Computers & Graphics 36 (2012) 1036–1047

Contents lists available at SciVerse ScienceDirect

Computers & Graphics

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

Graphics Interaction

A procedural method for irregular tree models

Ling Xu*, David Mould

Carleton University, Canada

ARTICLE INFO

Article history:
Received 2 July 2012
Received in revised form
18 August 2012
Accepted 22 August 2012
Available online 31 August 2012

Keywords:
Procedural modeling
Tree modeling
Natural phenomena
Geometry synthesis

ABSTRACT

We present a method to generate models for trees in which we first create a weighted graph, organized based on the Yao graph, then place endpoints and root point and plan least-cost paths from endpoints to the root point. The collection of resulting paths forms a branching structure. We create a hierarchical tree structure by placing subgraphs around each endpoint and beginning again through some number of iterations. Powerful control over the global shape of the resulting tree is exerted by the shape of the initial graph, composed of simple geometric primitives arranged in part manually and in part procedurally. Users can create desired variations by adjusting the initial graph shape; more subtle variations can be accomplished by modifying parameters of the graph and subgraph creation processes and by changing the endpoint distribution mechanisms. The method is capable of matching a desired target structure with a little manual effort, and can easily generate a large group of slightly different models under the same parameter settings. Environmental effects can also be incorporated into the models by automatic parameter adjustment. The final trees are both intricate and convincingly realistic in appearance.

1. Introduction

Trees are commonplace in the natural world, and tree models often appear in the virtual worlds of computer games and films. Fig. 1 shows some tree structures. In the top image, we see some typical characteristics of trees: irregular individual branches and multiple levels of detail. A typical global tree shape, seen in the lower image, includes a single trunk, some main branches, and smaller branches that form the tree crown in a hierarchical structure. The overall tree shape possesses a complex beauty, which unfortunately is tedious to create with human labor. To ease the burden on digital artists, procedural modeling methods have been devised, able to create many types of models but especially useful for complicated subjects such as trees.

This paper presents a procedural method to model trees, based on finding least-cost paths through a weighted graph, a modeling idea previously introduced by Xu and Mould [1–3]. The essential idea is to create a graph with random edge weights, then plan least-cost paths from a single root node to destination nodes. The resulting paths form a tree. By varying the graph shape and edge weights, the method can create a wide range of tree models.

An earlier version of this paper [3] described a modeling method involving sequences of graphs, using path planning to link endpoints in all graphs with a single root node. In this extended version, we build our initial graph using the Yao graph to reduce the number of edges without compromising quality; we propose refining the shapes of graphs to get more natural details in the resulting tree structures; and we suggest an approach to incorporate environmental factors into our tree growth process. Our method can create realistic, highly intricate tree models, with quite direct user control over the final tree shape through specifying the shapes of the graphs in which the tree is built.

The paper is organized as follows. Following the introduction, we review some previous work in tree modeling. In Section 3, we describe the algorithm. Results and evaluation are given in Section 4. Finally, we conclude and discuss future work.

2. Previous work

Tree modeling has a long history in computer graphics. The most notable modeling approach is the parallel rewriting grammar called L-systems, used for plant forms and even entire ecosystems [4–6]. General control over grammar-based methods is offered by Talton et al. [7], although their sampling process can be very time-consuming. The space colonization method of Runions et al. [8] offers a biologically motivated alternative with control over global shape, exploited by Palubicki et al. [9] for self-organizing tree modeling; here, the tree growth process follows the competition of branches for resources (e.g., light and space) with internal

---

* Corresponding author.
E-mail addresses: lxuc@scs.carleton.ca (L. Xu), mould@scs.carleton.ca (D. Mould).

0097-8493/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.cag.2012.08.005

Fig. 1. Examples of tree structures.

signaling mechanisms based on L-systems. The resulting forms can be controlled interactively, e.g., by sketching [10].

Geometric methods (e.g., that of Weber et al. [11]) explicitly vary geometric quantities such as segment length and angles; generating models involve adjusting a huge number of parameters. An alternative is to use input images to drive tree creation [12–14]: such methods can attain extremely high quality, although the need to supply input images is a drawback.

The basic idea of path planning [15] for tree modeling is due to Xu and Mould [1,2], who exploited path planning for general modeling of dendritic natural phenomena including trees, coral, and lightning. In their work, the graph containing the paths is a 2D lattice or 3D grid. The regular lattice imposes substantial penalties: the resolution of the model is limited by the spacing of the graph, and hence small-scale features (e.g., tiny twigs) need a high-resolution graph, incurring immense memory cost.

To enhance the control in tree modeling, sketch-based methods are used to provide clues of crown shape or main branches [16–19]. Based on L-systems, Ijiri et al.'s system [16] controls the growth direction of a tree by user-drawn strokes. However, to model a complex tree would require a lot of user interaction. Okabe et al. [17] build tree models using freehand sketches with the assumption that branches are spreading to maximize the distance between each other. In Chen et al.'s method [18], Markov random fields are used to infer the branch shape from the drawn sketches. Both methods use examples from a library of tree templates for branch propagation, which reduces the burden on user sketching. With similar stochastic optimization, Wither et al. [19] use a priori botanical knowledge to infer branch shapes from user sketched crown silhouettes at different scales, and can generate realistic tree models with good overall control.

Compared to the above methods, scanning methods focus on creating models of real trees, using point clouds of tree data obtained by 3D scanning. Xu et al. [20] build a tree skeleton by connecting neighborhood points to form a graph, where a single-source shortest path algorithm is applied to reconstruct branches. Bucksch et al. [21] extract a tree skeleton by subdividing the point cloud. Livny et al. [22] apply global optimizations to reconstruct multiple overlapping trees simultaneously. Scanning methods can achieve high quality of tree models, but are not intended to model novel trees.

3. Algorithm

We build on the method of Xu and Mould [1], who created least-cost paths through a regular lattice connecting multiple endpoints to a common root in order to build general tree-like structures. Since they used a regular lattice, they little investigated the task of building graphs; a significant portion of this paper is devoted to defining the graph shapes, which have an enormous impact on the shape of the final tree. The earlier work also did not pay much attention to the details of endpoint placement. We propose an iterative method whereby successive stages of endpoints are distributed within subgraphs, resulting in a high degree of visible structure; we discuss the details of the method next, to be followed by examples of our synthetic tree images.

3.1. Basic algorithm

We construct a graph and find the shortest paths from multiple endpoints to a common root point. The collection of the paths form the tree model. The basic algorithm can be decomposed into the following steps.

1. Build a graph and set edge weights randomly.
2. Choose a node to be the root point and some nodes as endpoints of the structure.
3. Find least-cost paths from the endpoints to the root.
4. Create geometry around the path segments and render the resulting model.

Xu and Mould used a graph consisting of a regular lattice, but the resulting paths suffered from lattice artifacts. We propose instead an irregular graph, obtained by creating a Poisson disc distribution of nodes within a designated volume; nodes are connected by edges by a policy, described below, and a random weight is assigned to each edge. Fig. 2 contrasts regular and irregular graphs: use of an irregular graph avoids lattice artifacts without necessitating higher resolution.

One option for the edge connection policy is simply to link two nodes whenever their distance is below some threshold. This policy is generally effective and was used for some examples in this paper. However, choosing the threshold is problematic: too large, and the number of edges per node is excessive; too small, and the graph can become disconnected. An alternative which

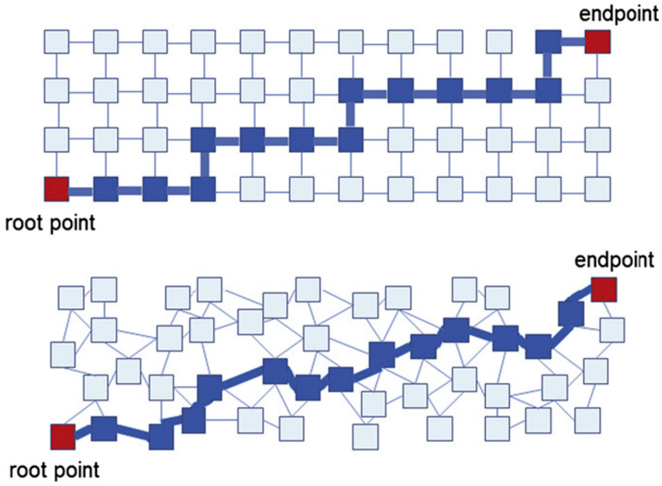

Fig. 2. A regular and an irregular graph.