

Shape Conformal Volumetric Interpolation with Interior Distances

Oleg Fryazinov, Mathieu Sanchez, Alexander Pasko
Bournemouth University, UK

Abstract

Source based heterogeneous modelling is a powerful way of defining gradient materials within a volume. The current solutions do not take into account the topology of the object and can provide counter intuitive results for complex objects. This paper presents a method to interpolate material properties and attributes based on the accessibility of the points in respect to the material features defined by the user. Our method requires the non overlapping source features with constant material to interpolate gradient materials, by using Voronoi diagrams on interior distances. It leads to intuitive material properties across the shape regardless of its topology or complexity. We show how the shape conformal field is defined inside the volume and can be extended outside the volume to create a valid operator for a heterogeneous modelling system dealing with scalar fields. The presented method is computationally efficient and has several applications, such as material property interpolation and shape aware procedural micro structures.

Keywords: Heterogeneous modelling, material interpolation, shape conformal, interior distance, procedural texturing

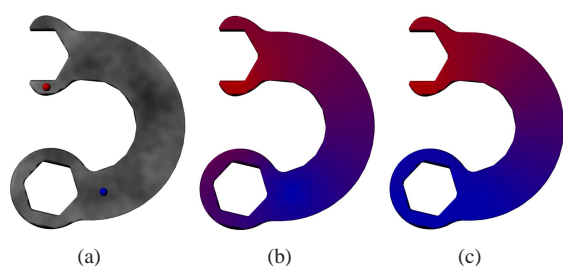


Figure 1: Material interpolation: a) Model with material features denoted, b) global transfinite interpolation, c) our method.

1. Introduction

Recent developments in the representation of real-life objects shows a paradigm shift from surface modelling where the interior of the object is homogeneous to a heterogeneous volume modelling approach. This allows us to take into account the complex internal structure of the object which can contain diverse materials and possess various physical properties. Heterogeneous modelling has found its applications in a wide range of diverse areas from biological and medical research and bio-engineering to multi-material design for fabrication and digital entertainment applications.

There exist several ways to define the interior structure of the object. The simplest one is explicit defini-

tion, e.g. for a large number of points that belong to the interior of the object we can specify the values of the attributes such as material, density etc. Alternatively if the geometry of the object is represented by a voxel set, the attributes can be defined per voxel. However in practice explicit definition is very inefficient because the more precise the materials are defined the more points or voxels should be set and the amount of data can be unbearable in this case. Therefore procedural methods become more suitable for the purposes of the material definition. A convenient way to do this is to specify the attributes only in a finite number of points or closed areas and interpolate the values elsewhere. Most of the interpolation methods, such as transfinite interpolation [1] however, do not take into account the shape of the object itself but rather providing a general interpolation within the whole space.

In this work we present a method that allows to define the interior structure of the object by using simple interpolation techniques yet taking the shape of the object into an account. We are working with the object defined by continuous scalar fields that allows us to work with arbitrary accuracy without increasing the complexity. As a result, we allow the user to set up the materials in a convenient way and efficiently evaluate the material in arbitrary point in space inside as well as outside the volume object. In this paper we present formulations for

shape-aware material distribution for the volume object represented by a scalar field, discuss the practical approach for interior distance fields and present some applications of our method for heterogeneous modelling.

2. Background and Related Work

2.1. Procedural material distribution

Different approaches exist to define the material distribution inside a volume object using limited information. Some methods use the attributes at the surface to extrapolate the attributes inside the volume by propagation. Thus, in [2] the texture coordinates for the point inside a source shape are obtained by finding the corresponding point on the support surface by letting particles flow through the gradient vector field. In general, this method can lead to discontinuities and is very limited in modifying attributes inside the volume. Source-based methods allow users to specify more information regarding attributes inside the shape by defining the sources for the material distribution and procedures to find the values of the attributes in the ambiguous areas where no source has been set up. Heterogeneous modelling using source features was introduced in [3] where the standard set-theoretic operations were used in the context of material properties. The gaps in the material distribution were filled by using a weighted sum of the materials defined by the neighbouring source features. In [4] the material distribution is defined by using optimisation techniques with parametric distances. In [5] is shown how the particular attributes of the model can be interpolated by using mass transport. However the generalisation for arbitrary attributes is not straightforward and the solution requires solving PDE systems for querying the attribute value in a point in space which makes it impractical for interactive applications. In [1] transfinite interpolation is used to define the material properties in space including the interior of the object. The simple formulation is used to find the attributes in the areas between the sources, however the method is global and does not take the shape of the object into account. The shape of the object can be taken into account for a limited types of input data. Thus, in [6] the medial axis is used to define shape-aware internal structure, which makes it uneasy to use for the geometry where the medial axis is hard to find, e.g. implicitly defined.

2.2. Interior distance fields

Interior (inner) distance measures the path between two points inside the object where all the points of the

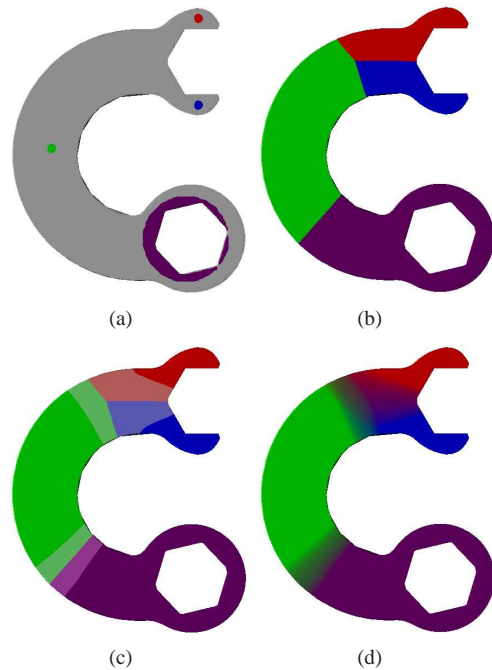


Figure 2: Algorithm details: a) Source model with the material feature defined, 3 point features (denoted by red, blue and green) and 1 solid (denoted by purple); b) Voronoi diagram applied for interior distance fields defined by the material features; c) Buffer zones based on the Voronoi diagrams; d) Result of the interpolation

path belong to the interior or the surface of the object. Various ways to define and calculate the inner distance between two points belonging to a closed object exist, including barycentric coordinates of a polygonal mesh [7] and a visibility graph for the volumetric object [8]. Also, the inner distance can be defined by using other ways, for example, in terms of Eikonal equation [9] and heat flow [10]. In general, as noted in [7], the interior distance can be expressed in a continuous setting, however in practical applications usually approximations are used.

3. Shape-aware interpolation

In our method we see the material distribution as a function $f_m : \mathbb{R}^3 \rightarrow \mathbb{R}^n$ where n denotes the number of real numbers to describe the attributes in the material distribution. In the simplest case the material is defined by a real-valued function, i.e. $n = 1$, the examples of this material can be value of the material density or the contrast of the colour defining the material (K component). In this paper for visualisation purposes for most of the examples we define the material by its RGB colour, $n = 3$.

We are working with the geometry defined in an implicit form, that means the volume object can be represented by its boundary (e.g. polygonal mesh), but we convert this boundary to a function for which the sign of the value in the given point denotes whether the point in question lies inside, outside or on the surface of the volume. Given the volume object S , the shape of the volume object is defined as a scalar field with a defining function $f_s(\mathbf{x})$. The sign of the defining function distinguishes the interior and the exterior of the volume, i.e. $f_s(\mathbf{x}) > 0$ for the exterior, $f_s(\mathbf{x}) < 0$ for the interior and $f_s(\mathbf{x}) = 0$ for the boundary of the volume object.

Inside the volumetric object, we define a number of material features, i.e. closed regions where the geometry and the material are known. The geometry of each material feature is also represented as a scalar field with the defining function $f_{m_i}(\mathbf{x})$, where $i = 1 \dots n$ is the index of the given material feature and n features are defined overall.

The steps of the method are shown on the figure 2. In brief, to make the interpolation, we use material source features (figure 2a) to propagate the interior distance field with respect to the input shape. Next, we construct a Voronoi diagram based on these distance fields (figure 2b) to build attribute cells for each material source and interpolation zones (figure 2c). Finally we use an interpolation between the scalar fields of the resulting Voronoi cells (figure 2d). Below we describe our method in detail.

3.1. Interior distance field for material features

In our work we use an approximate Euclidean shortest path [11] to define the interior distance field. The choice of this method over other existing in current state of the art (see section 2.2) is that it is easiest in terms of implementation and as efficient as some others (e.g.[9]).

Given the material feature i with the defining function $f_{m_i}(\mathbf{x})$, the value of the field for this material feature is defined as following:

$$d_i(\mathbf{x}) = \min_{\mathbf{x}_j: f_{m_i}(\mathbf{x}_j) > 0} (\mathbf{x} \ \mathbf{x}_j) \quad (1)$$

Here $(\mathbf{x} \ \mathbf{x}_j)$ denotes Euclidean shortest path between points \mathbf{x} and \mathbf{x}_j inside the volume object S and $\|\cdot\|$ denotes length of this path.

In practice the exact calculation of Euclidean shortest path is an NP-hard task and given the additional complexity of the global minimum operator, we only calculate the approximate Euclidean shortest path and hence approximate interior distance. We discuss the details of the process of construction and querying the value of the interior distance field in section 4.

3.2. Interior distance Voronoi diagram

An important concept we use in our material interpolation is the Voronoi diagram which we apply to scalar fields.

First, we revise the definition of the Voronoi diagram for Euclidean distance. Given the set of n points \mathbf{v}_i , $i = 1 \dots n$, the Voronoi diagram V is the collection of subsets V_i where each subset can be defined by the following [12]:

$$V = \{V_i; V_i = \{\mathbf{x} \mid \|\mathbf{x} - \mathbf{v}_i\| < \|\mathbf{x} - \mathbf{v}_j\|, j \neq i\}\} \quad (2)$$

Here $\|\mathbf{x} - \mathbf{v}_i\|$ denotes the magnitude of the difference between the vectors \mathbf{x} and \mathbf{v}_i and hence the Euclidean distance between them. The points \mathbf{v}_i are called Voronoi seeds and subsets V_i are Voronoi cells of \mathbf{v}_i . Each pair of Voronoi seeds i and j define a plane that is equidistant from these seeds:

$$\|\mathbf{x} - \mathbf{v}_i\| = \|\mathbf{x} - \mathbf{v}_j\| \quad (3)$$

In our work we use generalised Voronoi diagram. This generalisation has two steps: first, instead of points that define Voronoi seeds as objects with the same dimension as \mathbf{x} , i.e. we use volumes not just the points and secondly, instead of the Euclidean distance between Voronoi vertex and query point in the equations 2 and 3 we use the value of the interior distance field. Therefore the plane from the equation 3 becomes a surface in space that is defined as:

$$d_i(\mathbf{x}) = d_j(\mathbf{x}) \quad (4)$$

Here d_i and d_j represent the absolute value of the interior distance from the query point \mathbf{x} to the volumes defining material feature. This surface defines a half-space defined in implicit form as following:

$$h_{ij}(\mathbf{x}) = d_j(\mathbf{x}) - d_i(\mathbf{x}) \quad (5)$$

It can be seen that the shape of each Voronoi cell can be represented as an intersection of all the halfspaces for the Voronoi seed of the cell:

$$v_i(\mathbf{x}) = \bigcap_{j=1, j \neq i}^n h_{ij}(\mathbf{x}) \quad (6)$$

Here \bigcap denotes a set-theoretic operation (R-function) for intersection. Various functions exist for performing set-theoretic intersections, however for the purposes of modelling with scalar fields the following properties should be held:

The function \bigcap should have commutative and associative properties

The function should preserve the distance property, i.e. for any three points \mathbf{p}_1 \mathbf{p}_2 \mathbf{p}_3 the following condition is held:

$$f(\mathbf{p}_1) f(\mathbf{p}_2) f(\mathbf{p}_1) f(\mathbf{p}_3) f(\mathbf{p}_2) f(\mathbf{p}_3)$$

where $f(\mathbf{p})$ denotes the value of the scalar field at the point \mathbf{p} .

In practice, only a few R-functions satisfy these criteria:

R_1 system [13] also known as min-max functions

The system $\overset{0}{}$ [14]

Signed approximate real distance functions [15] (SARDF)

As the R_1 system results in C^1 -discontinuities and SARDF functions are expensive to compute, in our implementation we use the $\overset{0}{}$ system, where the intersection operation is defined by:

$$\begin{matrix} x_1 & x_2 & x_1 x_2 (x_1^n & x_2^n)^{\frac{1}{n}} & x_1 & 0 & x_2 & 0 \\ & & x_1 & & x_1 & 0 & x_2 & 0 \\ & & x_2 & & x_1 & 0 & x_2 & 0 \\ & & (1)^n & 1 (x_1^n & x_2^n)^{\frac{1}{n}} & x_1 & 0 & x_2 & 0 \end{matrix} \quad (7)$$

So the scalar field for each cell in the Voronoi diagram is therefore represented by:

$$v_i(\mathbf{x}) = \prod_{j=1}^n \left(\frac{d_j(\mathbf{x})}{d_i(\mathbf{x})} \right) \quad (8)$$

Here the \prod operation is described by equation 7.

The example of the generalised Voronoi diagram is shown on the figure 2b, where the cells are defined by 4 material source features, including three point features and one volume feature.

3.3. Material interpolation with Voronoi diagrams and interior distance fields

While presenting equation 8 we defined Voronoi cells using interior distances, which means that the boundary of the Voronoi cells is shape conformal. In the material interpolation step we use this boundary to perform material interpolation.

For material interpolation we distinguish two sub-spaces (figure 2c):

The interpolation zones where the materials from different cells are actually interpolated; these zones are defined around the border of each Voronoi cell for the given material source;

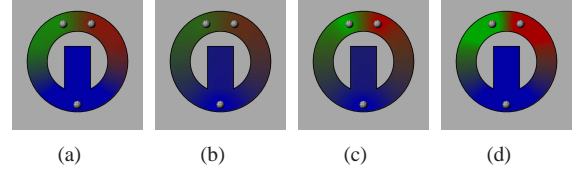


Figure 3: Various $\overset{0}{}$ set values s_{ij} : a) Small uniform $\overset{0}{}$ set value leads to a large constant blue block at the bottom, b) Large uniform $\overset{0}{}$ set value leads to washed out colours for close source features, c) Using an adaptive $\overset{0}{}$ set value based on distance between sources leads to intuitive results, d) The adaptive value can be scaled to $\overset{0}{}$ set where the interpolation starts.

The interior of the cell where only material defined for this cell is applied; this zone can be obtained by subtracting the point belonging to the interpolation zone from the point set for the Voronoi cell for the given material feature;

The interpolation zone can be defined by applying an $\overset{0}{}$ setting operation to the halfspace h_{ij} (the same as we use in the equation 6) separating cells i and j . The field for the interpolation zone between cells i and j can be defined as follows:

$$b_{ij}(\mathbf{x}) = h_{ij}(\mathbf{x}) \overset{0}{s_{ij}} d_j(\mathbf{x}) d_i(\mathbf{x}) \overset{0}{s_{ij}} \quad (9)$$

Here s_{ij} defines the $\overset{0}{}$ set value and hence the width of the interpolation zone between two feature materials i and j . Different values for the $\overset{0}{}$ set can be chosen. The first simplest option is to choose the value of the $\overset{0}{}$ set uniformly, i.e. such that $s_{ij} = s_{ji}$, however it might lead to undesirable effects. For example, in figure 3a, a small value for the $\overset{0}{}$ set was used so that the interpolation would keep the material features denoted by green and by red intact. As a result, the transition between the material features denoted by green and by blue is so short and the feature denoted by blue seems to expand more than it should. On the other hand increasing the uniform $\overset{0}{}$ set value causes the blur effect between two materials whose sources are close (figure 3b). Another option to choose the value for the $\overset{0}{}$ set is to make it dependent on the relative position of two material sources, such as the Euclidean or interior distance between these sources. The figure 3c shows $\overset{0}{}$ set values s_{ij} adjusted to fit interior distances between features. In figure 3d, the transition distances were scaled to give more room for constant material.

Finally, to find the final value for the interpolated material, we use a weighted summation of the materials participating for the given point:

$$\mathbf{c}(\mathbf{x}) = \frac{\sum_{k=1}^n w_k(\mathbf{x}) \tilde{\mathbf{c}}_k}{\sum_{k=1}^n w_k(\mathbf{x})} \quad (10)$$

Here n represents the number of material sources, \tilde{c}_k is the value of the k^{th} material properties and w_k is the weight:

$$w_k(\mathbf{x}) = \frac{\sum_{j=1}^n b_{kj}(\mathbf{x})}{\sum_{j=1}^n \tilde{c}_k} \quad (11)$$

This equation corresponds to the interpolation of n features in [1], using $b_{kj}(\mathbf{x})$ instead of the distance fields. Note that the denominator cannot be zero as Voronoi cells are not overlapping.

Note that the material parameters can be represented as a multidimensional vector and here we assume that simple linear interpolation can be used for such vectors. In some cases, however, additional interpolation constraints should be applied, for example, to interpolate between two colours that are defined in RGB colour space in a different colour space. In this case, equation 10 should be adjusted according to the interpolation scheme for the given attributes. In our examples all the attributes can be interpolated linearly and therefore other interpolation schemes are out of the scope of this paper.

4. Efficient approximate interior distance field for material sources

As we noted above, the calculation of the interior distance field is a fundamental part of our method. In this section we discuss a simple solution to create such field and query its values given any point in space, including outside the bounds of the object. We separate the procedure into two steps. First, we calculate the interior distance for the cells inside the object. In the second step we provide voxel values for the cells outside the shape. The values outside the shape are also estimated outside the bounds of the grid based on the grid boundary values.

In order to get a continuous field, we use trilinear interpolation. Since the values outside the bounds of the grid are extrapolated on the fly, the continuity is maintained throughout space.

4.1. Interior Distance Propagation

The use of a 3-dimensional regular grid is one of the most common ways to approximate the Euclidean shortest path inside a volume object by using the regular grid defined for the interior of the object. In [11] such a grid

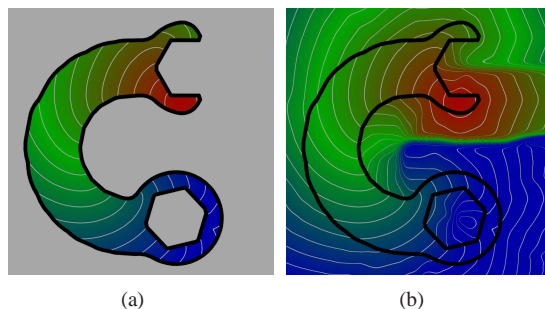


Figure 4: The interior distance field inside and outside the shape: a) The interior distance field for the shape, b) its corresponding extension to the exterior of the shape.

is called a cuboidal world. The representation of the object using distance fields gives us a fast and simple way to build such a structure.

For cells that were found to be inside the given shape, we find the set C_i of the cells that belong to the boundary of the material feature i by using the same approach: we process the cells that contain the zero-level set of the function f_{m_i} by checking the value of this function in the corners of the given cell.

Unlike methods to calculate the Euclidean shortest path between two points, we need to be able to calculate the shortest path between a point and the material feature. To achieve that, we use a propagation of the distance in the given regular grid. The grid is represented as a graph where cells are vertices of the graph and the adjacency information forms the edges of the graph. Then we use a modified A* search algorithm which we present in Algorithm 1.

If only the 6 direct neighbours are used, the distances will be Manhattan distances. By using 26 neighbours, the results are better although some patterns still appear. The results can be improved by finding the root of the straight segment leading to a sample, and checking for visibility. The details are in algorithm 1.

4.2. Exterior propagation

To construct the interior distance field in section 3, we use the Euclidean shortest path between the query point and the material feature shape. However this restricts the topology of the volume object and the positioning of the material sources. For example, if the object has two disjointed components, the shortest path between the point which lies inside the first component and the material feature which lies inside the second component is ill-defined. The similar situation occurs where the material feature is defined outside the volume object.

Input: *seeds*: Initial seeds
G: A grid
S: Shape which defines the boundaries
Output: Seeds for exterior propagation *O*
F: Priority queue of cell coordinates ;

```

foreach p in seeds do
  | insert p to F;
end
while F not empty do
  | cell coordinates current retrieved from front;
  | currentvalue value at current in G;
  | currentroot value at current in G;
  | rootvalue value at currentroot in G;
  | foreach Neighbour coordinate neighbour do
  | | if neighbour outside S then
  | | | frontdistance
  | | | distance(neighbour current);
  | | | recordeddistance value at
  | | | neighbour in G;
  | | | if frontdistance < recordeddistance
  | | | then
  | | | | value at neighbour in G
  | | | | frontdistance;
  | | | end
  | | | insert unique neighbour in O;
  | | else
  | | | if isVisible(neighbour, currentroot)
  | | | then
  | | | | frontdistance
  | | | | distance(neighbour currentroot)
  | | | | rootvalue;
  | | | | neighbourroot = currentroot;
  | | | else
  | | | | frontdistance
  | | | | distance(neighbour current)
  | | | | currentvalue;
  | | | | neighbourroot = current;
  | | | end
  | | | recordeddistance value at
  | | | neighbour in G;
  | | | if recordeddistance < frontdistance
  | | | then
  | | | | continue;
  | | | end
  | | | value at neighbour in G
  | | | frontdistance;
  | | | root at neighbour in G
  | | | neighbourroot;
  | | | insert neighbour to F;
  | | end
  | end
end

```

Algorithm 1: Interior propagation

To resolve this, our method to calculate the interior distance can be extended to query values outside the shape. The value at the given point outside the volume object can be found in either of two ways:

1. If the point lies inside the bounds of the grid, we use the same procedure as for interior distances, however we apply it twice - first, to calculate the shortest path to the material source within the shape and second, the shortest path to the shape;
2. If the point lies outside the grid, we extrapolate the cell data to preserve the rate of change.

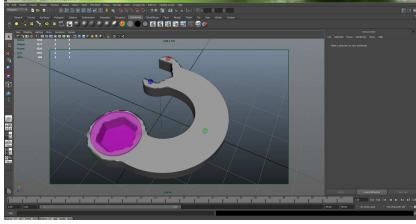
First, we evaluate grid samples within the bounds of the grid. The value of the sample can be found by using a similar procedure to algorithm 1, where the seeds are provided by the first propagation (*O* in 1). The only difference is that the algorithm does not check if it is crossing the field, but cannot overwrite values written by algorithm 1 (interior distances). The seeds are initialized to zero instead of carrying over the value of the interior distance. This is because we try to keep the discontinuity far from the surface. Once the cells have been computed, we add back the interior distance offset to all the cells calculated from this second step. This means that the boundary cells on the outside of the shape have close values to the ones on the inside. This is necessary to avoid our cubic interpolation to make the object surface values noisy.

Finally, we run several iterations of a local blur on all the cells outside the shape. We use a simple uniform blur, but better kernels exist. This additional step is made to reduce the high variations in the field.

To sample any point in space, we also need to extrapolate our samples outside the bounds of our grid. For simplicity, we extrapolate the discrete data on the fly from the boundary values of the grid. The extrapolation needs to maintain the rate of change the boundary cells have and carry over their value. We propose this simple formula which provides intuitive results:

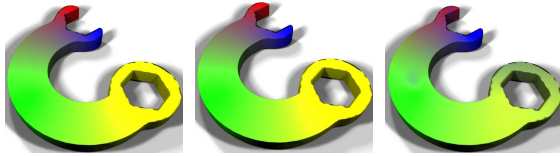
$$\begin{aligned}
 V(x, y, z) &= V(x_b, y_b, z_b) \\
 &+ V(x_d, y_b, z_b) \cdot d(x, x_b) \\
 &+ V(x_b, y_d, z_b) \cdot d(y, y_b) \\
 &+ V(x_b, y_b, z_d) \cdot d(z, z_b)
 \end{aligned} \tag{12}$$

Here x, y, z are the integer sample coordinates and are not bounded by the grid, x_b, y_b, z_b are the bounded coordinates of x, y, z within the grid, x_d, y_d, z_d are bounded coordinates offset by one within the grid. This means that if x is lower than x_b , then $x_d = x_b - 1$, otherwise if x is greater than x_b then $x_d = x_b + 1$. The function d simply returns the distance between the two coordinates.



(a)

Figure 5: Material properties are easily set up using our method.



(a)

(b)

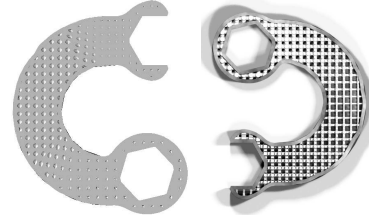
(c)

Figure 6: Influence of the resolution of the voxel grid on the interpolation: a) Resolution 256^3 , b) Resolution 64^3 , c) Resolution 32^3

5. Application and results

We have tested our method on a number of volume objects where the number of material sources were set up by the user. The implementation was done as a Maya plug-in (figure 5) allowing the user to set up the material sources in an easy and intuitive way. The average timings for the set up the interior distances for the model is about 5 seconds (all the measurements were performed on an Intel Xeon CPU at 3.33GHz, with 8 GB of RAM and without the use of multi threading) on a dense resolution we have tried (a 256^3 grid), however this resolution can be decreased without significant changes in the result of the interpolation (see figure 6). As expected, the number of polygons does not seem to have an impact on the computation and querying of a value in 16 million points takes around 8 seconds. This allows to potentially use our method for interactive applications, especially if interior distances are pre-defined in advance.

Most of the figures in this paper illustrate interpolation over the material represented as a colour distribution inside the shape. However many other attributes can be defined for the interior of the object. Another way to represent the heterogeneous nature of objects is to define microstructures (i.e. internal spatial geometric structures with size of detail orders of magnitude smaller than the overall size of the object) inside them. Microstructures can be described procedurally [16] by a function with several parameters such as orientation and thickness and therefore its parameters can be interpolated using our method.



(a)

(b)



(c)

Figure 7: Procedural microstructure created with our method: a) Porosity structure with interpolation over the pores size, b) Regular lattice with interpolated parameters, c) Foam structure with several parameters.

In figure 7a we show how simple parameters of the microstructure can be interpolated on an example of porosity parameter. More parameters can be defined for each material source and interpolated. Thus, in figure 7b different lattice structures are defined for different material sources: cylindrical lattice with different thickness for two material sources and cubical lattice with different thickness for another two material sources. Our interpolation method provides a weighting for each feature, and therefore we supply blend all the lattices together using a weighted sum of the function values of each lattice. Finally, in figure 7c, we use four features where several values are set. The objective is to control the attributes of the foam structure which are average cell size, wall thickness and roundness (blending amount). Each feature provides a value for each of those parameters. The large red feature uses a small cell size, large blending value and thick walls. The green feature contrasts with narrow walls, practically no blending and ample cell size. The last two features (yellow and blue) have high blending, thick walls. The yellow feature has the same cell size as the red feature while the blue is slightly larger.

6. Conclusions and discussion

Volumetric interpolation of object properties is an important operation for heterogeneous modelling of volumetric objects. While users can define some of the material sources, in the end one should be able to define the material distribution with as few material sources as possible. This is why the shape of the volume object should be taken into account. In this paper we show that the use of internal distances is a good way to take the shape of the volume object into account while interpolating the material attributes in the areas where the material is not defined. Our method allows for the definition of material sources as geometric objects defined by continuous scalar fields and therefore by using internal distances and Voronoi diagrams together achieve intuitive results to the user. It should be noted however that for many real-life applications additional constraints, such as gradients, profile curves and so on are often defined for the material distribution. While we have not discussed these constraints in this paper, we feel that it should be further researched.

In our experiments we have used small numbers of material sources. One can observe that if the number of sources is very large and limits to infinity, inside the shape the method converges to standard transfinite interpolation. As the formulation for transfinite interpolation is simpler and it is computationally less expensive than our method it would be interesting to find the point at which the number of material sources where our method produces the same or similar results with transfinite interpolation. In any case, our method still allows the user to make the input as little as possible.

Also in this paper we presented a way to extend the interior distances to allow the material attributes to be defined at any point outside the volume object. While the solution is straightforward, strictly speaking the idea of interior distances defined externally is mathematically weak. More research is required to prove that shape conformal methods are not ill-defined outside the given volume object.

Finally, we have presented a few applications in this paper. Heterogeneous modelling of volume objects is a relatively new topic and more applications are yet to be found and tested against our method.

Acknowledgements

This work was partially sponsored by the EU Interreg IVA project 5-023-FR_SHIVA.

References

- [1] A. Biswas, V. Shapiro, I. Tsukanov, Heterogeneous material modeling with distance fields, *Comput. Aided Geom. Des.* 21 (3) (2004) 215–242.
- [2] R. Zonenschein, J. Gomes, L. Velho, L. de Figueiredo, M. Tigges, B. Wyvill, Texturing composite deformable implicit objects, in: *Proceedings of the International Symposium on Computer Graphics, Image Processing, and Vision, SIBGRAPHI '98*, IEEE Computer Society, Washington, DC, USA, 1998, pp. 346–353.
- [3] Y. Siu, S. Tan, Modeling the material grading and structures of heterogeneous objects for layered manufacturing, *Computer-Aided Design* 34 (10) (2002) 705 – 716.
- [4] K. Samanta, B. Koc, Feature-based material blending for heterogeneous object modeling, in: A. Pasko, V. Adzhiev, P. Comninos (Eds.), *Heterogeneous Objects Modelling and Applications*, Vol. 4889 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, 2008, pp. 142–166.
- [5] N. Bonneel, M. van de Panne, S. Paris, W. Heidrich, Displacement interpolation using lagrangian mass transport, *ACM Trans. Graph.* 30 (6) (2011) 158:1–158:12.
- [6] A. K. M. Khoda, I. T. Ozbolat, B. Koc, Designing heterogeneous porous tissue scaffolds for additive manufacturing processes, *Computer Aided Design* 45 (12) (2013) 1507–1523.
- [7] R. M. Rustamov, Y. Lipman, T. Funkhouser, Interior distance using barycentric coordinates, in: *Proceedings of the Symposium on Geometry Processing, SGP '09*, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 2009, pp. 1279–1288.
- [8] Y.-S. Liu, K. Ramani, M. Liu, Computing the inner distances of volumetric models for articulated shape description with a visibility graph, *IEEE Trans. Pattern Anal. Mach. Intell.* 33 (12) (2011) 2538–2544.
- [9] M. Hassouna, A. Farag, Multistencils fast marching methods: A highly accurate solution to the eikonal equation on cartesian domains, *Pattern Analysis and Machine Intelligence*, *IEEE Transactions on* 29 (9) (2007) 1563–1574. doi:10.1109/TPAMI.2007.1154.
- [10] K. Crane, C. Weischedel, M. Wardetzky, Geodesics in heat: A new approach to computing distance based on heat flow, *ACM Trans. Graph.* 32 (5) (2013) 152:1–152:11. doi:10.1145/2516971.2516977.
- [11] F. Li, R. Klette, *Euclidean Shortest Paths - Exact or Approximate Algorithms.*, Springer, 2011.
- [12] B. Levy, N. Bonneel, Variational anisotropic surface meshing with voronoi parallel linear enumeration, in: X. Jiao, J.-C. Weill (Eds.), *Proceedings of the 21st International Meshing Roundtable*, Springer Berlin Heidelberg, 2013, pp. 349–366.
- [13] V. Shapiro, Semi-analytic geometry with R-functions, *Acta Numerica* 16 (2007) 239–303.
- [14] V. L. Rvachev, *Theory of R-functions and Some Applications*, Naukova Dumka, 1982.
- [15] P.-A. Fayolle, A. Pasko, B. Schmitt, Heterogeneous objects modelling and applications, Springer-Verlag, Berlin, Heidelberg, 2008, Ch. SARDF: signed approximate real distance functions in heterogeneous objects modeling, pp. 118–141.
- [16] A. Pasko, O. Fryazinov, T. Vilbrandt, P.-A. Fayolle, V. Adzhiev, Procedural function-based modelling of volumetric microstructures, *Graphical Models* 73 (5) (2011) 165 – 181.