3D Object Metamorphosis with Pseudo Metameshes

Bogdan MOCANU[1,2], Ruxandra TAPU[1,2], Titus ZAHARIA[2]
[1]*Faculty of ETTI, University "Politehnica" of Bucharest, 060042 Romania*
[2]*Institut Mines-Télécom / Télécom SudParis, ARTEMIS Department, UMR CNRS MAP5 8145, France*
ruxandra_tapu@comm.pub.ro, bcmocanu@comm.pub.ro, titus.zaharia@telecom-sudparis.eu

***Abstract*—In this paper we introduce a novel framework for 3D object metamorphosis, represented by closed triangular meshes. The systems returns a high quality transition sequence, smooth and gradual, that is visual pleasant and consistent to both source and target topologies. The method starts by parameterizing both the source and the target model to a common domain (the unit sphere). Then, the features selected from the two models are aligned by applying the CTPS C^2_a radial basis functions. We demonstrate how the selected approach can create valid warping by deforming the models embedded into the parametric domain. In the final stage, we propose and validate a novel algorithm to construct a pseudo-supermesh able to approximate both, the source and target 3D objects. By using the pseudo-supermesh we developed a morphing transition consistent with respect to both geometry and topology of the 3D models.**

***Index Terms*—3D mesh morphing, spherical parameterization, radial basis function, pseudo-supermesh.**

I. INTRODUCTION

In the case of 3D meshes, the term morphing refers to the change in appearance of a graphical object, between the source and the target geometries. The objective of morphing is then defined as the construction of an animated sequence corresponding to the gradual transition between two different objects.

Existing professional animation environments, such as 3DS Max or Lightwave, propose some basic morphing techniques. However, such methods cover only partially the aspects that need to be taken into account. In particular, they are able to morph solely meshes with the same topology and number of vertices and thus severely restrict the field of possible applications. Thus, one important objective is to make possible to morph 3D models described by different numbers of vertices and connectivities.

Because of such specificities, an initial stage is here required, which consists of establishing a correspondence between the two, source and target, 3D discrete surfaces defined by the meshes. Such a correspondence cannot be directly defined, because of the complexity of the topological and geometrical information involved. Instead, we determined the correspondence in an indirect manner with the help of parameterization techniques, which consists of establishing a one-to-one mapping between the mesh surface and a common 2D domain.

The process of object parameterization has a significant importance, conditioning directly the quality of the entire morphing sequence. Most parameterization algorithms, existent in the technical literature, are time consuming especially for complex 3D models. So, in order to reduce the processing time a mesh simplification step is required.

However, parameterizing the source and target meshes onto a common parametric domain does not entirely solve the correspondence problem. The mesh geometry (i.e., position of the mesh vertices in the 3D space) can be defined in arbitrary coordinate systems. Thus, a preliminary normalization and alignment processes are also required. Moreover, a feature alignment phase is necessary in order to guarantee a successful morphing process. This comes to: (1) define a set of interest features on both source and target models and (2) apply a warping/deformation of the parametric domain in order to guarantee that the parametric position of the corresponding features are as closed as possible for both models. We speak in this case of overlaid parameterizations. The features of interest are in general sets of points, lines, curves, regions, defined over the models to be morphed. They correspond to intuitive, semantic morphological characteristics.

In the case where the morphing sequence is considered for two face models, the features put in correspondence are, in most of the cases, the regions: mouth, nose, eyes or ears, common in both objects.

Once the source and target models are parameterized and aligned with respect to their corresponding features of interest, the final step necessary in the morphing process is the interpolation between objects. The simplest way to interpolate between these points is a linear interpolation.

In this paper, we propose a new 3D mesh morphing methodology which addresses all the above-presented aspects. The obtained morphing sequence ensures a high quality transition, smooth, consistent with respect to both geometry and topology, and visually pleasant.

The rest of this paper is organized as follows: after a synthesis of the 3D morphing techniques proposed in the state of the art (Section 2), in Section 3 we introduce and detail the proposed method. Section 4 presents the experimental results obtained, carried out on a test set of various 3D objects from the Princeton and MPEG 7 databases. Finally, Section 5 concludes our paper and opens some perspectives for future work.

II. RELATED WORK

The morphing process involves two main steps: (1) correspondence between the input, source and target models

The work has been funded by the Sectoral Operational Programme Human Resources Development 2007-2013 of the Ministry of European Funds through the Financial Agreement POSDRU/159/1.5/S/132395 (InnoRESEARCH).

Digital Object Identifier 10.4316/AECE.2015.01016

and (2) interpolation, in order to create intermediary states of morphing.

In most of the cases the correspondence issue is solved by employing different parameterization techniques that take into account the models topology. The parameterization is defined as a mapping Ω:M→D of a 3D model M onto a 2D parametric domain D. Most often, the domain D is either the unit disc (planar parameterization), or the unit sphere (spherical parameterization). The planar parameterization is useful in the case of 3D meshes that define an open surface with a unique connected component and border. Spherical parameterizations are necessary in the case of closed, connected 3D surfaces with genus-0 topology.

Concerning the first category, Kanai *et al.* [1] propose to use the harmonic mapping method in morphing of arbitrary triangular meshes with a topology equivalent to a disk. Here, the user has to specify a boundary loop for each object together with a boundary control vertex, which allows the models alignment. Each mesh is then embedded into a planar unit disk with the help of harmonic maps [2]. The boundary vertices are mapped onto the unit disc border, such that the angle formed by two successive vertices and the domain central point is proportional with the arc length determined by the considered vertices. The remaining (interior) vertices are mapped onto the interior of the unit disc by minimizing the total mesh energy. Once the harmonic maps of both source (M_S) and target (M_T) 3D meshes are computed, a new object, called supermesh or metamesh, is created by overlapping and merging the two mappings. The supermesh shares the connectivity of both original models and defines a one to one correspondence between the surfaces of the two meshes.

The merging algorithm proposed by Kent *et al.* [3] is based on the assumption that, after overlapping the parameterizations, no parametric vertices of the two models are coincident, and no parametric vertex of one model lays on an edge of the other model. However, this hypothesis limits its applicability in practice.

In order to overcome such a limitation, Kanai *et al.* [1] propose a slightly different method which is able to take into account coincident vertices/edges. In order to avoid numerical errors, the coincident vertices are first determined. The source and target parameterizations are then re-calculated by maintaining these vertices fixed to an average position. The operation is iterated until no coincident vertices are generated. The supermesh construction is performed similarly to [3].

For closed, genus-0 objects, which are topologically equivalent to the unit sphere, Alexa [4] proposes to establish the correspondence in the spherical domain. The mesh vertices are projected onto the unit sphere and then a relaxation process is performed that repeatedly places each vertex in the centre of its neighbors. In order to avoid triangle overlapping or mesh collapse problems, Alexa defines sets of anchor points in the parametric domain, which are modified several times during the relaxation process. Based on the feature pair vertices specified manually by user in both source and target models, the problem of feature alignment is established in the parametric domain with the help of the mesh deformation technique described in [5]. Next, in order to be able to transform from one object to another, a supermesh is constructed by overlapping the spherical embedding of the two models. Here, the problem of edge-to-edge intersection is transformed into a problem of arc-to-arc intersection. Finally, the morphing sequence is obtained using a linear interpolation scheme.

In all of the above-cited approaches, the resulting supermesh has a total number of vertices equal to $N_{M_1} + N_{M_2} + N_{Int}$, where N_{M_1} and N_{M_2} are the number of vertices in source and target respectively, and N_{Int} is the number of edge intersections. In practice, the number of edge intersection is prohibitively high, which leads to huge meshes in the case of objects described by a large number of faces/vertices.

In [6] the authors introduce a new perspective of the morphing process. In addition to the geometric transformation between models, they propose to gradually change the mesh connectivity. Using this strategy, the proposed method is creating an in-between mesh with a complexity inferior to a supermesh.

Even if it is not directly related to mesh morphing the method proposed by Wu et al. [7] aims at providing a solution for establishing a correspondence between arbitrary 3D meshes. In order to establish a shape-preserving correspondence between source and target meshes, a modified mean-value Laplacian fitting scheme is used. This operation is applied directly in the 3D space without requiring any model parameterization.

Athanasiadis *et al.* [8] introduce a completely automatic technique for 3D mesh morphing, which works well only for similar input models.

Using the method of concavity intensity, firstly introduced in [9], the system performs automatically the objects alignment, feature detection and feature point matching. Then, by analyzing the variation speed of the surface normal, the proposed framework introduces a region growing method used for extracting clusters of interest points, for each individual characteristics put in correspondence. The relations between the extracted features are represented using a connectivity graph.

The analysis of the state of the art shows that several phases involved in the morphing process are crucial for ensuring the quality of the resulting metamorphosis sequence. They notably concern: (1) the parameterization method involved, which should guarantee low geometric distortions in terms of lengths, angles and areas, (2) the warping of the source and target parametric domains, which should simultaneously guarantee a good match between corresponding feature points and a fold-over free deformation and (3) the connectivity-related issues which should ensure a smooth and local adaptation between source and target models.

Based on these considerations, we propose the 3D mesh morphing framework described in the following section.

III. Proposed 3D Mesh Morphing Method

The complete morphing process is illustrated in Fig. 1. First, by employing the user interaction some vertices are selected from the source and the target models and put into correspondence. The end user can interact with the models

by using the GUI interface of the proposed system. Because the entire framework is developed as a real time application we introduce next a decimation step designed to decrease the computation speed of the parameterization process. After simplification, each 3D object is parameterized to the unit sphere using our method, previously introduced in [10], which exploits a modified version of the Gaussian curvature.

Next, the mesh structure is reconstructed through a progressive mesh sequence which optimally reinserts the vertices removed in the simplification process. The characteristic features of the two models are aligned within the parametric domain by employing a warping scheme based on the radial basis functions. Here, additional constraints are introduced in order to maintain all vertices on the sphere during the deformation and to satisfy the bijective property. We introduce next a novel and simple method that creates a pseudo supermesh structure starting from the two overlapped embeddings of the models. The final step concerns the interpolation of the geometric positions of the source and target vertices onto the involved pseudo supermesh.

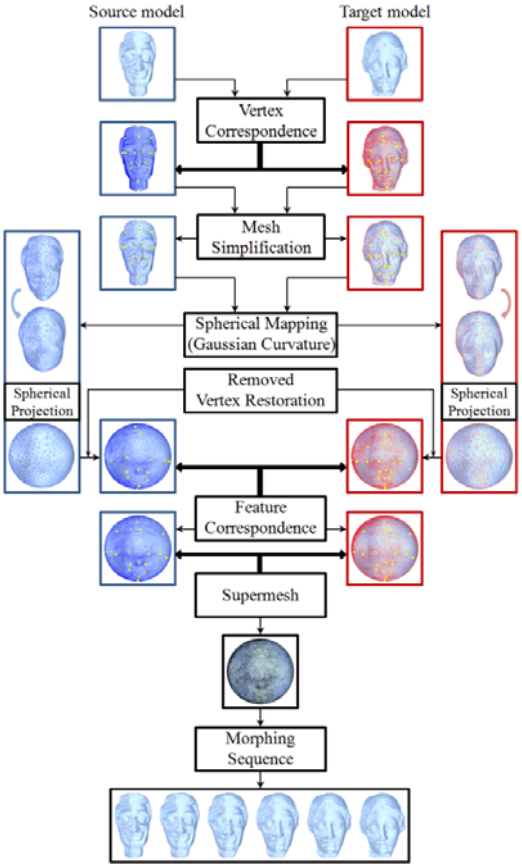

Figure 1. Proposed 3D mesh morphing scheme

The various steps involved in the proposed 3D mesh morphing algorithm are detailed in the following sections.

III.1. Feature Alignment

This phase is essential to guarantee a successful morphing transformation between the source and the target input models. Let us note that the necessity of feature alignment becomes more obvious when the models belong to the same semantic category. This is because the user has a strong a priori expectation of that transformation, and anticipates the preservation of common features of the models.

By using a graphical interface the end user can specify the features that need to be put in correspondence. Different authors [11], [12] try to automate the process for particular 3D objects that share a similar structure by performing preliminary normalization and a global alignment of the models. However, these approaches are limited for a general morphing process between arbitrary models since the global alignment will not be able to position the corresponding features into the same region. For this reason, we have developed an ergonomic graphical user interface with all the necessary interactivity features. Fig. 2 presents a screenshot of our interactive application.

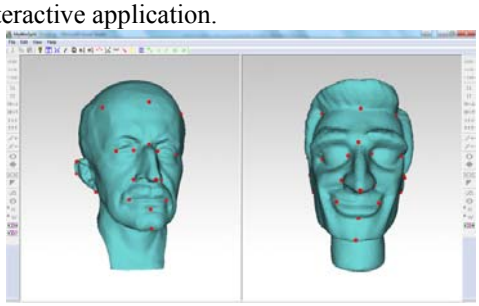

Figure 2. Graphical user interface

Two sets of corresponding feature points are here specified by the user on both source and target models.

Let us note that this step is the only one that requires the user intervention.

III.2. Mesh Simplification

The mesh simplification step is introduced in order to increase the processing speed of the parameterization process. Because the proposed framework is designed to work with any type of 3D model (characterized by complex structures or containing thousands of vertices and faces) we developed a simplification step that generates as output a coarser version of the original model. The parameterization obtained for the simplified mesh will be afterwards adjusted to handle the original vertices with the help of a reversed simplification process, which consists of iteratively inserting the removed vertices in the parametric domain.

The proposed technique relies on the edge collapse operator introduced by Hoppe *et al.* [13]. However, the original technique proposed in [13] involves a time consuming energy minimization process, needed to establish the edges that have to be removed from the mesh structure. For this reason, we have adopted the method introduced in [14] that uses the quadric error metric in order to compute the edge collapse cost. In contrast with [14], which allows contraction of any pair of vertices, in our case we authorize the collapse of solely vertices that are linked by an edge. This constraint allows us to preserve the original manifold characteristics of the models.

The simplification operation is controlled by a geometric criterion. Thus, the simplification process stops when the mean geometry deviation between two simplified versions of the original model exceeds a pre-established threshold T_{err}. The geometric error is computed between the two model surfaces S_1 and S_2 as:

$$e(S_1,S_2)=\sqrt{\frac{1}{Area(S_1)}\sum_{p\in S_1}d^2(p,S_2)} \qquad (1)$$

where d represents the approximate distance from a point p on surface S_1 to the surface S_2 given by:

$$d(p,S_2) = \min_{p' \in S_2} dist(p,p') \quad (2)$$

In our work, we have used a value for T_{err} of 0.0025. This constraint ensures that the obtained model still preserves the main geometrical characteristics of the initial object. The proposed decimation algorithm yields high quality results for the simplified model (Fig. 3) even for drastic decimation rates, in a relatively short time.

Figure 3. Model simplified with our mesh decimation technique

III.3. Embedding and Warping

For each vertex on the source (resp. target) model, a correspondent point on the target (resp. source) surface has to be identified. In our case, we adopt an indirect mapping method which consists of: (1) parameterizing both source and target models onto a common, spherical parametric domain; (2) warping the parametric domains in order to ensure a feature correspondence between the two 3D shapes to be morphed.

We start by employing our method, firstly introduced in [10], that creates spherical parameterizations for closed genius-0 3D models. Next, we focused on preserving the input features (specified by the user). The vertices already put in correspondence in Section III.2 (denoted anchor vertices) need to be displaced so that they share the same position in the parametric domain. In order to make a global displacement of the entire set of vertices the whole domain needs to be deformed. The resulted 2D maps should be smoothly warped and without foldovers.

In the technical literature various deformation methods were introduced, that can be classified in: space deformation [15], free-form warping [16], skeletal [17] or multi-resolution deformations [18], Laplacian mesh editing [19], and radial basis functions (RBF) [20]. From this set we have considered as appropriate for our 3D mesh morphing the RBF and Laplacian techniques because they do not require additional human interaction for the object editing. In the following section we give an objective comparative evaluation of these techniques.

Regarding the RBF functions, we have selected three radial basis functions, namely the CTPS C^2_a, CP C^2, and Gaussian [20]. Concerning the Laplacian deformation we have considered the classical form introduced in [19] that solves the deformation problem in the sense of least squares minimization.

However, such an approximate solution may affect the displacement accuracy (i.e., the anchor vertices may not reach the exact final position established). For this reason, we have also considered a modified version, in which the final positions of the control vertices are imposed as additional constraints.

In addition, two types of weights [19] are considered for specifying the Laplacian coordinates, including unitary weights (so-called Uniform Fix Laplacian Coordinate - UFLC) and mean value coordinates, so-called Tangential Laplacian coordinate deformation method (TLC).

The six retained techniques have been tested in terms of deformation quality, computational efficiency and displacement accuracy. As evaluation measures, we considered the size f_{size} and shape f_{shape} quality metrics [21] respectively defined as:

$$f_{size} = \min(\tau, \frac{1}{\tau}) \quad (3)$$

$$f_{shape} = \frac{\sqrt{3} r \sin\theta}{1 - r\cos\theta + r^2} \quad (4)$$

where τ represents the ratio between the area of a triangle in the deformed mesh and the area of the initial triangle. The angle θ is defined as:

$$\cos\theta = \frac{\lambda_{12}}{\sqrt{\lambda_{11}\lambda_{22}}} \quad (5)$$

and

$$r = \frac{\lambda_{11}}{\lambda_{22}} \quad (6)$$

with λ_{ij} , i,j = 1,2 being the metric tensor of the Jacobian matrix.

The measures f_{size} and f_{shape} can be combined to define a more general quality metric f_{ss}, defined as:

$$f_{ss} = f_{size} \cdot f_{shape} \quad (7)$$

Ideally, the f_{ss} value should be as close as possible to 1. Both the minimum ($min(f_{ss})$) and mean ($mean(f_{ss})$) values of the f_{ss} values, over the entire mesh, are retained.

In order to evaluate the displacement accuracy of each deformation technique, i.e., the average error between the actual displacement and the specified positions of the anchor points, we have also considered the following measure:

$$Dis = \frac{1}{n_c} \sum_{i=0}^{n_c} \frac{\left\| p_{c_i} - p_{c_i}^{dis} \right\|}{\left\| p_{c_i} - p_{c_i}^{init} \right\|} * 100 \quad (8)$$

where n_c denotes the total number of control vertices, $p_{c_i}^{init}$ and p_{c_i} respectively, represent the initial and target positions of point i, while $p_{c_i}^{dis}$ is the actual position reached. Ideally the value of Dis should be zero.

For evaluating the various methods, we have considered two test scenarios, one in 2D (and thus adapted to purposes of mesh warping in the parametric domain) and the other in 3D. Both scenarios employ a unit square domain with an inner rectangle (as control feature) that undergoes a high amplitude deformation corresponding to a combined rotation (with 90°), translation (12 intervals along the x direction, 4 intervals along the y axis (and for the 3D case, 10 units in the z direction)) and scaling (of factor of 2) (Fig. 4 and Fig. 5).

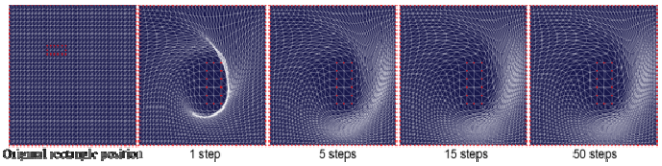

Figure 4. The influence of the number of steps on the deformed mesh

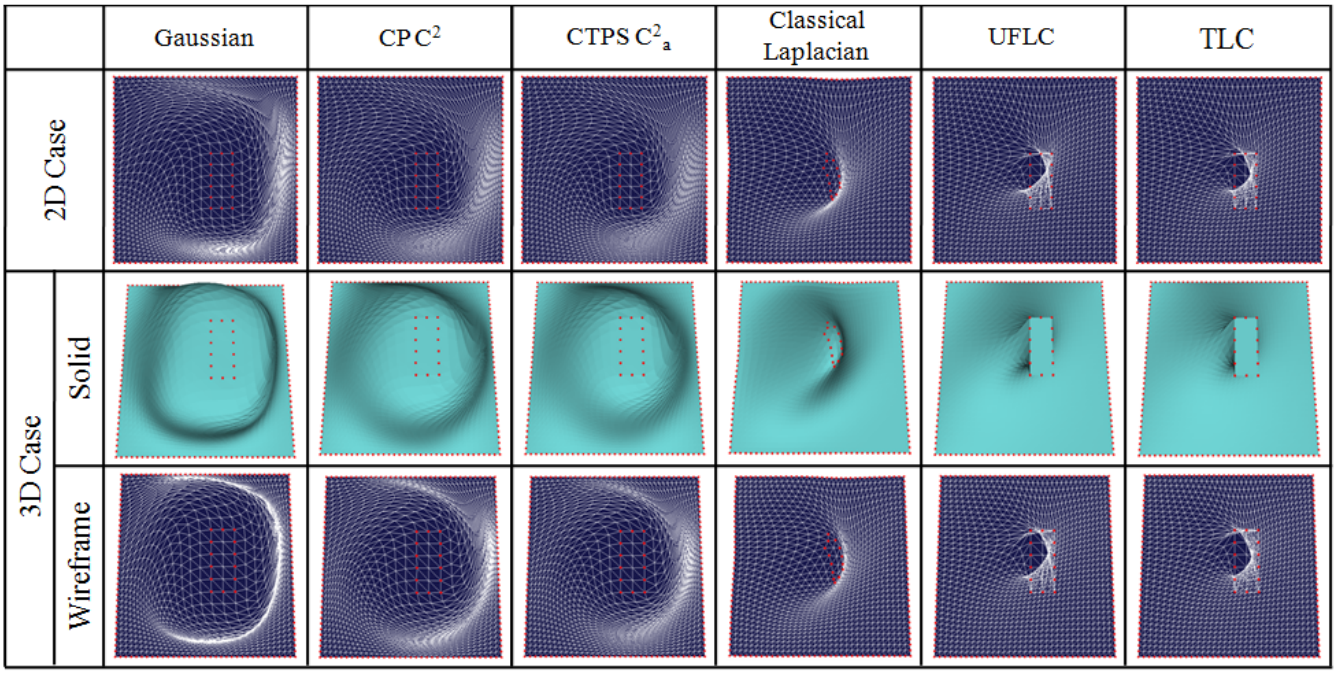

Figure 5. Visual analysis of mesh deformation

In addition, we perform the mesh deformation in a variable number of steps that iteratively displace the control vertices from their initial to the final positions. The number of intermediary steps ranges from 1 to 50 steps, and the distance between two positions is uniformly sampled. Figure 4 illustrates, in the 2D case, the influence of the number of steps on the deformed mesh for the CTPS C^2_a RBF method.

We can observe that a higher number of intermediary steps increases the quality of the deformation result, which is more smooth and foldover-free. However, starting from a number of 10 to 15 steps, the overall quality becomes equivalent.

Fig. 5 and Table I present the results obtained for both 2D and 3D scenarios. Each deformation method has been here applied in a number of 10 steps.

TABLE I. MESH DEFORMATION QUALITY ANALYSIS

Method	$minf_{ss}$	$meanf_{ss}$	*dis*	Inverted triangles	CPU Time (sec)
2D test case					
RBF-CTPS C^2_a	0.021	0.247	$8.34*^{10-6}$	0	21.1
RBF-CP C^2	0.017	0.232	0.334	0	18.6
RBF-Gaussian	0	0.198	0.043	22	21.3
Classical Laplacian	0.001	0.331	20.501	0	0.8
UFLC	0	0.354	$6.26*10^{-9}$	60	0.9
TLC	0	0.355	$1.03*10^{-9}$	51	1.5
3D test case					
RBF-CTPS C^2_a	0.118	0.465	$3.36*10^{-6}$	0	19.7
RBF-CP C^2	0.104	0.438	0.168	0	19.1
RBF-Gaussian	0.064	0.426	0.012	0	23.8
Classical Laplacian	0.228	0.613	14.751	0	1.12
UFLC	0.047	0.623	$4.53*10^{-9}$	0	0.31
TLC	0.045	0.618	$8.82*10^{-10}$	0	0.89

For the 2D case, the classical Laplacian method leads to a valid deformed mesh, but with relatively high distortions (50% greater than the RBF-related distortions) and also large values of the accuracy parameter *Dis*. For UFLC and TLC techniques a mesh overlapping is produced. This is caused by the strict conditions imposed on the control points, which are forced to reach exactly the target position. Concerning the RBFs, CTPS C^2_a and CP C^2, these return comparable results in terms of $min(f_{ss})$ and $mean(f_{ss})$, but a higher accuracy is obtained for CTPS C^2_a.

For the 3D case, the UFLC and TLC approaches avoid triangle overlapping, since the warping is performed in the 3D space. Moreover, they return the best results for the displacement accuracy. In terms of $mean(f_{ss})$ measures, these methods return comparable results as the ones obtained by the classical Laplacian, but with a $min(f_{ss})$ inferior with more than 68%.

Finally, the experimental results show that RBF method (and, in particular the CTPS C^2_a function) offers a good compromise between displacement accuracy and mesh quality. We have thus adopted it for warping of the meshes in the parametric domain.

The CTPS C^2_a RBF is applied in steps, by splitting the geodesic path between feature-pairs of correspondent vertices in 10 subintervals. This makes it possible to avoid fold-overs and self intersections.

In order to guarantee that the embedding remains valid, the deformed mesh is back-projected onto the unit sphere after each step.

Fig. 6 illustrates an example of two 3D closed genus-0 models with user-specified feature vertices, initial spherical mappings and final embeddings.

Let us note that the corresponding pairs of vertices become aligned in the parametric domain after applying the warping approach.

Next, we determined a one-to-one correspondence between the two models used for morphing (denoted by source and target) by overlapping the two embeddings.

For each vertex of the source model (respectively target model), in the parametric domain, we determined it corresponding triangle in which it can be projected in the target domain (respectively source domain). By using the barycentric coordinates we determine next the vertex

position on the original target surface. Because we want to create a 3D mesh morphing application for any type of model applied as input we cannot use a linear interpolation between the initial and final positions of the source (respectively the target) vertices. On the one hand, the structure of one object is insufficient to create visually pleasant transition and on the other hand, the approximation is inaccurate.

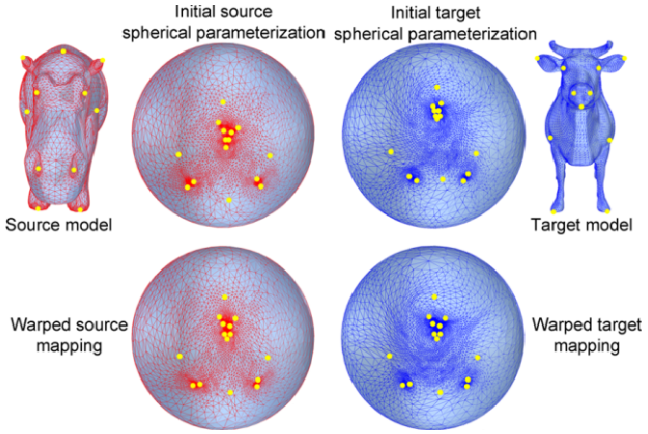

Figure 6. Feature alignment through spherical embeddings warping

In order to solve this issue a new mesh structure, usually called supermesh, which can represent both models, needs to be constructed.

III.4. Pseudo-Supermesh Construction

In the state of the art literature various authors [4], [11], [22] propose overlapping the two maps associated to the source and the target model in order to create a supermesh. Then, a new structure is developed by adding the edges of the target model (resp. source) in the structure of the source model (resp. target).

The created supermesh has the advantage of sharing the two topologies of the models used in the morphing application. However, because the size of the two models could exceed thousands of vertices the numerical instabilities the process of supermesh development becomes important (e.g. when computing intersections between source and target edges). Even so, the number of vertices of the resulted metamesh structure will drastically increase compared with the original models.

In this paper we introduced a novel approach called pseudo supermesh that tries to overcome the above limitation especially the creation and tracking of edge intersection. By using the proposed strategy the number of resulted vertices is significantly inferior to the one obtained in the case of the supermesh. We call our structure pseudo-metamesh since it does not require any edge intersections, and only approximates the two source and target shapes.

The pseudo supermesh construction process can be summarized as follows: First, we initialize the new structure with the one of the target embedding. Next, for any source parametric vertex, we determine the supermesh triangle in which it can be projected by employing a ray-triangle intersection test. Once we determine the face in which a source vertex falls, the triangle is split after a 1-to-4 scheme, as illustrated in Figure 7.

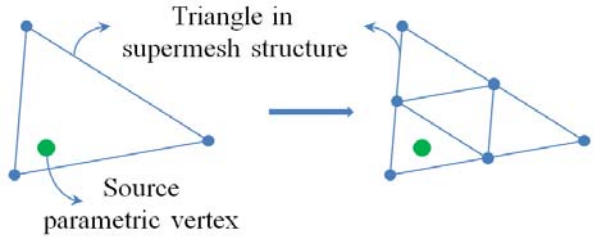

Figure 7 : Supermesh triangle split after 1-to-4 scheme

This operation is repeated until all source vertices are processed such that each triangle in the subdivided mesh includes a unique vertex of the source mesh. Obviously, the obtained pseudo-supermesh does not have a valid triangular mesh structure since, after a triangle subdivision the adjacent faces are not triangles anymore. Thus, a mesh retriangulation is performed, which consists of linking each mid-edge vertex generated by a triangle split to the vertex opposite to the considered edge in the adjacent triangle. This process is illustrated in Figure 8.

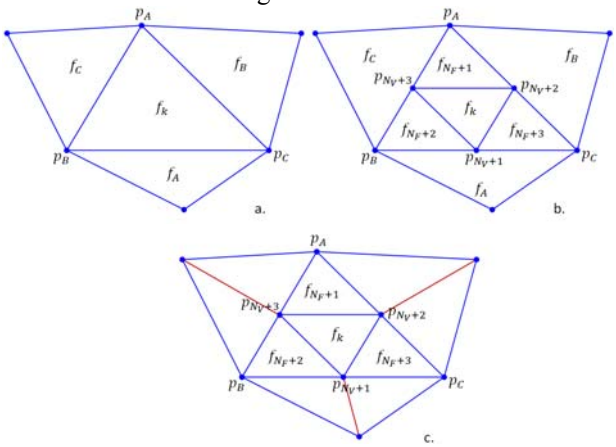

Figure 8 : Supermesh retriangulation process: (a) initial pseudo-supermesh; (b) pseudo-supermesh after 1-to-4 subdivision; (c) retriangulated pseudo-supermesh.

Once the pseudo-supermesh is created, we establish the 3D positions of its vertices relatively to the source and target shape. Actually, for the majority of its vertices we already know their positions relatively to the target shape since the supermesh was initialized with the target structure. The positions of the new vertices relatively to the target surface can be easily computed since after each split operation, the new vertices are inserted at the middle of an existing edge. The 3D position for each pseudo-supermesh vertex relatively to the source shape is computed employing a point-in-triangle test and using the barycentric coordinates (α, β, γ). Considering that a parametric supermesh vertex p^{H_M} lays in the source spherical domain (H^S) on a triangle $f^{H_S}(p_i^{H_S}, p_j^{H_S}, p_k^{H_S})$ then its position p^S relative to the original face $f^S(p_i^S, p_j^S, p_k^S)$ on the source shape can be computed using the barycentric coordinates:

$$p^S = \alpha p_i^S + \beta p_j^S + \gamma p_k^S \quad (9)$$

Fig. 9 illustrates an example of a pseudo-supermesh obtained with the proposed algorithm. It can be observed that the mesh structure remains simple without too many vertices and the supermesh feature regions are adaptively remeshed accordingly with the input models.

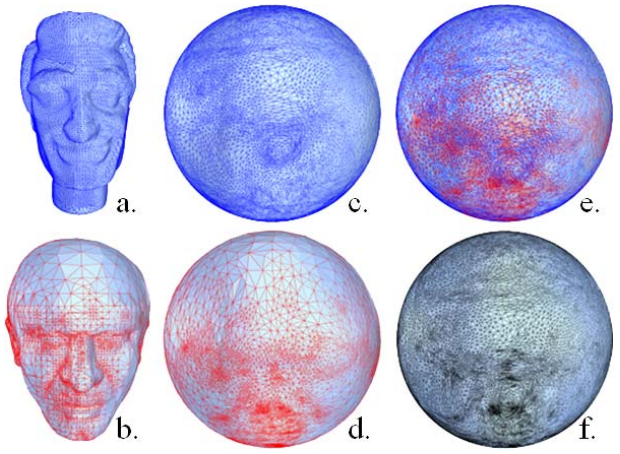

Figure 9. Pseudo-supermesh: (a) source model, (b) target model, (c) source map, (d) target map, (e) overlapped maps, (f) final pseudo-supermesh.

The main advantage of the pseudo-supermesh consists on its capability of accurately approximate both the source and target topologies. The supermesh will represent in the morphing sequence the source model at the first frame and the target model at the last frame. For intermediary frames, the vertex positions are linearly interpolated between their initial and final states.

Let us note that more sophisticated interpolation schemes, such as the ones introduced in [23], [24] can be considered in this framework.

IV. Morphing Results

Fig. 10-12 shows several key frames of the morphing sequences obtained between various 3D meshes.

The considered subset of objects consists of 3D closed genus-0 manifold models with various complexities and shapes. All the models are freely available over the Internet and are part of the Princeton and MPEG-7 databases.

The maximal number of feature points used for guiding the correspondence process was limited, respectively, to: 27, 24, 33 and 26 for models in Fig. 10-13.

Figure 10 : Morphing sequence between ManHead and Igea models

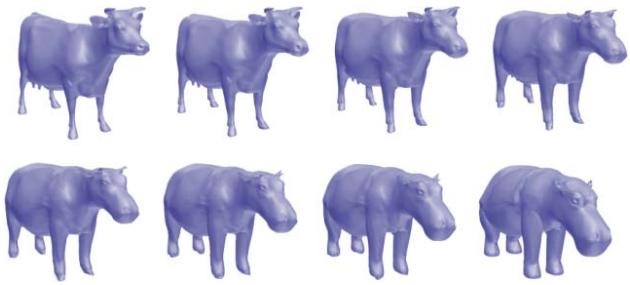

Figure 11 : Morphing sequence between Cow and Hipo models

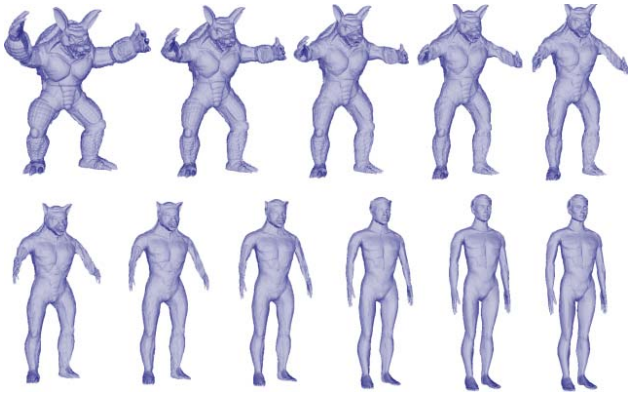

Figure 12 : Morphing sequence between Armadillo and Man models

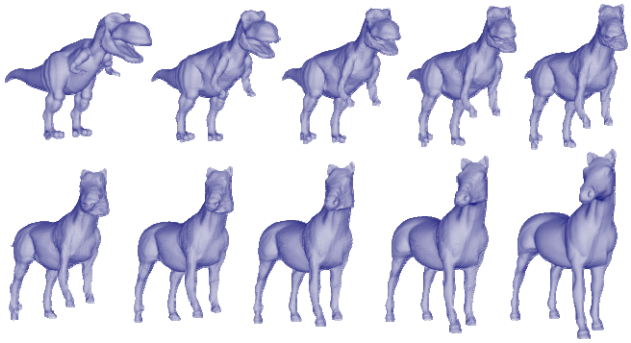

Figure 13 : Morphing sequence between Dino and Horse models

We can observe that in all cases the resulting morphing sequences ensure a gradual and visually pleasant transition between source and target models.

Table 2 presents the characteristics of some pseudo-supermeshes obtained, in terms of number of vertices and triangles compared with the original models.

Note that in most of the cases the pseudo-metamesh number of vertices does not exceed the sum of the source and target vertices, which is quite a remarkable result.

TABLE II. PSEUDO-SUPERMESH CHARACTERISTICS

	Model	Vertices	Faces	Pseudo Supermesh	
				Vertices	Faces
Source	Igea	15002	30000	27127	42250
Target	ManHead	17200	34396		
Source	Cow	11610	23216	20657	41310
Target	Hipo	8966	17928		
Source	Man	14603	29202	29411	58818
Target	Armadillo	15002	30000		
Source	Dino	16996	33988	31082	62080
Target	Horse	19851	39698		
Source	Head1	17358	34712	22467	44930
Target	Head2	7896	15788		

V. Conclusions and Perspectives

In this paper, we introduced a novel morphing technique that creates smooth and natural transitions between closed genus-0 3D models. The approach requires minimum user intervention which has to specify only some feature vertices of correspondence on both input models.

A spherical parameterization approach has been considered as intermediary step, followed by a feature matching that employs a mesh warping scheme based on

radial basis function.

Then, we have introduced a simple, yet efficient technique of mesh merging to create a pseudo supermesh structure that approximates well both source and target 3D shapes.

Once all the intermediary steps (feature alignment, mesh simplification and parameterization, warping, pseudo-supermesh construction and mesh interpolation) are performed as an offline process, the morphing sequence can be played in real time, continuously and smooth, whenever it is desired. The proposed algorithm can be used for specific activities in different areas like: medical industries (in order to construct detailed models of organs), movie industry (where objects are animated in such a manner to simulate the real world), video gaming industries (where 3D models are used as assets for computer games), scientific sector (for various simulation purposes), architecture industries (where they are needed to illustrate proposed buildings and landscapes) or CAD (in order to construct new devices, vehicles and structures based on predefined models).

Perspectives of future work concern: (a) the integration of more sophisticated interpolation mechanisms; (b) the re-triangulation of the pseudo-supermesh which leaves room for further optimization.

REFERENCES

[1] T. Kanai, H. Suzuki, F. Kimura, "Three-dimensional geometric metamorphosis based on harmonic maps," The Visual Computer, vol. 14(4), pp. 166–176, 1998. [Online]. Available: http://dx.doi.org/10.1007/s003710050132

[2] M. Eck, T. DeRose, T. Duchamp, H. Hoppe, M. Lounsbery, W. Stuetzle, "Multiresolution analysis of arbitrary meshes," Proc. SIGGRAPH, pp. 173–182, 1995. [Online]. Available: http://dx.doi.org/10.1145/218380.218440

[3] J. R. Kent, W. E. Carlson, R. E. Parent, "Shape transformation for polyhedral objects," Computer Graphics, SIGGRAPH Proceedings, vol.2, pp. 47-54, 1992. [Online]. Available: http://dx.doi.org/10.1145/133994.134007

[4] M. Alexa, "Merging polyhedral shapes with scattered features," The Visual Computer, vol. 16, pp. 26-37, 2000. [Online]. Available: http://dx.doi.org/10.1109/SMA.1999.749341

[5] N. Arad, D. Reisfeld, "Image Warping Using few Anchor Points and Radial Basis Functions," Computer Graphics Forum, vol.14, no.1, pp.23-29, 1995.

[6] C. H. Lin, T. Lee, "Metamorphosis of 3D polyhedral models using progressive connectivity transformations," IEEE Transaction on visualization and computer graphics, vol. 11(1), pp. 2-12, 2005. [Online]. Available: http://dx.doi.org/10.1109/TVCG.2005.12

[7] H.Y. Wu, C. Pan, Q. Yang, S. Ma, „Consistent correspondence between arbitrary manifold surfaces, " In ICCV, pp. 1–8, 2007. [Online]. Available: http://dx.doi.org/10.1109/ICCV.2007.4408908.

[8] T. Athanasiadis, I. Fudos, C. Nikou, V. Stamati, "Feature-based 3D morphing based on geometrically constrained spherical parameterization," Computer Aided Geometry Description, vol. 29, pp. 2-17, January 2012.

[9] V. Stamati, I. Fudos, "A Feature-Based Approach to Re-engineering Objects of Freeform Design by Exploiting Point Cloud Morphology," In Proc. of SPM 2007: ACM Symposium on Solid and Physical Modeling, Beijing, China, pp. 347-353, June 2007. [Online]. Available: http://dx.doi.org/10.1145/1236246.1236296

[10] B. Mocanu, T. Zaharia, "Direct Spherical Parameterization Based on Surface Curvature," *Workshop on Digital Media and Digital Content Management (DMDCM) 2011*, pp. 266-269, 15-16 May 2011. [Online]. Available: http://dx.doi.org/10.1109/DMDCM.2011.61

[11] R. Urtasun, M. Salzmann, P. Fua, "3D Morphing without user interaction," Eurographics Symposium on Geometry Processing 2004.

[12] T. Athanasiadis, I. Fudos, C. Nikou, V. Stamati, "Feature-based 3D morphing based on geometrically constrained sphere mapping optimization," 25th ACM Symposium on Applied Computing (SAC'10), Sierre, Switzerland, pp.1258-1265, 22-26 March 2010. [Online]. Available: http://dx.doi.org/10.1145/1774088.1774355

[13] H. Hoppe, "Mesh Optimization," In Proc. ACM SIGGRAPH, pp. 19-26, 1993. [Online]. Available: http://dx.doi.org/10.1145/166117.166119.

[14] M. Garland, P. S. Heckbert, „Surface Simplification Using Quadric Error Metrics," In 24th Annual Conference on Computer Graphics and Interactive, pp. 209-216, 1997. [Online]. Available: http://dx.doi.org/10.1145/258734.258849

[15] J. P. Lewis, M. Cordner, N. Fong, "Pose Space Deformation: A Unified Approach to Shape Interpolation and Skeleton-Driven Deformation," SIGGRAPH 2000 Proceedings of the 27th annual conference on Computer graphics and interactive techniques, no. 3, pp. 165-172, 2000. [Online]. Available: http://dx.doi.org/10.1145/344779.344862

[16] T. Ju, S. Schaefer, J. Warren, „Mean value coordinates for closed triangular meshes," ACM Trans. Graph., vol. 24, no. 3, pp. 561–566, 2005. [Online]. Available: http://dx.doi.org/10.1145/1186822.1073229

[17] O. Weber, O. Sorkine, Y. Lipman, C. Gotsman, "Context-Aware Skeletal Shape Deformation," Computer Graphics Forum vol. 26(3), pp. 265-274, 2007.

[18] D. Zorin, P. Schroder, W. Sweldens, „Interactive multiresolution mesh editing," In SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques, pp. 259–268, New York, 1997. [Online]. Available: http://dx.doi.org/10.1145/258734.258863

[19] O. Sorkine, "State-of-The-Art Report: Laplacian Mesh Processing," Eurographics appeared in Computer Graphics Forum, vol. 25(4), 2006.

[20] A. Boer, M.S. Schoot, H. Bijl, "Mesh deformation based on radial basis function interpolation," Computers & Structures, vol. 85, pp. 784-795, 2007. [Online]. Available: http://dx.doi.org/10.1016/j.compstruc.2007.01.013

[21] P. M. Knupp, "Algebraic Mesh Quality Metrics for Unstructured Initial Meshes," Finite Elements in Analysis and Design, vol.39, pp. 217-241, 2003. [Online]. Available: http://dx.doi.org/10.1016/S0168-874X(02)00070-7.

[22] Z. J. Zhu, M. Y. Pang, "Morphing 3D Mesh Models Based on Spherical Parameterization," International Conference on Multimedia Information Networking and Security, pp. 309–313, 2009. [Online]. Available: http://dx.doi.org/10.1109/MINES.2009.29

[23] M. Alexa, "Local control for mesh morphing," Proceedings of Shape Modeling International, pp. 209-215, 2001. [Online]. Available: http://dx.doi.org/10.1109/SMA.2001.923392

[24] J. Hu, L. Liu, G. Wang, "Dual Laplacian morphing for triangular meshes," Computer Animation and Virtual Worlds, vol.18(4/5), pp. 271-277, 2007. [Online]. Available: http://dx.doi.org/10.1002/cav.182