# Locally restricted blending of Blobtrees

Erwin de Groot
University of Calgary
2500 University Dr. NW
Calgary, Alberta, T2N 1N4, Canada
erwin@erwindegroot.nl

Brian Wyvill
University of Victoria
3800 Finnerty Road
Victoria, BC, V8P 5C2, Canada
blob@cs.uvic.ca

Huub van de Wetering
Technische Universiteit Eindhoven
Den Dolech 2
PO Box 513 - 5600 MB Eindhoven, The Netherlands
h.v.d.wetering@tue.nl

## Abstract

*Blobtrees are volume representations particularly useful for models which require smooth blending. When blending is applied to two or more Blobtree models, extra volume will be created in between the two surfaces to form a smooth connection.*

*Although it is easy to apply blending, it is hard to accurately control the resulting shape. More complications arise when the blended objects have large size differences. In this case the influence of the larger objects can overwhelm the influence of the smaller objects. As a result, the shape of the smaller objects can change drastically and the connection between surfaces can appear sharp instead of smooth.*

*This paper presents a locally restricted blend method that solves the blending problem described above. The locally restricted blend locally changes the blending influence of each of the surfaces in order to control blending with the other surfaces. Unlike previous methods, this blend method works with multiple Blobtree surfaces and offers intuitive control over the resulting shape.*

**CR Categories:** I.3.5 [Computer Graphics]: Curve, surface, solid, and object representations
**Keywords:** Implicit Surfaces, Blending

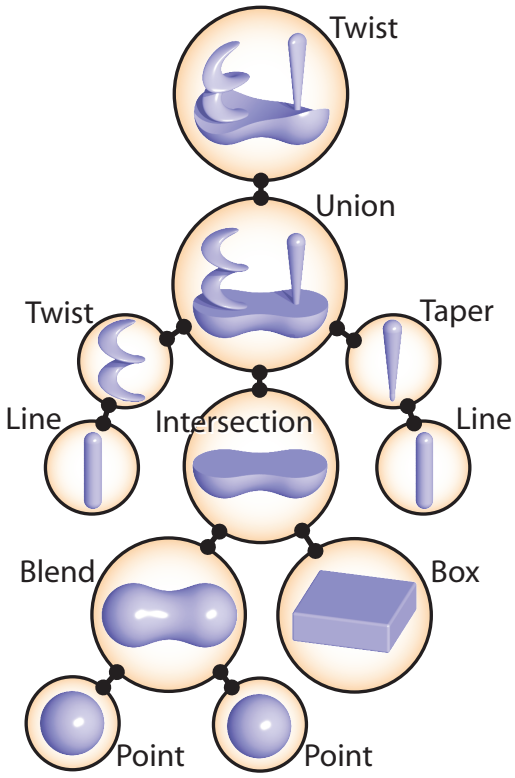


**Figure 1. Example Blobtree model**

## 1 Introduction

Blobtrees are implicit surfaces with a hierarchical structure particularly useful for modeling smooth objects of any topology. A number of Blobtree models can be blended together using simple operations like summation. In general the resulting shape of the blend is hard to control using these simple operations. More advanced blending techniques could be used to increase controllability, but most of these techniques can not easily be applied to Blobtree mod-

els. Existing blending techniques suitable for Blobtrees either have severe limitations or are not intuitive in their use. This paper describes the locally restricted blending method which can be applied to any set of Blobtree models and offers intuitive control of the resulting shape.

### 1.1 Implicit Surfaces and Blobtrees

Implicit surfaces are volume representations where the surface of the volume is defined by an iso-value of a real function. There are several variations of implicit surfaces [5] like Blobby Molecules [4], Soft Objects [28], Blobtrees [27] and FRep [20, 19], but in general implicit surfaces can be represented by a function $f : \mathbb{R}^3 \rightarrow \mathbb{R}$ which takes a point $p \in \mathbb{R}^3$ and returns the field value $f(p)$ of that point. A surface $S$ of a function $f$ can be defined as the collection of points with a certain field value $c \in \mathbb{R}$: $S = \{p \in \mathbb{R}^3 | f(p) = c\}$ ($c$ is called the iso-value of $S$). The volume inside surface $S$ is the collection of points for which the field value is greater than $c$ (or smaller than $c$ in some literature).

A Blobtree is an implicit surface organized in a tree structure built from skeletal primitives (leaves of the tree) and operations (inner nodes of the tree; see figure 1). The skeletal primitives are defined by simple skeletons like a point or a line. The field value in point $p$ of such a primitive $A$ is calculated by applying a field function $g : \mathbb{R} \rightarrow \mathbb{R}$ to the distance $d_A(p)$ between $p$ and the skeleton. An example of such a field function would be the Wyvill field function [28] (figure 2). Given a skeleton and its distance function $d_A(p)$ the field values for this primitive $A$ can be calculated as follows:

$$f_A(p) = g(d_A(p)). \tag{1}$$

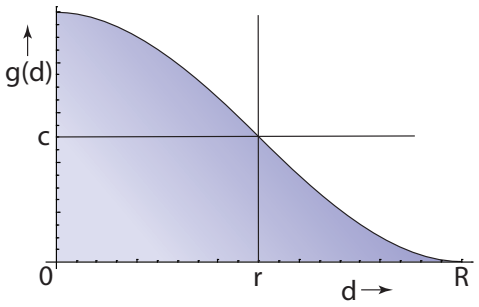


**Figure 2. The Wyvill Field function**

For points $p$ with distance $d_A(p)$ between 0 and $r$ (see figure 2) the field values will be greater than the iso-value $c$ and therefore these points will be part of the volume. The points $p$ with distance $d_A(p)$ between $r$ and $R$ are not part of the volume, but can still influence nodes higher in the Blobtree (blend operations in particular). The collection of points $p$ for which $r < d_A(p) < R$ (or $0 < f_A(p) < c$) is called the blending range of $A$ (or 'added material blend' in some literature).

The inner nodes of a Blobtree are operations. The field values of an operation $B$ are computed using the nodes in the tree directly below $B$: the child nodes of $B$. The child nodes of an operation can be both primitives or operations themselves. If the collection of child nodes of $B$ is denoted as $N_B$ than the function of $B$ generally can be written as:

$$f_B(p) = \otimes\{A \in N_B \mid f_A(p)\}, \qquad \otimes : \mathbb{R}^{|N_B|} \rightarrow \mathbb{R}. \tag{2}$$

In this equation $\otimes$ is the operation that is applied to the field values returned by the child nodes. For example, the function of a regular blend (summation blend) operation $B$ that has two child nodes $A_0$ and $A_1$ will look as follows:

$$f_B(p) = f_{A_0}(p) + f_{A_1}(p). \tag{3}$$

### 1.2 Blobtree blending

When the blending ranges of the child nodes of a blend operation overlap, extra volume called blending volume will appear that makes the connection between the child nodes smooth. When more than two Blobtree models are blended together, extra volume may appear between each pair of Blobtrees. The extra volume between such a pair is called partial blending volume and all partial blending volumes together form the final blending volume. Although blending implicit surfaces is easy, controlling the blending volume is not and in many cases the result will not have the desired shape.

One of the problems with blending implicit surfaces is unwanted bulging [7, 6]. This occurs when the blending volume grows too large. Image 5a of figure 3 shows the summation blend of three line primitives and image 5b shows the same blend with a shape cut out to see part of the inside. The summation blend results in a bulge all around the intersection of the primitives, while the desired shape in this case would only have extra volume in between the primitives to eliminate the sharp creases that can be seen in image 1a. In general unwanted bulging can be reduced by reducing the blending range and thus the blending volume in general (images 2a and 3a).

Another common problem with blending implicit surfaces is the influence problem. This occurs when the child nodes of the blend operation have relatively large size differences (as explained in [29]). When the size difference is caused by scaling certain nodes, the blending ranges will also be different in size. In this case the influence of the larger child nodes can cause drastic shape changes to the

smaller child nodes. Row 5 of figure 4 shows the summation blend applied to two point primitives with size ratios 16, 8 and 4 respectively. The summation blend results in a drastic size change of one of the primitives.

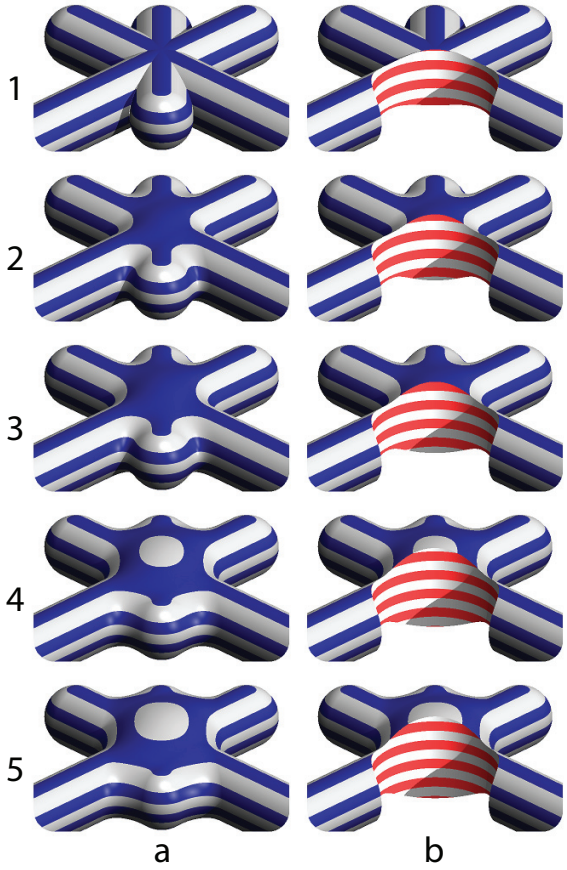


**Figure 3. Unwanted bulging; 1: Union, 2: Restricted blend, 3: Ricci blend (K=4), 4: Ricci blend (K=2), 5: Summation blend.**

There are several blend methods for Blobtrees. The most commonly used is the summation blend which is simply a summation of the field values of the child nodes:

$$f_B(p) = \sum_{A \in N_B} f_A(p). \tag{4}$$

The summation blend does not have any parameters to control the shape of the result and does not offer a way to reduce unwanted bulging (row 5 of figure 3) or unwanted shape changes (row 5 of figure 4). Another commonly used blend is the Ricci blend [23]:

$$f_{R_K}(p) = \sqrt[K]{\sum_{A \in N_{R_K}} (f_A(p)^K)}. \tag{5}$$

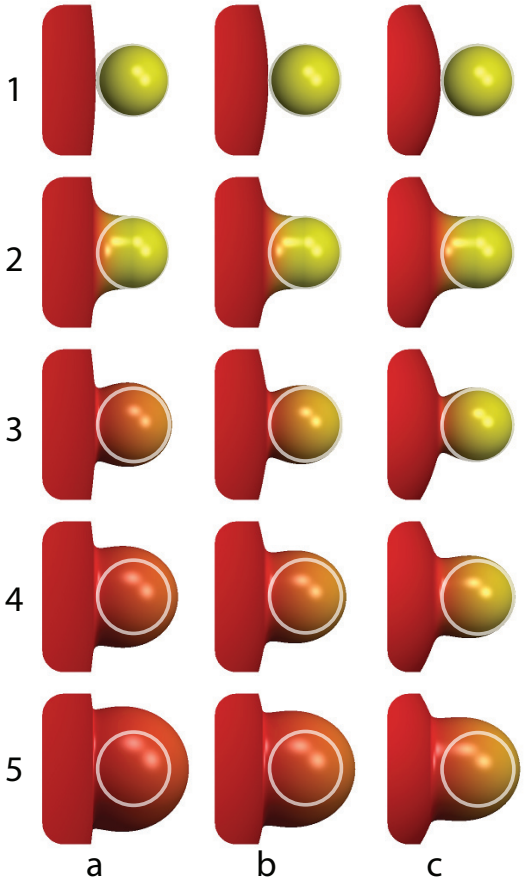


**Figure 4. The influence problem; 1: Union, 2: Restricted blend, 3: Ricci blend (K=4), 4: Ricci blend (K=2), 5: Summation blend.**

The difference between the Ricci blend and the summation blend is the parameter $K$. This parameter controls the amount of blending volume between the child nodes. When $K = 1$ the Ricci blend has the same result as the summation blend. When $K = \infty$ the Ricci blend is equal to the union operation (no blending volume). When the parameter $K$ is used correctly, unwanted bulging can be reduced (rows 3 and 4 of figure 3). The Ricci blend is less effective against unwanted shape changes due to large size differences (rows 3 and 4 of figure 4). When the parameter $K$ is set high enough to prevent severe shape changes, this can also results in loosing the smooth transition (row 3 of figure 4).

A controllable blending method for Blobtrees should offer enough control of the resulting surface to reduce unwanted bulging and solve the influence problem. The control over the blending shape should also be provided in an intuitive and user friendly way.

The locally restricted blending method described in this paper achieves these goals by locally deforming the blend-

ing range of each object. The locations of blending range deformations are directly related to the positions of surrounding objects. This way, the interference between the deformations is kept to a minimum. Because of this, the parameters that control the amount of deformation are also independent of each other and give an intuitive control.

## 2 Previous work

Some blending operations involve changing the field function of its child nodes [16, 29, 14]. These operations offer some control over the blending volume by changing the field functions of the child nodes before blending. Unfortunately this is only possible if the child nodes are primitives and it is not possible to change the partial blending volumes individually in case more than two child nodes are used (see section 3 for an example of this problem). Unwanted bulging can be reduced using this type of operation and the influence problem can be solved when only two primitives are blended. When more primitives are blended together, adjusting on of the partial blending volumes could change the other partial blending volumes introducing new problems.

Before a set of objects are blended together, a function can be applied to each of the objects to adjust the blending range as described in [2]. Unfortunately these functions reduce the range of field drastically, which reduces the usability of further blend operations (and other operations that use the blending range). Adjusting partial blending volumes is also not possible.

Incompatible fields (e.g. fields where the surface is represented using a different iso-value) need to be converted before they can be blended. Such a conversion can also provide parameters to influence the shape of the resulting blending range [2] and thus provide some control in blending. Since a conversion is only applied once to create compatible fields, blends relying on such a conversion suffer the same problems as the methods discussed above where only primitives can be adjusted.

A number of blending operations use a neighborhood graph to restrict blending to neighboring child nodes in the topology of the model [18, 12, 8]. These operations eliminate unwanted bulging in certain cases, but offer no control over the shape of the remaining blending volume.

The use of Convolution Surfaces reduces unwanted bulging in most blends [7, 6, 1]. Unfortunately blending can only be applied to skeletal primitives and does not offer any further control over the shape of the blending volume. Convolution Surfaces do not suffer from the influence problem when a primitive with a large skeleton is blended with a smaller primitive.

The pair-wise blends described in [15],[3] and [17] achieve blending of multiple objects by sequential blending. Each blend has two parameters to indicate the range of influence of each object to control the shape of the result. Unfortunately the binary blends force the user to treat two objects as one after they are blended together, which limits control in sequential blends with more than two objects.

The set of blending operations described in [9] and [3] use blending functions to describe the shape of the blending volume. This gives great control over the resulting shape when blending two child nodes, but only one blending function can be used even if more than two nodes are blended together. When the right blending function is used, unwanted bulging can be reduced, but the influence problem can usually only be solved for two child nodes.

The displacement method [24] uses a superelliptic definition for its resulting blend surface. Such a definition contains a weight parameters which can manipulate the influence of each child node in the blend. Just like methods that change the field functions of the child nodes, the displacement blend does not allow changing the partial blending volumes individually. Therefore it is not always possible to solve the influence problem and reduce unwanted bulging when more than two child nodes are used.

Bounded blending [21] uses a third implicit surface model to indicate the blend regions. With the use of the right bounding models and the right placement, the influence problem can be solved and unwanted bulging reduced. In some situations unwanted bulging can even be reduced without changing the rest of the blending volume, although this can result in visible discontinuities of the gradient. Bounded blending can also be used to form more complex blends such as 'partial edge blending' and 'multiple blending' as described in [21]. Unfortunately it is not always obvious what bounding models to use and correctly positioning the bounding models in 3D can be time consuming. Even though one can roughly predict what the results of a bounded blend will look like, the bounded blend does not offer intuitive control to fine tune the shape of the blend result.

In conclusion, most blending operations only partially solve the influence problem and unwanted bulging. Only bounded blending provides a complete solution, but unfortunately it lacks user friendly and intuitive control.

## 3 Adjusting the blending range

To partially solve the bulging problem and the influence problem, an adaptation is used of the methods that change the field function to influence the blend (as in [16, 29]). Instead of manipulating the field function directly (which is possible with skeletal primitives, but not with operations), a deformation function $m_k$ is used like the $gain_g$ function

used in Hypertexture [22]:

$$f_B(p) \quad = \quad \sum\{A \in N_B \mid m_{k_A}(f_A(p))\}. \qquad (6)$$

Here the amount of deformation of each child node $A$ is controlled by the parameter $k_A$. When $m_k$ is applied to a skeletal primitive this has a similar effect as manipulating the field function of this primitive. The main advantage of the use of $m_k$ is that it can also be applied to Blobtrees which are not a single skeletal primitive. There are a few requirements for $m_k$:

1. $m_k$ needs to be ascending to avoid unwanted field value peaks in the blending range,
2. $m_k(0) = 0$: Field values outside the bounds of the node must be $0$,
3. $m_k(c) = c$: The iso-value $c$ must be mapped to itself in order to preserve the shape of the node,
4. $m_k$ is at least $C^2$ continuous to prevent new $C^2$ discontinuities in the field values,
5. the parameter $k$ controls the amount of deformation.

The $gain_g$ function is not $C^2$ continuous (in Hypertexture [22] it is used for different purposes), so a new definition for $m_k(t)$ is calculated using Mathematica [26] that does fit the requirements:

$$m_k(t) \quad = \quad \begin{cases} 0 & \text{if } t \leq k \\ \frac{c(k-t)^3}{8(k-c)^5} e_k(t) & \text{if } k < t < 2c-k \\ 2c & \text{if } t \geq 2c-k \end{cases}, \qquad (7)$$

where

$$e_k(t) \quad = \quad 8k^2 - 25kc + 20c^2 + 9kt - 15ct + 3t^2, \quad 0 \leq k \leq c.$$

This definition of $m_k$ is easy to implement and fast to evaluate. There are other mappings that meet the requirements, but they will have similar results. Figure 6 shows the effect of $m_k$. Images a through d show cross sections of the point primitive. The part that is colored yellow represents the set of points inside the volume of the primitive and the blue part represents the points outside the volume (the blending range). The frequency of the contours represents the slope of the field values: regions with a high frequency contain a larger interval of field values while regions with a solid color contain a constant field value. Images b through d of figure 6 show a relationship between the frequency of the contours and the amount of deformation: a higher frequency means more deformation. When blending a set of nodes, each node $A$ will have its own parameter $k_A$ to control the amount of influence of that node:

$$f_B(p) \quad = \quad \sum\{A \in N_B \mid m_{k_A}(f_A(p))\}. \qquad (8)$$

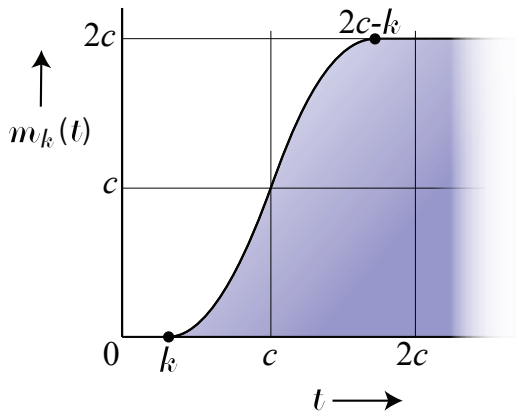


**Figure 5. The deformation function $m_k$ as defined in equation 7**

With the blend operation $B$ from equation 8 the influence of each child node of $D$ can be controlled individually. This gives some control over the resulting blend and can reduce the bulging and influence problem between two child nodes. When the bulging or influence problem spans more than two child nodes, the blend operation does not offer enough control for a solution. For example, in figure 7 nodes $A_0$ and $A_1$ are blended together and the blending volume (the red volume) is controlled by parameters $k_{A_0}$ and $k_{A_1}$. $k_{A_0}$ and $k_{A_1}$ are set to produce the wanted blend shape. When another node is added to the blend (node $A_2$ in figure 7), two more partial blending volumes need to be controlled (the yellow and green volumes) but only parameter $k_{A_2}$ can be used to do this without noticeably changing the blending volume between $A_0$ and $A_1$ (the red volume).

## 4 Local deformation

To solve the bulging and influence problem, more control is required which allows manipulation of the partial blending volumes individually. Ideally, changing the partial blending volume between two child nodes $A_0$ and $A_1$ would not significantly effect the rest of the blending volume. This can be achieved by localizing the deformations. In other words, the deformations of $A_0$ and $A_1$ will be stronger where $A_0$ and $A_1$ are close to one another. By localizing the deformations of a child node $A_0$, the blending volume between $A_0$ and each other child node can be controlled individually. This requires a set of parameters for every child node $A_0$, where each parameter will control the blending range in respect to the remaining child nodes. For a blend operation $B$ these parameters can be defined as $w_{A_0,A_1}$ where $A_0, A_1 \in N_B$. When $A_0 \neq A_1$, $w_{A_0,A_1}$ is the parameter to control the influence of $A_0$ in the blending

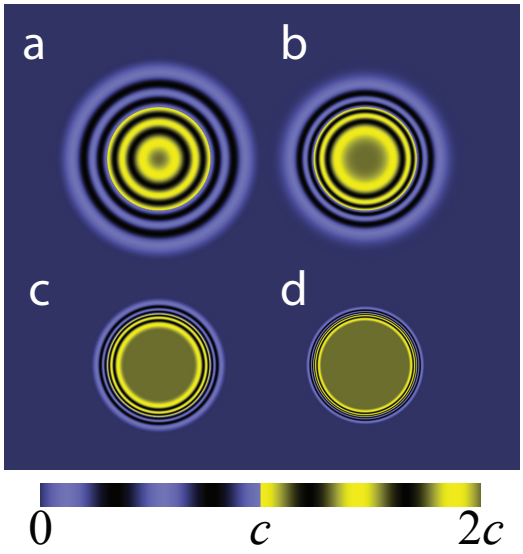


**Figure 6. Changing the influence of a node by applying $m_k$. a: Original point primitive. b, c and d: Deformed point primitives with respectively $k = 0$, $k = 0.5c$ and $k = 0.75c$.**

volume between $A_0$ and $A_1$. For $A_0 = A_1$, $w_{A_0,A_1}$ can be used to control the overall influence of $A_0$ like the $k_A$ parameter described in section 3.

The amount of deformation (or influence) of a child node $A_0$ in a point $p$ depends on the parameters $w_{A_0,A_1}, A_1 \in N_B$ and the proximity of the other child nodes. Since the field function used to construct skeletal primitives is descending (see figure 2), the field values of a Blobtree are in a direct relationship with the distance to the surface. A smaller field value translates into a greater distance from the surface, and vice versa. Therefore the field value $f_{A_1}(p)$ of a node $A_1$ in point $p$ can be used as a measurement for the proximity of $A_1$. The field value $f_{A_1}(p)$ combined with the parameter $w_{A_0,A_1}$ forms a localized contribution $s_{A_0,A_1}(p)$ to the deformation of $A_0$ in point $p$. Once all localized contributions are calculated they can be multiplied to form a final contribution in the range $[0, c]$. There are a number of requirements to the definition of $s_{A_0,A_1}(p)$:

1. when $A_1$ does not contribute to the deformation of $A_0$ in point $p$ ($f_{A_1}(p) = 0$), $s_{A_0,A_1}(p)$ will be equal to 1 (multiplication by 1 does not influence the final result),

2. the larger the contribution of $A_1$ towards the deformation of $A_0$, the smaller $s_{A_0,A_1}(p)$ will be (smaller values will result in a smaller final result),

3. when $f_{A_1}(p) \geq c$, $s_{A_0,A_1}(p)$ is constant (point $p$ lies inside the volume of $A_1$ and in this region the distance between $A_1$ and $p$ is a constant 0 so $s_{A_0,A_1}(p)$ should be constant as well)

4. the amount of contribution $s_{A_0,A_1}(p)$ should be directly related to the parameter $w_{A_0,A_1}$.

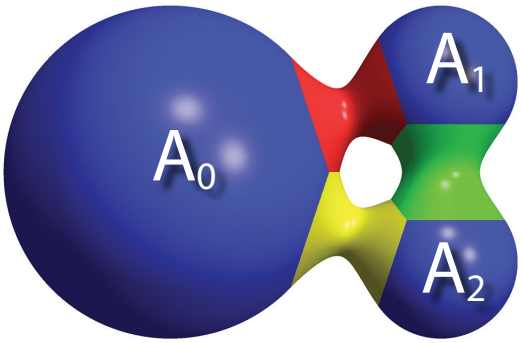


**Figure 7. Blending 3 nodes together results in 3 blending volumes (red, yellow and green)**

The following definition of $s_{A_0,A_1}(p)$ meets all requirements (see also figure 8):

$$s_{A_0,A_1}(p) = \begin{cases} 1 + (t^2 - 2t)(1 - w_{A_0,A_1}) & \text{if } t < 1 \\ w_{A_0,A_1} & \text{if } t \geq 1 \end{cases}, \quad \text{where } t = \frac{f_{A_1}(p)}{c}. \tag{9}$$

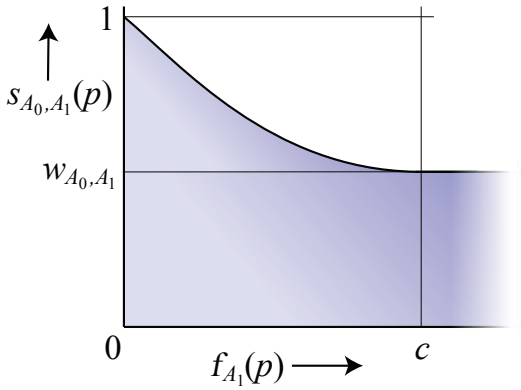


**Figure 8. The contribution $s_{A_0,A_1}(p)$ of node $A_1$ to the deformation of node $A_0$ in point $p$.**

After the localized contributions $s_{A_0,A_1}(p)$ have been calculated for each $A_1 \in N_B, A_1 \neq A_0$, they can be combined to form the final deformation contribution $k_{A_0}(p)$ as

follows:

$$k_{A_0}(p) = c\left(1 - w_{A_0,A_0} \prod_{A_1 \in N_B \setminus \{A_0\}} s_{A_0,A_1}(p)\right), \quad \text{where } s_{A_0,A_1}(p) \in (0,1] \tag{10}$$

Using equations 7, 10 and 9 the locally restricted blend can be constructed be as follows:

$$f_B(p) = \sum_{A \in N_B} m_{k_A(p)}(p). \tag{11}$$

## 5 Sequential blends

Besides a blending range with a more or less uniform width, the blending range should also remain wide enough. Figure 6b shows how the blending range of an object shrinks after applying the deformation function $m_k$ even when the parameter $k$ is at its smallest possible value. This means that after applying locally restricted blending, the blending range always shrinks compared to the blending ranges of the child nodes. When locally restricted blending is applied a number of times, the blending range will shrink to a point where the blending range is too small to be used for blending or other operations. To solve this problem, the deformation function $m_k$ needs to be modified so that it does not deform the blending range when $k = 0$. This can be achieved by replacing $m_k(t)$ with a smooth interpolation between $m_k(t)$ and the identity function $i(t) = t$. If we define $\widetilde{m}_k$ to be the replacement for $m_k$ (as defined in equation 7) then the definition of $\widetilde{m}_k$ is:

$$\widetilde{m}_k(t) = m_k(t)(1-s) + ts, \quad \text{where } s = \left(1 - \frac{k}{c}\right)^3. \tag{12}$$

The definition of $\widetilde{m}_k$ also meets the requirements for $m_k(t)$ (see section 3) and can therefore be used instead of $m_k(t)$.

Using $\widetilde{m}_k$ instead of $m_k(t)$ leaves the blend regions of the child nodes intact when they are not involved in the blend. At the same time other parts will still be deformed the same way as before. Using this alternative for $m_k$ may require minor adjustments of the parameters $w_{A_0,A_1}, A_0, A_1 \in N_B$ to get the same result. Figure 9 shows an example of a Blobtree model which contains sequential blend operations. Without the use of $\widetilde{m}_k$ (image a), the blending range of the final result has shrunk and is not suitable anymore for use in other blend operations. When $\widetilde{m}_k$ is used (image b), the blending range stays similar to the original blending ranges of the child nodes (image c).

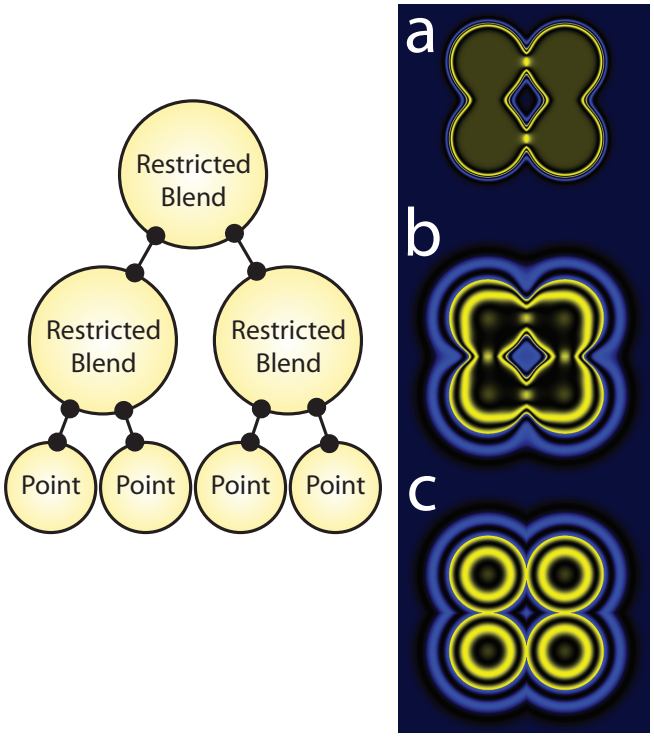


**Figure 9. Sequential blending. a: Restricted blend using $m_k(t)$, b: a: Restricted blend using $\widetilde{m}_k$, c: union operation for comparison.**

## 6 User interface

Blending $n$ Blobtree models together using locally restricted blending $B$ requires $n^2$ parameters. This may seem many, but their values can be set without much effort when a suitable user interface is used. The parameters $w_{A_0,A_1}, A_0, A_1 \in N_B$ can be split into two groups. The parameters where $A_0 = A_1$ are used to deforming the whole blending range of object $A_0$, whereas the parameters where $A_0 \neq A_1$ are used for local deformations of $A_0$. Using only selection of child nodes, the user can indicate which part of the blending range needs to be altered. The nodes can be selected from a list or directly from a visualization of the model using pick correlation [10].

After selecting a single child node $A_0$, the interface will allow the user to alter the blending range of $A_0$ as a whole (for example using a slider). The system will change the $w_{A_0,A_0}$ parameter accordingly, giving it a value between 0 (minimum blending range) and 1 (maximum blending range). This gives the user an intuitive control over the amount of blending volume everywhere around the node $A_0$.

When the user selects two child nodes $A_0$ and $A_1$, the interface will allow the user to alter the blending range in

between these objects. This can be done by changing two values: one to indicate the influence of $A_0$ and the other for $A_1$. The system will change the parameters $w_{A_0,A_1}$ and $w_{A_1,A_0}$ accordingly, giving them a value between 0 and 1. Parameters unchanged by the system will get the default value of 1. This gives the user an intuitive control of the blending volumes of $A_0$ and $A_1$ in the area in between these nodes. Figure 10 shows an example of the effect of changing these parameters where one parameter is represented by the columns and the other by the rows.

Visualization of the locally restricted blend is interactive (see section 7) and direct feedback can be given to the user when parameters are changed.

## 7 Results

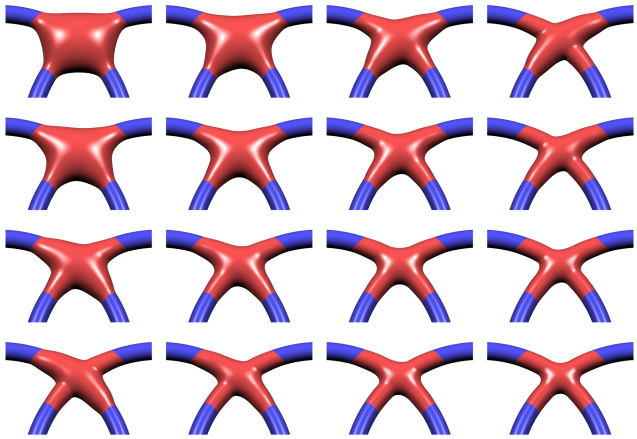

**Figure 10. Locally restricted blending between two tori $A_0$ and $A_1$. From left to right: $w_{A_0,A_1} = 1, 0.8, 0.6, 0.4$. From top to bottom $w_{A_1,A_0} = 1, 0.8, 0.6, 0.4$. In all cases $w_{A_0,A_0} = w_{A_1,A_1} = 1$. The red parts indicate the blending areas.**

Figure 10 shows the effect of changing the parameters $w_{A_0,A_1}$ and $w_{A_1,A_0}$ of the blend between two tori $A_0$ and $A_1$. Each parameter clearly controls the influence of one of the tori. In blends with more than two child nodes, the blending volume between each pair of child nodes can be controlled in the same way (see for example figure 7). Even though the number of parameters increases exponentially with the number of child nodes, these parameters can be set using the interface described in section 6 which is user friendly and intuitive.

Figures 13 and 15 demonstrate the possibilities of the locally restricted blend. In figure 13 the locally restricted blend has been applied to three child nodes (figure 11) and 6 out of 9 parameters needed to be set to get the wanted result. In figure 12 the locally restricted blend is compared to three other blends: the Ricci blend, a superelliptic blend and the bounded blend (the latter two are based on the blends from [24] and [21] adapted for Blobtree models). The Ricci blend does not allow the bulging around the handles to be reduced while maintaining the right size for the neck of the vase. The superelliptic blend allows a bit more control than the Ricci blend, but has similar problems. The bounded blend produces fairly good results (a slight bump remains at the top of the handles), but requires a significant amount of input from the user. In this case the bounding object is a union of four point primitives and one line primitive. Finding the right primitives, positioning and sizing of these primitives required 15 minutes of user interaction in an interactive system. The locally restricted blend produces the right shape, while user interaction required just under two minutes (setting 6 out of 9 parameters).

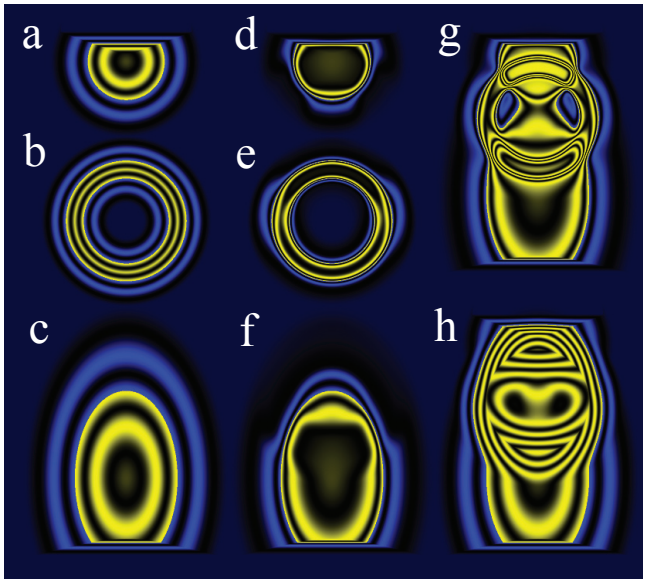


**Figure 11. Cross sections of the vase (figure 13): the child nodes (a,b,c), the deformed child nodes (d,e,f), locally restricted blending (g) and summation blending (h).**

The table below shows the average visualization times of the vase model in seconds. The second column shows the time needed to convert the Blobtree model into a polygonal mesh [28, 25] of approximately 9000 triangles. The third column shows the visualization times using a ray tracer for implicit surfaces [11, 13, 5] to create an image of 512 by 512 pixels. All results were achieved using a pc with an AMD Turion X2 processor at 1.9 GHz and 1.0 GB memory.

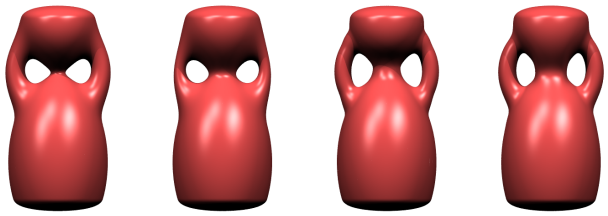

**Figure 12. From left to right: Ricci blend, superelliptic blend, bounded blend and locally restricted blend.**

**Figure 13. The vase.**

| Visualization times | Polygonization | Ray tracing |
|---|---|---|
| Ricci | 0.265 | 149 |
| Superelliptic | 0.250 | 153 |
| Bounded | 0.452 | 281 |
| Locally restricted | 0.343 | 204 |

In figure 15 locally restricted blending is applied to control the shape of some of the smaller details. 7 out of 36 parameters needed to be set to get the wanted result. With the user interface described in section 6, these parameters could be set in a matter of minutes. Partial blending volumes can be adjusted after selecting the right child nodes and the user does not have to worry about accidentally changing other partial blending volumes. Figure 14 shows the toy before and after applying the deformations of the locally restricted blend. Without applying the deformations, the locally restricted blend is equal to the summation blend.

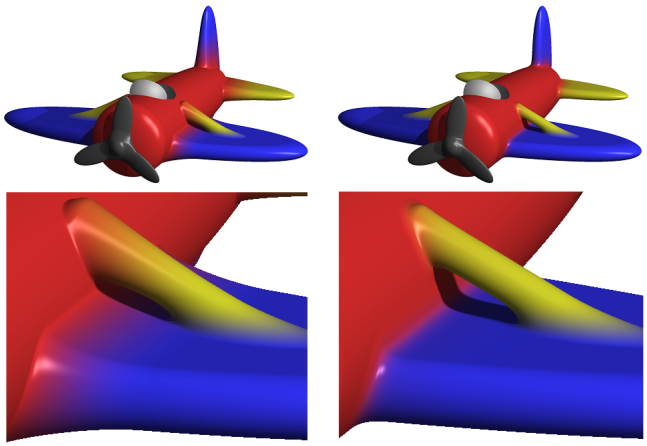

**Figure 14. Left: the toy before applying deformations. Right: the toy after applying deformations.**

**Figure 15. The toy.**

## 8 Conclusion and future work

Locally restricted blending gives the user more control over the blending volume. As can be seen in figures 4 and 3, the locally restricted blend solves the influence problem and reduces unwanted bulging, although reducing unwanted bulging can result in a reduction of the rest of the blending volume as well. With the user interface described in sec-

tion 6 the locally restricted blend can be applied in a user friendly and intuitive way. Locally restricted blends are somewhat slower to visualize compared to basic blend operations like Ricci and superelliptic blending. On the other hand, a locally restricted blend is faster to visualize than complex blends like bounded blends because there is no need to evaluate extra Blobtree nodes or compute gradients.

The disadvantage of locally restricted blending is that the blending range shrinks in the blend areas. The improvement discussed in section 5 significantly reduces this problem (see figure 9). Nevertheless the problem may remain in areas that require strong blending range deformations in order to get the required blend shape. Also, the blending range may become more irregular in places because it is a combination of deformed blending range of different objects. This may hinder further blending in those regions.

Even though the locally restricted blend offers control over the partial blending volumes individually, the locally restricted blend does not replace previous Blobtree blending methods. For example, it would be hard to imitate the 'partial edge blending' and 'multiple blending' features and the more flexible bulge reduction of bounded blending. A system which implements a number of blends e.g. the locally restricted blend and the bounded blend, would offer a more complete set of blending options the user could choose from depending on the users needs.

The locally restricted blend can only be applied when the blending ranges of the objects in the blend overlap. This is not only a restriction of this blending operation, but of all blending operations for bounded implicit surfaces. Even if it was possible to blend objects without overlapping blending ranges, the blending volume would stretch over the whole length of space in between the objects. In such cases it is often more practical to insert a third objects, like a thin line primitive to model the connection.

The locally restricted is designed for Blobtree models, but would be applicable to other bounded implicit surfaces with minor adaptations. An adaptation for unbounded implicit models would require more work, but is also possible.

Future work includes extending the locally restricted technique to smooth intersection and difference operations. Together with the blend (which is a smooth union), this would result in a complete set of CSG operations with smooth transitions. Furthermore, the restricted blend operation could be further improved by providing an automated system which creates initial values for the parameters to accommodate certain criteria. For example the values of the parameters could be adjusted to minimize curvature in the resulting blend. Finally, gradient information of the objects in the blend could be used to create an extension to our method which allows for a wider variety of blending shapes and better reduction of unwanted bulging.

## References


[1] A. Angelidis, P. Jepp, and M.-P. Cani. Implicit modelling with skeleton curves: Controlled blending in contact situation. In *Shape Modeling International*, pages 137–144. IEEE Computer Society, 2002.

[2] L. Barthe, V. Gaildrat, and R. Caubet. Combining implicit surfaces with soft blending in a CSG tree. In *Proceedings of CSG 98 , Ammerdown, UK*, pages 17–31, Information Geometers Ltd, 17-30 avril 1998. ISBN 1-874728-14-3. CSG Conference Series.

[3] L. Barthe, B. Wyvill, and E. de Groot. Controllable binary CSG operators for soft objects. In *International Journal of Shape Modeling*, volume 10, pages 135–154. december 2004.

[4] J. F. Blinn. A generalization of algebraic surface drawing. *ACM Transactions on Graphics*, 1(3):235–256, July 1982.

[5] J. Bloomenthal. *Introduction to Implicit Surfaces*. Morgan Kaufmann, ISBN 1-55860-233-X. Edited by Jules Bloomenthal With Chandrajit Bajaj, Jim Blinn, Marie-Paule Cani-Gascuel, Alyn Rockwood, Brian Wyvill, and Geoff Wyvill.

[6] J. Bloomenthal. Bulge elimination in convolution surfaces. *Computer Graphics Forum*, 16(1):31–41, 1997. ISSN 0167-7055.

[7] J. Bloomenthal and K. Shoemake. Convolution surfaces. *Computer Graphics*, 25(4):251–256, July 1991. Proceedings of SIGGRAPH'91 (Las Vegas, Nevada, July 1991).

[8] M.-P. Cani and S. Hornus. Subdivision-curve primitives: a new solution for interactive implicit modeling. In B. Werner, editor, *Proceedings of the International Conference on Shape Modeling and Applications (SMI-01)*, pages 82–88, Los Alamitos, CA, May 7–11 2001. IEEE Computer Society.

[9] D. Dekkers, K. van Overveld, and R. Goldsteijn. Combining csg modeling with soft blending using lipschitz-based implicit surfaces. *The Visual Computer*, 20(6):380–391, 2004.

[10] J. D. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes. *Computer graphics: principles and practice (2nd ed.)*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1990.

[11] A. S. Glassner. *An introduction to ray tracing*. Academic Press Ltd., 1989.

[12] A. Guy and B. Wyvil. Controlled blending for implicit surfaces using a graph. In *Implicit Surfaces '95*, pages 107–112, Apr. 1995.

[13] J. C. Hart. Sphere tracing: A geometric method for the antialiased ray tracing of implicit surfaces. *The Visual Computer*, 12(10):527–545, Dec. 1996.

[14] P.-C. Hsu and C. Lee. Field functions for blending range controls on soft objects. *Computer Graphics Forum*, 22(3):233–242, Sept. 2003.

[15] P.-C. Hsu and C. Lee. The scale method for blending operations in functionally-based constructive geometry. *Computer Graphics Forum*, 22(2):143–158, June 2003.

[16] Z. Kacic-Alesic and B. Wyvill. Controlled blending of procedural implicit surfaces. In *Graphics Interface'91*, pages 236–245, Calgary, Canada, June 1991.

[17] Q. Li. Smooth piecewise polynomial blending operations for implicit shapes. *Computer Graphics Forum*, 26(2):157–171, June 2007.

[18] A. Opalach and S. C. Maddock. Implicit surfaces: Appearance, blending and consistency. In *Proc. 4th Eurographics Workshop on Animation and Simulation*, pages 233–245, Sept. 1993.

[19] A. Pasko, V. Adzhiev, B. Schmitt, and C. Schlick. Constructive hypervolume modeling. *Graphical models*, 63(6):413–442, Nov. 2001.

[20] A. Pasko, V. Adzhiev, A. Sourin, and V. Savchenko. Function representation in geometric modeling: concepts, implementation and applications. *The Visual Computer*, 11(8):429–446, 1995.

[21] G. I. Pasko, A. A. Pasko, and T. L. Kunii. Bounded blending for function-based shape modeling. *IEEE Comput. Graph. Appl.*, 25(2):36–45, 2005.

[22] K. Perlin and E. M. Hoffert. Hypertexture. In *SIGGRAPH '89: Proceedings of the 16th annual conference on Computer graphics and interactive techniques*, pages 253–262, New York, NY, USA, 1989. ACM.

[23] A. Ricci. A constructive geometry for computer graphics. *Computer Journal*, 16(2):157–160, May 1973.

[24] A. P. Rockwood. The displacement method for implicit blending surfaces in solid models. *ACM Trans. Graph.*, 8(4):279–297, 1989.

[25] K. van Overveld and B. Wyvill. Shrinkwrap: An efficient adaptive algorithm for triangulating an iso-surface. *Vis. Comput.*, 20(6):362–379, 2004.

[26] Wolfram. Mathematica. http://www.wolfram.com/.

[27] B. Wyvill, A. Guy, and E. Galin. Extending the CSG tree. warping, blending and boolean operations in an implicit surface modeling system. *Computer Graphics Forum*, 18(2):149–158, June 1999. ISSN 1067-7055.

[28] B. Wyvill, C. McPheeters, and G. Wyvill. Data structure for soft objects. *The Visual Computer*, 2(4):227–234, 1986.

[29] B. Wyvill and G. Wyvill. Better blending of implicit objects at different scales. *ACM Siggraph 2000 presentation*, 2000.