Computers & Graphics xxx (2018) xxx–xxx

Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

Real-time field aligned stripe patterns☆

Q1 Nils Lichtenberg[a,*], Noeska Smit[b], Christian Hansen[c], Kai Lawonn[a]

[a] Institute for Computational Visualistics, University of Koblenz-Landau, Koblenz, Germany
[b] Visualization Group, University of Bergen, Bergen, Norway
[c] Computer Assisted Surgery Group, University of Magdeburg, Magdeburg, Germany

A R T I C L E I N F O

Article history:
Received 23 January 2018
Revised 27 April 2018
Accepted 29 April 2018
Available online xxx

Keywords:
Parameterization
Visualization
Computer graphics
Computational geometry,

A B S T R A C T

In this paper, we present a parameterization technique that can be applied to surface meshes in real-time without time-consuming preprocessing steps. The parameterization is suitable for the display of (un-)oriented patterns and texture patches, and to sample a surface in a periodic fashion. The method is inspired by existing work that solves a global optimization problem to generate a continuous stripe pattern on the surface, from which texture coordinates can be derived. We propose a local optimization approach that is suitable for parallel execution on the GPU, which drastically reduces computation time. With this, we achieve on-the-fly texturing of 3D, medium-sized (up to 70 k vertices) surface meshes. The algorithm takes a tangent vector field as input and aligns the texture coordinates to it. Our technique achieves real-time parameterization of the surface meshes by employing a parallelizable local search algorithm that converges to a local minimum in a few iterations. The calculation in real-time allows for live parameter updates and determination of varying texture coordinates. Furthermore, the method can handle non-manifold meshes. The technique is useful in various applications, e.g., biomedical visualization and flow visualization. We highlight our method's potential by providing usage scenarios for several applications.

1. Introduction

In surface visualization, there is often a need to visualize additional features of the data directly on the surface. If there is only one value that needs to be shown, color mapping is often employed to provide a qualitative impression of the value distribution over the surface. However, for multivariate data, the need can arise to visualize multiple values simultaneously, and simple color mapping will no longer suffice. Multiple views can be presented in such cases, but this requires mental integration for the viewer. Glyph-based or layering techniques are also able to convey multiple quantities, but may lead to clutter and occlusion [3]. To provide the user with an integrated view of multiple features, advanced visualization techniques such as illustrative visualization can be used to encode additional information. For such techniques, however, preprocessing is often required. This has the unfortunate side-effect that those techniques can no longer be employed to display dynamic changes, and there may be cases where preprocessing is undesirable or even impossible. Furthermore, when relying on precalculation, it is not possible to update any parameters involved at run-time. Therefore, there is a need for a method that is able to provide parameterization of a surface mesh without preprocessing and that can be adjusted on-the-fly.

We propose a technique to parameterize a triangulated surface and generate a global stripe pattern on the surface, based on an underlying tangent vector field. If no vector input is available, principal curvature directions could be computed as a backup strategy. This is also possible in real-time as stated by Griffin et al. [4]. The resulting parameterization can then be used for different visualizations tasks. Existing methods [5–7] already address this kind of problem. However, these do at most focus on interactivity, while we aim for a real-time visualization, allowing dynamic input properties. Further, our problem formulation is suitable for an optimized reconstruction of the parameterization in the fragment shader. This is beneficial, e.g., if our method is used to generate local texture coordinates. To make our method suitable for real-time applications, we adapt existing approaches and aim for a local solution through local iterative optimization steps. The locality of our approach allows handling of non-manifold surfaces. Also, we can update visualizations and their parameters on-the-fly, for instance driven by dynamic vector fields, or reactive to scene changes resulting from interaction. With this, our main contributions are the following:

☆ This article was recommended for publication by Prof. D Panozzo.
* Corresponding author.
E-mail address: nlichtenberg@uni-koblenz.de (N. Lichtenberg).

https://doi.org/10.1016/j.cag.2018.04.008

Please cite this article as: N. Lichtenberg et al., Real-time field aligned stripe patterns, Computers & Graphics (2018), https://doi.org/10.1016/j.cag.2018.04.008

- We propose a technique to derive local texture coordinates from tangent vector-fields on a surface mesh, through local iterative optimizations.
- Our technique can be executed in real-time for medium-sized meshes, and thus can be used in visualization of both dynamic meshes, as well as dynamic parameter input.
- We demonstrate the potential of our technique in several usage scenarios from various domains, and compare the performance of our technique both quantitatively and qualitatively to reference methods.

We obtain periodic 1D texture coordinates based on a 1D parameterization aligned to an unoriented vector field. This can be employed for field visualization using a stripe pattern. The parameterization based on two orthogonal vector fields can be used to obtain periodic 2D texture coordinates. These can be used to visualize vector fields or arbitrary scalar properties using different textures or patterns, as we demonstrate in several examples.

2. Related work

In this section, we examine related work from a technical perspective, as well as from a visualization application perspective.

Surface parameterization techniques. Surface parameterization has been intensely researched for a long time [8]. Global parameterization plays an important role in global quad remeshing algorithms in order to find an optimal remeshing across the whole mesh. A survey on this topic is provided by Bommes et al. [9]. Such methods are usually complex to implement, run at most at interactive timings [10] and thus, are not applicable in real-time applications. Jakob et al. [7] proposed a method that relinquishes global optimization, yet is still able to create meshes that align with features on a global scale. This local approach makes their method parallelizable, which makes finding a solution faster by several orders of magnitude. Such techniques define, or get as input, a direction field on the surface, along which the parameterization is aligned. Proper generation of such direction fields is crucial to guarantee mesh quality for these methods. Design of these direction fields has emerged from the above requirements as an additional research area. Details can be found in the state of the art report by Vaxman et al. [11]. Our work uses as input an unoriented vector-field and does not address its further optimization. The methods mentioned so far generate vector fields, or at least require an optimized vector-field as input, and use them in successive steps. The design and visualization of direction fields is often closely coupled to allow for a visual feedback of applied changes [12]. The visualization is often done using line integral convolution (LIC) [13,14]. However, LIC does only convey the ambiguous orientation of a vector direction $\mathbf{d} \sim -\mathbf{d}$ and cannot be used to display textures. Other methods, like the generation of texture coordinates, utilize vector-valued input to control texture orientation. Then, attention has to be paid to whether the vector field is oriented or non-oriented. Methods that take orientability into account can be used for a controlled display of orientable textures, but have to take care of visual seams [15–17], while methods that work on unoriented fields have to rely on symmetric textures [18,19].

The most important prior art to the work presented here are the position field optimization of the Instant Field Aligned Meshes (IFAM) algorithm by Jakob et al. [7] and the technique for stripe pattern synthesis on surfaces (SPS) by Knöppel et al. [6]. The IFAM algorithm has introduced a local and parallel solution to global parameterization and the patterns that result from applying SPS are globally smooth and applicable for design and texture synthesis tasks. The interpolation scheme by Knöppel allows for a globally continuous pattern away from isolated singular points. Global continuity refers to the property that no jumps in the pattern can be found across the surface (i.e., no seams are visible). More precisely, if a piecewise continuous pattern is given and the pattern is based on a periodic function, the periodicity results in repetitive piecewise continuity across the surface, hence achieving global continuity. In contrast to SPS, our technique finds a locally optimized solution through local iterative optimization steps, which makes it suitable for real-time applications without requiring any precalculation.

Related visualization applications. One of the potential application areas for our technique is to employ the generated stripe-patterns as an additional visual encoding channel for multivariate data visualization. Multivariate data is defined in the comprehensive survey by Fuchs and Hauser as information which has an attribute vector for each data item [20]. In the field of multivariate data visualization, Rocha et al. [21] recently proposed a real-time technique to map decals onto surfaces as a new way of representing multivariate data. The sets of images or patterns mapped to the surface are able to represent attributes of the data at the location they are mapped to, and can be used in combination with additional layered visualization elements. In contrast to their approach, we are able to handle dynamic flow patterns in addition to real-time texture coordinate synthesis, since we generate a globally continuous pattern. The work by Schroeder and Keefe [22] specifically caters to time-varying multivariate data visualization by providing an artist with an interface to sketch such visualizations. In their work, they allow artists to sketch illustrative elements that can be used as animated glyphs in a layered 2D visualization. However, their technique is focused on visualization design on a flat 2D surface. In earlier work by Kirby et al. [23], the potential of using illustrative techniques borrowing concepts from painting to visualize multivalued 2D flows was highlighted. Our technique is also able to generate illustrative strokes for flow, but extends to more complex 3D surfaces. Furthermore, we are able to animate these strokes to represent time-varying vector fields. Recent work by Roy et al. [24] use LIC to visualize the sheets of branched covering spaces. However, LIC is not suitable for expressing the unambiguous directionality of vector fields, and thus they require animations to express this aspect.

To the best of our knowledge, ours is the first work to use a globally smooth parameterization for visualization purposes, based on dynamic input data that can be updated in real-time. This concept, w.r.t. to visualization purposes, is inspired by the work by Knöppel et al. [6], who generate a continuous *stripe pattern* on a surface, based on an input vector field. They also present details on the proper visualization of their parameterization results and, e.g., how to obtain texture coordinates from that. Their approach in turn is based on the method to generate a periodic global parameterization (PGP) as described by Ray et al. [5], who focus on re-meshing purposes. The stripe pattern algorithm introduces several changes in order to drastically improve the performance. Jakob et al. [7] were the first to translate the problem addressed by the above mentioned methods to a formulation that allows a local and thus parallel execution of the optimization. However, their CPU implementation is suitable for interactive, but not for real-time performance. Furthermore, the frequency of their periodic pattern is limited by the mesh resolution.

We incorporate ideas and concepts of the above mentioned work and extend these with the goal to come up with an algorithm that allows for parameterization in real-time and is suitable for visualization purposes. We contrast the prior work in the way that we obtain coordinates for orientable textures, how these coordinates can be aligned with the underlying field on a pixel basis and we employ a convergence term for the optimization process. Furthermore, we show a range of application scenarios that can be

Please cite this article as: N. Lichtenberg et al., Real-time field aligned stripe patterns, Computers & Graphics (2018), https://doi.org/10.1016/j.cag.2018.04.008