Computers & Graphics 74 (2018) 161–170

Contents lists available at ScienceDirect

Computers & Graphics

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

Special Section on SMI 2018

Hierarchical representation for rasterized planar face complexes

Guillaume Damiand[a,*], Aldo Gonzalez-Lorenzo[a], Jarek Rossignac[b], Florent Dupont[a]

[a] CNRS, Univ Lyon, LIRIS, UMR5205, F-69622 France
[b] School of Interactive Computing, Georgia Institute of Technology, Atlanta, USA

ARTICLE INFO

Article history:
Received 27 April 2018
Revised 15 May 2018
Accepted 15 May 2018
Available online 24 May 2018

Keywords:
Planar polygonal meshes
Irregular representation
Hierarchical representation
Combinatorial maps
Compact representation
Topology preserving rasterization

ABSTRACT

A useful example of a Planar Face Complex (PFC) is a connected network of streets, each modeled by a zero-thickness curve. The union of these decomposes the plane into faces that may be topologically complex. The previously proposed rasterized representation of the PFC (abbreviated rPFC) (1) uses a fixed resolution pixel grid, (2) quantizes the geometry of the vertices and edges to pixel-resolution, (3) assumes that no street is contained in a single pixel, and (4) encodes the graph connectivity using a small and fixed number of bits per pixel by decomposing the exact topology into per-pixel descriptors. The hierarchical (irregular) version of the rPFC (abbreviated hPFC) proposed here improved on rPFC in several ways: (1) It uses an adaptively constructed tree to eliminate the "no street in a pixel" constraint of the rPFC, hence making it possible to represent exactly any PFC topology and (2) for PFCs of the models tested, and more generally for models with relatively large empty regions, it reduces the storage cost significantly.

1. Introduction

Consider a planar graph, G, that is embedded in the plane and comprises a ***connected*** network of finite and possibly curved edges and their bounding vertices. For example, each edge may represent a street and each vertex may represent a street junction. Their union decomposes the plane into faces that may be topologically complex. For example, G may have ***multi-edges*** (more than one edge joining any given pair of vertices). Furthermore, a face may contain, in its boundary, ***cracks*** (edges that bound a single face), ***dead-ends*** (vertices that bound a single edge—a crack), and ***loops*** (edges that start and end at the same vertex). The unbounded face is called ***exterior***. An example is shown in Fig. 1. We use the term ***Planar Face Complex*** (PFC) for such an arrangement.

Many applications need to represent and to traverse a PFC. Examples include street networks in ***Geographic Information System*** (GIS) [1], geological models [2], overlapping SVG elements [3], and multi-material structures [4].

Different solutions have been proposed to represent PFCs. Some approaches describe the connectivity of the graph [5–7] explicitly. This may yield a high storage cost for complex graphs. Other approaches use an image format (regular grid of pixels) to describe a rasterized approximation of the PFC [8–11], which assigns each pixel to a different face, without attempting to capture the topology inside the ***shared*** pixels that contains one or more edges.

The recently proposed ***rasterized Planar Face Complex*** (rPFC) [12] unifies these approaches by defining a compact representation of the topology of the PFC that decomposes it into per-pixel descriptors, each using a short string of bits to encode the topology of the intersection of the PFC with a pixel.

The rPFC model has many advantages: (1) It represents graph connectivity exactly and hence supports exact topological graph traversal; (2) It provides spatial indexing to both quantized geometry and exact topology; (3) It can represent non-trivial topology in a pixel (such as dangling edges, multiple vertices and multiple connected components); (4) It requires only a few bits per pixel.

However, the rPFC has a drawback: It cannot represent a graph that has an edge that fits entirely inside a pixel. Hence, to represent a graph with some relatively small edges, we either must use a high-resolution grid (see Fig. 1), which increases storage cost, or must simplify the graph by collapsing small edges in a preprocessing step, which implies the loss of the original topology. Furthermore, when the rPFC encoding stores a topology descriptor for each ***private*** pixel (a pixel that lies entirely in a face), the rPFC storage of large clusters of private pixels is wasteful.

1.1. Motivation

Our overarching motivation is to reduce the storage size of this graph, while preserving the benefits provided by w rPFC

* Corresponding author.
E-mail addresses: guillaume.damiand@liris.cnrs.fr (G. Damiand), aldo.gonzalez-lorenzo@liris.cnrs.fr (A. Gonzalez-Lorenzo), jarek@cc.gatech.edu (J. Rossignac), florent.dupont@liris.cnrs.fr (F. Dupont).

https://doi.org/10.1016/j.cag.2018.05.017
0097-8493/© 2018 Elsevier Ltd. All rights reserved.

representation, namely (1) random access and traversal (RAT) at constant amortized time (CAT) cost and (2) constant cost localization of the edges and vertices that intersect any given pixel. We also wish to provide efficient support for distributed processing, window-stream processing, and progressive refinements.

We believe that the above characteristics are important for navigation, query, and maintenance applications of huge databases of planar graphs, which may represent the geometry and connectivity of streets, rivers, or utility networks.

Our second main motivation is to use the 2D representation as the main tool to define 3D compact representation. This paper is the first step, necessary for the definition of a compact representation of 3D meshes.

1.2. Contribution

The high-level, novel contribution reported here is the combination of a hierarchical representation with the rPFC (per-pixel) encoding of geometry and connectivity. In this paper, we define the ***hierarchical rasterized Planar Face Complex*** representation (***hPFC***) which addresses the drawback of the previous rPFC.

The proposed hPFC is essentially a tree. Hence, our solution includes a quadtree as a special case. At the coarsest level, it is an rPFC, A. But some of the pixels of A, instead of containing the bitstring that encodes the local topology of the PFC, contain an index to a refined rPFC of the portion of the PFC inside that pixel. Such a more detailed rPFC, B, may, in turn contain pixels which, each, refer to even finer rPFC, C, and so on recursively.

This irregular representation allows us to remove the “no small edge” constraint imposed by the rPFC: When an edge fits entirely in a pixel, the pixel is subdivided.

Moreover, using an irregular (hierarchical) grid allows to reduce the storage cost of large clusters of private pixels.

For example, the rPFC shown in Fig. 1 uses a grid of 192 pixels and involves 79 ***crossings*** (points where an edge of G crosses a pixel border). A coarser grid would produce an invalid rasterization in which at least one edge is contained in a single pixel. As shown in Fig. 2, using the irregular grid of an hPFC solves the problem: The same PFC may be encoded as an hPFC that uses only a total of 42 pixels and involves only 30 crossings.

1.3. Organization

The paper is organized as follows. In Section. 2, we review the rPFC model and discuss other relevant prior art. In Section. 3, we present the hPFC model and the details of the operators needed to navigate through the PFC using its hPFC representation. In Section. 4 we give a compact encoding of hPFC that provides a good time-complexity for traversal operators, while allowing to navigate through the graph without needing to decode the whole data-structure, but only the current pixel. In Section. 5, we present experimental results, comparing our new solution with the previous regular version.

2. Prior art

2.1. Data-structures for polygonal meshes

A variety of edge-based data structures have been proposed in order to represent polygonal meshes, such as Combinatorial Maps, Corner Table, Doubly Connected Edge List, Half-Edge, Surface Mesh...[5–7,13–19]. They differ in their storage cost, in the type of operators that they support, and in the topological restrictions that they impose on the mesh. Many are reviewed in [20,21].

These data structures provide ***Random Access and Traversal*** (***RAT***) of the meshes, often in constant time, or sometimes in ***Constant Amortized Time*** (***CAT***). Their main drawback is to use a large number of bits per element (edge, vertex), which limits their applicability and performance for complex meshes.

Some solutions used rasterized images, where each pixel stores the color of the region that contains its center. The image can be compressed, for example by using RLE (Run Length Encoding). But the digitization does not represent street networks, removes all cracks and dead ends, and can disconnect regions.

Rasterized images were used in [22] to accelerate the rendering of antialiased vector graphics. That approach uses a coarse lattice in which each cell contains a variable-length encoding of the graphics primitives that overlap it. The proposed hPFC extends this previous work by capturing the connectivity (incidence and ordering) of the graph in a constant-length per-cell format and hence providing support for RAT operators.

In [9,23], a solution stores the crossing vertices between the mesh and an inter-pixel grid, and recompute (explicitly or implicitly) a simplified topology of the mesh. Such a representation can be used to accelerate Boolean operations [4]. But it only represents an approximated topological description of the mesh.

2.2. Compact representations of polygonal meshes

Several compression schemes propose to encode local mesh connectivity by using a few bits per element for polygon graphs [24,25] and for triangle meshes [7,26–32].

Often, the connectivity information is broken into a chunk per face, per edge, or per vertex. For example, the 2D version of Tetstreamer [33] organizes triangle faces into topological rings and divides connectivity information into one bit per edge (for some edges) and one or several bits per vertex. But extensions of this approach to more general (PFC) graphs would be challenging and the representation more expensive. More importantly, such schemes assume that the bits of the mesh encoding is received in a specific order. This compressed format must be decompressed first and converted into a more expensive format that is suitable for RAT in CAT.

More recent representations offer a much low storage costs while still supporting RAT in CAT for the most common access and traversal operations. For example, the Zipper format is restricted to triangle meshes, but uses on average only 6 bits per triangle and can be constructed in linear space and time [34]. Such representations rely on a specific ordering of vertices. The streamable version, Grouper [32], of this approach stores about two vertex-references per triangle.

2.3. Hierarchical representations of polygonal meshes

Many hierarchical solutions have been defined in order to reduce the memory space used in order to represent a mesh such as for example quadtrees [8,35]. [36] proposes a progressive mesh representation, a new scheme which allows to store and to transmit arbitrary triangle meshes. Several other hierarchical and pyramidal models were defined and used for example to represent multiresolution terrain models [37]. In [38], a compressed encoding of 3D triangular meshes is defined, based on a hierarchy and an encoding of split operators, which allows to encode both manifold meshes but also “triangle soups”. In [39], a compressed random-access tree is used in order to represent spatially coherent data. But these representations are either for grid of pixels, or for triangle meshes.

Quadtrees were also used to represent a set of points [40,41] or of line-segments [42–44]. These representations do not capture connectivity. The MX quadtree [45] does capture the connectivity of simple polygons, but does not support junction vertices with more than two incident edges. Hence, these previously proposed