Local Metamorphosis of Functionally Defined Shapes

B. Schmitt
Digital Media Professionals
Tokyo, Japan
schmitt_benjamin@yahoo.fr

A. Pasko
Hosei University
Tokyo, Japan
pasko@k.hosei.ac.jp

P.-A. Fayolle
University of Aizu
AizuWakamatsu, Japan
d8052103@u-aizu.ac.jp

Figure 1: One object is metamorphosed to another object of different topology. The most left object is the initial object and the most right the final object. The traditional global metamorphosis operation is used in the upper middle row, and the proposed local metamorphosis is used in the lower middle row.

Abstract
We propose a new operation to locally control the metamorphosis of two functionally defined shapes. To implement this operation, a set of non-overlapping space partitions is introduced, where the metamorphosis occurs locally. The sequence of local metamorphosis processes is controlled by a specific time schedule. The definitions of the partitions, of the time schedule, and finally of the local metamorphosis operation, are described in this paper. Experimental results and comparison with the traditional metamorphosis operations are also provided.

Introduction
Metamorphosis of two functionally defined shapes is considered in this paper. A metamorphosis is the transformation of an initial shape to a final shape. This animation technique is popular for polygonal objects. Nevertheless, for functionally defined shapes, few techniques have been proposed, especially for the locally controlled metamorphosis.

One of the first metamorphosis operations was proposed by Wyvill [WYV90], where initial and final shapes were defined as soft objects. Surface in-betweening is based on a weight paradigm applied to the initial shapes. The force property of each source key point is weighted. The weighting of the source keys is gradually changed from one to zero as the in-between progresses. The weighting of the force property of the destination keys changes from zero to one. This results in the interpolation between defining functions of source and destination soft objects.

In the advanced modeling technique proposed in [WEA99], still based on soft and/or blobby objects, metamorphosis is well defined by the use of a so-called warping operator [GLA00]. This operator allows fine control of the metamorphosis in some cases. In this context, initial and final shapes are defined by a constructive approach and are represented using a tree structure, so-called BlobTree. To define a metamorphosis with the proposed warping operators, one needs to define a correspondence between subtrees of the initial shape with some other subtrees of the destination shape. A subtree is a subset of the initial BlobTree, and can include several primitives (leaves of the tree) based on a distance function and a potential function, and several operations (nodes of the tree; set-theoretic, algebraic transformations etc.). This approach has the benefit to provide a fine control of the metamorphosis, and the concept of subtree provides a mean to define local metamorphosis.

However, the warping operator is applicable only to blobby objects. It can be hardly applied to other functionally defined shapes. In this paper, we intend to fill this gap and try to provide a new operation that enables the definition of local metamorphosis for functionally defined shapes. As
a framework for defining the shapes, we use the function representation model, FRep for short [PAS95]. In this model, shapes are defined by real valued function, so-called defining function. The interior of the shape is defined when this function is positive, its surface when the function is equal to zero and its exterior when it is negative. Each defining function is built using a constructive approach, resulting in a so-called constructive tree. The leaves of this tree can contain arbitrary functionally defined primitives (quadrics, blobby objects, trivariate BSpine, convolutions etc...), and its leaves can contain set-theoretic operations (union, intersection, subtraction), space transformations (rotation, point based deformations, twisting), and other types of operations (sweeping, Minkowski sum etc.). The left most and right most pictures of Fig. 1 show two objects defined by FRep models.

Given two FRep objects with the corresponding defining functions \( f_1 \) and \( f_2 \), a metamorphosis can be defined as a linear interpolation between these two real valued functions, where the time \( t \) is the parameter used for the interpolation:

\[
F = (1-t) \times f_1 + t \times f_2 \quad (1)
\]

It assumes that the shapes defined by \( f_1 \) and \( f_2 \) overlap in the space; otherwise in the resulting animation, one shape disappears and the other appears somewhere else, without actual shape transformation. This definition of the metamorphosis in the FRep model is global. Using this definition, one can not define metamorphosis locally. The upper middle row of images of Fig. 1 shows the result of such global animation. As one can see, initial shape is transformed into the final shape as a whole.

However, one may want to have a finer control of the metamorphosis process. For instance, as shown in the lower middle row of images of Fig. 1, one may want to have certain parts to disappear first (the vertical columns of the initial shape), then some parts to appear (the blended shapes inside the object), and finally to obtain the final shape. How to obtain this local control is the topic of this paper, and further explanations on the images shown in Fig. 1 will be given in the next sections.

In this paper, we define a new operation for metamorphosis that allows one to control locally its behavior. To define this operation, we need first to define a set of space partitions, where each local metamorphosis occurs and then a corresponding time schedule. With the use of these two items, we finally define the new metamorphosis operation. Comparison with the warping operator of the BlobTree and examples are given at the end of this document.

### Local metamorphosis

This section is organized as follows. First, we propose a mean to define partitions of the Euclidean space where the initial and final shapes are defined. Such partitions are called in the following metamorphosis partitions, and are denoted \( A_i \) where the lower script \( i \) indicates the partition number.

Second, we define a time schedule. We use in the following the global and local dynamic variables which can be mapped to time in order to generate animation. To control the time interval when a metamorphosis occurs inside a partition and the behavior of the metamorphosis in each partition, we assign a local dynamic variable to each partition. Finally, in the third subsection, we define the local metamorphosis operation, using the metamorphosis partitions, the global dynamic variable and the local dynamic variables.

### Partition definition

Metamorphosis partitions are defined using any real valued functions that fit to the FRep model. To define a metamorphosis partition, two different functions are needed. The first function defines the FRep model and takes values in \( \mathbb{R} \). The second function is defined depending on the first function, and takes values in the interval \([0,1]\). Figure 2 shows the initial setup for a simple example of 2D metamorphosis, where the initial geometry is a 2D block and the final a 2D ring.

Let \( P_i \) be the defining function of the \( i \)-th partition \( A_i \) in the Euclidean space. The function \( P_i \) defines two subsets of this space, one where \( P_i \) is greater or equal to zero and another where it is negative. In Fig. 2, one partition has been created, where \( P_i \) is defined as an ellipse. The subset of the Euclidean space where \( P_i \) is positive is shown in red and green (corresponding to the A and B zones).

Let \( F_{\mu_i} \) be a real valued function defined upon the defining function \( P_i \). In the example shown in Fig. 2, \( F_{\mu_i} \) is constructed to meet the following requirements. Given a user defined threshold value \( \Gamma_i \) (greater or equal to zero), the behavior of \( F_{\mu_i} \) is as follows:

\[
F_{\mu_i}(X) = \begin{cases} 
0 & \text{if} \quad P_i(X) < 0 \\
\Omega \left( \frac{P_i(X)}{\Gamma_i} \right) & \text{if} \quad 0 \leq P_i(X) \leq \Gamma_i \\
1 & \text{if} \quad P_i(X) \geq \Gamma_i
\end{cases} \quad (2)
\]
with the function $\Omega$ being one of the functions below:

$$\Omega(X) = \begin{cases} 
X, & 3X^2 - 2X^3 \\
6X^5 - 15X^4 + 10X^3, & \ldots
\end{cases} \quad (3)$$

Loosely speaking, $F_p$ is a simple step function that maps $P_i$ to the interval $[0,1]$, and $\Omega$ allows one to control the continuity of this mapping. The plots of the three different functions $\Omega$ are shown in Fig. 2(bottom). Other definitions for such a function can be found in the literature, see, for instance, [LI04]. We chose this one for its simplicity.

Considering Fig. 2, the following zones are defined:

1. $A$ (Red): Inside the partition, $F_{pi} = 1$, $P_i > \Gamma_i$ and $P_i \geq 0$
2. $B$ (Green): Border of the partition. $F_{pi}$ takes value in $[0,1]$, $P_i < \Gamma_i$ and $P_i \geq 0$
3. $C$ (White): Outside the partition, $F_{pi} = 0$, $P_i < \Gamma_i$ and $P_i < 0$

In the general case, a set of metamorphosis partitions is defined. From the user side, he/she models a set of partitions $P_i$ using FRep and associates each partition with a threshold value $\Gamma_i$. When he/she defines $N$ partitions, an additional metamorphosis partition, called the remaining partition $A_r$, must be defined in order to cover the entire Euclidean space. The remaining partition is defined as the set-theoretic difference between the Euclidean space and the union of all the partitions defined by the $P_i$ functions. The mapping function associated to this remaining function is expressed as:

$$F_{p,R}(X) = 1 - \sum F_{pi}(X) \quad (4)$$

For this partition, and only for this special partition, the function $F_{p,R}$ may take value that are not within the interval $[0,1]$, but, as it will be shown later in this document, this property does not influence the desired result.

Once a set of metamorphosis partitions has been defined, one needs to define a time schedule to indicate when a metamorphosis starts and stops evolving inside a partition. In the following subsection, we describe how to define it.
Time schedule

The whole metamorphosis is controlled by the time parameter, a global dynamic variable $t$, defined for simplicity in the interval $[0,1]$. We define for each partition $\Lambda_i$ a local dynamic variable $t_i$. Let us assume that $N$ partitions have been defined. Then, for each partition $\Lambda_i$, each local dynamic variable $t_i$ takes value in the interval $[0,1]$ for every value of the global dynamic variable $t$ in $[T_i, T_{i+1}]$, where $T_i$ and $T_{i+1}$ are user defined boundary interval values, defined in $[0,1]$, with $T_0=0$ and $T_N=1$. They correspond to the beginning and ending values where the metamorphosis starts and finishes inside the given partition $\Lambda_i$.

Figure 3 illustrates the concept of the metamorphosis partitions and the time schedule. The initial geometry is a 2D block, which is transformed into a 2D ring. In this example, three partitions are defined (and colored for explanation purposes), a red ellipse, a green disk, and the remaining metamorphosis partition (Fig. 3, top).

The plots, (below part of Fig. 3), shows the corresponding time schedule. First, the time $t_0$ corresponding to the green partition changes linearly from zero to one; then when $t_0$ is equal to one, $t_1$ starts to grow until it reaches one; finally, once $t_0$ and $t_1$ are equal to one, the dynamic variable $t_r$ corresponding to the remaining partition starts to increase, until it reaches one. When $t_r$ is equal to one, the metamorphosis is finished.

In the previous two subsections, we explained how to define space partitions and how to define the corresponding time schedule. In the next subsection, we define the operation for the local metamorphosis of two shapes.

Local metamorphosis definition

The local metamorphosis operation is defined by a mapping function that maps the global dynamic variable $t$ of Eq. 1 to another global dynamic parameter $t'$.

Given a set of $N$ metamorphosis partitions, a time schedule with local dynamic variables $t_i$, the remaining partition $\Lambda_r$ with its own local dynamic variable $t_r$, and a global dynamic variable $t$, the mapping of $t$ can be described as follows:

$$ t'(X) = \sum_{i=0}^{N-1} t_i F_{p_i}(X) + t_r F_{p_r}(X) $$(5)

Figure 3: (top) Metamorphosis partitions used to generate the frames in Fig. 4. (bottom) The corresponding time schedule. The time $t_0$ corresponds to the red ellipse, $t_1$ to the green disk and $t_r$ to the remaining space. Using this time schedule, the part of the block that belongs to the red ellipse will first be metamorphosed, then the part in the green disk, and finally the remaining parts.
Then, the mapped global dynamic variable $t'$ is simply used instead of $t$ in the linear interpolation for the global metamorphosis operation (see Eq. 1):

$$F = (1 - t') \times f_1 + t' \times f_2$$  \hspace{1cm} (6)

The equation 5 defines the mapping of the global dynamic variable depending on the metamorphosis partitions.

Let us suppose that a single partition has been defined with two local dynamic variables. The variable $t'$ is then defined as follows depending on the local dynamic variables:

<table>
<thead>
<tr>
<th>$T_0$</th>
<th>$t_r$</th>
<th>Global dynamic variable $t'$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>0</td>
<td>$t'(X) = t_0 F_{p_0}(X)$</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>$t'(X) = F_{p_0}(X)$</td>
</tr>
<tr>
<td>1</td>
<td>$0.1$</td>
<td>$t'(X) = F_{p_0}(X) + t_r F_{p_B}(X)$</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>$t'(X) = F_{p_0}(X) + F_{p_B}(X)$</td>
</tr>
</tbody>
</table>

The metamorphosis behaves as follows. When both local dynamic variables are equal to zero, the global dynamic variable is also equal to zero, i.e., it corresponds to the very beginning of the metamorphosis where the initial shape is unchanged. Then, $t_0$ linearly grows to one. For points that belong to the C zone of the partition $A_0$ (i.e., outside the metamorphosis partition, with $F_{p_0}=0$), the mapped dynamic variable is equal to zero. The initial shape remains unchanged. For points that belong to the A zone of $A_0$ (i.e., $F_{p_0}$ in $[0,1]$), the mapped dynamic variable is different from zero, and the metamorphosis in this zone occurs until the final destination shape is obtained (at $t_0$ equal to one). For points that belong to the B zone of the partition defined by $F_{p_0}$, some smooth metamorphosis occur corresponding to the transition area between the parts in the A zone and the C zone. Note that for points in the B zone, the metamorphosis is not completed during this time interval. It will be completed later (in this example, it will be during the time schedule corresponding to the remaining partition).

Once $t_0$ is equal to 1, the metamorphosis is completed in the A zone defined by the partition $A_0$. Recalling that in this example we consider only a single partition, then the local dynamic variable corresponding to the remaining partition $t_r$ starts to increase from zero to one, until the final shape is obtained.

The remaining partition is then defined as (see Eq. 4):

$$F_{p_B}(X) = 1 - F_{p_0}(X)$$

The last equation of the table can be rewritten as:

$$t'(X) = F_{p_0}(X) + F_{p_B}(X)$$

$$t'(X) = F_{p_0}(X) + (1 - F_{p_0}(X))$$

$$t'(X) = 1$$

It means that when the dynamic variable of the remaining space is equal to one, the destination

---

**Figure 4**: Metamorphosis of a block to a ring. (top) Global metamorphosis using linear interpolation. (bottom) Local metamorphosis using the partition shown in Fig. 3.
shape is obtained (c.f. the linear interpolation of Eq. 6).

In the case that two partitions are defined (three if one includes the remaining partition), when the metamorphosis is complete, the global dynamic mapping is defined as:

\[ t'(X) = F_{p_0}(X) + F_{p_1}(X) + F_{p_R}(X) \]

Replacing \( F_{p_R} \) by its definition, we obtain:

\[ t'(X) = F_{p_0}(X) + F_{p_1}(X) + \left(1 - (F_{p_0}(X) + F_{p_1}(X))\right) \]

This last equation shows explicitly that when \( t \) is in the interval \([0,1]\), the value of \( t' \) is can be outside \([0,1]\) in the case where the partition \( A_0 \) and \( A_1 \) overlaps (i.e. \( F_{p_0} \) and \( F_{p_1} \) are equal to one for instance). This is an undesirable property, as we assume that the global dynamic variable is in the interval \([0,1]\). This property leads a requirement on the metamorphosis partitions. They should be non-overlapping. This requirement is not very strong, because partitions are modeled for metamorphosis. To model overlapping partitions is conceptually difficult to understand, as the aim of the partition is to define a local metamorphosis, i.e. when a part of the shape starts and stops to evolve. If this part belongs to two or more metamorphosis partitions, the result that one expects is not clear.

Thus, using Eq. 6, a set of non-overlapping metamorphosis partitions modeled with FRep, and a time schedule, one can define the metamorphosis of two functionally defined objects with local control. The next subsection proposes a comparison with the BlobTree warping operator, and then we present several examples.

As a last remark concerning the definition of the local metamorphosis, and especially of the time schedule, one may wish to change the linear growth of the dynamic variables (both local and global) to some more complex behavior. As the \( \Omega \) function provides certain continuity, existing animation techniques can be easily included to this definition. Indeed, in the literature, non-linear time control has been widely discussed, and can be based on a 1D mapping function, using a spline function [FD95]. Such fine control of the time schedule can be included as an additional mapping of the dynamic variables as long as the resulting mapped variable remains in its dedicated interval.

**Comparison with the BlobTree warping operators**

As mentioned above, each metamorphosis partition is defined by a real valued function and can be modeled using a constructive approach, including all the existing modeling techniques related to this field. The only requirement is that the resulting real valued defining function of the partition is at least continuous.

An advanced technique to control metamorphosis of functionally defined shapes, restricted to the subset including blobby objects, soft objects are other objects based on the composition of a distance function and a potential function, has been proposed in [GLA00] by Galin et al. This technique consists in, given the constructive trees of the initial and final shapes, adding several warping operators inside these trees. A correspondence between these new nodes is then established, either automatically or manually, which allows one to control the metamorphosis of a given subset of the initial constructive tree to another subset of the final constructive tree. Although this technique is powerful, it has some limitations, as the metamorphosis trees are dependent of the initial constructive trees. Even if so-called “ghost shapes” can be added to the BlobTree to help the definition of the subtree correspondences, the dependency between the objects to be metamorphosed remains important. For example, it would be very difficult to create the metamorphosis shown in Fig. 4 with the proposed warping operator. Indeed, the initial and final constructive trees are reduced to a single primitive. The metamorphosis operation introduced in this paper allows one to define local metamorphosis anywhere in space independently from the initial constructive trees of the objects under consideration.

**Results and examples**

In this section, we give several examples of local metamorphosis. The first example of the metamorphosis between a 2D block to a 2D ring was used above in the mathematical definition of the proposed metamorphosis operation. Two partitions are defined, an ellipse and a disk. The metamorphosis setup is shown in Fig. 3. Figure 4 shows some frames of three different animations. The top row shows a simple linear interpolation between the initial block shape and the final 2D ring (Eq. 1). The bottom row shows the resulting frames while using the local metamorphosis operation defined in this paper. The assigned color corresponds to a given partition. The threshold
values for both partitions are different. The value corresponding to the green partition is close to one, and for the red partition it is close to zero. As one can see, in the first partition, the metamorphosis is smoother than in the second one. According to the time schedule, first, the metamorphosis occurs in the red area, then in the green area, and finally the remaining space is metamorphosed.

The definition for local metamorphosis proposed in this paper provides the possibility to define metamorphosis anywhere in space and is completely independent of the constructive trees of the objects under consideration. This feature is very important to create complex metamorphosis. Figure 1 shows a more advanced example of metamorphosis between two functionally defined shapes. The left most and right most figures are respectively the initial and the final shapes. In the upper middle row, two frames of the global metamorphosis are shown. In the lower middle row, two frames of the animation using local metamorphosis are shown. As one can see, we choose to morph some parts of the object that do not correspond to a precise subtree of the initial constructive trees. One can also notice that the initial object is textured with a procedural texture [EBE03]. To create the animation, we applied the local metamorphosis definition and the dynamic variable mapping function to interpolate the texture between the initial and final shapes.

In this paper, we defined the local metamorphosis operation and considered only the geometry. But as a matter of fact, nothing prevents one from using this operation in the constructive hypervolume model [PASS02]. Indeed, in this model, attributes, such as color, temperature or any other abstract point-wise attribute, are also functionally defined. The definition of local metamorphosis for attributes is straightforward, as one need to simply replace the defining function of the geometric shapes with the real valued function that defines the attribute. The only restriction on the metamorphosis is that the initial and destination attributes should be of the same nature (i.e., a color is changed to another color, for instance).

Figure 5 shows the metamorphosis between FRep models of a tank and a plane. The upper set of frames shows the result of the global metamorphosis, while the lower set of frames illustrates the usage of the local metamorphosis. Figure 6 shows the metamorphosis partitions. The metamorphosis occurs first in the red partition, then in the blue one, then in the yellow one and finally in the remaining space.

Figure 6: Metamorphosis partitions used for the animation shown in Fig. 5.

**Conclusion and future works**

We proposed a new metamorphosis operation for the FRep framework that permits local control of the metamorphosis behavior. To define this operation, a set of non-overlapping metamorphosis partitions and corresponding time schedule are introduced. The results obtained are convincing and clearly show that local control is now possible in the animation defined as the metamorphosis of two functionally defined shapes.

Our future research plan follows two main directions. First, we would like to include the SARDF operations [FPB04] and distance based primitives to check if the distance field provides better control of the metamorphosis, and also to help the user to define the partitions. Indeed, in the FRep model, to find a proper threshold value for the partition needs several tries from the user before it can be set. As the initial shapes are defined using a constructive approach, the resulting real valued function can have radically different behavior (very high or low slopes).

The other direction is to consider not only metamorphosis defined as a linear interpolation, but a more general metamorphosis operation. In this paper, we assume that the initial and the final shapes are overlapping in space. This is a very strong requirement. The metamorphosis proposed in [PPK05] based on space time blending allows for arbitrary positions of the shapes in space without overlapping. In its current formulation, the space-time blending has only a global character. Our future work includes the definition of local control for the space-time blending.
References


Figure 5: Metamorphosis of a tank to a plane. (top) Global metamorphosis. (bottom) local metamorphosis using the partitions shown in Fig. 6.