Ninth IEEE International Symposium on Multimedia 2007 - Workshops

Animating Tree Branch Breaking and Flying Effects for a 3D Interactive Visualization System for Hurricanes and Storm Surge Flooding

Khalid Saleem[1], Shu-Ching Chen[1], Keqi Zhang[2]

[1] *Distributed Multimedia Information system Laboratory, School of Computing and Information Sciences, Florida International University, Miami, FL, USA*
[2] *International Hurricane Research Center, Florida International University, Miami, FL, USA*
[1]{ksale002, chens}@cis.fiu.edu, [2]zhangk@fiu.edu

Abstract

This paper illustrates branch breaking and flying effect animation for trees in our 3D interactive visualization system for hurricanes and storm surge flooding. The tree branch breaking and flying effect animation extends our current tree animation framework by estimating the destination quadrant, direction (angle) and traveling speed with which a branch breaks and flies across the terrain. These calculations thus assist in simulating the potential threats to property and human lives posed by the flying branches during a hurricane, and add to the realism and appeal of our system. We utilize 3D tree models created with 3DSmax and implement the animation effects via OpenGL and nVidia CG shader language.

1. Introduction

Hurricanes are perilous natural phenomena that have constantly threatened the United States coastal regions across the Atlantic Ocean, Gulf coast and the Pacific Ocean. The wind speed intensity (74 mph and above) associated with hurricanes can cause massive and wide spread property/infrastructure damage while posing serious risk to human lives. Such high wind speeds across the coastal regions can result in storm surge flooding that further aggravates the financial/economical woes of the public/private sector after the hurricane landfall. The storm surge flooding associated with hurricanes poses an additional threat to human lives, owing to the increased possibility of death by drowning, and hence places immense responsibility on the emergency management officials and general public in executing timely and effective precautionary measures. Such effective preparation and planning can assist in mitigating the financial and human life losses associated with these hurricanes.

Recent hurricanes such as Katrina (2005) and Rita (2005) are evident examples of how untimely and ineffective planning resulted in a chaos. Katrina has proven to be one of the worst natural disasters in the history of USA, in part due to the lack of timely and effective planning by the state and federal emergency management agencies. Rita came after Katrina and was less intensive. However, due to the lack of knowledge on part of the public and ineffective planning by the emergency management officials, the mass exodus of 3 million people left the evacuating routes jammed across the state of Texas and many people had to spend around 25 hours on the interstate highways. The food and fuel shortages across the interstate highways further deteriorated the conditions for the evacuating population [1]. Many residents who were not required to evacuate panicked thus causing massive traffic gridlocks. Additionally, miscommunications and disruptive planning caused some people to evacuate using routes within the path of the impending storm.

Current methodologies for hurricane preparation/planning and dissemination of information to the general public involve 2D/3D maps or satellite images from National Hurricane Center (NHC) or print/electronic media indicating the storm track and illustrating different evacuation zones under mandatory evacuation orders on top of the county/state maps. This only provides the planners and general public some basic information about the impending threat and over time it has shown that it lacks the necessary visual detail with which the general public can relate to and using which the emergency management officials can effectively plan/prepare and disseminate hurricane related information. To assist the general public and emergency management officials in better understanding the threat and evaluating the extent of the threat and damages associated with an impending hurricane, we designed and implemented an Interactive 3D visualization system for hurricane and storm surge impacts [2, 3]. Our system allows for the users to visualize the possible effects of an impending storm around the location of their interest and thus, plan and strategize accordingly for preparation or evacuation. Our system uses LIDAR (Light Detection and Ranging) data for accurate terrain data representations along with 3D models and animation engines for visualizing and rendering hurricane and storm surge flooding impacts.

An important component of our 3D animation system is the tree animation engine that renders wind driven vegetation animations for the vegetation instances around a location of interest. The tree animation engine animates trunk and branch bending effects for trees along with the branch breaking and flying effects for a near realistic representation of hurricane impacts. The focus of this paper is the branch breaking and flying effect for the tree animation engine that animates the breaking and flying away effects for tree branches based on different wind speed thresholds. During a hurricane, tree branches can break and the massive wind force can cause these branches to hit and break windows or inflict other property damages such as broken wind shields for vehicles. The tree branch breaking and flying simulation estimates the destination quadrant where the broken branch will flow along with the branch travel direction (angle) and speed. The branch flow direction is computed via the initial wind direction, while the speed is computed utilizing the hurricane wind speed and scaling parameters for realistic rendering. These calculations assist in animating broken branches and their corresponding movement across the terrain. These animations simulate the potential dangers posed by broken branches to property and human lives, while ensuring near realistic and immersive experience to the users. Finally, we utilize the programmable graphics units for efficient animation and rendering of highly detailed 3D models in real time.

The rest of the paper is organized as follows. Section 2 provides literature review of previous works related to vegetation animation. Section 3 illustrates the branch breaking and flying effect methodology. Section 4 provides an overall view of our 3D visualization system and Section 5 presents conclusion.

2. Previous Work

There have been numerous research works and implementations that aimed at creating and rendering plant life in 2D/3D visualization systems or as part of rendering evolution of ecological systems. However, most of these works have been focused on the representation of vegetation instances rather than animating it. [4] presents a methodology for rendering 3D vegetation instances utilizing Level of Detail (LOD) triangular meshes. These meshes do assist in ensuring a realistic and detailed complex plant data representation. However, at present, it does not support wind driven tree animations which can assist in simulating the wind driven movement of plants or trees.

In [5], the authors provide a technique for rendering the tiny hairs that cover many plant organs. This technique does not scale well to the needs of our 3D visualization system, no matter how useful it might be in understanding and visualization of vegetation instances. The reason is that during or prior to an approaching hurricane, the concern is not to study the effects of wind on vegetation hair, but to understand the effect of high speed winds on big trees and their branches that can cause damages to buildings or other property/infrastructure. [6] makes use of 3DSmax [12] for rendering tree animations representing the evolution of plants over a period of time. Again, such techniques do not fit into our 3D visualization system.

Recently, some work has also been done on animating 3D vegetation models with respect to wind force. [7] makes use of single view videos representing plant motion with respect to wind and extracts 2D position and velocity of clustered points. It then employs frame level algorithms for generating 3D animations. However, this simply emulates the behavior of plants or trees as represented by the single view video and works best for shrubs under the controlled environments (indoor), while for branch movements in response to strong winds, it results in feature and texture loss that greatly affects the realistic rendering of hurricane wind speed driven 3D tree animations. Such feature and texture loss is unacceptable for our 3D visualization system.

The “Blustery Trees” demo in [8] simulates wind driven tree animations. However, this work is limited to animating the entire tree model with respect to the wind. Our previous work [9] made use of 3D tree models and the vertex weighting mechanism [10] for generating LOD meshes for branches and tree trunks facilitating their sway and bending according to the specified wind speed and wind direction variables.

All the above works mainly focus on either rendering vegetation data or animating branch and tree trunk movement with respect to wind direction and speed. However, they either lack the support for or do not facilitate the tree branch breaking and flying effect animations. On the other hand, our tree branch breaking and flying effect animations can assist in evaluating the on site conditions during or after the hurricane and thus facilitate a timely and effective preparation against impending hazardous conditions.

3. Branch Breaking and Flying Effect Animation

Our branch breaking and flying effect animation is an enhancement to our previous implementation of the tree animation engine that animated wind driven tree trunk and branch sway movements. In our earlier version [9], we utilized vertex weighting or vertex skinning technique [10] to smoothly animate bending effects for rigid bodies that would exhibit sharp seams or gaps in the absence of such an approach. Our current enhancements utilize the same techniques for trunk and branch bending animations of 3D tree models. However, additional algorithms are introduced for representing branch breaks and flying effects. Our new approach is also based on the wind speed and wind direction variables that drive the different animation engines of our 3D visualization system. Our enhancements involve two phases i) branch breaking effect, and ii) branch flying effect. For the purpose of this paper, we used 3D tree models for a Royal Palm tree as it is the most common trees in those locations currently supported by our system.

3.1 Branch Breaking Effect Animation

This phase involves the breaking of tree branches based on a wind speed threshold. Our tree animation engine first utilizes LOD meshes along with the vertex weighting technique to generate the trunk and branch portions of a 3D tree model and applies texture to the vertices in the meshes for a realistic representation. These meshes comprising a specific portion of the tree (i.e., trunk or branches) are then animated according to the wind direction and wind speed as illustrated by [9].

In order to adjust the above mechanism for the branch breaking effect, a wind speed threshold was introduced to our bending algorithm. Once the wind speed exceeds a certain threshold, the algorithm chooses random branches and disconnects them from the top of the trunk to animate the breaking effect. The number of broken branches depends on the wind speed and storm intensity as shown in Table 1. At this point, the branch is fed to the flying effect phase for rendering the required animations. The wind speed threshold can be adjusted accordingly. However, for the purpose of this paper, it has been set to 90 mph. If the wind speed variable increases beyond 90 mph, our branch breaking algorithm begins simulating the branch breaks for the trees in our 3D visualization system. The number of broken branches depicted in Table 1 is chosen only for the simulation purposes. Our approach can be further extended to utilize physics based models for estimating the number of broken branches. In this paper, we set the number of broken branches is two for wind speed values between 90-100mph, four for wind speed values between 100-120mph, and all the branches for wind speed values greater than 120mph. Table 1 shows the algorithm for the branch breaking and flying effect animation. Branch flying effect is explained in Section 3.2. Here, *ws* represents hurricane wind speed, *ws_threshold* represents wind speed threshold, *num_break* represents number of tree branches to be broken, *tot_branches* represents total number of branches on the tree, and *removedBranches* represents the vector of meshes containing removed branches for flying effect animation.

Table 1. Algorithm for branch breaking and flying effect animation

```
Load .3DS tree model;
Extract trunk and branches;
Apply vertex weighting;
// ws=wind speed;
// ws_threshold=wind speed threshold;
if (ws < ws_threshold)
   Animate trunk and branch bending;
else {
   Animate trunk and branch bending;

   // num_break=number of broken branches;
   // tot_branches=total number of branches;
   if (ws > 90 and ws ≤ 100)
      Set num_break=2;
   else if (ws > 100 and ws ≤ 120)
      Set num_break=4;
   else
      Set num_break= tot_branches;

   Remove num_break branches from the top of the
   trunk;
   for (each branch in removedBranches) {
       Calculate branch travel quadrant, direction,
       and speed;
   }

   // removedBranches=vector of meshes
   // containing removed branches for flying effect
   // animation
   Animate flying effect for removedBranches;
}
```

3.2 Branch Flying Effect Animation

Once the tree branches have been disconnected from the top of the tree trunk, they are sent to the flying effect animation portion of our tree animation engine.

In order to simulate near realistic animations for the branch breaking and flying effects, the branch flying effect animation needs to take into consideration numerous variables and parameters. On the whole, the branch movement across the terrain is driven by the wind speed and wind direction parameters defined for our 3D visualization system. This step involved 3 phases: i) Branch Travel Quadrant Estimation, ii) Branch Travel Velocity Calculation (XY plane), and iii) Branch Travel along vertical plane (Z axis).

3.2.1 Branch Travel Quadrant Estimation. The branch travel quadrant estimation takes into account the wind direction with respect to the location of the tree to determine the region where the branch will fly away across the terrain. Based on the wind direction and the location of the tree, a branch disconnected from the top of the tree trunk can fall and fly in various directions. For the purpose of our 3D visualization system, we divide the terrain space around a tree into a four-quadrant region space (XY plane) with the tree located at the centre of this region space. The z-axis (not shown in Figure 1) represents the vertical elevation level across the terrain and is used to determine the vertical distance traveled above the terrain by a branch. The four-quadrant region space is adjusted for our visualization system coordinates where the wind direction of 270° represents wind coming from the direction of the camera position, and the points inside each quadrant correspond to the difference from the origin. Different wind directions can be represented by different angles as shown in Figure 1.

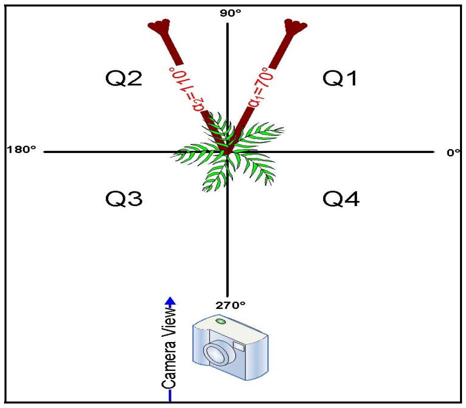

Figure 1. Wind Direction Angles

In Figure 1, Q1, Q2, Q3 and Q4 represent the four quadrants. As shown in this figure, a wind direction represented by an angle $\alpha_1 = 70°$ means that wind is flowing from Q1 and the branch after breaking will likely fall towards Q3. Similarly, a wind direction represented by an angle $\alpha_2 = 110°$ means that wind is flowing from Q2 and the branch after breaking will likely fall towards Q4. The location of any point (x, y) in a quadrant is denoted by the x and y coordinates respectively. Based on the above assumptions, we thus determine the region in which the broken branch is most likely to fall as follows.

i. $0° \leq$ Wind Direction Angle $\leq 90°$,
Branch falls in Quadrant = Q3.
ii. $90° <$ Wind Direction Angle $\leq 180°$,
Branch falls in Quadrant = Q4.
iii. $180° <$ Wind Direction Angle $\leq 270°$,
Branch falls in Quadrant = Q1.
iv. $270° <$ Wind Direction Angle $< 360°$,
Branch falls in Quadrant = Q2.

Currently, we utilize an initial wind direction provided by a hurricane forecast model to animate branch travel quadrant across the terrain. If the wind direction changes once the branches have broken, the new wind direction can then be used to animate the flying effects in the appropriate quadrant and direction.

3.2.2 Branch Travel Velocity Calculation. Once the branch travel quadrant has been determined, the traveling velocity (speed and direction) in the destination quadrant is computed to represent the branch flying effect. To provide realistic flying effects associated with the broken branches, the wind speed parameter is scaled at a rate of 4 frames per second (fps) to represent the flow relevant to the virtual terrain coordinates of our visualization system. Additionally, the branch flow along the terrain's XY plane is scaled as well to correspond to the wind direction angles. Equations 1 and 2 are defined to calculate the direction (angle) component of the branch travel velocity.

$$\theta = 180° + wd, \quad 0° \leq wd \leq 180° \qquad (1)$$

$$\theta = wd - 180°, \quad 180° < wd \leq 360° \qquad (2)$$

where θ is the branch travel direction (angle) and wd is the wind direction. To elaborate on this, considering a single broken branch with the wind direction equal to 300° (i.e., the wind is blowing from Q4), the branch should fly towards Q2 at an angle of 120° from the origin. Also, consider that the branch travels along the XY plane. In order to calculate the speed component of the branch travel velocity, we utilize Equation 3.

$$v_{speed} = \frac{(ws \times (1-\mu))}{FrameRate} \qquad (3)$$

where v_{speed} is the branch travel speed across the XY-plane, ws is the wind speed, μ is a constant whose value equals to 0.2 (the coefficient of friction for branch travel in air) and *FrameRate* is a constant whose value equals to 4 (i.e., frames per second rate). The branch travel speed is scaled to represent the speed across the terrain, e.g., 50 miles per hour (mph) represents 50 terrain points per second.

A coherent and near realistic representation of the branch flying effect requires some sort of friction and slow down effect to be introduced to the branch travel velocity calculation. The friction coefficient μ which represents the air friction effect across the terrain by limiting the travel speed of a broken branch in a certain direction, along with the *FrameRate* used in Equation 3 ensures that the motion of each branch with respect to the wind speed is rendered appropriately for the 4 fps rate and that the branch itself does not simply vanish. To elaborate further, a wind speed value of 90 mph would thus mean that the branch travel speed along the terrain is 18 points per frame or 72 points per second.

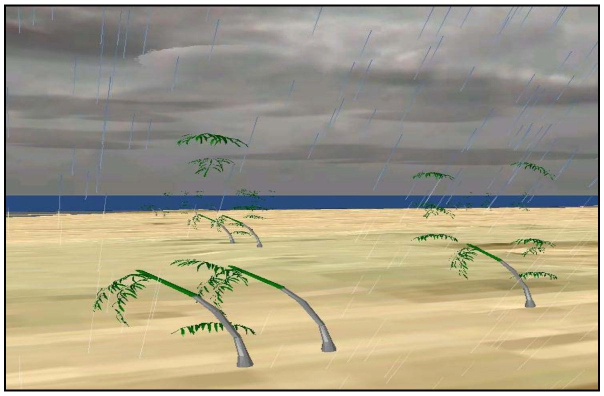
Figure 2. Branch Breaking and Flying Effect Animation

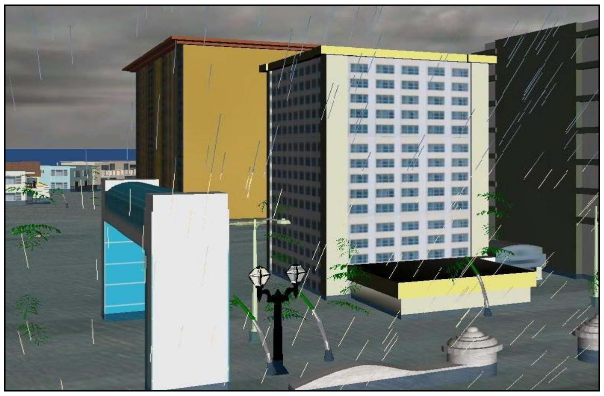
Figure 3. Branch Breaking and Flying Effect for Fort Lauderdale Beach Animation Scenario

3.2.3 Branch Travel along Vertical Plane. The discussion above considered the branch flow across the XY plane. However, wind driven movement of broken tree branches also exhibits motion along the vertical plane. In order to animate branch motion across the vertical plane, we simply randomize the branch motion on the vertical plane with the vertical height ranging from the bottom of the tree trunk (ground level) to one tree height above the trunk top. This along with the points obtained from the XY plane travel velocity calculation blends together extremely well to animate branch flying effects. Figure 2 and Figure 3 show our branch breaking and flying effect animation on a simple terrain and as part of the Fort Lauderdale Beach Animation scenario respectively.

Traditionally, vertex weighting methodology is a computationally expensive approach for the CPU, particularly since we have implemented the tree animation engine for the 3D visualization system which aims at performing in near real time on a single computer. In order to overcome the potentially expensive operations, we utilized the graphics processing unit (GPU) for processing these vertex weighting operations as mentioned in [9]. The vertex weighting method was implemented via nVidia Cg language [11].

4. 3D Visualization System for Hurricane Impact

Our tree animation engine is part of the 3D animation system for hurricane and storm surge flooding impacts. The system was designed to provide the general public and emergency management officials with a highly interactive visual tool for planning and preparing against an impending hurricane while analyzing and assessing the risks and threats associated with them. In order to provide realistic terrain level representations, LIDAR data is used. The LIDAR data is utilized to extract ground (terrain data) and non-ground (buildings, trees, etc.) measurements. Once these measurements are extracted, a visual representation of the terrain along with road, building, and vegetation instances is then generated to represent a 3D digital city model. The building/infrastructure, vehicle, and vegetation instances are represented via .3DS models created using 3DSmax [12].

We employed Virtual Terrain Project (VTP) [13] to generate the 3D terrain. VTP utilizes OpenGL [14] and C++ for rendering 3D terrain representation of different types of GIS data sets. To simulate hurricane and storm surge flooding effects, we developed animation engines which assist in animating the desired hurricane and storm surge flooding effects and are dependant upon the wind speed and wind direction variables. These animation engines include i) Ocean Water Animation, ii) Wind, Rain and Lightning Animation, iii) Tree Animation, iv) Traffic and Wake Effect Animation, and v) 3D sound engine for ensuring an immersive experience for the user. The system is highly interactive, allows the users to navigate across the terrain to visualize the extent of damages or possible threats associated with a hurricane, and can assist emergency management officials and general public to better prepare against an impending hurricane onslaught by providing near real time highly interactive 3D animation scenarios that the users can better relate with. Currently, we have implemented animation scenarios for 3 locations in Florida, i.e., Ft. Lauderdale Beach, Rickenbacker Causeway, and South Beach, Miami.

5. Conclusion

In this paper, we discuss the enhancements to our tree animation engine for generating wind driven branch breaking and flying effects. Our enhanced tree animation engine is part of the 3D visualization system for hurricane impacts. The enhanced tree animation engine determines the branch flow destination quadrant after breaking from the tree trunk and calculates the branch travel velocity components (direction and speed) to simulate the behavior of broken branches under hurricane conditions. We utilize GPUs for near realistic and efficient processing and simulation of our extended tree animation engine. Our approach allows the users to make better, educated and informed pre/post hurricane preparation and planning decisions.

We plan to further extend our tree animation engine to include relative trunk animation to adjust the tree trunk bending angle after the branches have been blown off the trees along with the destination quadrant wind direction perturbation calculation. Moreover, our animation engine can be further extended to support tree trunk breaking effects. We also aim to incorporate the window breaking animation effect for the buildings located along the broken branch travel paths across the terrain.

6. Acknowledgement

This project was supported in part by a grant from the National Oceanic and Atmospheric Administration.

7. References

[1] K. Mieszkowski, “How Rita drove Texas crazy,” Salon Online, http://dir.salon.com/story/news/feature/2005/09/27/traffic/index.html

[2] S-C. Chen, K. Zhang, and M. Chen, “A Real-Time 3D Animation Environment for Storm Surge,” *In Proceedings of the IEEE International Conference on Multimedia & Expo (ICME)*, vol. I, pp. 705-708, Baltimore, MD, USA, July 6-9, 2003.

[3] K. Zhang, S-C. Chen, P. Singh, K. Saleem, and N. Zhao, “A 3D Visualization System for Hurricane Storm Surge Flooding,” *IEEE Computer Graphics and Applications*, Vol. 26, Issue 1, pp. 18-25, Jan.-Feb. 2006.

[4] O. Deussen, C. Colditz, M. Stamminger, and G. Drettakis, “Interactive Visualization of Complex Plant Ecosystems,” *In Proceedings of IEEE Visualization 2002*, pp. 219-226, Boston, MA, USA, October 27-November 1, 2002..

[5] M. Fuhrer, H. W. Jensen, and P. Prusinkiewicz, “Modeling Hairy Plants,” In *Proceedings of Pacific Graphics 2004*, pp. 217-226, Seoul, Korea, October 6-8, 2004.

[6] Growing Tree, http://www.tutorialpedia.com/tutorial/growing_tree_animation_from_seed_to_giant_tree_using_particle_flow_8975.html

[7] J. Diener, L. Revéret, and E. Fiume, “Hierarchical retargetting of 2D motion fields to the animation of 3D plant models,” *In the Proceedings of the 2006 ACM SIGGRAPH/ Eurographics symposium on Computer animation*, pp. 187-195, Vienna, Austria, 2006.

[8] J. Thelen, “BlusteryTrees,” http://edgarapoe.home.mindspring.com/quixotic/blustery_trees.htm

[9] Peter A. Singh, Na Zhao, Shu-Ching Chen, and Keqi Zhang, “Tree Animation for a 3D Interactive Visualization System for Hurricane Impacts,” *In Proceedings of the IEEE International Conference on Multimedia & Expo (ICME)*, pp. 598-601, Amsterdam, The Netherlands, July 6-8, 2005.

[10] R. Woodland, “Filling the Gaps: Advanced Animation Using Stitching and Skinning,” *Game Programming Gems*, Edited by M. DeLoura, Charles River Media, pp. 480-483, Rockland, MA, USA, September 2000.

[11] NVIDIA CG Toolkit, http://developer.nvidia.com/object/cg_toolkit.html

[12] 3DSmax, http://www.autodesk.com/3dsmax

[13] Virtual Terrain Project, (VTP), http://www.vterrain.org

[14] OpenGL, http://www.opengl.org