Controlled Metamorphosis between Skeleton-Driven Animated Polyhedral Meshes of Arbitrary Topologies

Denis Kravtsov, Oleg Fryazinov, Valery Adzhiev, Alexander Pasko and Peter Comninos

The National Centre for Computer Animation, Bournemouth University, United Kingdom

Abstract

Enabling animators to smoothly transform between animated meshes of differing topologies is a long-standing problem in geometric modelling and computer animation. In this paper, we propose a new hybrid approach built upon the advantages of scalar-field-based models (often called implicit surfaces) which can easily change their topology by changing their defining scalar field. Given two meshes, animated by their rigging-skeletons, we associate each mesh with its own approximating implicit surface. This implicit surface moves synchronously with the mesh. The shape-metamorphosis process is performed in several steps: first we collapse the two meshes to their corresponding approximating implicit surfaces, then we transform between the two implicit surfaces and finally we inverse transition from the resulting metamorphosed implicit surface to the target mesh. The examples presented in this paper demonstrating the results of the proposed technique were implemented using an in-house plug-in for Maya^{TM}.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation

1. Introduction

Polygonal models are widely used in computer animation. Static polygonal models are commonly animated using an underlying rigging-skeleton that controls the deformation of the polygonal mesh. As the rigging-skeleton interpolates between key-frames, it automatically deforms the shape of this mesh. This technique, known as skeleton-driven animation, enables the artist to produce complex animation sequences in a relatively easy way. Artistic computer animations and digital effects in films often include a smooth transformation or metamorphosis of one moving character into another. However, performing complex metamorphosis between animated meshes of arbitrary shapes and topologies remains a challenging research problem. There is no general solution for this problem to date. In this paper we concentrate on the metamorphosis between two animated meshes which results in a smooth time-variant transition of their shapes. No restrictions are imposed on the number of vertices, polygons or even the topologies of these meshes.

There is a set of well-established techniques to perform metamorphosis (3D morphing) between static 3D meshes. However, most of the existing approaches require the surface topology (genus and number of components) to remain unchanged during the transformation and some of these techniques require that a one-to-one correspondence between elements of the meshes be established prior to the metamorphosis process. Methods that are applicable to meshes with different topologies typically require some a priori knowledge of the topological changes to occur and/or the user intervention in the process.

On the other hand, approaches to metamorphosis for voxel models, implicit surfaces and level sets are independent of the topologies of the objects and can easily handle a change of the genus and of the number of object components in an automatic manner. Polygonal meshes of different topologies, when converted to level sets or implicit surfaces, can also be automatically metamorphosed from one form into another. However this approach can not be directly applied to animated meshes, especially in real time. The main problem is that the conversion to another representations at each time step is computationally expensive, especially if voxelisation or implicitisation (i.e., conversion to an implicit form) is involved.

The novel technique presented in this paper allows us to produce, with great ease, metamorphosing transitions between animated meshes of arbitrary topologies using

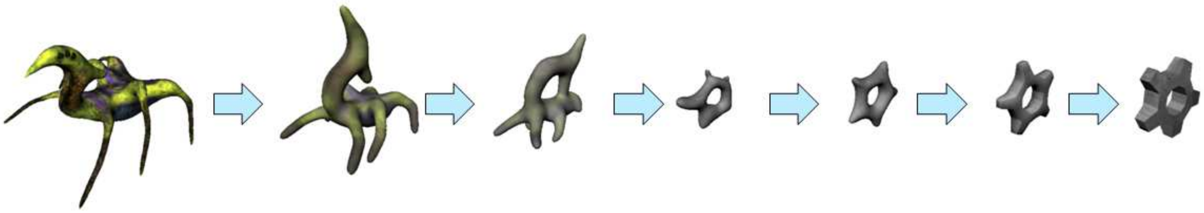


Figure 1: *Animation sequence illustrating controlled metamorphosis between animated meshes with varying topologies.*

polygonal-functional hybrid models. Our technique uses the meshes of the objects as well as their rigging-skeleton animations. As a result, we are able to generate metamorphosis animations of time varying meshes of arbitrary topologies in near real-time automatically or with minimal user intervention (see fig. 1, 12). It is important to note here that the problem of metamorphosis between animated meshes cannot be easily formalised. There is an infinite number of ways to perform a transition from one dynamic mesh to another. We can define a number of criteria which can be used in the search for an optimal solution. However, in practice artists prefer to have high-level control over the intermediate process in order to achieve specific effects. Hence we aim at introducing a practical method that can be used for the definition of general metamorphosis sequence between arbitrary animated meshes, while providing the user with additional high-level controls allowing him to achieve the desired effect according to his or her artistic vision.

The major contributions of this paper are: a novel technique utilising polygonal-functional hybrid models in several transition steps between model components in mesh metamorphosis and a set of algorithms for the forward and inverse transitions between an animated mesh and its functional approximation.

2. Related work

In this section, we briefly describe prior work related to the metamorphosis operation with different geometric models, which is related to the proposed technique.

2.1. Metamorphosis

From the first detailed survey on 3D metamorphosis [LV98] we can distinguish at least three types of approach that have been used in the metamorphosis process, namely methods based on polygonal meshes, scalar field based methods and conversion methods. Current approaches for the metamorphosis of polygonal meshes are predicated on one or more of the following assumptions: the equivalence of the topology of two given shapes (i.e. the two shapes are required to have the same genus and the same number of components), shape alignment (i.e. the two shapes are required to have a common coordinate origin and to have a significant overlap) and shape matching (i.e. there needs to be an established vertex-to-vertex feature correspondence between the two shapes). To deal with meshes with different topologies one typically needs to have some a priori knowledge of the topological changes that will occur and the user intervention to be applied to the process [DG96] [TKO01].

Discrete scalar fields (voxel models, level sets, etc.) allow for metamorphosis with arbitrary changes to the genus and to the number of components of the input shapes [Hug92] [YJ07]. Therefore, polygonal meshes converted to voxels using discrete 3D distance fields [COSL98] can also undergo arbitrary topological changes with no pre-established correspondence between their mesh vertices. However, such a conversion is time consuming and can not be directly applied to animated meshes, especially in real-time or in near-real-time, as it needs to be performed at each time step.

Implicit surfaces defined by continuous real functions appear to be the best suited for the most general type of topology-independent metamorphosis. Several algorithms for matching and interpolating skeletal soft objects have been proposed by Wyvill in [Blo97]. Some of these algorithms match shape elements according to their positions in space (cellular matching), while others require additional information regarding the shape elements (hierarchical matching). In such cases the intermediate implicit surface is generated during the process of interpolation of the main varying parameters (i.e., the positions of the skeletons and the field intensities). This method has the following major limitations: it may fail to match the components of different skeleton types and the shape components must be bijectively paired. A more general metamorphosis technique for soft objects, that was proposed in [GLA00], is constructed in the form of a BlobTree as a hierarchical combination of different skeleton primitives. This rather complex technique requires the creation of new overlapping BlobTrees, whose nodes and leaves are bijectively paired with the subsequent generation of an intermediate generic BlobTree with time varying nodes. Finer control over the metamorphosis is achieved by specifying the trajectory and transformation speed of its nodes and leaves.

Metamorphosis between arbitrary functionally-represented (FRep) objects can be described by the direct linear interpolation of the defining functions of two given shapes [PASS95], but this method can produce poor results for unaligned objects with different topologies. Turk and O'Brien [TO99] proposed a more sophisticated approach based on the interpolation of surface points (with assigned time coordinates) using radial basis functions in 4D space. This method is more applicable to unaligned surfaces with different topologies. However, for the initial implicit surfaces this method requires time-consuming surface sampling and interpolation steps. A new method of shape metamorphosis, called space-time blending, was introduced in [PPK04]. This method removes the requirement of shape alignment and is based on increasing the dimension of the object, on function-based bounded blending, and on consecutive cross-sectioning for animation.

The problem we are proposing to solve in this paper is to find a new method that is suitable for the metamorphosis of animated meshes, which is independent of the topologies of the shapes, requires a reasonable amount of processing time and is amenable to a real-time implementation. It is important to note that this problem has not been addressed before. Since successful conversion from meshes to voxels is feasible, a hybrid approach based on the combined use of animated meshes and implicit surfaces looks promising. Such an approach was applied in [KFA*10b], among other things, for the modelling of mesh interactions with viscous substances. The approach presented in that paper involved embedding an implicit convolution surface inside the animated mesh and synchronising the motions of its skeleton so that the embedded convolution surface stayed inside the polyhedral mesh during the animation process. This embedded convolution surface was subsequently blended, when required, with other external implicit surfaces that represented the environment of the mesh object (for instance, in the interaction between a mesh object and a liquid substance).

3. Method Outline

With our technique we use the concept of implicit stand-ins introduced in [KFA*10b]. For each mesh we create its approximation by an implicit surface based on skeleton information. Later in the text we refer to this approximation as the functional approximation. The metamorphosis between two meshes is performed not on those meshes directly, but on their corresponding functional approximations. The metamorphosis process consists of three distinct transitional stages:

1. a smooth transition from the animated source mesh to its functional approximation (see fig. 7a);
2. a continuous transition from the functional approximation of the source mesh to the functional approximation of the destination mesh (see intermediate frames in fig. 11);
3. a smooth transition from the resulting functional approximation to the animated destination mesh (see fig. 7b).

In this section we consider all these stages in detail.

3.1. The approximation of the animated mesh by an implicit surface

Given a mesh with skinning information and a skeletal animation, the functional approximation of the animated mesh can be derived from the skeletal implicit surface, whose skeleton information depends on the rigging-skeleton of the animated mesh. By "skeletal implicit surfaces" we mean the implicit surfaces that are derived using the set of skeletal line segments.

The correspondence between the line segments of the implicit surface skeleton and the joints of the rigging-skeleton of the animated mesh is established only once, using a single pose of the mesh model. Usually this is achieved using a rest pose or a T-pose of the mesh as it is the most natural way for the artist to depict the shape of an object (see fig. 2). However, any other pose of the mesh model can also be used, such as the initial pose of the animation sequence. As was shown in [KFA*10b], the correspondence between the skeleton of the mesh and the skeleton of the implicit surface does not have to be one-to-one, as some bones from the mesh rigging skeleton can be omitted and some additional segments can be added to the implicit surface skeleton depending on the artist's needs.

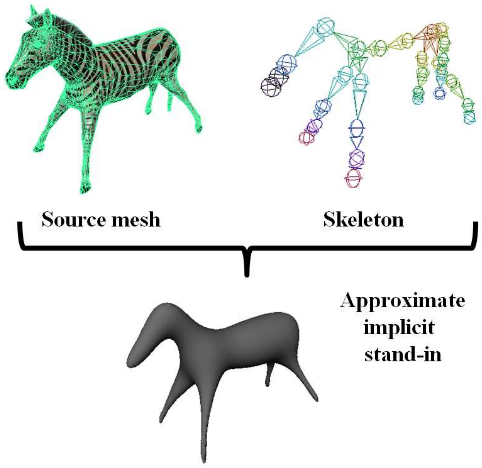


Figure 2: *Initial approximation*

The most appropriate skeletal implicit surface that can be used as a functional approximation is the convolution surface [Blo97]. In this work we use convolution surfaces that are based on linear segments with a Cauchy kernel [MS98]. One of the obvious advantages of a convolution surface is the smooth transition between its parts that are defined by the skeletal elements. The convolution surface follows the positions of its skeletal elements in a natural way that is useful for skeletal animation (see fig. 3).

An alternative to the skeletal implicit surface that could be used as a functional approximation is a set-theoretic union

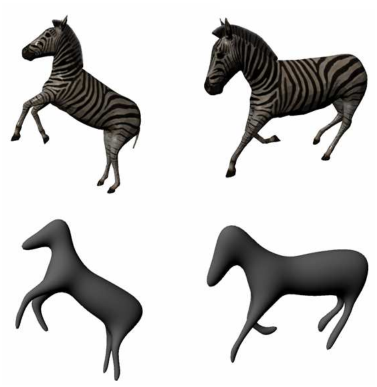


Figure 3: *Synchronisation of the animated mesh and the functional approximation.*

Figure 4: *Different functional approximations. Left to right: Initial model, convolution surface, union of canal surfaces.*

of canal surfaces. In this case, each segment of the skeleton would define a linear canal surface or a capsule. The implicit formulation for the linear canal surface can be found, for example, in [PS97]. The resulting implicit surface could be obtained by the set-theoretic union of these surfaces (see fig. 4). Compared to a convolution surface the defining function for a canal surface is easier to evaluate and because of its field properties it becomes easier to perform additional operations on such a surface, such as blending. The main disadvantage of this type of surface is the presence of visible sharp edges between its segments because of the set-theoretic operations that can later re-appear during the metamorphosis stage (see fig. 5). These edges can be avoided by using the blending versions of the set-theoretic operations (see [PASS95]), but in this case the defining function becomes more computationally expensive.

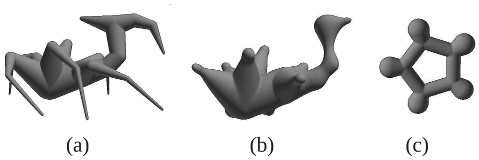


Figure 5: *Metamorphosis over the skeletal implicit surfaces defined as a set-theoretic union of canal surfaces: a) c) Functional approximation of initial models, b) Transitional stage of the metamorphosis between the shapes shown in (a) and (c)*

.

It is worth pointing out that because of the different field properties of canal and convolution surfaces, the metamorphosis between skeletal implicit surfaces of different types leads to unpredictable results. Therefore for a smoother metamorphosis transition we recommend that both functional approximations are defined by the same type of surface (i.e., that either both are defined by convolution surfaces or both by canal surfaces).

One useful extension which can help us improve the quality of the resulting animation sequence is that of automatic skeleton and motion matching. This improvement can potentially lead to a more natural and believable transition between the motions of both objects.

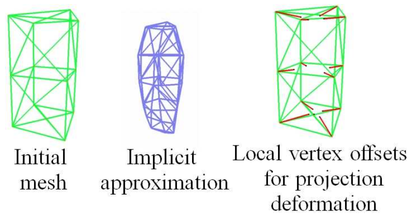


Figure 6: *Alignment deformation*

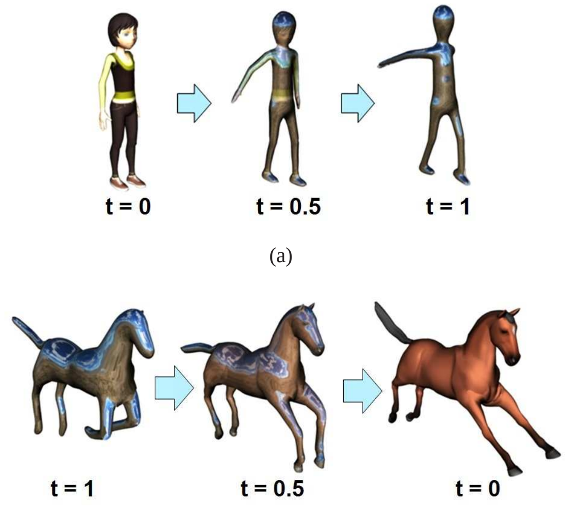
3.2. The transition between the animated mesh and its functional approximation

At the first stage of the metamorphosis between the two animated meshes we perform a transition between each animated mesh and its corresponding functional approximation. In our technique we use a projection of the vertices of the mesh onto the implicit surface approximation for each frame of this transitional stage. In a sense this is similar to a problem described in [LAG01], where coarse mesh is deformed and smoothed using an underlying set of implicit primitives.

This initial projection deformation is evaluated at the initialisation step. Each vertex of the mesh is projected onto the resulting convolution surface by using a gradient descent method (see fig. 6). After this projection step, each vertex is assigned an offset to its position on the convolution surface as well as to its resulting normal vector. This deformation allows us to align the mesh with the functional approximation, while applying the animation of the original mesh (see fig. 7):

$$\mathbf{p}'_i(t) = \sum_{j=1}^{N} w_i^j \cdot \mathbf{M}_j(t) \cdot (f_\alpha(\mathbf{p}_i, \mathbf{p}_i + \mathbf{dp}_i, t)) \qquad (1)$$

$$\mathbf{n}'_i(t) = \sum_{j=1}^{N} w_i^j \cdot \mathbf{M}_j^{-T}(t) \cdot (g_\alpha(\mathbf{n}_i, \tilde{\mathbf{n}}_i, t)) \qquad (2)$$



(a)

(b)

Figure 7: *Transition (a) between the animated mesh and its functional approximation (b) between the functional approximation and the original animated mesh.*

where w_i^j is the weight of the j-th joint of the skeleton deformation, $\mathbf{M}_j(t)$ is the transformation of the j-th joint at time instance t, f_α is the interpolation function used to perform a smooth transition from the initial point $\mathbf{p}_i$ to the deformed point $\mathbf{p}_i + \mathbf{dp}_i$ over time, $\mathbf{n}_i$ is the normal associated with the i-th vertex, $\tilde{\mathbf{n}}_i$ is the normal at the surface of the functional approximation at point $\mathbf{p}_i + \mathbf{dp}_i$ and g_α is the interpolation function for smooth normal transitions. In the simplest form both interpolation functions can be linear interpolations between the points/normals with respect to the parameter t.

On the transitional stage between the functional approximation and the animated mesh we utilise the same procedure with an inverse projection deformation (see fig. 7b).

Equations 1 and 2 are essentially a superposition of skinning deformation (weighted matrix transformation terms) and shape blending (arguments $\mathbf{f_a}$ and $\mathbf{g_a}$). The mesh is deformed over time and is gradually aligned with its functional approximation by smoothly offsetting vertices moved via the skinning deformation. At the end of the transition process the vertices of the source mesh are all located on the surface of the functional approximation. This results in both meshes being visually approximately equivalent.

In certain situations the transition process between the initial mesh and its functional approximation may not produce fine results. In this case some input from the artist might be needed to control the transition process. For a mesh with a very low number of polygons, the projection deformation alone might not be enough to avoid visual discontinuities during the transition. This is due to the fact that such a mesh cannot fully enclose the functional approximation after the applied deformation, if the sizes of the polygons are significantly bigger than those of the functional approximation. In case when subdivision of the animated mesh is not desirable, improvement of the transition process can be achieved by adding an alpha blending between the animated mesh and the functional approximation. Alternatively, this issue can be resolved by the artist. Also artistic control might be needed for a mesh with a very rough functional approximation, for example in the case of a mesh with large changes in curvature on the surface and small changes in the curvature in the functional approximation. This situation may occur when several vertices are projected to the same position on the implicit surface. In this case we need to control the direction of the vertex projection.

As was mentioned earlier, it is important to provide additional freedom for the artist to control the metamorphosis sequence. Therefore we introduce complementary "weight-maps", that define the order in which parts of the mesh have to be projected or un-projected. This allows the user to deform the meshes in a non-uniform way. For instance, fig. 8 shows that the head of the character can be "inflated" or "deflated" separately from other parts of its body. The "weight-maps" controlling the order of the projection deformation can be defined as per-vertex attributes or can be encoded in the alpha channel of the texture, thus giving an artist optional high-level control over the process.

3.3. The implicit metamorphosis stage

When the positions and normals of all the vertices are aligned with the convolution surfaces, then we switch from the polygonal object to the functional object (i.e. to its functional approximation). We can use different methods to generate the intermediate shapes based on the functional approximations of the source and destination animated meshes. These methods may include straightforward metamorphosis over two implicit surfaces, space-time blending or a complex user-controlled transition that employs the skeletons used to define the functional approximations. Here we describe a number of working options in order to provide the animator with a set of techniques that best suit specific needs.

3.3.1. Interpolation metamorphosis

Given two implicit objects represented by the functions $f_1(x,y,z) = 0$ and $f_2(x,y,z) = 0$, the metamorphosis between these two objects can be defined by the function:

$$\begin{aligned} &m_I(x,y,z,t) = (1 - w(t)) * f_1(x,y,z) + w(t) * f_2(x,y,z) \\ &w(t) \in [0;1], w(0) = 0, w(1) = 1; \end{aligned} \tag{3}$$

where t is time parameter which equals 0 at the first frame of the metamorphosis stage, equals 1 at the last frame of the metamorphosis stage and $w(t)$ is the interpolating function.

This type of metamorphosis can be evaluated very efficiently. However, it depends heavily on the field properties of the functions defining the source and destination objects. Additionally it provides the users with very limited control over the metamorphosis process [LV98].

3.3.2. Space-time blending

In space-time blending introduced in [PPK04], we apply a blending union operation between two generalised half-cylinders obtained from the source and target objects. These generalised cylinders are defined in R^{n+1}, where n is the dimension of the source and target objects. The intermediate shapes between the source and target objects is obtained by calculating the cross-section of the result of a bounded blending operation. Metamorphosis using space-time blending can be defined as follows:

$$\begin{aligned} g_1(x,y,z,t) &= f_1(x,y,z) \wedge (-t) \\ g_2(x,y,z,t) &= f_2(x,y,z) \wedge (t-1) \\ m_s(x,y,z,t) &= g_1(x,y,z,t) \vee g_2(x,y,z,t) \\ &\quad + disp_{bb}(g_1, g_2, t \wedge (1-t)) \end{aligned} \tag{4}$$

where g_1 is a generalised cylinder for f_1, g_2 is a generalised cylinder for f_2, $disp_{bb}$ is a displacement function for the bounded blending and $\vee$ and $\wedge$ denote R-functions for the set-theoretic union and intersection respectively. The displacement function $disp_{bb}$ can be defined as

$$\begin{aligned} {r_1}^2(f_1, f_2) &= \left(\frac{f_1}{a_1}\right)^2 + \left(\frac{f_2}{a_2}\right)^2 \\ {r_2}^2(f_3) &= \begin{cases} \left(\dfrac{f_3}{a_3}\right)^2, f_3 > 0 \\ 0, f_3 \leq 0 \end{cases} \\ {disp_{bb}}^2(f_1, f_2, f_3) &= \begin{cases} \dfrac{r_1^2(f_1, f_2)}{r_1^2(f_1, f_2) + r_2^2(f_3)}, r_2(f_3) > 0 \\ 1, r_2(f_3) = 0 \end{cases} \end{aligned}$$

Here parameters a_1, a_2 and a_3 are the blend parameters controlling the symmetry and influence of the field of the bounding function f_3.

In [PKP10] additional controls for this type of metamorphosis were presented. Overall, this method provides better results compared to linear metamorphosis and some aspects of it can be controlled by the artist. However, this method is more computationally expensive than linear metamorphosis and is more applicable to offline computation rather than to real-time applications.

3.3.3. User-controlled transition and deformations

For artists it is important to have more control over the metamorphosis process than what is available with the automatic methods described above. In this case additional control over the parameters of the functional approximation is needed. Such control can include the parametrisation of the radii of the canal surfaces or the parameters for the convolution surface. It is important to note that these parameters can be determined automatically or be set by the artist.

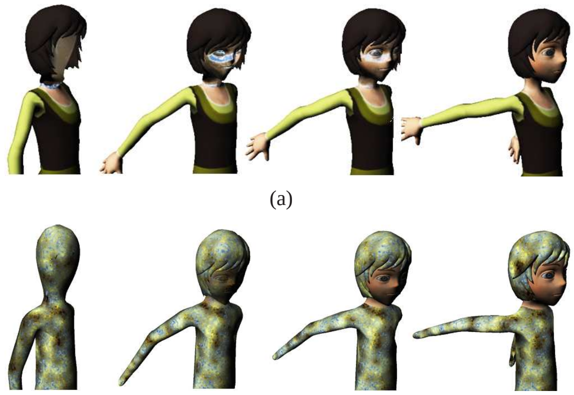
For example, at the centre of fig. 11 two possible variations are shown. In this example interpolation metamorphosis was used, however the parameters of the functional approximations are time-dependent. Control on the thickness of the shape of the functional approximation combined with the freedom to manipulate the defining skeleton provides better artistic control of the intermediate implicit surfaces. Variation of the intermediate metamorphosis transitions, shown in figures 11(a) and (b), was achieved through the application of different animations to the skeletons defining the underlying functional approximations. Animation shown in fig. 11(a) was produced using a simple metamorphosis technique performing interpolation of the two convolution surfaces with the original animations applied to the source and destination meshes. Figure 11(b) shows an animation where two skeletons were aligned in the middle of the transition process and thickness of the shape around the source skeleton was gradually decreased resulting in a smooth animation of the "collapsing" source shape and the "growing" destination shape. The resulting convolution surfaces and their topological changes are handled automatically. Manipulation of the skeleton provides the artist with a powerful tool allowing him to have additional control over the transitional stages of the metamorphosis process. Better visual results can be achieved when a correspondence between the two skeletons can be established (see fig. 11b). This can be achieved using the technique described in [BP07]. Another detailed example of user-controlled metamorphosis is described in section 4.

Additional user control over the shape of the intermediate surfaces is obtained by performing additional deformations. The nature of the resulting implicit surface after the metamorphosis process depends on the nature of the implicit surfaces that define the functional approximation. For example, if both functional approximations are convolution surfaces, the resulting implicit surface is most likely to have a blobby appearance. Modifications of the resulting shape by the artist may be required in order to fine-tune the behaviour of the transitional implicit surface. Such changes may include additional operations, such as warping deformations or solid noise applied to the transitional model.

3.3.4. Recommendations on metamorphosis

There is no single right way to define the metamorphosis between two animated meshes. Hence, we prefer to provide the user with additional options in order to let him choose the one that gives best results depending on the artistic vision and effect being created.

Interpolation metamorphosis requires no user intervention and is very simple to use. It allows the user to see intermediate results quickly without any effort. On the other hand its



(a)

(b)

Figure 8: *The application of "weight-maps" for non-uniform transitions between the meshes and their functional approximations (a) In the transitional stage the vertices for the head of the model are projected after all other vertices (b) In the transitional stage the vertices for the head of the model are projected before all other vertices.*

limited control would not suit the user willing to direct the transitional stage in a specific way.

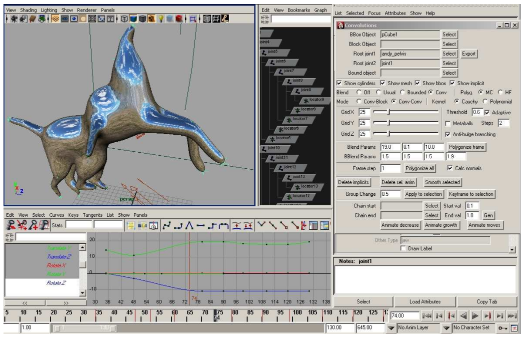
Space-time blending allows us to get more interesting results compared to simple metamorphosis. In the simplest case it hardly requires any user input but its main advantage is better control over the intermediate transition process through the use of additional parameters. These parameters allow the user to control the added volume and to direct the motion of the intermediate shape. Motions of the source and destination meshes affect the behaviour of the shape during the transition process. Additionally, improved space-time blending allows the user to get smoother transitions.

User-controlled transitions provide the animator with the most flexible way of controlling the metamorphosis process. Any change of the underlying lower-dimensional skeleton automatically affects the intermediate shape. Skeletons allow the user to take full control over the transition process. This may be a more labour intensive process compared to previous two approaches but it is the best method in cases when the user wants to achieve very specific transition behaviours according to his vision.

4. Implementation and Results

The technique proposed in this paper was implemented as a plug-in for Maya$^{\text{TM}}$. The interface allows the user to work with our plug-in in a easy and convenient way (see fig. 9), providing him/her with a near real-time evaluation of the results.

At the initial stage the user provides the source and the destination meshes together with their rigging-skeletons. The initial functional approximation, for each mesh, is built at this stage and the user can modify the parameters of the functional approximation in order to better fit the functional model to the mesh.

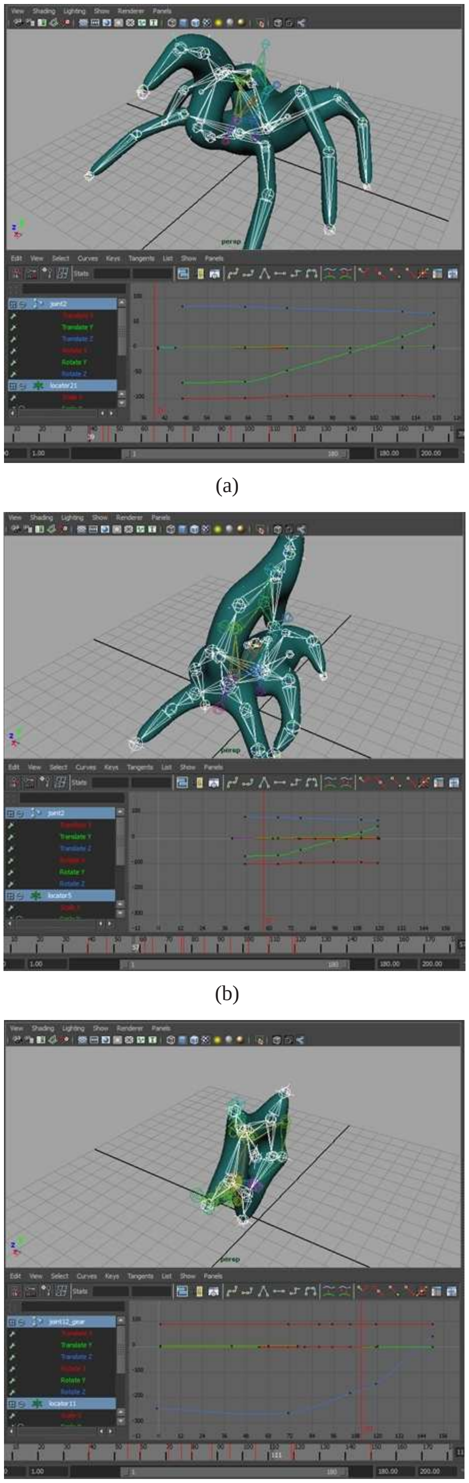


BBox Object: pCube1
Block Object:
Root joint1: andy_pelvis
Root joint2: joint1
Bound object:
☑ Show cylinders ☑ Show mesh ☑ Show bbox ☑ Show implicit
Blend ○ Off ○ Usual ○ Bounded ● Conv | Polyg. ● MC ○ HF
Mode ○ Conv-Block ● Conv-Conv | Kernel ● Cauchy ○ Polynomial
Grid X: 25
Grid Y: 25
Grid Z: 25
Threshold: 0.6 ☑ Adaptive
☐ Metaballs Steps: 2
☑ Anti-bulge branching
Blend Params: 19.0 0.1 10.0
BBlend Params: 1.5 1.5 1.5 1.9
Frame step: 1
☑ Calc normals
Group Change: 0.5
Chain start: Start val: 0.1
Chain end: End val: 1.0
Other Type: jaw
☐ Draw Label
Notes: joint1

Figure 9: *Screenshot of the working environment*

On each frame of the animation, we build a functional approximation and the resulting metamorphosis model as was described in section 3. Our plug-in allows the animator to produce lower-resolution models in near-real-time which in turn allows the animator to modify the parameters of the functional approximation in order to better control the shape of the resulting metamorphosis. The final sequence is produced using a higher resolution model.

We present an example of user-controlled metamorphosis in figure 10 corresponding to the animation shown in figure 1. Figure 10a shows working environment in Maya$^{\text{TM}}$. Joints marked in white control the skeleton of the source functional approximation. Locators attached to each joint control the radius of the shape around the joint. This approach provides an artist with a new powerful tool in a familiar environment. All the conventional animation tools available in Maya can be employed by the user in order to define the precise animation of the intermediate transition. In this example the user aligns the skeletons of the source and destination meshes over time. These skeletons along with locators controlling the radii of the shapes provide full control over the shape of the functional approximations, as was mentioned earlier. Since all controls can be animated in an arbitrary way, the user is given full artistic freedom in the definition of the actual transition. For instance, fig. 10b shows a possible transition from a monster to a gear, where the head of the monster is bent backwards while its limbs are collapsing. Additional control over the intermediate shape is provided through the use of blending (see section 3.3.1), allowing the user to adjust the contribution of each shape. Figure 10c shows the final functional approximation of the gear and its skeleton aligned with the skeleton of the monster. After this stage we can perform a smooth transition to the destination mesh controlled by the same skeleton.



(a)

(b)

(c)

Figure 10: *The metamorphosis animation setup in Maya^{TM}: (a) setting up the animation of the source functional approximation (b) user-controlled intermediate result of the blending of the two functional approximations (c) tweaking the animation of the destination functional approximation.*

Our technique can operate in a fully automatic mode. However, additional user control is often required to make the results appear more convincing. Through the use of additional user-controls one may obtain smooth transitions between models with similar skeletons (see figs. 11 and 12).

The animation sequence can be exported to a stand-alone application, where it can be re-played in real-time on the GPU. For each frame we copy the animated skeleton and the convolution parameters to the GPU and perform a projection deformation and an iso-surface extraction on the GPU. The projection deformation is applied in a vertex shader combined with a skinning deformation. The polygonisation of the metamorphosing shapes is performed using the NVIDIA CUDA SDK as described in [KFA*10a] and we have obtained similar near-real time performance in the current work.

5. Conclusions and future research

In this paper we have addressed a new problem of metamorphosis between two animated meshes, which are driven by rigging skeletons and have arbitrary non-related topologies (i.e. different genuses and number of components).

This novel approach is based on the observation that metamorphosis between implicit surfaces of differing topologies can be achieved automatically without any user intervention. The proposed hybrid metamorphosis process collapses the source mesh to an embedded approximating implicit surface, then applies several types of metamorphosis operations to it and then the obtained target implicit surface is projected to the target mesh.

An additional complication of using this type of metamorphosis technique is that of the smooth texture metamorphosis that needs to happen in parallel with the shape metamorphosis. Thus as well as metamorphosing the geometry and topology of the two key-frame mesh objects one must find a way of metamorphosing the colour of these objects so that their in-between shapes have sensible appearance. The texturing of implicit surfaces and texture metamorphosis are long-standing research problems that we are currently still working on. We intend to publish the results of our experiments in a forthcoming paper.

A CUDA implementation of our technique shows that we can achieve real-time rendering of the intermediate shapes.

However our technique has a number of limitations caused by the nature of the scalar field approximations used. One of the more serious limitations is the inability to fully control the shape of the object in the transitional stage of the metamorphosis process. Due to the nature of the functional approximation of the convolution surfaces, the object resulting from the metamorphosis operation looks to some extent blobby, which may be undesirable. To remedy this we have applied additional shape deformations to the resulting object, however these deformations are not always easy to evaluate and to control.

Future research will include a better approximation of the meshes by more sophisticated types of implicit surfaces and a more precise control of the implicit surface transformations.

References

[Blo97] BLOOMENTHAL J.: *Introduction to Implicit Surfaces, First Edition (The Morgan Kaufmann Series in Computer Graphics)*. Morgan Kaufmann, August 1997. 2, 3

[BP07] BARAN I., POPOVIĆ J.: Automatic rigging and animation of 3d characters. *ACM Trans. Graph. 26*, 3 (2007), 72–80. 6

[COSL98] COHEN-OR D., SOLOMOVIC A., LEVIN D.: Three-dimensional distance field metamorphosis. *ACM Trans. Graph. 17* (April 1998), 116–141. 2

[DG96] DECARLO D., GALLIER J.: Topological evolution of surfaces. In *Proceedings of the conference on Graphics interface '96* (Toronto, Ont., Canada, Canada, 1996), Canadian Information Processing Society, pp. 194–203. 2

[GLA00] GALIN E., LECLERCQ A., AKKOUCHE S.: Morphing the blobtree. *Computer Graphics Forum 19*, 4 (2000), 257–270. 2

[Hug92] HUGHES J. F.: Scheduled fourier volume morphing. *SIGGRAPH Comput. Graph. 26* (July 1992), 43–46. 2

[KFA*10a] KRAVTSOV D., FRYAZINOV O., ADZHIEV V., PASKO A., COMNINOS P.:. In *GPU Pro: Advanced Rendering Techniques*, Engel W., (Ed.). AK Peters Ltd., 2010, ch. Polygonal-Functional Hybrids for Computer Animation and Games, pp. 87–114. 8

[KFA*10b] KRAVTSOV D., FRYAZINOV O., ADZHIEV V., PASKO A., COMNINOS P.: Embedded implicit stand-ins for animated meshes: a case of hybrid modelling. *Comput. Graph. Forum 29*, 1 (2010), 128–140. 3

[LAG01] LECLERQ A., AKKOUCHE S., GALIN E.: Mixing Triangle Meshes and Implicit Surfaces in Character Animation. *Eurographics Workshop on Computer Animation and Simulation* (September 2001). 4

[LV98] LAZARUS F., VERROUST A.: Three-dimensional metamorphosis: a survey. *The Visual Computer 14*, 8/9 (1998), 373–389. 2, 6

[MS98] MCCORMACK J., SHERSTYUK A.: Creating and rendering convolution surfaces. *Comput. Graph. Forum 17*, 2 (1998), 113–120. 3

[PASS95] PASKO A., ADZHIEV V., SOURIN A., SAVCHENKO V.: Function representation in geometric modeling: Concepts, implementation and applications. *The Visual Computer*, 11 (1995), 429–446. 3, 4

[PKP10] PASKO G., KRAVTSOV D., PASKO A.: Real-time space-time blending with improved user control. In *Proceedings of the Third international conference on Motion in games* (Berlin, Heidelberg, 2010), MIG'10, Springer-Verlag, pp. 146–157. 6

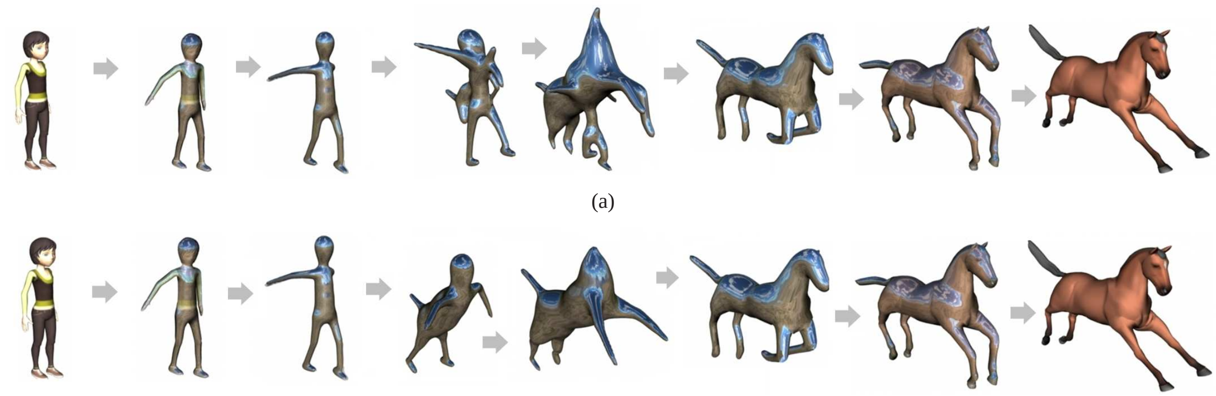
[PPK04] PASKO G., PASKO A., KUNII T.: Space-time blending. *Computer Animation and Virtual Worlds 15*, 2 (2004), 109–121. 3, 6

[PS97] PASKO A., SAVCHENKO V.: Projection operation for multidimensional geometric modeling with real functions. In *Geometric Modeling: Theory and Practice, W. Strasser, R. Klein, R. Rau (Eds.)* (1997), Springer-Verlag, Berlin/Heidelberg, pp. 197–205. 4

[TKO01] TAKAHASHI S., KOKOJIMA Y., OHBUCHI R.: Explicit control of topological transitions in morphing shapes of 3d meshes. In *Proceedings of the 9th Pacific Conference on Computer Graphics and Applications* (Washington, DC, USA, 2001), PG '01, IEEE Computer Society, pp. 70–. 2

[TO99] TURK G., O'BRIEN J. F.: Shape transformation using variational implicit functions. In *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques* (1999), pp. 335–342. 3

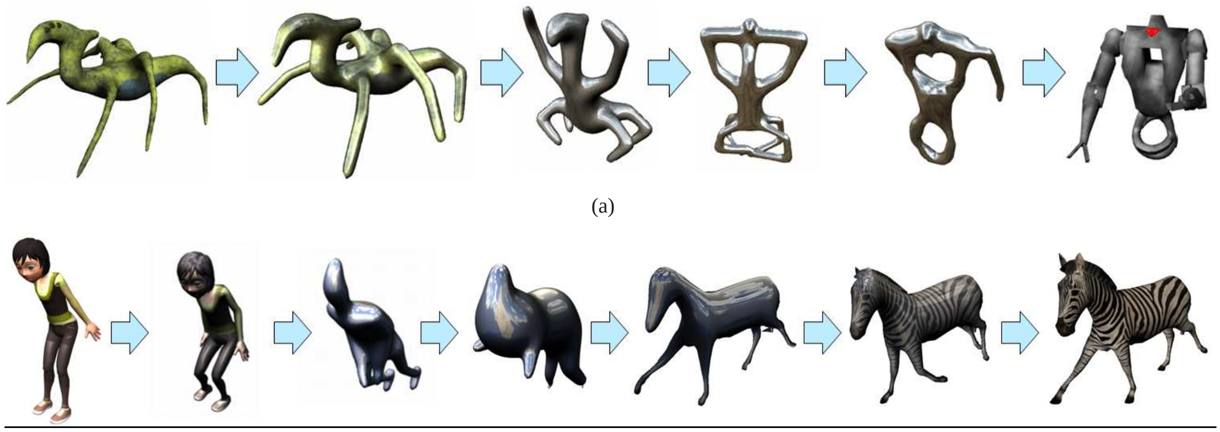
[YJ07] YANG H., JUTTLER B.: 3d shape metamorphosis based on t-spline level sets. *Vis. Comput. 23* (November 2007), 1015–1025. 2



(a)

(b)

Figure 11: *Metamorphosis animated sequence, first 3 frames represent the transitional stage from the source mesh to its functional approximation, the following 2 frames represent the metamorphosis stage and the last 3 frames represent the transitional stage between functional approximation and destination mesh: a) Automatic linear metamorphosis; b) Metamorphosis with additional user control of skeletons (performing alignment of skeleton segment positions along with gradual modification of the radii associated with each segment).*



(a)

(b)

Figure 12: *Additional examples of metamorphosis between animated meshes implemented in real-time in CUDA: a) Additional user-controlled alignment of the skeletons was performed for the transition in order to build a custom intermediate shape; b) user-defined skeleton alignment similar to 11(b) was used to match intermediate functional approximations.*