Topology Dictionary for 3D Video Understanding

Tony Tung, *Member, IEEE,* and Takashi Matsuyama, *Member, IEEE*

Abstract—This paper presents a novel approach that achieves 3D video understanding. 3D video consists in a stream of 3D models of subjects in motion. The acquisition of long sequences requires large storage space (2GB for 1min). Moreover it is tedious to browse datasets and extract meaningful information. We propose the topology dictionary to encode and describe 3D video content. The model consists in a topology-based shape descriptor dictionary which can be either generated from extracted patterns or training sequences. The model relies on (1) topology description and classification using Reeb graphs, and (2) a Markov motion graph to represent topology change states. We show that the use of Reeb graphs as high level topology descriptor is relevant. It allows the dictionary to automatically model complex sequences, whereas other strategies would require prior knowledge on the shape and topology of the captured subjects. Our approach serves to encode 3D video sequences, and can be applied for content-based description and summarization of 3D video sequences. Furthermore topology class labeling during a learning process enables the system to perform content-based event recognition. Experiments were carried out on various 3D videos. We showcase an application for 3D video progressive summarization using the topology dictionary.

Index Terms—3D video, dictionary, Reeb graph, topology matching, Markov model, editing, summarization, semantic description.

1 Introduction

Dynamic multi-view stereo reconstruction (or 3D video) is an image-based technique which produces free-viewpoint videos of 3D models in motion (cf. [1], [2], [3], [4], [5], [6], [7]). As a markerless motion capture system, subjects do not need to wear any 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 from multiple views. 3D models are then reconstructed using multi-view stereo reconstruction algorithms [8]. In addition, textures are rendered on the reconstructed 3D model surfaces to obtain high quality visual effects (e.g. for cloth rendering). Many methods have recently focused on performance and quality improvements (cf. [9], [7], [10], [11]). However, the acquisition of long sequences produces massive amounts of data which make the datasets difficult to handle: browsing and searching for relevant information quickly become intractable. Hence 3D videos are still mainly only used for display.

In this paper, we introduce the topology dictionary as a new technique to achieve *3D video understanding*. The proposed model is inspired from a dictionary-based encoding strategy, which consists of searching for matches between a set of patterns contained in a data structure (a dictionary or codebook) and the data to be encoded. As the encoder finds a match, it substitutes a reference to the data position in the data structure. Hence redundancies can be efficiently identified and processed (cf. Vector Quantization [12], [13]). The dictionary can be either generated from extracted patterns, or training sequences (eventually with semantic annotations), to respectively encode and describe 3D video sequences. Pattern extraction is obtained by unsupervised clustering of data stream using enhanced Reeb graphs as 3D shape-based descriptor. Reeb graphs have been used for shape matching in large databases and can perform efficient shape retrieval queries [14]. As a canonical representation of the topology of the surface, they suit to encode 3D meshes with temporal evolution [15]. The dictionary features a graph structure where nodes (or states) represent patterns and edges characterize topological changes (or state transitions). Assuming a Markov graph structure [16], the model allows us to compute statistical information on the video content such as duration and occurrence probability of topology classes in order to derive the relevant and redundant patterns. 3D video content-based encoding and summarization are obtained using a probabilistic selection process. Furthermore recognition and description of sequences can be achieved using annotated training datasets.

The rest of the paper is organized as follows. The next section discusses work related to techniques presented in this paper. Section 3 presents the topology-based shape descriptor dictionary which features a Markov motion graph. Section 4 describes topology classification using Reeb graphs. Section 5 present 3D video understanding using topology dictionary: encoding, editing and description. Section 6 shows experimental results. Section 7 concludes with discussion on our contributions.

2 Related work

The literature has provided few solutions to manipulate 3D videos (cf. Fig. 1). To date, most investigations have focused on data compression. The most straightforward

• *T. Tung and T. Matsuyama are with the Graduate School of Informatics, Kyoto University, Kyoto, Japan.*
E-mail: {tung, tm}@vision.kuee.kyoto-u.ac.jp

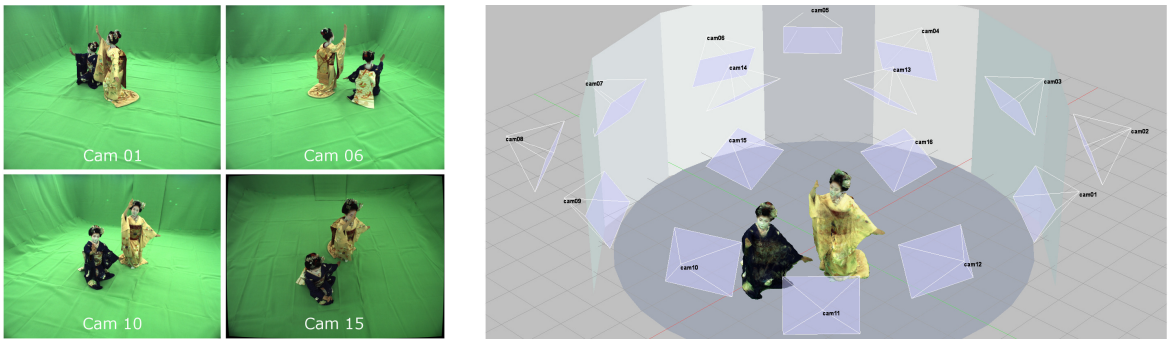

Fig. 1. **3D video framework.** 16 video cameras capture simultaneously subjects in motion. (Left) Video frame samples. (Right) 3D reconstruction from multi-view video frames of two traditional Japanese dancers (Maikos).

way to compress 3D video is to apply a compression technique to every 3D mesh of every single frame [17], [18]. However even though geometry and connectivity compression can generally guarantee lossless compression quality, they are not optimal since redundant information between frames is not handled. Various techniques dedicated to 3D animation sequence compression rely on basis decomposition [19], [20]. They are indeed dedicated to mesh sequences having the same connectivity and then cannot be applied to our data. Moreover Principal Component Analysis (PCA) methods usually require a registration step, which can be very time-consuming.

In [15], a 3D video compression strategy based on topology matching between consecutive frames was proposed. The algorithm consists in building enriched skeletons (namely aMRG graphs [14]) which embed 3D shape, texture, and temporal variations. The 3D video sequence is then compactly encoded and is reconstructed by skinning. In Sect. 4.1, we show the advantage of using this representation over skeleton fitting approaches such as [21], [22], [23], [24], [25]. 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 numerous complex topological changes. Moreover no information on the content of the sequence is given.

We propose a genuine content-based encoding strategy that can achieve 3D video understanding tasks such as editing, compression and description. The 3D video sequence is modeled by a topology dictionary with a Markov graph: a weighted directed graph where nodes represent topology classes (or states), and edges represent transitions between different classes [16]. The learning of dictionaries of feature vector clusters (or bags) have been successfully used for image categorization, segmentation and localization [26], [27], [28], and graphs have shown interesting applications for 2D video segmentation and summarization [29], [30]. Here we introduce the topology dictionary to learn and index 3D video patterns. In addition, semantic annotations of training datasets provide automatic video description and action recognition (as in [31], [32], [33], [34]). To our knowledge, although skimming techniques are successful for 2D videos [35], [36], the extension to 3D videos with automatic segmentation, encoding and reconstruction has not been treated yet. Preliminary versions of this paper appeared in [15] and [37].

Recently in [38] a framework to concatenate human motion 3D video sequences has been proposed. As in [37] a 3D shape descriptor is used to identify similar poses, and a graph is built to represent potential transitions. However in [38] a 3D histogram is used as 3D shape descriptor. This kind of representation cannot guarantee uniqueness of shape description, and therefore the processing of long and complex sequences might leads to inaccurate results and wrong graph designs. Moreover encoding and reconstruction of sequences are not possible and not handled whereas they are crucial in order to obtain smooth transitions between several potential states, in particular when topology changes occur. Our approach aims to manipulate and encode long sequences with various poses.

3 Topology Dictionary

The topology dictionary is proposed as an abstraction for data stream of geometrical objects. Data that evolve in time cannot be compactly represented using geometry only as the redundancy of information over time is not exploited. In particular it is challenging to manipulate a data stream when it becomes very complicated. Finding specific or relevant information is almost impossible. Moreover, when each data stream element is produced independently (such as in 3D video), geometrical representations quickly show their limitations as data structures are inconsistent to each other, i.e. complex matching processes are required to find geometric relations between consecutive data stream elements, and noises such as reconstruction artifacts have to be handled. However we can observe that the topology of data structure can remain very stable despite geometrical noises or short-term physical object motions. We therefore propose to use topology as a key property to characterize geometric data stream.

3.1 Model description

The model consists in a topology-based shape descriptor dictionary (or codebook) which can be either generated from extracted patterns or training sequences. The model relies on (1) topology description and classification using Reeb graphs (cf. Sect. 4), and (2) a Markov motion graph to represent topology changes. The graph is built upon the topology classes and the order of the frames to be encoded. The classes (or states) represent the words of the dictionary. Our approach serves to encode and describe 3D video sequences, and can be applied for content-based summarization of 3D video sequences (cf. Sect. 5). Furthermore, as mentioned previously, labeling and learning of topology classes enable the system to perform event recognition.

3.2 Dataset clustering

Let us assume a 3D video stream composed by a set of 3D mesh models $\mathcal{M} = \{m_1, ..., m_T\}$ where m_t is contained in the t^{th} video frame. A feature vector is extracted for every model based on a topology-based shape descriptor (cf. Sect. 4). As a feature vector is an abstraction of a mesh, we will refer to the mesh m_t or its feature vector equally. In order to cluster $\mathcal{M}$, the dataset is recursively split into subsets M_t and N_t as follows:

$$\begin{cases} M_t = \{n \in N_{t-1} : 1 - \mathrm{SIM}_k(m_t, n) < \tau\}, \\ N_t = N_{t-1} \setminus M_t, \end{cases} \quad (1)$$

where $M_0 = \emptyset$ and $N_0 = \mathcal{M}$. M_t is a subset of $\mathcal{M}$ representing a cluster containing m_t and similar elements. Similarities between elements of $\mathcal{M}$ are evaluated using a similarity function $\mathrm{SIM}_k : \mathcal{M} \times \mathcal{M} \in [0, 1]$ and a threshold $\tau \in \mathbb{R}$ (cf. below and in Sect. 4.2 for details).

The clustering step is a straightforward procedure: for each iteration step, from $t = 1$ to $t = T$, the closest matches to m_t are retrieved and indexed with the same cluster reference as m_t. Let us denote $\mathbf{C} = \{c_1, ..., c_N\}$ the N clusters created during the process by Eq. 1 (where the clusters are given by $\{M_t \neq \emptyset\}_t$ and $N \leq T$). Any visited element m_t already assigned to a cluster in $\mathbf{C}$ during a previous iteration step is considered as already classified and will be not processed subsequently. If $N_t = \emptyset$ or $t = T$, the recursive process terminates. As a result, the 3D video sequence $\mathcal{M}$ is clustered into the set of topology classes C. The next step consists in estimating topology class probabilities $\{P(c_1), ..., P(c_N)\}$ (cf. Sect. 3.3).

SIM_k is a similarity measure that computes a motion-to-motion matching score using a window of frames defined in the spirit of [39]:

$$\mathrm{SIM}_k(m_i, m_j) = \frac{1}{2k+1} \sum_{t=-k}^{k} \mathrm{SIM}(m_{i+t}, m_{j+t}). \quad (2)$$

The size of the window is chosen to be one third of a second in length ($k = 3$) as in [40]. Eq. 2 integrates consecutive frames in a fixed time window, thus allowing the detection of individual poses while taking into account smooth transitions. As defined, the formulation accounts not only for differences in body posture but also in motion speed. In practice, motion-to-motion matchings with SIM_k can be evaluated by first computing a frame-to-frame distance (or dissimilarity) matrix $\{1 - \mathrm{SIM}(m_i, m_j)\}_{ij}$, and then convolving the matrix using a window of size $2k + 1$ along diagonals (as in [41], [42]). Note that the pose invariance property of the Reeb graph allows us to compare poses (and motions) of subjects regardless of translation, rotation, and scaling.

Clustering evaluation. The clustering effectiveness is evaluated by the number of clusters found and should allow the identification of eventual redundant patterns. The threshold τ is set accordingly to the values of the similarity function SIM_k. The descriptor presented in this paper returns values in the range $[0, 1]$, and τ was defined experimentally. An optimal setting of τ should return a set of clusters similar to what a (hand-made) ground-truth classification would perform. As shown in Fig. 2 and Fig. 3, $\tau = 0.08$ returns qualitatively good clustering on humanoid datasets. Additional experiments are presented in Sect. 6.

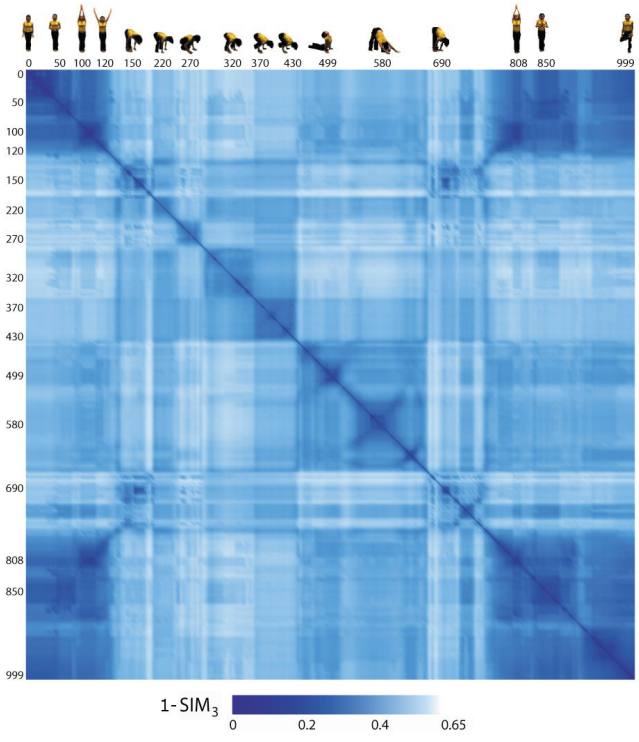

Fig. 2. **Distance matrix.** The matrix contains shape dissimilarity computation between 1000 frames of a 3D video sequence of a yoga performance. The blocks allow us to identify frames having similar topological structures.

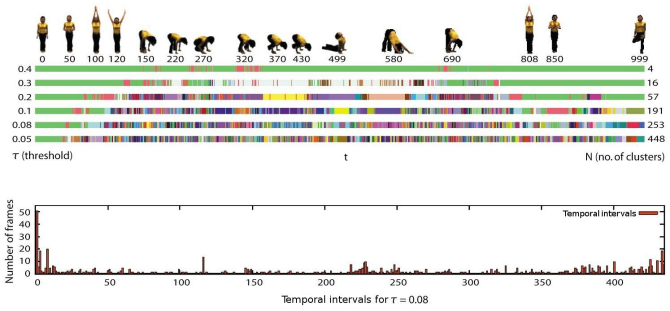

Fig. 3. **Clustering evaluation.**The topology dictionary allows the partition of data stream into atomic actions (sequence clips). Top color bars: number of clusters in a data stream with respect to the threshold τ (e.g. $\tau = 0.08$ returns 253 clusters from a partition containing 434 temporal intervals). Each color stands for a cluster. The histogram shows 1 long pose (>1s), 8 short actions (40ms to 1s), and 425 transition states (<40ms) for $\tau = 0.08$.

3.3 Markov motion graph

The structure of a data stream (e.g. video sequence) of an object in motion can be represented by a Markov motion graph that models the data evolution through successive states, such as scene changes [30]. In particular, when dealing with sequences of humans in motion, one can observe several repeated poses or actions, that can be efficiently encoded and exploited. Many researchers have indeed used statistical models of human motion to synthesize new animation sequences. For example in [43], [44], [45], [46] motion segments are identified using a motion database. A similar approach is introduced in the topology dictionary model in order to perform content-based manipulation of data stream.

Let us denote by $\mathbf{C} = \{c_1, ..., c_N\}$ the set of N clusters obtained by clustering the T frames of the sequence $\mathcal{S} = \{s_1, ..., s_T\}$, where $T = \sum_{i=1}^{N} N_i$ and $N_i = \text{card}(c_i)$ is the size of the i^{th} cluster c_i. Let us assume $\mathbf{G} = (\mathbf{C}, \mathbf{E})$ is a weighted directed graph, where $\mathbf{C}$ are the vertices and $\mathbf{E} = \{e_{ij}\}_{i\in[1,N]}^{j\in[1,N]}$ are the edges. We consider a probabilistic model where $\mathbf{C}$ represent states and $\mathbf{E}$ represent transitions between the states. The weights are defined as follows:

- The node weight $P(c_i) = \frac{N_i}{T}$ is the occurrence probability of a cluster c_i. If $P(c_i) \gg 0$, then c_i is a state representing a long or repeated pose preserving the same topological structure.
- The edge weight w_{ij} corresponds to the edge e_{ij} and models the transition probability between the two states c_i and c_j. w_{ij} is defined as the conditional probability:

$$P(c_j|c_i) = \frac{\sharp\{\text{frame transitions from } c_i \text{ to } c_j\}}{\sharp\{\text{frame transitions from } c_i\}}, \quad (3)$$

The probability is normalized so that assuming $\mathcal{N}_i^+ = \{c_j \in \mathbf{C} \setminus c_i : \exists (s_p, s_q) \in c_i \times c_j, q - p = 1\}$, then $\sum_{c_j \in \mathcal{N}_i^+} P(c_j|c_i) = 1$.

Let p_i^k denote the path in the motion graph $\mathbf{G}$ that links c_i to c_k. The path is defined as a set of successive nodes linked two by two by a single edge as follows: $p_i^k = \{c_{i_0}, e_{i_0 j_0}, c_{i_1}, e_{i_1 j_1}, ..., c_{i_K}\}$, where $i_0 = i$ and $i_K = k$. Then, the probability of p_i^k in $\mathbf{G}$ under Markov assumption can be evaluated using the following cost function:

$$E(p_i^k) = \sum_{i\in\{i_0,...,i_K-1\}} P(c_{i+1}|c_i)P(c_i), \quad (4)$$

where the most probable path p_{max} among all possible paths $\{p'^k_i\}$ that link c_i to c_k verifies:

$$p_{max} = \arg\max_{\{p'^k_i\}} E(p'^k_i). \quad (5)$$

The evolution of a data stream can be monitored using a graph representation as shown in Fig. 4. In the case of a sequence of animated 3D models, motion graph representation allows users (e.g. CG artists or animators) to design new sequences by navigating through probable paths as defined by Eq. 5, and concatenating sequence clips corresponding to the clusters belonging to the paths. For example, the 1000 frames of Yoga sequence shown in Fig. 3 were partitioned into 434 temporal intervals (at $\tau = 0.08$) which belongs to 253 clusters. Statistics on the clusters, such as the number of frames contained in each interval and each cluster, and the number of occurrences of clusters, return 1 long pose (>1s), 8 short actions (40ms to 1s), and 80 repeated atomic actions. The sequence can then be edited by shortening long poses and skipping atomic actions between redundant poses. The encoding strategy using the topology dictionary is presented in Sect. 5.1.

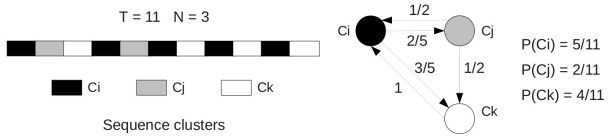

Fig. 4. **Probabilistic graph structure.** The sequence is represented as a Markov motion graph. It allows us to analyze the sequence content by modeling transitions between the different states. For example, cycles stand for repeated actions.

4 Topology Classification

This section gives a brief review of the augmented Multiresolution Reeb Graph (aMRG) [14], which is an enriched multiresolution Reeb graph [47]. The Reeb graph is an elegant solution to analyze 3D mesh topology and shape as it gives a graphical representation of surface properties. The reason we elected aMRG to cluster 3D video sequences is indeed threefold: (1) The Reeb graph extraction is fully automatic and does not require any prior knowledge on the shape, position, or topology of the captured subjects. It allows the system to model

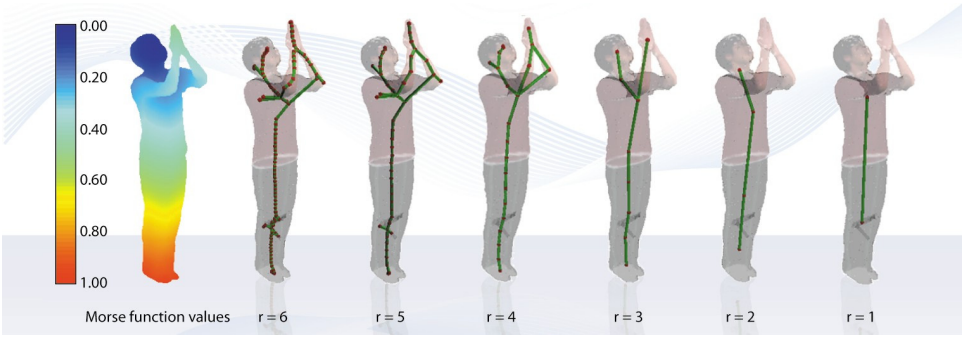

Fig. 5. **Topology-based 3D shape descriptor.** The Morse function allows the characterization of critical points on the mesh surface. Enhanced Reeb graphs are extracted at different levels of resolution to capture topology information.

complex sequences, especially when the fitting of a 3D skeleton of a priori known topology fails (e.g. subjects wearing loose clothing). (2) aMRG are high level 3D shape descriptors which have proven to be efficient for shape matching and classification tasks. (3) The multiresolution property with a hierarchical node matching strategy makes the search in large database tractable by by-passing the NP-complete complexity of the graph matching problem.

4.1 Reeb graph overview

Definition. We assume that 3D models are defined as compact 2-manifold surfaces approximated by 3D meshes. Let S be a surface mesh. According to the Morse theory, a continuous function $\mu : S \rightarrow \mathbb{R}$ 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 $\sim$ [48]. Assuming the points $x \in S$ and $y \in S$, then $x \sim y$ if and only if:

$$\begin{cases} \mathbf{y} \in \text{same connected component of } \mu^{-1}(\mu(\mathbf{x})), \\ \mu(\mathbf{x}) = \mu(\mathbf{y}). \end{cases} \quad (6)$$

The Morse function μ is the geodesic integral, as in [47]:

$$\boldsymbol{\mu(v) = \int_{p \in S} g(v, p) dS}, \quad (7)$$

where $g(v, p)$ is the geodesic distance on S between two points v and p belonging to S. That is, the Reeb graph of μ on S describes the connectivity of the level sets of μ. Note that the inverse function μ^{-1} is defined on $\mathbb{R}$ and returns regions on S corresponding to level sets of μ at some isovalues.

The function μ is further normalized with respect to its minimal and maximal values $\mu_{\min}$ and $\mu_{\max}$ as $\mu_N : S \rightarrow [0, 1]$, where $\boldsymbol{\mu_N(v) = \frac{\mu(v) - \mu_{\min}}{\mu_{\max} - \mu_{\min}}}$. Extremal values of μ_N return surface critical point locations which coincide to highly concave or convex regions.

Construction. The multi-resolution Reeb graph is a set of Reeb graphs of various levels of resolution. The construction process consists in first building the graph at the highest resolution $r = R$ ($R > 1$), and then iteratively deriving the graphs at lower resolution until $r = 1$ (cf. Fig. 5). At resolution $r = 0$, the graph consists in one unique root node. A Reeb graph at resolution level R is constructed by:

1) Partitioning the range of μ_N, i.e. $[0, 1]$, into 2^R regular intervals by iterative subdivisions, and assigning interval labels to surface points (i.e. mesh vertices) according to their μ_N values.
2) Creating a graph node for each surface *region* consisting of mutually connected surface points with the same interval label.
3) Linking the nodes that have their corresponding regions connected on the surface.

In practice, when the surface is represented by a triangular mesh, at each resolution each node corresponds to a set of connected triangles (and is placed at its centroid). The nodes created in the step 2 above stand for the Reeb graph nodes, and the links created in the step 3 stand for the Reeb graph edges. Reeb graphs at lower resolutions $r < R$ are obtained by first merging the intervals of μ_N values two by two using a hierarchical procedure. Then, a parent node is assigned to each group of nodes in the higher resolution graph whose corresponding surface regions are connected and share the same interval label of merged μ_N value. Hence, each node at resolution $r > 1$ has a unique parent node belonging to a Reeb graph at resolution $r - 1$ [47]. Note that the object surface is partitioned into regions with 2^r interval labels at the resolution level r.

Robustness. As the 3D models in 3D video data stream usually contain reconstruction artifacts, the Reeb graph extraction has to be particularly robust to surface noise. Fortunately, the normalized Morse function introduced in Eq. 7 is robust to local surface noise thanks to the integral formulation (as well as being invariant to rotation, translation and scale transformation). To evaluate the stability of the Reeb graph regarding surface noise, we tested the Reeb graph extraction on 3D models of

different resolution (high and low). We observed that extra nodes might appear occasionally, especially at the extremities of the graphs. These are due to surface sampling implementation artifacts, as geodesic distances between vertices of a surface mesh are computed using the mesh edges. Hence geodesic distances on two meshes having different connectivity may differ. However as defined Eq. 7 can usually cope with all kind of surface noise (including mesh connectivity change) thanks to the integral formulation which smoothes local variations.

Figure 6 shows (a) the 40th frame from the sequence Tony, captured and reconstructed in our laboratory, with 17,701 vertices and simplified to 1,335 vertices, and (b) the 25th frame from the sequence Free of [9] with 142,382 vertices and simplified to 6,500 vertices. The Reeb graphs are shown at the resolution $r = 3$. The simplifications were performed by edge collapsing in order to affect the geodesic measurements as much as possible. Despite some extra nodes at some extremities of the Reeb graphs, we can observe that their overall structure and topology are preserved.

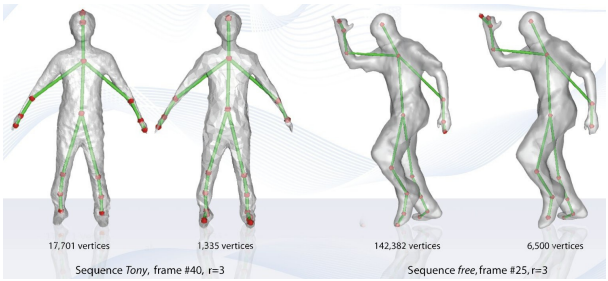

Fig. 6. **Reeb graph robustness to surface noise.** Reeb graphs are constructed for 3D models of different resolution (high and low). Overall structure and topology are preserved.

Advantage. Structure extraction from arbitrary shape is usually performed by fitting a 3D skeleton to the shape surface model, such as in [24]. When successful, this kind of approach is powerful because the kinematic structure of the object can be extracted, and the structure joints can be tracked while the object is in motion. However, fitting a skeleton requires to have prior knowledge on the shape to be described: the skeleton has to be defined beforehand and cannot be fitted to any arbitrary shapes [21]. On the other hand, the Reeb graph overcomes these limitations as it can characterize topology and shape of arbitrary 3D models. No a priori knowledge on the model shape and topology is required, and no initial pose is required. Figure 7 illustrates the advantage of using an automatic topological structure extraction method such as the Reeb graph, as opposed to a skeleton fitting technique (such as [24]). As can be observed, the Reeb graph can extract a consistent structure from arbitrary shapes without any prior knowledge, even though: (a) limbs are not visible, and regardless of the model (b) topology, (c) orientation, and (d) complexity[1]. Note that other approaches, such as the curve-skeleton [22], [23], [25], can be used to extract a graph with homotopy property. Nevertheless, as shown in the next section, our approach is the most suitable for shape matching as it features a hierarchical multi-resolution structure. Otherwise numerous graph matching computation in huge dataset can quickly become intractable.

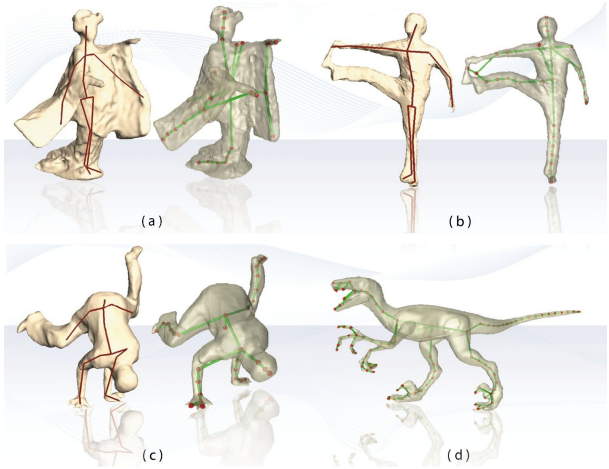

Fig. 7. **Comparison of structure extraction techniques.** Reeb graph (in green) vs. skeleton fitting approach (in red) [24].

4.2 Topology matching

Similarity evaluation. Dataset clustering is performed by similarity evaluation of aMRG graphs extracted from every 3D model. Assuming two aMRG graphs M and N, the similarity is obtained by computing the following SIM function:

$$\mathrm{SIM}(M,N) = \frac{1}{1+R}\sum_{r=0}^{R}\sum_{\{(m,n)\in\mathcal{C}_r\}}\mathrm{sim}(m,n), \qquad (8)$$

where $\mathcal{C}_r \subset M \times N$ contains all the pairs of topologically consistent nodes at the resolution level $r \in [0,R]$, and $\mathrm{sim} : M \times N \rightarrow [0,1]$ evaluates the similarity between two nodes m and n. SIM is obtained by summing similarity scores obtained for each pair of matching nodes (by sim) at every level of resolution from $r = 0$ to R. Each similarity evaluation of a pair of nodes returns a (positive) contribution to the global similarity score given by SIM, and is higher when nodes are similar. If $M = N$ then $\mathrm{SIM}(M,M) = 1$. Implementation details can be found in [14], where each node embeds topological attributes such as relative surface area and graph connectivity information, and geometrical attributes such as surface normal orientation histogram. Note that SIM is positive, reflexive, symmetric but not

1. Raptor model is provided courtesy of INRIA by the AIM@SHAPE Shape Repository.

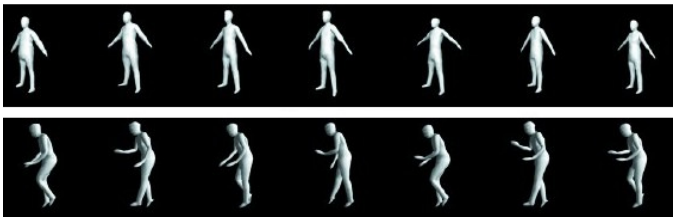

Fig. 8. **Synthetic dataset.** aMRG retrieval performance is evaluated using ground truth data. (Top) Example of models. (Bottom) Example of motion.

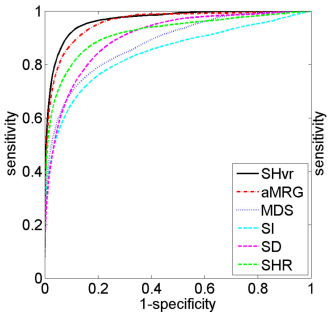

Fig. 9. **Evaluation of ROC curves.** aMRG is one of the top performers for shape retrieval in 3D video.

transitive as the matching depends on node structure.

Multiresolution strategy. The coarse-to-fine multiresolution matching strategy has two major advantages: (1) it is crucial for tractability when dealing with large database as it avoids the NP-complete problem of graph matching by stopping the matching process at low resolution if models are too different, and (2) it provides a judicious matching scheme as global shape and topology (the postures) account for more than fine details (e.g. arm positions, fingers). The nodes are matched hierarchically (starting at $r = 0$) using topology consistency rules and similarity evaluations (sim function); node children are matched recursively up to the highest resolutions. Irrelevant nodes are discarded in the graph matching process as they embed weak weights (e.g. nodes from a noisy surface). Further details concerning the matching step and sim function can be found in [14], [47].

4.3 Retrieval performance

The performance of aMRG is evaluated against various shape similarity metrics for 3D video sequences of people with unknown temporal correspondence [41]. Performances of similarity measures are compared by evaluating Receiver Operator Characteristics (ROC) for classification against ground-truth of a comprehensive dataset of synthetic 3D video sequences consisting of animations of several people performing different motions (cf. Fig. 8). The synthetic dataset is created using 14 articulated character models, each of which animated using 28 motion capture sequences. Recognition performances are evaluated using ROC curve, showing the true-positive or *sensitivity* in correctly defining similarity against the false-positive rate (FPR) or *one-specificity*. The evaluations on real 3D video sequences demonstrate that aMRG is one the top performers in the task of finding similar poses of the same person in 3D video compared to the state-of-the-art in shape matching techniques (cf. Fig. 9 and [42]). The comparison includes Shape Histograms (SHvr), Multi-Dimension Scaling (MDS), Spin Image (SI), Shape Distribution (SD) and Spherical Harmonics Representation (SHR) (cf. [41] for additional details).

5 3D VIDEO UNDERSTANDING

3D videos of animated subjects usually contain long poses, slow motions and repeated actions that can be represented efficiently using topology descriptors. We propose to use the topology dictionary to identify poses and encode 3D video sequences. We take advantage of the Markov graph structure to perform 3D video content-based progressive summarization (or skimming) by probabilistic discrimination of frames.

5.1 Content-based summarization

Encoding. We present a linear process to compactly encode 3D video data stream. Assuming the data stream $\mathcal{S}$ containing T frames, $\mathcal{S} = \{s_1, ..., s_T\}$, a data structure k_i is created for every of the N clusters $c_i \subset \mathcal{S}$, where $1 \leq i \leq N$. Each k_i encompasses a specific pose of the object and all its variations within each temporal interval c_i^j that forms $c_i = \bigcup_j \{c_i^j\}$.

Let $s_{i\min}^j$ and $s_{i\max}^j$ denote the frames having respectively the smallest and biggest indices in the j^{th} interval $c_i^j \subset c_i$. k_i contains one textured mesh (i.e. one mesh and one texture map), one graph structure (namely one Reeb graph at a chosen resolution), a table of node position offsets corresponding to all the node trajectories to transit from any $s_{i\min}^j$ to k_i, and a table of node position offsets corresponding to all the node trajectories to transit from k_i to any $s_{i\max}^j$. The node trajectories are obtained by tracking the node positions in each c_i^j, which is a trivial task as long as the graphs are consistent to each other (as it should be in each cluster). In practice, noisy nodes and edges are removed so that any Reeb graph constructed from a frame in c_i is topologically consistent to k_i. Practically, k_i should be chosen at the center of the cluster c_i:

$$k_i = \arg \min_{M_k \in c_i} \sum_{M \in c_i} \mathrm{SIM}(M_k, M), \tag{9}$$

where M_k and M are meshes in c_i, and SIM is the similarity function (Eq. 8).

The encoding process consists in sequentially substituting each identified cluster in a data stream by a cluster (or pattern) reference index. If a frame s_t does

not belong to the same cluster c_{i-1} as the previous frame s_{t-1}, then a new data structure k_i is created. If s_t belongs to the same cluster c_{i-1} as s_{t-1}, then only position offsets of the graph nodes at t are stored into k_{i-1}. It is then possible to recover the node trajectories between consecutive frames and reconstruct the mesh sequence by skinning (cf. next Section). Let s_m be the size of an encoded mesh plus a Reeb graph structure, and s_g be the size of an encoded set of node position offsets, the size σ of the encoded sequence is then $\sigma \leq s_m * N + s_g * T$.

Edition. 3D video content-based edition is performed by interacting with the motion graph as presented in Sect. 3.3. In particular path probability estimation (Eq. 5) allows users to evaluate scenario realism. In what follows we present an unsupervised scheme to skim 3D video sequences by processing 3D video clips of human performances. The goal is to automatically produce shorter sequences while preserving scenario consistency. The topology dictionary is used to identify isolated and non-relevant patterns and progressively remove them. First, the set **C** is sorted with respect to the cluster weights $P(c_i)$ in order to identify the frames belonging to clusters having the highest and lowest probabilities:

- If $P(c_i) \gg 0$, then c_i contains either: (1) a long sequence of successive frames belonging to c_i, or (2) a recurrent pose identified by frames belonging to c_i scattered in the sequence. In the case (1), long poses (e.g. low variations such as between frames $\#370$ and $\#430$ in Fig. 2) are compressed by encoding intermediate 3D video frames as described in the previous section. In the case (2), c_i is represented as a cycle (or loop) junction node in the motion graph **G**. The strategy consists therefore in gradually removing the small and non-relevant cycles:

$$S(\mathcal{L}) = \frac{\sum_{\{c \in \mathcal{L}\}} P(c)}{\text{card}\{c \in \mathcal{L}\}}, \quad (10)$$

$$P(\mathcal{L}) = \frac{\sum_{\{e_{ij} \in \mathcal{L}\}} P(c_j|c_i)P(c_i)}{\text{card}\{e_{ij} \in \mathcal{L}\}}, \quad (11)$$

  where the *size* $S(\mathcal{L})$ is the average weight of the cycle $\mathcal{L}$, and the *relevance* $P(\mathcal{L})$ is defined as the probability of the cycle $\mathcal{L}$ under Markov assumption. The following weight is used to sort the cycles:

$$W(\mathcal{L}) = \lambda.S(\mathcal{L}) + (1 - \lambda).P(\mathcal{L}), \quad (12)$$

  where $\lambda = 0.5 \in [0, 1]$. Practically, the skimming process consists in removing redundant frames from video sequences where small cycles with low probability are elected as first candidates for skimming. As discussed in Sect. 6, several other skimming strategies can be adopted.
- If $P(c_i) \sim 0$ then c_i contains few frames. Identified isolated patterns are reclassified into adjacent clusters: e.g. in the sequence $\{c_i, c_i, c_j, c_i, c_i\}$, the frames $\{s \in c_j\}$ are reclassified into c_i.

Finally, summarization can be processed iteratively up to some user-defined constraints such as a limitation on the sequence size or compression ratio (cf. Sect. 6).

5.2 Sequence reconstruction

3D video reconstruction from encoded frames is obtained by using a mesh skinning method where surface deformations are guided by Reeb graph nodes. Skinning [49], [50] is a popular method for performing character and object deformation in 3D games and animation movies (usually using CG software such as Blender, Maya, [51], [24]). During the skinning process, the graph is bound to a single mesh object, and the mesh is deformed as the graph nodes move. As node coordinates change, transformation matrices associated to the vertices of the mesh cause them to be deformed in a weighted manner. A weight defines how much a specific node influences vertices in the deformation process (e.g. 1.0 for rigid skinning, and less than 1.0 for smooth skinning). It is usual to set smoother skinning for vertices belonging to a joint area on a surface mesh.

The data stream reconstruction is performed for each cluster c_i by considering each temporal interval c_i^j independently, where $c_i = \bigcup_j \{c_i^j\}$, in order to avoid surface topology change issues. A unique data structure k_i, as introduced in the previous section, represents the cluster c_i: all of the 3D video frames whose corresponding feature vectors belong to c_i are reconstructed by deforming k_i according to encoded node coordinates (for each c_i^j) using a mesh skinning method as described above.

Figure 10 illustrates a sample of 3D video data reconstructed from an encoded data stream. Note that even though the implemented mesh skinning method is not optimal, no major reconstruction artifacts can be noticed at video frame rate (25fps). We measured 3D position distortions in a $400 \times 400 \times 400$ voxel grid (corresponding to a 2m $\times$ 2m $\times$ 2m volume having 5mm resolution), and we obtained the mean squared error $MSE \sim 0.005$ and the peak signal-to-noise ratio $PSNR \sim 75dB$ when computing surface distances using Hausdorff distance as metric.

5.3 Semantic description

Semantic description of data stream is obtained by labeling identified 3D video clips. The labeling can be performed on training datasets that are clustered using a topology dictionary model (as described in Sect. 3.2), and where each cluster is given a tag. The depiction is done using semantic annotations related to the data content (e.g. “stand up, hands on hips”, “stand up, hands joined over the head, head looking the hands”, etc.). In practice, any pose with annotation can be added into the dictionary, as learning and indexing can be performed on any

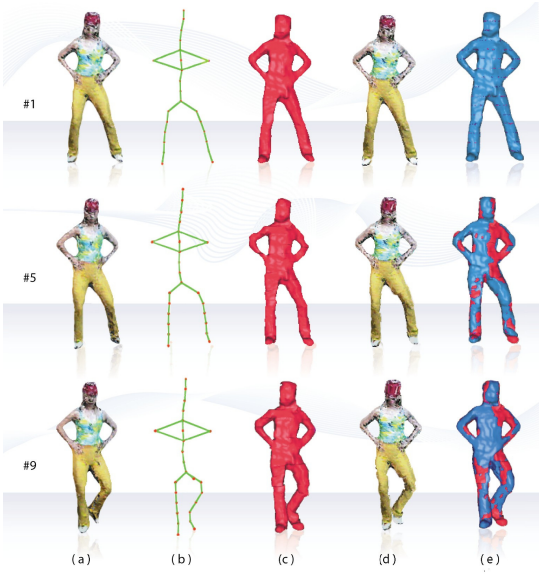

Fig. 10. **Sequence reconstruction.** (a) Textured mesh from initial 3D video data. (b) Reeb graphs extracted at resolution $r = 4$. (c) Reconstructed surfaces by mesh skinning. (d) Reconstructed surfaces with texture. (e) Overlay of both surfaces.

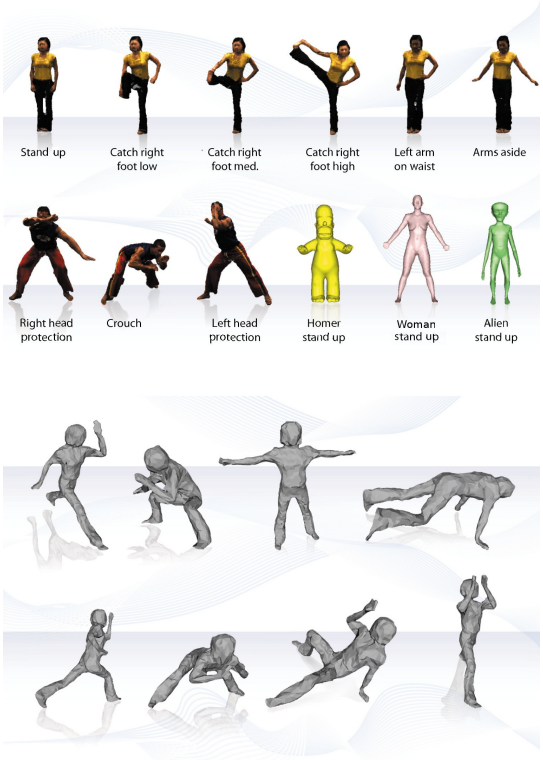

Fig. 11. **Samples of annotated data and CG models for topology dictionary learning.**

arbitrary data. For example as shown in Fig. 11, models from various sources can be annotated for action recognition application (e.g. 3D models from the Internet, CG models, etc.). As a consequence, queries on shape and/or semantic can serve to retrieve specific poses in 3D video data stream. In addition, classes can be built based on shape categorization in addition to topology. The Homer, woman and alien models have similar topology structure as they all stand up, but different shape features as the limbs have different lengths, and body build is different. The variations are captured by the Reeb graph node attributes. Thus, the topology dictionary can perform classification and description based on shape as well as topology. Furthermore, as the patterns we are considering are based only on shape and topology, there is no knowledge about content importance. However, an additional weight related to importance can be added along with labels (or annotations) to clusters belonging to training datasets. Hence, clusters with lower importance weight could be removed first in the skimming process described in Sect. 5.1.

5.4 On-line classification

The encoding of new frames is managed as a mapping problem. A classifier assigns new unclassified feature vectors to topology classes. The structure of the Markov motion graph $G = (C, E)$ is used to build a cluster priority list for each query. A cascade of classifiers is built to avoid slow and potentially intractable brute force search that would consist in systematically comparing new candidates to all clusters.

Assuming a new frame s_{T+1} is added to the sequence $\mathcal{S} = \{s_1, ..., s_T\}$, where s_T belongs to the cluster $c_i \subset C$, and let $\mathcal{N}_+(c_i) = \{c \in C | P(c|c_i) > 0\}$ and $\mathcal{N}_-(c_i) = \{c \in C | P(c_i|c) > 0\}$ denote the sets of clusters that are adjacent to c_i in G. A cascade of classifiers is created from the adjacent clusters and ordered by occurrence probability (high to low). s_{T+1} is successively compared to the sets: c_i, $\mathcal{N}_+(c_i)$ starting with higher probabilities $P(c|c_i)$, and $\mathcal{N}_-(c_i)$ starting with higher probabilities $P(c_i|c)$. For each cluster c, s_{T+1} is compared against its center $s_c \in c$. The search is performed until a match is found, as $1 - \mathrm{SIM}(s_c, s_{T+1}) < \tau$ (cf. Eq. 8 and Fig. 12). Alternatively, the search space is extended to the neighbors of the visited clusters up to a predefined depth of search. If no class is suitable for s_{T+1}, a new cluster is created and linked to c_i. Note that each similarity evaluation is achieved using a multiresolution matching scheme (embedded in Eq. 8). The process is tractable since unsimilar shapes are quickly rejected.

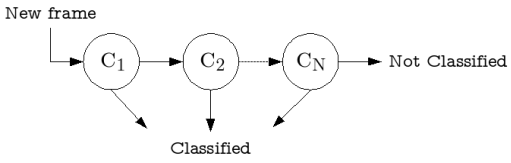

Fig. 12. **Mapping with a cascade of classifiers.**

6 Experimental Results

To assess the performance of the topology dictionary model, several experiments were performed on various 3D video sequences. The Yoga (Fig. 2) and Tony (Fig. 5) datasets are interesting as they contain many human poses. These sequences are useful to set the parameter τ for sequences of humanoid objects. The Maiko dataset is challenging for shape description as the subject wears a loose kimono which covers arms and legs. Fortunately the Reeb graph is a relevant tool to characterize arbitrary shapes. The Capoeira sequence represents quick moves of martial art.

Yoga, Tony, Maiko and Capoeira sequences contain respectively 7500, 250, 201 and 300 frames captured by multiple video cameras in a 3D video studio at 25fps. Every frame is composed of one 3D mesh of ∼30K triangles with texture. One uncompressed frame encoded in standard OFF format requires 1.5MB, which means 11.25GB for 7500 frames. Feature vectors were computed on a Core2Duo 3.0GHz 4GB RAM, nevertheless the process requires less than 512MB RAM. SIM computation between two models takes 10ms. A feature vector up to resolution level $R = 5$ is generated in 15s with the current implementation (cf. [52] for binaries). Other efficient computation of Reeb graphs can be found in the literature (e.g. [53]).

Topology dictionary stability. The core of the topology dictionary model relies on its ability to discriminate shape topology. The definition of the Morse function is therefore crucial (cf. Sect. 4). The ability of the dictionary to extract and classify patterns has been evaluated against different Morse functions and resolution levels R of Reeb graphs (cf. Fig. 13): the curves named geodesic $R = 5$, geodesic $R = 4$, and geodesic $R = 3$, were obtained when using the geodesic integral as Morse function as in Eq. 7, and by computing similarities up to the resolution levels $R = 5$, $R = 4$, and $R = 3$ respectively. The curve named geodesic $r = 4$ was obtained with the geodesic integral as Morse function, but without summation of coarse resolution levels when computing the similarity (i.e. only the contributions at level $r = 4$ were used in Eq. 8). The curve named height $R = 4$ was obtained when using the height function $\mu(\mathbf{v}) = z$ as Morse function, and by computing similarities up to the resolution level $R = 4$. The clustering performance is then evaluated with respect to the threshold τ. In Fig. 13 and 14, the sequences contain respectively 500 and 7500 frames of a Yoga session. They consist in a succession of various (complex) poses. The clustering behavior was analyzed with different values of τ and parameter setting.

It turned out that the integral geodesic functions with $R = 4$ and $\tau = 0.1$, and $R = 3$ and $\tau = 0.08$, give the best trade-offs for clustering performance and computation time for humanoid model sequences in comparison to a hand-made clustering. The full Yoga sequence (7500 frames) contains 1749 clusters: 44 long poses or repeated actions (>1s), 115 short actions (40ms to 1s) and 1590 transition states (<40ms).

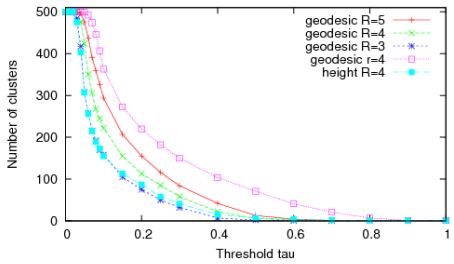

Fig. 13. **Clustering of Yoga (500f) with respect to τ.**

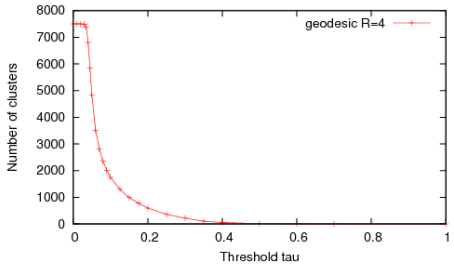

Fig. 14. **Clustering of Yoga (7500f) with respect to τ.**

Simple tests on 3D video data streams representing models with similar topological structures show that using the same parameters (i.e. μ function, R and τ), similarity estimations remain in the same range (cf. Fig. 15): a parameter set can be re-used for objects belonging to similar categories. aMRG graphs of Tony and Yoga have indeed similar skeleton-like structure.

Tony	145 vs 165	165 vs 100
SIM	0.33	0.39
Yoga	88 vs 115	115 vs 993
SIM	0.28	0.37

Fig. 15. **Similarity evaluation between data presenting similar topological structures**: frame #145 from Tony and frame #88 from Yoga, and frame #165 from Tony and frame #115 from Yoga.

Fig. 16. **3D video edition of the Yoga sequence.** Frames $\#4179$, $\#6987$ and $\#7000$ belong to the same cluster. Hence (1) they can be encoded using a unique data structure (cf. Sect. 5.1), and (2) they belong to cycles of the Markov motion graph in the topology dictionary model (cf. Sect. 3.3).

To cluster the Maiko sequence, the clustering threshold τ is set to 0.2 (cf. Fig. 17). The loose clothing makes the object shape more compact than humans with tight clothing. Hence the shape characterization at the lower resolutions of the aMRG is less discriminant, and therefore τ has to be increased. In fact, we believe that a relationship between τ and the global shape of models can be established (e.g. $\tau = 0.08$ for star-like shapes). This would allow us to set τ automatically. The Maiko sequence describes a dancer in action, performing a 360 degree rotation and kneeling. 201 frames were clustered in 24 clusters with $\tau = 0.2$ and a partition containing 95 temporal intervals. The statistics indicate that the sequence contains 2 short actions (40ms to 1s), and 17 repeated atomic actions. The longest state corresponds to the last part of the video, where the Maiko slows down her motion and remains still. The short actions and (quick) transitions allow the characterization of the pace and activity of the Maiko during her performance as she turns and moves her hand at the same time.

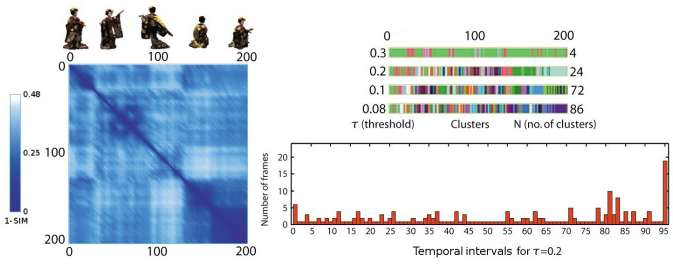

Fig. 17. **Clustering of Maiko (201f).** τ = 0.2 returns 24 clusters from a partition containing 95 temporal intervals.

3D video progressive summarization. The size of a sequence is growing linearly of 1.5MB per frame (11,250MB for 7500 frames). Hence it becomes very difficult to browse or search for specific information into long sequences. As presented in Sect. 5.1, our 3D video encoding process consists in two steps.

First, redundancies in long poses, slow motions and transition states are located in the data stream and compactly encoded using topology dictionary modeling. On the Yoga dataset, cluster encoding using the data structure presented in Sect. 5.1 returns a compression ratio of nearly 2:1, meaning a saving space of 50% (cf. Fig. 18). The 7500 frame sequence has been reduced to 3660 encoded frames as 1749 clusters were obtained: 44 long or repeated poses ($>$ 1s), 115 short actions (40ms to 1s) and 1590 transition states ($<$40ms).

Second, using the dictionary graph structure 656 cycle junction clusters and 9095 cycle combinations have been identified. Figure 16 show similar frames identified in the Yoga sequence: frames $\#4179$, $\#6987$ and $\#7000$ belong to the same cluster. Hence each of these frames corresponds to a repeated (atomic) action and correspond to a cycle junction cluster. Thus, frames contained in cycles can be removed for 3D video data skimming (and summarization) using the topology dictionary motion graph. 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%. Content-based summarization can then be progressively performed while keeping relevant information. Another possible skimming scheme consists in successively skipping the biggest cycles (in size $S(\mathcal{L})$). It returns

a sequence of 1439 encoded frames (the ratio is 5:1 and a saving space of 80%). Note that other strategies can be considered, such as maximizing content unicity by removing repeated instances of same actions and shortening long actions. Figure 19 presents progressive

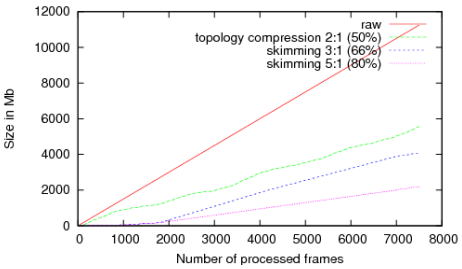

Fig. 18. **Summarization ratio of Yoga (7500f).**

summarization of the Yoga sequence. The result on a sample of 1000 frames is shown for the Yoga sequence instead of the results on the full sequence of 7500 frames for presentation clarity purpose. The 1000 frame Yoga sequence has 80 cycle junction clusters and 486 cycle combinations. Figure 20 presents progressive summarization of Maiko sequence. The Maiko sequence has 17 cycle junction clusters and 257 cycle combinations.

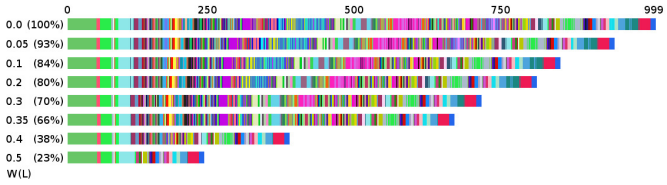

Fig. 19. **Progressive skimming of Yoga (1000f).** Values of $W(\mathcal{L})$ (Eq. 12) and summarization ratio.

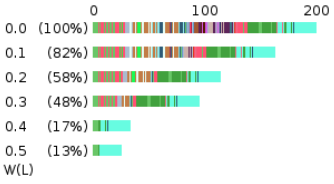

Fig. 20. **Progressive skimming of Maiko (201f).** Values of $W(\mathcal{L})$ (Eq. 12) and summarization ratio.

Semantic description. Semantic description of 3D video sequences is obtained by labeling topology clusters as described in Sect. 5.3. The system will then return a label as a class is identified. Training datasets can be populated with models from various sources, e.g. from the Internet or designed by CG software as shown in Fig. 5.3. Semantic description of 3D video data using a training dataset with annotation is shown in Fig. 21.

7 Conclusion

This paper presents the topology dictionary, a novel approach that achieves 3D video understanding for applications such as content-based encoding, summarization and semantic description. The topology dictionary has been proposed as an abstraction to represent data stream of geometrical objects. In particular, when the data streams become very complicated, the geometry features quickly show some limitations. It is then a challenge to manipulate the data stream, and especially to look for specific or relevant information. In some cases such as with 3D video data, every geometrical object composing the stream is obtained independently. Hence, no consistency exists between the geometrical structure of the objects, and the state-of-the-art does not provide any efficient technique to handle the data stream. However, the topology of data structure can be used as a stable feature to describe such kind of data. As one can observe, using a topology descriptor, stable topology features can be preserved even though the geometrical representation is challenged by noise of deformations. Taking advantage of this property, the topology dictionary has been developed as a combination of two ideas: (1) a dictionary-based encoding strategy identifies relevant patterns in data stream, and (2) a probabilistic graph models the structure of the stream, allowing data content-based manipulation.

In this paper, we show that this abstraction can be applied to 3D video data stream. Our implementation involves the use of the augmented Multi-resolution Reeb Graph (aMRG) as a robust topology descriptor for 3D shape matching and dataset clustering, and a Markov motion graph to model transitions between clusters. We present content-based encoding, summarization, and semantic description of various 3D video sequences. We believe the topology dictionary brings lots of perspectives to future research and applications on 3D video. As the reader may have noticed, several data structures were employed in this research work, which make the overall system rich and complex. For further studies, we should provide an interactive tool to pick clusters (or poses) using a visual representation of the motion graph. The best path on the motion graph between the selected clusters would return new sequences. As well, complex scenes containing animals or interacting people should be tackled.

Acknowledgments

This work was supported in part by the JST-CREST project "Creation of Human-Harmonized Information Technology for Convivial Society", and the Japan Society for the Promotion of Science (Wakate-B No.23700170). We thank Ms. Kinh Thang and Ms. Karine Tung for their support, as well as Ms. Bidda Camilla Solvang Poulsen for her graphic design work.

Fig. 21. **Semantic description of Capoeira and Yoga sequences, as well as arbitrary models.**

REFERENCES

[1] T. Kanade, A. Yoshida, K. Oda, H. Kano, and M. Tanaka, "A stereo machine for video-rate dense depth mapping and its new applications," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, p. 196, 1996.

[2] J. Starck and A. Hilton, "Model-based multiple view reconstruction of people," *Proc. 9th IEEE Int'l Conf. Computer Vision*, vol. 2, pp. 915–922, 2003.

[3] 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," *Computer Vision and Image Understanding*, vol. 96, no. 3, pp. 393–434, 2004.

[4] J. Franco, C. Menier, E. Boyer, and B. Raffin, "A distributed approach for real-time 3d modeling," *Proc. IEEE Conf. Computer Vision and Pattern Recognition Workshop on Real-Time 3D Sensors and their Applications*, p. 31, 2004.

[5] K. M. Cheung, S. Baker, and T. Kanade, "Shape-from-silhouette across time: Part ii: Applications to human modeling and markerless motion tracking," *Int'l J. Computer Vision*, vol. 63, no. 3, pp. 225–245, 2005.

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

[7] E. de Aguiar, C. Stoll, C. Theobalt, N. Ahmed, H.-P. Seidel, and S. Thrun, "Performance capture from sparse multi-view video," *ACM Trans. Graphics*, vol. 27, no. 3, 2008.

[8] S. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski, "A comparison and evaluation of multi-view stereo reconstruction algorithms," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol. 1, pp. 519–526, 2006.

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

[10] T.Tung, S.Nobuhara, and T.Matsuyama, "Simultaneous super-resolution and 3d video using graph-cuts," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2008.

[11] ——, "Complete multi-view reconstruction of dynamic scenes from probabilistic fusion of narrow and wide baseline stereo," *Proc. 13rd IEEE Int'l Conf. Computer Vision*, 2009.

[12] R. Gray and A. Gersho, "Vector quantization and signal compression," *Kluwer, Norwell, MA*, 1992.

[13] J. Ziv and A. Lempen, "A universal algorithm for sequential data compression," *IEEE Trans. on Information Theory*, vol. 23, no. 3, pp. 337–343, 1977.

[14] T. Tung and F. Schmitt, "The augmented multiresolution reeb graph approach for content-based retrieval of 3d shapes," *Int'l J. Shape Modeling*, vol. 11, no. 1, pp. 91–120, 2005.

[15] T. Tung, F. Schmitt, and T.Matsuyama, "Topology matching for 3d video compression," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2007.

[16] S. P. Meyn and R. Tweedie, "Markov chains and stochastic stability," *Cambridge University Press*, 2008.

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

[18] 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*, pp. 3–26, 2005.

[19] M. Alexa and W. Müllen, "Representing animations by principal components," *Computer Graphics Forum*, vol. 19, no. 3, 2000.

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

[21] J.Carranza, C.Theobalt, M.Magnor, and H.-P.Seidel, "Free-viewpoint video of human actors," *ACM Trans. Graphics*, vol. 22, no. 3, pp. 569–577, 2003.

[22] K.Palagyi and A.Kuba, "A parallel 3d 12-subiteration thinning algorithm," *Graph. Models and Image Proc.*, vol. 61, no. 4, pp. 199–221, 1999.

[23] N.Cornea, D.Silver, X.Yuan, and R.Balasubramanian, "Computing hierarchical curveskeletons of 3d objects," *The Visual Computer*, vol. 21, no. 11, pp. 945–955, 2005.

[24] I. Baran and J. Popovic, "Automatic rigging and animation of 3d characters," *ACM Trans. Graphics*, vol. 26, no. 3, p. 27, 2007.
[25] A.Sharf, T.Lewiner, A.Shamir, and L.Kobbelt, "On-the-fly curve-skeleton computation for 3d shapes," *Comput. Graph. Forum*, vol. 26, no. 3, pp. 323–328, 2007.
[26] J. Winn, A. Criminisi, and T. Minka, "Object categorization by learned universal visual dictionary," *Proc. 10th IEEE Int'l Conf. Computer Vision*, vol. 2, pp. 1800–1807, 2005.
[27] J. Shotton, M. Johnson, and R. Cipolla, "Semantic texton forests for image categorization and segmentation," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2008.
[28] B. Fulkerson, A. Vedaldi, and S. Soatto, "Localizing objects with smart dictionaries," *Proc. 10th European Conf. Computer Vision*, vol. 1, pp. 179–192, 2008.
[29] M. Yeung and B.-L. Yeo, "Segmentation of video by clustering and graph analysis," *Computer Vision and Image Understanding*, vol. 71, no. 1, pp. 94–109, 1998.
[30] C.-W. Ngo, Y.-F. Ma, and H.-J. Zhang, "Video summarization and scene detection by graph modeling," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 15, no. 2, pp. 296–305, 2005.
[31] J. Sullivan and S. Carlsson, "Recognizing and tracking human action," *Proc. 7th European Conf. Computer Vision*, 2002.
[32] M. Müller, T. Röder, and M. Clausen, "Efficient content-based retrieval of motion capture data," *ACM Trans. on Graphics*, vol. 24, no. 3, pp. 677–685, 2005.
[33] D. Weinland, E. Boyer, and R. Ronfard, "Action recognition from arbitrary views using 3d exemplars," *Proc. 11th IEEE Int'l Conf. Computer Vision*, 2007.
[34] L. Sigal, A.Balan, and M. Black, "Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion," *Int'l J. Computer Vision*, vol. 87, no. 1, 2010.
[35] M. Smith and T. Kanade, "Video skimming and characterization through the combination of image and language understanding techniques," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 775–781, 1997.
[36] Y. Pritch, A. Rav-Acha, and S. Peleg, "Nonchronological video synopsis and indexing," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 30, no. 11, pp. 1971–1984, 2008.
[37] T.Tung and T.Matsuyama, "Topology dictionary with markov model for 3d video content-based skimming and description," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2009.
[38] P. Huang, A. Hilton, and J. Starck, "Human motion synthesis from 3d video," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2009.
[39] A. Schödl, R. Szeliski, D. Salesin, and I. Essa, "Video textures," *Proc. ACM SIGGRAPH*, pp. 489–498, 2000.
[40] T. Mizuguchi, J. Buchanan, and T. Calvert, "Data driven motion transitions for interactive games," *Eurographics Short Presentations*, 2001.
[41] P. Huang, A. Hilton, and J. Starck, "Shape similarity for 3d video sequences of people," *IJCV Special Issue on 3D Object Retrieval*, vol. 89, no. 2-3, pp. 362–381, 2010.
[42] P. Huang, T. Tung, S. Nobuhara, A. Hilton, and T. Matsuyama, "Comparison of skeleton and non-skeleton shape descriptors for 3d video," *Proc. 3DPVT*, 2010.
[43] L.Molina-Tanco and A.Hilton, "Realistic synthesis of novel human movements from a database of motion capture examples," *In IEEE Workshop on Human Motion*, 2000.
[44] O. Arikan and D. Forsyth, "Interactive motion generation from examples," *ACM Trans. Graphics*, vol. 21, no. 3, pp. 483–490, 2002.
[45] L. Kovar, M. Gleicher, and F. H. Pighin, "Motion graphs," *ACM Trans. Graphics*, vol. 21, no. 3, pp. 473–482, 2002.
[46] J.Lee, J.Chai, P. S. Reitsman, J. Hodgins, and N. S. Pollard, "Interactive control of avatars animated with human motion data," *ACM Trans. Graphics*, vol. 21, no. 3, pp. 491–500, 2002.
[47] M. Hilaga, Y. Shinagawa, T. Kohmura, and T. L. Kunii, "Topology matching for fully automatic similarity estimation of 3d shapes," *Proc. ACM SIGGRAPH*, pp. 203–212, 2001.
[48] G. Reeb, "On the singular points of a completely integrable pfaff form or of a numerical function," *Comptes Rendus Acad. Sciences Paris*, vol. 222, pp. 847–849, 1946.
[49] S. Park and J. Hodgins, "Capturing and animating skin deformation in human motion," *ACM Trans. Graphics*, vol. 25, no. 3, pp. 881–889, 2006.
[50] O. Sorkine and M. Alexa, "As-rigid-as-possible surface modeling," *Proc. 5th Eurographics Symposium on Geometry Processing*, pp. 109–116, 2007.
[51] Y. Kho and M. Garland, "Sketching mesh deformations," *ACM Trans. Graphics*, vol. 24, no. 3, p. 934, 2005.
[52] T. Tung, "Shape similarity computation using amrg," *tonytung.org*.
[53] V. Pascucci, G. Scorzelli, P.-T. Bremer, and A. Mascarenhas, "Robust on-line computation of reeb graphs: Simplicity and speed," *ACM Trans. Graphics*, vol. 26, no. 3, p. 58, 2007.

Tony Tung Tony Tung received the M.Sc. degree in Physics and Computer Science from the Ecole Nationale Supérieure de Physique, France, with a double degree in Photonics and Image Processing in 2000, and the Ph.D. degree in Signal and Image processing from the Ecole Nationale Supérieure des Télécommunications de Paris in 2005. He has worked as an IT consultant (2000-2002) and R&D engineer (2005-2008) in private companies, and as a postdoctoral research fellow at Kyoto University (2005, 2008-2009). Since 2010, he is an Assistant Professor at Kyoto University, working jointly with the Matsuyama Laboratory at the Department of Intelligence Science and Technology, Graduate School of Informatics, and the Kawahara Laboratory at the Academic Center for Computing and Media Studies. His research interests include computer vision (3D video), pattern recognition, shape modeling, and human behavior analysis. He was awarded Fellowships from the Japan Society for the Promotion of Science in 2005 and 2008, and Grant-in-Aid for Young Scientists in 2011.

Takashi Matsuyama Professor Takashi Matsuyama received B. Eng., M. Eng., and D. Eng. degrees in electrical engineering from Kyoto University, Japan, in 1974, 1976, and 1980, respectively. He is currently a professor in the Department of Intelligence Science and Technology, Graduate School of Informatics, Kyoto University.

His research interests include knowledge-based image understanding, computer vision, 3D video, human-computer interaction, and smart energy management. He wrote more than 100 papers and books including two research monographs, A Structural Analysis of Complex Aerial Photographs, PLENUM, 1980 and SIGMA: A Knowledge-Based Aerial Image Understanding System, PLENUM, 1990.

He won ten best paper awards from Japanese and international academic societies including the Marr Prize at ICCV'95. He is on the editorial board of the Pattern Recognition Journal. He was awarded Fellowships from the International Association for Pattern Recognition, the Information Processing Society of Japan, and the Institute of Electronics, Information, and Communication Engineers Japan.