GPU Accelerated Method for Constructing and Rendering Trees
-
Literature Review

Thomas Mcloughlin

registration: 100203952

1 Introduction

The rendering of trees has advanced greatly from the early days of real-time applications. The earliest method of adding trees to an environement would be to simply load in a flat image of a tree and then have the image rotate to face the view point of the camera. This was acceptable for single placed trees but for areas with many trees the likely approach would be to repeat the image around the central axis of the tree to give a sense of volume.

Developments in graphics hardware allowed for more detailed models to be used in real-time applications. This would've started with more simple manually modelled trees with actual branches and leaves applied using bill-boarded textures or sometimes individually for games with less demanding environments.

Presently, trees are added to real time environments as fully developed realistic looking models. Trees are constructed and rendered using software and then loaded into a scene, with the exception of some that may be modelled manually if a specific shape or look is needed for the tree.

This project is a similar system, aiming to be able to produce realistic and varied looking trees in a real-time application to help bring life to an environment. The following section will provide a brief description of the project, the areas of knowledge required to complete the project and a general roadmap for the rest of the document.

1.1 Description

Generating natural environments can be costly. Creating and rendering realistic models of trees can be challenging. This projects goal is to investigate approaches for creating and rendering trees to be used in a real-time graphics application.

A key reason for wanting to include trees in computer generated environments is that trees, and other foliage, are what give life to that environment, a forest is not a forest without the trees and having an easy method of including trees in a landscape will mean that making that landscape more realistic and engaging becomes easier.

1.2 Knowledge

The key areas of knowledge required to complete this project are as follows:

- Branch Growth - The technical knowledge to produce natural looking branches will be most important to the success of this project, if the branches of the trees do not look naturally shaped then the final tree will not look realistic.

  The method used for naturally formed branches will be decided from researching multiple sources to find a suitable algorithm that can be implemented as efficiently as possible.

- Leaf Placement - Similarly to the issue of branch growth, the placement of leaves on the tree branches will also be important with respect to making the rendered trees look realistic.

  The method chosen will also similarly be researched from multiple sources to find the most appropriate algorithm for efficient implementation.

- Texturing - The textures applied to the trees after the geometry is finalised will not be as essential as the previous points but it will still play a key role in making the trees look realistic.

  A method will need to be used to correctly apply a bark texture to the trunk and branches of the tree without jarring edges being noticeable. The same attention will also need to be given to the leaf texture and the decision of whether to include some transparency in the leaves.

- Branch Pathing - This can be grouped with the branch growth section as a possible extension. The idea that an obstacle could be used to obstruct the growth of the tree's branches and the branch growth algorithm would take this into account and render the tree to look it has grown to avoid the obstacle.

  This would help the rendered trees properly integrate with the environment they're placed in and add a more realistic feel.

- Root System - A possible inclusion would be a ground level root system for the trees, not a fully rendered underground system as that would be pointless, but some of the roots of larger trees will be visible around the base of the tree.

  Some variation could be added to the trees by not having them magically sprout from the ground and instead showing some of the ground roots. A method would have to be chosen to model this, possibly using the sam method for the branches but reversed.

1.3 Roadmap

The following sections of this document will include: comparisons of material related to the project to be used to inform the direction that will be taken in design and development, following the material comparisons - a discussion of the key issues and themes related to the project and how some of the discovered materials can be used to understand these facets, and finally an evaluation of the document containing a critical review of the content presented throughout.

2 Comparisons of Related Material

This section will include a description of the methods used to find relevant materials to discuss around the project and a comparison of these materials where overlap is found.

2.1 Search Method

Relevant papers were found using google scholar and using search terms including "real-time", "rendering", "modeling" etc. effort had to be made to avoid papers referring to data structure related trees and focus was given to making sure that the chosen material was about real-time rendering in evironments and not just methods for generating premade models.

The different papers have varying focuses not all exactly about the rendering of trees, some were chosen to give some possibly needed background that will be relevant to the project less directly such as talking about fast rendering of large environments including trees.

Papers where not excluded based on date of release as the methods discussed are mostly algorithmic and not necessarily dampened by advances in technology, most relevant papers have been released withing the last 20 years however due to that being when hardware became powerful enough to begin this form of research in earnest. The papers chosen were also checked to have around 20 and above citations which should show that they have acceptable credibility.

2.2 Comparison of Material

In this section some of the key areas of required knowledge will be discussed by method with reference to various papers that have been found to contribute to each subject.

2.2.1 Branch Structure

The branching structure of rendered trees is the most commonly talked about subject across most papers relating to rendering trees. This is likely because the branch structure is most important with regards to making a tree look realistic, therefore it becomese the main focus of most relevant papers.

One of the earliest algorithmic methods of branch structure comes from Honda (1971) where a “mother” branch splits into two “daughter” branches using given angles and having a determined length from a ratio with the “mother” branch, this method can give various tree crown shapes providing a useful groundwork for tree construction. What’s missing is the ability to calculate thickness of branches, or the possibility of producing branches that are not completely straight.

Another pioneering work for branching structures was the creation of L-systems, proposed by Lindenmayer (1968) and later discussed by Prusinkiewicz et al. (1996), which is a more general method of producing fractal data but can be applied to graphical drawing techniques to produce relatively realistic looking branching structures. It works by using a dictionary of variables, represented as characters, that are assigned to certain drawing techniques, commonly examples using turtle graphics, and then from a given axiom the variables are built upon using certain rules. Once a chosen number of recursions is achieved the resulting string is then used to direct the drawing. With care taken to what rules are used this method can be successful in producing many different kinds of branching shapes. The use of L-systems also allows for the creation of bending branches unlike the method presented by Honda, however the methods presented by honda could be included in the ruleset of an L-system to produce trees of a simiar structure. Both Lindenmayer and Honda's papers are both included in the Journal of Theoretical Biology.

Weber and Penn (1995) and Prusinkiewicz et al. (2007) both make reference to Lindenmayer and Honda's work with their presentation for branching structures. Weber and Penn present the construction of a tree using truncated cones for "mother" branches and cones for terminating branches, by using cones they avoid the drawback of Lindenmayer and Honda's original methods that only produced a skeletal structure. They use a similar method for splitting "mother" branches into dichotomous "daughter" branches but presents a much more detailed model allowing for more control of the various facets of the tree.

Prusinkiewicz et al. suggest the use of "attraction points" which are used to guide the growth of a tree skeleton into a desired shape where each point has an area of attraction that applies some value to any branches within it's radius, this causes the branches to grow towards the points until all points are reached or a chosen number of iterations is completed. The method referenced by Prusinkiewicz et al. for giving the tree skeleton thickness is from Bloomenthal (1985) where the branches of the tree skeleton are treated as the centre point of many generalized cyclinders with varying radii, this requires the tree skeleton construction method to include the data for normals along sections of the branches so that the cyclinders can be constructed properly.

This method of generalized cyclinders could be applied to an implementation of Honda's method or a chosen L-system to give those trees thickness, avoiding the use of the cone method presented by Weber and Penn that may look too artificial. Bloomenthal also suggests the use of splines instead of lines, for the branches, to make the trees look more natural.

A common part of the methods posed by both Weber and Penn (1995) and Prusinkiewicz et al. (2007) that has not been mentioned is the use of a terminal enveloping of the tree skeleton to prevent the branches from growing out too far or to restrict the tree crown into a preferred shape, Prusinkiewicz et al. refer to this as just an envelope whereas

Weber and Penn call this overall method "Pruning".

A later paper by Weber (2008) presents another similar construction method to those mentioned above but with more attention paid to the possible simulation of the rendered trees. He puts forward a method for branches avoiding obstacles using collisions with environment objects, this is done by pushing any branches that collide with the given object away from its bounding area by changing their branching angles. This addition of realism for the tree growth moving around obstacles can help the trees feel more integrated with the environment they are placed into. Weber also begins to talk about the simulation of tree movements with respect to wind which will be discussed in a later section.

2.2.2 Leaf Placement

Leaf placement on tree models has seen many different methods applied over the years of development of graphical hardware. The chosen method of leaf placement for the rendering of a tree is coupled with the chosen branch structure method so that the leaves can be placed realistically along the branches giving the best possible result.

The earliest example comes with the L-systems method discussed in the previous section, proposed by Lindenmayer (1968) and discussed by Prusinkiewicz et al. (1996), where some of the variables chosen from the "alphabet" of possible permutations will include the drawing of branches and then leafs to go along with them, no direct method is proposed for how these leaves would be rendered however as these papers discuss mostly with turtle graphics but some of the methods about to be mentioned could likely be used in the L-system variables to render the leaves.

Weber and Penn (1995) present leaf placement as part of their overall algorithm using a level system for the recursive construction of the tree. Leaves are more densely populated on greater levels of the tree and the number of leaves per branch at each level is determined by a formula using some inputted variables to achieve various appearances. Weber and Penn also suggest an alteration to the orientation of the leaves after placement so that the leaves will face the light correctly. The rendering method chosen for the leaves is that there is a given set of simple geometric leaf shapes to choose from using an id called "LeafShape" these shapes range from a simple oval to more detailed split maple style leaves. This reduces the realistic looks of the tree as the leaves will not be textured but the rendering process will be very fast using such simple shapes.

Wesslén and Seipel (2005) actually directly uses the method given by Weber and Penn (1995) but makes a change to the leaf rendering process. Rather than use simple geometric shapes to represent the leaves, the leaves are modeled using textured primitives meaning that the quality of the leaves will look better when viewed from up close as compared to a plain shape from Weber and Penn's original model. This obviously comes with the large time gap between the two papers allowing for the use of more advanced hardware.

Prusinkiewicz et al. (2007) uses a different method to choose leaf placement but tries to achieve the same affect as Weber and Penn. They say that the spatial distribution of the leaves is determined with respect to the parallel transport frame and/or absolute directions in the world space in which the tree has been placed so that leaves can be placed along the branches curves while remaining parallel with respect to the connection (NEEDS CLARIFICATION).

Bloomenthal (1985) uses an interesting method for the leaf structure on his tree models, to note he his focussing on only maple trees, by having the leaves use a polygon that is hinged along the weaker sections of the leaf which allows for the shape to be varied depending on the strength of a hypothetical wind. Bloomenthal also clusters leaves together into what he calls leaf "configurations" and for each limb that does not exceed a given diameter and that has no outgoing limbs, a leaf configuration is chosen at random, scaled randomly and placed at the limb tip. This method is obviously limited as it is made specifically for maple trees but the ideas used could be transfered to other methods to add depth.

Candussi et al. (2005) discusses leaf placement with more of a focus on level of detail management but will be discussed here for completeness. Candussi et al. choose to represent leaves on the tree models using billboarding where various flat polygons are layered throughout the tree crown and then given a texture containing multiple leaves to give the look of a densely populated tree. This method is similar in execution to Bloomenthal (1985) but much simpler as it does not use separate polygons for each leave which saves geometry.

2.2.3 Wind Affect

The affect that wind has on a tree model can have a large impact on the realism of that model by adding a living element to the model rather than just having a static object. The chosen method for adding wind affects is not tied to the methods chosen for branch structure or leaf placement so supposedly any method could be used for the chosen branch and leaf methods. Weber and Penn (1995) model wind affecting trees as bending the branch segments by using the deflection of an elastic rod with a circular cross-section fixed on one end with a uniformly distributed force applied. The consider the rod as a pendulum and each branch is given a slightly varied oscillation to give a natural looking sway to the tree.

Wesslén and Seipel (2005) have their trees react to wind in two ways: in light wind only leaves flutter which they model as a quick rotation around the spine of each leaf and in heavy wind the tree will sway using a rotation from the base of the trunk, the rotation amount will be relative to the distance from the base of the trunk so that further, thinner branches will sway more than the nearer, thicker branches. This differs from Weber and Penn's method with the introduction of leaf movement, however their method of branch movement where each branch sways independantly will likely produce a more realistic result when compared to the method described above.

Candussi et al. (2005) model their trees differently again, with pre-computed animation properties, such as the flexibility of the branches. At run-time the tips of the branches are displaced and the rest of the branch movement is done by applying the same transformation to each vertex in the branch with alteration based on each vertices distance from the tip. The displacement of the branch tips are computed using the sum of two vectors, one in the direction of the wind and the other being orthogonal to the first to add some noise to the movement. When compared to the methods of Weber and Penn or Wesslén et al. this method would allow more control over the branch movement by using the target and noise vectors rather than rotations.

Weber (2008) and Bloomenthal (1985) both give the ability to apply affects of wind to their tree models but they are different from the previously mentioned methods in that they are not actively animated once rendered, the described wind affects they present are for adding to static models. Weber allows for the branch structure to be formed differently as though it grew under the pressure of heavy wind which would add to an environment that is meant to be under those conditions. Bloomenthal gives his leaf polygons hinge points to allow for the leaves to deform to show wind affect but does not describe a method for having this be an active animation rather than just an addition to producing a static model.

2.2.4 Detail Management

When rendering any scene in a real-time environment there needs to be attention paid to the performance of the program, care needs to be given to the rendering methods to allow for a smooth framerate (30fps or higher, ideally 60fps). This is done by reducing the details that get rendered when they wont have any affect on the perceived detail of the scene and due to the large amount of geometry involved in tree rendering, it is important to understand what measures can be taken in the rendering process to improve the performance of whatever environment they are placed in.

Weber and Penn (1995) discusses this issue of “degradation at range” and mentions the commonly used method for most 3d models where multiple versions of the model are stored, each of a varying level of fidelity, and depending on the distance of the camera from the object a certain version of the model is chosen that will give enough detail to look realistic but remove any details that would be impossible to see from that distance. The issue with this method that Weber and Penn bring up is that, with trees in a forest environment, this method would cause there to be very apparent “resolution waves” in the forest canopy that would not be acceptable when trying to make the scene look realistic. Instead of this method they propose that the separate levels of the tree models be simplified using a finer method that reduces detail more gradually sometimes one facet at a time. The tree geometry is organized into discrete geometric descriptions: 3 stem levels and the leaves. Any stems beyond the third level are grouped with the third level. The deeper levels of stems are rarely visible from a distance due to being obscured by the leaves. Weber and Penn refer to Oppenheimer (1986) that he recognised the use of polygonal tubes for large scale details and lines for smaller details. Over increasing distance, the rendering algorithm reinterprets the geometry by changing stem meshes into lines and leaf polygons into points and then at even longer ranges some stems and leaves will completely disappear. Weber and Penn's speculations on LOD were from 1995 however. In more recent years the problem with apparent "resolution waves" has been worked on to produce better detail degredation that doesn't show the obvious drops in detail throughout an environment.

References

Bloomenthal, J. (1985). Modeling the mighty maple. *ACM SIGGRAPH Computer Graphics*, 19(3):305–311.

Candussi, A., Candussi, N., and Höllerer, T. (2005). Rendering realistic trees and forests in real time. In *Eurographics (Short Presentations)*, pages 73–76. Citeseer.

Honda, H. (1971). Description of the form of trees by the parameters of the tree-like body: Effects of the branching angle and the branch length on the shape of the tree-like body. *Journal of theoretical biology*, 31(2):331–338.

Lindenmayer, A. (1968). Mathematical models for cellular interactions in development ii. simple and branching filaments with two-sided inputs. *Journal of theoretical biology*, 18(3):300–315.

Oppenheimer, P. E. (1986). Real time design and animation of fractal plants and trees. *ACM SiGGRAPH Computer Graphics*, 20(4):55–64.

Prusinkiewicz, P., Hammel, M., Hanan, J., and Mech, R. (1996). L-systems: from the theory to visual models of plants. In *Proceedings of the 2nd CSIRO Symposium on Computational Challenges in Life Sciences*, volume 3, pages 1–32. Citeseer.

Prusinkiewicz, P., Lane, B., and Runions, A. (2007). Modeling trees with a space colonization algorithm. *NPH*, 7:63–70.

Weber, J. and Penn, J. (1995). Creation and rendering of realistic trees. In *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '95, page 119–128, New York, NY, USA. Association for Computing Machinery.

Weber, J. P. (2008). Fast simulation of realistic trees. *IEEE Computer Graphics and Applications*, 28(3):67–75.

Wesslén, D. and Seipel, S. (2005). Real-time visualization of animated trees. *The Visual Computer*, 21(6):397–405.

Literature review

Introduction: brief description of project, areas of knowledge required, roadmap	First	2.1	2.2	3	Fail
Discovery of suitable quantity and quality of material	First	2.1	2.2	3	Fail
Description of key issues and themes relevant to the project	First	2.1	2.2	3	Fail
Evaluation, analysis and critical review	First	2.1	2.2	3	Fail

Quality of writing

Clarity, structure and correctness of writing	First	2.1	2.2	3	Fail
Presentation conforms to style (criteria similar to conference paper reviews)	First	2.1	2.2	3	Fail
References correctly presented, complete adequate (but no excessive) citations	First	2.1	2.2	3	Fail

Revised Workplan (if applicable)

Measurable objectives : appropriate, realistic, timely	First	2.1	2.2	3	Fail

Comments

Supervisor: Dr. Stephen Laycock

Markers should circle the appropriate level of performance in each section. Report and evaluation sheet should be collected by the student from the supervisor.