Topology matching for 3D video compression

Tony Tung[†] Francis Schmitt[†] Takashi Matsuyama[‡]

[†]Telecom Paris, CNRS UMR 5141, France, {tony.tung, francis.schmitt}@enst.fr

[‡]Graduate School of Informatics, Kyoto University, Japan, tm@i.kyoto-u.ac.jp

Abstract

This paper presents a new technique to reduce the storage cost of high quality 3D video. In 3D video [12], a sequence of 3D objects represents scenes in motion. Every frame is composed by one or several accurate 3D meshes with attached high fidelity properties such as color and texture. Each frame is acquired at video rate. The entire video sequence requires a huge amount of free disk space. To overcome this issue, we propose an original approach using Reeb graphs, which are well-known topology based shape descriptors. In particular, we take advantage of the augmented multiresolution Reeb graph properties [18] to store the relevant information of the 3D model of each frame. This graph structure has shown its efficiency as a motion descriptor, being able to track similar nodes all along the 3D video sequence. Therefore we can describe and reconstruct the 3D models of all frames with a very low-cost data size. The algorithm has been implemented as a fully automatic 3D video compression system. Our experiments show the robustness and accuracy of the proposed technique by comparing reconstructed sequences against challenging real ones.

1. Introduction

3D video [12] is a recent image media recording technique which produces high quality visual effects. The technology relies on a PC cluster system for reconstructing dynamic 3D object from multi-view video images. Temporal series of voxel representations of the 3D object motion can be obtained in real-time. Afterwards a 3D deformable model is used to accurately reconstruct the 3D mesh models. Finally, video textures are rendered on the reconstructed 3D object surfaces. Experimental results with quantitative performance evaluations demonstrate the effectiveness of these methods in generating high fidelity 3D video from multiview video images. The applications can cover various areas: entertainment (3D games, 3D TV), sports (performance analysis), scientific applications (3D surgery monitoring), cultural heritage (3D archive of traditional dances), and so on. At the moment data are still rare and focus on human motion records. Besides this high technology requires a huge amount of free disk space: for example, $\sim$ 250 Mo are required to store a 1 minute video sequence of a moving human model recorded at 10 frame-per-second (fps).

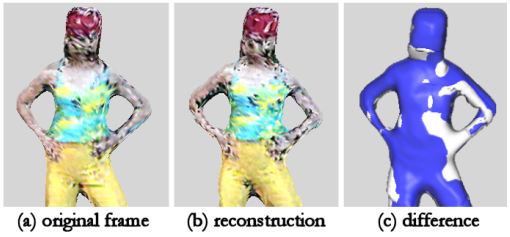


Figure 1. **3D video compression using Reeb graphs**. High fidelity 3D video consists on 3D model sequences reconstructed from multiview video images. The storage cost is huge for long video. This can be highly reduced using our Reeb graph approach with low-loss quality. (a) shows a real video frame. (b) shows a reconstructed frame. (c) shows the similarity of the two frames.

Our paper proposes an original scheme to reduce the data storage cost using a Reeb graph based approach. As human models are articulated objects with limited deformation, and present a relative simple topology, the Reeb graph suits very well to represent their shape as skeleton. It is built using a function μ based on the mesh connectivity. The surface of the object is divided in regions according to the values of μ, and a node is associated to each region. The graph structure is then obtained by linking the nodes of the connected regions. Then a multiresolutional Reeb graph can be constructed hierarchically, based on a coarse-to-fine approach node merging [8]. Keeping advantage of the multiresolutional representation, the augmented multiresolution Reeb graph (aMRG) [18] is an enhanced Reeb graph which includes topological, geometrical and visual (color or texture) information in each graph node. Therefore similarity between two aMRGs can be computed to retrieve the most similar nodes.

Our main contributions relie on a new augmented multiresolution Reeb graph design, which is dedicated to record 3D object shape continuous deformation along 3D video se-

quence. Thus one single graph is enough to reconstruct all 3D mesh models of the whole video. The size of the 3D video sequence is dramatically reduced and the visual quality is preserved as shown in Figure 1.

The next section discusses work related to the study presented in this paper. Section 3 presents the aMRG as a powerful 3D shape descriptor. Section 4 describes our 3D video compression scheme. Section 5 presents experimental results. Section 6 concludes with a discussion on our contributions.

2. Related works

3D video data are still rare as their creation requires a lot of hardware, and only few applications can be found in the scientific literature. However an increasing number of research groups are interested by this recent technology and have developed their own systems dedicated to real-time 3D shape reconstruction [10, 13, 2, 12, 4, 9], the main motivation being the total immersion in a virtual world.

The dancer sequences studied in this paper were acquired in real-time using a cluster of 30 node PCs and 25 cameras [12] (cf. layout illustration in Figure 2). A reconstruction method based on a shape-from-silhouette approach produces 3D mesh model sequences. Afterwards the surface of the 3D meshes are smoothed using a deformable model (3D snake) [7, 12]. Moreover a texture map is computed for each frame. The result is a high quality 3D video sequence recorded at a video speed frame rate (10 fps).

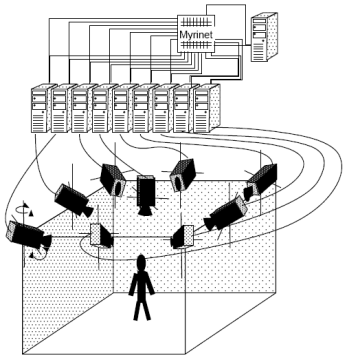


Figure 2. **3D video acquisition layout.** Cameras are linked to a PC cluster and are placed on the floor as well as at the ceiling to capture fully surrounding views of an object.

To address the problem of huge data size management, the literature offers only compression technique applied to 2D images (e.g. JPEG format) and 2D video (e.g. MPEG format), therefore none of these approaches suits our data. To reduce the size of 3D video, single 3D mesh model compression techniques [1] could be applied to each frame, but motion information would have to be added, as well as a framework to manage redundancies between consecutive similar frames. In [6] the *skin-off* scheme proposes to cut each 3D mesh surface along a path, and project geometrical and texture information on 2D images. The sequence is then compressed using MPEG format. However, the heuristic determination of cut path and mapping function for each frame is still to be improved. We believe the continuous aspect of video acquisition contains rich information. Therefore we have focused our effort on a 3D video compression strategy based on dynamical shape description.

To compactly describe 3D shapes, various 3D indexing methods were proposed. In particular, they are dedicated to shape-based retrieval in database of 3D mesh models [17]. Most of the time the encoded information is too coarse to finely reconstruct 3D models or not adapted to our purpose. As well, recent works on human motion analysis from video data presented in [16, 19] are based on single-view acquisition and do not aim to properly reconstruct 3D models.

However using a skeletal or graph representation appears very attractive as it gives a high level description of the shapes and keeps topology. Unfortunately, skeleton extraction is either time consuming, or too sensitive to noise on the object surface, or requires an interactive step to determine seed points. In addition, most of the time no matching scheme is provided, or no approach adapted to retrieval in large database [3, 5]. Hence the augmented Multiresolution Reeb Graph proposed in [18] shows promising skills to accurately describe and store 3D shapes. Its abilities rely on a topology based description, a multiresolution structure associated to rich embedded topological and geometrical information, as presented in the next section.

3. Motion description with aMRG

3.1. Overview of the aMRG

According to the Morse theory, a continuous function defined on a closed surface characterizes the topology of the surface on its critical points. Therefore, a Reeb graph can be obtained assuming a continuous function μ calculated over the 3D object surface.

In our framework, 3D models are defined by their surface and represented as 3D triangular meshes with vertices located in a Cartesian frame. We chose the function μ proposed in [8], which is defined as the integral of the geodesic distance $g(\mathbf{v}, \mathbf{p})$ from $\mathbf{v}$ to the other points $\mathbf{p}$ of the surface:

$$\mu(\mathbf{v}) = \int_{\mathbf{p} \in S} g(\mathbf{v}, \mathbf{p}) dS. \quad (1)$$

This function μ has the property to be invariant to rotations. Its integral formulation provides a good stability to local noise on surface and gives a measure of the eccentricity of the object surface points. A point with a great value of μ is far from the center of the object and from the opposite side. A point with a minimal value of μ is close to the center of

the object. The corresponding Reeb graph is then obtained by iteratively partitioning the object surface into regular intervals of μ_N values and by linking connected regions. For each interval, a node is associated to each different set of connected triangles.

To construct a Reeb graph of R levels of resolution, μ_N is subdivided into 2^R intervals from which the object surface is partitioned at the highest level of resolution. Afterwards, using a hierarchical procedure, Reeb graphs of lower resolution levels are obtained by merging intervals by pairs [8]. The multiresolutional aspect results from the dichotomic discretization of the function values and from the hierarchical collection of Reeb graphs defined at each resolution (cf. Figure 3).

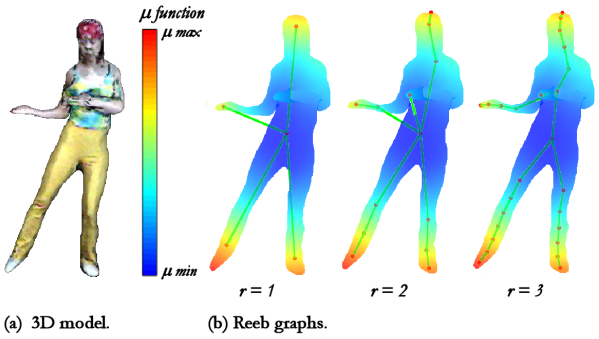


Figure 3. **Multiresolution Reeb graph.** (a) shows a 3D model. (b) shows values of function μ on the surface, with Reeb graphs at resolution $r = 1, 2$ and 3. The graph structure contains more topological and geometrical information at high resolution levels.

In [18], the multiresolution Reeb graph has been augmented by adding new characteristics to the nodes of the graphs, extending topological aspects of the graph matching procedure, and adapting the similarity calculation to the new features. The result is a flexible multiresolution and multicriteria 3D shape descriptor including merged topological, geometrical and colorimetric properties.

3.2. Additive topological features

In order to obtain a better control of the node matching, we propose to exploit the graph topology. Topological features can be deduced by the edge orientations given by μ values. In addition, multiresolution gives valuable information to characterize the global shape of models.

Our strategy is to introduce into each aMRG node n at maximal resolution $r = R$ the following attributes:

$\mathbf{Up_N(n)}$: the number of neighbor nodes linked to n and belonging to the next upper μ interval,

$\mathbf{Down_N(n)}$: the number of neighbor nodes linked to n and belonging to the next lower μ interval,

$\mathbf{Up_E(n) \in \{0,1\}}$: a flag telling if n is a "maximal" terminal node ($Up_E(n) = 1$) or not ($Up_E(n) = 0$),

$\mathbf{Down_E(n) \in \{0,1\}}$: a flag telling if n is a "minimal" terminal node ($Down_E(n) = 1$) or not ($Down_E(n) = 0$).

Then the following relations can be defined at level of resolution R:

- if $Up_N(n) = 0$ then n is a "maximal" terminal node, $Up_E(n) = 1$ and $Down_E(n) = 0$,
- if $Down_N(n) = 0$ then n is a "minimal" terminal node, $Up_E(n) = 0$ and $Down_E(n) = 1$,
- if $Up_N(n) = Down_N(n) = 0$ then the graph at maximal resolution R is represented by one unique root node and $Up_E(n) = Down_E(n) = 1$.

Thus we have a local topological information on each node at the finest level of resolution R.

At each lower level of resolution $r < R$, topological attributes are iteratively added (cf. Figure 4). Assuming the node m at level of resolution $r < R$:

- $Up_N(m) = \sum_{n \in \{children\ of\ m\}} Up_N(n)$,
- $Down_N(m) = \sum_{n \in \{children\ of\ m\}} Down_N(n)$,
- $Up_E(m) = \sum_{n \in \{children\ of\ m\}} Up_E(n)$,
- $Down_E(m) = \sum_{n \in \{children\ of\ m\}} Down_E(n)$.

Based on these rules, at the root resolution $r = 0$ of aMRG, we obtain $Up_N = Down_N =$ *# edges at maximal resolution r = R*. The information contained in Up_N and $Down_N$ are redundant and then miss relevancy. Therefore we propose the following additional rule at level of resolution R:

- if n is a terminal node then we set $Up_N(n) = 0$ and $Down_N(n) = 0$,

and we obtain the relation

$$\#edges \geq max(Up_N + Down_E, Down_N + Up_E).$$

Iterative additions of the topological attributes lead to a characterization of the descendant subgraph complexity of each node. In deed, assuming $\sigma = Up_N(m) + Down_N(m) + Up_E(m) + Down_E(m)$. If at resolution $r < R$, $\sigma \gg 1$ then node m has a lot of descendant nodes (children and grandchildren), and if $\sigma \to 1$ then m has few descendant nodes, and if $\sigma = 1$ then m is a terminal node.

Our study has shown topological attributes as efficient descriptors of the graph topology. For example, the study of the aMRG at resolution level $r = 5$ of a slightly deformed cube (cf. Figure 5) returns:

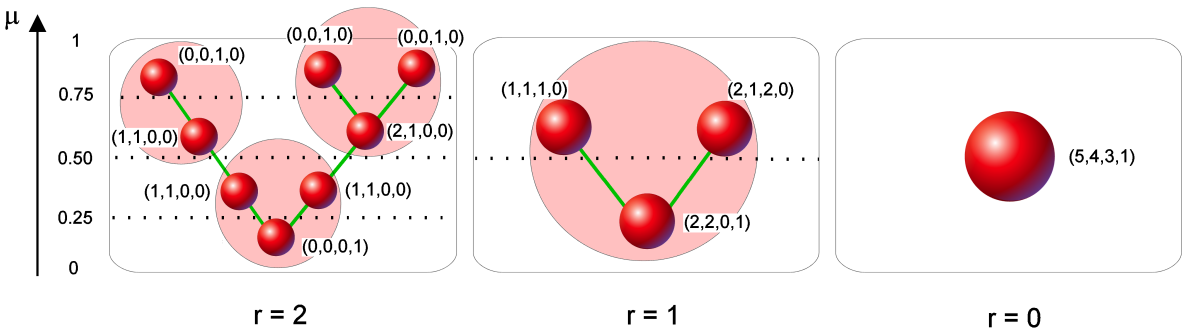


Figure 4. **Topological attributes.** Left: at resolution level $r = 2$, attributes $(Up_N, Down_N, Up_E, Down_E)$ are introduced to describe the local topology of each node. Middle: at lower level of resolution $r = 1$, topological information is cumulated. Right: at root ($r = 0$) all the topological attributes are cumulated. The embedded values characterize the global shape of the descendant subgraphs.

- $Up_N(m) = 22$,
- $Down_N(m) = 20$,
- $Up_E(m) = 8$ corresponding to the 8 cube vertices,
- $Down_E(m) = 6$ corresponding to the 6 cube faces.

Topological attributes allow to count the number of graph edges. In deed this number depends on the number of μ intervals which divide the object surface. Therefore the choice of the function μ and in particular its normalization have a great impact. The normalization of μ with μ_{max} reduces its dynamic span to $\mu_N \subset [0, 1]$ and then the number of possible intervals of μ on the objects as well. The normalization of μ with $\mu_{max} - \mu_{min}$ gives a complete dynamic span on $[0, 1]$ and then allows to reach the maximal number 2^R of μ intervals obtained by the dichotomic partitioning (considering a multiresolution graph construction at finest level of resolution $r = R$).

Then, it is possible to use the topological parameters at $r = 0$ to establish a 3D shape classification. For example, we obtain:

Star-like objects (hand, human model) : the more the object has appendages, the more it has "maximal" terminal nodes, and the bigger Up_E is. Therefore we have the following relation between maximal and minimal terminal nodes: $Up_N > Down_N$.

Elongated objects (cylinder) : μ_{max} lies at both extremities and μ_{min} is in the barycentric area. The graph lies in one branch and $Up_N = Down_N$, $Up_E = 2$ and $Down_E = 1$. Moreover the longer the object is, the more $(Up_N = Down_N) \rightarrow 2^R$.

Compact objects (sphere) : $\mu_{max} \sim \mu_{min}$ and the dynamic span of μ_N normalized with μ_{max} is very small. The graph has only few nodes, even at high resolution ($R = 5$). Up_N, $Down_N$, Up_E and $Down_E$ are small.

Polyhedral convex objects : minimal and maximal μ values lie on the polyhedral planar face centers and vertices respectively. If the maximal resolution level R is high enough, Up_E and $Down_E$ are close to the number of vertices and planar faces respectively, as shown in the example with the deformed cube (cf. Figure 5). $Up_E \sim \#$ vertices and $Down_E \sim \#$ faces.

The shape description is intuitive and completely topological. Its efficiency has been tested as described in the next section.

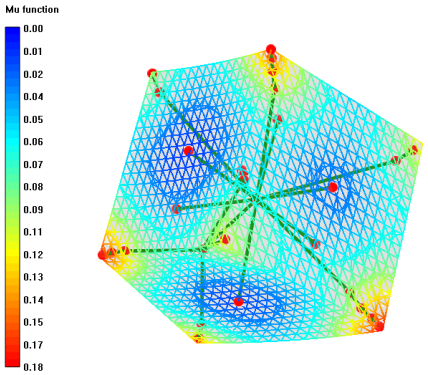


Figure 5. **aMRG of a deformed cube at** $R = 5$**.** μ is minimal at face centers and maximal on vertices.

3.3. Shape matching in 3D video

Assuming a 3D video sequence of an articulated body (e.g. a Japanese dancer [12]). It can be considered as a set of 3D mesh models with a continuous deformation between each consecutive frame.

The shape of the 3D models with its remarkable topology suits very well to a topological shape descriptor as the Reeb graph, especially when augmented with the topological features Up_N, $Down_N$, Up_E and $Down_E$ (cf. previous

Section 3.2). Therefore the aMRG can be used as a relevant 3D motion descriptor able to retrieve the similar postures of a choreography (cf. Figure 6).

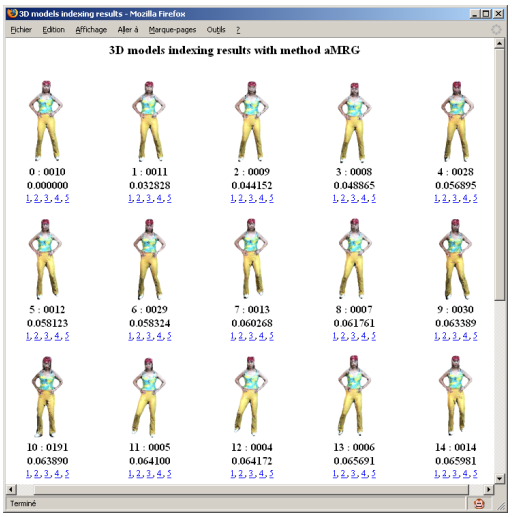


Figure 6. **Application of the aMRG as a motion descriptor in a 3D video sequence.** Similar poses of choreography can be retrieved in a 3D video sequence. Here, the query is the model on top-left and the video contains 334 frames.

Based on these results, we propose an original framework to compactly describe the motion of human model captured during a 3D video sequence.

4. 3D video compression

In this section we present the different steps of our automatic 3D video compression strategy. The 3D mesh model motion description by topology matching leads to a compact augmented Reeb graph representation. The data size is almost the storage cost one mesh model.

The pipeline of the proposed method is the following:

1. Compute all aMRG graphs of the 3D video sequence at all resolutions.

2. Using aMRG graphs properties, find similar nodes (by topology matching) for every consecutive frames .

3. The time varying 3D positions of each node of each frame of the sequence are stored in a list and associated to corresponding nodes from the first graph. A submesh is associated to each node as well.

4. Using the aMRG and its ability to retrieve similar frames, it is possible to reduce the node coordinate storage (by using index instead of coordinates for redundant postures).

5. The sequence can be simply reconstructed using submesh transformations (rotation+translation) with respect to the Reeb graph structure and the node position lists.

4.1. From topology matching to node tracking

The most critical part of the study is the node tracking. Especially, how to cope with topological structure changes is an important and difficult problem to be solved.

Our strategy relies on the node matching between aMRGs of consecutive frames using the topological attributes $(Up_N, Down_N, Up_E \text{ and } Down_E)$ described in Section 3.2.

First, the matching starts with the terminal nodes, which can be deduced using Up_E (equal to 1 at $r = R$) and geometrical attributes (the node coordinates, as in [18]). Afterwards a unique label is propagated from the matched nodes along the graph branches and then only nodes having the same labels are matched. Assuming the dynamic 3D model has a well known topology with limited deformations, simple rules are defined to match the critical nodes between the consecutive frames even in case of topological structure changes.

For instance, the surface topology of a human model is characterized by five meaningful singularities, standing for the head, the two hands, and the two feet. If a hand is close to the body, then a loop can appear in the graph representation, whereas a terminal node will disappear. However, the loop will contain a new node with a greater valence. This node stands for the hand and hence can be matched to a previous similar terminal node.

Assuming $(Up_E = Up_{E\,Head} + Up_{E\,Hands} + Up_{E\,Feet}) \in [2,5]$ with $Up_{E\,Head} = 1$, $Up_{E\,Hands} \in [0,2]$, and $Up_{E\,Feet} \in [1,2]$. Up_E is obtained from the topological attributes at $r = 0$. The nodes corresponding to the head and to the feet can be easily located at $r = R$ using the geometrical attributes (e.g. extremal values). Thus $Up_{E\,Feet}$ and afterwards $Up_{E\,Hands}$ can be deduced. As terminal nodes are identified, they are matched together (head with head, hands with hands, etc.). Labels are propagated to ensure the correct matching of the branches. Matching rules are then set up to cope with the different possible configurations. In particular, multiple matchings are allowed. As illustrated in Figure 7, if $Up_{E\,Hands} = 0$ then a node intersecting two loops stands for the two hands, if $Up_{E\,Hands} = 1$ then a node with attributes $(0,0,1,0)$ stands for one hand, etc. Intermediate nodes are matched using labels and geometrical attributes [18]. The process ends when all nodes are matched.

Note that the labelling brings the necessary information to reduce the complexity of the graph matching, avoiding an *NP*-complete problem computation.

Consecutive graph matchings from frame to frame start

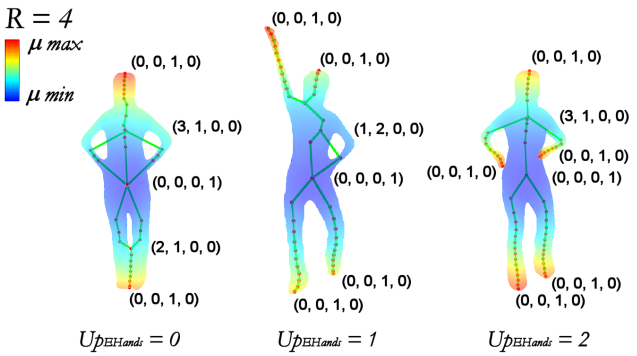


Figure 7. **Topology matching.** During the matching process, topological attributes are used to match the nodes between consecutive frames. In particular they can cope with topology changes.

at $t = 0$. As a consequence we are able to track any part of human model during the sequence.

To improve the topology change management, kinematic structure acquisition [14] could be used. This further step would require more tests with more complex shapes.

4.2. Mesh reconstruction from aMRG

Several sophisticated techniques exist to produce interactive surface deformation [11], as well as well-known softwares (e.g. 3D Studio Max, Maya). To reconstruct a deformed mesh driven by a graph, nodes are assimilated to control points and ad-hoc transformations are applied to the submeshes associated to the nodes. Mesh deformations are cast sequentially as nodes are manipulated.

Based on these ideas, we propose to combine the 3D mesh model deformations with the Reeb graph node positions. Assuming a 3D video record starting at time $t = t_0$, the submeshes associated to each component are stored as node attributes.

In order to automatically recover the 3D submesh transformations driven by the graph evolution between each frame, we propose to exploit the mesh surface partition order provided by the function μ values. The orientation is given by values of μ which increase as components are far away from the center, as the minimal value of μ lies in an area close to the center. Assuming I intervals of μ partition, N_i^t is the node in interval $[\mu_i]$ at time t. N_i^t is linked to n_{i-1} neighbor nodes in its lower bound at $[\mu_{i-1}]$. Then, the position of each vertex of the mesh associated to $N_i^{t+\delta t}$ at time $t + \delta t$ can be deduced by applying the following transformations to vertices at t:

- Translation

$$\begin{aligned} \mathbf{T} &= \frac{1}{n_{i-1}} \sum_{neighbors \in [\mu_{i-1}]} \mathbf{N_{i-1}^t N_{i-1}^{t+\delta t}} \\ &= \overline{\mathbf{N_{i-1}^{t+\delta t}}} - \overline{\mathbf{N_{i-1}^t}}, \end{aligned} \quad (2)$$

- Rotation R with respect to $\overline{N_{i-1}^{t+\delta t}}$ of angle $(A, \overline{N_{i-1}^{t+\delta t}}, N_i^{t+\delta t})$

where $A = \mathbf{T} + N_i^t$.

The figure 8 illustrates the local transformation to apply to the mesh vertices. The mesh subdivision depends on μ values and graph resolution. Attention should be paid to the first frame which should contain maximal number of terminal nodes. A non optimal choice of the first frame could produces degenerated frames.

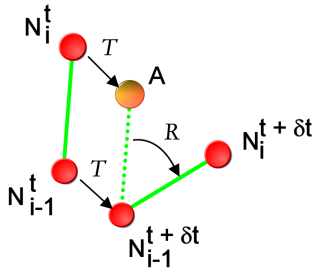


Figure 8. **Graph-driven submesh deformation.** The deformed submesh associated to the node $N_i^{t+\delta t}$ at time $t + \delta t$ can be deduced by applying a translation T and a rotation R to the submesh associated to the node N_i^t.

This reconstruction method is quite basic, but allows a quick previewing of the reconstructed 3D video. Obviously an additional post-processing step could be applied to the mesh surfaces to smooth the discontinuities between the connected submeshes, and mesh edition techniques can be adapted to improve the surface rendering [15]. In addition, as the mesh topology is the same for the whole sequence, only one texture map (one color per point or one texture patch per triangle) is sufficient to reproduce a fully textured sequence. The generation of texture map resulting from multiple view points is detailed in [7, 12]. The rendering with colors improve the visual effects as shown in the next section.

5. Experiments

The proposed approach has been implemented as a fully automatic system. 3D video data were obtained using a PC cluster as described in Section 2. A sequence is first recorded in real-time, and afterwards a 3D deformable model is applied to each frame. One convergence calculation requires approximatively 1 hour of computation. A resulting mesh of $\sim 23,000$ triangles has a size of ~ 450 Ko. Therefore 1 minute of 3D video recorded at 10 fps requires ~ 250 Mo. Note that additional post-processing can be applied to produce smooth meshes as intermediate frames, by which the mesh structure is kept the same for all 3D video frames [12]. This procedure suits well to our compression method as no topological issue has to be managed. Besides

it increases dramatically the computation time. Therefore this step was not included to our experimental tests.

Our compression scheme requires the aMRG computation of the 3D models of all frames as a pre-processing step. Considering the maximal construction resolution $R = 4$, the resulting graph of a human model has $n \sim 50$ nodes at the highest level. The computation of one aMRG up to $R = 4$ is performed in ~ 25 s, including the addition of a submesh, and geometrical and topological attributes, and depending on the quality of the data. aMRG calculations were performed on a laptop with Pentium(R) M processor 1.60 GHz and RAM 512 Mo. The calculation of the function μ remains the most time consuming, even with the Dijkstra coding scheme of $O(NlogN)$ complexity on N vertices. Our experiments have pointed out the importance of the choice of the function μ. As some data present artifacts on the surface, a robust function μ is necessary to extract exploitable graphs. We observe that the chosen μ is a good compromise. The local integral property can cope with surface noise and the computational time is acceptable. In addition the function suits well human model surface. Besides, rough meshes (triangle soups) and point clouds can be managed as well. The 3D object surface is intersected with an octree structure and a 3D fast-marching method is applied to compute the geodesic distances.

During the encoding step, similarity calculations between consecutive frames are performed. Nodes of the first aMRG at $t = 0$ are tracked along the sequence and their coordinates are reported in lists. This step requires ~ 2 s for 10 frames. The total size of a compressed sequence is $1\ mesh + n * T * (3 * c)$ where n is the number of nodes at $R = 4$, T is the number of frames, and c a coordinate value size (4 bytes). The formula is almost linear as n does not vary much. Color information can be coded as a texture map. In our experiments, a texture map (in PNG format picture) associated to $23,000$ triangles requires ~ 650 Ko. Therefore 1 minute of compressed 3D video recorded at 10 fps requires only $(630$ Ko + 650 Ko$) \sim 1.3$ Mo (vs. ~ 250 Mo), meaning an effective storage cost reduction of 99.5%. Moreover as said in Section 4, the coordinate storage can be optimized by coding the redundant node positions.

The proposed decoding step consists on reconstructing all the frames by recovering the successive node transformations. This step requires ~ 1 s for 10 frames (or ~ 1 minute for 1 minute of 3D video). Although this step does not produce an optimal surface, our experiments show that the reconstructed sequences are reliable to the original one. 3D model surfaces are reconstructed with a surface overlap error $\epsilon < 5\%$ (cf. Figure 9). However as said in Section 4.2, this reconstruction step can be improved using recent sophisticated mesh edition techniques as [15], and the rendering with colors reduces visual artifacts. Figure 10 illustrates different steps of the whole process.

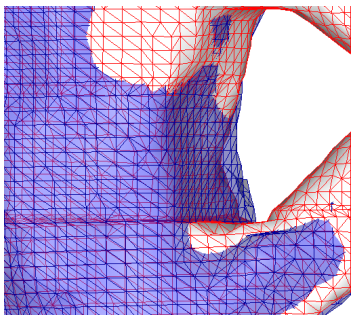


Figure 9. **Surface reconstruction results.** 3D mesh model from an original sequence (blue) is compared to a reconstructed model (red). The global shape of the model has been well recovered. Surface overlap error is estimated to $\epsilon < 5\%$.

6. Conclusion

This paper presents an original technique to compress 3D video data of an articulated body. 3D video is an innovative technology which produces time varying high quality 3D mesh sequences recorded at video speed. The applications are numerous and the research field very recent. One big issue to solve is the huge cost storage which increases linearly with the number of frames. Therefore we propose a novel application of the augmented multiresolution Reeb graph to store the data and recover the sequence with low-loss quality .

Additional topological attributes allow to track efficiently the similar nodes along the sequence. Moreover the challenging problem of topological structure changes is managed. Finally the 3D video data size is reduced to a single enhanced Reeb graph. The storage is almost the cost of only one mesh, and eventually one single texture map. The compression of the 3D video sequence is fully automatic. The 3D video encoding and decoding results are reliable although the reconstruction step is not optimized. Our approach is very promising and invites to investigate for a general matching scheme to cope with complex shapes. To our knowledge, this is the first automatic 3D video compression approach proposed in the literature.

References

[1] P. Alliez and C. Gotsman. Recent advances in compression of 3d meshes. *Advances in Multiresolution for Geometric Modelling. N.A. Dodgson, M.S. Floater, M.A. Sabin. Springer-Verlag editors*, pages 3–26, 2005.

[2] H. Briceno, P. Sander, L. McMillan, S. Gortler, and H. Hoppe. Geometry videos: A new representation for 3d animations. *ACM Symposium on Computer Animation*, pages 136–146, 2003.

[3] N. Cornea, M. F. Demirci, D. Silver, A. Shokoufandeh, S. Dickinson, and P. Kantor. 3d object retrieval using many-to-many matching of curve skeletons. *IEEE International*

Conference on Shape Modeling and Applications, pages 368–373, 2005.

[4] J. Franco, C. Menier, E. Boyer, and B. Raffin. A distributed approach for real-time 3d modeling. *IEEE International Conference on Computer Vision and Pattern Recognition Workshop*, page 31, 2004.

[5] J. Goldak, X. Yu, A. Knight, and L. Dong. Constructing discrete medial axis of 3-d objects. *International Journal of Computational Geometry and Applications*, 1(3):327–339, 2001.

[6] H. Habe, Y. Katsura, and T. Matsuyama. Skin-off: Representation and compression scheme for 3d video. *Picture Coding Symposium*, 2004.

[7] C. Hernandez-Esteban and F. Schmitt. Silhouette and stereo fusion for 3d object modeling. *International Journal on Computer Vision and Image Understanding*, 96(3):367–392, 2004.

[8] M. Hilaga, Y. Shinagawa, T. Kohmura, and T. L. Kunii. Topology matching for fully automatic similarity estimation of 3d shapes. *ACM SIGGRAPH*, pages 203–212, 2001.

[9] T. Kanade and P. J. Narayanan. Historical perspectives on 4d virtualized reality. *IEEE International Conference on Computer Vision and Pattern Recognition Workshop*, page 165, 2006.

[10] T. Kanade, P. Rander, and P. J. Narayanan. Virtualized reality: Constructing virtual worlds from real scenes. *IEEE Multimedia*, pages 34–47, 1997.

[11] Y. Kho and M. Garland. Sketching mesh deformations. *Proceedings of the ACM Symposium on Interactive 3D Graphics*, pages 147–154, 2005.

[12] T. Matsuyama, X. Wu, T. Takai, and S. Nobuhara. Real-time 3d shape reconstruction, dynamic 3d mesh deformation, and high fidelity visualization for 3d video. *International Journal on Computer Vision and Image Understanding*, 96(3):393–434, 2004.

[13] S. Moezzi, L. Tai, and P. Gerard. Virtual view generation for 3d digital video. *IEEE Multimedia*, pages 18–26, 1997.

[14] T. Mukasa, S. Nobuhara, A. Maki, and T. Matsuyama. Finding articulated body in time-series volume data. *The 4th International Conference on Articulated Motion and Deformable Objects*, pages 395–404, 2006.

[15] S. Park and J. Hodgins. Capturing and animating skin deformation in human motion. *ACM SIGGRAPH*, pages 881–889, 2006.

[16] C. Sminchisescu, A. Kanaujia, Z. Li, and D. Metaxas. Discriminative density propagation for 3d human motion estimation. *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 390–397, 2005.

[17] J. Tangelder and R.C.Veltkamp. A survey of content based 3d shape retrieval methods. *IEEE International Conference on Shape Modeling and Applications*, pages 145–156, 2004.

[18] T. Tung and F. Schmitt. The augmented multiresolution reeb graph approach for content-based retrieval of 3d shapes. *International Journal of Shape Modeling*, 11(1):91–120, 2005.

[19] A. Veeraraghavan, A. Roy-Chowdhury, and R. Chellappa. Matching shape sequences in video with applications in human motion analysis. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, pages 1896–1909, 2005.

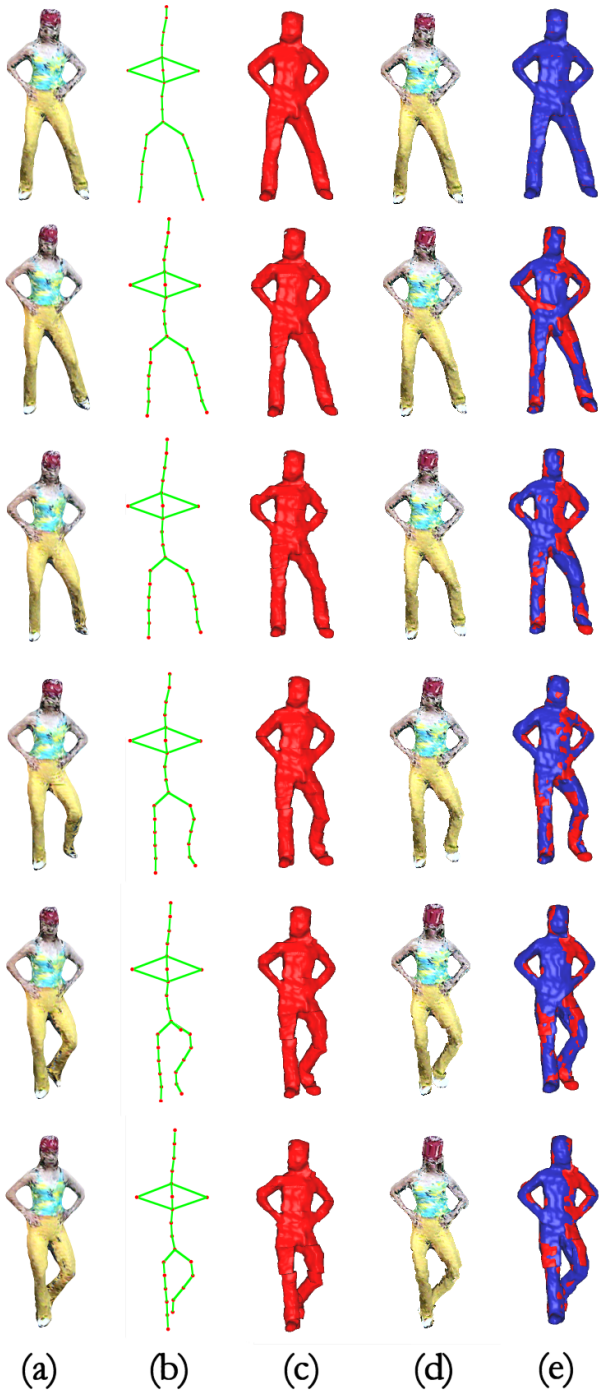


Figure 10. **Compression evaluation.** Continuous deformations of 3D shape can be tracked along a 3D video sequence using the aMRG representation. Embedded topological and geometrical information have been used for nodes similarity evaluation. Afterwards mesh reconstructions are obtained using time varying node positions and their corresponding submeshes. (a) presents initial frames. (b) shows Reeb graphs extracted at resolution level $r = 4$. (c) shows reconstructed sequences. (d) shows reconstructed sequences with texture. Finally, (e) shows the differences between the original and the reconstructed sequences.