MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

Depth Estimation for View Synthesis in Multiview Video Coding

Serdar Ince, Emin Martinian, Sehoon Yea, Anthony Vetro

TR2007-025 June 2007

Abstract

The compression of multiview video in an end-to-end 3D system is required to reduce the amount of visual information. Since multiple cameras usually have a common field of view, high compression ratios can be achieved if both the temporal and inter-view redundancy are exploited. View synthesis prediction is a new coding tool for multiview video that essentially generates virtual views of a scene using images from neighboring cameras and estimated depth values. In this work, we consider depth estimation for view synthesis in multiview video encoding. We focus on generating smooth and accurate depth maps, which can be efficiently coded. We present several improvements to the reference block-based depth estimation approach and demonstrate that the proposed method of depth estimation is not only efficient for view synthesis prediction, but also produces depth maps that require much fewer bits to code.

3DTV Conference 2007

DEPTH ESTIMATION FOR VIEW SYNTHESIS IN MULTIVIEW VIDEO CODING

*Serdar Ince**, *Emin Martinian, Sehoon Yea and Anthony Vetro*

Mitsubishi Electric Research Laboratories
201 Broadway, Cambridge, MA 02139

ABSTRACT

The compression of multiview video in an end-to-end 3D system is required to reduce the amount of visual information. Since multiple cameras usually have a common field of view, high compression ratios can be achieved if both the temporal and inter-view redundancy are exploited. View synthesis prediction is a new coding tool for multiview video that essentially generates virtual views of a scene using images from neighboring cameras and estimated depth values. In this work, we consider depth estimation for view synthesis in multiview video encoding. We focus on generating smooth and accurate depth maps, which can be efficiently coded. We present several improvements to the reference block-based depth estimation approach and demonstrate that the proposed method of depth estimation is not only efficient for view synthesis prediction, but also produces depth maps that require much fewer bits to code.

Index Terms— Depth estimation, view synthesis, multiview coding, regularization

1. INTRODUCTION

Emerging camera arrays [1] and eye-wear free 3D displays [2, 3] make 3D TV a feasible product in the future. In an end-to-end 3D system, the transmission and storage of multiple video streams is a concern because of the prohibitive amount of visual data. In response to this need, there is currently an MPEG activity on efficient coding of multiview video [4, 5].

One of the approaches in multiview coding is to use view synthesis to produce additional references for the view that is being encoded [6, 7]. Consider Fig. 1 where we would like to code $I_n(t)$, a frame at time t of camera n. As shown in Fig. 1, it is possible to use previous frames, such as $I_n(t-1)$, as references. Also, since the cameras share a common field of view, it is possible to use frames $I_{n-1}(t)$ and $I_{n+1}(t)$ of neighboring cameras as references as well. Moreover, by using view synthesis, it is possible to reconstruct a virtual view $V_n(t)$ for camera n using other cameras. Martinian *et al.* [6, 7] showed

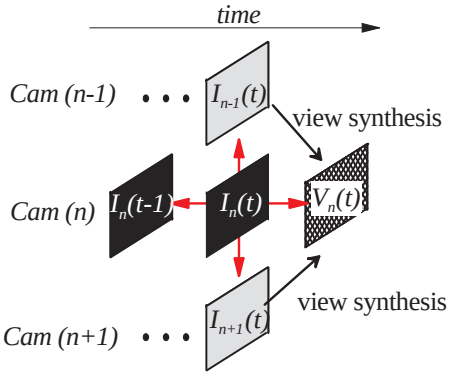


Fig. 1. Prediction using view synthesis in multiview coding

that using this synthesized view as an additional reference can introduce notable gains in compression efficiency.

Among many methods to synthesize a view, one approach is to compute the depth of the scene using available cameras and then to use this depth map to render virtual views [8, 9]. However, in the case of multiview video coding, one crucial step is the transmission of these maps, because they will be used by the decoder. In most cases, the depth of the scene is unavailable and must be extracted. Therefore, when depth maps are computed, the number of bits required to represent them must be considered as well [10, 11]. Depth maps for multiview coding should be smooth enough so that they can be coded efficiently, but they should also have enough variations to approximate the scene structure. Considering these needs, in this paper we focus on improving a block-based depth estimation tool to generate smooth and accurate depth maps. We progressively improve the results by introducing a hierarchical scheme, regularization and nonlinear filtering. We also extend the search into color components. These additional steps not only improve the smoothness of depth maps, but also lead to visual improvements in synthesized frames.

The paper is organized as follows: First, we introduce the depth estimation and then explain the improvements to the algorithm in detail. Finally, we show the efficacy of the resulting depth maps in view synthesis and multiview coding.

*S. Ince is with Boston University Department of Electrical and Computer Engineering Boston, MA 02215. He was an intern at MERL. Email: ince@bu.edu, emin@alum.mit.edu,[yea,avetro]@merl.com

2. DEPTH ESTIMATION FOR VIEW SYNTHESIS

Let A_n, R_n and t_n denote intrinsic matrix, rotation matrix and translation vector of camera C_n. Given a point $\mathbf{x}_n = [x_n, y_n]$ on image I_n captured by C_n and corresponding depth of the point $D(\mathbf{x}_n)$, it is possible to map $\mathbf{x}_n$ onto other cameras I_i where $i \in \{1 \dots N\}$, N being the number of cameras. First, the point is projected into three-dimensional space from two-dimensional image plane as follows:

$$X = R_n \cdot A_n^{-1} \cdot [\mathbf{x}_n\, 1]^T \cdot D(\mathbf{x}_n) + t_n \tag{1}$$

where X denotes the three-dimensional point. Next, X can be projected onto desired camera, for example I_{n-1}, as follows:

$$\mathbf{x}_{n-1} = A_{n-1} \cdot R_{n-1}^{-1} \cdot (X - t_{n-1}). \tag{2}$$

Combining these two equations we can write $\mathbf{x}_{n-1}$ as a function of $\mathbf{x}_n$ and $D(\mathbf{x}_n)$ within a scaling factor:

$$\begin{aligned}&\mathbf{x}_{n-1}(\mathbf{x}_n, D(\mathbf{x}_n)) = \\ &\quad A_{n-1}R_{n-1}^{-1}(R_n A_n^{-1}[\mathbf{x}_n\, 1]^T D(\mathbf{x}_n) + t_n - t_{n-1})\end{aligned} \tag{3}$$

Using (3), a depth estimation method seeks to minimize the following prediction error among possible depth values:

$$P(\mathbf{x}) = \Psi(I_n[\mathbf{x}_n] - I_{n-1}[\mathbf{x}_{n-1}(\mathbf{x}_n, D(\mathbf{x}_n))]) \tag{4}$$

where Ψ is an error function, for example quadratic or absolute value function. As a minimization problem this can be written as follows:

$$D(\mathbf{x}) = \underset{D_i(\mathbf{x})}{\operatorname{argmin}}\, P(\mathbf{x}) \tag{5}$$

where $D_i(\mathbf{x}) = D_{min} + iD_{step}, i = \{0 \dots (D_{max} - D_{min})/K\}$, K is the number of possible depth values.

If we consider a block-based model, then the frame I_n is divided into $M \times M$ blocks and prediction error in equation (4) is minimized for each block. Since this cost function computes the minimum prediction error, it is an excellent choice for minimizing prediction residual. However, the resulting depth maps are not suitable for a multiview codec. The problem with this depth estimation is that the resulting depth maps are usually very noisy and lack spatial smoothness. One frame from *Ballroom* sequence and its corresponding depth map estimated using 4×4 blocks are shown in Fig. 2.a and b respectively. Brighter pixels indicate the points that are far away from the camera while darker pixels indicate points that are close to the camera. Due to the lack of spatial and temporal correlation of these depth maps, conventional compression algorithms fail to achieve a high quality reconstruction while keeping the depth bitrate low. Moreover, the estimated depth values do not accurately represent the scene. It is clear that smoother depth maps are essential to achieve high compression ratios and accuracy of depth values.

3. IMPROVEMENTS ON DEPTH ESTIMATION

In this section, we progressively improve the results of the block-based depth estimation algorithm.

3.1. Hierarchical Estimation

In the example we show in Fig. 2.b, a block size of 4×4 is used to approximate the scene structure. However, carrying only 16 pixels of information, such a block fails to capture the local texture which will help to find a good match. Larger blocks tend to give better matches. On the other hand, larger blocks cannot define the local depth variations. Resulting depth maps by using large blocks will be over smoothed and blocky. A hierarchical estimation scheme is a good fit to solve both problems. The algorithm should start from a large block size so that a reasonable, but coarse, depth is estimated and then these values should be used as initial values and refined by smaller block sizes. Specifically, we start with 16×16 blocks, and refine the results using 8×8 and then 4×4 blocks. Results for each step of hierarchy are shown in Fig. 2.c-e. Notice that after successive steps, the depth map provides a better representation of objects in the scene. When compared to the original estimation (Fig 2.b), immediate improvements are visible in Fig. 2.e, especially in the background. However, this depth map still contains too much variation to be compressed efficiently.

3.2. Regularization

Regularization, a common tool in many inverse problems, introduces *a priori* knowledge to the problem [12]. In depth estimation, it can be assumed that neighboring points should have similar depth values because objects are rigid in the real world. Such a constraint is not enforced in equation (5), which yields noisy depth maps. Therefore in order to enforce regularization during depth estimation, we introduce a new term that penalizes when a block $\mathbf{x}$ has a different depth value than the neighboring points:

$$R(\mathbf{x}) = \sum_{k \in \Pi} \Psi(D(\mathbf{x}) - D(\mathbf{x}_k)) \tag{6}$$

where Π indicates the neighborhood of the current block and Ψ is an error function. We used second order neighborhood (eight surrounding neighbors) in the implementation, and absolute value function for Ψ. Combining the prediction error term in equation (4) and the new regularization term, we perform the following minimization:

$$D(\mathbf{x}) = \underset{D_i(\mathbf{x})}{\operatorname{argmin}}\, P(\mathbf{x}) + \lambda R(\mathbf{x}) \tag{7}$$

where λ is the regularization (smoothness) parameter. Large values of λ results in smoother depth maps. However, it should be noted that increasing λ to very large values yields over

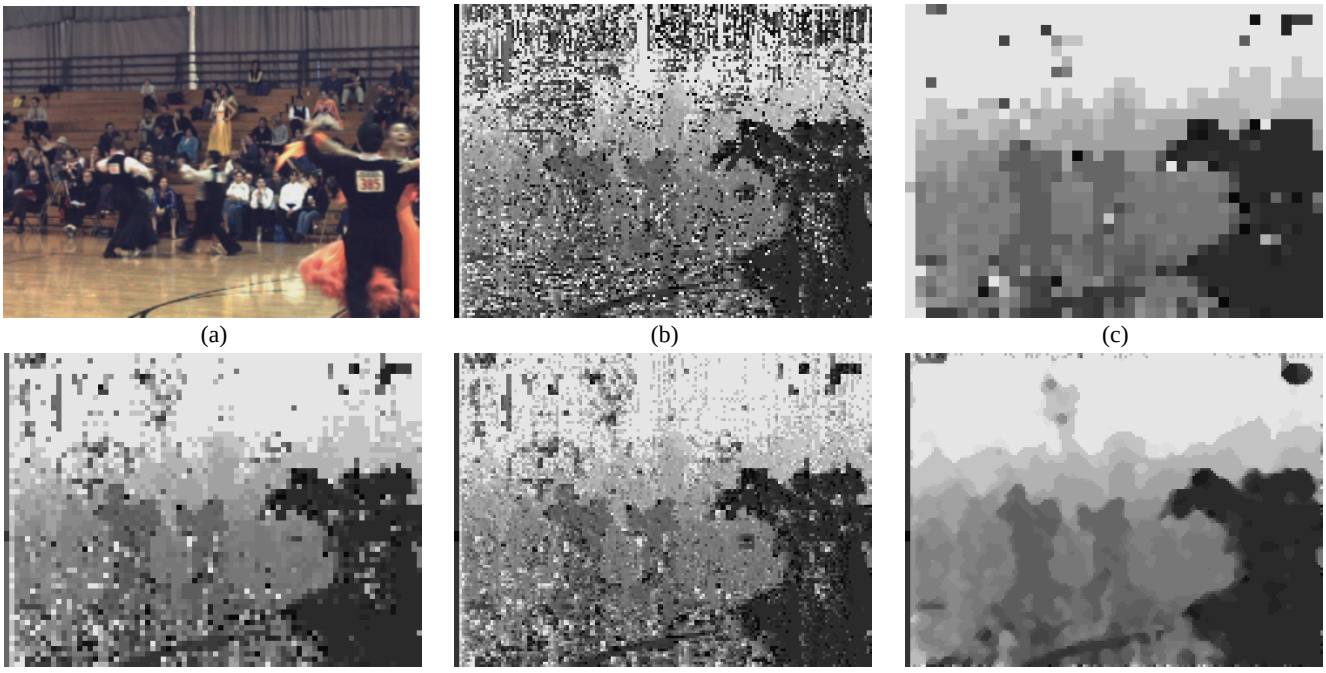


Fig. 2. Visual comparison of depth maps. (a) View #4, Frame #1 of *Ballroom* sequence. (b) Result of original block-based depth estimation. (c)-(e) Results of hierarchical scheme for each level, $16 \times 16, 8 \times 8, 4 \times 4$ respectively. (f) Final result of improved depth estimation algorithm. Clearly, improved algorithm generates smoother and more accurate depth maps.

smoothed depth maps which are as unusable as the unregularized ones. Therefore for best results, regularization parameter may need to be adjusted for different sequences. This is a common drawback of regularized methods.

3.3. Median filtering

Despite two previous steps which aim to achieve smooth depth maps, there may be still outliers in the depth map. The well-known median filter is a basic nonlinear filter used to suppress outliers in a data set. Median filtering is added to the algorithm as a post-processing step. Once a depth map is computed in each hierarchy level, the median filter is applied to eliminate the outliers. We used a fixed window size of 3×3 for median filtering.

4. COMPARISON OF DEPTH MAPS

The final depth map after improvements described in the previous section is shown in Fig. 2.f. It is clear that the resulting depth maps are much smoother, which should be easier to compress than the noisy depth maps in Fig. 2.b. We also observe that subjectively, the depth values are closer to the real depth.

As mentioned earlier, the smoother depth maps can be compressed more efficiently than noisy depth maps. To verify this claim, we tested the synthesized image quality *vs.* the bitrate required to encode the depth maps by using H.264/AVC reference software [13] on *Ballroom* sequence. Results are shown in Fig. 3. These results show that the new algorithm outperforms the original algorithm by up to 6dB. The original algorithm is better only at very high (and impractical) bitrates of 3 Mbits/sec.

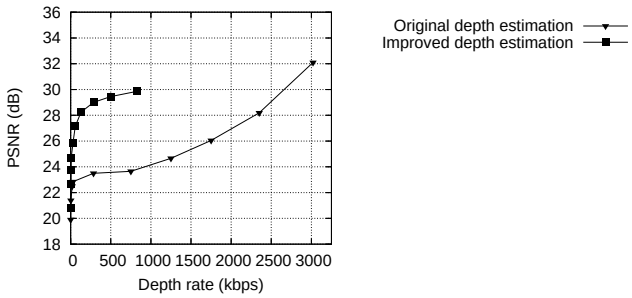
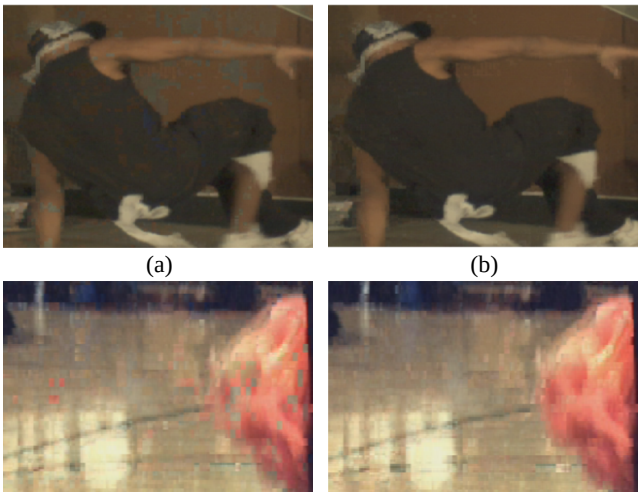


Fig. 3. Rate of depth *vs.* synthesized view quality.

Finally, we tested the synthesized frames generated by original and improved depth maps in the multiview codec described in [6]. Since bit-rate for depth was omitted in that study, we focus on the decoded image quality. Compared to results using depth maps obtained by reference block-based algorithm, we observed approximately the same PSNR using the new depth maps with less than 3% increase in the bit rate. This slight loss in prediction efficiency is expected due to the smoothness constraints imposed by the new algorithm. However, it should be kept in mind that the rate to code the new depth maps will be significantly less.



(a) (b) (c) (d)

Fig. 4. Synthesis results (a,c) without and (b,d) with using YUV search. (*Breakdancers* is courtesy of Microsoft.)

5. IMPROVEMENTS ON VISUAL QUALITY

For the sake of simplicity, usually only one color component, luminance, is used in depth estimation. However, two different textures in an image, especially areas with a smooth color, may have comparable luminance values. Due to this, depth estimation may yield incorrect depth values which in turn results in visual artifacts as shown in Fig. 4.a and c (Refer to electronic version of this paper for better quality). Therefore whether regularized or not, extension of depth estimation methods to include color components will contribute to improved visual quality of the synthesized view. Once minimization in equation (7) is carried on luminance and chrominance components jointly, such artifacts are significantly reduced as shown in Fig. 4.b and d.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we considered the estimation of smooth and reliable depth maps for view synthesis-based multiview coding. By adding several improvements, we showed these depth maps improve both compression efficiency and visual quality.

Further improvements in the depth estimation might be achieved by using variable block sizes instead of fixed sizes [14]. Synthesis correction vectors [7] can improve the results. Computation of possible depth values could also be reduced. Currently, the algorithm uses a fixed number of possible depth values and it linearly samples the depth range. Obviously, the number of possible depth values directly affects the synthesized image quality and depth maps. Therefore, a mechanism to adjust the depth range depending on available bandwidth may be considered. Moreover, linearly sampling the depth may not be always effective to approximate the scene. For example, objects closer to the camera will have more visible depth variations than far away objects, but possible depth values may not cover all structural details of this closer object and this may lead to artifacts in synthesized view. Visually, artifacts on objects closer to camera will have more degrading effects. So, nonlinear sampling of depth with emphasis on small depth values can be considered.

7. REFERENCES

[1] B. Wilburn *et al.*, "High performance imaging using large camera arrays," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 765–776, 2005.

[2] N.A. Dodgson, "Autostereoscopic 3D displays," *Computer*, vol. 38, no. 8, pp. 31–36, 2005.

[3] W. Matusik and H. Pfister, "3D TV: A scalable system for real-time acquistion, transmission and autostereoscopic display of dynamic scenes," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 814–824, Aug. 2004.

[4] "Updated call for proposals on multi-view video coding," MPEG Document N7567, Oct. 2005, Nice, France.

[5] A. Vetro *et al.*, "Coding approaches for end-to-end 3D TV systems," in *Picture Coding Symposium*, 2004.

[6] E. Martinian, A. Behrens, J. Xin, A. Vetro, and H. Sun, "Extensions of H.264/AVC for multiview video compression," in *IEEE Int. Conf. Image Processing*, 2006.

[7] E. Martinian, A. Behrens, J. Xin, and A. Vetro, "View synthesis for multiview video compression," in *Picture Coding Symposium*, 2006.

[8] S. E. Chen and L. Williams, "View interpolation for image synthesis," in *SIGGRAPH*, 1993, pp. 279–288.

[9] C. Buehler *et al.*, "Unstructured lumigraph rendering," in *SIGGRAPH*, 2001, pp. 425–432.

[10] A. A Alatan and L. Onural, "Estimation of depth fields suitable for video compression based on 3-D structure and motion of objects," *IEEE Trans. Image Process.*, vol. 7, no. 6, pp. 904–908, June 1998.

[11] J. Park and H. Park, "A mesh-based disparity representation method for view interpolation and stereo image compression," *IEEE Trans. Image Process.*, vol. 15, no. 7, pp. 1751–1762, 2006.

[12] W. C. Karl, "Regularization in image restoration and reconstruction," in *Handbook of Image and Video Processing*, A. Bovik, Ed. Academic Press, 2005.

[13] "H.264/AVC JM Reference Software," `http://iphome.hhi.de/suehring/tml`.

[14] A. Mancini and J. Konrad, "Robust quadtree-based disparity estimation for the reconstruction of intermediate stereoscopic imagessss," in *SPIE Stereoscopic Displays and Virtual Reality Systems*, 1998, pp. 53–64.