CONTENT-BASED VIDEO SEQUENCE REPRESENTATION

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

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

ABSTRACT

The compact representation of video sequences is important for many applications, including very low bit-rate video compression and digital image libraries. We discuss here a novel approach, called Generative Video, by which video sequences are compactly represented in terms of their contents. This is achieved by reducing the video sequence to constructs. Constructs encode video sequence contents, such as, the shape and the velocity of independently moving objects, and the camera motion. Constructs are of two types: world images and generative operators. World images are augmented images incrementally generated. Generative operators, access video sequence contents and reconstruct the sequence from the world images. The reduction of a video sequence to constructs proceeds in steps. First, the shape of independently moving regions in the image is tessellated into rectangles. Second, world images are generated using the tessellated shape representation. This is described with an experiment using a real video sequence.

1. INTRODUCTION

We discuss a framework, Generative Video (GV) [2], for content-based video sequence representation. GV is accomplished by integrating the spatial and temporal information of the video sequence. Video sequence contents, from the point of view of motion information, corresponds to either coherently moving regions, which we call image figures, or to the motion of the remaining region of the image, which we call image background. Image background motion is generated by a moving camera or observer. Spatial integration generates compact shape models for the image figures, and it proceeds as follows. First, we estimate image velocity, i.e., figure and background velocity. Second, we segment the figures by using a measure to detect the figures as they move in relation to the background. Third, the segmented figures are tessellated into a simple shape models. We discuss here a rectangular figure shape tessellation model. Temporal integration generates a compact video model in terms of constructs: world images and generative operators. World images are augmented images which contain all of the non-redundant information about the video sequence. The generative operators are applied to world images and they access video sequence content. To obtain world image and generative operators, we use the figure shape templates generated by the spatial integration stage. In other words, temporal integration is based on the results of spatial integration. Therefore, the generation of GV constructs is not realized using pixel information. Instead, the constructs are obtained from the higher level information given by the figure shape models. This enables us to represent the generative operators in terms of highly structured matrices which allows for the compact and efficient access to video sequence contents and the reconstruction of the video sequence. The world images are generated incrementally by using a small set of highly structured operators, called cut-and-paste operators.

The reduction of the video sequence to constructs and the sequence reconstruction from these constructs is realized in GV in three stages: (i) pre-processing; (ii) generation; and (iii) reconstruction. The pre-processing stage deals with motion estimation, segmentation, and shape tessellation. It involves the process of spatial integration discussed above. The pre-processing computes figure and background velocities, segments figures and tessellates them in terms of rectangles. The generation stage describes the process of obtaining world images and generative operators, and it corresponds to the process of temporal integration. Finally, the reconstruction stage corresponds to the synthesis of the video sequence or parts of it, by applying generative operators to world images.

This paper is divided in two parts. First, we describe the GV framework in terms the properties of world images and generative operators. Second, we present an experiment that implements GV on a video sequence of an outdoor 3-D scene.

2. FRAMEWORK

We describe the spatial and temporal integration steps of GV. Then we discuss briefly the representation that GV uses for the sequence reconstruction and for the accessing of video sequence contents.

Spatial Integration: Shape Tessellation. In this paragraph, we assume that the segmentation stage has already been accomplished, i.e., that image (figure and background) velocities have been computed, and that each figure has been segmented[1]. Each image figure, which corresponds to an independently moving object in the world, is now tessellated into sets of rectangles according to the Tessellation Principle.

Tessellation Principle. Let A^{F_k} be the area of figure F_k.

[1] In the experimental section we discuss in detail the preprocessing operations implemented for the real video sequence.

0-8186-7310-9/95 $4.00 © 1995 IEEE

We tessellate F_k into N^R rectangles $R^F_{\alpha,k}$ by minimizing

$$E_T = [A^{F_k} - (\sum_{\alpha=1}^{N_R} A^{R^F_{\alpha,k}})]^2 + \lambda N_R, \quad (1)$$

where $A^{R^F_{\alpha,k}}$ is the area of the rectangle $R^F_{\alpha,k}$. The first term in (1) is the square of the difference between the area of the figure F_k and that of the sum of rectangles $R^F_{\alpha,k}$. The second term is proportional to the total number of rectangles N_R. The constant of proportionality λ determines the relative strength between these two terms, i.e., the trade-off between shape accuracy, determined by the first term, and complexity, determined by the second term. This criterion is similar to the Minimum Description Length (MDL) of Rissanen [3].

As a result of minimizing (1), we obtain the tessellated figure F_k in terms of a set of rectangles $\{R^F_{\alpha,k}\}_{\alpha=1,\cdots,N^R}$:

$$F_k = \sum_{\alpha=1}^{N_R} R^F_{\alpha,k}. \quad (2)$$

This describes the rectangular shape tessellation model. As a result of this, the shape of figures is described by a small set of parameters, which makes the process of world image generation and of video sequence reconstruction manageable from the point of view of its computational complexity.

Temporal Integration: World Image Generation. A World image Φ is an augmented image, e.g., a panoramic image, which is generated recursively from the original sequence by what we call cut-and-paste operations. Assume that, for a sequence $I_1, \cdots, I_N$ of N images, we know figure and camera velocities, and that figure segmentation and tessellation have been completed. The figure or background world image Φ is generated by the following recursion:

1. For $r = 1$

$$\Phi_1 = M_1 I_1. \quad (3)$$

2. For $r \geq 2$

$$\Phi_r = A_r \Phi_{r-1} + B_r(M_r I_r). \quad (4)$$

A_r is decomposed as

$$A_r \stackrel{df}{=} (\mathrm{I} - A_{2,r}) A_{1,r}, \quad (5)$$

where I is the identity operator. Φ_r represents the world image at the recursive step r, I_r is the rth image from the sequence, and M_r is the masking operator which selects from image I_r the tessellated figure or the image background region. The operators A_r and B_r perform the operations of *registration*, *intersection*, and of *cutting*.

The operators $A_{1,r}$ and B_r register Φ_r and I_{r-1} by using the information about figure and/or camera motion. Once registered, the operator $A_{2,r}$ selects from Φ_{r-1} that region which it has in common with I_r, and $\mathrm{I} - A_{2,r}$ cuts out of Φ_{r-1} this region of intersection. Finally, the resulting regions are pasted together. This algorithm is shown in Fig. 1.

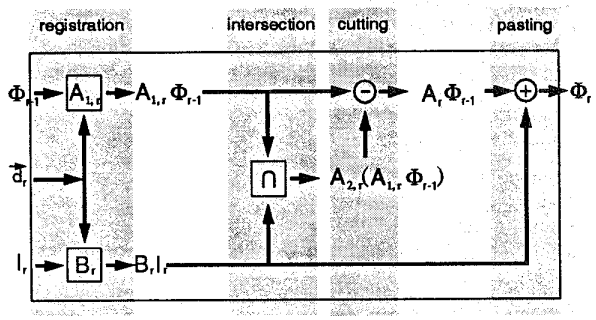

Fig. 1: *Cut and paste operations. The four operations of registration, intersection, cutting, and pasting are shown in sequence, from left to right, as they are applied to Φ_{r-1} and I_r. The motion information, denoted by $\vec{d}_r$, is precomputed. The symbol $\bigcap$ denotes the operation of intersection.*

It is important to stress out that this process of world image generation is feasible because it uses shape templates for the tessellated figures. A pixel-based approach [4] does increase the computational complexity of generating world images and it also make this process less robust.

Stratification Principle. This principle describes how world images are stratified in layers [1, 4]. For each figure we associate a different figure world image Φ^F_k. Similarly, with the image background, which distinguishes itself from the image figures by being static or slowly varying in time, we associate the background world image Φ^B_k. The Stratification Principle states that the figure and background world images are distributed in layers according to how the corresponding objects in the world are distributed at different depth levels, or, accordingly, with different velocities. This encodes the information about how the figures occlude each other or the image background.

Representation. Given that world images have been generated and that the dynamics of figure and background motion are known, we can access video sequence content and/or reconstruct the original sequence by applying generative operators to world images.

The generative operators are divided into window, motion, and signal processing operators.

Window Operators. They select rectangular regions of world images. The image window operator selects individual images of the sequence. Given a world image Φ_k, a particular image I_k of the sequence is selected by applying the image window operator W^I_k to Φ_k:

$$I_k = W^I_k \Phi_k. \quad (6)$$

Φ_k is a combination of figure and background world images. Similarly, figure window operators select rectangular regions of image figures. Using the Tessellation Principle, i.e., expression (2), we can select a rectangular figure region $R^F_{\alpha,k}$ at instant k by applying the figure window operator W^F_k to figure F_k:

$$R^F_{\alpha,k} = W^F_{\alpha,k} F_k. \quad (7)$$

This defines $W^F_{\alpha,k}$.

The image and figure window operators are decomposed into the Kronecker product of two highly structured matrices representing row and column properties. Therefore

$$W^I_k = V^I_k \bigotimes H^I_k \quad , \quad W^F_{\alpha,k} = V^F_{\alpha,k} \bigotimes H^F_{\alpha,k}, \qquad (8)$$

where $\bigotimes$ is the Kronecker product. The pre-multiplication matrices V^I_k and $V^F_{\alpha,k}$ encode image and figure row positions, respectively, and the post-multiplication matrices H^I_k and $H^F_{\alpha,k}$ encode image and figure column positions, respectively. Each row of these component matrices is given by a unit column vector $\underline{e}_i$. The *premultiplication* in the Kronecker product by a *row* $\underline{e}_i$ of V^I_k or $V^F_{\alpha,k}$ selects the ith *row* of the image I_k or of the rectangle $R^F_{\alpha,k}$, respectively. The *postmultiplication* by a *row* $\underline{e}_j$ of H^I_k or $H^F_{\alpha,k}$ selects the jth *column* of the image I_k or of the rectangle $R^F_{\alpha,k}$, respectively.

Motion Operators. They describe temporal transformations on window operators and/or world images. They generate global/regional image motion. This shows the power of representing video sequences in terms of constructs: global or regional image motion is synthesized by simple operations on window and/or world images, without the need to encode local image velocity. GV deals with rigid translational, rotational, and scaling motion. The structure of motion operators is simple and compact. Given that (6) and (7) can be condensed to

$$\Psi_k = W_k \Phi_k, \qquad (9)$$

where Ψ_k denotes an image or figure at instant k, W_k denotes an image or figure window operator, and Φ_k denotes a world image, the motion operators are represented by two types of transformations. One transformation affects only the window operator, as in translational motion, and the other transformation affects only the world image, as in rotational and scaling motion. This is described with more detail next.

Translational motion is represented by the transformation

$$W_k \rightarrow W_{k+1} = T^F_k W_k T^B_k, \qquad (10)$$

for fixed Φ_k. The translational motion operators T^F_k and T^B_k generate figure and background translational motion, respectively. It is interesting to notice the asymmetry between how T^F_k and T^B_k operate on Φ_k, i.e., they pre-multiply and post-multiply Φ_k, respectively. T^F_k and T^B_k, similarly to W^F_k and W^B_k, are decomposed as

$$T^F_k = T^F_{V,k} \bigotimes T^F_{H,k} \quad , \quad T^B_k = T^B_{V,k} \bigotimes T^B_{H,k}. \qquad (11)$$

Each component operator is given by powers of the dislocation operator D defined by

$$D = [\underline{e}_2, \underline{e}_3, \cdots, \underline{e}_{N-1}, \underline{e}_N]^T, \qquad (12)$$

where $\underline{e}_i$ is the ith unit column vector.

These powers correspond, in lattice units, to the vertical (row) and horizontal (column) image background and/or figure translational velocity.

Rotational and scaling motion are represented, for a fixed W_k, by the transformations

$$\Phi_k \rightarrow \Phi_{k+1} = R_k \Phi_k, \quad , \quad \Phi_k \rightarrow \Phi_{k+1} = S_k \Phi_k, \qquad (13)$$

respectively. The rotational motion operator R_k describes how world image elements on discrete lattices are mapped between themselves; this requires the introduction of some distortion in the position of the rotated elements. The scaling motion operator S_k corresponds to spatial multiresolution transformations, e.g., spatial pyramids. In order to represent scaling motion at different rates, we developed a model for pyramids indexed by a fractional parameter. We call these rational pyramids. The magnitude of the scaling velocity is proportional to this fractional number. The dyadic pyramid is a particular instance of a rational pyramid.

In addition to the window and motion operators, GV is described by a set of signal processing operators, e.g., spatial and temporal multiresolution operators.

3. EXPERIMENT

We describe an experiment using a real video sequence of an outdoor 3-D scene. The video sequence of thirty 240 × 256 pixels images is generated through a static handheld camera. It shows a walking person, as shown in Fig. 2.

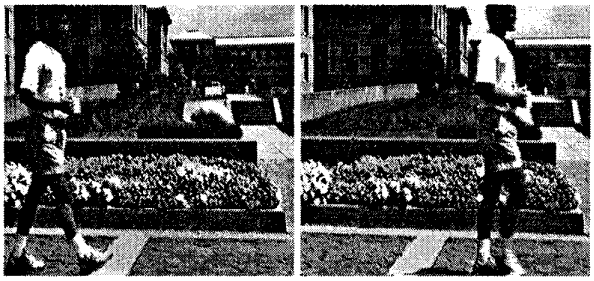

Fig. 2: *The 1st (left) and 30th (right) images of the sequence.*

The goal of this experiment is to generate two world images, one for the image background and the other one for the figure (walking person), and the translational motion operators. This requires two sets of operations: pre-processing and generation.

3.1. Pre-Processing

Pre-processing is divided into three parts.

First, we compute the image background velocity. This is realized through a gradient-based method on pyramids. In this experiment the image background velocity is approximately equal to zero, because the camera is held fixed.

Second, we segment the walking person. This requires that we compensate for the image background velocity computed before, i.e., by registering the pixels of each image according the background velocity. Following this, we apply for each consecutive pair of registered images the motion detection measure operator $M(x,y,t)$

$$M(x,y,t) = \frac{|\frac{\partial I(x,y,t)}{\partial t}| \cdot \sqrt{|\nabla I(x,y,t)|}}{|\nabla I(x,y,t)| + C}, \qquad (14)$$

where $I(\cdot,\cdot,\cdot)$ is the image intensity function, C is a constant necessary to avoid numerical instabilities, and $\frac{\partial I(x,y,t)}{\partial t}$

and $\nabla I(x,y,t)$ correspond to the temporal and spatial image gradients, respectively. $\frac{\partial I(x,y,t)}{\partial t}$ and $\nabla I(x,y,t)$ are approximated by first order differences in a $2 \times 2 \times 2$ cube in the neighborhood of each pixel. Following this operation we binarize the images by applying a threshold T ($0 \leq T \leq 255$) to each of the 29 images obtained through the motion detection measure. This allows for the robust figure segmentation. The parameters used here are $C = 5.0$ and $T = 6.0$. In Fig. 3 we show an image representing the segmented figure.

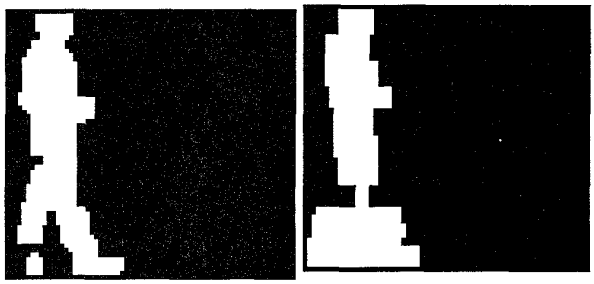

Fig. 3: *Segmented (left) and tessellated (right) image figure. The right figure describes the shape template* M_1^F.

Third, we tessellate the segmented (binary) figure images into rectangles according to (1). Figure tessellation is realized by fitting rectangles to the binary images. The tessellation was performed separately on the figure torso and feet; this is because the torso moves in horizontal rigid translational motion, and the feet move semi-rigidly. The result for the tessellation into 16 rectangles for the torso and 4 rectangles for the area representing the feet region is shown in Fig. 3.

3.2. Generation

World images are generated recursively through cut-and-paste operations, as discussed above.

We generate the background and figure world images according to (3). For $r = 1$, $\Phi_1^B = M_1^B I_1$ and $\Phi_1^F = M_1^F I_1$, where M_1^B removes from I_1 the background region, and M_1^F removes from I_1 the figure region. This is shown in Fig. 4.

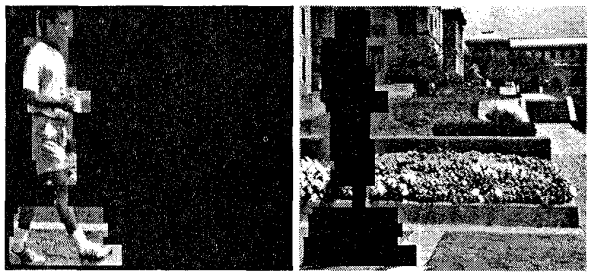

Fig. 4: *World images after one step.* $\Phi_1^F = M_1^F I_1$ *(left) and* $\Phi_1^B = M_1^B I_1$ *(right).*

Subsequently, for $r \geq 1$, we generate the background world image Φ^B by using (4) and (5) as in Fig. 1. The result of this process for $r = 5, 8, 11$ and $r = 28$ is shown in Fig. 5. The background world image corresponds to $r = 28$. In this particular experiment, the figure world image Φ^F is identical to its first instance shown in Fig. 4. (left), i.e., $\Phi^F = \Phi_1^F$. It is important to remark that (3) and (4) incorporates the solution to the occlusion problem.

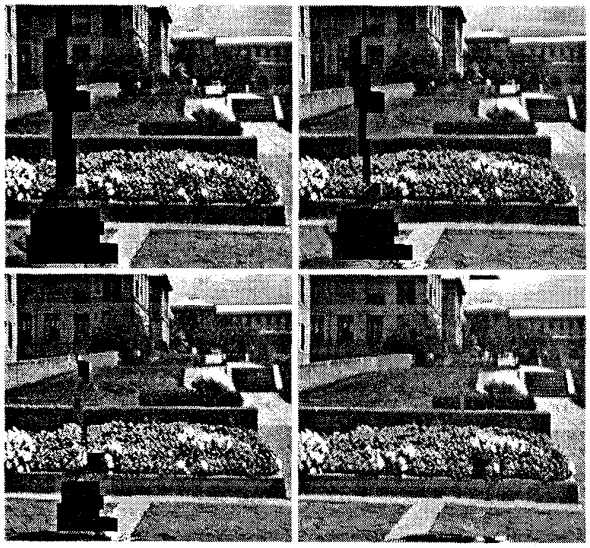

Fig. 5: *Some instances of the background world image as it is being recursively generated. In the upper row* Φ_5^B *(left) and* Φ_8^B *(right), and in the lower row* Φ_{11}^B *(left) and* $\Phi_{28}^B = \Phi^B$ *(right).*

The importance of using tessellated figure shape is shown in Fig. 5. Without this simple shape model the process of "filling in" the image background region being unoccluded by the walking person would be very unstable and it would generate many artifacts. In our framework the recursive process of world image generation using tessellated figure shape templates is robust and it describes in simple mathematical terms the problem of image occlusion.

4. CONCLUSION

We described a novel framework for content-based video sequence representation. Its main elements are world images which encode the non-redundant information on the sequence, and generative operators which are used to access video sequence contents.

5. REFERENCES

[1] T. Darrell and A. Pentland, "Robust estimation of multi-layered motion representation", *Proc. of IEEE CVPR*, 296-302, 1991.

[2] R.S. Jasinschi, J.M.F. Moura, J.C. Cheng, and A. Asif, "Video compression via constructs", *Proc. of ICASSP*, Detroit, Michigan, **Vol. 4**, 2165-2168, 1995.

[3] J. Rissanen, "Universal coding, information, prediction, and estimation," *IEEE Trans. on Information Theory*, **Vol IT-30**, 629-636, 1981.

[4] J.Y.A. Wang and E.H. Adelson, "Layered representation for motion analysis", *Proc. of IEEE CVPR*, 361-366, 1993.