Three LOD for the Realistic and Real-Time Rendering of Crowds with Dynamic Lighting

Jean-Marc Coic, Celine Loscos, Alexandre Meyer

Three LOD for the Realistic and Real-Time Rendering of Crowds with Dynamic Lighting

Jean-Marc Coic[δ] Celine Loscos[δ] Alexandre Meyer [ω]

[δ] c.loscos-at-cs.ucl.ac.uk, Department of Computer Science, University College London, UK

[ω] Alexandre.Meyer-at-liris.cnrs.fr, LIRIS Lab, Universit Cflaude Bernard, Lyon, France.

Abstract

Populated urban environments are very important in many applications such as urban planning and entertainment. Techniques that populate a 3D virtual city with hundreds of pedestrians walking in real-time have previously been developed using image based rendering techniques. However, when visualising these avatars nearby the viewpoint, such methods provide a poor rendering quality. In this paper, we propose a method based on three rendering techniques used as levels of details. We provide continuous transitions between them by maintaining a similar dynamic lighting on each level. For closed objects, we use a polygonal representation. At the intermediate level, we adapt the existing layered impostor technique to animated objects and we add a dynamic lighting. As the object gets further, this layered representation is used with a decreasing number of layers to reach the third level, a one-polygon impostor. Our technique contributes to improve the realism of virtual humans in a crowd while maintaining an interactive frame rate.

I. INTRODUCTION

Populated urban environments are very important in many applications such as urban planning, entertainment, architectural or cultural applications. In this context, a single realistic and animated character (or agent) with which the users can interact are becoming more and more common. On the other side of the scale, the rendering of crowd with hundreds of virtual humans is becoming more and more efficient, specially with Image-Based Rendering techniques. Indeed, Image Based Rendering methods simplify the model by using images instead of the geometry. In the context of crowd rendering, previous techniques use **impostors**, which is a quad mapped with a texture representing the model. Using this method, it is possible to render hundreds pedestrians in real-time, as has been demonstrated in [TC00].

Although the use of an impostor is appropriate when the objects are far from the viewpoint, certain issues appear when the viewpoint is closer to objects. First, it becomes apparent for the user that a flat impostor is used instead of a real geometric model. Second, the pixel may become visible. Increasing the number of view and the resolution of the texture is possible but it will not be possible to store the required textures in the GPU memory. Furthermore, for very close view, it could be interesting to switch to a geometrical representation. Dobbyn *et al.* in [DHCS05] presented recently an interesting paper going in this direction of using LOD in crowd rendering. Indeed, they provide an efficient technique based on a ”pixel to texel” ratio to switch from geometry to impostor representation.

The method presented in this paper add a third LOD in this crowd rendering scheme to interactively render crowd from near to far viewpoints.

II. Previous Work

Tecchia *et al.* in [TC00], [TLC02] present an Image Based Rendering technique to achieve the real-time rendering of hundreds pedestrians. Each pedestrian is represented by an one-polygon impostor, pre-calculated from sample views around the upper hemi-sphere for each step of the walking movement. The right texture to use at a given time is determined by the angle between the direction of the normal of the polygon supporting the impostor and the camera and the current phase of the walking process. Thus, it is possible to render hundreds pedestrians in real-time, as has shown figure 1 on the right. However, for the avatar near the viewpoint, these methods provide a poor visual aspect because of the impostors flat aspect and the lighting conditions fixed on the image as shown on figure 1 on the left. Loscos *et al.* in [LCT01] propose a technique for generating the shadows of the pedestrians cast on the ground. In a second pass the model is rendered projected on the ground and drawn in black . These fake shadows are valid only on a ground with a parallel light source but give realistic results in the available computing time.

Fig. 1. Previous impostor techniques from Tecchia *et al.* [TLC02] allows large number of pedestrians in real-time. However, human near the viewpoint has a poor visual aspect.

Aubel *et al.* [ABD99] present a technique to render a virtual human based on multi-planed impostor where each part of a body is rendered by a textured polygon. Since it is a one level model, it would be difficult to integrate it in our hierarchies of LODs (see section V) for the safe of continuity.

Dobbyn *et al.* presents recently in [DHCS05] a novel hybrid rendering system for crowds that solves the problem of degraded quality of image-based representations at close distances by building an impostor rendering system on top of a full, geometry-based, human animation system. This enables almost imperceptible switching between the two representations based on a pixel to texel ratio, with low popping artefacts. The shading of the impostors and the introduction of variety into

the virtual humans geometric and impostor model is achieved at run-time through programmable graphics hardware. Instead of 2 LOD as Dobbyn *et al.* proposed, we use 3 LOD. Indeed, we think that for some angles of view, the visual gap between flat impostor and geometry is so big to completly avoid popping artefacts. We then introduce a volumetric layered based impostor between flat impostor and geometry to help to achieve continuity during transitions.

In [Sch95], Schaufler presents a dynamic image based method to render static objects. The impostor's image is generated between two frames only when necessary, reusing the most possible number of impostor in next frames. This dynamic generation provides low popping effect and decreases texture memory consumption. Later, Schaufler [Sch98] demonstrates the benefits of saving the depth of each texel of the colour texture generated. Instead of one single textured polygon, several layers of the colour texture are drawn, depending on the texel's depth. These layers fill a volume in the 3D scene instead of a single polygon. From a close point of view, the layered impostor gives a realistic 3D effect. However, to be able to keep an impostor along several frames, the object has to be static which is not our case.

The same year, Meyer *et al.* [MN98] presents a method consisting in slicing 3D geometry into a series of thin rectangle containing the shaded geometry that falls in that slice. These layers are used as volumetric texture on a surface. This layers are precomputed and cannot be built on the fly. So the texture memory consumption is too high to apply this technique on animated humans.

In the work of Decoret *et al.* [DSSD99], the rendering method design for fixed objets like facade is a combination of pre-generated and dynamically updated image-based representations. By capturing the relevant depth information in the model, the occlusion artifacts between impostors are reduced.

Many works exist on human and crowd simulation [Don01], [UT02], [LMM03]. As our goal in this paper is realistic rendering independent of the global behavioural simulation, we will not describe them further.

III. Overview

Our goal is to improve the realism of humans when the camera is close and to maintain a fast rendering process for far away humans. For that, we adapt in section IV the layered impostor technique to animated humans and we add dynamic lighting. This allows us to build in section V a set of three representations as various LODs: geometry, adapted layered impostors and flat impostors. Each of these three representations may also contain sub-level of details to form a continuous hierarchy of LODs. As result in section VI, we obtain realistic images of crowd from close to far distance with continuous transitions while maintaining interactive frame rates.

IV. Crowd Layered Impostor with Dynamic Lighting

At the intermediate LODs, we extend the single-polygon fixed-shaded impostors by our dynamically-shaded layered impostors approach, thus enhancing the texture information with depth and normal. Depth information will help to give a 3D aspect to the impostor and the normal will be used to compute a per pixel dynamic lighting.

The Schaufler's layered impostor technique [Sch98] is dynamic, i.e. when the limit of validity of a view is reached, the algorithm renders a new views of the object on the fly. Since we deal with animated objects -walking humans-, this approach would lead to render a new view each time the human moves. Instead, we choose a precomputed approach based on [TC00]. We consider sample views around the upper hemisphere for each step of the walking movement of the pedestrian as shown on figure 2 *(a)*. The appropriate view to use at each frame is determined by the angle between the direction of the impostor and the camera position, and by the current phase of the walking process.

For each direction, we need to have several images as shown on figure 2 storing the colour, the depth and the normal vector of each pixel. The colour image represents the object without any shading since it will be computed on the fly with the normal informations. For generating these images, we place the viewpoint in the desired position, we draw the object and save the resulting image.

A first approach is to store these three images as three textures. Due to hardware constraints, Schaufler [Sch98] stores the depth information into the alpha channel of the RGBA colour texture and is limited to 1,2,4,8, 16 or 32 layers. Today, with the possibilities of fragment programs and multi-texturing, we have more flexibilities. The number of layers is no more a constraint but the texture memory limitation is still an issue. In our case, storing the colour information and the normal information into the same texture avoids two rendering passes during the preprocessing. With the goal to save a maximum of texture memory, we can store the colour on 16 bits (5 bits per components) in the RG channels, the normal in the B channel and the depth in the alpha channel. We then fit in 4 bytes per pixel. In choosing this packing scheme, there is a trade off between the texture memory consumed and the precision of the data. An other scheme consuming more memory but providing more precision would be to store the normal in the alpha channel and the depth in another 1 byte per pixel texture with the multi-texturing.

We render the impostor in layers parallel to the viewpoint. The process of choosing the number of layers will be discussed in the next section. For each of these layers, we select the pixels that correspond to a certain depth. After selecting the required number of layers, we can divide the volume captured during the preprocessing in as many intervals as the number of layers, defining the intervals of depth for selecting pixels in the colour texture. By this mean, we reconstruct the initial volume

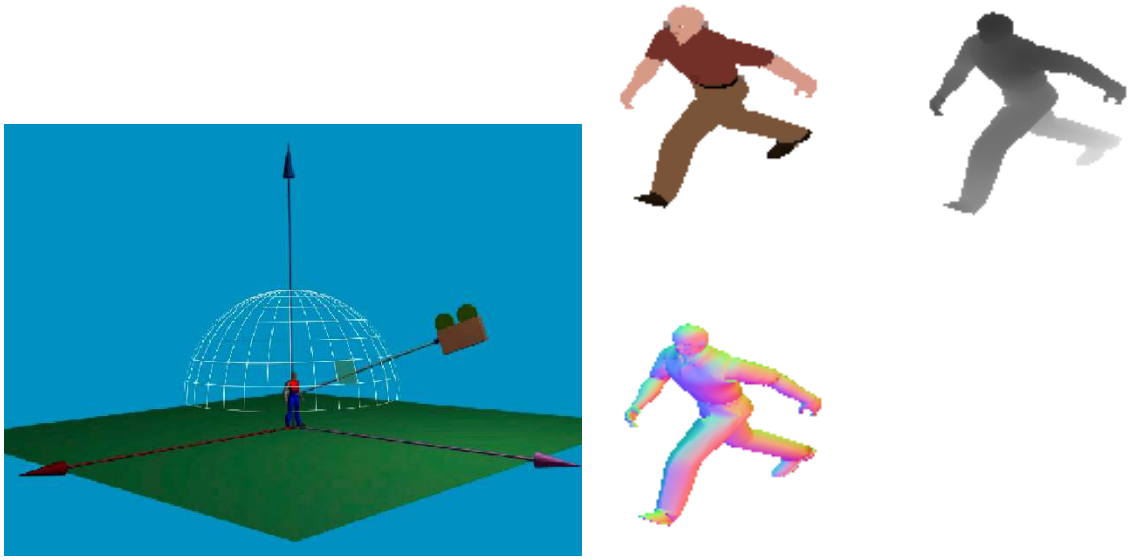

Fig. 2. For each keyframe of the animation, we sample the upper hemisphere (*left*). For each sample, we compute a view (*right*) with the colour (*top left*), the normal vector (*bottom*) and the depth information for each pixel (*top right*).

with several layers. The selection of the right pixels for each depth interval is done using the fragment program or the registers combiners functionalities. During this per pixel selection, we also compute the lighting (Phong model) according to the normal vector stored for each pixel. To extend the validity of the layered impostors, we can use the overlapping depth intervals. Without overlapping, cracks appear on the layered impostor as soon as the viewpoint differ a little bit from the pre-computed one (Figure 3 *left*). By drawing a small part of the previous and next layer, we can avoid these gaps from more farther viewpoints (Figure 3 *right*), extending the lifetime of the layered impostor and decreasing the density of precomputed views.

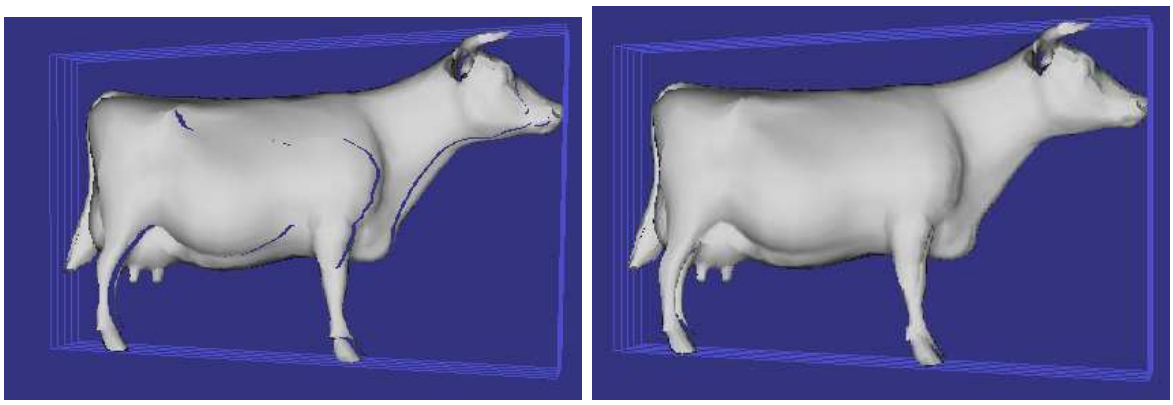

Fig. 3. A cow rendered with 5 layers and our dynamic lighting, without (*left*) and with overlapping (*right*).

V. Level of Details

To maximize the visual quality of the simulation while keeping an interactive frame rate, we introduce an hybrid system based on a set of LODs as described in figure 4. We use three representations where each of them may contains also some sub-level of details. For close view, we render the polygonal model, possibly with LODs computed by mesh simplification [Hop97]. In the mid-distance, we used layered impostors with fewer and fewer layers as the distance increases, possibly until just one layer. This single polygon for far-distance can also be mip-mapped. We maintain a visual continuity between these levels because of the dynamic lighting present in each technique.

To well synchronise the transitions in this hierarchies, we need criteria to choose how many layers are drawn and when the layered representation has to be replaced by the polygonal representation.

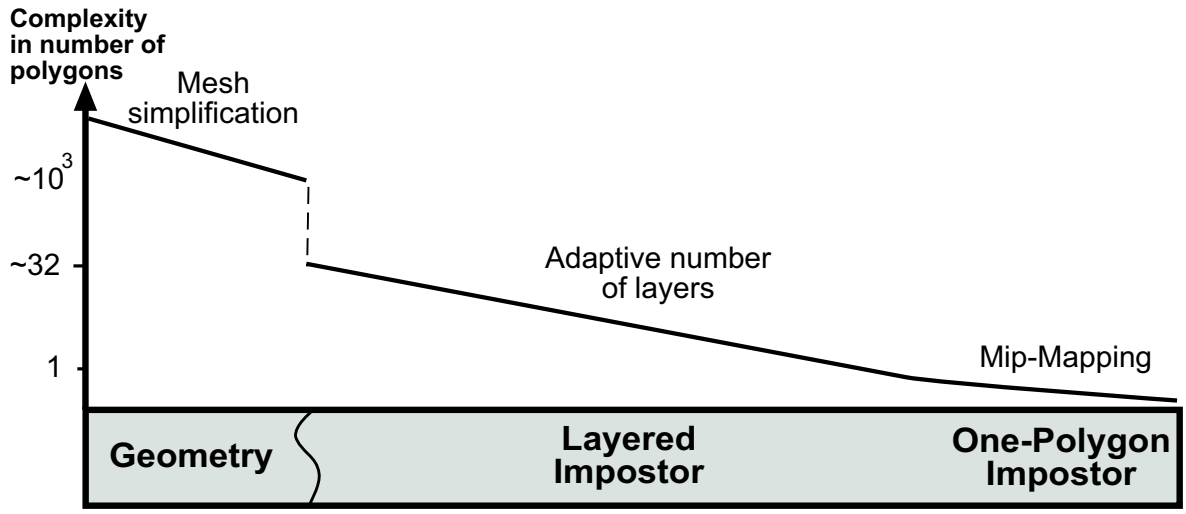

Fig. 4. Our three representations used as levels of detail.

Number of Layers and Precomputed Data: Schaufler in [Sch98] gives a criterion to decide how many layers has to be drawn. This criterion is based on how many pixels can be seen between two layers inside the object. This number of pixels depends on the angle of view and on the distance. We use a similar criterion during the rendering to choose the number of layers drawn.

As previously mentioned, Schaufler's technique is dynamic and a new view of the object is computed when this criterion demands very large number of layers. Since we deal with animated objects for which a dynamic approach would be inefficient, we precompute the different views (See section IV). Nevertheless, we use the Schaufler's criterion to decide how many precomputed views are needed. For that the user chooses an overlapping size he judges acceptable. There is a trade off between quality and texture memory consumption: small overlapping means good quality but with many precomputed views and vice versa.

Transition: It would be interesting in terms of frame rate to use the layered representation as much as possible. Nevertheless, viewing layered impostors from a close viewpoint leads to notice the discrete resolution of the textures. Indeed, when the pixel size of the drawn layered impostor is greater than the texture resolution, a texture pixel covers several pixels on the screen, decreasing the visual quality. This gives us a criterion for choosing between polygonal representation and layered impostors: when the projected pixel of the layer is greater than one screen pixel, we switch to the polygonal representation. This criterion depends on the distance of the object and on the resolution of the texture.

VI. Results

Since our target was the most common PC, all the implementations and tests were realized on a Intel Pentium 4 2.0 GHz laptop with the register combiner functionality of a Nvidia Geforce 4 Go 4200 with 64 MB of video memory. We sample the sphere by 256 viewpoints giving a texture of 2048×512, since each viewpoint gives an image of 64×64. With the compression on the GPU, the 8 steps of the animation consume $8 \times 512\text{KB} = 4\text{MB}$ of the texture memory.

As we consider the polygonal representation as the best quality representation, we have measured the visual quality of our system of LODs. We have compared the quality of the layered impostor representation and a polygonal-only rendering, as show on figure 5 and the video provided with the paper. The main differences are on the silhouette due to the small resolution of our texture. Notice that on recent graphics cards this resolution can easily be increase.

We continuously render from 100 to 1000 humans in a field with each of the different rendering method separately, from a fixed viewpoint. The polygonal mesh of the humans contains about 5000 triangles. For comparison, we also run the test with the one-polygon method for render crowd. Figure 6 presents the results. The layered impostor are really faster rendered than the polygonal representation. The cost of layered impostor compared to the one-polygon impostor is a loss in the frame rate of about 10%. Thus, it is really a good intermediate representation.

VII. Conclusion

At the cost of few additional data per pixel - normal vector and depth value - we have improved the quality of a crowd simulation. We have filled the gap which existed between the polygonal representation and the one-polygon impostor in the context of animated humans rendering. We have added a dynamic lighting to the layered impostors technique to be able to use it as an intermediate level, providing continuous transitions from near to far viewpoints. Our global set of LODs allows interactive frame rates for the rendering of hundreds of realistic humans.

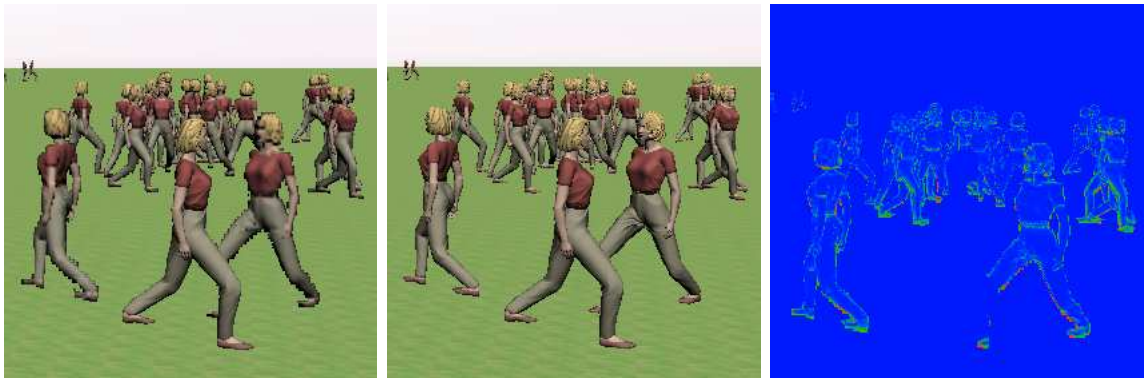

Fig. 5. Comparison (*right*) in CIE XYZ space between the layered impostors technique (*left*) and the only-polygonal rendering (*middle*).

In the future, we would like to improve the diversity of humans by adding dithered impostor colours like in [TLC02] and [DHCS05]. The second point is to add shadows of humans on other humans and on buildings. Finally, we need to work on the behaviour.

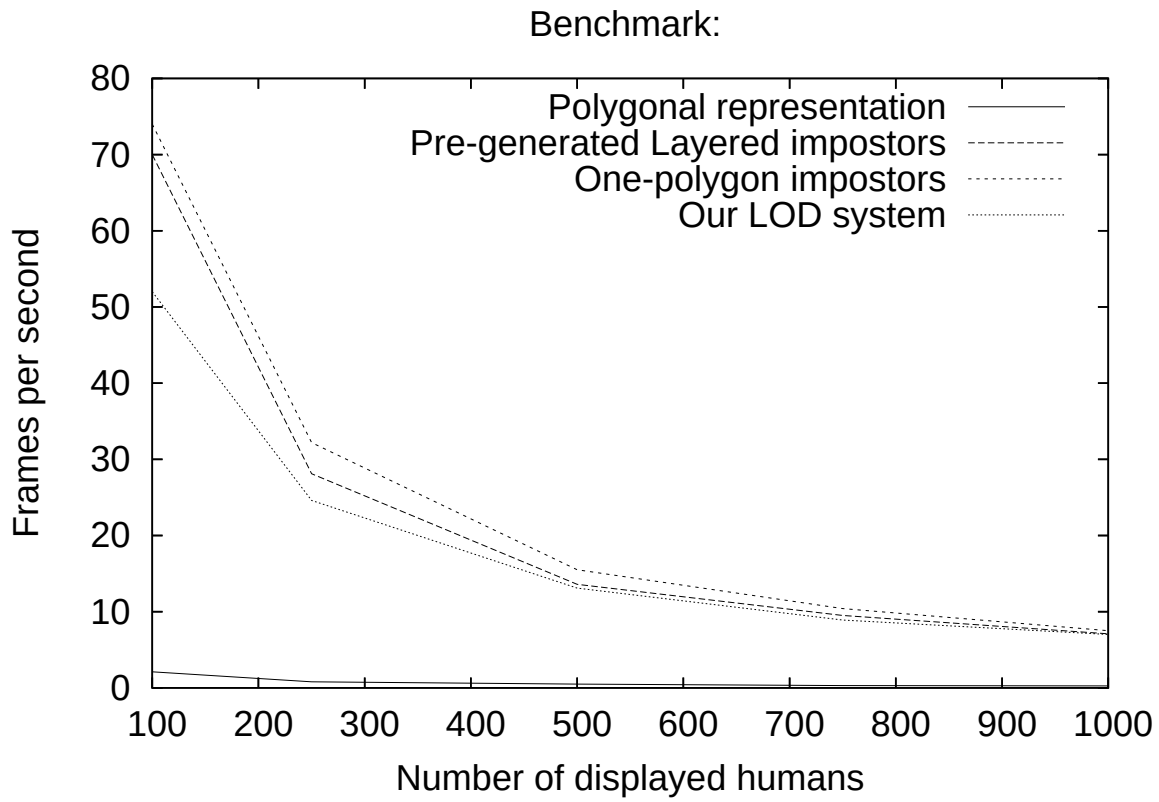

Fig. 6. Frame rates depending on number of humans and rendering method.

Fig. 7. Crowd with close view on human.

REFERENCES

[ABD99] AUBEL A., BOULIC R., D.THALMANN: Lowering the cost of virtual human rendering with structured animated impostors. In *Proc. WSCG* (1999).

[DHCS05] DOBBYN S., HAMILL J., CONOR K. O., SULLIVAN C. O.: Geopostors: A real-time geometry / impostor crowd rendering system. In *Symposium on Interactive 3D 2005* (2005).

[Don01] DONIKIAN S.: Hpts: a behaviour modelling language for autonomous agents. In *Fifth International Conference on Autonomous Agents* (2001).

[DSSD99] DCORET X., SCHAUFLER G., SILLION F., DORSEY J.: Multi-layered impostors for accelerated rendering. In *Computer Graphics Forum (Eurographics)* (1999).

[Hop97] HOPPE H.: View-dependent refinement of progressive meshes. In *SIGGRAPH 1997* (1997).

[LCT01] LOSCOS C., CHRYSANTHOU Y., TECCHIA F.: Real-time shadows for animated crowds in virtual cities. In *ACM Symposium on Virtual Reality Software and Technology* (2001).

[LMM03] LOSCOS C., MARCHAL D., MEYER A.: Intuitive crowd behaviour in dense urban environments using local laws. In *Eurographics UK* (2003).

[MN98] MEYER A., NEYRET F.: Interactive volumetric textures. In *Eurographics Workshop on Rendering* (1998).

[Sch95] SCHAUFLER G.: Dynamically generated impostors. In *GI Workshop on Modeling, Virtual Worlds, Distributed Graphics* (1995).

[Sch98] SCHAUFLER G.: Per-object image warping with layered impostors. In *Eurographics Workshop on Rendering* (1998).

[TC00] TECCHIA F., CHRYSANTHOU Y.: Real-time rendering of densely populated urban environments. In *Eurographics Rendering Workshop* (2000).

[TLC02] TECCHIA F., LOSCOS C., CHRYSANTHOU Y.: Image-based crowd rendering. *IEEE Computer Graphics and Applications* (2002).

[UT02] ULICNY B., THALMANN D.: Towards interactive real-time crowd behavior simulation. In *Computer Graphics Forum* (2002).