance.

4.6 Results

An example of the fusion process is illustrated in figure 5, where the color, motion and depth segmentation results are shown along with the fusion result of the previous frame. More examples are given in figures 6–17 for two video sequences. On the first one the camera is still and the person is moving from left to right with his right hand extended. A 6-frame sample of this sequence is shown in figures 6–11. In this sample, the color segmentation produces about 140-150 segments; the motion segmentation produces two segments, one for the background and one for the moving person; the depth segmentation produces three to five segments: one for the background, one for the person's body, one for his hand and additionally some false segments especially in areas with low energy in the high horizontal frequencies.

It can be seen that the person's boundaries in the fused result are more precise than the boundaries of the cue sources. This is not due to the higher resolution of the color segmentation but due to the fusion of color and depth segmentations; a depth segmentation at higher resolution would not have the accuracy of the color segmentaton. The person on the fused image has been seperated succesfully from the background; his hand, having a different depth from his body, is split in a different segment. At frame 59 in figure 10 the hand is separated in two segments due to the vertical expansion of the hand segment of the depth channel.

On the second sequence, both the camera and the person are still but the person is waving his hand. A 6-frame sample is shown in figures 12–17. As it can be seen, up to two motion vectors are created on the palm and arm areas which when combined with the depth and color channels result in the fused images. Note that the area below the the person's right arm (at the bottom left) is considered to be in a different segment than the rest of the background, because the corresponding color segments are not adjacent; they are separated by the arm's color segments. In figure 12 two motion segments have been detected due to the higher linear velocity of the palm. Because of this, the palm is a seperate segment in the fusion result (Fig. 12) and this separation propagates to the following segmentations. If this separation is

desirable, the M-RSST algorithm could be adjusted to be more sensitive and produce more segments. If the separation is not desirable, the use of a full parametric motion model would detect the circular motion and thus preserve the arm and palm in one segment.

Because of the inaccurate depth segmentation, one can observe an extra segment in figure 15 near the palm (in black color). This segment was not merged with the rest of the palm because of the previous and motion segmentations, and thus it was not propagated to the following frames.

5 Conclusions

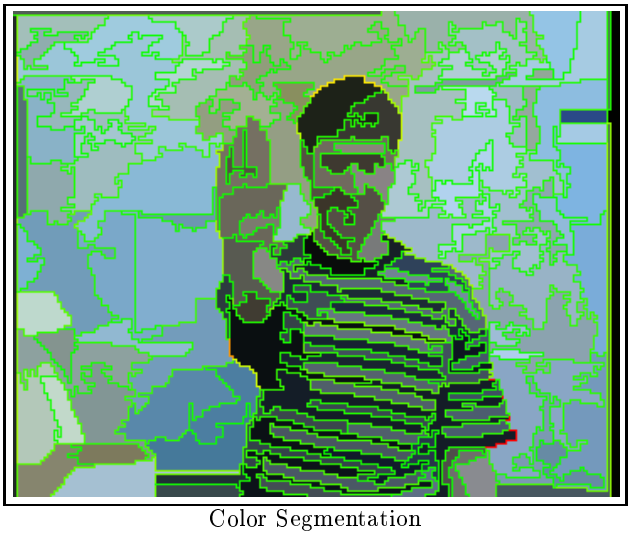
In this chapter, we examined the integration of multiple cue segmentations and the use of fuzzy decisions for handling inconsistencies and improving segmentation accuracy. The resulting segmentation is shown to be superior to the individual cue segmentations.

One way to improve the fusion performance is by augmenting its memory: more than one previous frames could be taken into account thus minimizing the error propagation. Kalman filters could be introduced in combination with the fused-result memory, to implement object tracking and further enhance segmentation reliability.

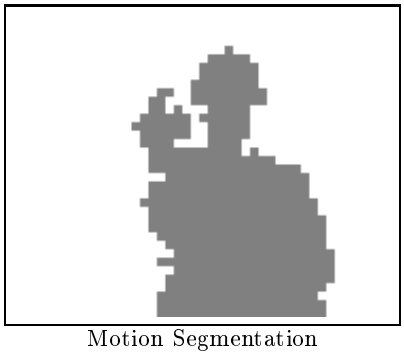
Fuzziness may also be introduced in the partitions themselves, so that the transition between neighboring regions may be gradual, as well as in the generation of the partition trees, by means of link weights expressing the strength of decomposition of a coarse region into finer ones. Fuzzy partitions can be defined on the partition trees, giving linguistic meaning to low level visual descriptors and allowing the numerical interpretation of inference rules. Moreover the fusion module parameters could be dynamically adapted with the aid of higher level object description information.

References

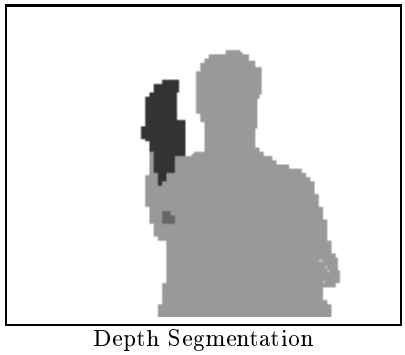
1. T. Meier and K. N. Ngan *Automatic Segmentation of Moving Objects for Video Object Plane Generation* IEEE Trans. Circuits and Systems for Video Technology, Vol. 8, No. 5 pp. 525-538 1998
2. M. Pardas and P.Salembier *3-D morphological segmentation and motion estimation for image segmentation* Signal Processing, vol. 38 pp. 31-43 Sept. 1994
3. M. M. Chang, A. M. Tekalp, and M. I. Sezan *Motion field segmentation using an adaptive MAP criterion* Proc. of IEEE Int. Conf. Acoust., Speech, Signal Processing ICASSP'93 Minneapolis, MN, Apr. 1993, vol. V pp.33-36
4. D. Geiger and A. Yuille *A common framework for image segmentation* Int. Journal Comput. Vision, vol 6 pp.227-243 1991
5. O. J. Morris, M. J. Lee and A. G. Constantinides *Graph Theory for Image Analysis: an Approach based on the Shortest Spanning Tree* IEE Proceedings, Vol. 133 pp.146-152 April 1986



Color Segmentation

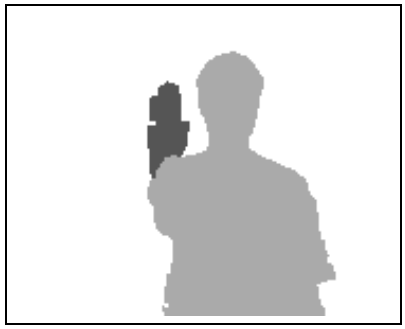


Motion Segmentation



Depth Segmentation

Previous Fusion



Fusion Result

Fig. 5. Sequence 2, Frame 60, Large format

6. 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 July 1999
7. R. C. Gonzalez and R. E. Woods *Digital Image Processing* Addison-Wesley 1992
8. D. Tzovaras, N. Grammalidis and M. G. Strintzis *Disparity Field and Depth Map Coding for Multiview 3D Image Generation* Image Communication, No. 11 pp. 205-230 1998
9. C. Gu and M.-C. Lee *Semiautomatic Segmentation and Tracking of Semantic Video Objects* IEEE Trans. on Circuits and Systems for Video Technology, Vol. 8, No. 5 pp. 572-584 Sept. 1998
10. A.A. Alatan, L. Onural, M. Wollborn, R. Mech, E. Tuncel, and T. Sikora *Image Sequence Analysis for Emereging Interactive Multimedia Services - The European COST 211 Framework* IEEE Trans. on Circuits and Systems for Video Technology, Vol. 8, No. 7 pp. 802-813 Nov. 1998
11. Chong-Wah Ngo, Ting-Chuen Pong, and R. T. Chin *Video Partitioning by Temporal Slice Coherency* IEEE Trans. on Circuits and Systems for Video Technology, Vol. 11, No. 8 pp. 941-953 Aug. 2001
12. A. Smolic, T. Sikora, and J.R. Ohm *Long-Term Global Motion Estimation and Its Application for Sprite Coding, Content Description, and Segmentation* IEEE Trans. on Circuits and Systems for Video Technology, Vol. 9, No. 8 pp. 1227-1242 Dec. 1999
13. G.B. Stamou and S.G. Tzafestas *Fuzzy Relation Equations and Fuzzy Inference Systems: An Inside Approach* IEEE Transactions on Systems, Man and Cybernetics - PART B: Cybernetics, Vol. 29, No. 6 Dec. 1999
14. G.J. Klir and B. Yuan *Fuzzy Sets and Fuzzy Logic: Theory and Applications* Englewood Cliffs, NY: Prentice Hall, 1995
15. L.A. Zadeh *Fuzzy logic=computing with words* IEEE Transactions on Fuzzy Systems pp. 103-111 1996
16. E. Saber and A. Tekalp and G. Bozdagi *Fusion of Color and Edge Information for Improved Segmentation and Edge Linking* Image and Vision Computing vol 15 pp. 769-780 1997
17. P.Salembier, F.Marques, M. Pardas, J.R.Morros, I. Corset, S.Jeannin, L.Bouchard, F.Meyer, and B.Marcotegui *Segmentation-Based Video Coding System Allowing the Manipulation of Objects* IEEE Trans. on Circuits and Systems for Video Technology, Vol. 7, No. 1 pp. 60-74 Feb. 1997
18. I. Koprinska, S. Carrato *Temporal video segmentation: A survey* Signal Pricessing: Image Communication 16 pp. 477-500 2001
19. 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
20. T. Sikora *The MPEG-4 Video Standard Verification Model* IEEE Trans. Circuits and Systems for Video Technology, Vol. 7, No. 1 pp. 19-31 Feb. 1997
21. A.M. Tekalp *Digital Video Processing* Prentice Hall 1995

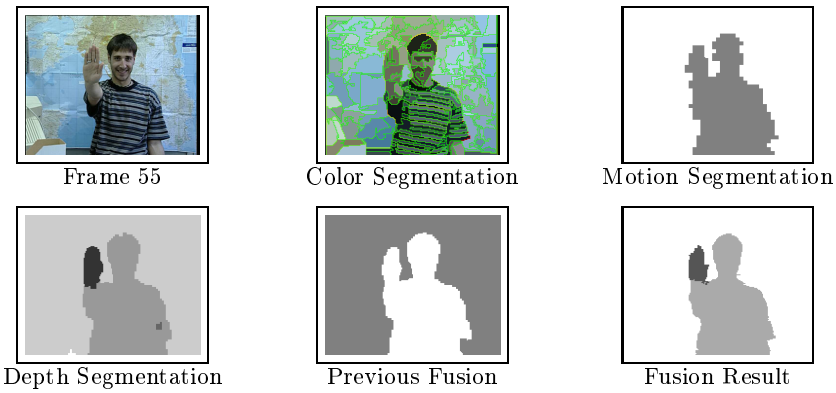


Fig. 6. Sequence 2, Frame 55

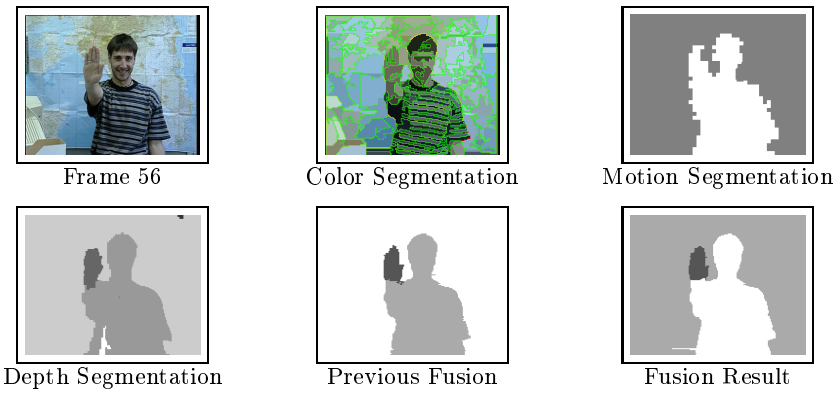


Fig. 7. Sequence 2, Frame 56

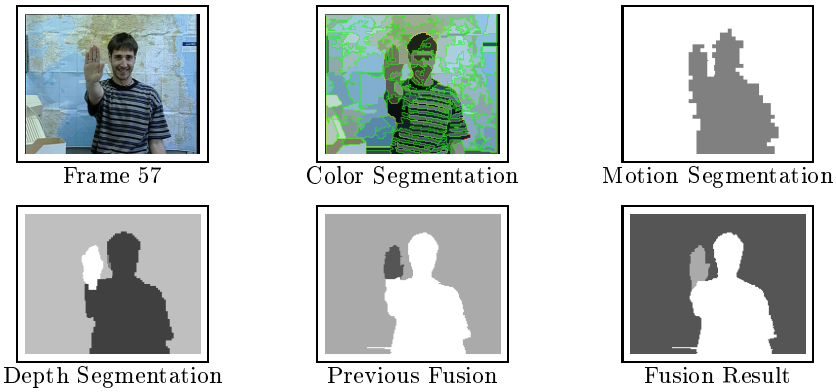


Fig. 8. Sequence 2, Frame 57

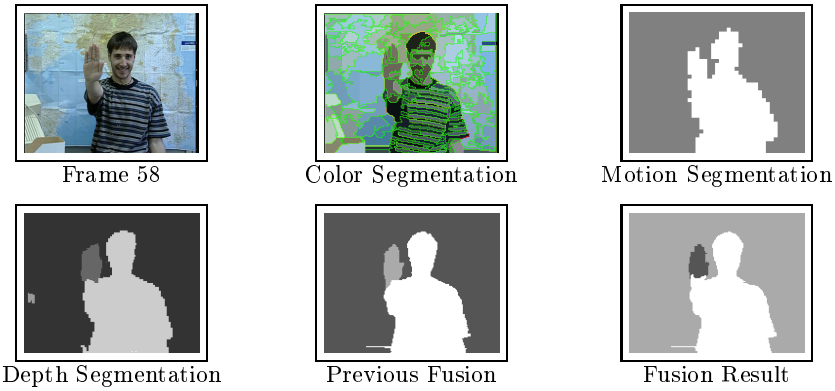


Fig. 9. Sequence 2, Frame 58

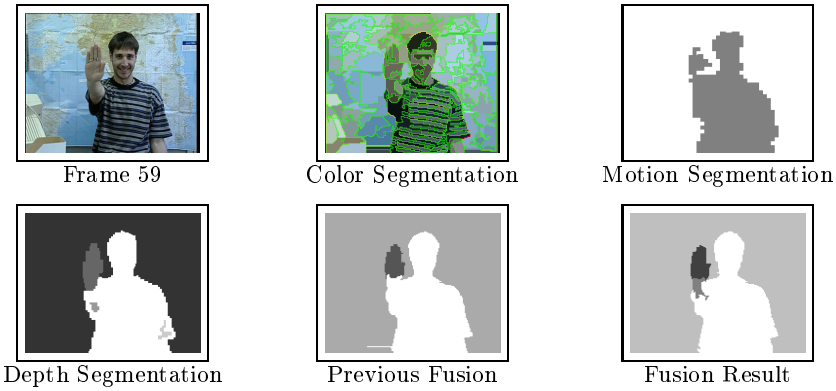


Fig. 10. Sequence 2, Frame 59

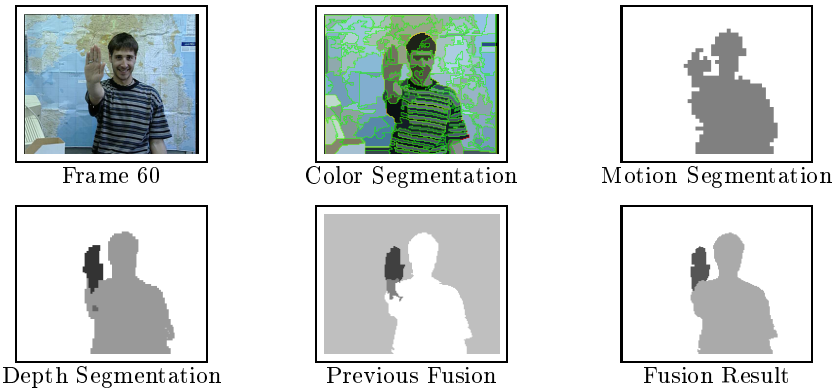


Fig. 11. Sequence 2, Frame 60

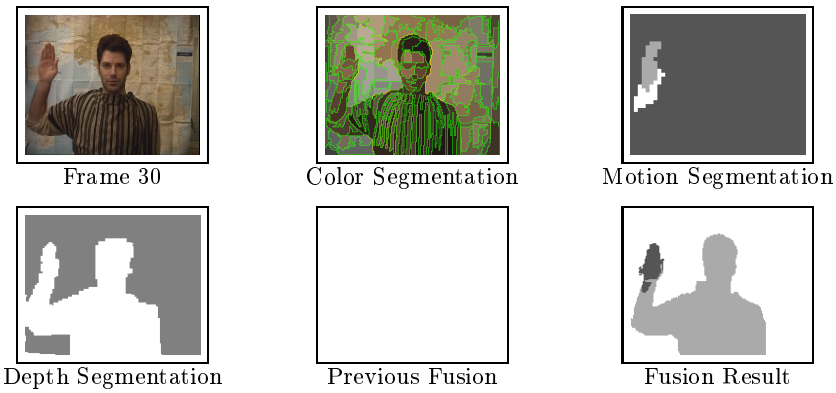


Fig. 12. Sequence 1, Frame 30

Fig. 13. Sequence 1, Frame 31

Fig. 14. Sequence 1, Frame 32

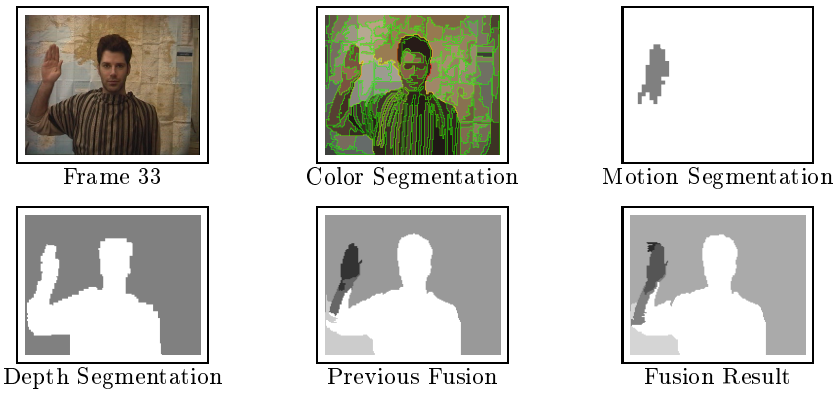


Fig. 15. Sequence 1, Frame 33

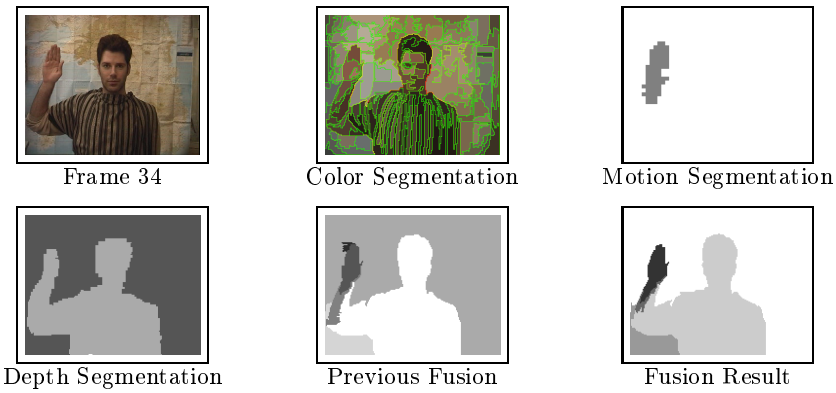


Fig. 16. Sequence 1, Frame 34

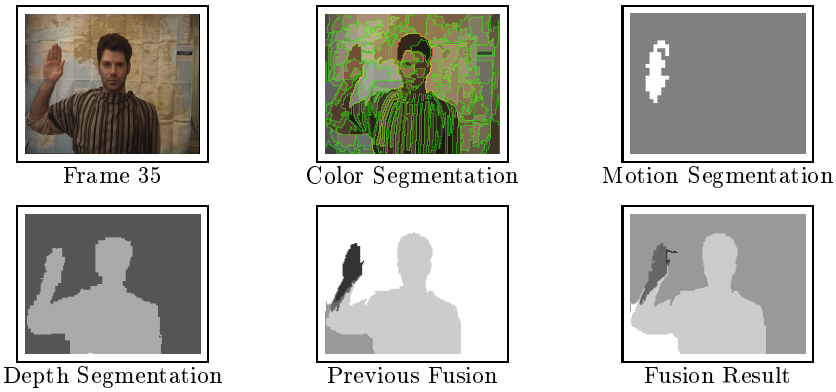


Fig. 17. Sequence 1, Frame 35