Diffusion-based parametrization of surfaces on 3D-meshes

Martin Schmidt
Philipps-University of Marburg
Hans-Meerwein-Str. 6
Germany, 35032 Marburg
schmidtma@informatik.uni-marburg.de

Michael Guthe
Philipps-University of Marburg
Hans-Meerwein-Str. 6
Germany, 35032 Marburg
guthe@informatik.uni-marburg.de

Volker Blanz
University of Siegen
Hölderlinstr. 3
57076 Siegen
blanz@informatik.uni-siegen.de

ABSTRACT

This work presents a new approach towards parametrization of three-dimensional wireframe models. The method is derived from a specific phenomenon of cellular development in nature. It recreates the effect of diffusion of messengers through tissue, which leads to the formation of extremities and other anatomical structures depending on the position on the tissue surface. This process of diffusion on the surface is analyzed and simplified for usage as a parametrization of mesh surfaces. The presented approach uses the similarity of wireframe meshes and graphs in order to carry out the mechanism of diffusion. For this it implements a specialized algorithm based on Dijkstra's algorithm for finding the shortest paths.
The results of this mechanism are saved and organized in a binary tree structure, which allows for simple and efficient construction of correspondence between two distinct meshes. The paper concludes with an outlook on possibilities of further development and enhancements of the approach.

Keywords

mesh, parametrization, diffusion, geometric analysis

1 INTRODUCTION

Whenever computers analyze or display visual objects, they need to represent these objects in a suitable mathematical form that is appropriate to the processing tasks. The typical representation of an object is a polygon mesh, but for many purposes such as texturing, object retrieval, shape comparisons, differential geometry or for computing point-to-point correspondence between pairs of objects, it is important to describe shapes as parameterized surfaces. Hence, the purpose of this paper is to introduce a new approach in parametrization and its evaluation.

1.1 Parametrization

In the following, we give a short introduction on the topic of parametrization, its applications and related work. This is concluded by the analysis of the distinction of our approach in contrast to previous work.

Basics

An important aspect of 3D shape processing and analysis is the connection of objects with each other. By a correct link between two objects, algorithms can transfer knowledge and properties from one object to the other. It is also possible to describe the similarity between these objects. This link is generated by a surface parametrization that is consistent across different objects.

A parametrization of surfaces is defined as a mapping from a parameter domain onto the surface. This is also called *Surface-to-Surface-Mapping*, if the parameter domain is a surface itself. The parametrization of a parameter domain maps each point on the domain onto a certain point of the surface.

Bijectivity is an important property of parametrizations, as many applications rely on a complete cover of the originating surface without introducing ambiguity. That means each point on the surface maps to exactly one point on the parameter domain and vice versa.

Possible applications in computer graphics

There are several possible applications for parametrizations in computer graphics. The three most important applications are briefly described in the following.

1.1.1 *Transfer of detail:*

One of the first times a parametrization of objects became necessary was the application of *texture mapping* in rendering (see Figure 1. As a part of the rendering of a scene, texture mapping increases the detail of the surfaces of objects by drawing a pregenerated image on the surface.

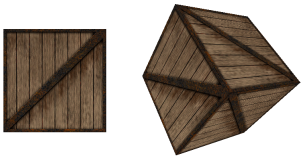

Figure 1: Texture mapping

Further improvements of this approach include *Normal Mapping*, which transfers the shading of a high quality mesh consisting of thousands of polygons to a mesh with reduced polygon count to increase the detail on the latter mesh without decreasing render speed. More approaches like this are *Bump Mapping* and *Displacement Mapping*, which – similar to *Normal Mapping* – also apply more detail onto a mesh without significant impact on the render time.

1.1.2 *Remeshing:*

Polygonal meshes are created by several methods like scanning with lasers and modeling by hand in a special software. This leads to meshes of different resolution and different surface generation techniques (e.g. triangles, quads, mixes of both). Sometimes an application only allows for a certain kind of triangulation and resolution.

This is where *Remeshing* becomes important. Remeshing parametrizes a mesh and then maps a regular and desired triangulation on the parameter domain to retriangulate the original mesh [13]. The application of subdivision on the parameter domain can also lead to good results in regard to desired mesh quality [17], [14].

1.1.3 *Correspondence:*

If two meshes should be analyzed, it is sometimes desired to link those meshes to each other by means of correspondence. The correspondence of two meshes means that the relationship between a region A on the first mesh and a region B on the second mesh is known. This can be used to transfer details from one mesh to the other.

To create this link, both meshes need to be mapped to the same representation. By using a bijective parametrization, an algorithm can map each point on the first mesh to the corresponding point on the second mesh simply by choosing the same parameter values.

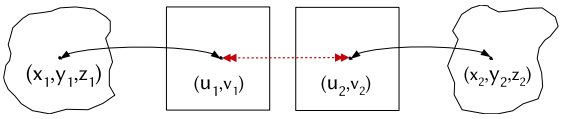

Figure 2: Correspondence between two surfaces

1.2 Related Work

The increasing requirements for parametrizations on surfaces have led to the development of different approaches with different pros and cons. The following section summarizes these attributes.

Criteria for the evaluation of parametrizations

The attributes, by which parametrizations can be measured regarding their potential application, are:

- the degree of distortion
- the aspect of bijectivity
- the limitation on certain mesh types

Distortions in the parameter domain are a direct result of parametrization and can be of different type. It has been proven by differential geometry that there is no distance-preserving parametrization for generic surfaces [8]. The distortion can be minimized, but not fully prevented.

Possible types of distortion are the distortion of distance – an irregular distribution of parameter values in one dimension – and angular distortion. Type and degree of distortion are important criteria in judging methods of parametrization. It is often dependent on the specific application which methods are more suitable than others. This leads to compromises almost every time since no parametrization is completely free of distortion.

The methods of parametrization also show different bijective behavior. This can be separated into global and local bijectivity. Global bijectivity is maintained over the whole mesh, while local bijectivity is given only for local regions on the surface. Not every parametrization leads to bijectivity of one of these kinds.

The third criterion is the limitation on certain mesh types. Some parametrizations need convex meshes, while other approaches can also use more complex meshes.

Approaches

Perhaps the oldest method of parametrization was developed by William Tutte in 1963 [27]. Tutte used *graph embedding* as the basis of the approach, which led to a bijective parametrization with distortion in distance and angular aspects. Using the same approach,

two algorithms developed by Floater show similar behavior, but at the same time reduce angular distortion [10], [11].

Eck et al. use a parametrization for remeshing at different resolutions [9]. This method is based on harmonic maps and therefore preserves angular dimensions.

DCP is a different parametrization developed by Desbrun et al. [6] and combines the already known Dirichlet-energy [20] with a new quadratic Chi-energy E_χ, which describes the inner angles of triangles. This method is not bijective in every situation, but can be used without limitations (e.g. only convex meshes).

Implementing the *Least-Squares*-method, the parametrization called LSCM preserves the orientation of each triangle and angle. It is independent of resolution but cannot guarantee bijectivity for every mesh [18].

Linear parametrizations like those mentioned above tend to create an increased distance-based distortion on meshes with sharp slopes. Using non-linear parametrizations helps to reduce these distortions. An example is MIPS [15], which divides the mesh into several linear maps. A special functional reduces the distortion map by map and creates a parametrization which is bijective and can be used without limitation.

The parametrization ABF differs from the mentioned methods, as it does no work on the vertices, but on the angles of the triangles [22]. It reduces angular distortion and shows local bijectivity. A variation of the algorithm called ABF++ increases calculation speed and stability on large meshes.

Kharevych et al. adopt a similar approach by defining circumcircles on every vertex [16]. Cutting circumcircles define the angles between vertices, which are then minimized. This approach works best on Delaunay-triangulations [5].

The introduced parametrizations are primarily based on mathematical ideas and concepts from computer science. These are the topics of differential geometry, topology and graph theory, which are combined to represent a mesh in parameter domain.

In the past, several approaches to carry concepts over from natural processes to computer science were successful and lead to groundbreaking and novel methods, for example genetic algorithms, routing algorithms and graph algorithms [19, 7, 1]. Other approaches that use similar diffusion-based processes called *reaction-diffusion* create textures automatically [26, 28].

In the topic of parametrizations, this transfer is yet to be made.

1.3 Motivation

This work applies insights from biology and chemistry to the problem of consistent surface parametrization. Diffusion helps cellular development and differentiation by giving hints on the position in the organism to individual cells. The aim of this paper is to show that the natural processes of diffusion can help in the development of algorithms for parametrizations.

The proposed method guarantees local bijectivity on convex mesh parts, which are constructed from the original mesh. The parametrization retains the proportions of distance of the mesh parts. Because the mesh is viewed as a graph, already available and highly optimized algorithms from graph theory can be used to achieve an efficient and stable parametrization.

In the following, the course of the paper is to present an introduction to physical and biological diffusion, followed by evaluation of algorithms which can be appropriate for serving as a basis for further development. Later on, the modifications to the chosen algorithm are explained. The paper concludes with a discussion and and an outlook on possibilities for further research.

2 DIFFUSION

The basis of our approach originates from morphogenesis. Morphogenesis controls the differentiation and development of cells in multicellular organisms to organs and extremities, and it produces patterns on skin, fur or shells of animals. Well-directed flow and diffusion of activators through the tissue leads to specific development of the stem cells depending on the structure they are going to form.

The gradients of concentrations of specific substances form a metric in the tissue and on the surface. On different shapes of the same type, these gradients and thus the induced metric are similar, so they describe objects regardless of position in space, scale, orientation and resolution of the mesh.

2.1 Physical diffusion

Diffusion occures whenever particles – for example atoms, molecules or charge carriers – are aggregated in a carrier medium and there exists a concentration gradient between these particles. The reason for this movement is called pedesis or Brownian motion [3]. The atoms and molecules are in an undirected motion, depending on temperature: Due to collisions, their direction changes randomly over time. If there is a concentration gradient, there will be an overall net motion along this direction, which forms the diffusion process described by the following equation:

$$\frac{\delta u}{\delta t} = D \cdot \Delta u = D \cdot \operatorname{div}(\operatorname{grad} u) \tag{1}$$

2.2 Diffusion in developmental biology

Diffusion plays an important role in biological processes. By diffusion through membranes cells are supplied with nutrients and metabolic waste products are removed. Patterns on the skin or fur are also controlled by diffusion of messengers.

This is described in detail by Gierer and Meinhardt, who present a model which depicts the formation of structures in biological cell structures [12]. The interaction of two messengers – the so called activator/inhibitor pair – play a major role in position dependent pattern formation. Both agents diffuse through the tissue and lead to different concentrations depending on position. This process – called morphogenesis – has been described by Turing [25] and was specified further by Gierer and Meinhardt. A gradient of concentration values runs between the cells producing the activator and between the cells producing the inhibitor. The resulting patterns of cell differentiation are aligned along this axis.

One special form of this mechanism is observable in the polyps of the genus *Hydra*. Their heads and feet are shaped depending on the concentration of the activator [12]. Gierer and Meinhardt showed that the diffusion of the morphogenes in Hydra takes a gradual course. This gradient of source density gives orientation in the tissue over the longitudinal axis of the animal's body. The gradient serves as an indicator of relative position within the animal.

2.3 Formulation of the idea

Several criteria influence the parametrization and construction of correspondence, depending on the geometry and quality of the mesh. These are:

- Scale of the mesh
- Position of the mesh relative to the coordinate system
- Rotation of the mesh
- Internal geometry of the mesh (for correspondence a rough similarity is sufficient)

There may be considerable variation in these parameters and properties, which makes many surface analysis problems challenging. In many cases, surface parametrizations help to find more simple and efficient solutions.

Our diffusion-based parametrization is inspired by the gradient of source density, which defines a polar orientation. We will ignore most of the details of the biological mechanism and develop the idea towards parametrization of surface patches.

2.4 Diffusion as a description of surfaces

To successfully compute the diffusion on the surface, the physical and biological model needs to be simplified. The first simplification is to consider the surface as a two-dimensional carrier medium and discarding any effects of diffusion into depth. Both complexity and computational costs are decreased due to the fact that only two dimensions are considered.

We assume a dynamic equilibrium where a substance is produced in a source point or a line-shaped set of sources, travels along the surface due to diffusion and is diluted, washed away, absorbed or deactivated chemically over time. Therefore, in our simplified model, the concentration of the substance is proportional to the distance from the source.

To obtain relative distances on the surface, independent of the scale of the object, we introduce pairs of antagonist substances that have concentrations d_1 and d_2 and are defined on each vertex of the mesh. To calculate the relative distance between each of the reference points, the following equation is used:

$$\hat{d} = \frac{d_1}{d_1 + d_2} \tag{2}$$

This formula makes sure that $\hat{d}$ takes the value 0 on the source points of the first substance and value 1 on the source points of the second substance. It then increases smoothly in between. The possible values of distance $\hat{d}$ are clamped to the interval $[0,1]$.

This operation is the necessary first step in parametrizing the whole mesh. To get an even distribution of the parameters, sets of points form the source points. This leads to a smooth and near parallel distribution of the isolines on the surface (see Figure 3).

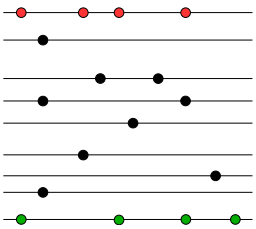

Figure 3: An ideal distribution of isolines of $\hat{d}$ is obtained if sources for d_1 (*red*) and d_2 (*green*) are not only single points, but sets of points along the edges

For a unique identification of each point a second axis in parameter distribution is needed (see Figure 4). If this second axis is orthogonal to the first direction of diffusion, two linearly independent parameters result and are able to describe every point on the surface by a pair of the interval $[0,1]^2$. We call these two axes the gradients of diffusion ∇u and ∇v. The parameters are called u and v and form pairs (u_i, v_i) for each point P_i on the surface.

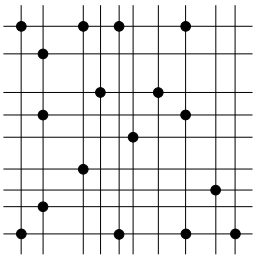

Figure 4: A 2D parametrization (u,v) is obtained by two separate diffusion processes, one along the horizontal and one along the vertical direction, with two variables $\hat{d}_u$, $\hat{d}_v$

2.5 Organization of the parameter values in a kd-tree

By simply saving the parameter values to each point, one can easily retrieve the tuple (u,v) for every given point. This is done in a linear array, which can be accessed in $\mathcal{O}(1)$. For an adequate use the other way around is also important, therefore it is necessary to have an efficient data structure for finding a point corresponding to a given value. Different data structures exist, which help to find a given value in a set of points quickly without comparing each point to the search term. In our work we chose the kd-tree [2], which scales well with a certain number of points. The complexity of searching for a value lies in $\mathcal{O}(\log n)$, where n is the number of vertices of the mesh.

2.6 Viewing surfaces as a discrete point set

Polygonal surfaces are usually intended to be approximate models of smooth surfaces such as geometrical primitives (spheres), vehicles, human faces or characters. The higher the number of polygons is, the better the approximation of the surface can be. Because of this approximation it is difficult to model the process of diffusion, as it runs through a continuous substance. The triangulated surface has no continuous nature, but instead has gaps between discrete points.

The algorithm to solve this problem needs to simulate the continuous progression of the diffusion along the surface with special regard to the distance between the points. One algorithm with these characteristic is Dijkstra's algorithm for shortest paths. With a given starting point, Dijkstra's algorithm assigns each following point the shortest distance to the starting point. It propagates the distance values iteratively in a process that is similar to diffusion.

We modified the algorithm in such a way that it takes a complete set of points to start instead of a single point. After a successful pass, each distance d_1 between point and start is measured. A second pass in the reverse direction leads to the second value d_2 of diffusion, which is used in order to determine the value of relative distance $\hat{d}$ in the calculation.

We deliberately decided to propagate the values along edges. Alternatively one could choose the geodesics, i.e. direct paths along the surface which are not constrained to points in the set. Geodesics, however, do not respect the structure of the mesh. Interpolation is required to calculate the geodesic path through the polygons. Because of this computational increase the idea of geodesics had been discarded.

2.7 Segmentation

Whether global bijectivity of the parametrization can be achieved at all depends on the topology of a mesh. A mesh which is topologically equivalent to a disc can be mapped to a parameter domain in the plane. These simple meshes do not need to be partitioned and easily achieve global bijectivity. Other meshes with more complex topology cannot be mapped to a plane and therefore violate global bijectivity.

Consider the simple example of a sphere. From topology, we know that we cannot find a homeomorphism from parameter space $[0,1]^2$ to the sphere, so we expect singularities and ambiguities if we apply our algorithm to the entire sphere. For the first parameter u and its relative distance function $\hat{d}$, consider a line from pole to pole (half of a great circle) as a source for d_1, and the other half of the same great circle as source for d_2. This defines a diffusion which spreads over both hemispheres. However, both poles will be singularities because they are sources of both d_1 and d_2. On the other hand, if we use two opposite points as sources for d_1 and d_2, respectively, we violate the uniqueness criterion as soon as both parameters u and v are computed: As shown in Figure 5, two arbitrary isolines of each diffusion cintersect twice, so these intersection points obtain the same parameter tuple (u,v).

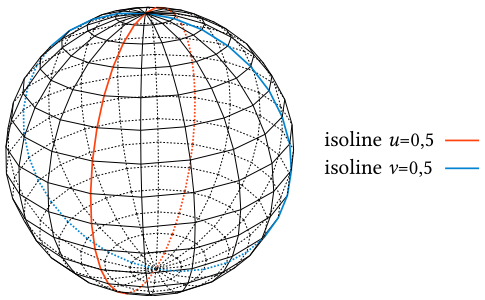

Figure 5: Ambiguity on certain points on meshes that are not homeomorphic to a disc.

The solution to this undesired effect is the segmentation of the mesh into hemispheres. Two passes parametrize each hemisphere separately. This guarantees local bijecitivity on each partial mesh. If the segmentation of the mesh is adequate, the whole mesh can be partitioned in topological discs with guaranteed uniqueness.

Global bijectivity is not given and therefore the direct mapping of (u,v)-tuples from two meshes is not possible. A mechanism is required to assign regions of different meshes to the same or similar areas, which introduces a new level of hierarchy.

The segmentation must lead to reproducible regions. Only then is the parametrization useful, particularly in regard to the correspondence between meshes. Several criteria define the usefulness of the segmentation. Besides the reproducibility these are the number of segmented regions, similar segmentation on similar meshes and interaction of the user to mark logical areas.

To achieve a $1:1$ connectivity between meshes, the segmentation must separate both mesh A and mesh B to each set of regions $r_{A_0}, r_{A_1}, .., r_{A_n}$ and $r_{B_0}, r_{B_1}, .., r_{B_n}$. Each vertex must be assigned to a region. Both sets of regions can be related to each other if both meshes have the same number of regions. To create a meaningful correspondence, the regions must map to similar parts on the meshes in the first place. This means that the segmentation must follow a fixed orientation over the mesh, which also adds to the reproducibility.

Meshes from different classes of objects should segment to semantic similarities of the same type. The algorithm should react on the user's input, which denominates the points of interest regarding the logical areas. This simplifies the finding of correspondence between meshes and enhances quality.

Existing approaches for reproducible segmentations lack the possibilities to work on a predefined number of regions and only respect the user's input marginally. Therefore, we chose to implement a simple, but efficient approach that gives the user the freedom of choice in segmenting the mesh (see Figure 6).

The idea is that the user paints the regions on the surface manually with an interactive tool. This defines the segments of the patchwise parametrization. Moreover, and this is the core idea of the user interface, the boundaries of the segments are a natural choice of source-sets for the diffusion algorithm. Depending on the label of the adjacent region, a boundary vertex will be a source point of d_1 or d_2 for parameter u or v, respectively. The goal of the segmentation is to produce patches with closed and connected boundaries that can be divided into 4 parts, similar to a rectangle.

Those parts which form the ends of extremities (e.g. hands and feet in Figure 6) of the depicted mesh need special treatment. They could be seen as single regions, because they fulfill the topological criterion. The problem of this naive approach is the nature of their edges. These edges overly stretch during the mapping to the plane (as seen in 7). This leads to increased angular distortion, with higher extremes in parts with longer stretched edges.

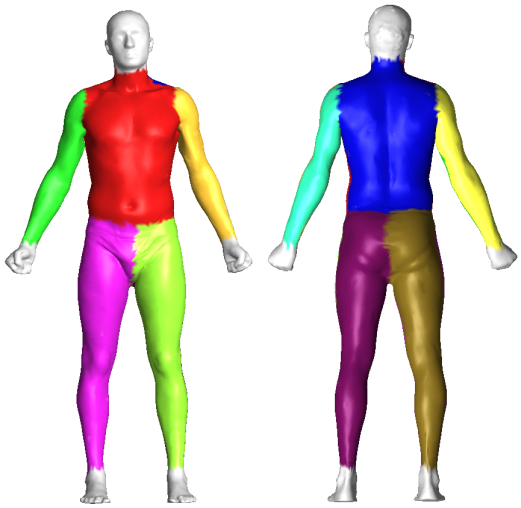

Figure 6: Segmentation example

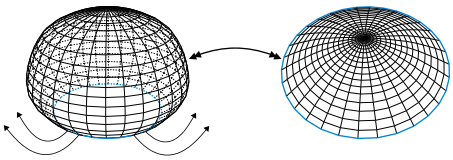

Figure 7: Edge stretching and angular distortion

Separating such meshes into two halfs helps preventing these unfavorable results. By splitting the meshes in question before mapping them to the plane a lower distortion is achieved, as the change of length of the edges is reduced.

After choosing the regions carefully to avoid bad segmentation, the next step is the selection of starting and ending edges. As mentioned before, the process of diffusion needs the four edges of the surface to be parametrized. The edges resemble the sets of starting and ending points for the modified Dijkstra. Because the diffusion is not limited to four-sided surfaces, it must be possible to choose four logical edges on arbitrary convex surfaces, leading to edges $E_{u_1}, E_{u_2}, E_{v_1}, E_{v_2}$. The finding of suitable edges can be based on the segmentation into regions, as the borders between two regions already form suitable edges.

In the case that there is only one continuous border between two regions, the edge is simply the overlapping part of both regions. In order to not include a set of points in two edges at once, the method can choose to include or exclude these vertices. This discrimination is necessary if a point takes part in both regions as seen in Figure 8.

A user who is familiar with the process of segmentation can handle it very quickly. The segmentation in Figure 6 did not take longer than 10 minutes with the interactive, mouse-based tool.

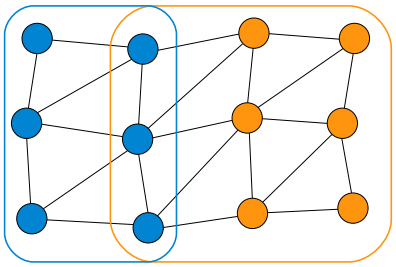

Figure 8: Overlapping of vertices in two regions

2.8 Process of diffusion

The diffusion processes each region on the mesh separately. This is done by separation and transformation of the whole mesh into partial graphs. As all regions are combined to the original mesh, the partial graphs of each parametrized region form the whole graph that represents the mesh.

The construction of partial graphs shifts the process of diffusion away from the original mesh. Each partial graph leads to an own uv-map, which in turn is independent to the uv-maps of other partial graphs, thus solving the global bijectivity problem. But as shown later, the local bijectivity on each partial graph is given.

The modified Dijkstra algorithm is initialized just like the original algorithm. Each vertex that does not belong to the starting points is assigned $d_i = \infty$. Unlike the original Dijkstra's algorithm, which starts from a single vertex, it is here the initial starting (source) set that is initialized as $d_{i_S} = 0$. After inserting each point of the whole partial graph into a *min-heap*, the algorithm can start.

Taking the first element out of the min-heap gives the point with the minimal value of diffusion $d_{i_{min}}$. In the first iterations this will be the whole set of starting points, but later on this will govern all points in the whole partial graph. Each extracted point is treated in exactly the same way. Every edge of this vertex will be relaxed, using the following equation 3.

$$d_B = \begin{cases} d_A + w(A,B), & \text{if } d_A + w(A,B) < d_B \\ d_B, & \text{otherwise} \end{cases} \tag{3}$$

The function $w(A,B)$ is the weighting function which returns the length of the edge between vertices A and B. Vertex A is the vertex that is currently being processed and vertex B is the target of the edge that the algorithm actually relaxing. Therefore, A stays the same for the vertex that is processed and B changes to each of the adjacent vertices during relaxation of the adjacent edges.

This relaxation leads to updated values of diffusion in each adjacent vertex. By updating the value, the process of diffusion iteratively spreads over the whole graph, until reaching its end (see Figure 9 for an illustration). Since the relaxation affects the value of diffusion, which in turn is the key to the min-heap, the residing vertices are sorted after each update to reflect the correct sequence of increasing key values.

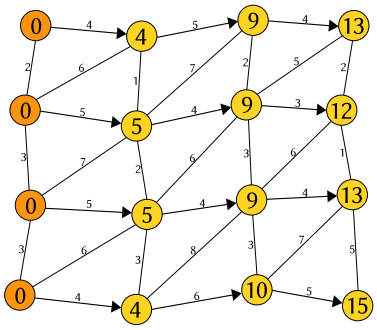

Figure 9: The process of diffusion illustrated with a simplified mesh.

As soon as the min-heap is empty, the process of diffusion finishes. At this point, each vertex holds its distance to the set of starting points. This distance is the path to the closest vertex in the set of starting points. This algorithm is run twice, once for d_1 and once for d_2. From these, $\hat{d}$ is computed, which is equivalent to the parameter u. With another pair of source sets, the parameter v is computed in the same way.

After the completion of the second two-pass, each vertex holds the values d_{u_1}, d_{u_2}, d_{v_1} and d_{v_2}. These values are the direct values of distance to their corresponding starting edge. By inserting these into the two equations 4, we normalize the values into the interval $[0,1]$:

$$\hat{d_u} = \frac{d_{u_1}}{d_{u_1} + d_{u_2}} \qquad \hat{d_v} = \frac{d_{v_1}}{d_{v_1} + d_{v_2}} \tag{4}$$

These values of diffusion $\hat{d_u}$ and $\hat{d_v}$ are then saved in the kd-tree. By combining the kd-tree with a simple linear array, we get a uv-map that supports fast retrieval of (u,v)-tuples for a given vertex and also grants an efficient search for a vertex, which lies as close as possible to a given (u,v).

3 RESULTS

We implemented the modified Dijkstra's algorithm. We parametrized a scan from a human and a cow and manually developed a segmentation of the meshes. The human mesh (see Figure 10) containes about 11k vertices with 3 to 5 edges adjacent to each vertex. The cow (see Figure 11) is made up of ca. 3.1k vertices, also connected to neighbors with 3 to 5 edges. The parametrizations were done in 68 ms for the cow and 224 ms for the human mesh on a Pentium i7-2600 3.40 GHz.

The parametrization of the whole mesh took place as a sequence of parametrizations on the list of partial

meshes after the segmentation. Like the original algorithm by Dijkstra our modified approach has a complexity lying in $\mathcal{O}(|E|log|V|)$, where $|E|$ is the number of edges in the mesh and $|V|$ is the number of vertices in the mesh. We then applied a grid on the mesh, using the parametrized values as texture coordinates (see Figure 10).

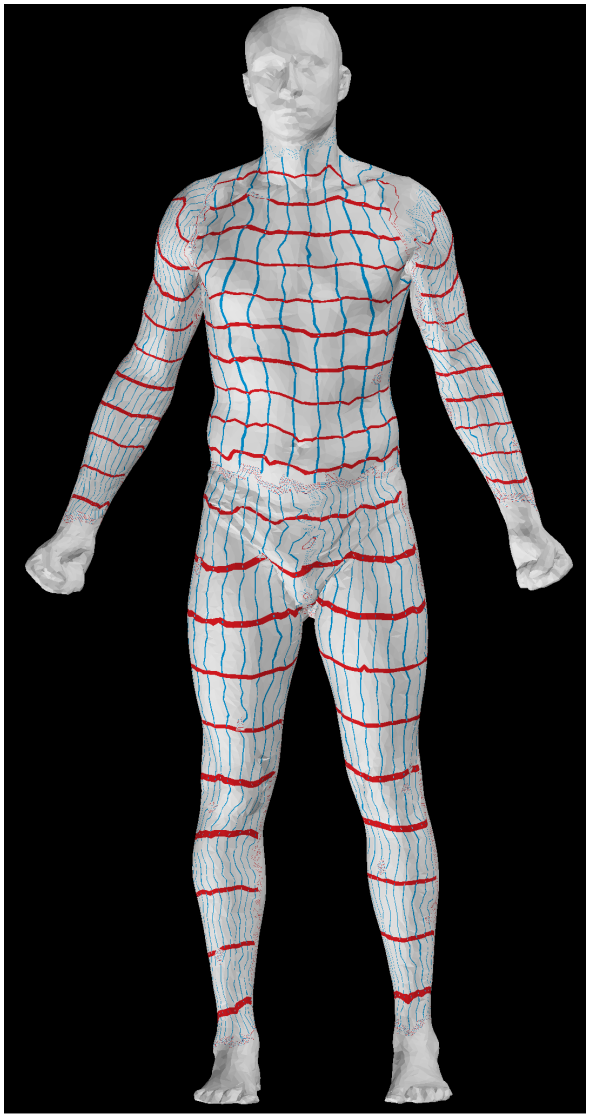

Figure 10: The parametrization visualized by a grid

From these pictures we can judge the quality of the parametrization. The distribution depends on the mesh structure and it is clearly visible that this leads to an increased jitter of parameter values.

On the large areas like chest and back of the human mesh (Figure 10) there is less distortion than on structurally smaller areas. These areas (e.g. the arms and legs) show a ratio between horizontal and vertical distribution which is not aspect-preserving.

The different length of the edges in a mesh is directly visible in the grid. For example, the edges on the side of the cow are longer than the edges on other parts of the mesh. The apparent gap between the isolines shows that the edges on this side are almost double in length. Right behind the right shoulder of the cow there is a loop that continues on the bottom side of the mesh. This loop is part of the isoline that moved downwards due to the larger distance between the vertices on the side.

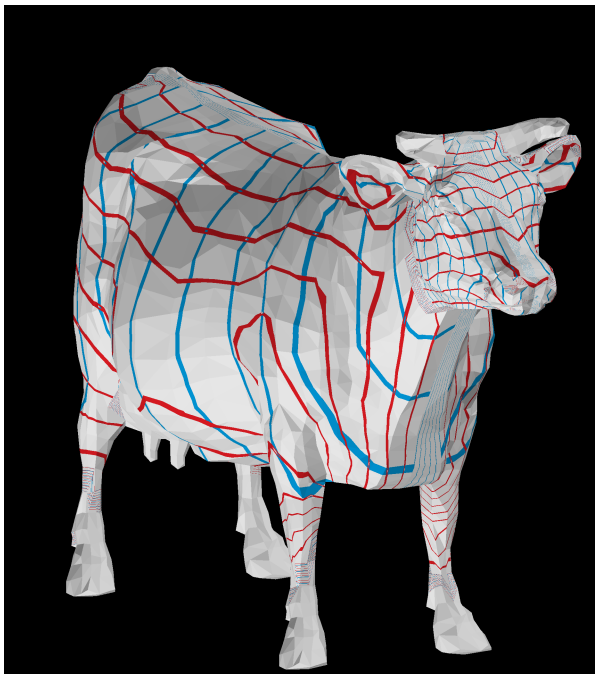

Figure 11: Parametrization of another object

An important aspect is the examination of the boundaries between the regions. They show irregularities which probably stem from the different parameter values that overlap on these vertices. Regardless of which set of (u,v)-values is chosen, there will always be a residual distortion, even though the algorithm contains mechanisms that guarantee a continuous parametrization across boundaries of segments: Consider a segment pattern that looks like a distorted rectangular grid on the 3D surface, and a boundary segment S between two patches A and B. Let the parameter u be 1.0 along the boundary in A, and 0.0 in B, while v varies from 0.0 to 1.0 continuously. Then v is determined by the boundaries (source sets) adjacent to S in A and in B. If these adjacent boundaries in A and B fit together as one continuous curve, like in a rectangular grid, the parameter v on S will be the same in A and B, so the patch-wise global parametrization is continuous across patch boundaries.

3.1 Comparison

Based on Alla Sheffer's work [24], we compared our parametrization with other approaches. Table 1 shows various parametrizations [24]. We added our parametrization in the last line.

The images show clearly that the algorithm does hardly prevent distortion. Especially in parts with complex curvature (e.g. the head and breast of the cow) angular distortion increases. Deficiencies on the mesh (e.g. different edge lengths, jumps, doubled vertices) further increase the distortion. Adding the evaluation of an error metric before the parametrization can value the quality that can be expected.

Especially the uniform, harmonic and mean-value weighted parametrizations [27, 9, 11] show similarities between different patches on the surface, e.g. the breast region of the cow. Algorithms like DCP and LSCM have similar problems preserving areas and distances, but are better in preserving angles than our algorithm. This is because they sacrifice the distance preservation in favor of minimized angular distortion. Our algorithm has problems with angular distortion in specific regions.

Singularities like the ears of the cow show cycles that are direct result of the diffusion process. Other parametrizations handle such singularities more graceful. Circle patterns [16] and stretch minimizing [21] approaches excel at these parts.

4 CONCLUSION AND FUTURE WORK

In this paper we showed the development of a new approach in parametrization, inspired from nature. Our approach can lead to a patchwise bijective parametrization, which concentrates on local bijectivity. User interaction makes global bijectivity possible. The main targeted application is the creation of correspondence between two objects. Our approach simplifies this by using a combination of a *kd*-tree and a linear array named *uv*-map, which stores tuples of (u,v) and provides fast and efficient two way searches.

The approach is limited by the heavy dependency on the users input. The segmentation process is entirely controlled by the user as is the assignment of regions between two parametrized meshes. Here lies further potential for improvement. A fully automated segmentation method, which leads to reproducible partition and comparable results between different, but similar objects, would enhance the application of this approach. We will investigate current and future segmentation algorithms for suitability.

Our limitation to convex patches is also subject to possible research, as other parametrizations do not depend on convex meshes. By surpassing this limitation it could be possible to completely omit the segmentation and any user input. We will investigate this, too.

ACKNOWLEDGEMENTS

We want to thank the anonymous reviewers for their valuable comments.

5 REFERENCES

[1] Yehuda Afek, Noga Alon, Omer Barad, Eran Hornstein, Naama Barkai, and Ziv Bar-Joseph. A biological solution to a fundamental distributed computing problem. *Science*, Vol. 331, No. 6014:183–185, 2011.

[2] Jon Louis Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18:509–517, 1975.

[3] Robert Brown. A brief account of microscopical observations made in the months of june, july and august, 1827, on the particles contained in the pollen of plants; and on the general existence of active molecules in organic and inorganic bodies. *Philosophical Magazine*, Vol. 2:161–173, 1828.

[4] Patrick Degener, Jan Meseth, and Reinhard Klein. An adaptable surface parameterization method. In *12th International Meshing Roundtable*, 2003.

[5] Boris N. Delaunay. Sur la sphère vide. *Bulletin of Academy of Sciences of the USSR*, 7:793–800, 1934.

[6] Mathieu Desbrun, Mark Meyer, and Pierre Alliez. Intrinsic parameterizations of surface meshes. *Computer Graphics Forum*, Vol. 21:209–218, 2002.

[7] Gianni DiCaro and Marco Dorigo. Antnet: Distributed stigmergetic control for communications networks. *Journal of Artificial Intelligence Research*, Vol. 9:317–365, 1998.

[8] Manfredo P. do Carmo. *Differential geometry of curves and surfaces*. Prentice Hall, 1976.

[9] Matthias Eck, Tony DeRose, Tom Duchamp, Hugues Hoppe, Michael Lounsbery, and Werner Stuetzle. Multiresolution analysis of arbitrary meshes. In *ACM SIGGRAPH*, 1995.

[10] Michael S. Floater. Parameterization and smooth approximation of surface triangulations. *Computer Aided Geometric Design*, Vol. 14, No. 3:231–250, 1997.

[11] Michael S. Floater. Mean value coordinates. *Computer Aided Geometric Design*, Vol. 20, No. 1:19–27, 2003.

[12] Alfred Gierer and Hans Meinhardt. A theory of biological pattern formation. *Kybernetik*, 12:30–39, 1972.

[13] Xianfeng Gu, Steven J. Gortler, and Hugues Hoppe. Geometry images. In *ACM SIGGRAPH*, 2002.

[14] Igor Guskov, Kiril Vidimce, Wim Sweldens, and Peter Schröder. Normal meshes. In *ACM SIGGRAPH*, 2000.

[15] Kai Hormann and Günther Greiner. *MIPS: An Efficient Global Parameterization Method*. Vanderbilt University Press, 2000.

[16] Liliya Kharevych, Boris Springborn, and Peter Schröder. Discrete conformal mappings via circle patterns. *ACM Transactions on Graphics*, Vol. 25, No. 2:412–438, 2006.

[17] Aaron Lee, Hugues Hoppe, and Henry Moreton. Displaced subdivision surfaces. In *ACM SIGGRAPH*, 2000.

[18] Bruno Lévy, Sylvain Petitjean, Nicolas Ray, and Jérome Maillot. Least squares conformal maps for automatic texture atlas generation. In *ACM SIGGRAPH*, 2002.

[19] Melanie Mitchell. *An Introduction to Genetic Algorithms*. MIT Press, Cambridge, 1996.

[20] Ulrich Pinkall and Konrad Polthier. Computing discrete minimal surfaces and their conjugates. *Experimental Math*, Vol. 2, No. 1:15–36, 1993.

[21] Pedro V. Sander, John Snyder, Steven J. Gortler, and Hugues Hoppe. Texture mapping progressive meshes. In *ACM SIGGRAPH*, 2001.

[22] Alla Sheffer and Eric de Sturler. Surface parameterization for meshing by triangulation flattening. In *9th International Meshing Round Table Conference*, 2000.

[23] Alla Sheffer, Bruno Lévy, Maxim Mogilnitsky, and Alexander Bogomyakov. Abf++: Fast and robust angle based flattening. *ACM Transactions on Graphics*, Vol. 24, No. 2:311–330, 2005.

Name	**Distortion minimized**	**Bijektivity**	**Boundary**	**Source**
Uniform	none	yes	convex	[27]
Harmonic	angular	no	convex	[9]
Shape preserving	angular	yes	convex	[10]
Mean-value	angular	yes	convex	[11]
DCP	angular	no	free	[6]
LSCM	angular	no	free	[18]
MIPS	angular	yes	free	[15]
ABF/ABF++	angular	yes (only local)	free	[22]/[23]
Circle patterns	angular	yes (only local)	free	[16]
Stretch minimizing	distance	yes	free	[21]
MDS	distance	no	free	[29]
Adaptable surface	area	yes	free	[4]
Our approach:				
Diffusion-based	to a degree:			
distance	yes (only local)	convex		

Table 1: Comparison of parametrizations with our approach

[24] Alla Sheffer, Emil Praun, and Kenneth Rose. Mesh parameterization methods and their applications. *Foundations and Trends in Computer Graphics and Vision*, Vol. 2, No. 2:105–171, 2006.

[25] Alan Mathison Turing. The chemical basis of morphogenesis. *Philosophical Transactions of the Royal Society of London*, Series B, Vol. 237, No. 641:37–72, 1952.

[26] Greg Turk. Generating textures on arbitrary surfaces using reaction-diffusion. *SIGGRAPH Comput. Graph.*, 25:289–298, 1991.

[27] William T. Tutte. How to draw a graph. *London Mathematical Society*, Vol. 13:743–768, 1963.

[28] Andrew Witkin and Michael Kass. Reaction-diffusion textures. In *Computer Graphics*, pages 299–308, 1991.

[29] Gil Zigelman, Ron Kimmel, and Nahum Kiryati. Texture mapping using surface flattening via multi-dimensional scaling. *IEEE Transactions on Visualization and Computer Graphics*, Vol. 8, No. 2:198–207, 2002.