VIDEO COMPRESSION VIA CONSTRUCTS

R.S. Jasinschi, J.M.F. Moura, J.C. Cheng, and A. Asif

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

ABSTRACT

Current video compression standards compress video sequences at NTSC quality with factors in the range of 10-100, like in MPEG-1 and MPEG-2. To operate beyond this range, that is, MPEG-4, radically new techniques are needed. We discuss here one such technique called Generative Video (GV). Video compression is realized in GV in two steps. First, the video sequence is reduced to constructs. Constructs are world images, corresponding to augmented images containing the non-redundant information on the sequence, and window, figure, motion, and signal processing operators, representing video sequence properties. Second, the world images are spatially compressed. The video sequence is reconstructed by applying the various operators to the decompressed world images. We apply GV to a 10sec video sequence of a real 3-D scene and obtain compression ratios of about 2260 and 4520 for two experiments done with different quantization codebooks. The reconstructed video sequence exhibits very good perceptual quality.

1. INTRODUCTION

We describe a novel approach to video compression, called Generative Video (GV) [1]. The codec structure of GV is as follows. At the encoder, the video sequence is reduced to constructs. These constructs are world images, which correspond to augmented images, and window, figure, motion, and signal processing operators. Following this, world images are spatially compressed. The error of the compressed world images and the pointers to the various operators are then transmitted. At the decoder, the video sequence is reconstructed by applying the various operators to the decompressed world images. Next, we explain each of these codec elements.

World images are augmented images representing the non-redundant information in the video sequence. Intuitively, a world image describes all that can be seen in the sequence by integrating its information along the images. For example, if we pan a camera with respect to a static 3-D scene, the corresponding world image describes the panoramic view of this scene. A world image is recursively generated through cut-and-paste operators. At each recursive step, the cut-and-paste operators combine the world image obtained at the previous step with a corresponding image of the sequence; they remove from the previous world image the region of overlap it has with the corresponding image of the sequence and paste this to the image. World images are stratified in layers [2, 3]. For each independently moving image region, which we call image figure, we associate a figure world image. For the image background, which is supposed to be static or slowly varying in time, we associate a background world image. The background and figure world images are stratified in layers according to how the corresponding objects in the world are distributed at different depth levels.

Window operators select individual images of the sequence. Window operators are applied to world images. Individual images of the sequence correspond to partial snapshots of a real or synthetic scene. This matches our intuitive idea that, e.g., as we move a camera with relation to a 3-D scene, we sample at each instant a different view represented by an image.

Figure operators represent image figures [3]. As a matter of compactness, arbitrary image figures, i.e., independently moving image regions, are tessellated into a set of rectangles which move coherently. Each of these rectangles is represented by an operator. These rectangle operators are recursively generated through a set of figure cut-and-paste operators. These figure cut-and-paste operators are defined through a set of nested transformations [3] which satisfy the conditions for the solution of the figure-background and figure-figure occlusion problems.

Motion operators represent temporal transformations of world images, window and/or figure operators. Motion operators describe how image regions move according to a given motion pattern, e.g., by translating the image window operator with relation to a world image. GV deals with regional or global translational, rotational, and scaling motion [1].

Signal processing operators include cut-and-paste operators which generate world images from video sequences, and multiresolution operators which transform world images between levels of space-time pyramids, besides other smoothing and filtering operators.

World images are spatially compressed using a non-causal Gauss Markov random field [4] approach. This approach delivers compression ratios in the range of 20-50, above by a factor of 3 of what JPEG [5] produces.

In GV, we transmit error world images and a set of pointers to the various operators described above. These pointers describe the operator parameters. The video sequence reconstruction is realized by applying the various operators, in the order determined by the motion and sig-

This work was partially supported by a grant from Bellcore/INI.

0-7803-2431-5/95 $4.00 © 1995 IEEE

nal processing operators, to the (stack) of the decompressed world images.

The main goal of this paper is to describe a complete set of results involving the processing of a video sequence of a real 3-D scene. In Section 2, we describe the theory for world images, image window operator, and motion operators for rigid image background translational motion. In Section 3, we present an experiment that illustrates GV by compressing a video sequence of a real 3-D scene. Finally, in Section 4, we summarize the paper.

2. THEORY

2.1. World Image

We consider world images associated with video sequences. Images are defined on rectangular lattices. The world image Φ_k at instant k occupies the lattice $\mathcal{L} = \{(i, j) : 1 \leq i \leq N_x^{\Phi_k}, 1 \leq j \leq N_y^{\Phi_k}\}$. The image coordinate system is oriented from left to right on the horizontal axis and from top to bottom on the vertical axis. Using lexicographic ordering, Φ_k is represented by a $(N_x^{\Phi_k} \cdot N_y^{\Phi_k}) \times 1$ vector

$$\Phi_k = [\Phi_{1,k}^T \cdots \Phi_{N_x^{\Phi_k},k}^T]^T, \tag{1}$$

where the vector collecting the intensities of the pixels in row i of Φ_k is

$$\Phi_{i,k} = [\Phi_{i,1,k} \cdots \Phi_{i,N_y^{\Phi_k},k}]^T. \tag{2}$$

2.2. Image Window Operator

An image I_k at instant k corresponds to a rectangle of dimensions $N_x^{I_k} \times N_y^{I_k}$ defined on the rectangular lattice $\mathcal{L}$. In lexicographic ordering, it is represented by a $(N_x^{I_k} \cdot N_y^{I_k}) \times 1$ column vector. The image I_k is generated from Φ_k by applying the image window operator W_k^I to Φ_k:

$$I_k = W_k^I \Phi_k. \tag{3}$$

W_k^I is a $(N_x^{I_k} \cdot N_y^{I_k}) \times (N_x^{\Phi_k} \cdot N_y^{\Phi_k})$ matrix, with $N_x^{I_k} \leq N_x^{\Phi_k}$ and $N_y^{I_k} \leq N_y^{\Phi_k}$. It is decomposed as

$$W_k^I = V_k^I \bigotimes H_k^I, \tag{4}$$

where $\bigotimes$ is the Kronecker product. The window operator components V_k^I and H_k^I are highly structured and sparse matrices, given by

$$V_k^I = [\underline{e}_{i^I}, \cdots, \underline{e}_{(i^I + N_x^{I_k} - 1)}]^T, \tag{5}$$

$$H_k^I = [\underline{e}_{j^I}, \cdots, \underline{e}_{(j^I + N_y^{I_k} - 1)}]^T, \tag{6}$$

where (i^I, j^I) is the position of the upper left image element $I_{1,1,k}$ with relation to the origin of the world image coordinate system located at the position of $\Phi_{1,1,k}$, and $\underline{e}_i$ is the unit column vector. In general, the *premultiplication* in the Kronecker product by a *row* $\underline{e}_i$ of V_k^I selects the *ith row* of the image I_k and the *postmultiplication* by a *row* $\underline{e}_j$ of H_k^I selects the *jth column* of the image I_k. Therefore, by changing the positions of the $\underline{e}_i$'s in V_k^I or H_k^I, we generate arbitrary shifts of the image window operator W_k^I with relation to the world image Φ_k. This is important for motion representation.

2.3. Motion Operator

Our framework includes motion operators for translational, rotational, and scaling motion [3]. Here we discuss the image (background) translational motion operator. This operator is represented by temporal transformations on the image window operator. These transformations generate inside the individual images global background translational motion patterns. The translational motion operator T_k is defined by

$$W_{k+1}^I = W_k^I T_k. \tag{7}$$

T_k is decomposed, similarly to W_k^I, as

$$T_k = T_{V,k} \bigotimes T_{H,k}, \tag{8}$$

where the operators $T_{V,k}$ and $T_{H,k}$ generate transformations on V_k^I and H_k^I, respectively. These component operators are given by powers of the dislocation operator D or of its transpose; these powers correspond to the magnitude of the translational motion with relation to the horizontal and vertical directions. The dislocation operator D is defined by

$$D = [\underline{e}_2, \underline{e}_3, \cdots, \underline{e}_{N-1}, \underline{e}_N, \underline{0}]^T, \tag{9}$$

where $\underline{0}$ is a zero column vector. D has the role of translating all image background rows or columns by one unit. In summary, image background motion is generated by the translation of the image window operator with relation to the background world image; its velocity is *opposite* to that of the image window operator.

2.4. World Image Generation

For reasons of conciseness we detail here only the case for which the world image is time invariant, i.e., $\Phi_k = \Phi$. World image generation is realized through cut and paste operations. Let $\{I_1, \cdots, I_N\}$ be a sequence of N images representing image (background) translational motion, as described in the previous subsection. The background world image Φ is generated by the recursion

$$\Phi_r = A_r \Phi_{r-1} + B_r I_r, \tag{10}$$

where Φ_r represents the partial description of the world image at step r after processing the first r ($1 \leq r \leq N$) images, I_r is the rth image, A_r and B_r are the cut-and-paste operators. The cut operator A_r is decomposed as

$$A_r = (\mathrm{I} - A_{2,r}) A_{1,r}, \tag{11}$$

where I is the identity matrix. The operator $A_{1,r}$ has the role of incrementing the dimensions of Φ_{r-1} by adding to it *zero* rows and columns; the resulting matrix has the same dimensions as Φ_r. The operator $\mathrm{I} - A_{2,r}$, when applied to $A_{1,r}\Phi_{r-1}$, selects the elements of Φ_{r-1} which are not present in I_r. Finally, the paste operator B_r increments I_r by the same number of *zero* rows and columns as $A_{1,r}$ increments Φ_r.

For $r = 1$, $\Phi_1 = I_1$, where, according to (3), $I_1 = W_1^I \Phi$. Next, for $r = 2$, given that $I_2 = W_2^I \Phi$ and $W_2^I = W_1^I (T_{V,2} \bigotimes T_{H,2})$, where $T_{V,2}$ is equal to $D^{d_x^I}$ or $[(D)^T]^{d_x^I}$, and $T_{H,2}$ is equal to $D^{d_y^I}$ or $[(D)^T]^{d_y^I}$, equation (10) adds up to I_2 a number of d_x^I rows and d_y^I columns of Φ_1 which

are not contained in I_2 and scale up appropriately the dimensions of I_2. This process is repeated for all N images in the sequence. The image velocity magnitudes d_x^I and d_y^I and the corresponding directions are estimated from the sequence through a coarse-to-fine Gaussian pyramid method [3]. These velocities are used to process the recursion (10).

2.5. Spatial World Image Compression

We compress world images through a non-causal Gauss Markov random field (GMRF) [4] technique. A GMRF is a random field whose variables are jointly Gaussian. We consider here quadratic GMRF's, i.e., with pairwise interaction and second-order self-interactions. For them, the joint probability distribution is given by

$$\frac{1}{Z}\exp[-\frac{(\Phi)^T P \Phi}{2\sigma^2}], \tag{12}$$

where Z is a normalization constant which does not depend on the world image Φ configurations, P is the *potential matrix*, and σ^2 is a positive constant. The potential matrix P is highly structured [6, 4].

The spatial compression of the world image Φ is done as follows. First, the elements of Φ are estimated. Second, the estimated values are quantized.

The estimated $\hat{\Phi}_{i,j}$ at pixel (i,j), according to a first order minimum mean square error prediction model, and its error $e_{i,j}$, are given by

$$\hat{\Phi}_{i,j} = \beta_v(\Phi_{i-1,j} + \Phi_{i+1,j}) + \beta_h(\Phi_{i,j-1} + \Phi_{i,j+1}), \tag{13}$$

$$e_{i,j} = \Phi_{i,j} - \hat{\Phi}_{i,j}, \tag{14}$$

respectively. We can summarize (14) and (13) by inserting (13) into (14), and by stacking the field values into column vectors. We get

$$P\Phi = e, \tag{15}$$

where e has covariance $\Sigma_e = \sigma^2 P$. The constants β_v and β_h determine the vertical and horizontal couplings between elements of Φ. The expression (15) represents a non-causal prediction model. This introduces a non-recursive structure in the estimation process. We can transform (15) such that the estimation process becomes recursive [4]. This is realized by the Cholesky factorization $P = U^T U$, given that P is positive definite, where matrix U is upper triangular. Using this decomposition in (15) we get

$$U\Phi = w, \tag{16}$$

for which the error w is now a Gaussian white noise with covariance $\Sigma_w = \sigma^2 I$, and $U^T w = e$. It can be verified that (16) is such that Φ at a given pixel (i,j) depends only on Φ defined for "future" pixels. A similar factorization of P allows Φ elements to depend just on "past" pixel values.

We use a quadtree mean removal and cascaded vector quantization (VQ) scheme to quantize the error image w. Cascaded VQ [6] has the advantage that the codebooks are much smaller than for single stage VQ, which reduces the computational complexity of codebook generation. On the other hand, cascaded VQ comes at the expense of a reduced peak SNR. This is compensated by quadtree mean removal [6].

3. EXPERIMENT

We process a video sequence of a real 3-D scene of 300 images each of 240 × 256 pixels. Two images of this sequence are shown at the top row in Fig. 1. Between each pair of images, we compute the global image velocity through a coarse-to-fine pyramid-based algorithm. The average image background velocity thus obtained is equal to 1 pixel per frame (ppf). In general, for GV, we determine at each pyramid level, regions moving independently from the background; this uses a detection measure followed by thresholding [3]. For the experiment described here, we have only image background motion, and no independently moving region. The world image generated according to (10) is shown at the top in Fig. 2. The generated world image leads to the "lossless" compression ratio

$$C_\Phi = \frac{F \cdot B \cdot N_x^I \cdot N_y^I}{B \cdot N_x^\Phi \cdot N_y^\Phi} \tag{17}$$

corresponding to the ratio of bits needed to encode the sequence over the bits needed to encode the world image.

In (17), F is the number of frames (f), B is the number of bits per pixel (bpp), and N_x^I, N_y^I, and N_x^Φ, N_y^Φ represent the image and world image dimensions in pixels (p), respectively. In our experiment, $F = 300$f, $B = 8$bpp, $N_x^I = 240$p, $N_y^I = 256$p, $N_x^\Phi = 376$p, and $N_y^\Phi = 462$p, leading to a compression ratio $C_\Phi = 106.1$.

For the cascaded VQ, we used a two-stage VQ. Codebooks of length (4+16) and (2+4) were used leading to spatial compression ratios of $C_{Intra} = 21.3$ and $C_{Intra} = 42.6$, respectively. The compounded compression ratio is defined by $C_{CR} = C_{Intra} \cdot C_\Phi$. For the spatial compression ratios $C_{Intra} = 21.3$ and $C_{Intra} = 42.6$, we obtain the compound compression ratios $C_{CR} = 2259.9$ and $C_{CR} = 4519.8$, respectively. The compressed world images are shown in the middle ($C_{CR} = 2259.9$) and bottom ($C_{CR} = 4519.8$) levels in Fig. 2. The middle and bottom rows in Fig. 1 show us two images reconstructed by applying the image window operator to the compressed world images with compound compression ratios of $C_{CR} = 2259.9$ and $C_{CR} = 4519.8$, respectively. When shown dynamically at a frame rate of 30fps the reconstructed video sequences exhibit very good perceptual quality.

4. SUMMARY

In this paper we discuss GV, a framework for video compression. In GV a video sequence is reduced to constructs which are then used to reconstruct the sequence. Through an experiment on a video sequence of a real 3-D scene we show that GV delivers compression ratios in the range of $10^3 - 10^4$. This illustrates the potential of GV to effectively deal with the problem of very low bit-rate video compression.

5. REFERENCES

[1] R. Jasinschi and J.M.F. Moura, "Generative video," Tech. report, Dept. of Elec. and Comp. Engineering, Carnegie Mellon University, July 1994.

[2] E. Adelson and J.Y.A. Wang, "Representing moving images with layers," Tech. Report 228, M.I.T. Media Lab., November 1993.

[3] R. Jasinschi and J.M.F. Moura, "Generative vision: content-based image sequence processing," submitted to ICCV-95.

[4] J. Moura and N. Balram, "Recursive structure of noncausal gauss-markov random fields," *IEEE Trans. Inform. Theory*, vol. IT-38(2), pp. 334–354, 1992.

[5] D. Le Gall, "MPEG: A video compression standard for multimedia applications," *Commun. of the ACM*, vol. 34(4), pp. 47–58, 1991.

[6] A. Asif and J.M.F. Moura, "Non-causal codec with residual quadtree cascaded vector quantization," Tech. report, Dept. of Elec. and Comp. Engineering, Carnegie Mellon University, December 1993.

Fig. 1: Reconstructed images. In the top row we have the 1st (left) and the 100th (right) images of the original video sequence. In the middle and bottom rows we show the reconstructed images at the same positions as the original images, with compression ratios of $C_{CR} = 2259.9$ (middle) and $C_{CR} = 4519.8$ (bottom).

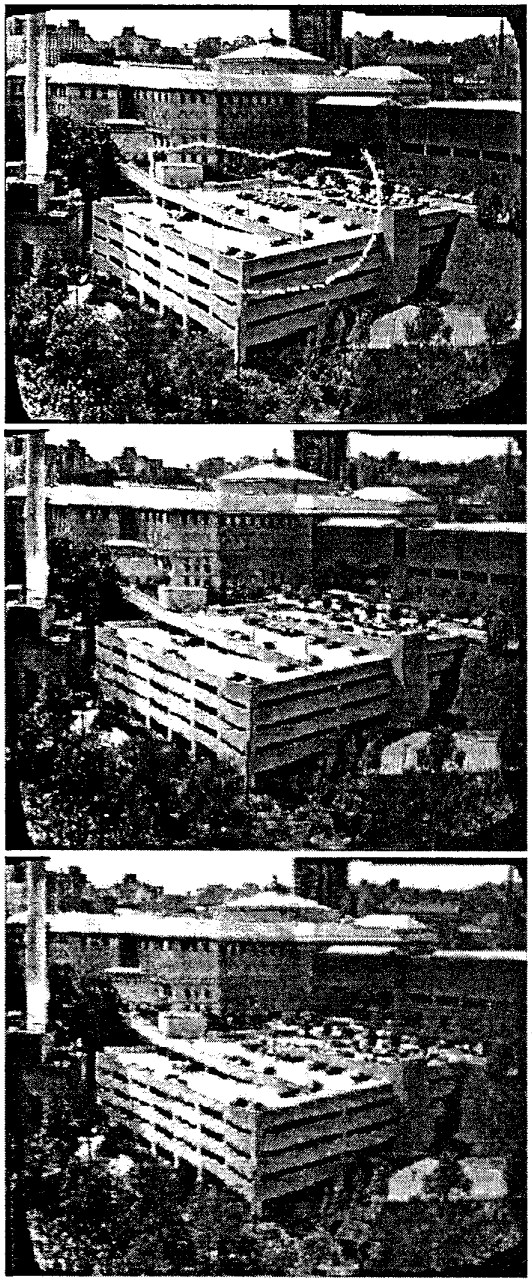

Fig. 2: World images. The top world image is generated from the original video sequence with a compression ratio $C_\Phi = 106.1$. The trajectory of the image center is shown superimposed to the world image. The middle and bottom world images correspond to the result of the spatial compression of the world image, shown at the top, with compression ratios of $C_{CR} = 2259.9$ (middle) and $C_{CR} = 4519.8$ (bottom). The border regions shown in black inside the three world images correspond to scene parts not detected by the camera.