

# SCALABLE AND EFFICIENT CODING OF 3D MODEL EXTRACTED FROM A VIDEO

*Raphaèle Balter*<sup>1,2</sup>, *Patrick Gioia*<sup>1</sup>, *Luce Morin*<sup>2</sup> and *Franck Galpin*<sup>2</sup>

<sup>1</sup> FRANCE TELECOM R & D, 4 rue du Clos Courtel, 35512 Cesson-Sevigne, France

<sup>2</sup> IRISA-INRIA, Campus de Beaulieu, avenue du General Leclerc, 35042 Rennes, France  
email: {raphaele.balter,patrick.gioia}@rd.francetelecom.com and {rbalter,lmorin}@irisa.fr

## ABSTRACT

This paper presents an efficient and scalable coding scheme for transmitting a stream of 3D models extracted from a video. As in classical model-based video coding, the geometry, connectivity, and texture of the 3D models have to be transmitted, as well as the camera position for each frame in the original video.

The proposed method is based on exploiting the interrelations existing between each type of information, instead of coding them independently, allowing a better prediction of the next 3D model in the stream. Scalability is achieved through the use of wavelet-based representations for both texture and geometry of the models.

A consistent connectivity is built for all 3D models extracted from the video sequence, which allows a more compact representation and straightforward geometric morphing between successive models. Furthermore this leads to a consistent wavelet decomposition for 3D models in the stream.

Quantitative and qualitative results for the proposed scheme are compared with the state of the art video coder H264, 3D model-based Galpin coder and independent MPEG4-based coding of the information. Targeted applications include distant visualization of the original video at very low bitrate and interactive navigation in the extracted 3D scene on heterogeneous terminals.

## 1. INTRODUCTION

3D reconstruction from images and video is an attractive alternative for complex and tedious manual modeling of large and complex natural scenes.

In target applications such as virtual tour in urban environments, metric precision is not a key point whereas photo-realism and more generally visual quality is an essential requirement. 3D information has to however be realistic enough to allow 3D based functionalities, such as interactive visualization and hybrid synthetic-natural composition.

A natural extension of such applications is their distant version, where the scene representation is available on a server and must be transmitted to a distant terminal for visualization and navigation. In this case, the 3D scene representation must verify several additional requirements: it must be encoded in a compact and streamable format and the output stream must be scalable, i.e. it can be divided into sub-streams compatible with several different network and terminal capabilities.

Compression and transmission of 2D contents (images and videos) for distant visualization has been studied for long and lead to very efficient standardized methods such as the H264(MPEG-AVC) codec[SW02]. Scalable schemes have been standardized for fixed images in the JPEG2000 specification [Tec00], and they are

being discussed for video coding in MPEG and ITU standardization consortium.

However, such schemes are not easily extended to 3D contents. In 3D model-based video coding [Ga99] [Mar00] [PM98], the video is represented as one or several textured 3D models and a set of camera/object parameters, but it is restricted to scenes of specific contents (for instance a human face as in video-conferencing), which is known both at the coder and decoder side.

In the computer graphics area, compression and transmission of synthetic 3D meshes has focused on scene partitioning for data streaming, and on efficient and scalable coding of 3D models geometry and connectivity [AG03]. It is assumed that texture will be transmitted as a simple image or as a few parameters. However, in the case of 3D models extracted from videos, texture is the most expensive information to be transmitted, and it is a major factor in the final visual quality. Texture and geometry interactions should thus be taken into account for optimizing scene representation compactness together with visual quality.

In both image/video coding and synthetic 3D models coding, wavelets [Dau92] have been chosen to achieve scalability in an elegant and efficient way, in particular second generation wavelets [LDW97] which allow to apply wavelet decomposition on arbitrary shaped supports.

We thus propose a coding scheme for 3D models extracted from a video of a static scene based on a stream of 3D models and a scalable representation using interrelations between geometry, connectivity and texture. The compression, transmission and visualization requirements are taken into account as soon as the 3D reconstruction and representation steps.

To meet coding requirements we do not want to do any assumption on camera parameters, scene contents or length of the sequence. In this context we consider that the final navigation will be visually acceptable only for viewpoints close to the original acquired views. We thus propose to estimate a set of local 3D models, instead of a unique one containing all the information viewed in the entire video sequence, as in previous automatic shape-from-motion schemes [FZ98] [Nis00]. This choice has several advantages:

- Global consistency of extracted 3D information is not required. This allows to simplify estimation and use inaccurate camera parameters.
- The set of 3D models directly provides a streamed format, and avoids the complex and symmetrical steps of fusion and partitioning.
- Global illumination changes along time are modeled and reconstructed.

- Sequences of arbitrary size can be processed with on-the-fly estimation and streaming of the 3D models.
- Camera motion is unconstrained as long as it is not degenerated (e.g example a pure rotation).

Our first experiments have validated this scheme [GMD02]. However, a complex and computationally expensive morphing and re-meshing process was used at the decoder side to ensure smooth visual transition between successive 3D models [GBMD04] and mesh geometry had fixed resolution. We here propose to build a hierarchical representation based on wavelet decomposition over a common connectivity for all 3D models. This representation provides time and spatial scalability. It also allows straightforward morphing between successive models which ensures visual continuity at no additional cost. An efficient coding scheme is presented which takes into account the interrelations between geometry, connectivity, texture and camera positions.

In the following, we first briefly describe the extraction of the 3D models from a video of a static scene captured with a moving camera and then pay more attention on hierarchical representation. We then present interrelations between media flows and the coding and decoding schemes. Results on real video sequences are finally shown and discussed.

## 2. 3D RECONSTRUCTION

3D models are automatically extracted from the input video sequence of a static scene using shape from motion methods developed in computer vision [PVV<sup>+</sup>01] [KPG00] [MG00].

Each 3D model is extracted and used for a small portion of the video sequence called a GOF (Group Of Frames). The GOFs overlap each other, that is the last image of a GOF is the same as the first image of the next one (cf. Fig. 1). We call these images *keyframes*. Keyframes are automatically selected adaptively to video contents, based on several criteria depending on motion, percentage of outgoing points in images and 3D reconstruction stability [GM02].

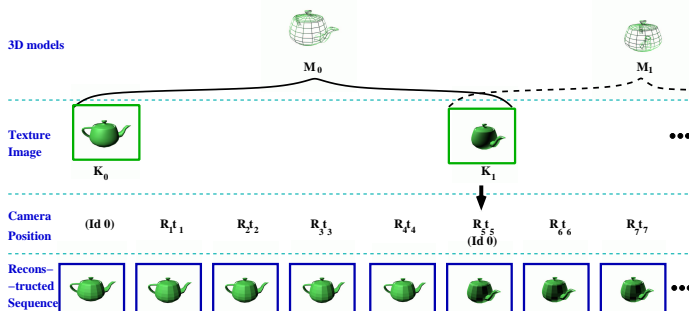


Fig. 1. Reconstruction of the original sequence

For each GOF, a 3D model is automatically extracted on-the-fly. Disparity estimation is performed by a dense mesh-based motion estimator using multi-grid and multi-resolution approaches [PMCM01]. The camera intrinsic parameters are estimated using a self-calibration algorithm or fixed to approximate values. Accurate intrinsic parameters are not required since we design our representation to be robust to inaccurate intrinsic parameters. The extrinsic

parameters are computed using classical calibration methods and adapted bundle adjustment method [GM02]. The dense motion field from the first to the last image of the GOF and camera parameters for these two images allow to reconstruct a dense depth map of the first image of the GOF. Fig. 2 shows an example of such a depth map extracted from the *Street* video sequence (cf. results section). Camera extrinsic parameters are then retrieved for each image in the video sequence using a pose estimation algorithm.



Fig. 2. An example of a depth map (bottom) extracted from the *Street* video sequence. Pictures on top is the corresponding image in the sequence.

The 3D reconstruction step thus provides for each GOF:

- the 3D model geometry: a dense depth map of the scene viewed from the first image in the GOF
- the 3D model texture: the first image in the GOF
- camera parameters for each frame in the GOF.

## 3. HIERARCHICAL 3D MODELS

We now explain how the hierarchical 3D triangular meshes are constructed from the dense depth maps.

### 3.1. Notations

The following notations will be used in the rest of the paper

- $M_k^i$  is the 3D model associated with GOF  $k$  at resolution  $i$
- $BM_k$  is the coarsest 3D model associated with GOF  $k$
- $FM_k$  is the finest 3D model associated with GOF  $k$
- $K_k$  is the keyframe of GOF  $k$  (used as texture image  $T_k$  for  $M_k^i$ ).
- $C_k$  is the viewpoint associated with camera position of the key frame  $K_k$ .  $C_k$  is defined by translation  $t_{t_k}$  and by rotation  $R_{t_k}$ .
- $CM_k$  is the 3D model associated with GOF  $k$  including correspondents of vertices of the previous model  $M_{k-1}^i$  that are still visible from  $C_k$  and the associated faces
- $Pr(M, T, C)$  is the image issued from the  $3D \rightarrow 2D$  projection of  $M$  textured using  $T$  with camera position  $C$ .

### 3.2. Single connectivity mesh and global indexing

Though theoretically elegant and commonly used for generic 3D models, it is not straightforward to apply a multi-resolution wavelet scheme to the 3D model stream we deal with. Indeed, our approach is based on swapping smoothly from one model to another when changing GOF, causing changes in connectivity and issues due to hidden parts.

In the model-based setting, one is given a collection of coarse base meshes, that are refined by inverse wavelet transform into dense models for every GOF. This setting is not appropriate at all for defining a single continuous scene, because switching from one set of wavelet coefficients to another introduces discontinuities, and it is very hard to setup the geometry correspondence from one GOF to another. Applying wavelets for such a streaming enforces the use of a standalone indexing reference for the wavelet trees.

Since zero-trees are indexed by edges of the base meshes [KSS00], this reference should take the form of a planar graph, denoted as the *single connectivity mesh (SCM)*. The purpose of this single connectivity mesh is to gather the connectivity information of every base mesh in the sequence, regardless of the geometry of the base meshes.

The generation of this SCM is based on correspondences between vertices of base meshes given by a dense motion field. We introduce a new indexing system for the vertices of base meshes. A new index is associated to each vertex, but this index is the same for all corresponding vertices of all base meshes integrating it. The SCM is described as a list of triangles expressed in the new global indexing system.

### 3.3. Base meshes construction

Base meshes use non-uniform triangulation in order to ensure global consistency and smooth transition between models. Furthermore to better represent the video content, the base mesh has also to fit features of the scene.

For the first GOF, the adaptive triangular mesh is based on interest points computed on the first frame in the GOF (key image), using the Harris corner detector [HS88]. A 2D Delaunay triangulation of these points under the constraint of image borders gives us our base mesh. The 3D model is then obtained by elevation of this 2D mesh with 3D information given by the depth map.

In order to smooth transitions between model, each base mesh is constrained to contain the correspondents of the previous base mesh vertices, if they are still visible in the GOF. Triangles of these correspondents, whose set is denoted  $CM_k$ , are included in the base mesh:  $CM_k \subseteq BM_k$ .

When adding vertices on the border of the model, the new triangulation has to preserve the connectivity derived from the preceding GOF without edge crossing. This is achieved by a 2D Delaunay triangulation constrained not only by image borders but especially by correspondent mesh  $CM_k$  borders.

Furthermore this constraint of including correspondents vertices enables to smooth transitions between 3D models through implicit morphing that can be achieved at low computing cost, using a simple linear interpolation between vertices :

$$M_c = \alpha * M_k^i + (1 - \alpha) * M_{k+1}^i$$

$$\text{with } \alpha = \frac{\|t_{t_{k+1}} - t_{t_c}\|}{\|t_{t_{k+1}} - t_{t_k}\|}$$

### 3.4. Wavelet decomposition

We define the “base mesh”, denoted  $BM_k$  as the mesh at coarsest level and the “fine mesh”, denoted  $FM_k$  as the dense mesh at finer level. The refinement from coarser to finer level is then expressed as wavelets coefficients ( $r_i^j$  on Fig. 4) using a second-generation wavelet transform.

To obtain the hierarchical meshes we then use canonical facets quadrisections in place to define scaling functions and wavelets [LDW97]. Each base mesh  $BM_k$  is refined by recursive facets quadrisections into a hierarchy of finer meshes, up to fine mesh  $FM_k$  with resolution equivalent as the dense depth map. The information to encode with wavelet analysis is the depth difference  $p$  between a vertex  $M^{i+1}$  and the middle of the associated edge as shown on Fig. 3 where  $i$  represents the resolution of the mesh.

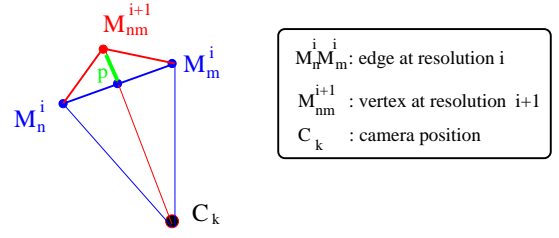


Fig. 3. Estimation of the depth to encode with wavelets

The representation to encode is given on Fig. 4. The fine models  $FM_i$  are represented by the base meshes  $BM_i$  and wavelet coefficients  $r_i^j$ . The scale coefficients  $e_i^j$  expressing the geometry of the base meshes are gathered and indexed by the single connectivity mesh.

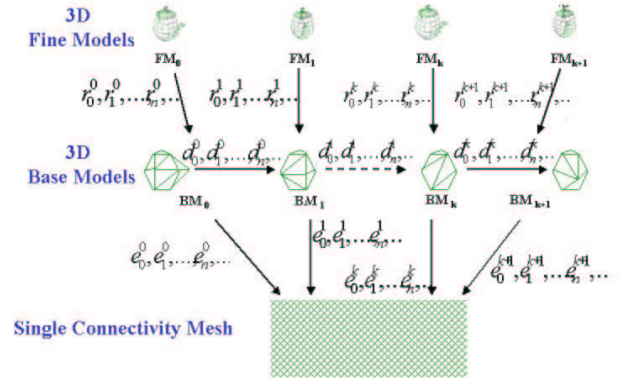


Fig. 4. Proposed representation based on a 3D model stream and second generation wavelets.

At this stage, we have obtained a set of multi-resolution meshes based on non-uniform triangulation, with corresponding vertices. This representation has several advantages: vertex positions can be adapted to scene contents; vertex to vertex correspondence between successive models is implicitly provided by the mesh structure and has thus not to be transmitted or estimated at the decoder; smoothed transitions between 3D models through

implicit morphing can be achieved at low computing cost, using a simple linear interpolation between vertices.

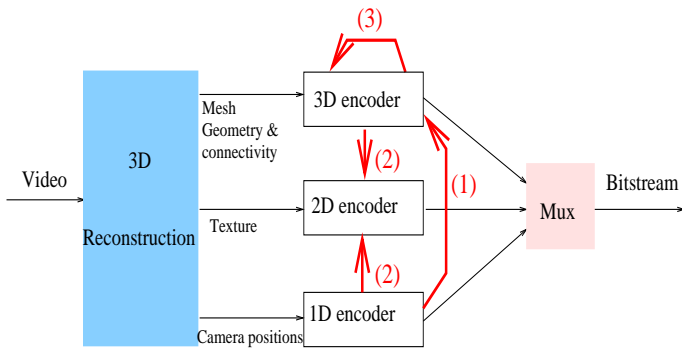
This 3D representation for videos induces several media streams, such as topology (the connectivity of the single base mesh), geometry (wavelet coefficients and incremental model representation), and texture as well as camera parameters for each frame.

Efficient coding of these streams is described in the next section.

#### 4. INTERRELATIONS BETWEEN DIFFERENT MEDIA

A key observation is that the information streams to be transmitted are not independent and an efficient coding algorithm should take into account this redundancy for both compression rate and quality of the reconstructed sequence.

Here is a non-exhaustive list of interrelations occurring within 3D model-based coding. Those interrelations are summarized on Fig. 5.

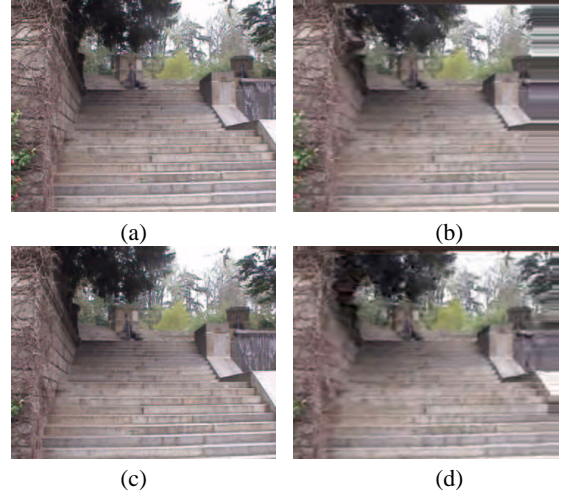


**Fig. 5.** Interrelations between the media in 3D model-based coding

First of all camera positions can help on the transmission of 3D models. Indeed, for each vertex of the non-uniform meshed model 5 coordinates have to be transmitted: 3 coordinates for vertex location and 2 for texture coordinates. If camera positions are known on the decoder side only 3 parameters instead of five are required. Indeed texture coordinates can be retrieved by reprojecting 3D vertices  $M_i$  on camera position of the GOF key frame viewpoint. As the key frame is also the texture image, the coordinates of the obtained projection  $m_i$  are the texture coordinates for vertex  $M_i$ . These parameters can be exact positions of 3D vertices  $M_i$  or texture coordinates  $m_i$  and the associated depth  $d_i$  in the case of the 3D model being an elevation map.

Furthermore since 3D models represent overlapping parts of the scene, the associated textures include redundant information. So it is interesting to compress texture images using a classical MPEG scheme IPP where first image is in Intra mode and others in Predicted mode. Using 1D and 3D information predicted images can be estimated thanks to the reprojection of the textured model onto associated camera. This is represented by arrows (2) on Fig. 5. So texture decoding is strongly dependent on the transmission of both 3D model and camera position. An example of a predicted images is given on Fig. 6.

In the same way, 3D models geometry share common information. This redundancy can be reduced by using an IPP scheme for 3D information. Predicted models are given by the common part of the precedent model. Furthermore the hidden region can be



**Fig. 6.** “Thabor” sequence: Predicted images. Image 98 from original sequence (CIF, 25Hz) (a) and associated predicted image (b), Image 107 from original sequence (c) and associated predicted image (d)

managed over single connectivity mesh by changing its connectivity in time.

Depending on the envisioned applications, texture or geometry has to be favored. For instance, texture is very important in video broadcasting applications, because human vision is very sensitive to texture. In virtual reality applications, 3D geometry has to be accurate. An appropriate trade-off thus must be found so that the allocated bitrate is shared between the streams. This is an important interaction between our 3 streams because rate allocated to one depends on the others. With a unique stream instead of several we can update online the rate associated for each medium whose particular encoders are presented in the next section.

### 5. MEDIA ENCODERS

#### 5.1. Camera encoding

Key frame camera positions are differentially encoded and intermediate camera positions are retrieved by linear interpolation between key positions:

$$C_c = \alpha * C_k + (1 - \alpha) * C_{k+1} \text{ with } \alpha = \frac{\|t_{k+1} - t_c\|}{\|t_{k+1} - t_k\|}$$

#### 5.2. Texture encoding

A IPP scheme is used where predicted image  $P(I_{k+1})$  are obtained by the reprojection of the precedent textured model on the current key position as described in Section 4. Padding is used in areas where prediction does not apply.

$$P(K_{k+1}) = Pr(M_k^i, T_k, C_{k+1});$$

#### 5.3. 3D Geometry and connectivity encoding

3D information encoding is based on transmitting the base mesh  $BM_k$  and a set of wavelet coefficients for refinements.

2D texture coordinates are not encoded as explained in section 4 as they can be retrieved by reprojecting the 3D model on the associated key position.

The base mesh associated with the first GOF is encoded using TS encoder (Topological Surgery encoder) [TR98] (geometry and connectivity).

Because of implicit morphing the order of vertices in the list of a base mesh is really important. So we cannot use TS encoder for other base meshes. Thus 3D coordinates for model  $BM_k$  having a correspondent in the previous base mesh  $BM_{k-1}$  are differentially encoded with respect of their correspondent's position, as coefficients  $d_i^j$  on Fig. 4.

Indeed we can rapidly identify vertices having a correspondent in the next model by projecting vertices of the current model on the key image of the next GOF. By this way, we retrieve the common part between two models at the decoding stage without transmitting additional information. The new global indexing system introduced in Section 3 helps to implicitly encode correspondences between successive base meshes. Furthermore thanks to the unique connectivity mesh, introduced in Section 3, and this new indexing system, faces of corresponding vertices have not to be encoded either.

In the particular case of elevation map more advantage can be taken from interrelations. Indeed projection of vertices of the current model on the key image of the next GOF gives 2D texture coordinates of the correspondent vertices in the next base mesh. Indeed only the depth has to be transmitted. In order to avoid numerical errors a stage of robust selection of base mesh vertices is added to Harris corner selection. In this case vertex consistency really allows a more efficient coding of 3D information through differential encoding.

3D coordinates of new vertices added to a base mesh and the associated faces are encoded with TS encoder and the 2D texture coordinates of those vertices are retrieved by reprojection.

The wavelet representation is further compressed with an adaptation [KSS00] of the classic SPIHT zero-tree encoder, adding bit-plane scalability to spatial and frequency scalability provided by the wavelets themselves.

## 6. RESULTS AND DISCUSSION

We show results on two sequences, illustrating the compression rates reached by comparison with Galpin and H264 encoders at low and very low bitrates on both constrained and free navigation.

We use the classical lazy wavelet analysis, which is a bi-orthogonal transformation realizing a sub-sampling [SS95].

In order to show the interest of including interrelations introduced in section 4, we propose to compare results of our encoder with an encoder denoted *naive encoder* where the single stream is obtained by concatenating all streams or blindly multiplexing them without exploiting the interrelations between the different media.

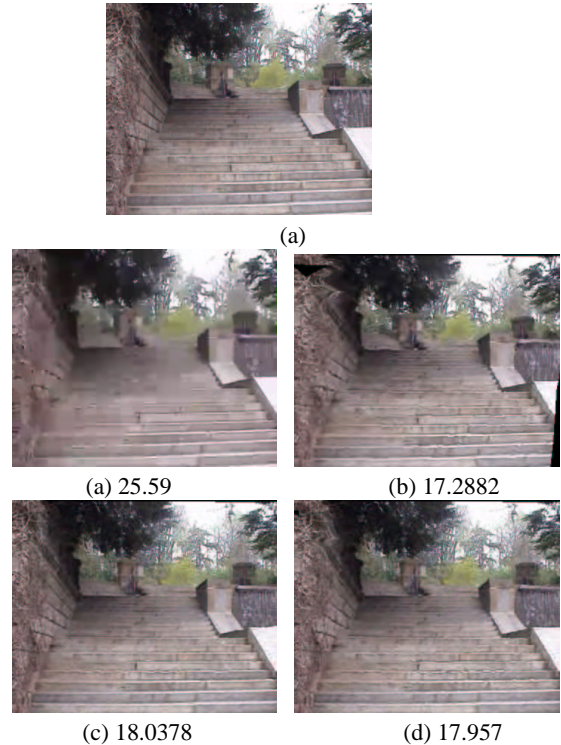
For this naive encoder the same encoders are used than for our method:

- camera positions : key positions are differentially encoded,
- texture : all images are encoded in intra-mode with EBCOT [Tec00]
- 3D : a scalable encoding is done based on a coarse mesh (encoded with TS encoder) and a set of wavelets coefficients for refinements. 2D texture coordinates of vertices are encoded with 3D coordinates by TS encoder.

### 6.1. Compression results

We give compression results for a sequence of 110 frames of the *Thabor* sequence (lateral motion) for low and very low bitrates. This sequence No comparison can be done with H264 or with the naive encoder if such a low rate cannot be reached at 25Hz.

We show images on Fig. 7, 8 and 9. While PSNR is appropriate for measuring block based errors, it has little meaning when it comes to geometric distortion (e.g image translated of one pixel). This is shown by PSNR values given for the reconstructed images on Fig. 7 and 8 but we thus rather use visual quality of the reconstructed images as an evaluation.

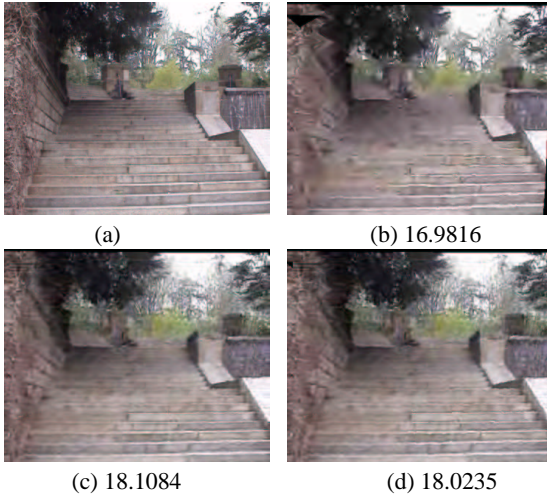


**Fig. 7.** “Thabor” sequence: Image 125 from original sequence (CIF, 25Hz) (a) and reconstructed images at 125kb/s with naive coder (b), with Galpin coder (c) and with our coder (d)

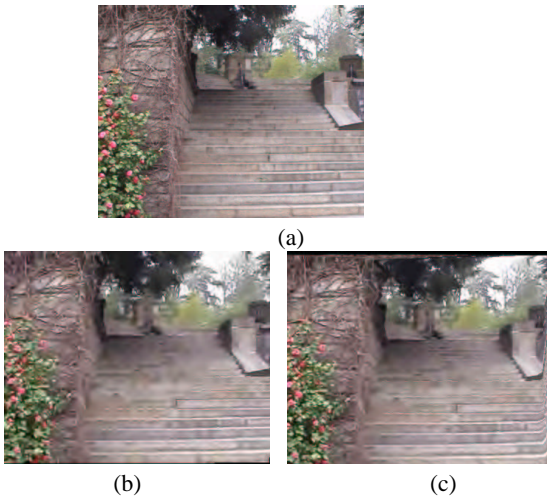
We choose to encode a precise 3D information with our coder even for very low bitrate in order to be adapted to compression as well as other functionalities such as free view point generation. Of course this 3D information could have been reduced using space scalability to favor texture information.

In order to adapt the information to transmit for Galpin encoder we choose to favor texture information and thus reduce the rate allocated to 3D information in order to optimize the quality of the reconstructed sequence because of the context of compression tests. In Galpin's method depth maps were encoded as an image with EBCOT thus the number of the vertices of the uniform mesh is then reduced to be competitive with the rate achieved by our progressive coder (15kb for 2400 vertices against 22kb for 1600 vertices for Galpin's coding for the stairs sequence).

But for 3D bases applications, such as free viewpoint generation or augmented reality, visual quality if the reconstructed images with Galpin encoder is lower as the model is coarser. In order



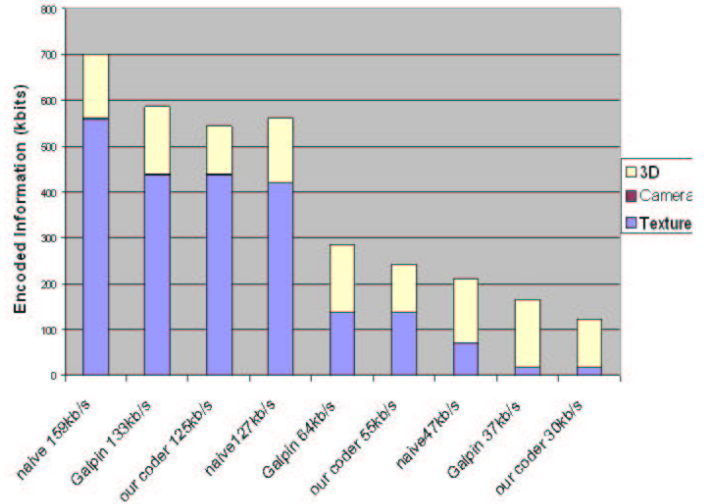
**Fig. 8.** “Thabor” sequence: Image 122 from original sequence (CIF, 25Hz) (a) and reconstructed images at 55kb/s with naive coder (b), with Galpin coder (c) and with our coder (d)



**Fig. 9.** “Thabor” sequence: Image 71 from original sequence (CIF, 25Hz) (a) and reconstructed images at 30kb/s with Galpin coder (b) and with our coder (c)

to get the same quality we have to encode the same information, i.e about the same number of vertices.

Compression results for the encoding of the same information is given in Fig. 10 showing the profits for the bitrate by using our coder against Galpin coder. Since texture information is pre-heminent over geometry for low bitrates, this profits is particularly interesting for very low bitrate.



**Fig. 10.** Compression results for the same encoded information for Thabor sequence

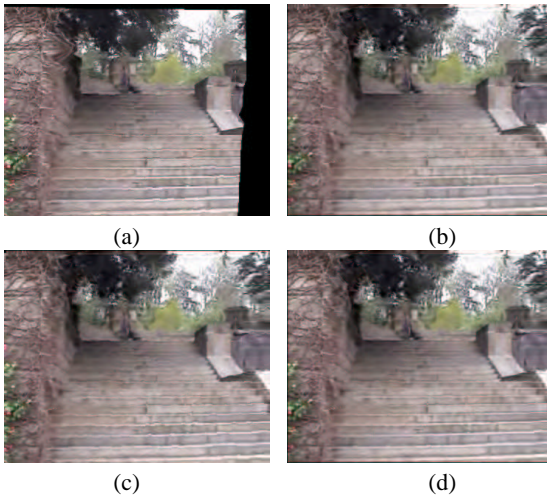
Benefits given by using the interrelation (cf. Fig 10) must be increasing with the size of the sequence as data for the first model are encoded in Intra mode. Furthermore for the naive encoder we have not taken into account the cost of 2D coordinates for vertices retrieved with wavelets coefficients.

## 6.2. Virtual navigation results

Not only including interrelations into the coder dramatically decreases compression rates as shown on Fig. 10 but it also increases visual quality of the reconstructed sequence by linking up different models together.

As introduced in the precedent section by reprojecting a 3D model on the camera position associated with the following model we can rapidly retrieve the common part between two models at the decoding stage without transmitting additional information. If vertices are encoded in a particular order we can so implicitly retrieve correspondences between vertices of the models. Transition between models brings visual discontinuities due to connectivity and geometry pops. If we know the correspondences between vertices of successive models we can easily morph one into the other with a linear interpolation between corresponding vertices. This morphing can be done at each level of subdivision thanks to the consistent connectivity of all base meshes. So at each level transition between models can be smooth using a morphing based on implicit matching.

Fig. 11 shows successive images around a transition between two GOFs. We can see on these images the discontinuity between two successive frames of the video. Fig. 11 shows also the same images reconstructed with our method. The artefacts are greatly reduced by the morphing enabled by model correspondences.



**Fig. 11.** “Thabor” sequence: Reconstruction of original path. Successive reconstructed images with naive coder (a) and (b), Successive reconstructed images with our coder (c) and (d).

Fig. 12 shows the results obtained with Galpin 3D fading [GM02]. Artifacts near model transitions are reduced but it also produces ghost effects.



**Fig. 12.** “Thabor” sequence: Very low bitrate reconstruction on the original path (30kb/s). With Galpin coder (a), with our coder (b).

The implicit morphing strongly contributes to the visual quality of the scene avoiding those ghost effects while smoothing transitions between models.

One should note that this implicit morphing does not add computational time thanks to the SCM contrary to classical morphing procedures defining correspondences and remeshing and still including a connectivity jump between GOFs..

Fig. 13 gives reconstruction results during free navigation, i.e when the viewer is not restricted to the camera path defined during capture. In a similar way results on the original path visual quality of reconstructed images is largely increased by eliminating artifacts of ghost effects.

## 7. CONCLUSIONS AND FUTURE WORK

We have presented a new model-based coding scheme for video of static scenes with fine-grain scalability, allowing content adaptation over a very wide spectrum of terminals and networks. In



**Fig. 13.** “Street” sequence: Reconstruction on virtual path. With uniform mesh and fading (a), with proposed method (non uniform mesh and implicit morphing) (b).

particular, 3D can be streamed adaptively in applications of free navigation over networks. The coder, showing better compression results and finer scalability than previous schemes, exploits all the power of second generation wavelets and implicit morphing thanks to the design of a connectivity mesh gathering every GOF information.

To still improve this scheme it would be interesting to express of the temporal increments in a wavelet basis. Furthermore, reconstructed images have shown the need for a suitable error measure for reconstructed images taking into account the geometric distortion to meet visual quality measure.

## 8. REFERENCES

- [AG03] Pierre Alliez and Craig Gotsman. Recent advances in compression of 3d meshes. In *Proceedings of the Symposium on Multiresolution in Geometric Modeling*, 2003.
- [Dau92] I. Daubechies. *Ten lectures on wavelets*, volume 61. CBMS-NSF regional conf. series in appl. math., Philadelphia, PA, society for industrial and applied mathematics edition, 1992.
- [FZ98] Andrew W. Fitzgibbon and Andrew Zisserman. Automatic camera recovery for closed or open image sequences. In *ECCV (1)*, pages 311–326, 1998.
- [Ga99] B. Girod and al. 3d image models and compression - synthetic hybrid or natural fit? In *Proc. ICIP*, October 1999.
- [GBMD04] F. Galpin, R. Balter, L. Morin, and K. Deguchi. 3d models coding and morphing for efficient video compression. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, Washington, USA*, 2004.
- [GM02] F. Galpin and L. Morin. Sliding adjustment for 3d video representation. *Eurasip Journal ASP, special issue on Signal Processing for 3D Imaging and Virtual reality*, 2002.
- [GMD02] F. Galpin, L. Morin, and K. Deguchi. Compression performance of computer vision based coding. In *Proc. MVA, Nara, Japan*, 2002, 2002.
- [HS88] C. Harris and M. Stephens. A combined corner and edge detector. In *Proc. 4th Alvey Vision Conf.*, 1988.

- [KPG00] Reinhard Koch, Marc Pollefeys, and Luc Van Gool. Realistic surface reconstruction of 3d scenes from uncalibrated image sequence. *Journal of Visualization and Computer Animation*, 11:115–127, 2000.
- [KSS00] A. Khodakovsky, P. Schroder, and W. Sweldens. Progressive geometry compression. In *SIGGRAPH 2000 Conference Proceedings*, 2000.
- [LDW97] M. Lounsbery, T. D. DeRose, and J. Warren. Multiresolution analysis for surfaces of arbitrary topological type. *ACM Transactions on Graphics*, 16(1):34–73, January 1997. ISSN 0730-0301.
- [Mar00] G. Martínez. Joint position estimation for object-based analysis-synthesis coding. In *Proc. VCIP*, June 2000.
- [MG00] M. Magnor and B. Girod. Model-based coding of multi-viewpoint imagery. In *Proceedings of the SPIE Conference on Visual Communications and Image Processing, Perth, Australia*, June 2000.
- [Nis00] D. Nister. Reconstruction from uncalibrated sequences with a hierarchy of trifocal tensors. In *Proceedings of the 5th European Conference on Computer Vision ECCV'2000, Dublin, Ireland*, 2000.
- [PM98] F. Preteux and M. Malciu. Model-based head tracking and 3d pose estimation. In *Visual Conference on Image Processing, San Jose, California*, pages 94–110, 1998.
- [PMCM01] 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.
- [PVV<sup>+</sup>01] M. Pollefeys, M. Vergauwen, F. Verbiest, K. Cornelis, and L. Van Gool. From image sequences to 3d models. In *Third International Workshop on Automatic Extraction of Man-made Objects from Aerial and Space Images*, 2001.
- [SS95] P. Schroder and W. Sweldens. Spherical wavelets : Efficiently representing functions on the sphere. In *Siggraph 95*, pages 161–172, 1995.
- [SW02] H. Schwarz and T. Wiegand. The emerging jvt/h.261 video coding standard. In *Proc. IBC, Amsterdam, 2002*, 2002.
- [Tec00] Information Technology. Jpeg2000 image coding system, iso/iec fdis 15444-1, 2000.
- [TR98] Gabriel Taubin and Jarek Rossignac. Geometric compression through topological surgery. *ACM Trans. Graph.*, 17(2):84–115, 1998.