NONLINEAR EDITING BY GENERATIVE VIDEO

R.S. Jasinschi *J.M.F. Moura*

Department of Electrical and Computer Engineering
Carnegie Mellon University
Pittsburgh, PA 15213-3890

ABSTRACT

Nonlinear video editors manipulate video sequences by contents irrespective of frame order. These are computer based tools that contrast with analogue linear tape editing technologies. The latter are extremely taxing of videographers' time and resources. Current computerized editing methods represent video in terms of individual images. This poses a formidable task to the manipulation task due to the large data volumes associated with them. We discuss a framework – Generative Video, which deals with this problem in an efficient way. Generative Video represents video sequences in terms of constructs – compact models. These are world images and generative operators. World images are augmented images, which contain the non-redundant information in the video sequence, and they describe video contents information. For each independently moving object we have a different world image. World images are stratified in layers according to occlusion information. The generative operators access video contents information, such as, the shape and motion of objects moving in the sequence. nonlinear video editing is realized by applying generative operators to world images. This approach to nonlinear editing facilitates the access, storage, and manipulation of video contents information. We describe the main properties of Generative Video and demonstrate nonlinear editing on a real video sequence.

1. INTRODUCTION

Nonlinear video editing is becoming increasingly important in multimedia technology used, for example, in sports video editing and in the movie industry. It requires the processing of large amounts of video data, which has to be annotated, manipulated, compressed, stored, and transmitted. In current methods this is a difficult task, given that these methods represent video sequences by the information conveyed by individual images. Computerized video editing provides random access which represents a remarkable improvement over the current analogue tape editing technologies. These are said to be linear editors. Computer based video editing however faces enormous challenges. The large volumes of information created by digital video requires compression. On the other hand, random frame access requires that we deal with uncompressed data. In particular, if we want to edit video by its contents, for example, objects in motion, the access to uncompressed data is difficult to be accomplished using current technology. For example, video sequences compressed according to MPEG do not preserve contents information; the compressed video data is represented by, for example, DCT coefficients which are not directly correlated with the original contents information.

We discuss here Generative Video [2, 3], a framework for content-based video representation. In Generative Video nonlinear editing is made simple. In Generative Video video contents is represented by means of constructs. Constructs encode in a compact way all the non-redundant contents information in the video sequence. Constructs are world images and generative operators. World images are augmented images describing video contents information. Generative operators, such as shape templates – select image regions – or image window operator – selects individual images of the video sequence, access the video contents. In general, video contents is the information that can be identified and classified in terms of visual attributes, such as, color, depth, shape, motion, texture, or functionality. Depending on the application, the video contents are indexed in terms of different visual attributes. In this paper we focus on applications for which contents information is derived from shape and motion information.

Nonlinear editing by Generative Video deals with contents information. It is realized based on construct manipulation. More precisely, contents information is accessed and transformed by applying generative operators to world images. This means that the image, shape, and the velocity of objects moving in and out of view, and the image of the background region of 3-D scenes are processed by the application of generative operators for world images. Since world images represent, by design, the non-redundant video sequence information, their generation results in a substantial compression. Nevertheless, all content information about the video sequence is preserved in its original format, that is, in terms of its visual attributes, and not in terms of parameters in some transform domain.

Next, we present the structure of Generative Video and discuss its application to nonlinear editing. Following this, we discuss in details an experiment describing nonlinear video editing by Generative Video of a real video sequence. We finalize by describing further applications of Generative Video to nonlinear editing.

2. GENERATIVE VIDEO

Generative Video is a framework for video sequence representation. Generative Video is divided into analysis and

0-7803-3192-3/96 $5.00

synthesis phases. It involves multiple levels of abstraction:

1. Video Contents Information. Generative Video represents a video sequence in terms of constructs, that is, world images and generative operators, which encode video sequence contents.
2. Spatial Integration. Generative Video organizes the video sequence in regions according to their information contents. These regions provide a structured contents representation rather than a pixel based representation. Regions are determined by their shape and motion properties. Regions are divided into background and foreground. The background region corresponds to that part of the video sequence which is static or slowly varying. The foreground region corresponds to objects moving in and out of view. We call the latter regions 'image figures'. The goal of spatial integration is to segment the figures with respect to the image background, to determine the image figure and background velocities, and to tessellate the figures into compact shape models. Consequently the image regions are segmented and indexed by a small set of parameters.
3. Temporal Integration. Generative Video integrates video information across the sequence by generating world images and generative operators. World images are generated through cut-and-paste operations. These cut-and-paste operations are implemented by using compact shape models obtained at the spatial integration stage.

The challenge in Generative Video is to reduce the video sequence to world images and to generative operators. This is realized by the analysis phase, and it involves the spatial and temporal integration stages. The synthesis phase describes how, from the world images and generative operators, we reconstruct the input video sequence or we construct a new video sequence. Next, we describe the analysis phase.

The first part of the analysis phase describes spatial integration which determines the image background and figure velocities, figure segmentation, and figure tessellation. The second part corresponds to the determine world images and generative operators.

A. Spatial Integration. It is initialized by estimating the image velocity. This is realized, for reasons of computational complexity, on a dyadic pyramid. First, we take pairs of successive images defined at a coarse level, e.g., the Lrd level. We compute the optical flow using a gradient-based approach coupled with an affine model [4]. The resulting optical flow is propagated coarse-to-fine to the $L-1$nd level, and the images are registered using this velocity. We reapply the gradient-based method to the pairs of registered images to determine sub-pixel corrections to the propagated optical flow; this is added to the latter velocity, and the result is propagated to the $L-2$st pyramid level for which the process is repeated.

The next step is the computation of the image background velocity. This uses the velocity histogram method [4]. At each pyramid level, we determine the velocity histogram. This is obtained by counting the number of votes in the velocity space which is the 2-D space spanned by the two image velocity components. To determine the image background velocity we use the dominant motion assumption: the majority of the image pixels, say, 80%, belong to the image background. Therefore, the velocity histogram mode is assumed to correspond to the image background velocity.

Following this we segment image figures and estimate their velocity. This is done by registering the successive images of the sequence using the image background velocity. The image figures are segmented by applying to registered successive images the motion detection operator:

$$M(x,y,t) = \frac{|\frac{\partial I(x,y,t)}{\partial t}| \cdot (|\nabla I(x,y,t)|)^{\frac{1}{2}}}{(|\nabla I(x,y,t)|)^{\frac{1}{2}} + C}, \qquad (1)$$

where $\nabla I(x,y,t) \stackrel{df}{=} (\frac{\partial I(x,y,t)}{\partial x}, \frac{\partial I(x,y,t)}{\partial y})$; C is a constant necessary to avoid numerical instabilities. After this, we binarize the images obtained through $M(x,y,t)$ by using a fixed threshold for the grey-level scale. The resulting images are shown in white (255 on the grey-level scale) where there exists figure motion and in black (0 on the grey-level scale) where there occurs background motion. The figure velocity is given by temporal correlation of these binary images for all successive frames of the sequence.

Finally, the segmented figures are tessellated into rectangles. This is realized according to the Tessellation Principle [3]: it minimizes a cost function which is the sum of the square of the difference between the areas of the segmented figure and that of the sum of the rectangles, and a term proportional to the total number of rectangles. As a result of this we obtain for each figure, and at each instant k, a figure template M_k^F. This template, when applied to the image I_k, selects the figure region. The complement operator M_k^B selects from image I_k the background region.

As a result of these operations we obtain shape templates for the image figures (and for the image background), and figure and background velocities. These velocities are described by compact models for rigid translation, scaling, and rotation. Complementing this, we obtain image occlusion information which determines how image figures occlude each other and/or the image background. This completes the spatial integration stage.

B. Temporal Integration. This describes the process of obtaining world images and generative operators.

B1. World Image Generation. World images come in layers. For each figure we associate a different world image – the figure world image, and for the image background we associate the background world image. The figure and background world images are ordered in layers as determined by occlusion information; this is described by the Stratification Principle [4]. In general, world image generation is done recursively. This uses, for each figure, a set of figure templates $M_1^F, \cdots, M_N^F$, a set of background templates $M_1^B, \cdots, M_N^B$, figure and background velocities, and occlusion information. The world image Φ_r at step r is obtained by combining the information about the world image Φ_{r-1} at step $r-1$ and the rth image I_r of the sequence. The following operations are used for world image generation.

1. Initialization. It determines the first instance of the world image in terms of the first image I_1 of the sequence.
2. Registration. It aligns image regions. It consists of two parts: motion compensation and dimensional expansion.
3. Intersection. It determines the common regions between the registered Φ_{r-1} and I_r.
4. Cutting. It removes the region of intersection, determined in 3., from the registered Φ_{r-1}.
5. Pasting. It adds the result obtained through the cutting operation to the registered image I_r.

The operations *2.* to *5.* are repeated $N-1$ times, by recursively processing world image instances Φ_{r-1} and images I_r of the video sequence. At the last step, we obtain the world image $\Phi = \Phi_N$. As an example, the following algorithm describes background world image generation in the absence of figures.

Algorithm

Background World Image Generation

Task: Given the video sequence $\{I_1, \cdots, I_N\}$ of N images and the pre-computed image background translational velocities $\{(d^B_{1,x}, d^B_{1,y}), \cdots, (d^B_{N-1,x}, d^B_{N-1,y})\}$, generate the background world image Φ^B.

Recursion:

1. Initialization. For $r = 1$, $\Phi^B_1 = I_1$.
For $r = 2$ to N do:
2. Registration. Register Φ^B_{r-1} with I_r:
2.1. World image registration: $A_{1,r}\Phi^B_{r-1}$.
2.2. Current image registration: $B_r I_r$.
3. Intersection. Intersect Φ^B_{r-1} and I_r: $A_{2,r}(A_{1,r}\Phi^B_{r-1})$.
4. Cutting. Cut intersection in **3.** from **2.1.**: $[(\mathrm{I} - A_{2,r})A_{1,r}]\Phi^B_{r-1}$.
5. Pasting. Paste result of **4.** with that of **2.2.**:
$\Phi^B_r = A_r\Phi^B_{r-1} + B_r I_r$,
where $A_r \stackrel{df}{=} [(\mathrm{I} - A_{2,r})A_{1,r}]$.

Output: After N steps, output $\Phi^B_N = \Phi^B$.

The operator A_r in the algorithm describes the operations of image registration, intersection, and cutting. The operator $A_{1,r}$ registers Φ^B_{r-1} with I_r, while the operator B_r registers I_r with Φ^B_{r-1}, such that the common regions of $A_{1,r}\Phi^B_{r-1}$ and $B_r I_r$ are perfectly aligned. The operator $A_{2,r}$ determines the region of intersection of $A_{1,r}\Phi^B_{r-1}$ and $B_r I_r$.

World image generation will result in one background world image: Φ^B, and a set of $\mathcal{F}$ figure world images $\{\Phi^F_{n,k}\}$, where $n = 1, \cdots, \mathcal{F}$.

B2. Generative Operators. These comprise the image window operator W^I_k, the figure window operator W^F_k, the figure template M^F_k, the background template M^B_k, the cut-and-paste operators, and the motion operators for rigid translational, scaling, and rotational motion.

For each tessellated figure we associate a different figure world image. Given a set of N^{R^F} rectangles into which a figure F is tessellated, the figure world image is given by $\Phi^F_k = \sum_{\alpha=1}^{N^{R^F}} R^F_{\alpha,k}$. Each rectangle $R^F_{\alpha,k}$ is selected from Φ^F_k by applying to it a rectangular figure window operator $W^{R^F_\alpha}_k$, such that $R^F_{\alpha,k} = W^{R^F_\alpha}_k \Phi^F_k$. The sum of the rectangular figure window operators gives the figure window operator $W^F_k = \sum_{\alpha=1}^{N^{R^F}} W^{R^F_\alpha}_k$. W^F_k is obtained by integrating the information given by the figure templates M^F_k along time.

Individual images of the sequence are selected by the image window operator W^I_k. If $\Phi_{n,k}$ describes a stack of n figure and background world images, the image at instant k is given by $I^n_k = W^I_k \Phi_{n,k}$. W^I_k clips the kth frame of the sequence.

Temporal information is given by the motion operators. For rigid translational, scaling, and rotational motion the motion operators are applied either to the (image, figure) window operators or to world images. For example, background translational motion is represented by slidding the image window operator with respect to the world image; for the case of one figure, this is described by the transformation $I^F_k \rightarrow I^F_{k+1} = W^I_{k+1}\Phi_{1,k}$, where $W^I_{k+1} = W^I_k T$, and T is the translational motion operator. In matrix terms T is given by powers of the dislocation matrix $D \stackrel{df}{=} [\underline{e}^T_2, \cdots, \underline{e}^T_N]$, where $\underline{e}_i$ is the ith unit column vector. These powers correspond to the image background velocity components along the vertical and horizontal directions. The image background velocity is equal in magnitude to the rate at which W^I_k slides in respect to the world image, and with opposite direction. Scaling motion is given by multiresolution (pyramid) transformations on figure windows or on world images; the image window operator is maintained fixed. Finally, rotational motion is described by transformations on the figure window operator, for figure motion, and on the world images, for background rotation.

3. NONLINEAR EDITING

Nonlinear video editing by Generative Video is content-based. It is realized by applying the figure and image window operators, suplemented by motion and occlusion information, to a stack of world images. This describes video synthesis. This approach selects, manipulates, and synthesizes video contents information irrespective of the original video frame order. The contents information, such as the shape and motion of image figures, is determined at the analysis phase, as described above.

We describe nonlinear editing in terms of an experiment. A video sequence of a real 3-D outdoor scene was recorded with a camcorder fixed on a tripod. The camera pans the scene from left to right, back to left, and finally from left to right. The sequence thus generated contains 594 images with dimensions 237 × 318. Its contents describe 12 independently moving cars plus camera motion. Two images are shown in Fig. 1. In order to generate the figure and background world images we have, first, to spatially integrate the information about the video sequence. According to the previous section, this requires the operations of image background velocity estimation, image registration, figure segmentation, figure tessellation, and figure velocity estimation. In Fig. 2 we show the images for the segmented (left) and tessellated (right) car shown in the left of Fig. 1.

In Fig. 3 we display the 964 × 258 background world image for the sequence. It is generated according to an algorithm which uses the cut-and-paste operations [4] similarly to the one described in the previous section. The areas shown in black correspond to the background regions which are not panned by the camera, e.g., the small black region at the right upper corner.

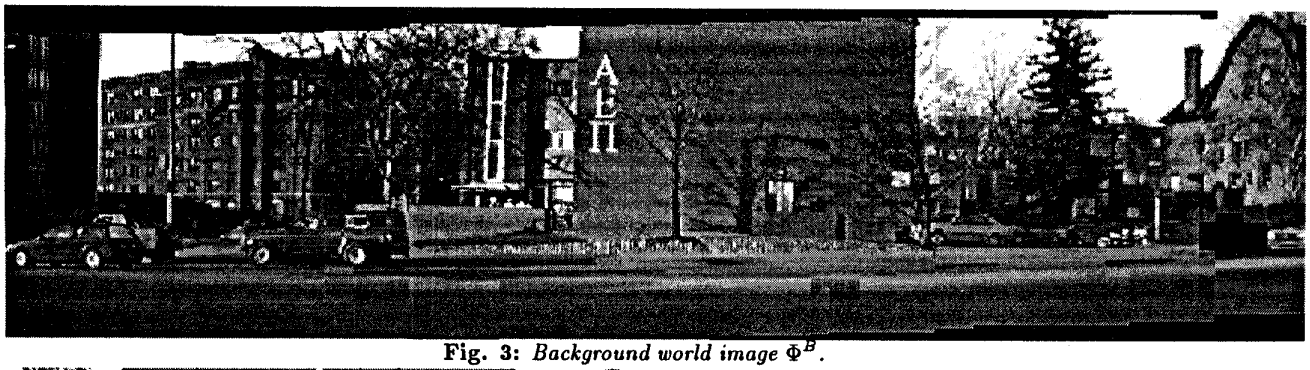

Fig. 3: *Background world image Φ^B.*

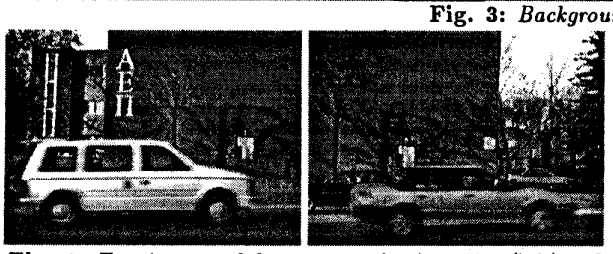

Fig. 1: *Two images of the sequence for frame 76 (left) and frame 262 (right).*

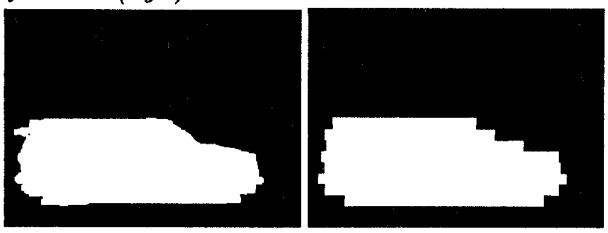

Fig. 2: *Segmented (left) and tessellated (right) car figure.*

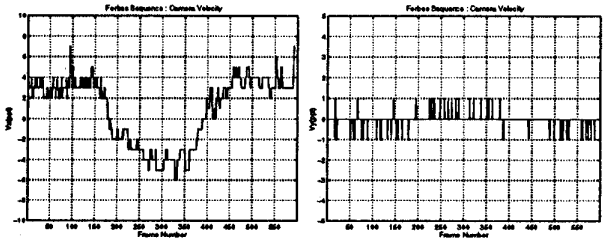

Fig. 4: *Horizontal (left) and vertical (right) image camera velocity along the 594 frames.*

Fig. 5: *Background region of frame 262 (left) and car (right) selected by applying the car shape template of Fig. 2 (right).*

We want to demonstrate nonlinear editing of this video sequence by taking the car shown on the left of Fig. 1 and inserting it in a different frame, say, corresponding to the image shown on the right of Fig. 1. This is done in a simple way.

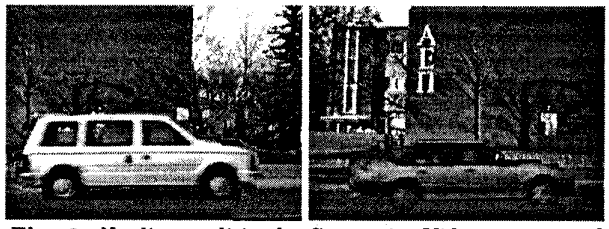

Fig. 6: *Nonlinear editing by Generative Video: cars pasted in reversed order compared to the original images (Fig. 1).*

First, we use camera motion information, as shown in Fig. 4, to select from the background world (Fig. 3) the background region corresponding to frame 262. The result is shown in the left of Fig. 5. Next, we apply the figure mask, shown in white on the right of Fig. 2., to image in the left of Fig. 1. This is shown on the right of Fig. 5. This nonlinear editing experiment is repeated for the car appearing on the right of Fig. 1. The results of these two experiments are displayed in Fig. 6.

The power of nonlinear editing by Generative Video is shown in Fig. 6. Both images do not occur in the original sequence. Also, it is not possible to generate them through frame/pixel-based editing techniques without introducing a substantial amount of artifacts. For example, to generate the image on the left of Fig. 6 through a frame/pixel-based technique we have to generate a large area for which there exists no background information.

4. CONCLUSION

Generative Video is a framework discussed here for compact video sequence representation. It is a powerful tool for nonlinear editing, given that it represents the video sequence in terms of its contents allowing for random access and manipulation of this contents in the original format.

REFERENCES

[1] E.H. Adelson and J.Y.A. Wang, "Representing moving images with layers", *M.I.T. Media Lab. Tech. Rep. N° 228*, November, 1993.

[2] R.S. Jasinschi, J.M.F. Moura, J.C. Cheng, and A. Asif, "Video Compression via Constructs", *IEEE ICASSP*, Vol. 4, 2165-2168, 1995.

[3] R.S. Jasinschi and J.M.F. Moura, "Content-based video sequence representation", *IEEE ICIP*, Vol. 2, 229-232, 1995

[4] R.S. Jasinschi, "Generative Video: a Meta Video Representation", Ph.D. Thesis, ECE-CMU, 1995.