View Synthesis for Multiview Video Compression

Emin Martinian, Alexander Behrens, Jun Xin, and Anthony Vetro
email:{martinian,jxin,avetro}@merl.com, behrens@tnt.uni-hannover.de

Mitsubishi Electric Research Labs
201 Broadway
Cambridge, MA 02139, USA

Abstract. We consider multiview video compression: the problem of jointly compressing multiple views of a scene recorded by different cameras. To take advantage of the correlation between views, we compare the performance of disparity compensated view prediction and view synthesis prediction to independent coding of all views using H.264/AVC. The proposed view synthesis prediction technique works by first synthesizing a virtual version of each view using previously encoded views and using the virtual view as a reference for predictive coding. We present experimental coding results showing that view synthesis prediction has the potential to perform significantly better than both disparity compensated view prediction and independent coding of all views.

Index Terms view synthesis, view interpolation, multiview video compression, H.264/AVC

1 Introduction

Advances in display and camera technology make recording a single scene with multiple video signals attractive. While there are many applications of such multiview video sequences including free viewpoint video [1], three dimensional displays [2, 3] and high performance imaging [4], the dramatic increase in the bandwidth of such data makes compression especially important. Consequently, there is increasing interest in exploiting the inherent correlation in multiview video through disparity compensated prediction [5, 6], mesh-based view prediction [7], wavelet transforms, and related techniques. In response to recent advances in coding technology and the emerging applications for multiview video, MPEG has recently issued a Call for Proposals on multiview video coding [8].

The main issue in any multiview compression algorithm is how to predict the signal in a given camera from one or more neighbors. Most existing multiview compression systems work without any explicit knowledge of the camera parameters and apply local or global translations to predict a given block in the target from one or more references. The advantage of this approach is that it can reuse many of the same tools as traditional temporal motion compensated prediction. But the correlations between frames captured at the same time in different cameras may be quite different from frames captured in the same camera at different times. Specifically, while block translation is a good model for predicting temporally adjacent frames, it is less accurate for predicting spatially adjacent frames because the disparity of an object in one frame relative to another depends on the distance of the object to the camera (*i.e.*, the object depth), as well as the camera and scene geometry.

As illustrated in Fig. 1, we explore whether knowing the camera and scene geometry can improve prediction (and hence compression performance) of a given camera from its neighbors. Specifically, we compare the rate-distortion performance of H.264/AVC to two different H.264/AVC multiview codecs using disparity compensated prediction and view synthesis prediction. The former adds the ability to use a previously encoded frame from other cameras as a prediction reference in addition to conventional temporal prediction. The latter adds the ability to synthesize a virtual version of the frame to be encoded from previously encoded frames of other cameras and uses the virtual frame as a prediction reference. Furthermore, in studying view synthesis prediction, we also compare performance when synthesis correction vectors are allowed to correct for slight inaccuracies in the camera parameters.

This paper is organized as follows. After describing disparity compensated prediction and the proposed view synthesis prediction tools in Section 2, we present experimental results in Section 3 and close with some concluding remarks in 4.

2 Prediction Tools

2.1 Disparity Compensated Prediction

In the following we describe the disparity compensated view prediction (DCVP) method that is used

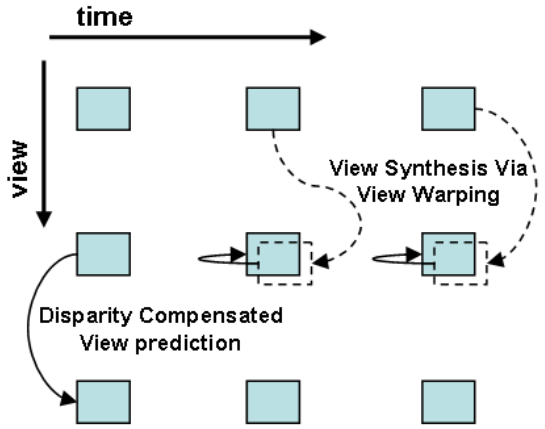

Fig. 1. Diagram of multiview prediction tools explored in this paper. In disparity compensated view prediction, a frame at time t in camera c is predicted from another frame at time t in camera c'. In view synthesis prediction, a virtual version of frame t in camera c is synthesized from frame t in camera c' using camera parameters and depth information. The synthesized frame is then used as a prediction reference.

in our system. We define $I[c,t,x,y]$ as the intensity of the pixel in camera c at time t at pixel coordinates (x,y). With conventional temporal prediction for each camera c, frame t in sequence c is typically predicted only from other frames in sequence c. With DCVP, for each c, the value of $I[c,t,x,y]$ may also be predicted from other cameras (*i.e.*, from $I[c',t,x-m_x,y-m_y]$ where (m_x,m_y) is a disparity vector computed in a blockwise manner and $I[c',t,x,y]$ is a frame from an already encoded sequence from another camera). One natural camera prediction structure is the sequential structure where $I[c,t,x,y]$ is predicted from $I[c-1,t,x,y]$, which is analogous to the IPPP Group of Pictures (GOP) structure in conventional temporal coding. Other camera prediction structures are also possible and may be better depending on the camera geometry.

2.2 View Synthesis Prediction

While DCVP provides improvements over pure temporal prediction, it does not take advantage of some essential features of multiview video. First, while temporal motion can be accurately modeled using translational motion compensation, the differences between multiple views of a scene usually cannot. For example, in moving from one camera to another the disparity in the screen pixel coordinates of an object between cameras will depend on the depth of the object. Objects closer to the camera will move much more than objects that are far from the camera. Also, effects such as rotations, zooms, or different intrinsic camera properties are often difficult to model using pure translational motion compensation. Finally, since many applications of multiview video such as 3D displays or free viewpoint video require accurate camera parameters, this information is often available at encoding time and should ideally be used to improve compression.

As illustrated in Fig. 1, we exploit these features of multiview video by synthesizing a virtual view from previously encoded views and then performing predictive coding using the synthesized views. Specifically, for each c, we first synthesize a virtual frame $\hat{I}[c,t,x,y]$ based on the on the unstructured lumigraph rendering technique of Buehler *et al.* [9] (described in more detail shortly) and then use disparity compensated view prediction as described in Section 2.1 to predictively encode the current sequence using the synthesized view.

To synthesize $\hat{I}[c,t,x,y]$, we require a depth map $D[c,t,x,y]$ that describes how far the object corresponding to pixel (x,y) is from camera c at time t, as well as an intrinsic matrix $A(c)$, rotation matrix $R(c)$, and a translation vector $T(c)$ describing the location of camera c relative to some global coordinate system. Using these quantities, we can apply the well-known pinhole camera model to project the pixel location (x,y) into world coordinates $[u,v,w]$ via

$$[u,v,w] = R(c)\cdot A^{-1}(c)\cdot[x,y,1]\cdot D[c,t,x,y] + T(c). \quad (1)$$

As a further refinement to (1), we can apply a small synthesis correction $(\alpha_{x,y},\beta_{x,y})$ to each block of original pixel locations to correct for slight inaccuracies in the camera parameters. With the synthesis correction, the world coordinates are computed as

$$[u,v,w] = R(c)\cdot A^{-1}(c) \cdot[x-\alpha_{x,y},y-\beta_{x,y},1]\cdot D[c,t,x,y] + T(c). \quad (2)$$

Next, the world coordinates are mapped into the target coordinates $[x',y',z']$ of the frame in camera c' which we wish to predict from via

$$[x',y',z'] = A(c')\cdot R^{-1}(c)\cdot\{[u,v,w]-T(c')\}. \quad (3)$$

Finally, to obtain a pixel location, the target coordinates are converted to homogeneous form $[x'/z', y'/z', 1]$ and the intensity for pixel location (x, y) in the synthesized frame is $\hat{I}[c, t, x, y] = I[c', t, x'/z', y'/z']$.

2.3 Depth Maps

An important issue in view synthesis is computing, coding, and transmitting accurate depth maps. In many scenarios such as free viewpoint video and 3D displays, such depths maps may be required as part of the application itself and can therefore be used in the compression process without requiring any extra coding overhead or computational effort. In general, however, one must both obtain the required depth maps and define a method for the encoder to convey them to the decoder.

Depth maps were not available for most of the multiview test sequences in the MPEG Call for Proposals [8]. For sequences where depth maps obtained using computer vision techniques were available (*i.e.*, the breakdancers test sequences from Microsoft Research [10]) our tests indicated that using H.264/AVC to compress the depth map at 5-10% of the total bit rate produced acceptable view synthesis performance.

In the results reported in this paper, we used a block based depth search algorithm to extract the optimal depth for two reasons. First computer vision based depth maps were unavailable for most sequences. Second, extracting depth maps for multiview compression is different from extracting depth using standard computer vision techniques. Specifically, in most computer vision based depth extraction algorithms [11], the goal is to take two images from different cameras, $I[c, t, x, y]$ and $I[c', t, x, y]$ and estimate the depth of the scene (which is unknown to the depth extraction algorithm). Performance is judged by evaluating how accurate the resulting depth map is relative to the true depth (usually obtained via direct measurement).

In contrast, for the view synthesis prediction methods explored here, the goal is to produce an accurate synthesized view $\hat{I}[c', t, x, y]$ from a reference image $I[c, t, x, y]$ and a depth map. Good performance corresponds to a low error between the synthesized and actual versions of the frame (which can be measured by the depth extraction algorithm since the image to be synthesized is available to the encoder) and the true accuracy of the depth map itself is not directly relevant. Consequently, to obtain depth maps for view synthesis prediction, we use a depth search algorithm optimized to produce accurate view synthesis.

Specifically, we define minimum, maximum, and incremental depth values $D_{\min}$, $D_{\max}$, D_{step}, and a block size D_{block}. Then, for each block of B pixels in the frame that we wish to predict, we choose the depth to minimize the error for the synthesized block:

$$D(c, t, x, y) = \min ||\hat{I}[c, t, x, y] - I[c', t, x', y']|| \quad (4)$$

where the minimization is carried out over the set

$$d = \{D_{\min}, D_{\min} + D_{\text{step}}, D_{\min} + 2D_{\text{step}}, \ldots, D_{\max}\} \quad (5)$$

and $||\hat{I}[c, t, x, y] - I[c', t, x', y']||$ denotes the average error between the block of size D_{block} centered at (x, y) in camera c at time t and the corresponding block that we are predicting from. Note, that the depth influences the error by affecting the coordinates (x', y') of the block we are predicting from.

Figures 2 and 3 present a visual comparison of the two kinds of depth maps for the breakdancers sequence. In general, the depth as computed in (4) yields a smaller error in the synthesized view (and hence a higher PSNR after compression) than depth obtained from classic methods of computer vision.

Fig. 2. Computer vision based depth map provided by Microsoft Research [10].

3 Results

Fig. 4 compares the rate-distortion performance of different multiview compression methods averaged over 250 frames (at 25 frames per second) and 8 views of the ballroom sequence. The lowest

Fig. 3. Block based depth map obtained using (4).

curve (labeled "Simulcast") represents independent coding of each view using version JM 9.5 of the H.264/AVC reference software [12] using the coding conditions from [8] summarized in Tab. 1. The next curve (labeled "View Prediction") represents the performance when using disparity compensated view prediction (DCVP) as described in Section 2.1. Evidently, DCVP provides a PSNR gain of 1-1.5 dB or a bit rate reduction of 20-30% relative to simulcast, which is consistent with previously reported results [13, 6]. The next set of curves (labeled "SV Radius 0/1/2") represent the performance of view synthesis prediction (VSP) as described in Section 2.2 with synthesis correction vectors of 0, 1, and 2 respectively.

Table 1. Coding conditions for H.264/AVC.

Feature / Tool / Setting	AVC Parameters
Rate control	Yes, basic unit=1 MB row
RD optimization	Yes
Specific settings	Loop filter, CABAC
Search range	±32
# Reference pictures	5
Temporal random access	(Open GOP) 0.5 sec
GOP Structure	IBBP...
Direct mode	Spatial
FRExt (*e.g.*, adaptive block transform)	Yes

These three curves representing VSP illustrate that efficient use of camera parameters in the form of view synthesis can provide significant improvements over DCVP. Furthermore, these curves show that while VSP improves performance relative to DCVP, to gain the full benefit of view synthesis it is essential to use a synthesis correction vector to account for slight inaccuracies in camera parameters. Specifically, using no synthesis correction vector (*i.e.*, SV Radius 0) yields a gain of only about 0.25 to 0.5 dB relative to DCVP. In contrast, using a synthesis correction vector of either $0, \pm 1, \pm 2$ in each dimension (*i.e.*, SV Radius 1) yields more than a 2 dB PSNR gain relative to DCVP and more than a 3 dB gain in PSNR or equivalently half the bit rate relative to simulcast.

Since Fig. 4 does not include the overhead required to encode the depth map and view synthesis correction vectors, these results should be viewed as upper bounds representing the potential of VSP. Still, even for small synthesis correction vectors, we believe the gains for VSP are large enough to warrant significant effort to find efficient encoding methods for depth and synthesis correction information.

4 Concluding Remarks

In this paper, we considered various methods of jointly compressing multiple views of a scene recorded by different cameras and found that a new technique we called view synthesis prediction has the potential to obtain significant gains over existing methods such as disparity compensated view prediction or independent coding of all views. Specifically, we found that (when overhead for depth maps and synthesis correction vectors was ignored), the proposed view synthesis prediction can achieve PSNR gains of more than 3 dB relative to independent coding of all views and almost 2 dB more than disparity compensated view prediction.

References

1. Smolic, A., Kauff, P.: Interactive 3-d video representation and coding technologies. Proceedings of the IEEE **93** (2005) 31–36
2. Tanimoto, M.: FTV (free viewpoint television) creating ray-based image engineering. In: International Conference on Image Processing. Volume 2. (2005) 25–28
3. Dodgson, N.A.: Autostereoscopic 3d displays. IEEE Computer **38** (2005) 31–36
4. Wilburn, B., Joshi, N., Vaish, V., Talvala, E.V., Antunez, E., Barth, A., Adams, A., Horowitz, M., Levoy, M.: High performance imaging using large camera arrays. ACM Transactions on Graphics **24** (2005) 765–776
5. Girod, B., Chang, C.L., Ramanathan, P., Zhu, X.: Light field compression using disparity-compensated lifting. In: International Conference

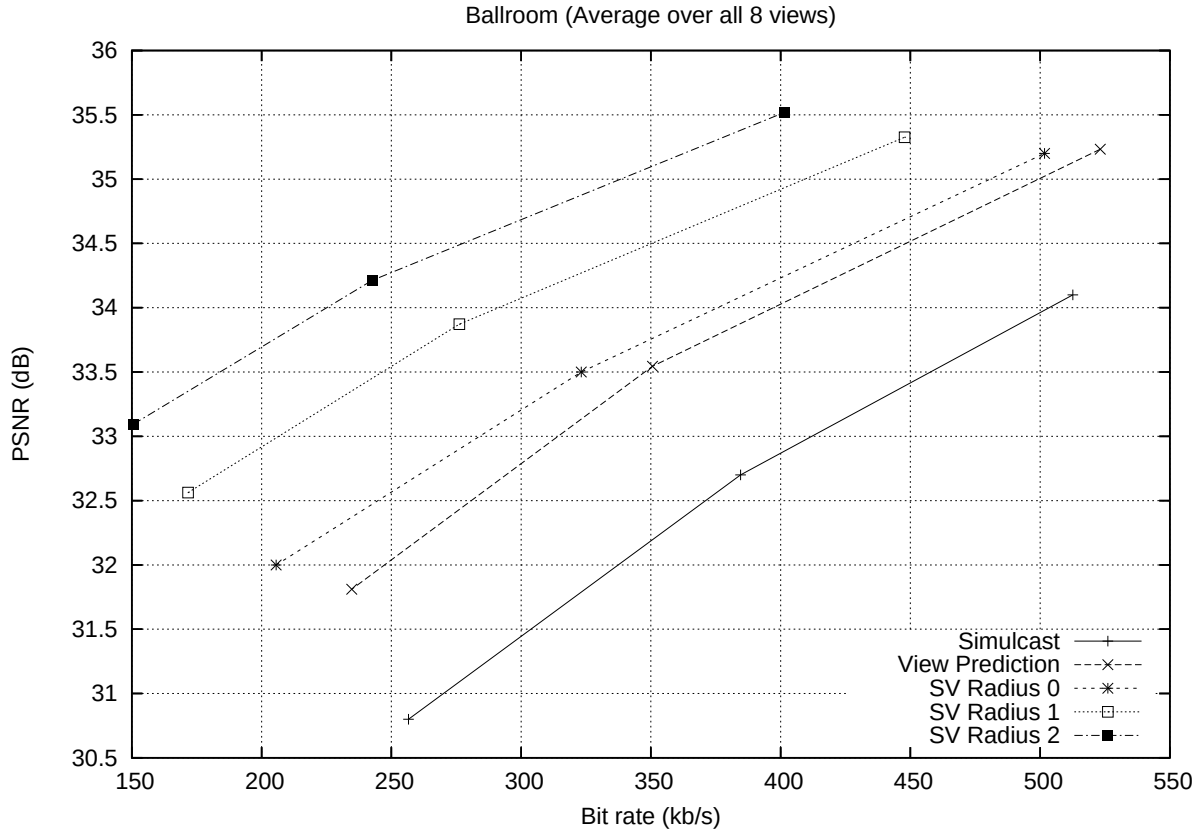

Fig. 4. Multiview compression results for the ballroom sequence from the MPEG Call for Proposals on Multiview Video Coding [8]. The bit rates for "View Prediction" and "SV Radius 0/1/2" do not include the rate required for depth maps or synthesis correction vectors.

on Acoustics Speech and Signal Processing. Volume 4. (2003) 760–763

6. Chan, S.C., Ng, K.T., Gan, Z.F., Chan, K.L., Shum, H.Y.: The compression of simplified dynamic light fields. In: International Conference on Acoustics Speech and Signal Processing. (2003) 653–656
7. Wang, R.S., Wang, Y.: Multiview video sequence analysis, compression, and virtual viewpoint synthesis. IEEE Transactions On Circuits and Systems for Video Technology **10** (2000) 397–410
8. ISO/IEC JTC1/SC29/WG11: Updated call for proposals on multi-view video coding. MPEG Document N7567 (2005) Nice, France.
9. Buehler, C., Bosse, M., McMillan, L., Gortler, S., Cohen, M.: Unstructured lumigraph rendering. In: Proceedings of ACM SIGGRAPH. (2001) 425–432
10. Zitnick, C.L., Kang, S.B., Uyttendaele, M., Winder, S., Szeliski, R.: High-quality video view interpolation using a layered representation. In: Proceedings of ACM SIGGRAPH. (2004) 600–608
11. Scharstein, D., Szeliski, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. International Journal of Computer Vision (2002) 7–42
12. MPEG: H.264/AVC reference software JM 9.5. (http://iphome.hhi.de/suehring/tml/doc/lenc/html/)
13. ISO/IEC JTC1/SC29/WG11: Report of the subjective quality evaluation for multi-view coding CfE. MPEG output document N6999 (2005)