

# Automatic Design of Morphological Operators for Motion Segmentation

ROBERTO HIRATA JR.<sup>1</sup>,  
J. BARRERA<sup>1</sup>,  
F. C. FLORES<sup>1</sup> AND  
R. A. LOTUFO<sup>2</sup>

<sup>1</sup> Instituto de Matemática e Estatística - USP  
PO Box 66.281  
05315-970 São Paulo - SP - Brazil  
<hirata,jb,fcflores>@ime.usp.br

<sup>2</sup> Faculty of Electrical and Computer Engineering - UNICAMP  
PO Box 6101  
13081-970 Campinas - SP - Brazil  
lotufo@dca.fee.unicamp.br

**Abstract.** A problem of interest in digital video edition is the elimination of moving objects from one video and the introduction of pieces of other videos in their places. A fundamental problem to build computational tools for this purpose is the segmentation of moving objects. This paper approaches this problem by a new technique, based on Beucher-Meyer's paradigm, with markers detected by morphological operators designed by computational learning techniques. The objects in the first frames of the video are marked manually and used to train the markers detector. Then, the operator designed is used to mark the objects in the other frames and the paradigm is applied to all frames marked by the detector. Some synthetic and real world examples illustrate the application of the technique proposed. Complex situations as occlusion are examined.

## 1 Introduction

A *digital video* is an ordered sequence of digital images, called *frames*. The order is defined by the instant the frame was taken, for instance: frame 1 is the image taken at the instant  $t_0$ , frame 2 is the image taken at instant  $t_0 + \Delta t$ , frame 3 is taken at  $t_0 + 2\Delta t$ , and so on.

Digital video technology has a strong impact on video production, reducing costs and opening a spectrum of new possibilities. A great advantage of digital videos, in relation to the analogic ones, is that the edition tasks can be done by computational tools. Besides performing easily conventional tasks, the computational manipulation offers several new edition techniques such as new alternatives for video *mixing* (i.e., combining pieces of different videos for the production of a new video.)

A well known technique for video mixing is *chroma key* [1]. This technique permits the mixing of two videos, however it supposes some special cares. When filming the main video (i.e., the video in which there will be inserted pieces of other videos), the objects to

be substituted should be covered with a color (usually blue or green) different of the colors of the other objects in the scene. For example, if the main video presents a room that has a window with a blue covering, it may be substituted by the corresponding region of the other video. Thus, the mixing could simulate a room filmed in New York to be, for example, in Paris. The image processing technique applied in the chroma key is classical pattern recognition, using pixel color intensities as attributes.

A much less simple image processing task is introducing pieces of other videos in place of non covered objects. This task depends essentially on segmenting the moving objects that will be substituted, what usually is not easy. There are no commercial products available that properly performs this task. So, usually, it is avoided by video producers. When this is not possible, the only alternative is the manual solution, that is boring and expensive. The literature on the subject is also very restrict.

This paper presents a new approach for this problem based on Beucher-Meyer paradigm, with the object

markers detector designed by computational learning techniques.

Beucher-Meyer paradigm is one of the most powerful known techniques for image segmentation. The great quality of this strategy is changing problems of edge detection into problems of marker detection (i.e., finding a small connected component inside the object to be segmented), that usually is much simpler. The markers detected are used as reference for filtering the gradient of the input image and, finally, the watershed gives the exact (without blur) edges of the desired objects.

In spite of the evolution due to Beucher-Meyer paradigm, the problem of having a systematic approach to design operators that detect object markers still persists. An alternative is to design these operators by computational learning.

The technique proposed consists of designing operators that perform object tracking (or, equivalently, detect object markers) by training with a few frames of the sequence and applying the designed operator to the other frames. Having marked the objects in all frames, Beucher-Meyer paradigm is applied for detecting the edges of these objects.

A condition for applying this technique is that the training frames must be statistically similar to the other frames where the objects appear. When a set of frames with different statistical characteristics appear in the sequence, due to changes in image illumination, resolution, camera position, etc., a new training is necessary before processing the next frames.

Following this Introduction, Section 2 reviews Beucher-Meyer paradigm. Section 3 presents a family of operators, the aperture filters, used in the statistical design of operators for object tracking. Section 4 shows the application of the technique proposed on synthetic and real world images. Section 5 gives some conclusion and future steps of this research.

## 2 Beucher-Meyer Paradigm

Image segmentation is an important and difficult problem. It is important because it is part of most image analysis solutions [2, 3]. It is difficult because it is an ill-posed problem [4].

Mathematical Morphology provides us with a powerful segmentation method, called Beucher-Meyer paradigm [5]. This method is useful to find exact borders of specified objects. It simplifies the segmentation process reducing the problem of segmenting objects directly to the problem of finding markers for the specified objects [6].

The known approaches for image segmentation use basically two ideas: finding the borders of the objects

in an image or grouping the points of the image which are similar in order to get the points of the objects. Beucher-Meyer paradigm is based on finding the borders of the objects one wants to segment using the watershed operator [7, 8, 9] composed with some other morphological operators that are useful to prepare the image to be segmented. This preparation is necessary to eliminate the borders of the objects we are not interested in, and also the borders that appear due to noise in the image.

Borders are, in general, discontinuities in the image and they can be detected by differencing operators, like the morphological gradient [10]. This operator is, however, very sensitive to noise in the image, i.e., it enhances the gray-level transitions due to the borders of the objects but also the transitions due to noise. Hence, the gradient image is usually a noisy image in the sense that it carries more information than it is necessary. The application of the watershed operator to the gradient image usually results in an over-segmented image [9, 6]. The solution to eliminate the borders one does not want is to apply an operator that changes the homotopy [10] of the gradient function [5, 6].

Let  $\mathbb{E}$  be a non empty set (usually  $\mathbb{E} \subset \mathbb{Z} \times \mathbb{Z}$ ). A gray-level image in the discrete space, i.e., an image where all points can take values in  $\mathbb{K} = [0, k]$ ,  $k \in \mathbb{Z}$  can be represented by a function  $f : \mathbb{E} \rightarrow [0, k]$ . The set of all gray-level images is represented by  $\mathbb{K}^{\mathbb{E}}$  and an image  $f$  is a point of that space ( $f \in \mathbb{K}^{\mathbb{E}}$ ).

The homotopy operator is a filter [10, 11] which main property is to guarantee that the watershed lines found in the filtered image are a subset of the watershed lines found in the gradient image.

After the application of the homotopy filter on the gradient image, the watershed operator finds only the borders of the objects one wants to segment. This method is known as “Beucher-Meyer paradigm” [5].

Beucher-Meyer paradigm simplifies the segmentation problem into the problem of finding markers to the objects we want to segment. However, this can still be a hard segmentation problem because it is usually heuristically done.

Figures 1 to 3 show an example of the paradigm applied to an image of muscle fibers. Figure 1a shows the image to be segmented and Fig. 1b shows the result (inverted to show the details) of the gradient applied to that image. Fig. 2a shows the over-segmentation due to applying the watershed operator directly on the gradient image. Figure 2b shows markers for the dark objects (the objects we want to segment). Figure 3a shows the result (inverted to show the details) of the homotopy operator applied to the gradient image using the markers image to specify the desired objects.

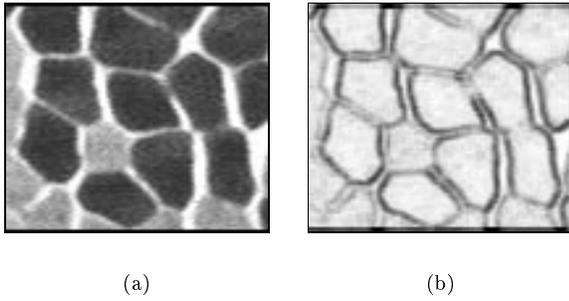


Figure 1: Beucher-Meyer paradigm (a) muscle; (b) gradient (inverted).

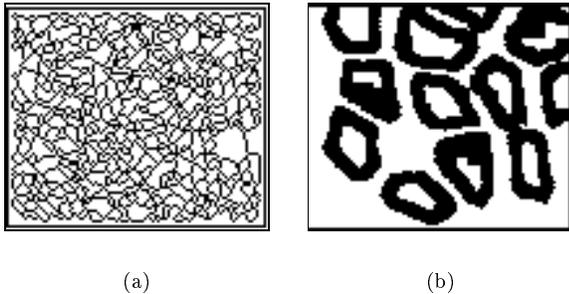


Figure 2: Beucher-Meyer paradigm (a) over segmentation; (b) markers.

Figure 3b shows the final result (the watershed lines) composed with the original image.

### 3 Aperture Operators

The basic operator design problem for a pair of random images,  $h$  to be observed and  $g$  to be estimated, is to find an operator  $\Psi$  in a class of operators that minimizes an error measure between  $\Psi(h)$  and  $g$ . A common way to proceed is to define  $\Psi$  locally by some function  $\psi$  operating on  $h$  in some window  $W_p = \{w_1 + p, w_2 + p, \dots, w_n + p\}$  about a point  $p$  by  $\Psi(h)(p) = \psi(h(W_p))$ , where  $h(W_p)$  is the restriction of  $h$  to  $W_p$ . For digital images,  $W_p$  is the translate of a finite window  $W$  to  $p$  and the problem is to find an optimal estimator of the random variable  $g(p)$  in terms of the observed random variables in  $W_p$ . Often  $\psi$  is constrained to a subclass of functions over  $W$ .

The optimal image operator  $\Psi$  is called a  $W$  operator and is defined by  $\Psi(h)(p) = \psi_W(\mathbf{X}) = E[Y|\mathbf{X}]$ ,

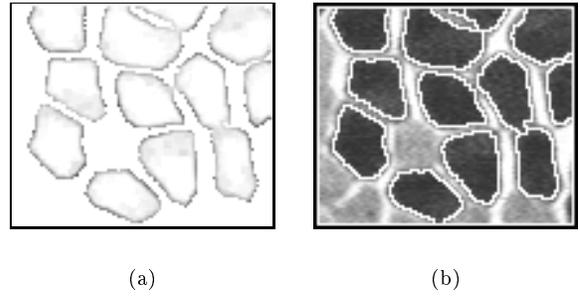


Figure 3: Beucher-Meyer paradigm (a) homotopy (inverted); (b) final result.

where  $E[Y|\mathbf{X}]$  is the conditional expectation of  $Y$  given  $\mathbf{X}$  and  $\mathbf{X} = (X_1, X_2, \dots, X_n)$  is observed in  $W_p$  [12]. Excluding the situation where  $X_1, X_2, \dots, X_n$ , and  $Y$  are jointly Gaussian, in which case the optimal  $W$ -operator coincides with the optimal linear operator [13], one is typically confronted with estimating the optimal operator from realizations (sample data) under various forms of constraint on optimization. Moreover, for a discrete gray-scale range, which is our concern in this paper,  $\psi_W(\mathbf{X})$  is obtained by quantizing  $E[Y|\mathbf{X}]$ . In the unconstrained discrete case, in which there are no modeling assumptions, optimization reduces to estimating the quantized conditional expectation. When  $X_1, X_2, \dots, X_n$ , and  $Y$  are binary and the window is not too large, this can be successfully accomplished [14, 15]; for larger windows, various forms of constraint have been employed [16]. In particular, a recursive error representation has been used to estimate the morphological basis of the optimal increasing operator [17]. Optimal binary operators have found application in digital document processing [18].

A  $W$ -operator  $\psi_W$  locally defines an operator  $\Psi_W$  by

$$\Psi_W(h)(p) = \begin{aligned} &\psi_W(h(W_p)) = \\ &\psi_W(h(p + w_1), \dots, h(p + w_n)) \end{aligned} \quad (1)$$

By appropriately indexing  $w_1, w_2, \dots, w_n$ , the window can be centered at  $p$ . When using sample data to estimate the conditional expectation, there is an implicit assumption that, as a random function, the image is stationary (or sufficiently so) that the same computational operator  $\psi_W$  is applied at each point  $p$ .

In this paper the windowing is in both the domain and gray-level range of the operator. Not only are the observations constrained to the domain window  $W$ , the values of the observations are constrained to a range

window  $K = \{-k, -k + 1, \dots, k\}$ . For  $j = 1, 2, \dots, n$ , define the truncated random variable by

$$X_j^* = \begin{cases} X_j & : -k \leq X_j \leq k \\ k & : X_j > k \\ -k & : X_j < -k \end{cases} \quad (2)$$

and let  $\mathbf{X}^* = (X_1^*, X_2^*, \dots, X_n^*)$ . The filters of the form  $\Psi(\mathbf{X}^*)$  will be called *aperture filters*, where the aperture  $\mathcal{A}$  is the product set  $W \times K$ . Geometrically, observations within an aperture  $W \times K$  are unchanged and those outside of  $W \times K$  are projected vertically into the boundary of  $W \times K$  (from above or below). The morphological representation of aperture filters has been treated previously (under the name of  $WK$  operators) [19].

The aperture operator ( $\mathcal{A}$  operator)  $\Psi_{\mathcal{A}}$  is also defined via an aperture function  $\psi_{\mathcal{A}}$ , and the aperture  $W \times K$  must be placed into the product space of the domain and the range. At  $t$ , the observed image value is  $h(p)$  and we center  $W \times K$  at  $(p, h(p))$ .

Aperture placement is illustrated in Figs. 4 and 5, which show additive white noise and blurring, respectively. In each figure, part *a* gives the ideal (uncorrupted) signal and part *b* gives the observed (corrupted) signal. Signals are shown as solid dots,  $\times$  marks the center of the aperture, shadowed dots show vertical projections of signal points into the aperture and black squares with a white  $+$  inside shows the observed value in the ideal signal. In part *b* of Fig. 4, the aperture is placed vertically at the observed value,  $\mathbf{x}^*$  (realization of  $X^*$ ) differs from  $\mathbf{x}$  (realization of  $X$ ) at four points and in part *b* of Fig. 5,  $\mathbf{x} = \mathbf{x}^*$ .

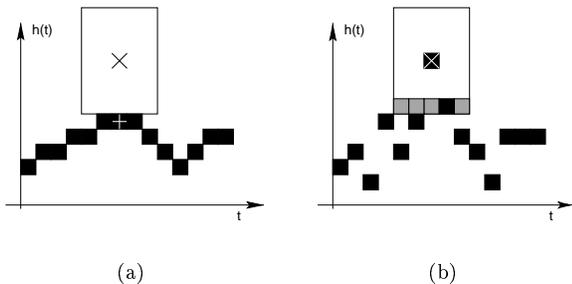


Figure 4: Aperture placement for additive white noise: (a) aperture on ideal signal; (b) aperture on observed signal.

Design of aperture operators from signal realizations requires estimation of the local operator  $\psi_{\mathcal{A}}$  from sample signal data and then representation of  $\psi_{\mathcal{A}}$  in

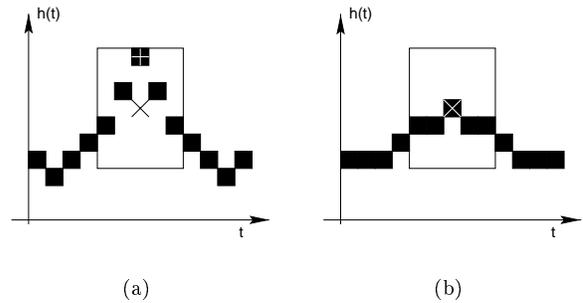


Figure 5: Aperture placement for blurring: (a) aperture on ideal signal; (b) aperture on observed signal.

some computational form.  $\psi_{\mathcal{A}}$  is estimated from training pairs of the form  $((\mathbf{x} - h(p))^*, y)$  extracted from pairs of signal realizations. Since we are interested in defining  $\psi_{\mathcal{A}}$  on a set of vectors in the aperture  $\mathcal{A}$ , we will suppress the vertical displacement  $h(t)$  when discussing training pairs and just write  $(\mathbf{x}^*, y)$ .  $\mathbf{x}^*$  is a *training example with label y*. If a particular  $\mathbf{x}^*$  is observed  $m$  times, then there are  $m$  labels  $y_1, y_2, \dots, y_m$  associated with it in training pairs. The mean of these labels provides an estimate  $\psi_{\mathcal{A}, N}(\mathbf{x}^*)$  of  $\psi_{\mathcal{A}}(\mathbf{x}^*)$ , where  $N$  is the total number of sample pairs over all possible vectors. Using the language of machine learning, we refer to  $\mathbf{x}^*$  as a *pattern* and  $\psi_{\mathcal{A}, N}(\mathbf{x}^*)$  as a *final label*. We denote the set of labeled patterns by  $\mathcal{M}_{\mathcal{A}}$ .

Having associated labels with the vectors observed during training, we need to provide a representation for the operator. Here we need to recognize that  $\mathcal{M}_{\mathcal{A}}$  is often (and in our case will almost always be) a proper subset of the set  $\mathcal{X}$  of all possible vectors and that  $\mathcal{X}$  is the true domain of  $\psi_{\mathcal{A}, N}$ . A representation for  $\psi_{\mathcal{A}, N}$  will complete the definition of the operator. It is critical that the representation extends the definition of  $\psi_{\mathcal{A}, N}$  from  $\mathcal{M}_{\mathcal{A}}$  to  $\mathcal{X}$  in a consistent manner, meaning that if  $l$  is the final label for  $\mathbf{x}^* \in \mathcal{M}_{\mathcal{A}}$ , then, upon representation, the operator must assign  $l$  to  $\mathbf{x}^*$ . Given this consistency, the definition of  $\psi_{\mathcal{A}, N}$  is unambiguous. In machine learning, the extension of the definition of  $\psi_{\mathcal{A}, N}$  from  $\mathcal{M}_{\mathcal{A}}$  to  $\mathcal{X}$  is called *prediction*, or *generalization*, because the extension “predicts”, or “generalizes” a learned concept. Prediction is important because the performance of a designed operator depends on the way in which it maps vectors not observed during training, those lying in  $\mathcal{X} - \mathcal{M}_{\mathcal{A}}$ .

In this paper, we use decision trees for operator representation. They are fast to be implemented, extend the learned operator consistently, and tend to pro-

vide good generalization [20, 21, 22]. For details on how they have been used in this context see [12].

## 4 Applications

In this section we will show the methodology of the system, the problems involved in the motion segmentation, the solutions that are not implemented yet and some applications of the method.

### 4.1 Methodology

The methodology applied for the automatic design of morphological operators for motion segmentation is simple because one of the objectives of the automatic design of operators is that it can be done by a non specialist in image processing. The steps to be followed are:

- To train an operator,  $\Psi$ , giving some pairs of images ( $\{f_i, g_i\}$  where  $f_i$  is the  $i$ -th frame of the sequence and  $g_i$  is the corresponding marker image) which should reflect statistically the different situations the object of interest appear. That means that the indexes  $i$  do not need to form a sequence with uniform step like  $\{f_i, g_i\}, \{f_{i+1}, g_{i+1}\}$ , etc. Since one is interested on only one object of the image, it is natural that the operator does not need to be trained using the whole image but just from some neighborhood of the object of interest. This is done by a mask  $m_i$  that will be given to the system together with the training pair  $\{f_i, g_i\}$ . The mask will restrict the domain where the training samples  $(x^*, y)$  will be observed to the area inside the mask. Figure 6 shows an object in the  $i$ th frame  $f_i$  and a possible mask  $m_i$  used to restrict the learning for the pair  $\{f_i, g_i\}$ . The radius of the circle that forms the mask is based on a parameter given by the user that reflects the speed of the object in the sequence. Since the system supports the classification of several objects, a tool to track more than one object is being designed. In the training of the operator, the user may have to try different window sizes and ranges, i.e., different  $W$  and  $K$  values, because this part is not automatized yet.

- To apply the operator to the other frames of the sequence, the user gives the location of the object in the first frame of the application sequence manually and a parameter related to the speed of the objects in the sequence. The application of the operator is also restricted to a mask but this time the mask  $m_{i+1}$  is built from the result of the segmentation of the  $i$ th frame  $f_i$  plus the initial speed parameter given by the user (for instance, a dilation with a structuring element larger than  $\Delta S = V_0 \Delta T$ ). Figure 6 shows an object in the  $(i+1)$  frame and the respective application mask built

from the information of the object segmented in the  $i$ th frame.

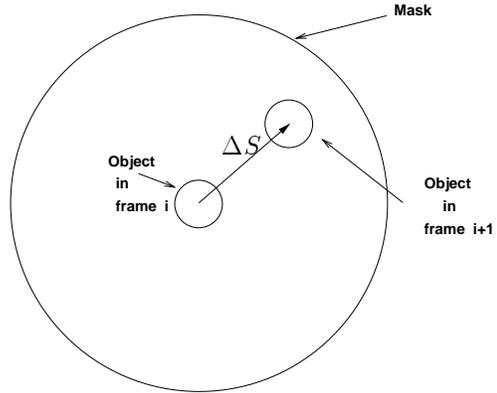


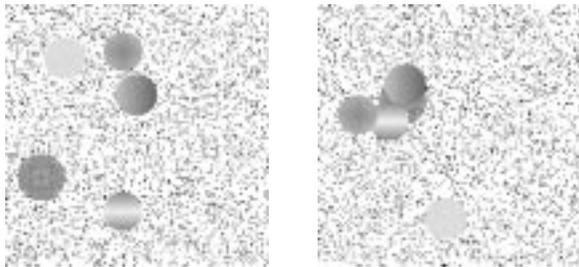
Figure 6: Restriction mask

- To filter the image because the result of  $\Psi$  may not be perfect (which is usually the case). The filter is necessary to eliminate the markers of low statistical confidence (usually isolated points or small connected components.) This filter is a composition of a connected filter and a pruning operator. The connected filter is parameterized by a structuring element  $B$  (to define the connectivity) and it eliminates connected components of area less than a value  $a$  specified by the user. This process of eliminating a small connected component is done by assigning the gray-value of the largest nearest neighbour connected component to it. The value  $a$  specifies to the system the minimum size of connected objects that can be considered a marker, i.e., the system will eliminate the markers smaller than  $a$  considering the connectivity specified by  $B$ . The prunings are necessary because the marker resulting after the filtering may be larger than the object.
- To apply Beucher-Meyer paradigm using the homotopy and the watershed operators.

### 4.2 Simulated experiment

To test the ideas shown above, a synthetic image sequence of a simulation of moving disks filled with different patterns has been generated. Figure 7a shows an example of such image. The sequence has 50 frames and each frame  $i$  of the sequence has five disks moving in the domain area; moreover, the disks do not collide with each other, only with the borders of the frame (where the collisions are elastic). This is like a 3D model of balls that, because of the initial conditions, do not collide with each other. This model is useful to test the system against occlusion (total or partial, see Fig. 7b). To complicate a little more the model, additive Gaussian noise ( $N(0, \sqrt{10})$ ) was applied to 20% of

the pixels.



(a) (b)

Figure 7: Synthetic application: (a) First frame; (b) Occlusion.

The operator has been trained to track the largest ball (the one with the finger print like pattern). Ten frames were given to train the operator and they were chosen in a way to observe basically three situations: isolated ball of interest, interaction of the balls with and without occlusion. For the frames where the ball of interest interacts with the other balls, different labels were given to each ball (the ball of interest has the same label during all the training) in order to distinguish the ball of interest from the others.

The application was done to all the 50 frames. Figure 8 shows the result of the operator for the 40th frame restricted to an application mask centered in the position of the object of interest in the last frame. Note that the operator classifies the four balls which are seen inside the mask and there are several small misclassified areas.

Figure 9a shows the result of the filtering. This result is the marker image to be used in the paradigm. Figure 9b shows the result of the segmentation after the application of Beucher-Meyer paradigm (the watershed line is enhanced in black around the ball of interest for each image). Note that the system segmented both parts of the occluded ball (the one that the system was tracking).

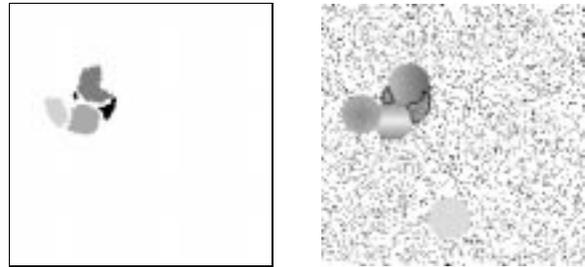
Figure 10 and Fig. 11 show part of the segmented sequence (frames 35 to 46).

### 4.3 Real sequence

In the second experiment used to test the ideas, we used the known sequence of the table tennis player [23]. Figure 12 shows part of the first image of the sequence. There are several objects that can be interesting to segment, for instance: the ball, the racket, the player's



Figure 8: Result of the restricted classification



(a) (b)

Figure 9: Synthetic application: (a) Marker; (b) Result.

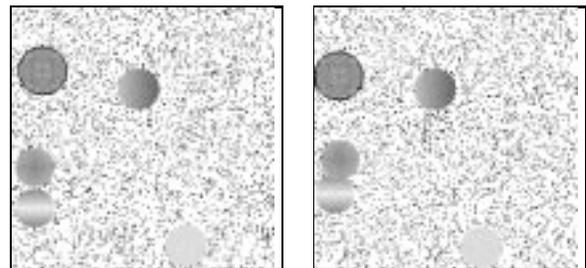


Figure 10: Application on the synthetic sequence

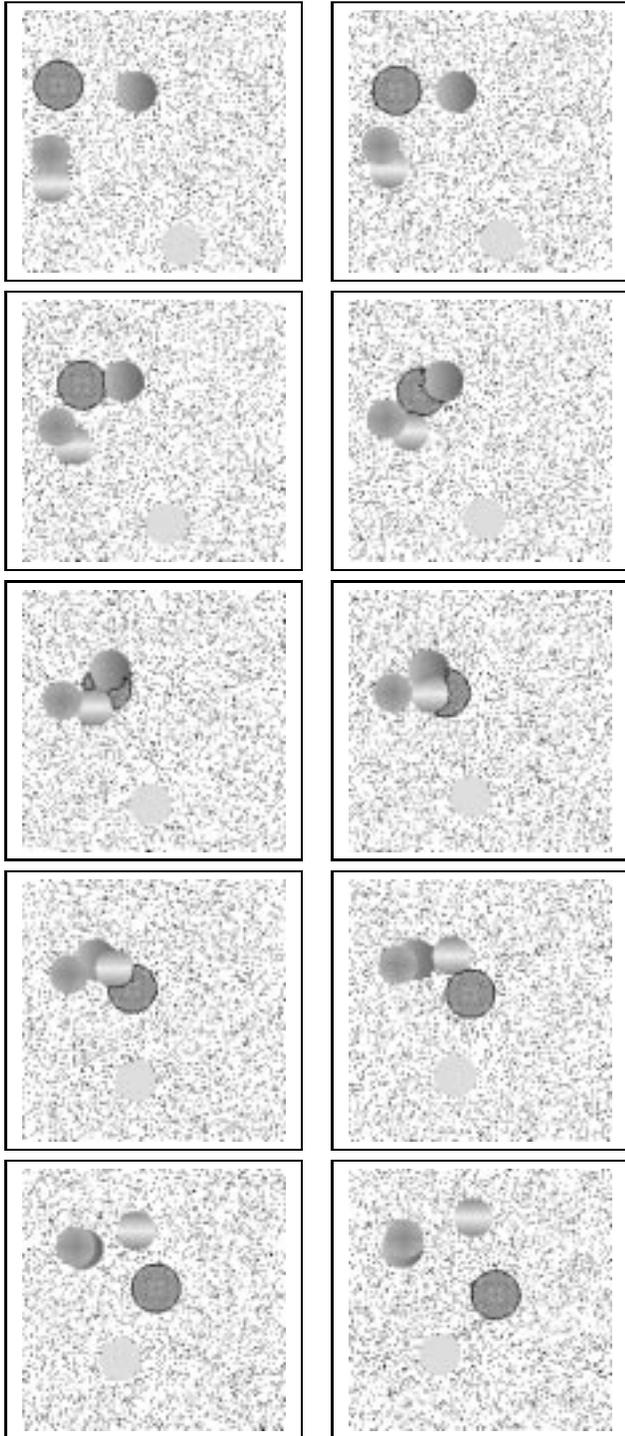


Figure 11: Application on the synthetic sequence



Figure 12: Table tennis player sequence

face, etc. We have chosen to track the ball, but more experiments will be done to track the racket, as well.

The table tennis ball is relatively easy to segment in the first 8 frames of the sequence because there it appears isolated. In the next 3 frames the ball interacts with the racket and in the last 5 frames that the ball appears in such part of the sequence (after this the ball disappears, returning much later to the scene) the form of the ball is not a disk anymore but a cylinder because the lack of temporal resolution of the movie.

In order to track the ball during that first part (16 frames) of the movie, we used 5 frames to train the operator, the first 3 frames and 2 frames where the ball interacts with the racket, one of them the ball form is a cylinder.

Figures 13 to 15 show the result of the application of the methodology for the first 9 frames. The watershed line around the ball is colored white because of the background.

## 5 Conclusion

Nowadays, the segmentation of moving objects is done manually by video editors. As digital videos are composed of hundreds or thousands of frames, this technique becomes too expensive and, consequently, almost not applied in practice. This paper proposes a new technique that may reduce significantly the cost of the



Figure 13: Part of the first 3 frames



Figure 14: Part of the frames 4 to 6



Figure 15: Part of the frames 7 to 9

operation.

The technique proposed for the segmentation of moving objects is based on Beucher-Meyer paradigm with the markers detector designed by computational learning. The objects from few frames of the video are marked manually and used to train the markers detector. Then, the operator designed is used to mark the objects in the other frames and Beucher-Meyer paradigm is applied to all frames marked.

The markers detector designed are aperture filters, conditioned to a mask around the target object. The size of this mask depends on the maximum velocity of the object in the video sequence, that is supposed known. This prior information reduces significantly the statistical complexity of the operator design problem.

In fact, before the application of Beucher-Meyer paradigm the markers were filtered by a connected filtering, that eliminates the markers of low statistical confidence. This is done by eliminating all flat zones of small area and aggregating each of them to an adjacent flat zone of maximum area.

The technique proposed has been applied to a synthetic video, that presents moving balls degraded by Gaussian noise, and to a real world video, showing a table tennis play. In both experiments, the object segmented was a ball, for which is known the position in the first frame and its maximum velocity in the video. In the first experiment, there were cases of occlusion, but not in the second. The results were quite satisfactory: 10 and 5 frames were marked manually for experiments 1 and 2, respectively, while 50 and 16 frames were segmented, respectively, in the first and in the second experiment.

We recognize that the method should be tested more intensively on real world images for a further conclusion. This will be the immediate next step of our research. We also have in mind the extension of this technique for colored videos.

### Acknowledgments

The authors acknowledge partial support from CNPq and FAPESP.

### References

- [1] Christopher Finch. *Special Effects: Creating Movie Magic*. Abbeville Press, 1984.
- [2] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Addison-Wesley Publishing Company, 1992.

- [3] W. K. Pratt. *Digital Image Processing*. John Wiley and Sons, 1991.
- [4] T. Poggio. Early Vision: from computational structure to algorithms and parallel hardware. *CVGIP*, 31(1):139–155, 1985.
- [5] F. Meyer and S. Beucher. Morphological Segmentation. *Journal of Visual Communication and Image Representation*, 1(1):21–46, September 1990.
- [6] R. Hirata Jr. Segmentação de Imagens por Morfologia Matemática. Master's thesis, Instituto de Matemática e Estatística - USP, março 1997.
- [7] S. Beucher. Watersheds of Functions and Picture Segmentation. In *ICASSP 82, Proc. IEEE Intern. Conf. on Acoustics, Speech and Signal Processing*, pages 1928–1931, Paris, May 1982.
- [8] P. Soille and L. Vincent. Determining Watersheds in Digital Pictures via Flooding Simulations. In *Visual Communications and Image Processing*, pages 240–250. SPIE, 1990. volume 1360.
- [9] L. Vincent and P. Soille. Watersheds in Digital Spaces: An Efficient Algorithm Based on Immersion Simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6):583–598, June 1991.
- [10] J. Serra. *Image Analysis and Mathematical Morphology*. Academic Press, 1982.
- [11] J. Serra. *Image Analysis and Mathematical Morphology. Volume 2: Theoretical Advances*. Academic Press, 1988.
- [12] R. Hirata Jr., E. R. Dougherty, and J. Barrera. Aperture Filters. *submitted*, 1999.
- [13] E. R. Dougherty. *Random Processes for Image and Signal Processing*. SPIE and IEEE Presses, Bellingham, 1998.
- [14] E. R. Dougherty and R. P. Loce. Optimal Mean-Absolute-Error Hit-or-Miss Filters: Morphological Representation and Estimation of the Binary Conditional Expectation. *Optical Engineering*, 32(4):815–827, April 1993.
- [15] J. Barrera, E. R. Dougherty, and N. S. Tomita. Automatic Programming of Binary Morphological Machines by Design of Statistically Optimal Operators in the Context of Computational Learning Theory. *Electronic Imaging*, 6(1):54–67, January 1997.
- [16] E. R. Dougherty. Binary Filter Estimation for Large Windows. In *SIBGRAPI-Simpósio Brasileiro de Computação Gráfica e Processamento de Imagens*, Rio de Janeiro, October 1998. IEEE Press.
- [17] R. P. Loce and E. R. Dougherty. Optimal Morphological Restoration: The Morphological Filter Mean-Absolute-Error Theorem. *Journal of Visual Communication and Image Representation*, 3(4):412–432, December 1992.
- [18] R. P. Loce and E. R. Dougherty. *Enhancement and Restoration of Digital Documents: Statistical Design of Nonlinear Algorithms*. SPIE - The International Society for Optical Engineering, Bellingham, 1997.
- [19] J. Barrera and E. R. Dougherty. Representation of Gray-Scale Windowed Operators. In Henk J.A.M. Heijmans and Jos B.T.M. Roerdink, editors, *Mathematical Morphology and its Applications to Image and Signal Processing*, volume 12 of *Computational Imaging and Vision*, pages 19–26. Kluwer Academic Publishers, Dordrecht, May 1998.
- [20] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. The Wadsworth Statistics/Probability Series. Wadsworth International Group, 1984.
- [21] T. M. Mitchell. *Machine Learning*. McGraw-Hill Series in Computer Science. McGraw-Hill, March 1997.
- [22] S. K. Murthy, S. Kasif, and S. Salzberg. A System for Induction of Oblique Decision Trees. *Journal of Artificial Intelligence Research*, 2:1–32, 1994.
- [23] P. Salembier and M. Pardàs. Hierarchical Morphological Segmentation for Image Sequence Coding. *IEEE Trans. on Image Processing*, 3(5):639–651, September 1994.