

# Fuzzy Segmentation of Color Video Shots

Bruno M. Carvalho, Lucas M. Oliveira, and Gilbran S. Andrade

Departamento de Informática e Matemática Aplicada,  
Universidade Federal do Rio Grande do Norte,  
59072-970, Natal, RN, Brazil

bruno\_m\_carvalho@yahoo.com, lmoliveira@gmail.com, gilbransilva@yahoo.com.br

**Abstract.** Fuzzy segmentation is a region growing technique that assigns a grade of membership to an object to each element in an image. In this paper we present a method for segmenting video shots by using a fast implementation of the fuzzy segmentation technique. The video shot is treated as a three-dimensional volume with different  $z$  slices being occupied by different frames of the video shot. The volume is interactively segmented based on selected seed elements, that will determine the affinity functions based on their intensity and color properties. Experiments with a synthetic video under different noise conditions are performed, as well as examples of two real video shot segmentations are presented, showing the applicability of our method.

## 1 Introduction

Digital image segmentation is the process of assigning labels to different objects in a digital image, where the level of detail indicated by the labeling is related to the application at hand. To perform object identification in digital or continuous, moving or still images, humans make use of different visual cues and high-level reasoning and knowledge. The difficulty of incorporating such type of reasoning into a computer program makes the task of segmenting out an object from its background a hard one. This task is even more challenging for a computer program when, instead of intensity values, what distinguishes the object from the background is some textural property, or when the image is corrupted by inhomogeneous illumination and/or noise.

Video digital segmentation consists of segmenting objects on sequences (shots) of images (frames). Conceptually, digital video segmentation is different from individually segmenting a sequence of digital images in the sense that there needs to be a consistency between the segmentations of a frame and of the frames that come before and after it on the segmented sequence. For example, the level of detail of the labels has to be consistent through out the segmented sequence.

Region growing algorithms segment an image by appending pixels to regions defined by seed pixels until all pixels in the image have been assigned to a set defined by one of the seed pixels, where the decision of to which object a pixel will be assigned is based on some predefined criteria. The selection of the seed pixels can be performed automatically, based on the nature of the problem. However, when no *a priori* information is available, these algorithms are semi-automatic, requiring the selection of the seed pixels by a user.

Fuzzy segmentation [1, 2] is a region growing method that has been successfully used for segmenting images corrupted by inhomogeneous illumination and/or noise [2]. In this paper we propose a method for segmenting color video shots based on the fuzzy segmentation method of [2, 3], that calculates simultaneously the fuzzy segmentation of multiple objects. The proposed method considers the frames of the video sequence as  $z$  slices of a 3D volume and uses the fast implementation of the algorithm presented in [3]. The speed of the segmentation allows an user to add and/or delete seed pixels, rerun the algorithm a few times, and still get the final segmentation in a reasonable amount of time.

The reason why we consider the video sequence as a 3D volume is that we want to stylize objects in pre-acquired video shots using Non-Photorealistic Rendering (NPR) techniques. This work was performed as a part of a project for providing computational tools to non-experienced users for generating animations using NPR techniques. NPR techniques aim to reproduce artistic techniques renderings, trying to express feelings and moods on the rendered scenes. Another way of defining NPR is that it is the processing of images or videos into artwork, generating images or videos that can have the visual appeal of pieces of art, expressing the visual and emotional characteristics of artistic styles such as impressionism and watercolor painting.

NPR techniques can be applied to still images, to 3D models or to video sequences, a task also called video stylization. If the input for the NPR video is a normal video, not maintaining temporal coherence of elements of the stylization, such as brush strokes, incurs in severe flickering on the output video [4]. This flickering comes not only from changed objects being rendered with elements that follow the object movement but also from static areas being rendered differently each time. Because of this problem, some authors have used optical flow techniques for enforcing temporal coherence [4,5]. However, the local characteristic of the optical flow techniques and their sensitivity to noisy images somehow limit their applicability. To overcome those problems, segmentation algorithms have been applied to video shot segmentation to produce end-to-end segmentations that are later used to enforce temporal coherence [6,7]. Here we propose the use of a fast implementation of fuzzy segmentation for segmenting color video shots as 3D volumes interactively.

## 2 Fuzzy Segmentation

Fuzzy segmentation is segmentation technique that computes the fuzzy connectedness, a concept introduced in [8], for every pixel in an image. The original fuzzy segmentation algorithm, introduced in [1], was generalized in [2], where the technique deals with an arbitrary finite set  $V$ , composed of *spels* (short for *spatial elements*). These spels can represent many different things, such as pixels of an image (as in [1, 9]), voxels placed on a simple cubic grid (as in [3]) or on a face-centered cubic grid (as in [10]), dots in the plane (as in [11]) or feature vectors (as in [12]). The theory and the algorithms discussed in [2, 3] are independent of the specifics of the application area, and so, can be applied to data clustering [13] in general.

The objective of the fuzzy segmentation algorithms of [2, 3] is to produce a partition of the set  $V$  into a specified number of objects, but in a fuzzy way; i.e., in addition to assigning an object label for every spel, the algorithm also assigns a grade of

membership for that object. A grade of membership is a number between 0 and 1, where 0 indicates that the spel definitely does not belong to the object, and 1 indicates that it definitely does). The formalization of such fuzzy partitioning is achieved by the concept of an  $M$ -semisegmentation (where  $M$  is the number of objects), defined below.

An  $M$ -semisegmentation of  $V$  is a function  $\sigma$  that maps each  $c \in V$  into an  $(M + 1)$ -dimensional vector  $\sigma^c = (\sigma_0^c, \sigma_1^c, \dots, \sigma_M^c)$ , such that

1.  $\sigma_0^c \in [0, 1]$  (i.e.,  $\sigma_0$  is nonnegative but not greater than 1),
2. for each  $m$  ( $1 \leq m \leq M$ ), the value of  $\sigma_m^c$  is either 0 or  $\sigma_0^c$ , and
3. for at least one  $m$  ( $1 \leq m \leq M$ ),  $\sigma_m^c = \sigma_0^c$ .

In the definitions above,  $\sigma_m^c$  represents the grade of membership of the spel  $c$  in the  $m$ th object, and  $\sigma_0^c$  is always  $\max_{1 \leq m \leq M} \sigma_m^c$ . It is easy to see that this definition of  $M$ -semisegmentation allows the case where a spel belongs to more than one object, as long as it has the same grade of membership in all of them. An  $M$ -segmentation of  $V$  is an  $M$ -semisegmentation where  $\sigma_0^c > 0$  for all spels  $c \in V$ .

In this context, a *chain* is defined as a sequence  $\langle c^{(0)}, \dots, c^{(K)} \rangle$  of distinct spels, and its *links* are the ordered pairs  $(c^{(k-1)}, c^{(k)})$  of consecutive spels in the sequence. The strength of a link  $(c, d)$ , or  $\psi$ -strength of a link, is also a fuzzy concept, with a real value between 0 and 1 being assigned to it by an appropriate *fuzzy spel affinity* function  $\psi : V^2 \rightarrow [0, 1]$ . The  $\psi$ -strength of a chain is defined as the  $\psi$ -strength of its weakest link if the chain has two or more spels on it, and 1 if the chain has only one spel in it. A set  $U (\subseteq V)$  is said to be  $\psi$ -connected if, for every pair of spels in  $U$ , there is a chain in  $U$  of positive  $\psi$ -strength from the first spel of the pair to the second. If one wants to segment multiple objects, it is reasonable to define different fuzzy spel affinities for each one of them.

A  $M$ -semisegmentation in our theory is determined by an  $M$ -fuzzy graph, that is a pair  $(V, \Psi)$ , where  $V$  is a nonempty finite set and  $\Psi = (\psi_1, \dots, \psi_M)$  with  $\psi_m$  (for  $1 \leq m \leq M$ ) being a fuzzy spel affinity. A *seeded  $M$ -fuzzy graph* is a triple  $(V, \Psi, \mathcal{V})$  such that  $(V, \Psi)$  is an  $M$ -fuzzy graph and  $\mathcal{V} = (V_1, \dots, V_M)$ , where  $V_m \subseteq V$  for  $1 \leq m \leq M$ . We say that a seeded  $M$ -fuzzy graph  $(V, (\psi_1, \dots, \psi_M), (V_1, \dots, V_M))$  is *connectable* if

1. the set  $V$  is  $\phi_\Psi$ -connected, where  $\phi_\Psi(c, d) = \min_{1 \leq m \leq M} \psi_m(c, d)$  for all  $c, d \in V$ , and
2.  $V_m \neq \emptyset$ , for at least one  $m$ ,  $1 \leq m \leq M$ .

For an  $M$ -semisegmentation  $\sigma$  of  $V$  and for  $1 \leq m \leq M$ , we define the chain  $\langle c^{(0)}, \dots, c^{(K)} \rangle$  to be a  $\sigma m$ -chain if  $\sigma_m^{c^{(k)}} > 0$ , for  $0 \leq k \leq K$ . Furthermore, for  $W \subseteq V$  and  $c \in V$ , we use  $\mu_{\sigma, m, W}(c)$  to denote the maximal  $\psi_m$ -strength of a  $\sigma m$ -chain from a spel in  $W$  to  $c$ . (This is 0 if there is no such chain.)

**Theorem 1.** *If  $(V, \Psi, \mathcal{V})$  is a seeded  $M$ -fuzzy graph (where  $\Psi = (\psi_1, \dots, \psi_M)$  and  $\mathcal{V} = (V_1, \dots, V_M)$ ), then*

- (i) *there exists an  $M$ -semisegmentation  $\sigma$  of  $V$  with the following property: for every  $c \in V$ , if for  $1 \leq n \leq M$*

$$s_n^c = \begin{cases} 1, & \text{if } c \in V_n, \\ \max_{d \in V} (\min(\mu_{\sigma, n, V_n}(d), \psi_n(d, c))), & \text{otherwise,} \end{cases} \quad (1)$$

*then for  $1 \leq m \leq M$*

$$\sigma_m^c = \begin{cases} s_m^c, & \text{if } s_m^c \geq s_n^c, \text{ for } 1 \leq n \leq M, \\ 0, & \text{otherwise;} \end{cases} \quad (2)$$

- (ii) this  $M$ -semisegmentation is unique; and  
 (iii) it is an  $M$ -segmentation, provided that  $(V, \Psi, \mathcal{V})$  is connectable.

What Theorem 1 says in general terms, is that the  $m$ th object of an  $M$ -semisegmentation can “claim” a spel  $c$  as part of it if, and only if,  $s_m^c$  is maximal. We can see from the definition of an  $M$ -semisegmentation in shown previously that  $\sigma_m^c$  has a positive value only for such objects. The calculation of  $\sigma$  for a spel  $c$  (determined by a local condition) is performed by computing the values of the  $s_n^c$  using (1) and satisfying the conditions of (2) at  $c$ . Theorem 1 says that there is a unique  $M$ -semisegmentation which satisfies this reasonable property simultaneously everywhere, and that this  $M$ -semisegmentation is in fact an  $M$ -segmentation provided that the seeded  $M$ -fuzzy graph is connectable.

The original algorithm for computing the fuzzy segmentation according to the specifications above, called MOFS (multi object fuzzy segmentation) can be found in [2, 3], as well as the proofs for Theorem 1.(ii) and 1.(iii). The proof of Theorem 1.(i) can be found on [3].

The affinities  $\psi_m$  can be specified in several ways. In [2], as well as in here, the affinities and the  $V_m$ ,  $1 \leq m \leq M$  are specified by using the information on the neighborhood of seed spels clicked by a user. When the user clicks on a spel and associates it with an object  $m$ , he/she is saying that it is certain that the spel belongs to the object  $m$ . Then, we collect information on a  $3 \times 3$  neighborhood, defining  $g_m$  to be the mean and  $h_m$  to be the standard deviation of the average brightness for all edge-adjacent pairs of spels in  $V_m$  and  $a_m$  to be the mean and  $b_m$  to be the standard deviation of the absolute differences of brightness for all edge-adjacent pairs of spels in  $V_m$ . We then define  $\psi_m(c, d)$  to be 0 if  $c$  and  $d$  are not edge-adjacent and to be  $[\rho_{g_m, h_m}(g) + \rho_{a_m, b_m}(a)] / 2$  if they are, where  $g, h, a, b$  are as defined above and the function  $\rho_{r,s}(x)$  is the probability density function of the Gaussian distribution with mean  $r$  and standard deviation  $s$  multiplied by a constant so that the peak value becomes 1.

After the initialization steps, the central part of the MOFS algorithm updates the best guesses for the final values of the  $\sigma_m^c$  for all  $c \in V$ . A current value is replaced by a larger one if it is found that there is a  $\sigma_m$ -chain from a seed spel in  $V_m$  to  $c$  of  $\psi_m$ -strength greater than the old value (the previously maximal  $\psi_m$ -strength of the known  $\sigma_m$ -chains of this kind) and it is replaced by 0 if it is found that (for an  $n \neq m$ ) there is a  $\sigma_n$ -chain from a seed spel in  $V_n$  to  $c$  of  $\psi_n$ -strength greater than the old value of  $\sigma_m^c$ .

The total computational complexity of the algorithm is  $O(N(\log N + ML))$ , where  $N$  denotes the number of elements of  $V$ ,  $M$  is the number of objects, and  $L$  denotes the number of neighbor spels, usually a small number in the application of image segmentation (4 or 8 in 2D square grid images and 6, 18 or 26 in 3D cubic grid volumes).

## 2.1 Fast Fuzzy Segmentation

Even though we were able to segment a 3D image with more than 7,000,000 spels in approximately 4 minutes, as shown on [3], this response time may not be sufficiently fast for some applications. A fast implementation of the original MOFS algorithm for

computing the simultaneous fuzzy segmentation of multiple objects, introduced in [2], was presented in [3]. (The segmentation of the same 3D image mentioned above using the fast implementation was achieved in 35 seconds.) This fast implementation is briefly described below.

The original MOFS algorithm belongs to the class of greedy algorithms [14], and was implemented using a binary heap to keep a partial ordering according to the spels' current  $\sigma^0$  values. The values of the chains stored in the heap are updated as new chains with values greater than the ones currently stored are found. Since a heap is used to keep the partial ordering needed by the algorithm, the operations of spel insertion, deletion and  $\sigma^0$  update take  $O(\log N)$ , where  $N$  is the number of spels in  $V$ . (A heap has to be used because the values of  $\sigma^0$  can assume any real value between 0 and 1.)

However, suppose that the set of nonzero fuzzy spel affinities for a particular class of problems is always a subset of a fixed set  $A$ . Let  $K$  be the cardinality of the set  $A \cup \{1\}$ , and let  $1 = a_1 > a_2 > \dots > a_K > 0$  be the elements of  $A$ . For example, in many applications the quality of the fuzzy segmentation is not significantly affected if we round each fuzzy spel affinity to three decimal places. If we use such rounded spel affinities, then we can take  $A = \{0.001, 0.002, \dots, 0.999, 1.000\}$ , so that  $K = 1000$  and  $a_k = 1.001 - k/1000$ .

By restricting the affinity values to a fixed set of values, as above, we can use an  $M \times K$  array  $U[m][k]$  of sets of nodes that represent spels, where  $M$  is (as before) the number of objects. This array stores the spels according to their  $\sigma^0$  values, thus, maintaining the partial ordering of the  $\sigma^0$  values. Now, the cost of the operations of spel insertion and removal and the update of the  $\sigma^0$  values becomes proportional to a constant, i.e.,  $O(1)$ . (Similar ideas were used in [15] to speed up the algorithm of [1].) This implementation (shown below in pseudo-code first published in [3]) is most effective if all of its data structures (with space complexity  $O(M(K + V))$ ) can be held in the main memory.

### Fast implementation of the MOFS algorithm

1. **for**  $c \in V$  **do**
2.     **for**  $m \leftarrow 0$  **to**  $M$  **do**
3.          $\sigma_m^c \leftarrow 0$
4.     **for**  $m \leftarrow 1$  **to**  $M$  **do**
5.         **for**  $c \in V_m$  **do**
6.              $\sigma_0^c \leftarrow \sigma_m^c \leftarrow 1$
7.              $U[m][1] \leftarrow V_m$
8.         **for**  $k \leftarrow 2$  **to**  $K$  **do**
9.              $U[m][k] \leftarrow \emptyset$
10.     **for**  $k \leftarrow 1$  **to**  $K$  **do**
11.         **for**  $m \leftarrow 1$  **to**  $M$  **do**
12.             **while**  $U[m][k] \neq \emptyset$  **do**
13.                 remove a spel  $d$  from the set  $U[m][k]$
14.                  $C \leftarrow \{c \in V \mid \sigma_m^c < \min(a_k, \Psi_m(d, c)) \text{ and } \sigma_0^c \leq \min(a_k, \Psi_m(d, c))\}$
15.                 **while**  $C \neq \emptyset$  **do**
16.                     remove a spel  $c$  from  $C$
17.                      $t \leftarrow \min(a_k, \Psi_m(d, c))$
18.                     **if**  $\sigma_0^c < t$  **then do**

- 19. remove  $c$  from each set in  $U$  that contains it
- 20. **for**  $n \leftarrow 1$  **to**  $M$  **do**
- 21.      $\sigma_n^c \leftarrow 0$
- 22.      $\sigma_0^c \leftarrow \sigma_m^c \leftarrow t$
- 23.     insert  $c$  into the set  $U[m][l]$  where  $l$  is the integer such that  $a_l = t$

This fast version of the MOFS algorithm was chosen to be used in the color video segmentation because the small running times achieved allows an user to include or delete seed spels and rerun the algorithm a few times in a reasonable amount of time.

### 3 Color Video Segmentation

The fuzzy affinity function is now defined as an average of six components, two for each YUV channel, the color model chosen to code the input images. The two components of each channel used in the fuzzy affinity function are defined in the same way as before, i.e., using the mean and the standard deviation of the average value for all edge-adjacent pairs of spels in  $V_m$  and the mean and the standard deviation of the absolute differences of values for all edge-adjacent pairs of spels in  $V_m$ . Once more,  $\psi_m(c, d)$  was defined to be 0 if  $c$  and  $d$  are not edge-adjacent and to be

$$[\rho_{gY_m, hY_m}(g) + \rho_{aY_m, bY_m}(a) + \rho_{gU_m, hU_m}(g) + \rho_{aU_m, bU_m}(a) + \rho_{gV_m, hV_m}(g) + \rho_{aV_m, bV_m}(a)] / 6 \tag{3}$$

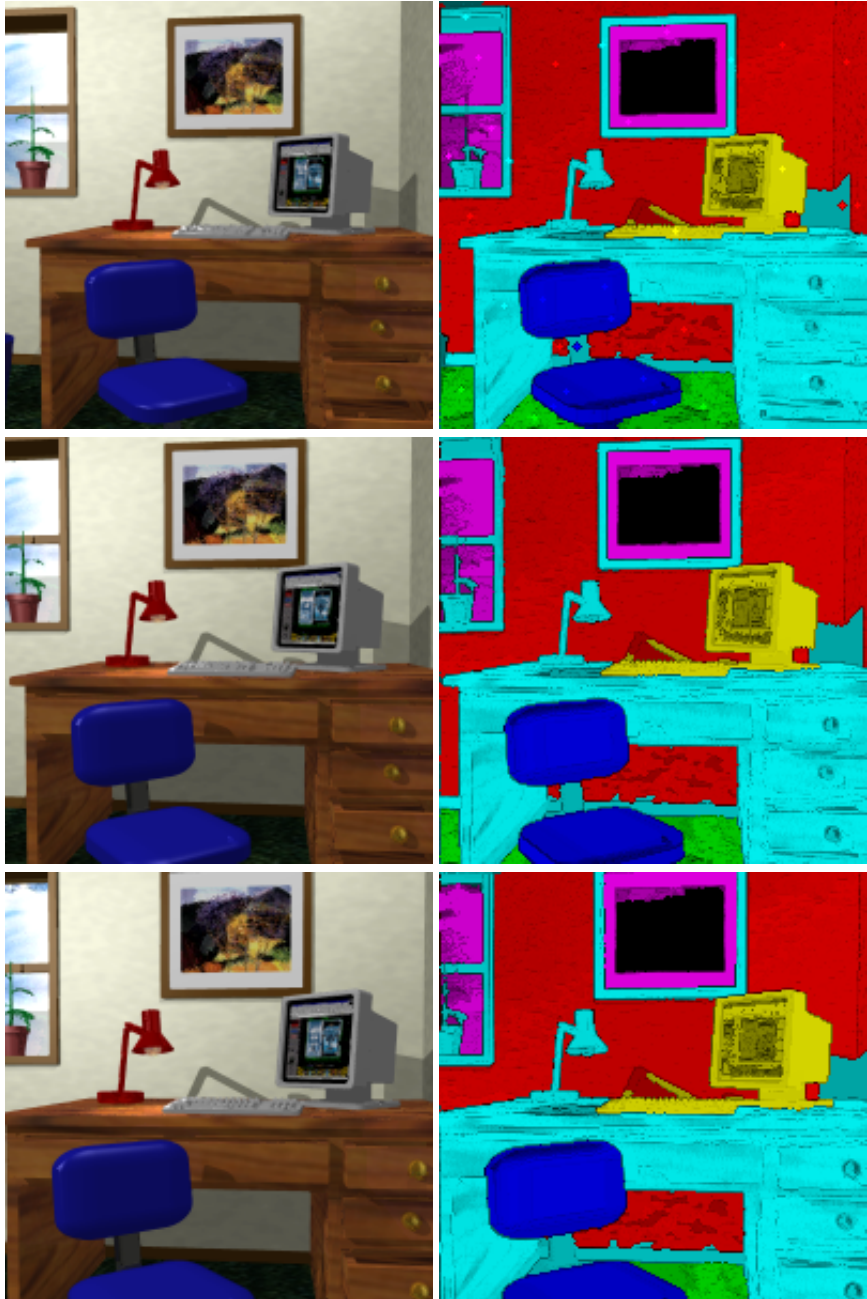
if they were, where  $g$  is the mean and  $a$  is the absolute difference of the Y, U or V channel values of  $c$  and  $d$  and the function  $\rho_{r,s}(x)$  is the probability density function of the Gaussian distribution with mean  $r$  and standard deviation  $s$  multiplied by a constant so that the peak value becomes 1.

The affinity function defined above can be used for segmenting both 2D or 3D images. The volume composed by the video sequence frames is treated as a volume on a cubic grid with face adjacency, and the user can select seed spels on any slice (frame) of the volume. By doing this, the user can successfully segment objects that do not appear on the first frame. Moreover, the user can identify disconnected objects in the time direction as being the same one, as in the case of an object that is visible, occluded for some frames and visible again in the video sequence.

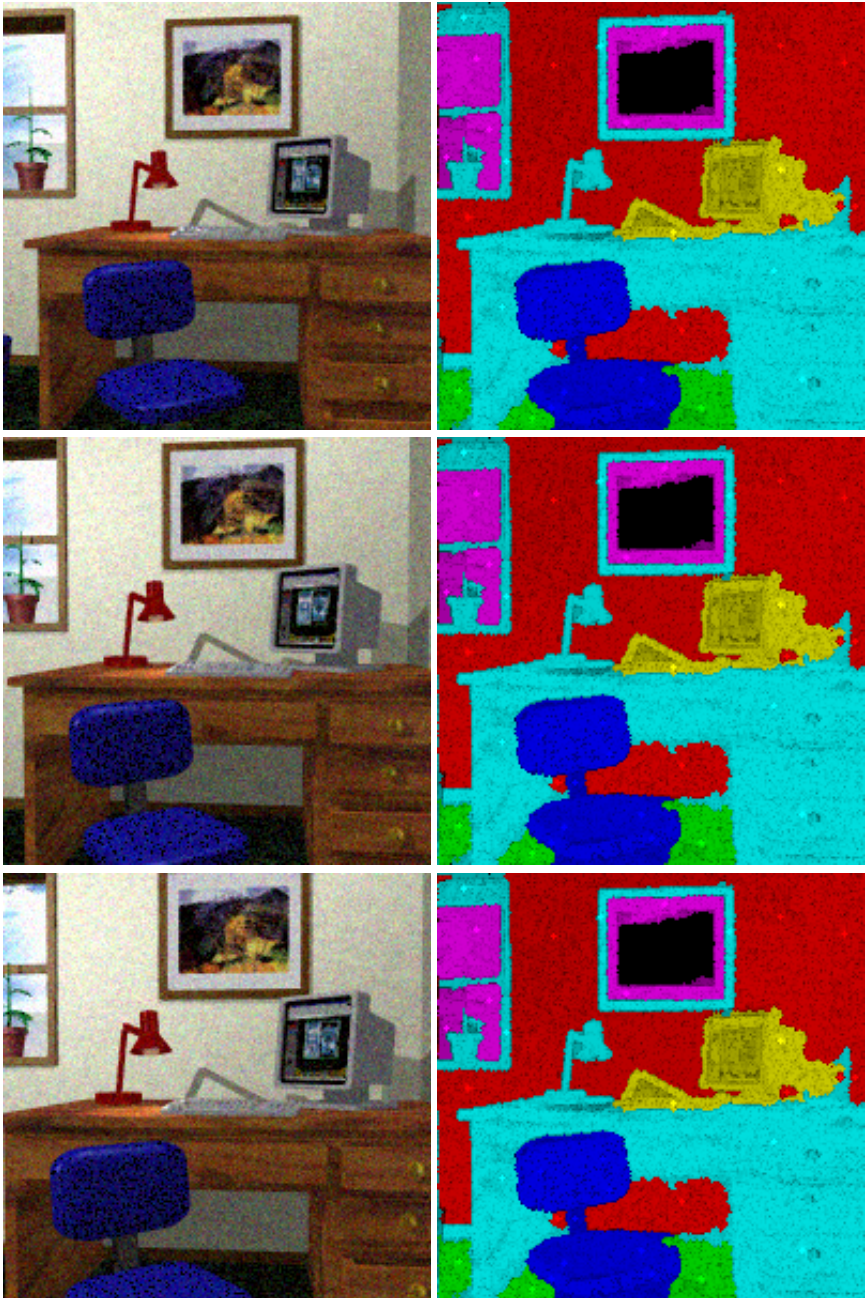
The idea of using the fast implementation of the fuzzy segmentation algorithm described here is that besides the ability of the fuzzy segmentation algorithm for segmenting images corrupted by noise and/or varying illumination, the small total segmentation time of the algorithm (not including the user interaction) allows the user to inspect the results and add or remove seed spels to achieve a better segmentation. This adjustment process of the segmentation can be repeated a few times and still be finished in a short period of time (in the order of seconds or a few minutes). The interactivity and the speed of the method makes it possible for the user to include high level knowledge in the segmentation process.

### 4 Experiments

First, we used the fast MOFS algorithm to segment a synthetic video shot (obtained from [16]) with and without added noise. The noisy sequence was corrupted by a

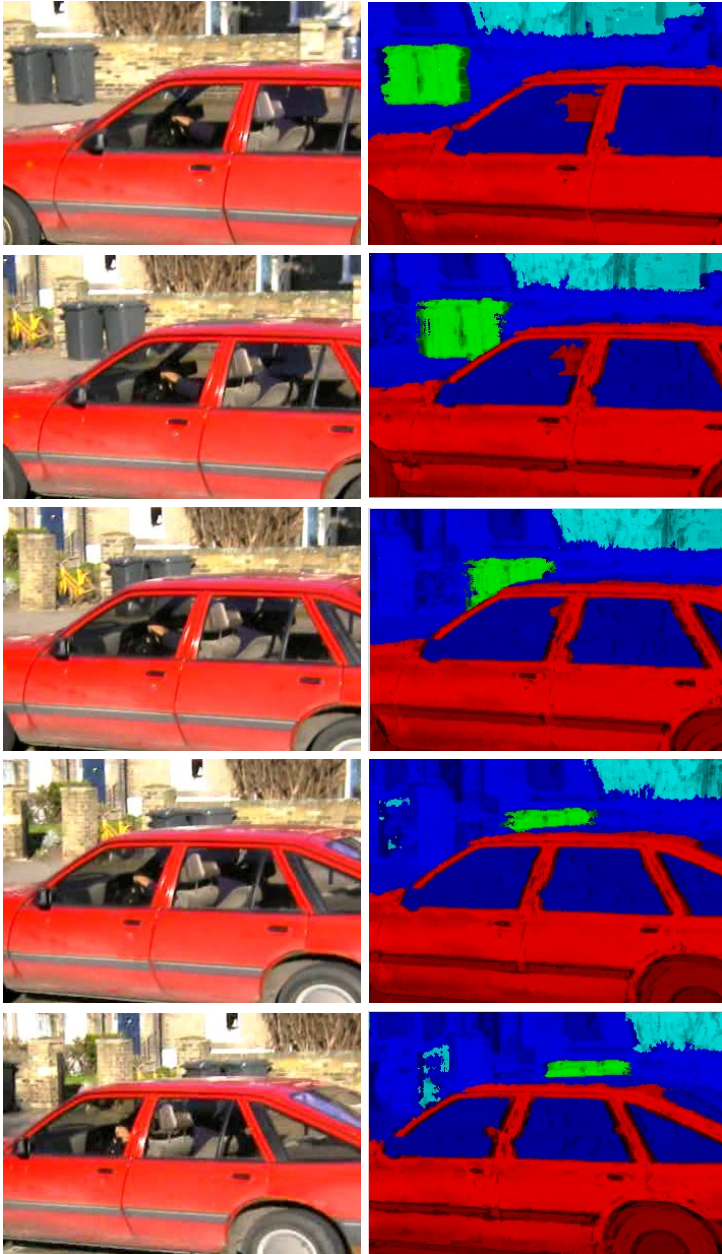


**Fig. 1.** Original frame images (left) and the corresponding fuzzy segmentation connectedness maps (right) for the 1st, 10th and 20th frames of the noiseless video sequence

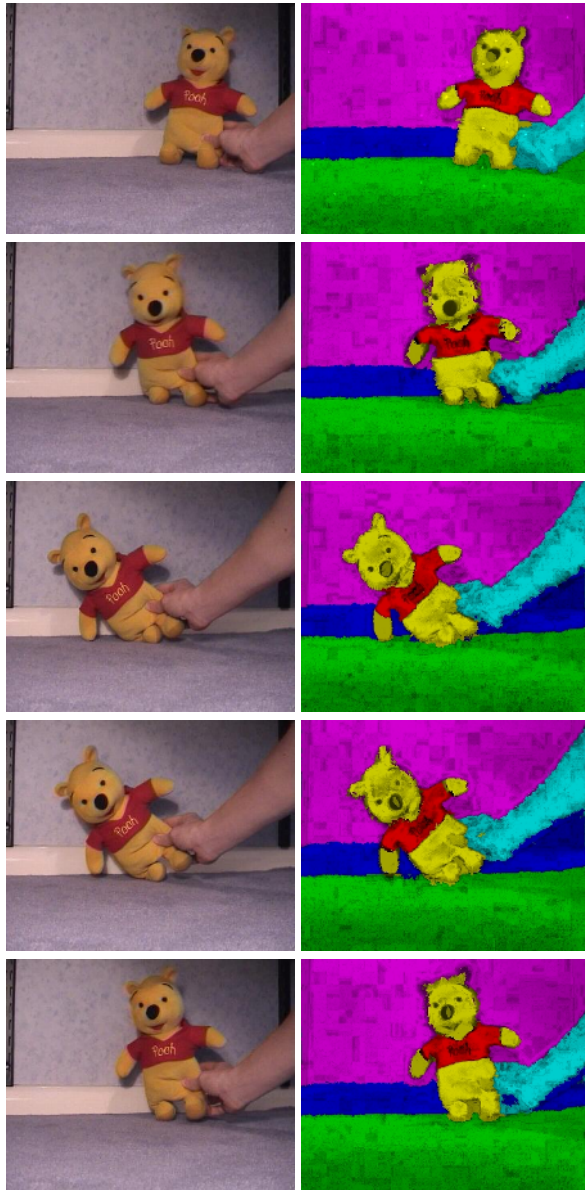


**Fig. 2.** Original frame images (left) and the corresponding fuzzy segmentation connectedness maps (right) for the 1st, 10th and 20th frames of the video sequence corrupted by noise with  $\sigma = 10$  units in all three channels





**Fig. 3.** Original frame images (left) and the corresponding fuzzy segmentation connectedness maps (right). The rows correspond to the 1st, 6th, 11th, 16th and 21st frames (slices) of the video sequence (volume).



**Fig. 4.** Original frame images (left) and the corresponding fuzzy segmentation connectedness maps (right). The rows correspond to the 1st, 16th, 31st, 46th and 61st frames (slices) of the video sequence (volume).

Gaussian noise with mean 0 and standard deviation of 10 units (in the range  $[0, 255]$ ) for all three channels, and pseudo-random values of this distribution were added to all three channels of the video sequence. The 20 frame long video sequences of  $200 \times 200$

pixels were then segmented interactively on a single try, and the accuracy of the segmentations were measured by counting the number of misclassified pixels for all frames of the sequenced when compared to the ground truth. Figures 1 and 2 show the 1st, 10th and 20th frames of the noiseless and noisy synthetic video sequences, respectively.

The segmentations shown in Figures 1 and 2 were obtained by running the algorithm after the user selected seed spels for seven objects. The segmentation times for these sequences were the same, 2 seconds. As mentioned before, if the user wants to refine a segmentation, he/she can add or remove seed spels, thus, providing temporal and spatial high-level information about the relationships of the objects in the video sequence.

We then determined the accuracy of the segmentations as the percentage of pixels correctly segmented when compared to hand segmented frames of the noiseless synthetic sequence. The accuracy of the noiseless and noisy segmentations are 94.3% and 92.0%, respectively.

To show the usefulness of the algorithm to the segmentation of real color video sequences, we performed the segmentation of two real videos. The real images of the 21 frame car sequence with  $352 \times 240$  pixels shown in the left column of Figure 3 are the 1st, 6th, 11th, 16th and 21st frames of the video sequence, while on the right are the corresponding fuzzy connectedness maps. These maps were obtained by running the algorithm after the user selected seed spels for five objects. The segmentation time for this sequence was 3 seconds.

Figure 4 shows 5  $360 \times 288$  frames of the 71 frame long Pooh sequence (left) and the corresponding segmentations (right). These maps were obtained by running the algorithm after the user selected seed spels for six objects. The segmentation time for this sequence was 23 seconds. Notice the faulty segmentation right below the arm on the right of the image on the 3rd, 4th and 5th rows. After visualizing this result, the user can add seeds for a new object for that area and rerun the algorithm. (The interaction time needed for selecting the seeds in these examples range from 30 to 60 seconds.) The high level information that this new object actually belongs to the same object as the rest of the wall can be used afterwards, for example, when rendering the background wall of the video sequence using NPR techniques.

## 5 Conclusion

In this paper we propose to use the fast implementation of the simultaneous fuzzy segmentation of multiple objects for segmenting pre-acquired color video sequences. We consider the frames of the sequences as slices of a 3D volume, since our objective is to segment end-to-end pre-acquired video sequences, for rendering them later using NPR techniques. The end-to-end segmentation has to be performed, so its result can be used for enforcing temporal coherence in the scene generated by the rendering technique.

The results show a good accuracy rate of the segmented pixels for the synthetic sequences, with and without noise, and high quality segmentations of real video sequences, all with very small running times. This allows the user to rerun the algorithm after adding/removing seed spels, if the result of the segmentation was not satisfactory.

Future work will concentrate on incorporating motion cues (such as the algorithms presented in [17, 18]) in the fuzzy affinity functions for more accurate segmentation of different objects that have similar colors.

## References

1. Udupa, J., Samarasekera, S.: Fuzzy connectedness and object definition: Theory, algorithms and applications in image segmentation. *Graph. Models Image Proc.* **58** (1996) 246–261
2. Herman, G., Carvalho, B.: Multiseeded segmentation using fuzzy connectedness. *IEEE Trans. on Pattern Anal. Mach. Intell.* **23** (2001) 460–474
3. Carvalho, B., Herman, G., Kong, T.: Simultaneous fuzzy segmentation of multiple objects. *Disc. Appl. Math.* **151** (2005) 55–77
4. Hertzmann, A., Perlin, K.: Painterly rendering for video and interaction. In: *Proc. ACM Non-Photorealistic Animation and Rendering (NPAR)*. (2000) 7–12
5. Litwinowicz, P.: Processing images and video for an impressionist effect. In: *Proc. ACM SIGGRAPH*. (1997) 407–414
6. Collomosse, J., Rowntree, D., Hall, P.: Stroke surfaces: Temporally coherent artistic animations from video. *IEEE Trans. Visualiz. and Comp. Graph.* **11** (2005) 540–549
7. Wang, J., Xu, Y., Shum, H.Y., Cohen, M.: Video tooning. *ACM Trans. on Graph.* **23** (2004) 574–583
8. Rosenfeld, A.: Fuzzy digital topology. *Inform. and Control* **40** (1979) 76–87
9. Carvalho, B., Gau, C., Herman, G., Kong, T.: Algorithms for fuzzy segmentation. *Pattern Anal. Appl.* **2** (1999) 73–81
10. Carvalho, B., Garduño, E., Herman, G.: Multiseeded fuzzy segmentation on the face centered cubic grid. In Singh, S., Murshed, N., Kropatsch, W., eds.: *Advances in Pattern Recognition: Second International Conference - ICAPR 2001 (LNCS 2013)*, Rio de Janeiro, Brazil, Springer-Verlag (2001) 339–348
11. Zahn, C.: Graph-theoretic methods for detecting and describing Gestalt clusters. *IEEE Trans. Comp.* **1** (1971) 68–86
12. Duda, R., Hart, P.: *Pattern Classification and Scene Analysis*. John Wiley & Sons, New York (1973)
13. Jain, A., Murty, M., Flynn, P.: Data clustering: a review. *ACM Comput. Surveys* **31** (1999) 264–323
14. Cormen, T., Leiserson, C., Rivest, R.: *Introduction to Algorithms*. MIT Press, Cambridge, MA (1990)
15. Nyul, L., Falcão, A., Udupa, J.: Fuzzy-connected 3D image segmentation at interactive speeds. *Graph. Models* **64** (2002) 259–281
16. Computer Vision Homepage: University of Otago, available at <http://www.cs.otago.ac.nz/research/vision/Research/OpticalFlow/opticalflow.html> (2006)
17. Galun, M., Apartsin, A., Basri, R.: Multiscale segmentation by combining motion and intensity cues. In: *Proc. IEEE Comp. Soc. Conf. on Comp. Vision and Patt. Recog.* (2005) 256–263
18. Khan, S., Shah, M.: Object based segmentation of video using color, motion and spatial information. In: *IEEE Comp. Soc. Conf. on Comp. Vision and Patt. Recog. Volume 2*. (2001) 746–751