Rendering Grass Terrains in Real-Time with Dynamic Lighting

Kévin Boulanger, Sumanta Pattanaik, Kadi Bouatouch
August 1st 2006
Goal:

Rendering millions of grass blades, at any distance, in real-time, with:

- Per-pixel dynamic lighting
- Grass density management
- Shadows
- Animation
- Anti-aliasing
Rendering at any distance

627 million grass blades on a football field
Previous work

• Geometry and billboards:
  – “Animating prairies in real-time”
    Perbet F. and Cani M.-P.

    ACM Interactive 3D Graphics, 2001

• BTF:
  – “Real-time rendering of realistic-looking grass”
    Shah M. A., Konttinen J. and Pattanaik S.

    Proceedings of the 3rd international conference on Computer graphics and interactive techniques in Australasia and South East Asia, 2005
Content

• Terrain subdivision
• Three levels of detail
• Grass density management and smooth transitions
• Shadows
Content

• Terrain subdivision
• Three levels of detail
• Grass density management and smooth transitions
• Shadows
Terrain subdivision

• Uniform subdivision of the terrain
• One grass patch per cell
• Random symmetries to remove repetitive patterns
• Quadtree structure (macro-cells for faraway grass)
Content

- Terrain subdivision
- **Three levels of detail**
- Grass density management and smooth transitions
- Shadows
Three levels of detail

- Near (geometry)
- Medium distance (vertical and horizontal slices)
- Far (horizontal slice only)
Nearby grass: geometry

- Textured quadrilateral strips
- Alpha channel to define the blade shape
- Alpha test, blending and multisampling
  → Order-independent rendering with anti-aliasing
Medium distance grass: slice generation

- Volume rendering using axis-aligned semi-transparent slices
- Slices data generation from a patch defined with geometry (preprocessing step)
Medium distance grass: lighting

- Slices data are BTFs (Bidirectional Texture Functions), allowing per-pixel lighting
- Five light directions
- Two view directions
- Inbetween light and view directions: interpolate BTF images (spherical barycentric interpolation)
Medium distance grass: transparency

- Alpha layer for semi-transparent slices
- Custom filter: to create mipmap images representing the BTF

Example
Medium distance grass: volume rendering advantages

- Good parallax effect (better than billboards)
- Accurate occlusions accounting for external objects
Faraway grass: horizontal slice

- Keep the horizontal slice of the previous method
- When patches are faraway from the viewer, render macro-cells (quadtree)
Content

• Terrain subdivision
• Three levels of detail
• Grass density management and smooth transitions
• Shadows
Grass density management

- Density = number of grass blades per surface unit
- A user-defined *density map* gives the density for each point of the ground (*local density*)
Density threshold

- Each patch of the terrain is made of the same number of blades
- Each grass blade of the repeated patch is assigned a density threshold in $[0,1]$.
- A grass blade on the terrain is rendered if its density threshold is lower than the local density of its root.
Grass density example

- Density threshold represented by a gray level (0 → black, 1 → white)

<table>
<thead>
<tr>
<th>Density</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>All grass blades are rendered</td>
</tr>
<tr>
<td>0.5</td>
<td>Every grass blade with a density threshold in [0,0.5] is rendered</td>
</tr>
<tr>
<td>0.1</td>
<td>Every grass blade with a density threshold in [0,0.1] is rendered</td>
</tr>
</tbody>
</table>
Density threshold for slices

- For slices, same algorithm but density threshold stored per texel
- Each texel covered by the same grass blade has the same density threshold (gray level)
- Some artifacts due to occlusions, not really visible since grass is quite far
Smooth transitions

Horizontal slices

Vertical slices

Geometry
Smooth transitions

Each LOD local density is weighted by a function of the distance from the viewer.
Content

- Terrain subdivision
- Three levels of detail
- Grass density management and smooth transitions
- Shadows
Shadows on the ground

- Projection of the grass blades onto the ground
- Drawn into the stencil buffer
- Then, black full screen quadrilateral with blending
Shadows from other blades

- Simple projection, shadow mapping and shadow volumes too expensive or not precise enough
- We need an approximation to allow real-time performances while keeping convincing results
Shadows from other blades: our method

- Each grass blade surrounded by a cylinder
- *Shadow mask* mapped onto the cylinder, representation of the occluders of the neighborhood
- Shadow mask computed once in a preprocess
- Equivalent to a visibility function of the environment
Shadows from other blades: our method

For each vertex:

• Send a ray to the light source
• Determine the coordinates of the intersection point with the cylinder

For each pixel:

• Compute shadow mask coordinates using interpolation
Speed results

- GeForce 7800 GTX, 1024x768, 4X antialiasing:

<table>
<thead>
<tr>
<th></th>
<th>Park scene</th>
<th>Football field</th>
</tr>
</thead>
<tbody>
<tr>
<td>Horizontal view</td>
<td>27 fps</td>
<td>18 fps</td>
</tr>
<tr>
<td>at low altitude</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Human height</td>
<td>80 fps</td>
<td>100 fps</td>
</tr>
<tr>
<td>Faraway view</td>
<td>150 fps</td>
<td>250 fps</td>
</tr>
</tbody>
</table>

- An infinite number of grass blades can be rendered in real-time (slight speed difference between 627 million and 11 billion grass blades)
Video
Contributions

• Dynamic lighting
• Three levels of detail with smooth transitions
• Volume rendering with axis-aligned slices, resulting in a convincing parallax effect, and per-pixel lighting using BTFs
• Custom filter to create mipmaps for slices
• Density management for the three levels of detail with user-defined density map
• Shadows from blades to blades
Current and future work

• Uneven terrains
  – Height map
  – Different space for all lighting computations
  – Vertical slices at the terrain silhouettes with smooth transition
  – Adaptive quadtree depending on the relief

• Wind simulation
Uneven terrain example
Questions ?