Normal Mapping for Surfel-Based Rendering

Mathias Holst
University of Rostock
Albert-Einstein-Str. 21
18059 Rostock, Germany
mholst@informatik.uni-rostock.de

Heidrun Schumann
University of Rostock
Albert-Einstein-Str. 21
18059 Rostock, Germany
schumann@informatik.uni-rostock.de

ABSTRACT

On the one hand normal mapping is a common technique to improve normal interpolation of low tesselated triangle meshes for a realistic lighting. On the other hand today's graphics hardware allows texturing of view plane aligned point primitives. In this paper we illustrate how to use textured points together with normal mapping to increase surfel splatting quality, especially when using larger splats on lower level of detail. In combination with a silhouette refinement this results in a significant decimation of needed surfels with small visual disadvantages only. Furthermore, we explain how to create a normal map for points within a point hierarchy.

Keywords

Normal Mapping, Surfel Splatting, Point-Based Rendering, GPU-Programming.

1 INTRODUCTION

In recent years point-based rendering has been proven to be effective and efficient for rendering highly detailed complex geometric models. Point-based rendering bases on the idea, that polygonal representations get less efficient with increasing polygon number, because in this case each polygon covers only a few pixels in image space [LW85]. Additionally triangle meshes, or polygonal meshes in general, are not easy to handle and to simplify because of their connectivity. Points on the other side do not have any connectivity and can be stored and merged very easily using simple subdivision schemes [PGK02].

Since points only have a position but no dimension, they are parameterized with other attributes that describes their look. Usually these are a normal and a radius to represent circular disks in 3D (see fig. 1), known as *surfels* (from surface elements). With surfels a dense, opaque and smooth surface approximation can be described.

To get a high quality rendering result small and many surfels have to be used. However, the number of vertices (e.g. points) that is processed by the GPU is a framerate limiting factor. Thus, it is useful to render fewer but larger surfels instead. To attenuate the loss of surface features we propose to apply normal mapping. This is possible, because today's graphics hardware allows texturing point primitives. This texturing is given for image space only. Therefore we show in this paper how object-space texture coordinates can be obtained by adding only a 2D texture coordinate and a scaling factor as additional surfel attributes. Of course, normal mapping does not decimate the total number of pixels/fragments to shade but in the case of non-parallel vertex-fragment processing this results in a significant framerate increase.

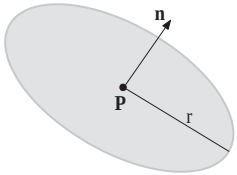

Figure 1: Surfel geometry.

When working with normal mapping normal maps have to be associated to the lower levels of detail of the original object. In this paper an algorithm is presented to get normal maps for surfels of point hierarchies. In past several point hierarchies has been developed. Therefore, we take the two most popular hierarchies into account: point and hybrid bounding-sphere hierarchies.

Since normal mapping only affects the shading of pixels, but not the shape of the underlying geometry, the silhouette looks very coarse when using low tesselated polygonal or point-based models. Therefore, it is useful to enhance the rendering process by silhouette refinement using more primitives at these surface parts. In contrast to polygonal-based approaches silhouette refinement can be easily done using point hier-