Geotextures: A Multi-Source Geodesic Distance Field Approach for Procedural Texturing of Complex Meshes

Guilherme N. Oliveira*, Rafael P. Torchelsen*, João L.D. Comba*, Marcelo Walter*, Rui Bastos†

*Instituto de Informática - UFRGS.

e-mail: {gnoliveira,comba,marcelo.walter}@inf.ufrgs.br, {rafael.torchelsen@gmail.com}

†NVIDIA Corporation.

e-mail: {rbastos@nvidia.com}

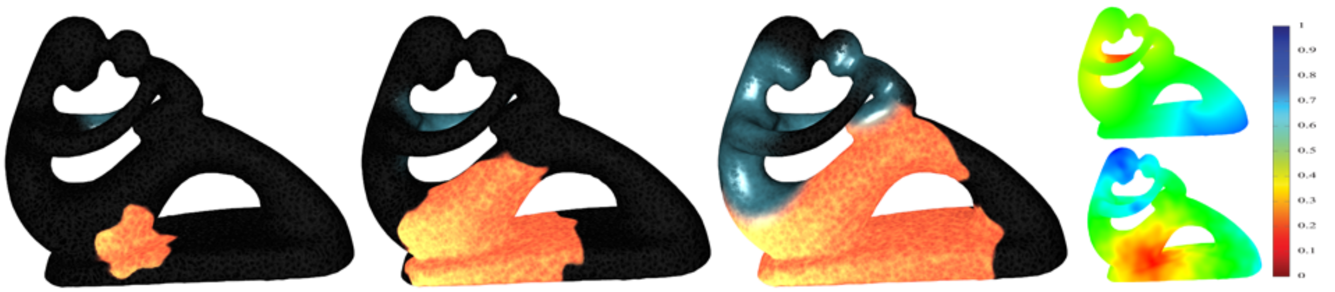


Figure 1. Continuous and surface conforming procedural water and lava-like materials are propagated over a complex surface from two distinct sources. In the right, the geodesic distance fields defined by the two propagation sources are color-mapped over the surface according to the color scale on the right.

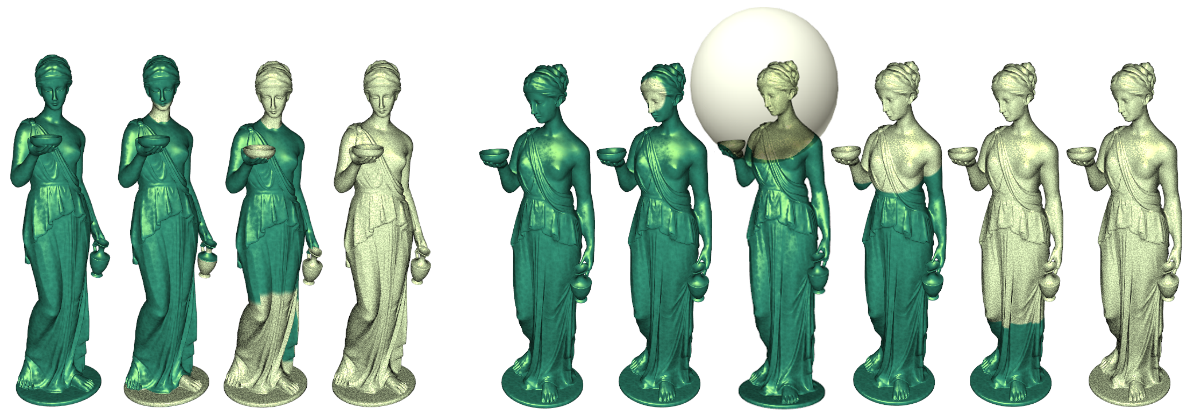
***Abstract*—Texture mapping is an important technique to add visual detail to geometric models. As an alternative to traditional image-based texture mapping, procedural textures are described by a function, with interesting properties such as compact representation, resolution independency and parametric adjustment of the visual appearance. Procedural textures are usually defined in the 2D texture space, making the result dependent on texture mapping coordinates assigned to the model, or in the 3D object space, implying in no correlation with the surface model. In this work we introduce *GeoTextures*, an approach that uses the geodesic distance, defined from multiple sources over the model, as a parameter that is taken into account by time-varying procedural textures. The use of geodesic distances allows the process to be both independent from the mapping of texturing coordinates and also conforming with the model surface. We validate the proposal by applying real-time procedural textures in complex surfaces.**

***Keywords*-Procedural texturing; distance fields; geodesics;**

I. INTRODUCTION

Modeling geometric models with detailed visual attributes is an important problem in Computer Graphics and related fields. Texture mapping is a viable alternative since it allows adding detail to often simplified geometric models. In texture mapping, one can choose among several alternatives, such as image-based and procedural approaches. Procedural textures were introduced as an alternative to reduce memory usage of image-based approaches. Since current computer systems often have plenty of memory available, in some situations it becomes questionable whether the memory savings due to the compact representation of procedural textures compensates the difficulty of creating a function that reproduces the desired appearance. This is certainly a valid point for static textures or predefined animations, but for dynamic scenes it is clear that procedural texturing is still very important since image-based approaches imply in multiple images for each timestep. In addition, having an artist to create the textures for each frame is time consuming, while having a proper procedural model can reproduce an animation by interpolating parameters in real-time. Following the saying that an image is worth a thousand words, a model is then worth a thousand or more images, and particularly for texture mapping for animated sequences the model offered by procedural texture is much more rich.

There are, however, limitations on the current application of procedural texturing. For instance, the texture mapping space used is closely related to the results obtained. For example, in 2D the interpolation along the surface of a model conforms with the texture coordinates assigned to each mesh vertex. For the resulting texture to show the desired appearance and avoid discontinuities in the appearance, the texture coordinates must be assigned consistently. In Figure 2(a) we illustrate a problem when texture coordinates are not properly assigned. In this example, a texture resembling sand is defined over a mesh as a function of the distance to a given point (say the point with texture coordinates $(0,0)$). A sequence of four images from left to right displays the resulting texture when the distance threshold is set



(a) (b)

Figure 2. (a) Texturing using the 2D texture mapping space. The sand-like feature should propagate from a single point, but instead appears on different spots on the surface due to the mapping of replicated texture coordinates. (b) Texturing using the 3D object space. The sand-like feature propagates from the head along the surface, however it jumps to the bowl in the right hand before passing by the right elbow. Notice the spherical nature of the propagation as highlighted in the third image with the forehead centered sphere.

to $0.0, 0.12, 0.5$ and 1.0. Notice that in the second image the sand has started to spread from four distinct points: the neck, the floor, and two distinct places in the vase. This happens because texture coordinates are replicated on different vertices of the surface. In the third image it can be seem that features did not spread uniformly in the lower part of the dress, which show that texture coordinates were not consistently defined. An even worst problem happens when texture coordinates are not defined. A consistent texture mapping would be a global mapping over the entire surface, however such approach results in high distortion level for complex surfaces. To avoid distortion, one can break the mesh into several charts and apply separate mappings [1].

On the other hand, in 3D the vertex coordinates are used as texture coordinates. This approach is called solid texturing, and is suitable for cases where the surface has to be textured as if it was carved from some solid material, such as a chair carved from a single wood trunk. We refer the interested reader to [2] and [3] for more about the subject. Features defined in 3D are independent from the 2D texture mapping coordinates and have no correlation with the mesh surface. Even if the function is continuous in the 3D object space the feature will not follow the surface and for complex surfaces will show incoherent behavior. In figure 2(b) we illustrate the 3D object space solid texturing approach. A given point is chosen in 3D as the source of the propagation. In this case this point lies over the forehead of the model, and the sand feature defined on the area where the distance to the source is less than a given threshold. As the threshold increases, the feature seems to spread uniformly along the surface, but in the third image the feature "jumps" from the body to the bowl before even reaching the right elbow, revealing the spherical nature of the propagation front.

In this work we present *GeoTextures*, an approach that extends procedural texturing to take into account the geodesic distance from multiple sources, and applies it to define features on time-varying procedural textures over complex surfaces (Figure 1). We show that the use of geodesic distances as parameter solves the problems pointed previously, showing surface-conforming behavior while introducing no discontinuity such as the ones discussed in the 2D texture mapping case. We demonstrate with results how it improves the controllability of the procedural models. The remaining sections are organized as follows. First we review some related works regarding geodesic distances calculation, procedural texturing and animation. In section III the method implemented for distance computation is detailed justifying relevant decisions and exposing others possibilities and tradeoffs. Our scheme for application of the geodesic distances on procedural texturing is then presented in section IV. The results are shown with some procedural textures examples and we conclude the paper with directions for future researches.

II. Related Work

The appearance of certain materials have a strong relation with the topology of the surface on which they are applied. In such cases, taking into account surface attributes in the texturing process allows for more appealing results. In [4], Walter et al. modeled the creation of different patterns found on mammalian coats as well as its behavior as the body grows. Xu et al. [5] applied surfaces salient feature curves to texture synthesis resulting in shape-revealing textures which reinforce or even highlights the shape essential characteristics. Distance is usually an important parameter for texturing. For example, feature propagation is an example of such scheme as shown in [6] where a distance field is used to define the water flow on rivers. In 2D the Euclidian distance

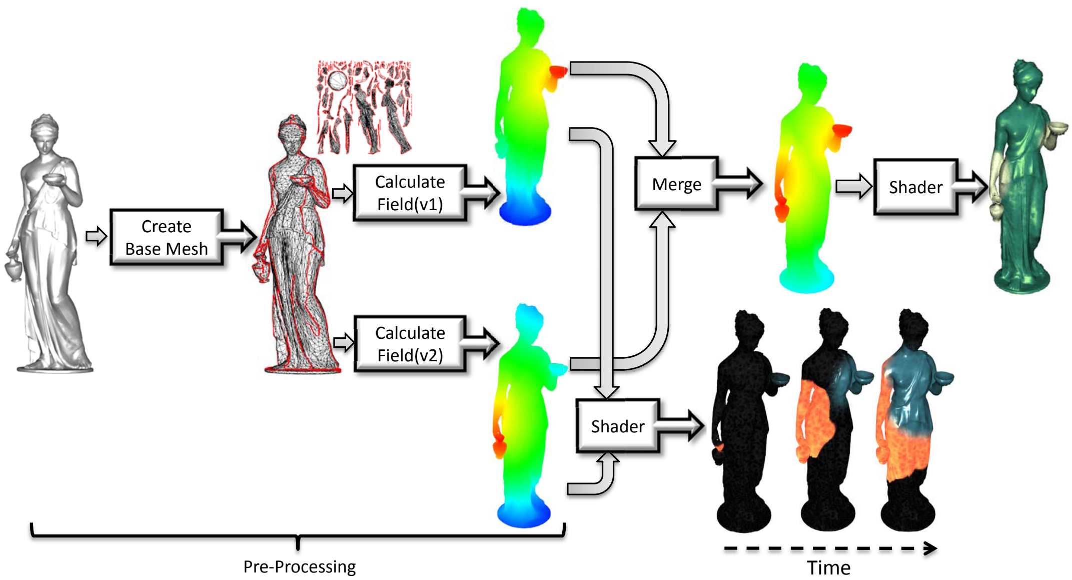


Figure 3. GeoTextures Architecture. The input mesh is processed by the algorithm described in [1] to partition into charts, which are assigned a 2D parameterization that guides a simplification of each chart. Distance field computation is performed over the base mesh. In this example, two distance fields are computed. For single material, both fields are combined before procedural texturing is applied (sand example). For multi-materials (lava and water) both distance fields are input to the rendering shader. Although only two distance fields are used in the diagram, the method imposes no restriction in the number of distance fields and materials used.

is enough, but for 3D surfaces, inconsistencies will show up as the complexity of the surface increases. Hegeman at al. [7] presented an approach for modeling relief conforming fluids flow over surfaces limited to genus zero. Stam [8] solved the flow of fluids over surfaces of arbitrary topology, but restricted to Catmull-Clarck surfaces [9].

In [1] Torchelsen et al. introduced a scheme for faster computation of geodesic distance fields over complex surfaces exploiting the simplification of charts and calculations on the parametric domain. Later, the same approach was successfully applied to calculate geodesic distance fields to guide the navigation of agents over complex surfaces [10]. While the method for geodesic distance computation used in Torchelsen et al. works [1] and [10] was the one in [11], the scheme can be adapted to use any other method. Bommes and Kobbelt [12] extended [11] to the computation of geodesic distance fields taking polygonal curves as sources instead of only points. Weber et al. presented a parallel method in [13] that also uses the parametric domain and reaches an even faster computation of geodesic distance fields, however it depends on a regular connectivity of the surface mesh and leads to less accurate results.

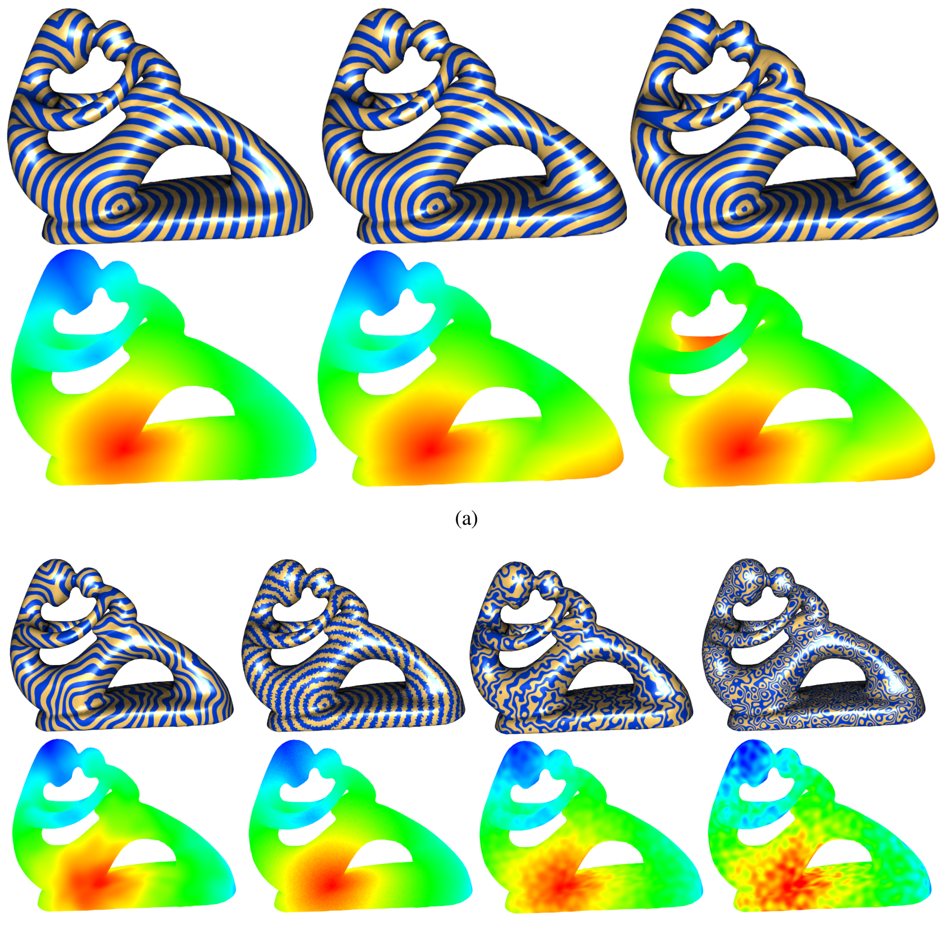
III. Geodesic Distance Computation

Geotextures relies on the geodesic distance computation in a fast efficient manner. We use the geodesic distance computation described by Torchelsen et al. [1]. Weber et al. [13] could also be applied but would restrict us to regular connectivity meshes, while the former fits triangular meshes with irregular connectivity and, with Surazhsky et al. [11] method, give us more accurate distances. We apply Cohen et al. [14] to generate the charts, followed by the method of Wang [15] to reduce the number of charts by merging the ones that results in charts with less than a given threshold of distortion. This way we can control the distortion by defining the amount of charts to be generated with [14] trading accuracy for speedup once that lower numbers of charts results in faster geodesic field computation. As the plane parameterization method we used Sheffer et al. [16] angle based flattening approach.

Being the final charts quasi-developable surfaces, the distance variation inside each chart is constant in any direction and each chart can be re-triangulated to reduce the number of primitives, but preserving the borders is essential so connectivity between neighbors charts is not lost in the process. Finally [11] can be applied on the final atlas of simplified charts, where the lower number of triangles allows faster wave front propagation.

IV. Geodesic Texturing

Our proposal is to define the features of a procedural texture over a surface according to a given distance field increasing the controllability of the procedural model. The diagram in figure 3 gives an overview of our scheme. Starting from a 3D model represented by a two-manifold unstructured triangular mesh we first create a version of this initial surface suited for the computation of the geodesic distances, we call this final version ”the base mesh”. This



(a)

(b)

Figure 4. (a) Different geodesic distance fields color mapped from red to blue on a surface and isolines of the respective distance fields. (b) Distance field distortion with different noise scales.

stage consists in the application of the methods in [14] and [15] for charts generation, followed by the ABF++ parameterization [16] and the internal simplification of the charts, as explained in the previous section. Now we have the original mesh associated with an atlas of quasi-developable and simplified charts as the output of the this stage.

With the base mesh ready, [11] can be adapted and applied to calculate the distance fields as shown in [1]. In [10] the use of an hierarchy of distance fields was presented to reduce the stall time of the agents when their goal changes, thus allowing the manipulation of such points in real-time. Such solution could be easily adapted to the case of texturing, since the main idea is the use of a hierarchy of versions of the same mesh with different simplification levels. Similar to the agent navigation example, the difference in the accuracy of distance fields is not so noticeable in the simulation, it would lead to perceptible changes in the texture appearance when the applied distance field is replaced by a more accurate version. For such reason we leave the definition of the features source points and the respective distance fields in the preprocessing stage.

The distance field calculation algorithm takes as input a vertex of the surface mesh. In our texturing scope we call this one a ”feature source”, and then defines for each vertex of the surface mesh a value representing the shortest geodesic distance from this vertex to the feature source. These resulting distances are normalized by the longest one and then provided to the procedural texture shader as an array of 1D texture coordinates. In figure 3 the distance fields, calculated taking V1 and V2 (in the bowl on the left hand and in the jar on the right one respectively) as feature sources, are mapped with a red-to-blue color spline on the surface according to the color scale in figure 1.

Although Surazhsky et al. [11] algorithm can create a distance field with more than one source point, doing so showed to be, in execution time terms, equivalent as creating distinct distance fields for each source point, making the second approach more interesting since we can exploit CPU parallelism to compute multiple distance fields faster, also giving more flexibility to the procedural texture by providing

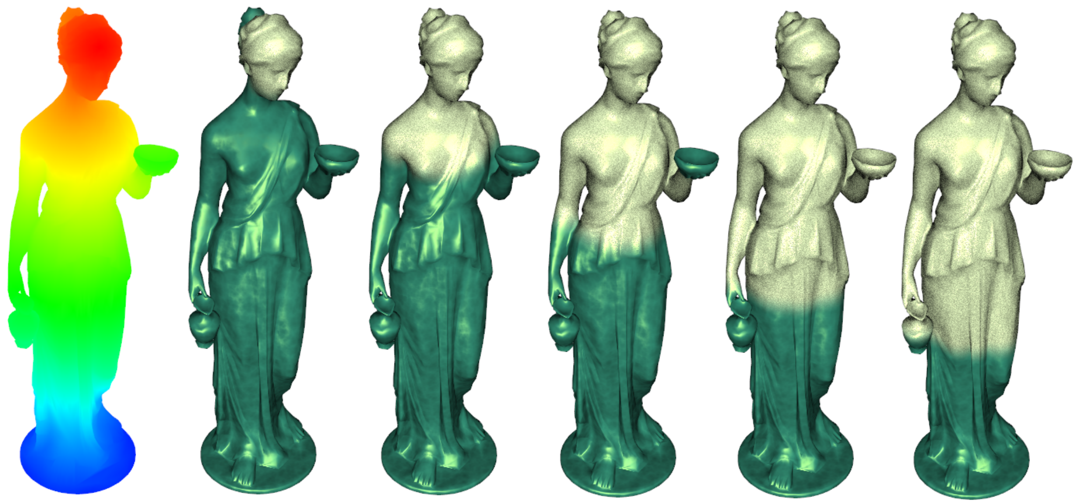


Figure 5. Geodesic distance field applied to define sand isle with the center on the model's forehead.

multiple independent distances.

Several distance fields with different feature sources can be used for procedural texturing in a shader with no extra cost in the evaluation of the functions used for the texture generation. They can be used separated to define different features with different sources, or can be easily merged in a preprocess stage or even in real-time. In preprocessing, we can consider for each vertex only the smallest value associated to it among a set of the distance fields provided. An alternative is to apply the same concept in real-time in the vertex shader or in the pixel shader where the distances values have already been interpolated for each fragment resulting in a merged field with better resolution.

Figure 4(a) shows the isolines of different geodesic distance fields over the surface of the fertility model. The middle and right examples use a distance field created by the merge of the left immediate neighbor example and one created taking a new location as source (the right bottom of the base and the elbow of the back arm respectively). Notice how the distances are defined over the whole surface in a consistent manner. The isolines cover the surface in a relief conforming way. By adding noise to the distance field, we can modify the regular nature of the propagation front achieving more realistic effects. This idea is better illustrated in figure 4(b), where different noise scales and amplitudes are applied to the distance fields.

V. Results

All the preprocessing stage, from the creation of the base mesh to the computation of the distance fields, was implemented in C++, the simulation rendered with OpenGL being the procedural textures defined as glsl shaders. For all the simulations we used a Intel Core2 CPU 6420 2.13GHz and a NVIDIA GeForce 8600 GT. Although the images depicts results with one to three feature sources, we have chosen such amount in order to provide a better understanding of the process. Our method imposes no limitation on the number of feature sources used, as becomes clear in figure's 4(b) right most image, where the applied distortion creates a set of virtual feature sources around the real one, or in the number of materials defined by the texture, having the last one influence only in the computational cost of the procedural texture.

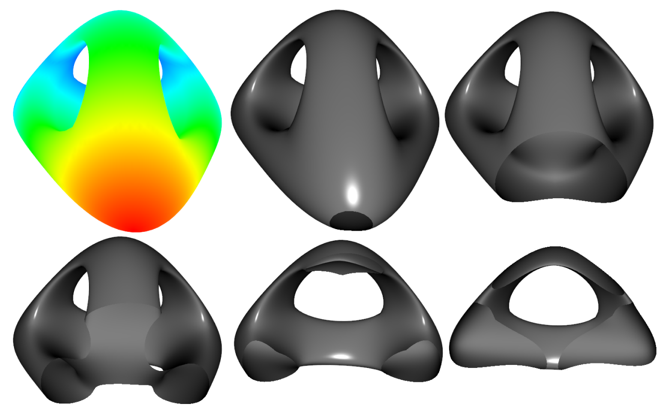


Figure 6. Application of geodesics to define the opacity along the mesh's surface.

In figure 5 we used the distance field to define the location as well as the extension of a portion of some sand-like material on the surface textured with a material resembling sea water. A given value of threshold defines in the procedural model the transition between the two materials as the interpolation parameter of a smoothstep for the color and bump function. In this example, the sand-like material is applied when the geodesic distance value is less than the provided threshold. Notice the behavior of the shore in the texture as the threshold is interpolated from 0 to 1.

The shore expands from a single point in a uniform way in every direction, following the gradient of the distance field and conforming to the surface relief. It shows no undesired behavior like the ones formerly shown on figure 2. The sand-like appearance was achieved by the use of a high frequency simplex noise as bump function, and the water-like one resulted from a fractal sum of Worley noise also as bump function. We refer the interested reader to [17] and [18] for more about simplex noise and Worley noise respectively and [2] for procedural texture synthesis.

Figure 6 shows the result of the same use of the geodesic distance threshold to alter the surface's opacity. The use of geodesic distance to define transparency can be useful to explore the topology of complex surfaces as well as its interior. As shown in the pictures, the continuous variation of the threshold can fully reveal shape of complex surfaces with no change on the point of view.

To propagate the same material from several sources at the same rate, a single distance field can be used. The spreading of moss in figure 7 uses a single distance field with three distinct sources: two at the base of the sculpture and one at the elbow of the further arm. A sum of simplex noise turbulence and fractal Worley noise was employed to model the bumps of the moss texture, while simplex noise turbulence alone defined the bumps for the grainy stone.

In figure 8 we simulated the dispersion of two fluids to exemplify the independent propagation of materials with different properties from different sources. In such case, one distance field is required for each material, in the example we used only a single source point for each material, but multiple sources can be defined as well. The water and lava appearance was created the same way as the sea water in figure 5 example, the dark burn texture is the result of bump-mapped simplex noise turbulence. Also some noise was added to the threshold value associated to the lava material shore to introduce a slight irregularity to it, giving a higher viscosity.

As a last application, we used a single distance field to the modeling of rust evolution on a metal chair. Three materials are defined according to the distance to three different points. In the chair model it becomes even more evident the relief conforming nature of the propagation due to the several holes of the chair shape. The propagating front of each material follow strictly the silhouette of the model. The paint layer is simply blue color with some specular and a smoothstep bump variation on the threshold with the metal material, the last one is result of a sum of fractal sum of simplex noise and high frequency simplex noise as bump with low specular influence. The rusty appearance is also created with fractal sum, and a fourth threshold is defined, with some high frequency noise added to its threshold in order to change the opacity of the surface simulating the degeneration of the chair due to the rust action. By adding noise only to threshold value that defines the front of this last "transparent material", we distort only this propagation front instead of modifying the whole field, this way we can use the same field to propagate fronts with different perturbation levels.

Table I relates the complexity of the models in terms of number of faces and charts to the time in seconds to calculate a geodesic distance fields in the preprocessing stage. In table II the frame rates for rendering the previous examples are shown. The isolines frame rate was added for comparison purposes as an example of rendering time for a simple texture. Rendering time is affected only by the complexity of the texture function and the number of rendered fragments, so for every texture example on table II there is the frame rate when the model is rendered close to the camera (Near), with more fragments, and farther (Far), with less fragments rendered. The number of distance fields has no influence on the rendering performance.

Model	Triangles	Charts	Average(s)	Deviation
Fertility	10000	14	2,97	0,12
Genus3	13312	19	5,28	0,37
Chair	25524	127	9,36	0,27
Hebe	7989	101	1,78	0,15

Table I
MODEL COMPLEXITY AND DISTANCE FIELD CALCULATION TIME

VI. CONCLUSION AND FUTURE WORK

We presented in this work a scheme to define relief conforming procedural features on complex surfaces as well as it's propagation over time. To ensure consistent propagation and low apparent distortion we applied geodesic distances as a mapping parameter to the procedural models. As examples of applications we simulated the propagation of multiple materials from different sources over complex surfaces of different genus.

Although the idea showed interesting results, there's still room for several improvements especially regarding the geodesic distance field creation stage. A faster computation of geodesic distances could allow the modification of feature sources in real-time opening way for new applications. The use of the geodesic distance fields can also be further explored to define other behaviors of the modeled materials as to induce flow direction.

ACKNOWLEDGMENT

The work of Guilherme Oliveira is sponsored by a CNPq scholarship process 579744/2008-6. The work of João Comba is funded by CNPq processes 305770/2008-0 and 569239/2008-7.

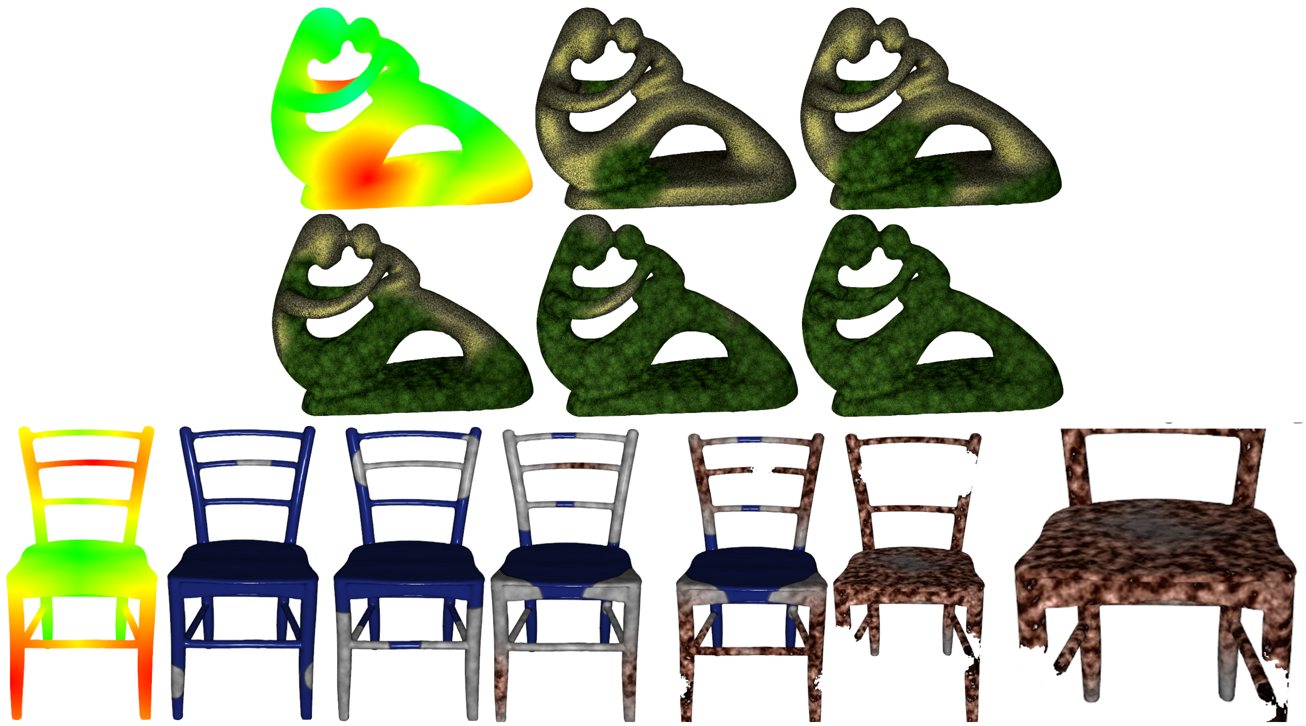


Figure 7. (Top) Moss spreading from multiple sources over granite. (Bottom) Different materials propagating from the same sources simulating rust evolution on a metal chair.

	Water/Sand		Moss		Rust		Water/Lava		Isolines	
Model	Near(fps)	Far(fps)	Near(fps)	Far(fps)	Near(fps)	Far(fps)	Near(fps)	Far(fps)	Near(fps)	Far(fps)
Fertility	9	75	7	75	9	97	3	48	71	232
Genus3	8	52	6	53	9	83	4	45	100	239
Chair	10	46	7	39	10	55	6	32	124	219
Hebe	15	77	13	78	18	117	10	64	148	247

Table II
RENDERING TIMES: AVERAGE FRAME RATES FOR THE DIFFERENT MODELS AND TEXTURE EXAMPLES.

REFERENCES

[1] R. P. Torchelsen, F. Pinto, R. Bastos, and J. L. D. Comba, “Approximate on-surface distance computation using quasi-developable charts,” *Comput. Graph. Forum*, vol. 28, no. 7, pp. 1781–1789, 2009.

[2] D. S. Ebert, F. K. Musgrave, D. Peachey, K. Perlin, and S. Worley, *Texturing and Modeling: A Procedural Approach*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2002.

[3] D. R. Peachey, “Solid texturing of complex surfaces,” in *SIGGRAPH ’85: Proceedings of the 12th annual conference on Computer graphics and interactive techniques*. New York, NY, USA: ACM, 1985, pp. 279–286.

[4] M. Walter, A. Fournier, and D. Menevaux, “Integrating shape and pattern in mammalian models,” in *SIGGRAPH ’01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. New York, NY, USA: ACM, 2001, pp. 317–326.

[5] K. Xu, D. Cohen-Or, T. Ju, L. Liu, H. Zhang, S. Zhou, and Y. Xiong, “Feature-aligned shape texturing.”

[6] Q. Yu, F. Neyret, E. Bruneton, and N. Holzschuch, “Scalable real-time animation of rivers,” mar 2009, to appear. [Online]. Available: http://artis.imag.fr/Publications/2009/YNBH09

[7] K. Hegeman, HongyuWang, M. Ashikhmin, X. Gu, and H. Qin, “Gpu-based conformal flow on surfaces.”

[8] J. Stam, “Flows on surfaces of arbitrary topology,” in *SIGGRAPH ’03: ACM SIGGRAPH 2003 Papers*. New York, NY, USA: ACM, 2003, pp. 724–731.

[9] E. Catmull and J. Clark, “Recursively generated b-spline surfaces on arbitrary topological meshes,” pp. 183–188, 1998.

[10] R. P. Torchelsen, L. F. Scheidegger, G. N. Oliveira, R. Bastos, and a. L. D. Comba, Jo “Real-time multi-agent path planning on arbitrary surfaces,” in *I3D ’10: Proceedings of the 2010 ACM SIGGRAPH symposium on Interactive 3D Graphics and Games*. New York, NY, USA: ACM, 2010, pp. 47–54.

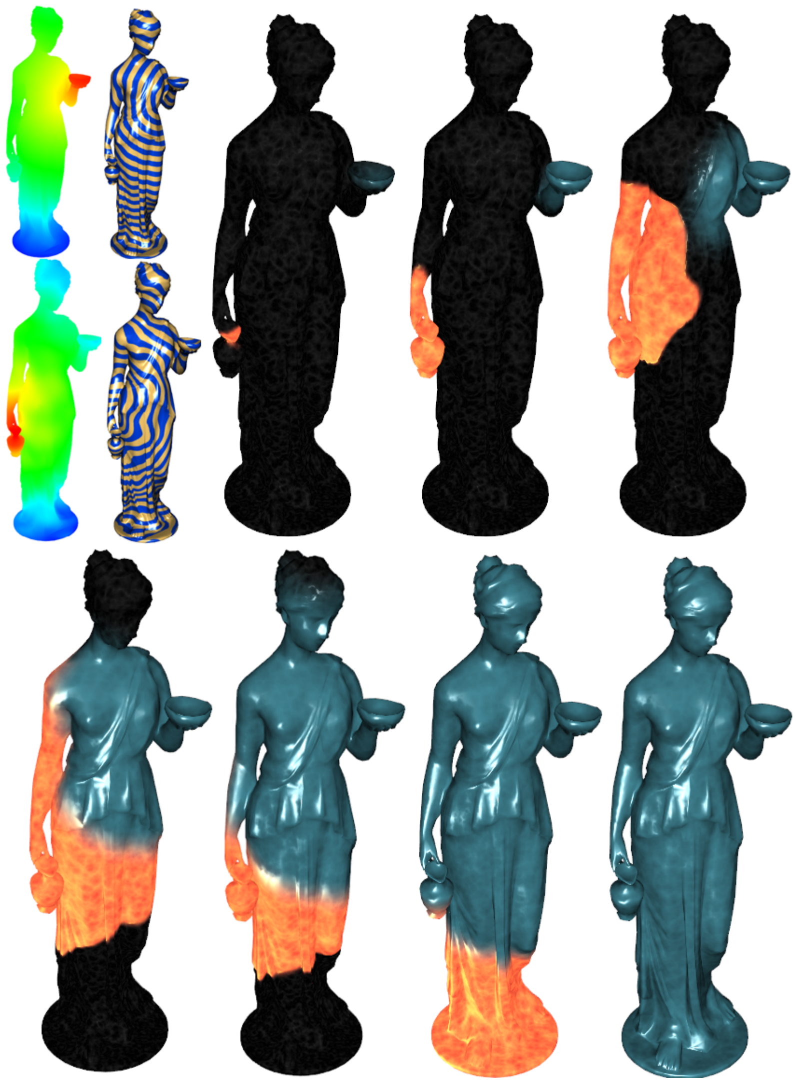


Figure 8. Propagation of different features from distinct sources. Noise is added to distance field associated to the lava material introducing irregularities in the propagation front. Water propagation has no noise addition and overlaps the lava flow. Both propagations are independent from each other and continue to evolve unaffected by the other front.

[11] V. Surazhsky, T. Surazhsky, D. Kirsanov, S. J. Gortler, and H. Hoppe, "Fast exact and approximate geodesics on meshes," in *SIGGRAPH '05: ACM SIGGRAPH 2005 Papers*. New York, NY, USA: ACM, 2005, pp. 553–560.

[12] D. Bommes and L. Kobbelt, "Accurate computation of geodesic distance fields for polygonal curves on triangle meshes," in *VMV*, 2007, pp. 151–160.

[13] O. Weber, Y. S. Devir, A. M. Bronstein, M. M. Bronstein, and R. Kimmel, "Parallel algorithms for approximation of distance maps on parametric surfaces," *ACM Trans. Graph.*, vol. 27, no. 4, pp. 1–16, 2008.

[14] D. Cohen-Steiner, P. Alliez, and M. Desbrun, "Variational shape approximation," in *SIGGRAPH '04: ACM SIGGRAPH 2004 Papers*. New York, NY, USA: ACM, 2004, pp. 905–914.

[15] C. Wang, "Computing length-preserved free boundary for quasi-developable mesh segmentation," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 1, pp. 25–36, 2008.

[16] A. Sheffer, B. Lévy, M. Mogilnitsky, and A. Bogomyakov, "Abf++: fast and robust angle based flattening," *ACM Trans. Graph.*, vol. 24, no. 2, pp. 311–330, 2005.

[17] K. Perlin, "Chapter 2 noise hardware," 2001.

[18] S. Worley, "A cellular texture basis function," in *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. New York, NY, USA: ACM, 1996, pp. 291–294.