Topology Dictionary with Markov Model for 3D Video Content-Based Skimming and Description

Tony Tung Takashi Matsuyama
Graduate School of Informatics, Kyoto University, Japan
{tung,tm}@vision.kuee.kyoto-u.ac.jp

Abstract

This paper presents a novel approach to skim and describe 3D videos. 3D video is an imaging technology which consists in a stream of 3D models in motion captured by a synchronized set of video cameras. Each frame is composed of one or several 3D models, and therefore the acquisition of long sequences at video rate requires massive storage devices. In order to reduce the storage cost while keeping relevant information, we propose to encode 3D video sequences using a topology-based shape descriptor dictionary. This dictionary is either generated from a set of extracted patterns or learned from training input sequences with semantic annotations. It relies on an unsupervised 3D shape-based clustering of the dataset by Reeb graphs, and features a Markov network to characterize topological changes. The approach allows content-based compression and skimming with accurate recovery of sequences and can handle complex topological changes. Redundancies are detected and skipped based on a probabilistic discrimination process. Semantic description of video sequences is then automatically performed. In addition, forthcoming frame encoding is achieved using a multiresolution matching scheme and allows action recognition in 3D. Our experiments were performed on complex 3D video sequences. We demonstrate the robustness and accuracy of the 3D video skimming with dramatic low bitrate coding and high compression ratio.

1. Introduction

Dynamic 3D multi-view stereo reconstruction (or 3D video) is an image recording technique which produces free-viewpoint videos of 3D models in motion [9, 19, 11, 5, 4, 2]. It is a markerless motion capture system where subjects do not need to wear special equipment. The technology can be employed in many areas of applications such as cultural heritage preservation, entertainment, sports, medicine, and so on. It requires a set of calibrated and synchronized video cameras to capture temporal series of subjects in motion at video rate. 3D mesh models are then reconstructed using multi-view stereo reconstruction algorithms [16]. In addition, textures are rendered on the reconstructed 3D object surfaces to obtain high quality visual effects (e.g. for cloth rendering). Many methods have recently focused on performance and quality improvements [20, 22]. In particular, one issue is the huge amount of disk space required which makes the dataset difficult to handle: the navigation and search for relevant information among gigabytes of data is intractable. To date, 3D video compression has been addressed regardless of content. Moreover existing techniques are not designed to cope with complex patterns (e.g. 3D shape topology changes) [7, 24].

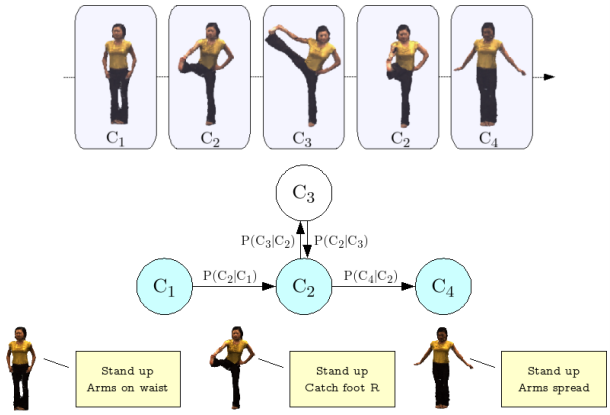

Figure 1. **3D video content-based skimming**. 3D video sequences of humans in motion contain slow motions and several repetitive poses. We introduce the dictionary of topology-based shape descriptors with Markov model to identify poses and compactly encode 3D video sequences. The proposed technique is used to perform 3D video skimming and 3D action recognition.

We propose to use a dictionary-based encoder approach [28]. It consists in searching for matches between a set of patterns contained in a data structure and the data to be compressed. As the encoder finds a match, it substitutes a reference to the data position in the data struc-

ture. Hence redundancies can be efficiently encoded. In this paper, we introduce the topology dictionary. The dictionary can be generated from a set of extracted patterns or learned from training input sequences. Pattern extraction is obtained by unsupervised clustering of the dataset using enhanced Reeb graphs. They have been designed for shape matching in large database and can perform efficient shape retrieval queries [23]. A Reeb graph is a canonical representation of the topology of the surface, and can be used to encode 3D meshes with temporal evolution [24]. Furthermore, 3D video compression and skimming is performed by segmenting the sequence and encoding the successive subsequences. The dictionary features a weighted directed graph structure where nodes (or states) represent patterns and edges characterize topological changes (state transitions). This Markov network structure allows to draw statistical information on the video content such as duration and occurrence probability of frame sets in order to preserve relevant information and remove redundant data. Then as an extension, semantic annotations are used to perform automatic description of video sequences. In addition, we take advantage of a Reeb graph multiresolution matching scheme to encode forthcoming frames and achieve 3D action recognition. When the learned dictionary is used to encode video sequences, it produces low bitrate coding with high compression ratio (cf. Figure 1).

The rest of the paper is organized as follows. The next section discusses work related to the techniques presented in this paper. Section 3 describes the Reeb graph construction algorithm. Section 4 presents the topology-based shape descriptor dictionary with semantic annotations. Section 5 presents 3D video skimming and 3D action recognition. Section 6 shows experimental results. Section 7 concludes with a discussion on our contributions.

2. Related work

The most direct way to compress 3D video (cf. Figure 2) is to apply 3D mesh compression techniques to every single frame [7, 3]. Geometry and connectivity compression can generally guarantee lossless compression quality but is not optimal since redundant information between frames is not handled. Various techniques dedicated to 3D animation sequence compression rely on basis decomposition [1, 10]. Indeed, they are dedicated to mesh sequences having the same connectivity and then cannot be applied to our data. Moreover Principal Component Analysis (PCA) methods are still limited to low resolution meshes due to the significant computing resources required. In [24], a 3D video compression strategy based on topology matching between consecutive frames was proposed. The algorithm consists in building enriched skeletons which embed 3D shape, texture, and temporal variations. The 3D video sequence is compactly encoded and can be reconstructed by skinning techniques. However the pitfall is that topological changes of 3D shapes between consecutive frames have to be managed, and so far the proposed algorithm relies on semi-heuristic matching rules which are not easy to set when facing complex topological changes.

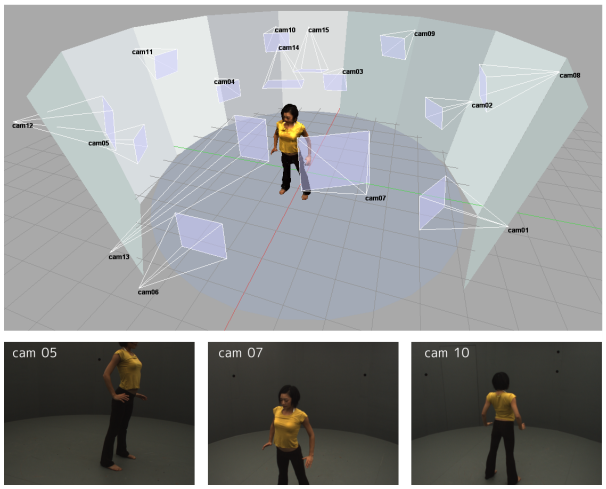

Figure 2. **3D video.** (Top) Multi-view reconstruction from video cameras. (Bottom) Corresponding video frames.

We propose a genuine content-based encoding strategy to achieve 3D video compression, skimming and description. The 3D video sequence is modeled by a topology dictionary with Markov network: a weighted directed graph where nodes represent clusters of topology descriptors, and edges represent transitions between different poses. The learning of dictionaries of feature vector clusters (or bags) have been successfully used for image categorization, segmentation and localization [26, 17, 6], and graphs have shown interesting applications for 2D video segmentation and summarization [27, 12]. Hence, we introduce the topology-based shape descriptor dictionary to learn and index 3D video patterns. In addition, semantic annotations provide automatic video description and action recognition (cf. [21, 25]). To our knowledge, although skimming techniques are successful for 2D videos [18, 14], the extension to 3D videos with automatic segmentation, encoding and reconstruction has not been treated yet.

3. Enhanced Reeb graphs of 3D models

The Reeb graph is a high level 3D shape descriptor. It is an elegant solution to analyze 3D mesh topology and shape as it gives a graphical representation of surface properties. This section gives a brief review of the augmented Multiresolution Reeb Graph (aMRG) [23, 24]. The aMRG is an enhanced version of a multiresolution Reeb graph. It embeds topological and geometrical information and enables shape matching and mesh skinning.

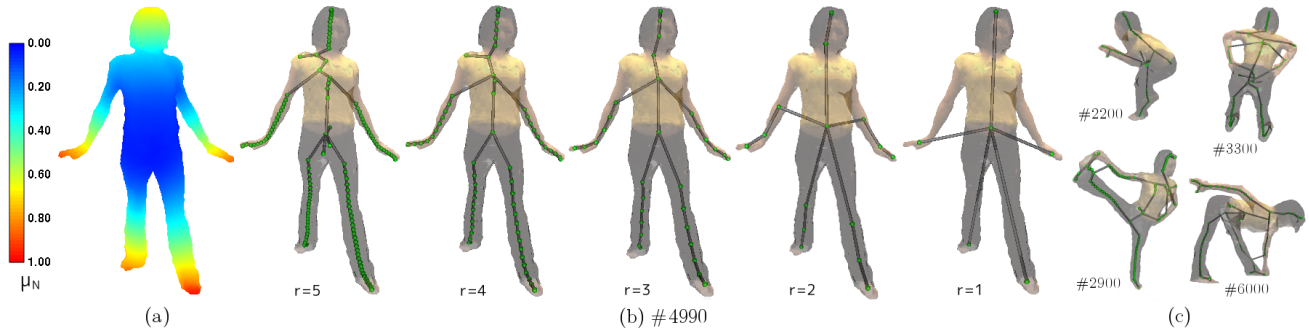

Figure 3. **Topology-based shape descriptors.** Enhanced Reeb graphs are extracted from 3D video sequences to capture topology information. (a) The Morse function μ allows to detect critical points on the mesh surface. (b) shows Reeb graphs of frame #4990 at resolutions $r = 5$ to 1. Every node embeds information related to the geometry and topology of an assigned surface region. (c) Examples of Reeb graphs for different 3D video frames ($r = 5$). As illustrated surface topology is captured by Reeb graphs.

Reeb graph construction. In our framework, we assume 3D models defined as compact 2-manifold surfaces approximated by 3D meshes. Let S be a mesh surface. According to the Morse theory, a continuous function μ defined on S characterizes the topology of the surface on its critical points. The surface connectivity between critical points can then be modeled by the Reeb graph of μ, which is the quotient space defined by the equivalence relation $\sim$:
let $\mathbf{X} \in S$ and $\mathbf{Y} \in S$, then $\mathbf{X} \sim \mathbf{Y}$ if and only if: (1) they belong to the same connected component of $\mu^{-1}(\mu(\mathbf{X}))$, and (2) $\mu(\mathbf{X}) = \mu(\mathbf{Y})$ [15]. The Morse function μ is defined as in [8]:

$$\mu(\mathbf{v}) = \int_{\mathbf{p} \in S} g(\mathbf{v}, \mathbf{p}) dS \quad \text{and} \quad \mu_N(\mathbf{v}) = \frac{\mu - \mu_{\min}}{\mu_{\max} - \mu_{\min}} \tag{1}$$

where $g(\mathbf{v}, \mathbf{p})$ is the geodesic distance on S between two points $\mathbf{v}$ and $\mathbf{p}$ belonging to S, and $\mu_N : S \rightarrow [0, 1]$ is the function μ normalized with respect to its minimal and maximal values $\mu_{\min}$ and $\mu_{\max}$. As defined, μ_N is invariant to rotation, translation and scale transformations. Moreover the integral formulation provides robustness to local surface noise such as outliers (e.g. due to 3D reconstruction errors). Extremal values of μ_N return surface critical point locations which coincide to highly concave or convex regions. The Reeb graph is then constructed by: (1) partitioning the object surface into regular intervals based on μ_N values, (2) assigning a node to every different region in each interval, and (3) linking nodes of connected regions. The level of resolution R of the Reeb graph is defined upon the number of intervals 2^R obtained by iterative subdivisions of $[0, 1]$. Lower resolution Reeb graphs are obtained by merging intervals by pairs using a hierarchical procedure where nodes are linked to a unique parent-node from the next lower resolution [8]. Thus the multiresolution Reeb graph is a set of Reeb graphs of various levels of resolution $r = 0...R$ deduced from the Reeb graph computed at the highest resolution R (cf. Figure 3).

Node features. In [23, 24], the multiresolution Reeb graph is augmented with geometrical and topological features in order to obtain efficient topology-based shape matching in large dataset of 3D mesh models. Each node of the graph embeds the relative area of its assigned region, and the connectivity layout of neighboring nodes. The following attributes are set for each graph node n at the highest resolution level: $\{U_N(n), D_N(n), U_E(n), D_E(n)\}$ standing for the number of connected nodes having a higher μ_N value, the number of connected nodes having a lower μ_N value, a binary value indicating if n has no neighbor with a higher μ_N value, and a binary value indicating if n has no neighbor with a lower μ_N value respectively. Then, at every lower resolution level, the node attributes consist in the addition of their children-node attributes. Note that ending nodes have always either $U_E = 0$ or $D_E = 0$ at any level of resolution $r > 0$. The embedded features allow to quickly classify the nodes. They are encoded together with multiresolution Reeb graph structures as feature vectors.

Similarity function. The similarity evaluation between two models M and N is obtained by computing the SIM function on the set $\mathcal{C}_r$ of pairs of topologically consistent nodes within a coarse-to-fine approach from $r = 0$ to R:

$$\mathrm{SIM}(M, N) = \sum_{r=0}^{R} \sum_{(m,n) \in \mathcal{C}_r} \mathrm{sim}(m, n) \tag{2}$$

where sim measures the difference between two node features [23]. We observe that the feature vectors do not have a fixed size (as they depend on the number of graph nodes) and the multiresolution matching scheme is crucial for tractability as it avoids the NP-complete problem of graph matching.

4. Topology dictionary

We propose to analyze the content of training 3D data to identify poses, and encode sequences using pattern references. The search is operated by Reeb graph matchings. The dataset is clustered into topology classes and a weighted directed graph is built upon the timing of the sequence as a Markov network. The overall structure stands for the topology-based shape descriptor dictionary.

4.1. Dataset unsupervised clustering

We assume a training 3D video sequence composed by a set of 3D mesh models $\mathcal{M} = \{m_1, ..., m_T\}$. Feature vectors (of aMRG) are computed for every model at resolution level R. Let $M_t \subseteq \mathcal{M}$ denote the set of feature vectors associated to an element m_t. The training dataset $\mathcal{M}$ is recursively split in subsets M_t and N_t according to a threshold τ on the SIM function (cf. Eq.(2)):

$$M_t = \{n \in N_{t-1} | \text{SIM}(m_t, n) < \tau\}, \quad (3)$$

$$N_t = N_{t-1} \setminus M_t, \quad (4)$$

where $N_0 = \mathcal{M}$. Hence for each similarity query m_t on $\mathcal{M}$, starting at $t = 1$, the closest feature vectors are retrieved and corresponding frames are indexed with the same cluster reference as m_t. If a frame has already been assigned to a cluster c_i, then it is considered as classified and is not processed subsequently. As $N_t = \emptyset$ or $t = T$, the recursive training stops and topology class distributions $P(c_i)$ are estimated. Note that if τ is underestimated, then the dataset will be overpartionned, but without any impact on the quality of the sequence reconstruction (cf. Figure 4).

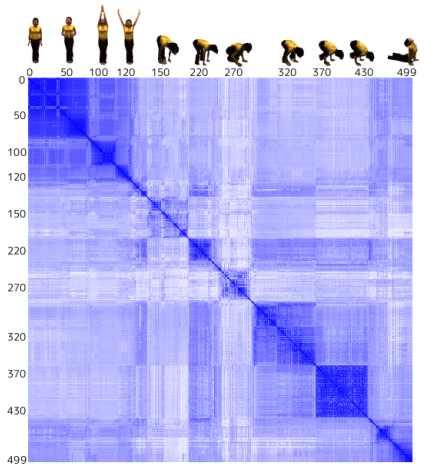

Figure 4. **Similarity matrix.** The matrix shows the similarity computation of 500 3D video frames. Dark and light colors denote strong and weak similarities respectively. The blocks show successive frames belonging to the same topology class. 220 clusters were found with a threshold $\tau = 0.1$.

4.2. Probabilistic graph structure

The structure of a video sequence can be designed as a Markov network that models the video evolution between different states (e.g. scene changes [12]). In our approach, we propose to partition 3D videos of human models into poses of different topology. We denote by N the number of clusters in the sequence $\mathcal{S} = \{s_1, ..., s_T\}$, $N_i = \text{card}(c_i)$ the size of the i^{th} cluster c_i, and $T = \sum_{i=1}^{N} N_i$ the number of frames in $\mathcal{S}$. Let assume $\mathbf{G} = (\mathbf{C}, \mathbf{E})$ is a weighted directed graph, the set of vertices $\mathbf{C} = \{c_1, ..., c_N\}$ stands for the topology-based shape descriptor clusters, and the set of edges $\mathbf{E} = \{e_{ij}\}_{i\in[1,N]}^{j\in[1,N]}$ stands for the chronological transitions which connect every cluster ($\text{card}(\mathbf{E}) < N$). $P(c_i) = \frac{N_i}{T}$ is the occurrence probability of a cluster c_i, which weights the occurrences of a pose in $\mathcal{S}$. The weight w_{ij} is associated to the edge $e_{ij} \neq \emptyset$ and models the transition probability between the two states c_i and c_j. w_{ij} is defined as the conditional probability

$$P(c_j|c_i) = \frac{\sum_{(s_p,s_q)\in c_i \times c_j} \delta(q-p)}{\sum_{(s_p,s_q)\in c_i \times \mathbf{C}\setminus c_i} \delta(q-p)}, \quad (5)$$

where s_p is the p^{th} frame of $\mathcal{S}$ and $\delta(x) = \begin{cases} 1 & \text{if } x = 1 \\ 0 & \text{else} \end{cases}$. The probability is normalized so that assuming $\mathcal{N}_i \subset \mathbf{C}$ is the set of adjacent clusters in the neighborhood of c_i, $\mathcal{N}_i = \{c \in \mathbf{C} \setminus c_i | \exists (s_p, s_q) \in c_i \times c, q - p = 1\}$, then $\sum_{c_j \in \mathcal{N}_i} P(c_j|c_i) = 1$. The evolution of the model shape can be analyzed along $\mathcal{S}$ using the graph $\mathbf{G}$ (cf. Figure 5). As well, the structure allows to design new sequences by navigating through the graph.

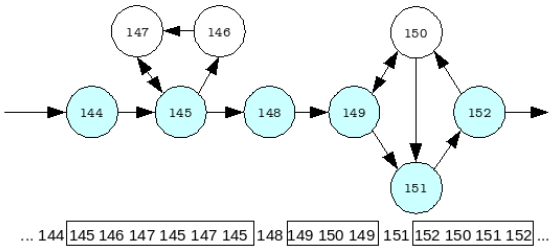

Figure 5. **Probabilistic graph structure.** The sequence is represented as a Markov network. It models transitions between the different states and allows to analyze the sequence content. For example, cycles stand for redundant poses.

4.3. Semantic description

Human models have an articulated body with constraints and a finite number of positions. In order to achieve automatic semantic description of 3D video sequences, training data models are depicted by semantic annotations (e.g. “stand up, hands on hips”, “stand up, hands joined over the

head, head looking the hands", etc.). Any additional pose with annotations can be learned and indexed in the dictionary. Hence similar poses of a model can be retrieved in 3D videos by queries on shape and/or semantic (cf. Figure 6). In our framework, 20 poses of yoga and from various sources where annotated for 3D action recognition purpose.

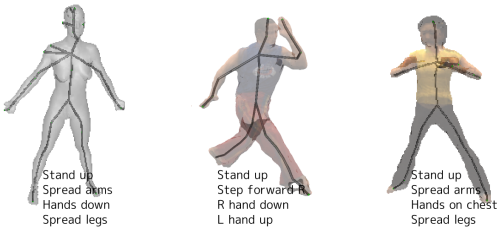

Figure 6. **Training data with annotations.** Any human poses with annotations can be learned and indexed in the topology dictionary (e.g. 3D models found on the web).

5. 3D video skimming and description

3D videos contain slow motions, long poses and repetitive actions. We propose to use topology information to identify the poses and efficiently encode the sequences. We take advantage of the Markovian dictionary structure to perform 3D video skimming by probabilistic discrimination of frames. In addition automatic semantic description and 3D action recognition is made possible with annotated training data.

5.1. Content-based encoding

We propose a linear process to compactly encode 3D video sequences. Assuming the sequence $\mathcal{S} = \{s_1, ..., s_T\}$, a keyframe k_i is created for every cluster $c_i \subset \mathcal{S}$. k_i contains one graph structure with a textured submesh associated to each node, and the coordinates of all the graph nodes of the frames $\{s \in c_i\}$. Indeed each keyframe encodes a specific pose of the model and its variants with respect to similarity constraints. If a visited frame s_t does not belong to the same cluster c_{i-1} as the previous frame s_{t-1}, then a new keyframe k_i is created. If s_t belongs to the same cluster c_{i-1} as s_{t-1}, then only the graph node coordinates of the model at t are encoded into k_{i-1}. Thus it is possible to recover the node transformations between consecutive frames and reconstruct the mesh sequence without topology matching issues. Let m be the size of an encoded mesh with a Reeb graph structure, and g be the size of an encoded set of node positions, then the size of the compressed sequence is $\sigma \simeq m * N + g * T$.

To further compress the sequence, our strategy is to identify and skip isolated and redundant patterns using the dictionary properties. First, the cluster set $\mathbf{C}$ is sorted with respect to the weights $P(c_i)$ in order to find the frames belonging to clusters having the highest and lowest probabilities. Then:
- if $P(c_i) \gg 0$, then c_i may either contain: (1) a long sequence of successive frames representing a long pose with slow or low variations which can be shorten by skipping intermediate frames (e.g. frames #370 to #430 in Figure 4), or (2) a recurrent pose modeled as a cycle junction node in the graph $\mathbf{G}$. Frames are identified by repetitive sequences in the sequence timeline. Each cycle $\mathcal{L} \subset \mathbf{C}$ is weighted according to its *size* $S(\mathcal{L}) = \frac{\sum_{c \in \mathcal{L}} P(c)}{\text{card}\{c \in \mathcal{L}\}}$ and *relevance* $P(\mathcal{L}) = \prod_{e_{ij} \in \mathcal{L}} P(c_i|c_j)$. Cycles are removed from $\mathbf{G}$ according to $S(\mathcal{L})$ and $P(\mathcal{L})$.
Indeed if $P(c_i|c_j) = 0$, then c_i and c_j are disjoint ($e_{ij} = \emptyset$), if $P(c_i|c_j) = 1$, then the state space of the transition from c_j is reduced to the singleton $\{c_i\}$, and else if $P(c_i|c_j) \in]0, 1[$, then c_j is a cycle junction node.
- if $P(c_i) \ll 1$, then c_i contains few frames. We identify isolated patterns and reclassify the frames into an adjacent cluster (e.g. in the sequence $\{c_i, c_i, c_j, c_i, c_i\}$, $\{s \in c_j\}$ are reclassified in c_i).
Hence content-based skimming can be processed iteratively until a threshold on the compression rate is reached. Note that skimming small cycles is equivalent to skip short actions.

5.2. 3D action recognition

The encoding of new frames is processed as a mapping problem. A classifier assigns a topology class (or cluster) to new unclassified feature vectors. The process relies on the Markov network of the learned dictionary $\mathbf{G} = (\mathbf{C}, \mathbf{E})$ and the multiresolution Reeb graph matching scheme. As a new frame s_{T+1} is added to the sequence $\mathcal{S} = \{s_1, ..., s_T\}$, a cascade of classifiers ordered by relevance of probability is built from the cluster weights. Let assume s_T belongs to the cluster $c_i \subset \mathbf{C}$, and let denote the sets of adjacent clusters $\mathcal{N}_+(c_i) = \{c \in \mathbf{C} | P(c|c_i) > 0\}$, and $\mathcal{N}_-(c_i) = \{c \in \mathbf{C} | P(c_i|c) > 0\}$. s_{T+1} is successively compared to: (1) c_i, (2) $\{c \in \mathcal{N}_+(c_i)\}$ starting with higher probabilities $P(c|c_i)$, and (3) $\{c \in \mathcal{N}_-(c_i)\}$ starting with higher probabilities $P(c_i|c)$. At each step, the similarity of s_{T+1} is evaluated against a model s_c (the keyframe) belonging to the visited cluster c. The search of c is performed until a match is found, as $\text{SIM}(s_c, s_{T+1}) < \tau$ (cf. Eq.(2) and Figure 7). Annotations associated to c are then displayed to the user (i.e. the action is recognized). If no positive classification has occurred after the step (3), a new cluster containing s_{T+1} is created and linked to c_i. Alternatively, the search space can be extended to the neighbors of the visited clusters and so on recursively. As well, a depth of search can be set in order to limit the number of classifiers. Note that each similarity evaluation is achieved using a mul-

tiresolution matching scheme. Hence unsimilar shapes are quickly rejected and the mapping process is not particularly time-consuming.

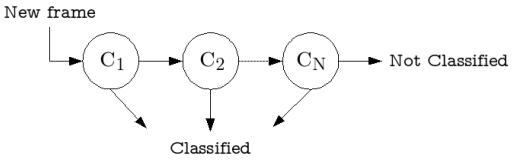

Figure 7. **Mapping with a cascade of classifiers.** A cascade of classifiers ordered by relevance weights is built as a new frame is added to the sequence.

5.3. Sequence reconstruction

The 3D video reconstruction from compressed frames relies on a mesh skinning technique where Reeb graphs serve as skeletons and submeshes are associated to each node. As models of a same class have the same topology which is represented by a unique skeleton, topology change issues are avoided during the skinning process [24]. An initial mesh M_i is assigned to each cluster c_i as a *keyframe* (cf. Section 5.1), and every 3D video frame belonging to c_i is reconstructed by deforming M_i according to recovered node transformations. By measuring 3D position distorsions on a $400\times400\times400$ voxel grid (corresponding to a $2\text{m}\times2\text{m}\times2\text{m}$ volume at resolution 5mm), we obtain $MSE \sim 0.005$ and $PSNR \sim 75dB$, which stand for the mean squared error and the peak signal-to-noise ratio respectively. In our experiments the implementation of the reconstruction step was not optimized. However due to the video rate (25fps), no major reconstruction artifacts were noticed.
Note that intraclass interpolations of graph node coordinates are necessary during the reconstruction process to ensure smooth transitions as frames are dropped during the skimming process. Fortunately this step is straightforward (e.g. using quaternions and spherical linear interpolations to compute transformations).

6. Experimental results

To evaluate the performance of our approach, we have tested our algorithm on real 3D video sequences. In particular, this paper is illustrated with yoga sequences. This dataset is interesting and challenging as it consists in succession of various complex human poses. The model turns several times during a session, making action recognition from a single viewpoint very limited. In this case, 3D shape understanding can be efficiently performed by topology description. The sequences contain 7500 frames acquired by a set of multiple video cameras at 25fps. Every frame contains a 3D mesh of 30000 triangles with texture. Hence an uncompressed frame encoded in standard (ASCII) format requires 1.5Mb, which means 11.25Gb for 7500 frames. Feature vectors were computed on a Core2Duo 3.0GHz 4Gb RAM, nevertheless it requires less than 512Mb RAM. The current unoptimized implementation in C++ takes 15s to generate a feature vector at resolution level $R = 4$. Efficient computation of Reeb graphs can be found in the literature (e.g. [13]). The similarity computation between two models takes 10ms. Although our algorithm contains several independent steps, it has been fully automatized.

Topology dictionary stability. The core of our approach relies on the ability of the dictionary to discriminate shape topology. In particular, the definition of the Morse function and the similarity measure are crucial (cf. Section 3). The sensitivity of the dictionary has been evaluated against different Morse functions and different resolutions R of Reeb graphs (cf. Figure 8). The integral geodesic function with $R = 4$ has shown the best trade-off between clustering performance (quality) and computation time. Different formulations of the similarity measure SIM have been tested. For example without summation of coarser resolution contributions (cf. geodesic $r = 4$ on Figure 8). As well, the clustering power of the topology descriptor depends on the threshold $\tau \in [0, 1]$ (as any models which similarity score is smaller than τ belong to the same topology class). We have observed similar behavior of the dataset clustering with respect to τ using sequences of different lengths (cf. Figure 9). This helps to set τ and check the validity of training datasets. By experience, dataset of human models are well clustered with $\tau = 0.1$.

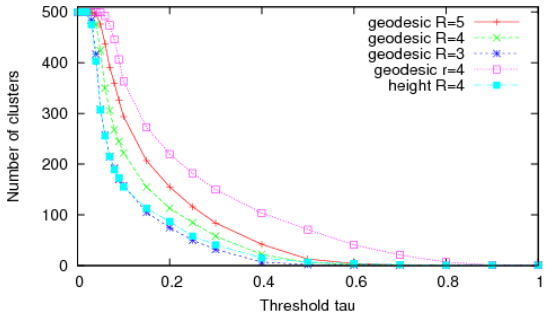

Figure 8. **Comparison of different similarity measures for the clustering of 500 frames with respect to τ.**

3D video compression and skimming. An uncompressed sequence size is growing linearly of 1.5Mb per frame (11,250Mb for 7500 frames). Hence it becomes very difficult to search for specific information or navigate in long sequences. As presented in Section 5.1, our 3D video encoding process consists in two steps. First, long poses, slow motions and variations belonging to the same topology class are automatically located using the weights of the

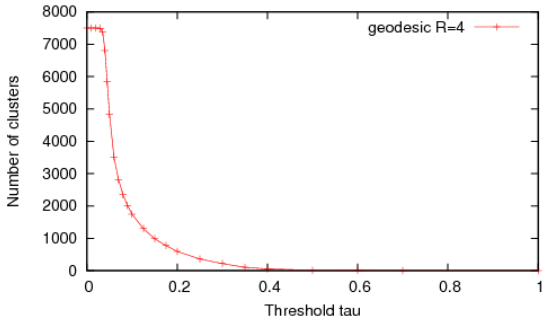

Figure 9. **Clustering of 7500 frames with respect to τ.**

dictionary clusters, and then efficiently compressed. The topology-based compression step achieves a compression ratio of 2:1, meaning a saving space of 50% (cf. Figure 10). The 7500 frame sequence has been reduced to 3660 encoded frames and 1749 clusters were found. Intraclass reconstruction ensures accurate recovery of the sequence. Second, using the dictionary graph structure, 656 cycle junction nodes have been identified (cf. Figure 11). The skimming of short actions ($<$ 2s) produces a 3D video sequence of 2716 encoded frames, which is equivalent to a compression ratio of 3:1 and a saving space of 66%. Another possible skimming scheme consists in successively skipping the biggest cycles. It returns a sequence of 1439 encoded frames (the ratio is 5:1 and a saving space of 80%). Intraclass interpolations ensure seamless frame transitions. Indeed the sequence can be automatically or progressively reduced while keeping relevant information. Appropriate file size, poses to keep or to skip can be chosen.

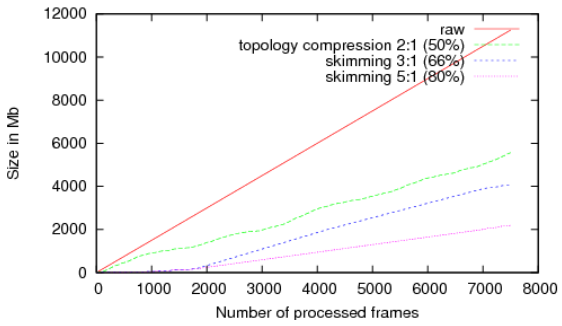

Figure 10. **Content-based compression and skimming gain.**

3D action recognition. Automatic 3D video description and 3D action recognition of new frames are performed using the topology dictionary (cf. Figure 12).

7. Conclusion

The contributions of this paper consist in the introduction of a topology dictionary with Markov model to perform 3D video content-based compression, skimming, and 3D action recognition. A 3D video consists in a stream of 3D models. Long sequences require several gigabytes of disk space, and the navigation and information retrieval in huge datasets are intractable. Hence we propose to use a topology dictionary to reduce the storage cost of 3D videos while keeping relevant information. As a matter of fact, slow motions and repetitive poses occur frequently in sequences of humans in action. Topology description has shown its robustness to extract similar patterns. Thus the sequence can be compactly encoded with pattern references. In addition, the 3D video sequence is modeled by a weighted graph having Markovian property. The skimming of 3D video is then possible by probabilistic discrimination of frames. We show results for seamless skimmed videos with compression ratio up to 5:1, meaning a saving space of 80%. Furthermore the dictionary features learned annotated poses, allowing to perform automatic semantic description of videos and 3D action recognition of forthcoming frames. We believe the topology dictionary will bring lots of perspectives to future 3D video research and applications.

Acknowledgement

The authors would like to gratefully thank Pr. Francis Schmitt from TELECOM ParisTech for the precious and insightful discussions.

References

[1] M. Alexa and W. Müllen. Representing animations by principal components. *Computer Graphics Forum*, 19(3), 2000.

[2] J. Allard, C. Ménier, B. Raffin, E. Boyer, and F. Faure. Grimage: Markerless 3d interactions. *SIGGRAPH - Emerging Technologies*, 2007.

[3] 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.

[4] K. M. Cheung, S. Baker, and T. Kanade. Shape-from-silhouette across time: Part ii: Applications to human modeling and markerless motion tracking. *IJCV*, 63(3):225–245, 2005.

[5] J. Franco, C. Menier, E. Boyer, and B. Raffin. A distributed approach for real-time 3d modeling. *CVPR Workshop on Real-Time 3D Sensors and their Applications*, page 31, 2004.

[6] B. Fulkerson, A. Vedaldi, and S. Soatto. Localizing objects with smart dictionaries. *ECCV*, 2008.

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

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

[9] T. Kanade, A. Yoshida, K. Oda, H. Kano, and M. Tanaka. A stereo machine for video-rate dense depth mapping and its new applications. *CVPR*, 1996.

[10] Z. Karni and C. Gotsman. Compression of soft-body animation sequence. *Computers & Graphics*, 28:25–34, 2004.

Figure 11. **3D video skimming.** Topology is used to locate the similar frames in 3D videos. Frames #1783, #4179 and #6987 belong to the same topology class. Hence: (1) they are encoded and reconstructed using a unique mesh model, (2) they are represented by cycle junction nodes in the topology dictionary and all intermediate frames can be skimmed seamlessly.

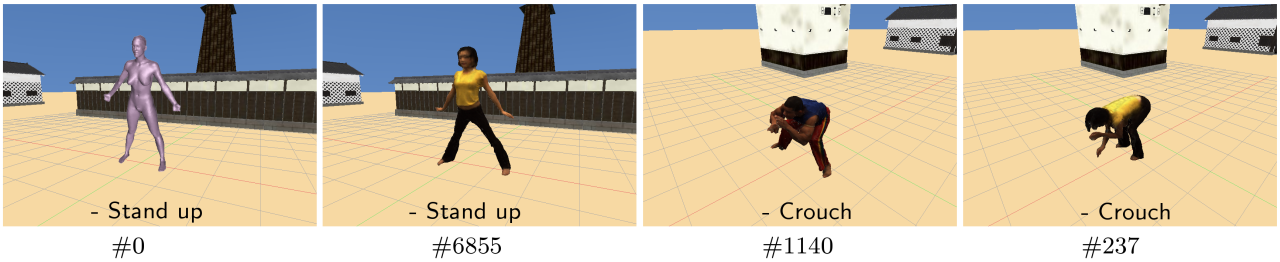

Figure 12. **3D action recognition using the topology dictionary.** Poses with annotations are indexed in the training dataset of the topology dictionary and mapped on the sequence frames as actions are recognized.

[11] 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. *CVIU*, 96(3):393–434, 2004.

[12] C.-W. Ngo, Y.-F. Ma, and H.-J. Zhang. Automatic video summarization by graph modeling. *ICCV*, 2003.

[13] V. Pascucci, G. Scorzelli, P.-T. Bremer, and A. Mascarenhas. Robust on-line computation of reeb graphs: Simplicity and speed. *SIGGRAPH*, 2007.

[14] A. Rav-Acha, Y. Pritch, and S. Peleg. Making a long video short: Dynamic video synopsis. *CVPR*, 2006.

[15] G. Reeb. On the singular points of a completely integrable pfaff form or of a numerical function. *Comptes Rendus Acad. Sciences Paris*, 222:847–849, 1946.

[16] S. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. *CVPR*, 2006.

[17] J. Shotton, M. Johnson, and R. Cipolla. Semantic texton forests for image categorization and segmentation. *CVPR*, 2008.

[18] M. Smith and T. Kanade. Video skimming and characterization through the combination of image and language understanding techniques. *CVPR*, pages 775–781, 1997.

[19] J. Starck and A. Hilton. Model-based multiple view reconstruction of people. *ICCV*, 2003.

[20] J. Starck and A. Hilton. Surface capture for performance-based animation. *IEEE Computer Graphics and Applications*, 2007.

[21] J. Sullivan and S. Carlsson. Recognizing and tracking human action. *ECCV*, 2002.

[22] T. Tung, S. Nobuhara, and T.Matsuyama. Simultaneaous super-resolution and 3d video using graph-cuts. *CVPR*, 2008.

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

[24] T. Tung, F. Schmitt, and T.Matsuyama. Topology matching for 3d video compression. *CVPR*, 2007.

[25] D. Weinland, E. Boyer, and R. Ronflard. Action recognition from arbitrary views using 3d exemplars. *ICCV*, 2007.

[26] J. Winn, A. Criminisi, and T. Minka. Object categorization by learned universal visual dictionary. *ICCV*, 2005.

[27] M. Yeung and B.-L. Yeo. Segmentation of video by clustering and graph analysis. *CVIU*, 71(1):94–109, 1998.

[28] J. Ziv and A. Lempen. A universal algorithm for sequential data compression. *IEEE Trans. on Information Theory*, 23(3):337–343, 1977.