

Rendering Grass Terrains in Real-Time with Dynamic Lighting

Kévin Boulanger*

Sumanta Pattanaik†

Kadi Bouatouch‡

University of Central Florida, USA,
IRISA Rennes, France



Figure 1: From left to right: painted grass of a football field, seamless transitions between the three rendering methods, dynamic shadows on grass rendered using geometry, per pixel lit grass seen from a far viewpoint.

1 Introduction

We present a new approach to the rendering of realistic grass fields in real-time with dynamic lighting, shadows, antialiasing, animation and density management. We propose a method with three levels of detail, chosen depending on the distance from the camera: geometry rendering, volume rendering with per pixel lighting, rendering of 2D texture map with per pixel lighting. Usually, due to the expensive computations required to render large amounts of geometry for the grass blades, coarse approximations such as modeling using billboards are used in practice. Our approach, however, allows rendering of very detailed grass with shadows in close proximity from the viewer, and far grass with per pixel lighting and with a convincing parallax effect. Our method is based on the instancing of a patch of grass along a terrain, but does not exhibit repetitive patterns due to our simple aperiodic tiling scheme. Our management of grass density allows the rendering of grass patches of arbitrary shape at any distance from the camera and seamless transitions between the levels of detail.

2 Rendering method

The first level of detail, say geometry, is defined using semi-transparent quadrilateral strips textured using a scanned real grass blade, with the alpha channel defining its shape. These polygons can be rendered in batch since our blending approach is totally order-independent, fast and allows antialiasing if desired. The two other levels of detail make use of BTFs (Bidirectional Texture Functions) and image-based techniques. Each BTF stores the bidirectional reflectance properties of an axis-aligned slice of a patch of grass. Rather than using a single layer mapped onto a terrain as done in [Shah et al. 2005], we use multiple slices in a volume rendering approach. Very far grass is rendered by using only the horizontal slice of the volume. The BTFs are generated by rendering a patch of geometric grass in a preprocessing step.

For each level of detail, grass density is managed using a density map covering the terrain. Density management by simple manipulation of the opacity does not create natural looking images. To improve the results, we follow a threshold based approach where each blade of grass is assigned a density threshold. A blade of grass is rendered only when the local density taken from the density map

is greater than its threshold. By simply creating a hand-drawn density map, any shape of grass patch can be obtained. We also use this density management to achieve seamless transitions between the levels of detail by dynamically modulating the local density as a function of the distance from the camera.

Shadows play an important role to achieve realism in the final rendering. For the BTF based method, shadows are included in the reflectance data. For geometry, we use an approximation that is fast and visually convincing: a single precomputed shadow mask mapped onto a cylinder around each grass blade is used as a visibility function for the point light sources.

A football field covered by 627 million virtual grass blades (Figures 1 and 2) is rendered at 200 fps from a distance where the whole terrain is visible, 80 fps at human height, and 14 to 50 fps at very close proximity of the grass using a *GeForce 7800* graphics card, at a resolution of 1024×768 pixels and with $4 \times$ antialiasing.



Figure 2: Comparison between a rendered image of the football field with a real picture.

We are thankful to *Electronic Arts* for their valuable support.

References

SHAH, M. A., KONTTINEN, J., AND PATTANAİK, S. 2005. Real-time rendering of realistic-looking grass. In *Proceedings of GRAPHITE '05*, 77–82.

*e-mail: kevinb@cs.ucf.edu

†e-mail: sumant@cs.ucf.edu

‡e-mail: kadi@irisa.fr