Every Last Detail: Density based level of detail control for crowd rendering

Micheal Larkin* Trinity College Dublin
Sebastian Paris† Trinity College Dublin
Simon Dobbyn‡ Trinity College Dublin
Carol O'Sullivan§ Trinity College Dublin

Figure 1: *Representations used: Geometrical level of detail, point based meshes, point cloud representation*

Keywords: point based rendering, level of detail, crowd rendering

1 Introduction

Rendering large crowds while maintaining realism is an important application in many fields. Performing this task at real time frame rates is a challenging task. We propose that instead of achieving real time frame rates by managing the level of detail of each character individually, high framerates may be guaranteed if detail is managed for crowds as a whole.

To render the higher levels of detail we use a hardware skinned pseudo-instancing method. We pre-create a number of batches that may be used to render geometry, up to a number that we know we can render at real time frame rates.

By limiting the resources that may be used to display each frame and distributing those resources among the crowd as a whole, we can guarantee that a crowd of any size may be rendered at interactive frame rates. The realism of this crowd may be maintained by sharing the available resources appropriately. Choosing points as the rendering primitive for lower detail representations allows us to distribute the available points among appropriate characters, something not possible with triangles, as they must maintain an explicit connectivity to represent an object.

As the lowest level of detail will only be viewed from great distances, we simplify rendering by creating a point based progressive mesh that averages the shapes of all the characters in our system. This allows us to update our point cloud structure very efficiently.

2 Method

To render our low level of detail crowd representation, we must first determine the appropriate level of detail at which to render agents. Performing a Delaunay triangulation of each agents position allows us to characterise the amount of personal free space for each agent. This amount of free space will be used directly to characterise the agent's grouping property: the more free space, the less the crowd density.

*e-mail:larkinmi@cs.tcd.ie
†e-mail:parissebastien@free.fr
‡e-mail:sdobbyn@cs.tcd.ie
§e-mail:carol.osullivan@cs.tcd.ie

I3D 2010, Washington, DC, February 19 – 21, 2010.
ISBN 978-1-60558-938-1/10/0002

In order to optimise the points distribution based on the distance to the camera, we propose to create groups of agents with decreasing levels of detail. To do so, we first order the triangulation's vertices by increasing distance from the camera.

We order the vertices instead of the agents for two reasons: first, the number of vertices is lower than the number of agents because of the grouping performed by the triangulation, thus reducing the computation time of the ordering operation; second, all the agents of a vertex share the same free space value and are almost equally distant from the camera, allowing the same point distribution to be used for all of them.

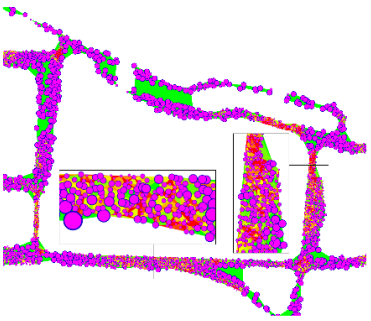

Figure 2: *Density Representation*

We precalculate the number of points that we may render at a real time frame rate. This number of points must be shared between the entire crowd. To achieve this we generate a point based representation that can be shared among all characters in the crowd, by averaging the shape of all character representations to be used. This point cloud has its points ordered from highest to lowest importance by removing nearest neighbours in order. Each character has colour variation, meaning the colour must be unique for each character. The colour combinations are pre-calculated and stored along with the character positions.

Density values are then used to render the crowd representations. Each vertex in the triangulation has a density value, which is used to calculate the number of points to render. This number of points is copied into the vertex buffer, along with the appropriate colours. This entire vertex buffer is then rendered in a single draw call, representing the entire low level of detail crowd.