

# High Quality Shadows for Real-Time Crowds

G. Ryder and A.M. Day

School of Computing Sciences, University of East Anglia, Norwich, NR4 7TJ, UK

---

## Abstract

*Real-time shadowed crowds, especially under locally lit environments, pose several distinct challenges. Current solutions for crowds are either very simplistic for multiple lights or only handle a single directional light with shadows. This work focuses on bringing hundreds of local lights with high fidelity shadows for crowds into the real-time domain. Each light is given the ability to cast shadows for all objects (dynamic and static) within the scene, and all the levels of detail used for representing the crowds correctly interact with the lighting conditions. In this paper we introduce depth augmented billboards for crowds, to allow for improved quality under local lights, including self-shadowing. Through use of the GPU fragment programs; this extension is entirely implemented on the graphics card for minimal performance impact.*

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism

---

## 1. Goal & Introduction

Local lighting, e.g. street lights, room lights, shop signs etc, feature heavily in real-life and has been largely ignored to date in real-time crowd rendering. In this paper we bring shadow handling for an extremely large number of lights, along with depth augmented billboards, to the field of crowd rendering to solve the problem of lighting crowds under common lighting conditions, for example street lights. We improve on existing image based rendering methods for crowds by introducing correct shadow casting and receiving, including self shadowing for all levels of detail.

## 2. Background

The presence of shadows serves two main purposes in crowd simulations. Firstly it visually locks the crowds' feet to the ground, without shadows the virtual humans often seem to float above the floor. The second purpose is to improve the visual realism of the simulation, as shadows play a significant part in real-world lighting. Loscos et al. [LTC01] present a planar based shadow method for image based crowds, within which they identify the following categories of shadows:

1. Shadows between static geometry
2. Shadows from static onto dynamic geometry
3. Shadows from dynamic onto static geometry
4. Shadows between dynamic objects

In the paper they proposed part solutions to 2 & 3 for a single fixed directional light. Dobbyn et al. [DHOO05] improve slightly on the method by the inclusion of the use of a stencil buffer to remove z-buffer fighting. Our research aims to allow for all of these cases, including higher visual quality solutions to the tasks tackled in [LTC01].

Nailboards presented by Schaufler [Sch97] introduced the idea of adding depth to impostors in a software render to allow for correct depth ordering. Shade et al. [SGwHS98] use warped "Sprites with Depth" as a simplified version of Layered Depth Images, with the goal of allowing for parallax effects with billboards. Policarpo and Oliveira [PO06] present a more general, though more computationally expensive method, for self shadowing impostors than the one presented in this paper.

Combining image based impostors and geometry based representations have been shown to be an effective way of merging the benefits of both methods for crowd rendering [RFD04, DHOO05, RFD05]. This allows for the lightweight representation of the crowd at a distance with image based rendering, combined with the high quality geometry based rendering close to the camera.

Planar shadows would not give the fidelity of representation desired and thus were discounted. Due to the use of billboard representations in our implementation, we are unable to use shadow volumes for all levels. Also their high fill rate con-

sumption would make them unsuitable for a crowd of a significant size being lit by a large number of lights. Our current work focuses on the use of shadow maps [Wil78] in rendering a large locally lit crowd scene.

### 3. Rendering Techniques

In the following sections, we present our methods for handling extremely large numbers of shadow casting lights upon crowds. These methods allow for per pixel lighting upon the maximum human level of detail, along with the automatic construction and use of "Depth Augmented Impostors" (Section 4) for improved lighting of billboard impostors under positional lights.

#### 3.1. Level of Detail for Shadow Maps

With each light having the ability to cast dynamic shadows, a naïve solution would be to assign a separate off-screen depth buffer to each streetlight to use as a shadow map. However context switching to an off-screen buffer is expensive and repeating this operation a large number of times per frame would cause a significant impact on the total frame time. Thus it would be advantageous to bind one large shadow map which is shared between many (or all) of the lights in the scene. Our method is to "tile" smaller shadow maps into a larger map, only using a single off screen rendering target for all streetlights in the scene. Our test implementation divided a 2048x2048 shadow map in 31 sections, ranging from 1024x1024 to 128x128, which was a sufficient number for the test scene without popping.

The overhead of using shadow maps comes in two areas, creating the shadow map and using the shadow map in the lighting calculations. In our system the off screen render target is bound once to draw and again once to be used, thus giving a significant saving on state changes. The level of detail algorithm dramatically reduces the fill rate required to fill the buffer compared to a 1024x1024 shadow map per visible light. The results section shows that it can maintain real-time frame rates for a large number of shadow casting lights (over 30 in view at any one time). The method can be implemented upon the current fixed function pipeline, with only the extra overhead of loading a 4x4 matrix per light.

A depth-sorted list of the lights within view is maintained. Due to the spatial coherency of consecutive frames, the overhead of maintaining this list is minimal, as there is no need to recalculate the distance for every light each frame (also as it is only for comparison, distance squared is sufficient). There is the further advantage of drawing in a loosely sorted front to back order, saving on fill-rate by reducing overdraw. In the shadow map construction, the closer the light is to the camera, the higher the precision of the shadow map sub-segment assigned to it.



**Figure 1:** Looking along a street, with many street lights in view

#### 3.2. Large number of real-time local lights

In a system with a freely moving camera, allowed to raise above building level, many hundreds of lights may come into view at any one time. Even if the camera is restricted to ground level, it is still possible for a large numbers of lights to be in view, as shown in Figure 1. For each street light in the scene, all the triangles in the scene are tested against the cone of influence of the light (a future optimization would be to pre-process the scene into a spatial data structure, however triangle in cone tests are relatively cheap, taking under 5 seconds for all 258 lights against 175,000 triangles). For all those that fall within the cone of light, an OpenGL display list is created and associated with the streetlight. This is sufficient for the static geometry; however it will not identify which people are visible to the light.

The human behaviour system uses a regular grid for collision detection between the virtual humans. This is exploited at run time to identify which humans fall within each light's influence. All grid cells that fall within the light are stored in a list for each light. Two final lists are built to hold the items not contained in the other lists, one for the scene polygons and the other for the grid cells.

To render, the global direction lights are enabled and an OpenGL light is placed at the first light's position. The display list associated with each light is called along with all humans in the grid cells within the light's list. The light is then moved to the second light position and the process repeated for all lights.

It is clear that any triangles that fall within two (or more) cones of influences are drawn multiple times (blending will sum the contributions). However, the number of triangles drawn multiple times is low in comparison to the number of triangles in the scene. Depending on the number and state of these triangles, it may well be more expensive to handle these triangles separately in a single (with multiple lights enabled) pass than to draw them in multiple blended passes, as visually the results are the same.

#### 4. Depth Augmented Impostors

Lighting artifacts are one of the major pitfalls of a multiple level of detail system, "pops" in the lighting often clearly identify when a change in representation has occurred. Normal maps can only be used to correctly relight billboard impostors under directional light, due to light direction being invariant to world position. Under positional lights, e.g. street lights or car headlights, the position of the fragment, in addition to the fragment normal, affects the amount of light received. This leads to lighting discrepancies when changing from geometry based representations to a billboard impostor.

Through high level fragment shaders, all the information needed to reconstruct the true position for each point can be made available in the fragment shader. The position of the pixel in the 2D plane is easily calculated, through interpolating the vertex positions of the billboard. It is this position that is displaced along the billboard's normal by a scalar amount.

When the billboard is created, the depth values of each pixel are stored encoded in the range 0-255, along with the maximum depth and the depth range.

$$DCP = P + (MD - DS * DR) * BN \quad (1)$$

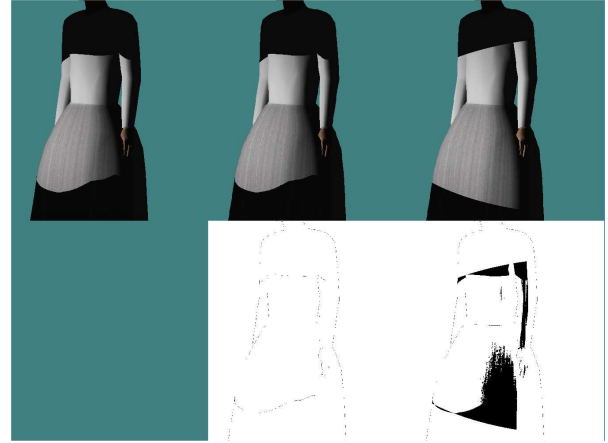
Where:  $DCP$  = Depth Corrected Position,  $P$  = Position of Fragment,  $MD$  = Max Depth,  $DS$  = Depth Scale,  $DR$  = Depth Range,  $BN$  = Billboard Normal

At run time, in the fragment shader, an implementation of Equation 1 is used to reconstruct the pixel position for the lighting calculations. This correction comes at nominal extra work and storage. The depth value can be added as an alpha channel to the normal map texture, causing only a small increase in texture overhead, and no increase in the number of texture accesses.

Figure 2 shows that with depth correction the billboard impostor almost entirely matches the lighting characteristic of the geometry representation model. The geometry model (top left) clearly distorts the outline of spot light due to the differences in depth in the model. On a flat billboard (top right), the light forms an elliptical boundary. However with the added depth information (middle), the billboard almost exactly matches the geometry. On examining the pixel difference between the two images (bottom middle), it is clear the majority of the errors present are due to aliasing. Thus a depth augmented billboard under a positional light will receive light correctly.

##### 4.1. Shadow receiving & Self Shadowing

Non augmented billboards suffer from "popping" into and out of shadows when traversing shadow boundaries parallel to its orientation. To overcome this problem the depth adjusted value can be used in the shadow receiving code, allowing for correct shadowing. Also without depth augmentation, self shadowing billboards are not possible. The methods presented above will



**Figure 2:** Comparison under a positional spot light of a geometry model (top left), depth augmented billboard (top middle) and a standard normal mapped billboard (top right). Also presented are the per-pixel difference of the geometry model and the depth augmented billboard (bottom middle) and the difference between the geometry model and the standard normal mapped billboard (bottom right)

allow for self-shadowing, if the shadow map contains the correct depth values for a virtual human. If the billboard is drawn into the shadow map in its native form, the values written to the depth buffer are of the polygon that holds the texture, not the values of the model itself. Therefore a second shader is required for the shadow map construction. The shader is based upon the depth augmented method, however it requires no lighting calculations as only depth is written during this pass. The adjusted pixel position is calculated in the same way and is outputted to the depth buffer.

Figure 4 demonstrates the difference between the geometry (top left), a depth augmented billboard (top middle) and a traditional normal mapped billboard (top right). To help clarify the differences between the images, two further sections of the image have been included. A per pixel difference between the geometry model image and the depth augmented billboard (bottom middle), and the pixel difference between the geometry model and the standard billboard (bottom right). Without self-shadowing, significant differences appear around the nose, side of head and arm when comparing the billboard to the geometry model. The depth augmented billboard is clearly displaying self shadowing and without studying the per pixel difference image, the differences are not immediately apparent. The image comparison given does indicate that due to inherent aliasing problems associated with billboards, on the boundaries of the shadows and the silhouette there are differences. This demonstrates that depth augmented billboards, when combined with two appropriate shaders (one for the

shadow map generation pass and another the main rendering pass), can handle self-shadowing and other lighting situations.

## 5. Results

The techniques present in this paper are designed to facilitate real-time rendering of a city scene with lights. For this reason the test scene chosen to test the methods is based on a real city. The scene is a 500m x 500m tile of a city, generated automatically from land line data, consisting of 174,690 triangles.

Within the scene there are two global directional lights (to give the non spot lit areas a low level of lighting) and 258 positional shadow-casting streetlights. The crowd consists of 1000 virtual humans, with 7,000 triangles for the top detail level, 3500 triangles for the middle and 2 triangles for the third level. This high level of complexity for the top mesh allows for high quality deformations.

The shadow requirements have been met by all the levels of detail in framework, including the billboard impostor level. Table 1 shows the frames per second of the multiple paths through the scene. All paths maintained average frame rate comfortably above the goal of 25 frames per second. Occlusion culling has not been included in the results present here, to stop the algorithm chosen heavily colouring the results.

	Avg. FPS	Min FPS	Max FPS	Avg. Lights in View
Path 1	42	24	80	105
Path 2	38	27	53	136
Path 3	50	37	85	58

**Table 1:** Frame rate comparisons for the three paths



**Figure 3:** Example of a geometry avatar showing self-shadowing and casting a shadow onto an un-even floor

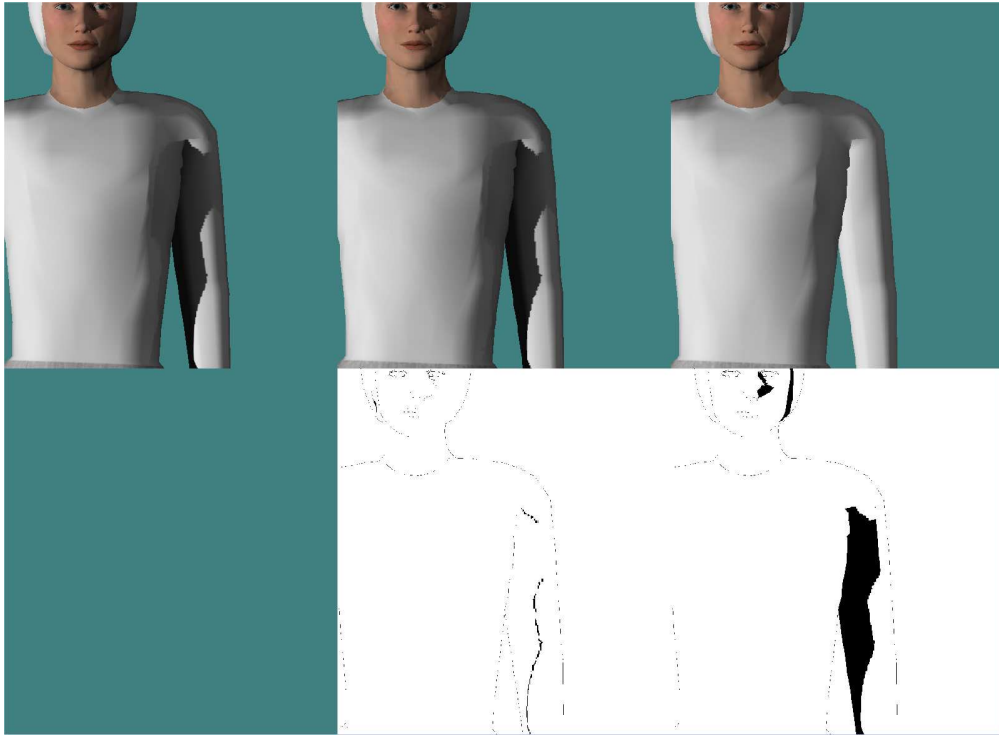
## 6. Conclusion

There were a number of visual elements that this work has addressed. The four types of shadowing as stated by [LTC01] which the framework aimed to solve, along with an additional condition added, that of self-shadowing have been successfully incorporated into crowd rendering in real-time for large numbers of lights. The depth augmented billboard impostors help lift the quality of the lowest level representation, allowing the change over threshold to be closer than it would otherwise. Using OpenGL lights for each spotlight allows for full flexibility for dynamic effects and the integration with other advanced lighting techniques. The shadows help lock the crowd onto the ground, and due to all dynamic and static elements sharing the same lighting conditions there is no feeling of the humans being "superimposed" upon the scene.

Future work will include examining adaptive subdivision of the shadow map to handle different light configurations, along with examining methods for efficiently including dynamic lights (e.g. car head lights).

## References

- [DHO05] DOBBYN S., HAMILL J., O'CONNOR K., O'SULLIVAN C.: Geopostors: a real-time geometry/impostor crowd rendering system. *ACM Trans. Graph.* 24, 3 (2005), 933–933.
- [LTC01] LOSCOS C., TECCHIA F., CHRYSANTHOU Y.: Real-time shadows for animated crowds in virtual cities. In *Proceedings of the ACM symposium on Virtual reality software and technology* (2001), ACM Press, pp. 85–92.
- [PO06] POLICARPO F., OLIVEIRA M. M.: Relief mapping of non-height-field surface details. In *SI3D '06: Proceedings of the 2006 symposium on Interactive 3D graphics and games* (New York, NY, USA, 2006), ACM Press, pp. 55–62.
- [RFD04] RYDER G., FLACK P., DAY A.: Adaptive crowd behaviour to aid real-time rendering of a cultural heritage environment. In *Vast 2004* (2004), Chrysanthou Y., Cain K., Silberman N., Niccolucci F., (Eds.), pp. 29–36.
- [RFD05] RYDER G., FLACK P., DAY A.: A framework for real-time virtual crowds in cultural heritage environments. In *Vast 2005, Short Papers Proceedings* (2005), M. Mudge N. R., R S., (Eds.), pp. 108–113.
- [Sch97] SCHAUFLE G.: Nailboards: A rendering primitive for image caching in dynamic scenes. In *Proceedings of the Eurographics Workshop on Rendering Techniques '97* (London, UK, 1997), Springer-Verlag, pp. 151–162.
- [SGWHS98] SHADE J., GORTLER S., WEI HE L., SZELISKI R.: Layered depth images. In *SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1998), ACM Press, pp. 231–242.
- [Wil78] WILLIAMS L.: Casting curved shadows on curved surfaces. In *SIGGRAPH '78: Proceedings of the 5th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1978), ACM Press, pp. 270–274.



**Figure 4:** Comparison of self shadowing on a geometry model (top left), depth augmented billboard (top middle) and a standard normal mapped billboard (top right). Also presented are the per-pixel difference of the geometry model and the depth augmented billboard (bottom middle) and the difference between the geometry model and the standard normal mapped billboard (bottom right)