SIFT-BASED MODELING AND CODING OF BACKGROUND SCENES FOR MULTIVIEW SOCCER VIDEO

Haopeng Li and Markus Flierl

School of Electrical Engineering
KTH Royal Institute of Technology, Stockholm
{haopeng, mflierl}@kth.se

ABSTRACT

This paper presents a content-adaptive modeling and coding scheme for static multiview background scenes of soccer games. We discuss a content-adaptive modeling approach for static multiview background imagery that is based on piecewise geometric models of the content. We propose an approach that uses the Scale Invariant Feature Transform (SIFT) to extract the parameters of the geometric models. Moreover, a content-adaptive rendering approach is presented for handling occlusion problems in large baseline scenarios. The experimental results show that our content-adaptive modeling and coding scheme outperforms conventional DIBR schemes.

Index Terms**—** Immersive networked experience, content-adaptive modeling and rendering, SIFT features.

1. INTRODUCTION

In recent years, content-based coding techniques have been considered for efficient video coding [1]. In our earlier work [2], we proposed a content-adaptive coding scheme for immersive networked experience of soccer games. The content-adaptive coding scheme extracts from an input image sequence several sub-sequences depending on the static (e.g., soccer stadium) and dynamic (e.g., player) content of the input. Then, each sub-sequence is encoded according to optimally allocated bitrates among static and dynamic content items. This scheme provides not only improved rate-distortion performance but also more flexibility by allowing users to access content items easily.

To enable a free-viewpoint experience, Depth Image Based Rendering (DIBR) is a widely used technique which utilizes one or more reference texture images and their associated depth images to synthesize virtual camera views [3]. Consider an example of multiview static content captured by four high-definition cameras as shown in Fig. 2. A DIBR approach to virtual view synthesis uses dense depth images, which are usually generated by pixelwise matching [4]. However, due to the large baseline scenario for soccer video (usually more than 10 meters horizontal distance between each camera) and occlusion problems, the accuracy of generated depth imagery is relatively low and the computational complexity is high. Additionally, conventional pixelwise depth estimation approaches usually utilize pairs of reference views instead of considering multiple views jointly. This leads to poor global consistency among all reference views.

To overcome the disadvantages of pixelwise approaches, piecewise planar models are often used to model man-made urban objects by approximating them with piecewise planar structures [5]. Such approaches provide fast and visually superior result by simplifying the geometric structure of objects. However, applying such methods to non-planar objects, like the grandstand with spectators in Fig. 2, will decrease the rendering quality significantly.

In this paper, we discuss a content-adaptive modeling and coding scheme for static multiview background scenes from a free-viewpoint rendering perspective. We divide the entire background scene into multiple content items and model each item with one piecewise planar or non-planar model. For an efficient selection of planar and non-planar models, we propose an approach to estimate the model parameters that uses the Scale Invariant Feature Transform (SIFT) [6]. With that, a free-viewpoint experience can be realized by knowing a small number of geometric parameters instead of dense depth images. This approach will also be advantageous for coding and transmission. Additionally, as we consider large baseline scenarios, our content-adaptive rendering technique prevents potential occlusion problems and improves overall rendering quality.

2. CONTENT-ADAPTIVE MODELLING AND CODING

The static content captured by an array of static cameras in a soccer stadium, comprising mostly of areas depicting the field and the background objects, is varying slowly over time and its piecewise structure can be utilized to efficiently generate geometric models for virtual view rendering purposes.

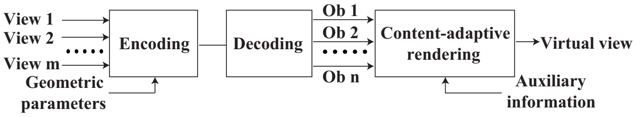
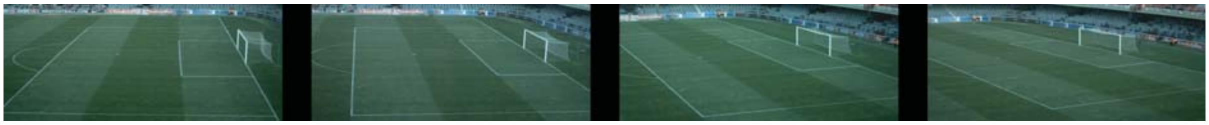


Fig. 1. Content-adaptive coding and rendering scheme.

The content-adaptive coding and rendering scheme, as shown in Fig. 1, comprises the encoding of multiple views and geometric parameters of piecewise models, and a rendering unit at the decoder that facilitates the reconstruction of the virtual view video. The content-adaptive rendering approach will offer an additional advantage when using auxiliary information for handling occlusions.

2.1. Piecewise Geometric Models for Background Scenes

In our work, the entire background scene is divided into multiple content items. We use a 3D piecewise geometric model to characterize the geometry of each item. We choose a heuristic method to find an approximation by allowing both planar and non-planar models. This approach offers three advantages: First, by using a



(a) Camera 1 **(b) Camera 2** **(c) Camera 3** **(d) Camera 4**

Fig. 2. Multiview background scenes in a large baseline scenario.

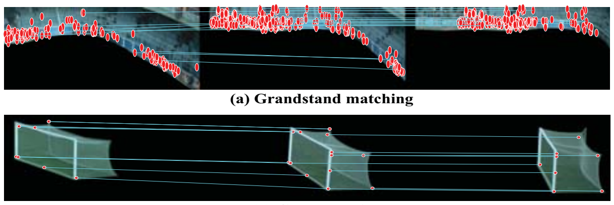
piecewise model, the smoothness and texture continuity of the background content guaranties better visual experience. Second, combining both planar and non-planar models avoids an over-simplification of the appearance of objects and allows a trade-off between rendering quality and simplicity of the model. Third, any required depth image can be generated easily by projecting the globally consistent 3D piecewise model onto a 2D image plane. Thus, the cost of transmitting view-dependent depth imagery is saved.

2.2. Parameter Estimation via SIFT Features

The multiview imagery of the background will help us to find the 3D piecewise geometric model. We propose an approach that uses SIFT features to estimate the parameters of the 3D model. In particular, the SIFT features will be exploited to establish feature correspondences between multiple observations of each content item.

2.2.1. SIFT Feature Matching and Refinement

First, we extract SIFT features in the multiview imagery and find correct correspondences. As they relate to the same 3D point in the scene, we use them to estimate the parameters of the 3D model. We extract and match SIFT features from adjacent observations. An example is depicted in Fig. 3.



(a) Grandstand matching

(b) Goal matching

Fig. 3. Matching SIFT features among multiple observations of grandstand and goal.

Let $p_i^l \leftrightarrow p_j^r$ be a feature correspondence between two references I^l and I^r, where p_i^l denotes the i-th feature point with the image coordinate (x_i^l, y_i^l) in the view I_l and p_j^r the j-th feature point with the image coordinate (x_j^r, y_j^r) in the view I_r. Knowing the camera calibration parameters, we are able to check the correctness of feature matches based on a geometric constraint. If it is a correct correspondence, p_i^l and p_j^r are originally projected from the same 3D world coordinate. Thus, by using projection relations for corresponding points, the following relations hold [3]:

$$R_l^{-1} \cdot A_l^{-1} \cdot [x_i^l, y_i^l, 1]^T \cdot d_l + \begin{pmatrix} C_l^x \\ C_l^y \\ C_l^z \end{pmatrix} = \begin{pmatrix} X_i^l \\ Y_i^l \\ Z_i^l \end{pmatrix}, \qquad (1)$$

$$R_r^{-1} \cdot A_r^{-1} \cdot [x_j^r, y_j^r, 1]^T \cdot d_r + \begin{pmatrix} C_r^x \\ C_r^y \\ C_r^z \end{pmatrix} = \begin{pmatrix} X_j^r \\ Y_j^r \\ Z_j^r \end{pmatrix}, \qquad (2)$$

where $[X, Y, Z]^T$ is the 3D world coordinate, and where R, A and C are the camera calibration parameters which depend on the camera position. The factors d_l and d_r in (1) and (2) define the position of the 3D point on the rays, known as depth. To determine the depth, let the third row of the 3×3 matrix $R_l^{-1} \cdot A_l^{-1}$ be $[\alpha_l, \beta_l, \gamma_l]$. Thus, the factor d_l is given by

$$d_l(Z_i^l) = \frac{Z_i^l - C_l^z}{\alpha_l x_i^l + \beta_l y_i^l + \gamma_l}. \qquad (3)$$

Similarly for the factor d_r. Therefore, the depth d_l and d_r are a function of the world coordinate Z.

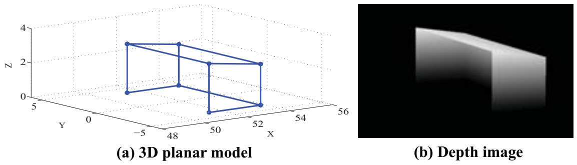
With the scaling factors d_l and d_r, (1) and (2) need to be equal for corresponding points $p_i^l \leftrightarrow p_j^r$. As we assume to know the true camera calibration parameters, the resulting expression is overdetermined. For our practical application, we determine the least square error solution of Z^* according to

$$\begin{aligned} Z^* \; = \; & \arg\min_Z \| R_l^{-1} \cdot A_l^{-1} \cdot \begin{pmatrix} x_i^l \\ y_i^l \\ 1 \end{pmatrix} \cdot d_l(Z) + \begin{pmatrix} C_l^x \\ C_l^y \\ C_l^z \end{pmatrix} \\ & - R_r^{-1} \cdot A_r^{-1} \cdot \begin{pmatrix} x_j^r \\ y_j^r \\ 1 \end{pmatrix} \cdot d_r(Z) - \begin{pmatrix} C_r^x \\ C_r^y \\ C_r^z \end{pmatrix} \|_2. \qquad (4) \end{aligned}$$

The two resulting 3D world coordinates $[X, Y, Z]_i^l$ and $[X, Y, Z]_j^r$ are obtained by the least square error solution (4) with respect to p_i^l and p_j^r. However, some small misalignment caused by calibration parameters should also be considered. Thus, we use an additional criterion. If $\|[X, Y, Z]_i^l - [X, Y, Z]_j^r\|_2 < \delta_d$, where δ_d is a small threshold for the Euclidean distance in 3D space, the correctness of the correspondence $p_i^l \leftrightarrow p_j^r$ is sufficiently reliable.

2.2.2. Geometric Parameters from SIFT Features

Now, we use the set of accurate 3D features to estimate the parameters of the 3D model. For content with planar structure, such as goals, poles, and flags, we use a single or an assembly of planar models. A planar model is easily defined by the 3D positions of the corner points. Therefore, we defined those as the parameters of a planar model. An example for an assembly of 3D planar models to characterize the structure of a goal is shown in Fig. 4.



(a) 3D planar model **(b) Depth image**

Fig. 4. Planar model of a goal as obtained by SIFT features.

On the other hand, for non-planar structured content such as the grandstand, 3D piecewise non-planar models are used. We define a

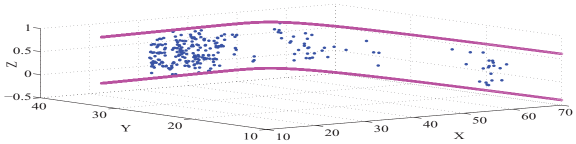


Fig. 5. Non-planar model of the grandstand. Blue circles indicate 3D SIFT features, red lines indicate fitted model.

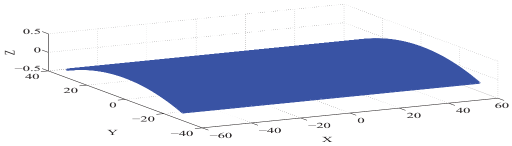


Fig. 6. Non-planar model of the field.

set Q_s for the 3D SIFT features of the grandstand, as depicted by blue circles in Fig. 5. We assume that the model plane is perpendicular to the field plane at $Z = 0$. In this case, the model plane is specified by fitting the $[X, Y]$ components of the 3D features with an exponential curve

$$\min \sum_{X(i),Y(i)\in Q_s} \|Y(i) - a \cdot e^{b\cdot X(i)} - c \cdot e^{d\cdot X(i)}\|_2, \qquad (5)$$

where a, b, c, d are parameters of the model. They are determined by the least square error solution and will specify the model for the grandstand. In Fig. 5, the model is depicted by red lines.

Additionally, SIFT features can be used efficiently to choose between planar and non-planar models due to their accuracy. One example is the modeling of the soccer field. Intuitively, we can use a plane to model the field. However, by investigating the differences of the Z components of the 3D features, the field appears non-planar. More precisely, the center of the field is usually about 30 cm above the borderline of the field. In practice, we choose a cylindrical surface to model the field, where the X axis is parallel to the generatrix of the cylinder.

$$\min \sum_{Y(i),Z(i)\in Q_f} \|Y(i)^2 + (Z(i) - c + r)^2 - r^2\|_2, \qquad (6)$$

where Q_f is the set of features extracted from the field, r is the radius and c is the Z-intercept. The parameters are obtained by the least square error solution. The model of the field is shown in Fig. 6.

2.3. Coding of Parameters

After modeling the static background by a piecewise geometric model, the model parameters are encoded with high accuracy such that the model can be reconstructed at the decoder. For our background content, we use 48 parameters in total. Considering the need for accuracy and the small number of parameters, we choose a double-precision representation for each parameter. The resulting 8 bytes per parameter incur a small cost compared to the overall bitrate. Similar to [7], we use arithmetic coding to encode the entire parameter set without any loss.

2.4. Content-Adaptive Rendering for Large Camera Baselines

After decoding the multiview sequences, virtual views may be rendered by warping reference images to target viewpoints by using DIBR. However, distances between adjacent cameras are usually very large for our application. Therefore, handling occlusion problems and adjusting lighting conditions turn out to be two major challenges.

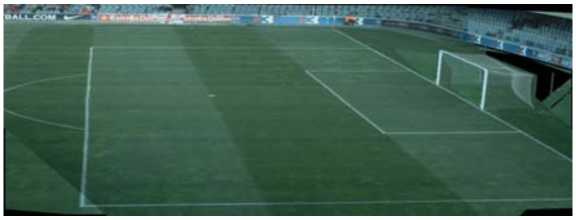
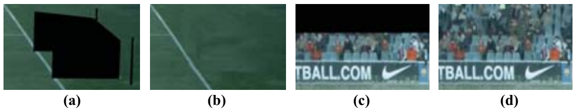


Fig. 7. Occlusions and lighting differences in a virtual view.



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

Fig. 8. Compensation of reference content by inpainting; black areas in (a) and (c) are occlusions caused by content extraction or potential occlusion; (b) and (d) show occlusion-compensated images.

Large distances between adjacent cameras may cause large occluded areas in the virtual viewpoint. For our application, we observe that up to 10% of the pixels in an image may be affected. This problem can be partially solved by warping multiple references to the target view. However, the large distances between the cameras is still limiting as shown in Fig. 7. Note the areas behind the goal and the borders of the image.

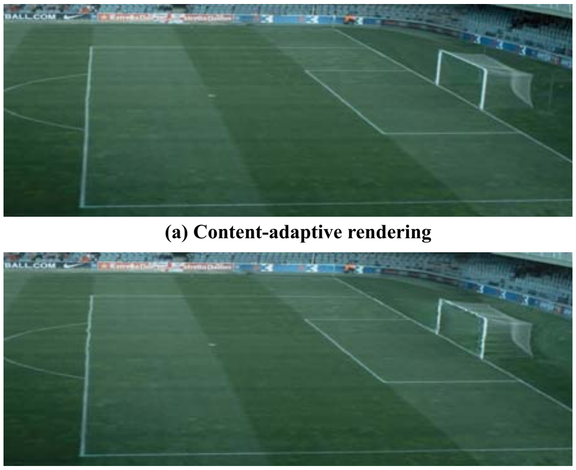
To approach this problem, we consider a content-adaptive rendering method. As discussed in Section 2.1, for each content item of the static background, we have one corresponding 3D piecewise model. By projecting the 3D model onto the reference image plane, a 2D mask for each content item is obtained. The content item in the reference image is extracted according to this 2D mask and the occlusions are compensated by auxiliary information based on inpainting [8]. An example is shown in Fig. 8(b). At the border of the image, we use exemplar-based inpainting, as shown in Fig. 8(d). With this method, we reduce the interference between content items.

Content-adaptive warping can be accomplished for each available reference view. As we have multiple references, we are able to improve the quality of the target view by merging. In our implementation, available content from the closest reference is prioritized.

In our large baseline scenario, lighting conditions vary significantly. This can be observed in Fig. 7. We adjust the intensity when merging different references by using fast linear intensity adjustment methods [9]. With that, we balance computational complexity and visual quality.

3. EXPERIMENTAL RESULTS

We evaluate our content-adaptive modeling and coding scheme with the soccer test video set *Barca-St. Andreu*, which is provided by the MEDIAPRO group. The videos are captured by four fixed broadcast cameras, as shown in Fig. 2. The resolution of the videos is 1080 × 1920 at 25 fps. The average Y-PSNR between the reconstructed view and the corresponding original camera view will be used to evaluate the performance of our scheme. We use 100 successive frames from the test sequence. H.264/AVC encoding and decoding is accomplished by the x264 implementation [10].



(a) Content-adaptive rendering

(b) Conventional DIBR

Fig. 9. Subjective comparison between content-adaptive rendering and conventional DIBR.

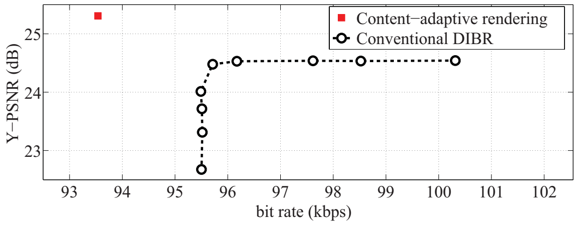


Fig. 10. Performance comparison between our content-adaptive modeling and coding and conventional texture plus depth coding with DIBR.

We compare our scheme to conventional texture plus depth coding with DIBR. We encode multiple textures from cameras 1, 3 and 4 with H.246/AVC in a simulcast setting at high quality for both schemes. For our scheme, we encode the parameters of the 3D piecewise model by arithmetic coding. For the reference scheme, the depth sequences of cameras 1, 3 and 4 are encoded at various bitrates using H.264/AVC simulcast. As discussed in our earlier work [2], we reduce the frame rate of the static background to 1 fps to optimize the overall rate-distortion performance of the content-adaptive coding scheme. Thus, we encode both texture and depth of the background sequences at 1 fps. For a fair comparison, the depth images are generated by our 3D piecewise model. In our scheme, we use decoded texture images plus decoded model parameters to synthesize the texture images of camera 2. In our reference scheme, the texture images of camera 2 are synthesized by decoded texture and depth images.

Fig. 9 shows a subjective comparison of one frame. Using the piecewise geometric model, the continuity and smoothness of the content items is preserved. On the other hand, the coded depth images of our reference scheme cause a visible degradation of the rendered image. As shown in Fig. 10, our content-adaptive modeling and coding scheme also outperforms conventional DIBR in an objective comparison. The camera baseline is 10m for this data set. Hence, the varying lighting conditions are challenging and affect the average Y-PSNR. Note that we require only 248 bits to code our model parameters.

4. CONCLUSIONS

We discussed a content-adaptive modeling and coding scheme for static background scenes to enable a free-viewpoint experience of soccer video. We use piecewise geometric models to characterize the structure of content items in the static background and utilize SIFT features to estimate the model parameters. Further, a content-adaptive rendering approach is introduced to realize a free-viewpoint experience in a large camera baseline setting. This approach handles occlusion and lighting problems more efficiently than conventional DIBR. The experimental results show that our content-adaptive modeling and coding scheme outperforms conventional texture plus depth coding with DIBR for both subjective and objective evaluation.

5. ACKNOWLEDGMENTS

This work was supported in part by the European Commission in the context of the project ICT-FP7-248020 “FINE – Free-Viewpoint Immersive Networked Experience”.

6. REFERENCES

[1] P. Ndjiki-Nya, T. Hinz, A. Smolic, and T. Wiegand, “A generic and automatic content-based approach for improved H.264/MPEG4-AVC video coding,” in *Proc. of the IEEE International Conference on Image Processing*, Sept. 2005.

[2] H. Li and M. Flierl, “Rate-distortion-optimized content-adaptive coding for immersive networked experience of sports events,” in *Proc. of the IEEE International Conference on Image Processing*, Sept. 2011.

[3] Y. Mori, N. Fukushima, T. Fujii, and M. Tanimoto, “View generation with 3d warping using depth information for FTV,” *3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video*, pp. 229–232, May 2008.

[4] D. Scharstein and R. Szeliski, “A taxonomy and evaluation of dense two-frame stereo correspondence algorithms,” *International Journal of Computer Vision*, pp. 7–42, Apr. 2002.

[5] B. Micusik and J. Kosecka, “Piecewise planar city 3D modeling from street view panoramic sequences,” in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, July 2009.

[6] D. Lowe, “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision*, vol. 60(2), pp. 91–110, 2004.

[7] A. Dasu, S. Raghavan, N. Raghavendra, and S. Panchanathan, “Arithmetic precision for perspective transform in sprite decoding of MPEG-4,” in *Proc. SPIE Media Processors*, Jan. 2000, vol. 3970.

[8] A. Criminisi, P. Perez, and K. Toyama, “Region filling and object removal by exemplar-based image inpainting,” *IEEE Trans. on Image Processing*, vol. 13, pp. 1200 –1212, 2004.

[9] D. Milgram, “Computer methods for creating photomosaics,” *IEEE Trans. on Computers*, vol. C-24, pp. 1113 – 1119, 1975.

[10] L. Aimar, L. Merritt, E. Petit, M. Chen, J. Clay, M. Rullgard, R. Czyz, Ch. Heine, A. Izvorski, A. Wright, and J. Garrett-Glaser, “X264 – A free H264/AVC encoder,” http:// www. videolan. org/ developers/ x264.html, Jan. 2011.