HAL open science

Progressive Shape Models

Antoine Letouzey, Edmond Boyer

▶ To cite this version:

Antoine Letouzey, Edmond Boyer. Progressive Shape Models. CVPR - Computer Vision and Patern Recognition - 2012, Jun 2012, Providence, United States. pp.190-197, 10.1109/CVPR.2012.6247675 . hal-00677506

HAL Id: hal-00677506

https://inria.hal.science/hal-00677506

Submitted on 8 Jan 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution - NonCommercial - ShareAlike 4.0 International License

Progressive Shape Models

Antoine Letouzey Edmond boyer
INRIA Grenoble Rhône-Alpes
655, Avenue de l'Europe, 38334 St Ismier France
firstname.lastname@inria.fr

Abstract

In this paper we address the problem of recovering both the topology and the geometry of a deformable shape using temporal mesh sequences. The interest arises in multi-camera applications when unknown natural dynamic scenes are captured. While several approaches allow recovery of shape models from static scenes, few consider dynamic scenes with evolving topology and without prior knowledge. In this nonetheless generic situation, a single time observation is not necessarily sufficient to infer the correct topology of the observed shape and evidences must be accumulated over time in order to learn the topology and to enable temporally consistent modelling. This appears to be a new problem for which no formal solution exists. We propose a principled approach based on the assumption that the observed objects have a fixed topology. Under this assumption, we can progressively learn the topology meanwhile capturing the deformation of the dynamic scene. The approach has been successfully experimented on several standard 4D datasets.

1. Introduction

Visual acquisition systems have become a common way to acquire 3D shape models. Their applications, such as scene understanding, cover a wide range of domains from computer vision to computer graphics and medical applications. Recent systems, using for instance color or depth information, enable the acquisition of 3D models over temporal sequences, hence allowing for objects that deform and evolve over time. Unless strong prior knowledge on shapes is available, *e.g.* a template obtained with a laser-scan [1, 5], such systems usually produce temporal series of inconsistent 3D models (4D data), often in the form of meshes with different connectivities and possibly different topologies. A key issue in the acquisition process is then how to obtain a model that is consistent with all the observations, hence allowing for tracking and motion related applications.

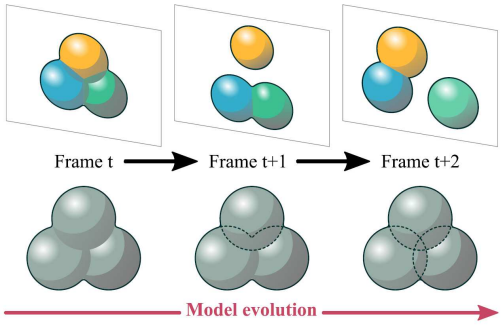


Figure 1: Top row, temporal observations of a scene composed of several interacting components. Below the learned model evolution over time. Note that none of the time frames identifies the correct topology of the scene.

In the general case, such a template model cannot be assumed for dynamic scenes since they may be composed of several interacting disconnected components. Moreover, even a single object may be difficult to model *a priori* with a static acquisition system, like for example animals. Furthermore, considering one of the frames in a temporal sequence as a template, *e.g.* [4], does not solve the problem either since a single frame seldom provides enough topological information on the observed shape (see figure 1). In this paper we consider this issue and devise a simple yet efficient algorithm that accumulates the topology information provided by a temporal sequence of 3D models and progressively learns and tracks a model of the true observed shape.

Recently, significant efforts have been put into the recovery of precise model geometry given multiple images [11], or the tracking of deforming objects given multiple videos and a template topologically consistent with the observations, [15, 5, 4], or even the matching of shapes, as in [3], with possibly different topologies, [13]. More recent approaches are able to improve shapes models using temporal sequences of observations, *e.g.* [10]. However, the problem of recovering a proper shape models from temporal observations where topological changes occur is still open.

To learn and track a template model over a time sequences of 3D models we introduce progressive shape models. Based on a growth strategy, these models evolves both the topology and the geometry of a reference model along a temporal sequence. Topology refers here to properties such as the number of connected components and holes and geometry refers to the location in space. The proposed estimation approach alternates between model evolution for the topology and model tracking for the geometry. With respect to the first aspect, we assume that topological properties can only grow in the progressive shape model, as revealed over time by the acquisition process and up to some noise tolerance. This assumption is grounded on the fact that most objects of interest, in a real context, presents a fixed topology that is preserved by deformations. Hence, fusions can not occur in the model evolution. Under this assumption, our approach provides a well founded solution for the recovery of both geometry and the topology from temporal sequences of 3D surfaces and hence enables the capture and the analyses of scenes with increasing complexity.

The remainder of the paper is as follows: §2 reviews the related works. The proposed method is detailed in §3. §4 presents both quantitative and qualitative evaluations on synthetic and real data and §5 discusses the approach.

2. Related Work

In the past years, several works have considered the recovery of shape deformations and evolutions from temporal sequences of visual data such as color and depth images. These approaches can be roughly classified with respect to the amount of prior knowledge they assume. First, several of these approaches assume a known model of the shape, either a surface or volume model as in [5, 7, 6], or a structure model such as a skeleton in [15]. While allowing for partial observations only, such an *a priori* model is however seldom available, in particular with complex scenes. Another strategy in this category is to consider one frame of a temporal sequence as the model, as in [4]. While sensible in some situations, this is not guaranteed to provide a model topologically consistent with other frames. In this paper, we tackle the problem of building such a model.

Other methods have been proposed that consider less prior information. For instance, [9] registers large sets of unstructured points clouds in a global manner by assuming smooth motion and a dense sampling over time and space. [18] also registers point clouds of deforming objects by assuming that they present skeletal structures that are consistent over time and hence can be aligned. These approaches solve for the registration over time however they implicitly assume a static and simple topology. It is worth also mentioning the works of [12] and [8]. In both methods, the information is accumulated over a time window in order to improve the individually reconstructed shape models. While improving over individual reconstruction, these approaches do not learn a model; in addition our approach gathers information from the whole sequence in order to build a model as accurate as possible.

Closer to our purpose, a few methods learn a shape model using temporal evolutions. For instance, [16] introduces a joint estimation of deformations and shapes. Interestingly [10] proposes a hierarchical strategy that progressively improves a shape model by accumulating pairwise shape differences in a binary tree built on top of the frames. While allowing for some topological changes, by thresholding shape differences, both these approaches are mainly designed to fill holes in the shape model by accumulating depth map information. Our scenario is different since we consider full shape information, *i.e.* meshes, instead of point clouds. Our strategy differs as well since our approach does not estimate topological changes by approximation but learns them from clear evidences in the observed frames.

Our contributions with respect to the aforementioned works is twofold. First we introduce progressive shape models that incrementally builds models of dynamically moving and deforming objects from a sequence of independently reconstructed surfaces. Second the model estimation is grounded on a theoretically well founded treatment of evidences in the observations to detect and to handle topology changes.

3. Method

3.1. Principle

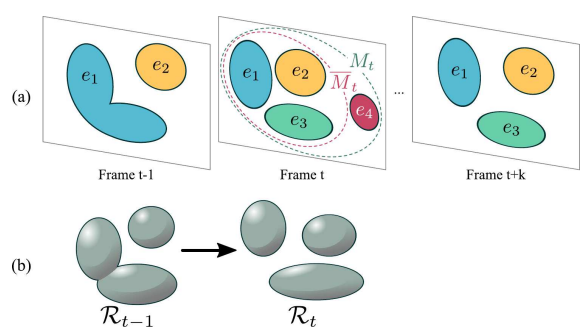


Figure 2: Sequential evolution of the topology of the progressive model. (a) Observed shape components that present an outlier at t and (b) the corresponding model $\mathcal{R}$ evolution.

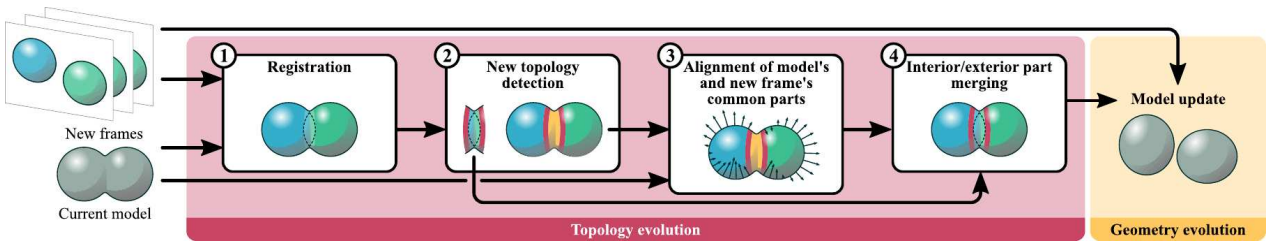


Figure 3: Overview of the progressive modelling.

The objective is to recover the shape and topology of a dynamic scene observed over a time sequence. The approach takes as input a temporal sequence of n inconsistent triangular meshes and estimates a template mesh model of the scene. Starting with the shape and topology of the first frame as the model, the approach evolves the shape of this model so that it fits spatially the observation at each time instant while recording topology changes such as : object apparition, split and hole creation. The observed scene under consideration can be composed of any number of non-rigidly moving components but they are assumed to have a fixed topology, *i.e.* the scene's Betti numbers are fixed. Note that the recovered topology is the best that can be estimated given the observations but it is not necessarily the true one, unless enough information is available in the observations.

The approach evolves the progressive model topology along the sequence. Using a set formulation, the principle is as follows. Let $M_t = \{e_j\}$ be the set of observed shape components at time t (see figure 2). In order to account for noise in the acquisition process we consider that components are valid only if they appear in at least k consecutive frames, thus:

$$\overline{M}_t = \{e_j \mid e_j \in \bigcap_{i=0}^{k} M_{t+i}\}, \tag{1}$$

is the set of valid components at time t. Let R_n be the set of components in the template model at frame n. Following our assumption that the topology of the observed shape is fixed and thus necessarily includes all the valid components that appear along the sequence:

$$R_n = \bigcup_{i=1}^{n} \overline{M}_i = R_{n-1} \bigcup \overline{M}_n. \tag{2}$$

Figure 2 shows how the model R_n is evolving with a simple example.

The progressive modelling proceeds in four main steps iterated over the frames, as depicted in figure 3. Each of these step is further detailed in a specific section.

1. The observed meshes in the new frame are registered with the current progressive model §3.2.
2. New topology information are detected in the registered meshes as the interior faces §3.3.
3. The current model is precisely aligned with the registered meshes minus the interior faces §3.4.
4. The new topology, *i.e.* interior faces, is added to the current model. §3.5.

In what follows, we consider a single iteration of the approach and call the incoming meshes with a new frame $\mathcal{M}$ and the current progressive model $\mathcal{R}$.

3.2. Registration

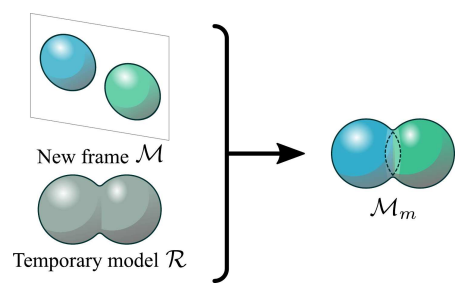


Figure 4: Model Registration: the new frame $\mathcal{M}$ is registered to the current progressive model $\mathcal{R}$. The result is a deformed version $\mathcal{M}_m$ of $\mathcal{M}$ that best fits $\mathcal{R}$.

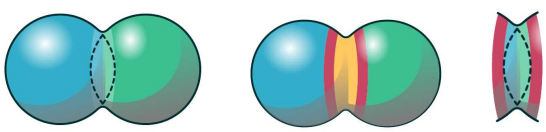
The first step of the approach consists of registering the new observations $\mathcal{M}$ with the current model $\mathcal{R}$. This step allows to identify topological differences between the model and the observations. The main difficulty lies in the fact that their associated meshes present different connectivities and possibly different number of components. To solve the problem, we cast the registration as an optimization where the distance between the computed deformation of $\mathcal{M}$ and the current model is minimized while enforcing a locally smooth deformation function over the surface.

Denoting Θ the parameters of the deformation, *i.e.* the vertex displacements, the optimization consists in maximising the log-likelihood of the joint probability distribution of the observations $\mathcal{M}$, the model $\mathcal{R}$ and the deformation parameters Θ:

$$\arg\max_{\Theta} \quad \ln P(\mathcal{M}, \mathcal{R} \mid \Theta) P(\Theta) \,. \tag{3}$$

Several approaches can be considered for that purpose using either photometric or geometric information. For instance, in Popa *et al*. [10] a tracking method based on optical flow is used. However, the optical flow, as a result of finite difference approximations, does not correctly handle large deformations and we use a purely geometric approach proposed by Cagniart *et al*. [4]. This approach was originally designed to track a mesh model over a temporal sequence of independently reconstructed meshes with identical topologies. By making use of a probabilistic framework it can handle multiple deforming meshes and missing data and hence adapts well to the problem considered here. We implemented a simplified version that registers the new surface $\mathcal{M}$ coming with the next frame to the current progressive model $\mathcal{R}$, as shown in figure 4. $\mathcal{M}_m$ is then the deformed version of $\mathcal{M}$ after this process.

3.3. Topology Change Detection



(a) $\mathcal{M}_m$ (b) $\mathcal{M}_{out}$ (c) $\mathcal{M}_{in}$

Figure 5: Detecting topology changes: (a) Input Mesh with self-intersections due to topology evolution. (b) Self intersection removal - red: vertices where cuts occur, orange: new geometry added to $\mathcal{M}_m$. (c) The identified *inside* part $\mathcal{M}_{in}$. The red vertices in (b) and (c) are common to $\mathcal{M}_{out}$ and $\mathcal{M}_{in}$.

Once the new frame $\mathcal{M}_m$ is registered with the current progressive model $\mathcal{R}$, the next step is to detect topology changes in $\mathcal{M}_m$ in order to update $\mathcal{R}$ later on. With a compact surface, and in the general case, these topological changes can be of four types: *split*, *hole formation*, *merge* and *hole loss*. As mentioned earlier, we assume that the observed shape has a fixed topology. Therefore a *split* and a *hole formation* can be observed, as a result of missing information to be added to the model, whereas a *merge* or a *hole loss* can not occur since they imply a change in the observed shape topology. Figure 1 shows a simple example of this principle.

The case where new objects enter the scene is trivially handled by a connected component detection. The idea is to verify whether a connected component of $\mathcal{M}$ is classified as an outlier during the previous registration step (see §3.2). In this case, the component does not belong to the template model and thus is simply added to it.

Other non-trivial topology changes are detected when self-intersections of the registered frame $\mathcal{M}_m$ occur. The resulting inside facets are considered as missing parts in the model $\mathcal{R}$. To this aim, we build on the approach proposed by Zaharescu *et al*. [17]. This approach identifies the *outside* part of a self-intersecting mesh and produces a manifold that represents this part (see fig. 5b). The main feature of this method is that it preserves the geometry of the mesh and only modifies its connectivity (2D Delaunay triangulations are performed inside intersecting facets that are subdivided). Applying this on $\mathcal{M}_m$ we compute $\mathcal{M}_{out}$: a manifold mesh with the geometry of the registered observations and the topology of the model $\mathcal{R}$, and we compute the inside part $\mathcal{M}_{in}$ as $\mathcal{M}_m \setminus \mathcal{M}_{out}$. We also mark the vertices where cuts are made (see fig. 5. If no self-intersection occurs, the model $\mathcal{R}$ is simply kept unchanged and we proceed to the next frame.

The output of this step is therefore two separated meshes. $\mathcal{M}_{out}$ a re-meshed version of the outside part of $\mathcal{M}_m$ and $\mathcal{M}_{in}$ the set of inside or partially inside facets of $\mathcal{M}_m$. In both meshes, vertices corresponding to the cut made while removing inner parts of $\mathcal{M}_m$ are marked to ease their access during the next steps.

3.4. Precise Alignment

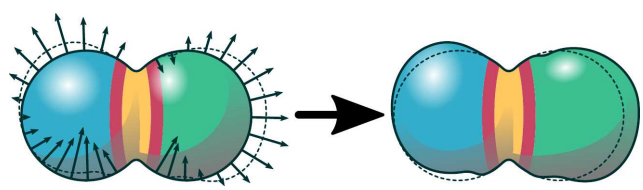


Figure 6: Precise Alignment of $\mathcal{M}_{out}$ onto the current progressive model.

The third step of the approach consists in precisely aligning the registered partial observation $\mathcal{M}_{out}$ to the model $\mathcal{R}$, this in order to add $\mathcal{M}_{in}$ to $\mathcal{R}$ in a further step. $\mathcal{M}_{out}$ is topologically equivalent to $\mathcal{R}$ or, at least, $\mathcal{M}_{out}$ does not have topology information that are not present in $\mathcal{R}$. Thus, we apply a morphing strategy where $\mathcal{M}_{out}$ is deformed by moving every vertex toward $\mathcal{R}$. Precisely, at each iteration, vertices are moved along their normal axis using the following displacement vector $\mathbf{d}$ computed at each vertex p on $\mathcal{M}_{out}$:

$$\mathbf{d}_p = \gamma^{\mathcal{R}}(p)\, N(p) \,, \tag{4}$$

where $N(p)$ is the normal to $\mathcal{M}_{out}$ at p and $\gamma^{\mathcal{R}}()$ a signed distance function to $\mathcal{R}$. In order to preserve manifoldness

during this step, we again use the approach in [17] on the deformed version of $\mathcal{M}_{out}$.

In the above deformation process, vertices marked in the previous step (the red and orange points in fig. 6) are not moved for two main reasons. First, by nature they do not correspond to any part of the current model $\mathcal{R}$ and as such should not participate in the deformation. Second, they correspond to part of the observation that should not be modified since they are on both $\mathcal{M}_{out}$ and $\mathcal{M}_{in}$ and will be used in the next step to merge inside and outside parts.

3.5. Progressive Shape Model Update

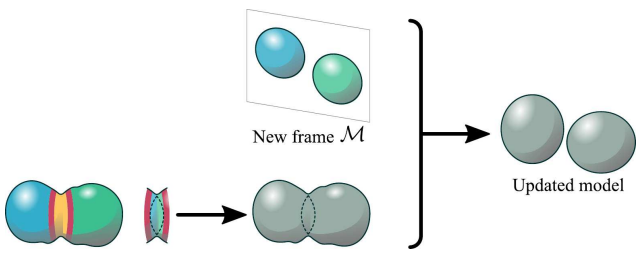


Figure 7: Progressive model update: merging of the deformed version of $\mathcal{M}_{out}$ and $\mathcal{M}_{in}$ and registration with the current observation.

At this stage, two meshes are available. $\mathcal{M}_{in}$ that encodes the topological evolution arising from the new observation $\mathcal{M}$ and $\mathcal{M}_{out}$ that is geometrically equivalent to the current model $\mathcal{R}$ but contains additional information on where cuts were made. The final step of the proposed method simply merges this two meshes together. This process is eased by the fact that tracks of the vertices where cuts were made during the topology detection step are kept. Those vertices are merged on $\mathcal{M}_{out}$ and $\mathcal{M}_{in}$ and the geometry that $\mathcal{M}_{in}$ will replace is removed on$\mathcal{M}_{out}$. In figure 5, this corresponds to merging red vertices on $\mathcal{M}_{out}$ and $\mathcal{M}_{in}$ while removing orange vertices on $\mathcal{M}_{out}$.

The newly obtained mesh contains both the information gathered in $\mathcal{R}$ from all previous frames as well as the newly available information brought by the new observation $\mathcal{M}$. The reference model $\mathcal{R}$ is then replaced with this new mesh. Before proceeding to the next frame, the new template $\mathcal{R}$ mesh is deformed so that it fits the observation of the current frame. This is the tracking step where the geometry of the model evolves with respect to the observations. To this aim, the registration step described in §3.2 is performed with the source being the model and the target the observation (see fig. 7).

Following this process, at the end of the pipeline, we have a new shape model whose topology is consistent with all the previous observations and whose geometry is aligned with the last observation. In order to build a temporally consistent sequence, this model can be tracked back over the complete sequence.

3.6. Implementation Notes

During the precise alignment step (see §3.4), a part of $\mathcal{M}_{out}$ is not deformed (the red and orange areas in fig. 5). This part corresponds to the geometry added to $\mathcal{M}_m$ during the topology change detection. To avoid sharp transitions at the boundaries between deformed and non-deformed regions in the resulting model we extend the non-deforming constraint by defining a smooth transition area. To this aim, the N-neighbourhoods of non-moving vertices are marked as belonging to the neighbouring transition area. In this area, connectivity is preserved by preventing removal and creation of vertices while letting them move freely. Once the convergence is reached, their final positions are computed as a smooth blending of their previous and new positions, where the latter are weighted by their distances to the cutting edge. The size of this area is the only data-dependent parameter of our method. It can easily be chosen by looking at the data, and in particular the ratio between edge sizes and topological merges.

4. Evaluations

Evaluations were performed on both synthetic and real data with the aim to cover a wide range of deformations and topology changes.

We created two synthetic datasets. The first is composed of three balls moving and intersecting each others. The generated triangular meshes of this short sequence contain around $2k$ vertices. In this sequence all the objects are rigid and none of the input frames reveals the true topology of the scene. The second sequence represents a Y-shaped object that merges and splits. The input meshes contain approximately $6k$ vertices.

Experiments were also conducted with three different sets of real data, all publicly available. The first one was taken from the flashkick sequence from the University of Surrey [14]. The second and third ones come from INRIA. Both these datasets provide temporal sequences of inconsistent meshes obtained using a standard 3D modelling approach. Note that meshes in these two sequences show a number of temporal discontinuities in their topologies.

Computation times mostly depend on the mesh complexity and are in the order of one minute per frame on a standard PC (Core 2 Duo 3GHz, 4GB ram) for the most complex dataset ($10k$ vertices).

4.1. Synthetic Data

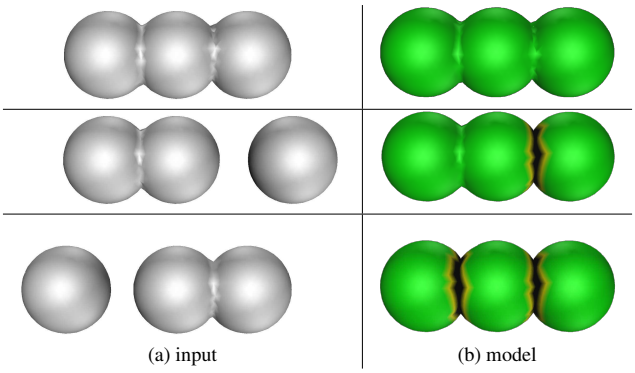


Figure 8: Result on the first synthetic dataset. (a) Three consecutive input meshes and (b) the corresponding progressive model evolution.

In order to demonstrate the ability of the presented approach to create a model of a scene that is better than any individual mesh of a sequence, we generated a simple sequence where three spheres intersect each other. Figure 8a shows three consecutive frames from this sequence. As we can see, none of the input meshes include the global topology information of the scene. Figure 8b shows the evolution of the template. As shown in the figure, the proposed approach gradually fits the real shape of the scene, and the topology information is correctly accumulated over the frames in the progressive shape model. Green areas represent the common parts of the previous and newly computed model. Black areas denote the vertices added to the previous model, *i.e*. the physical boundaries. The final progressive model contains the three independent and complete spheres. The yellow ring around black areas denotes the N-ring neighbourhood where vertices are protected from the precise alignment step as discussed in §3.6. For this dataset, a 1-ring neighbourhood was used.

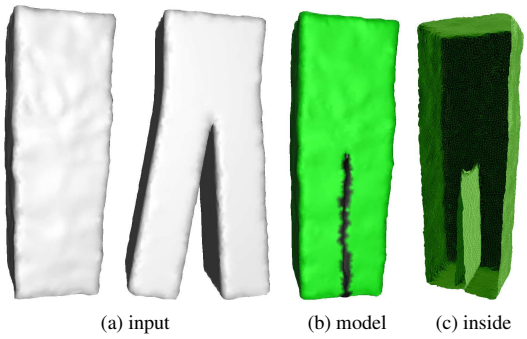


Figure 9: Result on the y-shaped object. (a) Two first meshes; (b) The inferred template evolution re-aligned with the first frame; (c) Inside view of the template re-aligned with the first frame.

The results on the second synthetic dataset show that the method does not only recover deformations with Betti numbers changes in the number of holes and connected components, but also less radical deformations. In the first frame, which is also the initial model, legs are merged due to the convex hull simplification. Later in the sequence the legs get separated (see figure 9a). As show in figure 9b and 9c the proposed method correctly splits the legs in the initial template. In this case the size of the deformation-free area was set to 0.

4.2. Real Data

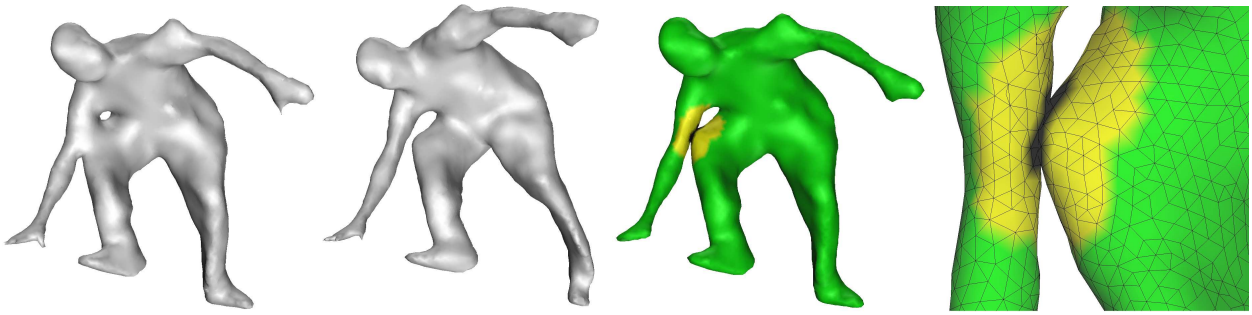
The first real data sequence was obtained with eight cameras shooting a dancer. The low number of cameras available combined with the use of a coarse 3D modelling algorithm, yields to a large number of topological artefacts in the reconstructed meshes. Figure 10a and 10b shows two input meshes and the associated improvement of the progressive model between these two frames. The model is initialised with the mesh in (a). Figure 10c shows how the model evolves correctly with respect to the input data. Due to the higher resolution and wide merging area in the input data, the size of the deformation-free area (yellow) was set to 4 (4-ring neighbourhood) in this case.

The second dataset used for the experiments on real data shows a man moving his hands from hips to head. This sequences was acquired using a 32-camera setup, leading to a set of much more accurate input meshes than with the previous dataset. However, the topology is still not correct when contacts in real world occur. Figure 11a shows the first frame (also used as the initial model) of this sequence along with two other frames (#37 and #69) where the topological changes are illustrated. As shown in figure 11b the model mesh is correctly estimated. In this sequence the size of the deformation-free area (yellow) was set to 2.

The final dataset considered shows a man and a kid playing with a ball. Meshes were obtained using a 16-camera setup. Note that the sequence is quite complex with three deforming objects that interact. As a result, only some frames exhibit a correct topology and hence a model based approach would be likely to fail without strong prior knowledge. Figure 12a shows two consecutive frames of the sequence, the first one being used as the initial template. Figure 12b and 12c show that the two hands are correctly detached from the ball in the resulting model. In this case the size of the deformation-free area was set to 1.

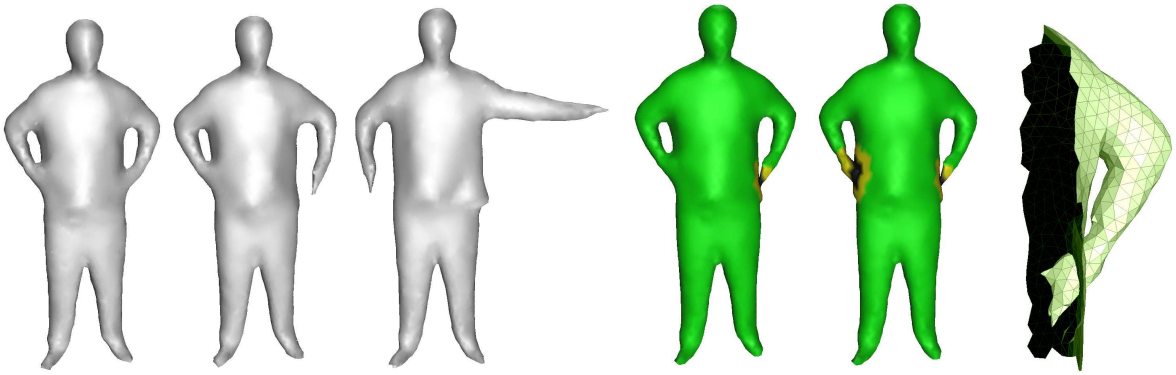
4.3. Quantitative Evaluation

Quantitative evaluation were also performed using the Hausdorff distance (see Table 1). The mean distance be-



(a) first frame (b) subsequent frame (c) model evolution (d) close-up

Figure 10: Results on data from the University of Surrey. (a) The first mesh, the initial model; (b) A subsequent mesh with a change in the topology; (c) The inferred progressive model evolution re-aligned with the first frame; (d) Close-up on the knee-elbow intersection.



(a) input (b) model evolutions (c) inside the model

Figure 11: Results on the INRIA data. (a)-1 The first mesh of the input sequence and also the initial model. (a)-2 and -3 The meshes (#37 and #69 respectively) of the input sequence where new topology information appears. (b) the respective evolutions of the progressive model aligned with the first frame. (c) close-up on the inside of the left arm of the model, we can see that the arm and hip are separated.

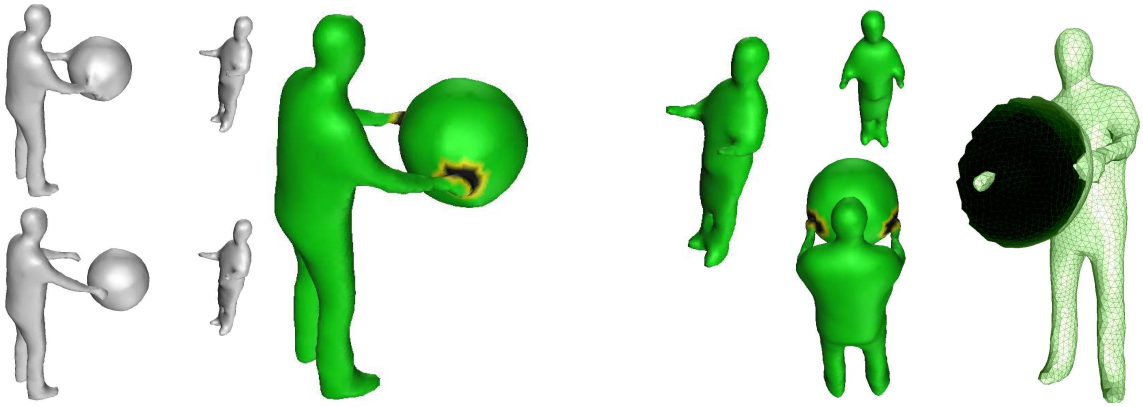
	average distance to input	average distance to ground truth
Spheres	0.003410	0.003194
Legs	0.002036	0.001952
Flashkick	0.000680	none
INRIA - man	0.000550	none
INRIA - man, kid & ball	0.000537	none

Table 1: Average distances between progressive models and input data as well as ground truth when available normalized by the diagonal of the bounding box.

tween the final model, aligned on the first observation, and the first input mesh was computed. Also for the synthetic sequence, an additional comparison with the ground truth is provided. The numbers given in table 1 are distances normalized by the diagonal of the model bounding box [2]. The results show that the distances are very small, mainly due to the precise alignment step §3.4. Note that values are higher for the synthetic datasets than for real datasets due to the low mesh resolution that increases noise effect in the Hausdorff distance. It is also worth noticing that for the synthetic data the final progressive model is closer to the ground truth than it is to the observed meshes. This demonstrates the ability of the method to progressively learn a model that converge to the true shape by accumulating evidences over a time sequence of incomplete observations.

5. Conclusion

This paper introduced progressive shape models. Based on the assumption that real object topologies are fixed they allow to learn both the evolving geometry and topology of a shape along temporal sequences. This provides a useful information for any subsequent mesh processing steps such as surface tracking, scene flow estimation or marker-less motion capture. Experiments on synthetic and real datasets



(a) input (b) model evolution (c) inside the model

Figure 12: Results on the INRIA data. (a) The first two meshes of the sequence (b) two views of the new template, note that the topology evolved in two different areas during the a single model improvement pass. (c) Inside view of the improved model, this view shows that the ball and the man are two separated objects.

demonstrate the effectiveness and the accuracy of the proposed method.

Yet the approach relies on purely geometric information. However most of the datasets obtained using multi-camera setups comes with photometric information. We believe that the proposed method could benefit from such photometric information, either during the registration step §3.2 or during the precise alignment step §3.4. In the latter case, a multi-view stereo algorithm could be used to improve the model, not only at a coarse level for the topology, but also at finer levels to recover small surfaces details.

References

[1] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, and J. Davis. Scape: Shape completion and animation of people. *ACM Trans. Graphics (Proc. SIGGRAPH)*, 2005. 1

[2] N. Aspert, D. Santa-Cruz, and T. Ebrahimi. Mesh: Measuring errors between surfaces using the hausdorff distance. In *IEEE ICME*, 2002. 7

[3] A. M. Bronstein, M. M. Bronstein, and R. Kimmel. Calculus of non-rigid surfaces for geometry and texture manipulation. *IEEE TVCG*, 2007. 1

[4] C. Cagniart, E. Boyer, and S. Ilic. Probabilistic Deformable Surface Tracking From Multiple Videos. *IEEE ECCV*, 2010. 1, 2, 4

[5] 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 (Proc. SIGGRAPH)*, 2008. 1, 2

[6] Y. Furukawa and J. Ponce. Dense 3D Motion Capture for Human Faces. *IEEE CVPR*, 2009. 2

[7] H. Li, B. Adams, L. J. Guibas, and M. Pauly. Robust Single-view Geometry and Motion Reconstruction. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 2009. 2

[8] H. Li, L. Luo, D. Vlasic, P. Peers, J. Popovič, M. Pauly, and S. Rusinkiewcz. Temporally Coherent Completion of Dynamic Shapes. *ACM Trans. Graphics (Proc. SIGGRAPH)*, 2011. 2

[9] N. J. Mitra, S. Flöry, M. Ovsjanikov, N. Gelfand, L. Guibas, and H. Pottmann. Dynamic geometry registration. In *Proceedings of the Symposium on Geometry Processing*, 2007. 2

[10] T. Popa, I. South-Dickinson, D. Bradley, A. Sheffer, and W. Heidrich. Globally Consistent Space-Time Reconstruction. *Computer Graphics Forum*, 2010. 1, 2, 4

[11] S. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A Comparison and Evaluation of Multi-View Stereo Reconstruction Algorithms. In *IEEE CVPR*, 2006. 1

[12] A. Sharf, D. A. Alcantara, T. Lewiner, C. Greif, A. Sheffer, N. Amenta, and D. Cohen-Or. Space-Time Surface Reconstruction using Incompressible Flow. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 2008. 2

[13] A. Sharma, R. Horaud, J. Cech, and E. Boyer. Topologically-Robust 3D Shape Matching Based on Diffusion Geometry and Seed Growing. In *IEEE CVPR*, 2011. 1

[14] J. Starck and A. Hilton. Surface Capture for Performance-Based Animation. *IEEE CGA*, 2007. 5

[15] D. Vlasic, I. Baran, W. Matusik, and J. Popović. Articulated Mesh Animation from Multi-view Silhouettes. *ACM Trans. Graphics (Proc. SIGGRAPH)*, 2008. 1, 2

[16] M. Wand, P. Jenke, Q. Huang, M. Bokeloh, L. Guibas, and A. Schilling. Reconstruction of deforming geometry from time-varying point clouds. In *Proceedings of the Symposium on Geometry Processing*, 2007. 2

[17] A. Zaharescu, E. Boyer, and R. Horaud. Topology-Adaptive Mesh Deformation for Surface Evolution, Morphing, and Multi-View Reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2011. 4, 5

[18] Q. Zheng, A. Sharf, A. Tagliasacchi, B. Chen, H. Zhang, A. Sheffer, and D. Cohen-Or. Consensus Skeleton for Non-rigid Space-time Registration. *Computer Graphics Forum*, 29, 2010. 2