MVA2002 IAPR Workshop on Machine Vision Applications, Dec. 11-13, 2002, Nara-ken New Public Hall, Nara, Japan

10—3

Compression Performances of Computer Vision Based Coding

Franck Galpin *, Luce Morin † and Koichiro Deguchi ‡
Graduate School of Information Sciences, Tohoku University,
Aoba-campus 01, Sendai 980-8579, Japan
Telephone: 022-217-7014; Fax: 022-217-7015.

Abstract

This paper presents new results in the field of very low bitrate coding and compression using 3D informations. Contrary to prior art in model-based coding where 3D models have to be known, the 3D models are automatically computed from the original video sequence. The camera parameters and the scene content are supposed unknown and the video sequence is processed on the fly. A stream of 3D models is then extracted and compressed, using adapted compression techniques. We finally show results of the proposed compression scheme, and show the efficiency of these approach.

1 Introduction

Instead of representing video sequence as a set of pixels, like in classical video coding algorithms, 3D model based coding aims at representing the video sequence with one or severals textured 3D models and a set of camera/objects parameters. This topic has been addressed for many years, particularly in the field of visio-conference where a 3D model of the human face is used to represent the video sequence of the speaker [1]. In this paper, we address the problem of the representation of unknown static scenes viewed by a monocular moving camera.

The 3D-based coding has several advantages compared to image based coding: first it allows very low bitrate coding since the only informations to transmit are the camera parameters, the 3D model of the scene (if it is unknown on the decoder side) and corresponding texture image mapped on this 3D model. Moreover, with such a coding, we can benefit of 3D features such as viewpoint transformation, illumination change with respect to geometry or synthetic 3D-model coding. However many works [2] [3] are based on known 3D-models, and few results are available concerning 3D-model based video compression with unknown 3D-models. On the other hand, computer vision techniques, and particularly shape from motion techniques allow to extract 3D-models of a static scene viewed by a moving camera [4] [5] [6]. However, the shape-from-motion process still is a difficult problem without simplifying assumption on the scene or the camera parameters. Another solution is to take into account a very large amount of data among the video sequence in order to recover the whole 3D structure of the scene [7] [8].

In this paper, we present an automatic scheme for 3D-model based video compression of static scene viewed by a moving camera. Video coding framework brings some particular constraints: first we must consider that camera parameters (both extrinsic and intrinsic) and scene content are unknown. We must also take into account that video coding requires an on-the-fly process: we must process only a small part of the video sequence, not waiting for all data to process the video sequence. We then have chosen to based our process on the extraction of a stream of 3D models, instead of a unique 3D model which requires too much informations in the case of video coding. Extraction of the 3D models is briefly described, and we pay more attention on the compression stage.

2 3D models stream generation

We first present the structure of the representation: each 3D model is extracted and used for a small portion of the video sequence called GOP (Group Of Picture). The GOPs overlap themselves, that is the last image of a GOP is the same as the first image of the next one. We call these images *Keyframes*. For each GOP, a 3D model is automatically extracted on the fly. The principle of extraction is based on shape-from-motion method: we use a dense mesh-based motion estimator using multi-grid and multi-resolution approaches [11]. The camera intrinsic parameters are estimated or fixed to approximated values. The extrinsic parameters are computed using classical calibration methods and an adapted bundle adjustment method [9]. The dense motion field from the first to the last image of the GOP and camera parameters for these two images allow to reconstructed a dense depth map of the first image of the GOP. A uniform triangular mesh is then applied on each first image of each GOPs which gives a 3D model and its associated texture image (the first image of the GOP). Camera extrinsic parameters are retrieved for each image of the video sequence. The original video sequence reconstruction is then realized by the projection of each 3D textured models. A 3D fading is also used to smoothly switch from one GOP to the next one [9]. One must must notice that the decoding stage uses classical 3D algorithms and is realized in real time.

3 Stream compression

Once the original video sequence has been encoded as a 3D model stream, we want to compressed it. The

*Graduate School of Information Sciences, Tohoku University, E-mail: galpin@fractal.is.tohoku.ac.jp

†Irisa/INRIA Rennes, Campus de Beaulieu, 35042 Rennes Cedex, FRANCE, E-mail: luce.morin@irisa.fr

‡Graduate School of Information Sciences, Tohoku University, E-mail: kodeg@fractal.is.tohoku.ac.jp

compression of the different parts of the representation, that is 3D model, texture images and camera parameters use adapted compression techniques.

3.1 Texture images

We use a representation similar to MPEG format for low bitrate (IPP), that is: one *Intra* (I) image at the beginning of the video stream followed by *Predicted* (P) images.

The first image of the sequence is compressed using a method similar to JPEG2000 standard [10], that is: a wavelet transformation, an adaptive quantification (using *EBCOT*) and an entropic coding. The following texture images are predicted using the previous decoded texture image (of the previous GOP) and the previous 3D model: the previous 3D model is mapped with the previous texture image and projected using the camera parameters of the last image of the previous GOP (i.e. the first image of the current GOP). It produces a image prediction of the texture image similar to a forward motion compensated image in classical video coding scheme (fig. 1). We then encode the difference image using the same method as for the *Intra* image, adapted to the *Predicted* image characteristics.

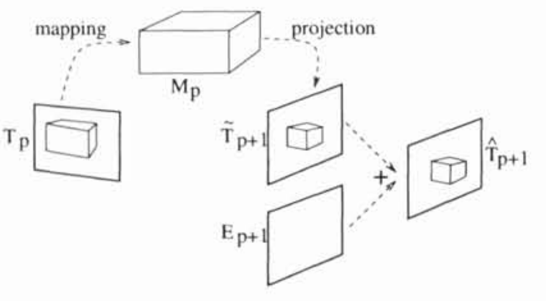

Figure 1: Principle of predictive coding method of the texture images.

3.2 3D models

Each 3D model can be see as a 3D model extruded from the uniformly meshed depth map. The 3D models stream, that is the decimated depth map, is also compressed using a wavelet transformation. However, we use an adapted quantification law: the depth (from the camera position to the 3D vertex of the 3D mesh) of each point of the 3D model is quantified using a normalized inverse law. Assuming a pure translational motion, this quantification allows to compressed the 3D model with a linear reprojection error regarding to the quantification step. The mesh size also allows to adapt the bitrate of the compressed video sequence. We typically fix the mesh size to 8 for low bitrate and 12 for very low bitrate. The quantization step is chosen such as the projection error is less than 0.5 pixel.

3.3 Compressed stream generation

The representation is effectively compressed, allowing to compute the real coding cost. We can accurately choose the target bitrate R_c thanks to *EBCOT* coding of texture images. The coding cost R_{text} of the current texture image is simply chosen as:

$$R_{text} = \frac{R_c.(t_{n+1} - t_n)}{\text{fr}} - (R_{cam} + R_M) \qquad (1)$$

with fr the framerate of the video sequence, R_{cam} the coding cost of the camera parameters (100 bits per *Keyframe* images and 50 bits for other intermediate images) and R_M the coding cost of the 3D model.

One must notice that the proposed representation intrinsically allows 3D manipulations like stereovisualization or scene manipulation [9].

4 Results

The compression scheme performance is first compared to state of the art low bitrate encoder (H26L) and we show new results in very low bitrate coding area.

On figure (5), we show a result with the *Street* video sequence [1]: format is CIF 4:2:0 at 25Hz, global motion is a translation along z-axis, neither camera parameters nor scene content are known. The internal camera parameters are fixed to typical values and focal length is fixed to an approximate value. The video sequence is compressed with H26L video encoder [2] at 82kb/s which is the minimum bitrate with this coder. The sequence is also encoded with the proposed method denoted *Rec3D* with the same parameters. The figure (5) shows *PSNR* score along the sequence: *Texture* curve denotes the *PSNR* of the compressed texture images mapped on the successive 3D models. The *Image* curve denotes the *PSNR* of the reconstructed sequence with the *Rec3D* method and *H26L* with the H26L encoder. The curves show that *PSNR* score are similar for both methods, and better for *Rec3D* method when the 3D model is refresh (peak on the second curve). On figures (2-4), we compare visual quality of reconstructed video sequence: the images clearly show that *Rec3D* provides better visual quality. For such a bitrate, H26L method introduces classical block artifacts and texture degradation whereas *Rec3D* provides better texture images resulting in good visual quality of the reconstructed video sequence.

On figures (6-9), we show similar results on the *Stairway* video sequence. The format of these sequence is CIF 4:2:0 at 25Hz, global motion is a translation along x-axis, neither camera parameters nor scene content are known. The internal camera parameters are fixed to typical values and focal length is fixed to an approximate value. One must notice that these video sequence is very shaky. The video sequence is compressed with the H26L video encoder at 125kb/s which is the minimum bitrate with this coder, and the proposed coder. The figure (9) shows that the *PSNR* of H26L coder is better than the one of *Rec3D* coder. However, the *PSNR* of the texture images and figures (6-8) show that the visual quality of the reconstructed sequence still is better with the *Rec3D* coder.

Moreover, the proposed method allows very low bitrate coding (up to 16kb/s for CIF, 25Hz format) which are not reachable by classical image based coders. The figures (10-12) shows results with the *Street* video sequence for a target bitrate of 16kb/s. The figure (12)

[1] video sequence from Thomson Multimedia and FTRD
[2] TML coder v.6

shows the *PSNR* score along the video sequence: *Texture* curve shows that *PSNR* score remains good for such a bitrate. The figures (10) and (11) show an image extracted from the reconstructed video sequence: we see that global visual quality is quite good, despite compression artifacts. One must notice that such a bitrate is considered as very low for CIF/25Hz format video sequence.

5 Conclusion

We shows results on real video sequence and evaluate performance of vision computer based approach for video compression. The results clearly show that such an approach is very efficient for low bitrate coding and also show new results for very low bitrate coding. We show that the proposed scheme allows better performance than classical scheme like H26L for static scene video sequences. However, quality still is difficult to evaluate for such a coding approach: the method is intrinsically not based on a *pixel coding* approach, making classical quality measures (such as *PSNR* score) unsuitable. Moreover, we plan to extend the proposed scheme to scene with moving objects, making the approach more general for video coding purpose.

Acknowledgments

This work is supported by the Japan Society for the Promotion of Science (JSPS) through JSPS Postdoctoral Fellowship for Foreign Researchers.

References

[1] F. Prêteux and M. Malciu, "Model-based head tracking and 3d pose estimation," in *Visual Conference on Image Processing*, San Jose, California, 1998, pp. 94–110.

[2] B. Girod and al., "3d image models and compression - synthetic hybrid or natural fit?," in *Proc. ICIP*, Oct. 1999.

[3] G. Martinez, "Joint position estimation for object-based analysis-synthesis coding," in *Proc. VCIP*, June 2000.

[4] M. Pollefeys, R. Koch, M. Vergauwen, B. Deknuydt, and L. Van Gool, "three-dimensional scene reconstruction from images," in *proceedings SPIE Electronic Imaging, Three-Dimensional Image Capture and Applications III*, 2000.

[5] R. Kochand M. Pollefeys and L. Van Gool, "Realistic 3d scene modeling from uncalibrated image sequences," in *Proc. ICIP*, Oct. 1999.

[6] M. Magnor and B. Girod, "Model-based coding of multi-viewpoint imagery," in *Proc. VCIP*, June 2000.

[7] Andrew W. Fitzgibbon and Andrew Zisserman, "Automatic camera recovery for closed or open image sequences," in *ECCV (1)*, 1998, pp. 311–326.

[8] D. Nistèr, "Reconstruction from uncalibrated sequences with a hierarchy of trifocal tensors," in *Proc. ECCV*, Dublin, Ireland, 2000, pp. 649–663.

[9] Franck Galpin and Luce Morin, "Sliding adjustment for 3d video representation," *Eurasip Journal on Applied Signal Processing, special issue on Signal Processing for 3D Imaging and Virtual Reality*, vol. to be published, 2002.

[10] Information Technology, "Jpeg2000 image coding system, iso/iec fdis 15444-1," 2000.

[11] S. Pateux, G. Marquant, and D. Chavira-Martinez. Object mosaicking via meshes and crack-lines technique. application to low bit-rate video coding. In *Proceedings of Picture Coding Symposium 2001*, pages 417–420, Seoul, Korea, April 2001.

Figure 2: *Street* video sequence at 82kb/s: image 100 reconstructed by H26L coder.

Figure 3: *Street* video sequence at 82kb/s: image 100 reconstructed by Rec3D coder.

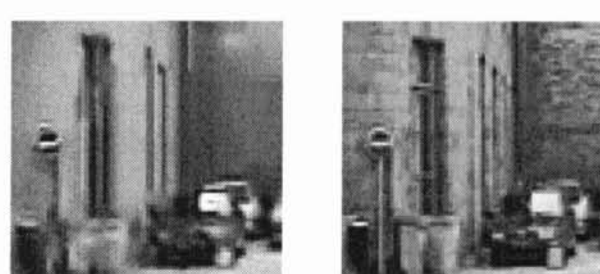

Figure 4: *Street* video sequence at 82kb/s: zoom on image 100 reconstructed by H26L (left) and Rec3D (right) coder.

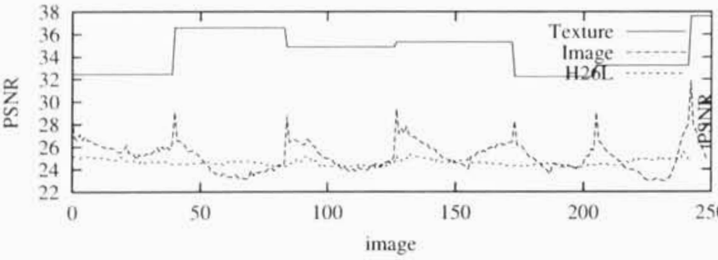

Figure 5: *PSNR* of *Street* video sequence at 82kb/s: comparison between H26L and Rec3d method.

Figure 6: *Stairway* video sequence at 125kb/s: image 67 reconstructed by H26L coder.

Figure 7: *Stairway* video sequence at 125kb/s: image 67 reconstructed by Rec3D coder.

Figure 8: *Stairway* video sequence at 125kb/s: zoom on image 67 reconstructed by H26L (left) and Rec3D (right) coder.

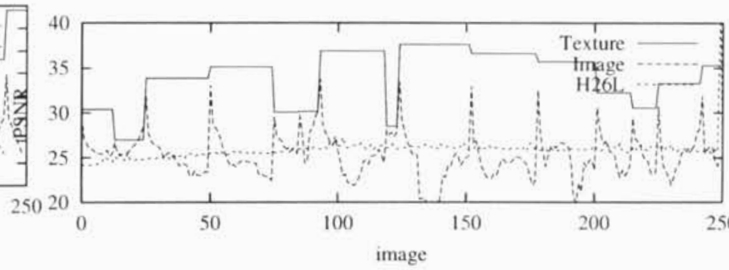

Figure 9: *PSNR* of *Stairway* video sequence at 125kb/s: comparison between H26L and Rec3d method.

Figure 10: *Street* video sequence at 16kb/s: image 100 reconstructed by Rec3D coder.

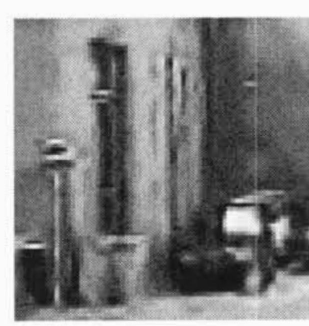

Figure 11: *Street* video sequence at 16kb/s: zoom on image 100 reconstructed by Rec3D coder.

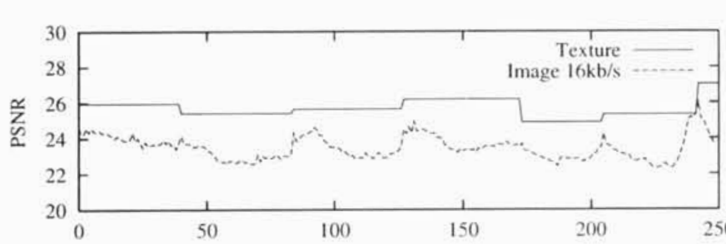

Figure 12: *PSNR* of *Street* video sequence at 16kb/s compressed with Rec3D method.