Meshless Statistical Occlusion Computation

François Fouquet, Jean-Philippe Farrugia, Jean-Claude Iehl, Jonathan Dupuy

Université de Lyon, CNRS
Université Lyon 1, LIRIS, UMR5205, F-69622, France

Abstract

This paper presents a new method to evaluate visibility from a point cloud by using a meshless statistical representation. The resulting evaluation uses much less memory than previous work while still producing high quality images, making it suitable for memory limited systems. The obtained results are exposed through the evaluation of shadow maps using a point cloud created from range images and are compared to current methods, which rely on mesh reconstruction techniques.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation—

1. Background

In augmented reality, the shadowing of the inserted 3D objects produced by the captured environment greatly contributes to the quality of the simulation. Naturally, such an effect requires the scene information to be available in 3D. Most existing methods extract meshes from point clouds generated by range images, which then allows one to use polygon-based rendering methods to determine the virtual object's final appearance. Although such methods can be executed in realtime, the meshing pass can be very memory intensive, which makes them incompatible with memory limited platforms, such as mobile phones.

In this work, we use the point sets to build statistical –rather than polygonal– surfaces. Once the points are acquired, we generate a set of trivariate normal distributions which locally approximates the surface geometry of the scene. A simple raytracing-based method then determines if the scene is occluding the object or not. Additionally, we show how to efficiently support incremental refinement by using a space partitioning algorithm, such as a KD-tree. Overall, obtained results show that the method consumes much less memory than previous work while still producing convincing results.

2. A Statistical Model of Construction

Instead of building an explicit polygonal surface from a point cloud or using direct visibility computation [KTB07], we try to fit a trivariate normal distribution function which approximates the whole set. When the point set becomes very dense, a single distribution only provides a very crude approximation. Hence, we use a spatial sorting algorithm to generate multiple distributions from subsets of points which are spatially near from each other.

To construct these distributions, each point is inserted into the root node of a KD-tree. While descending in the tree, mean vector $\vec{\mu_n}$ and co-variance matrix Σ_n of local trivariate normal distribution $\mathcal{N}(\vec{\mu_n}, \Sigma_n)$ are evaluated. Note that such a scheme allows incremental refinement and level-of-detail management since a node holds an approximate distribution of its child nodes. When the point reaches a leaf node, it is discarded. The subdivision process is controled using a cell size criterion, which also limits the memory footprint of the complete tree. This part of our work is very similar to the one of Pajarola [Paj03].

3. Using a Statistical Model in a Rendering Process

Once the distributions are built, we use them to determine the amount of occlusion found along a view ray. The occlusion factor is evaluated by accumulating the maximal value of a univariate normal distribution, generated from a condition on the two other dimensions which are evaluated on the plane normal to the view ray. This process is illustrated in Figure 1. This process does not produce an exact occlusion

DOI: 10.2312/conf/EG2012/posters/023-024

factor, but rather a probable occlusion factor, that is used to compute the final appearance of our 3D models.

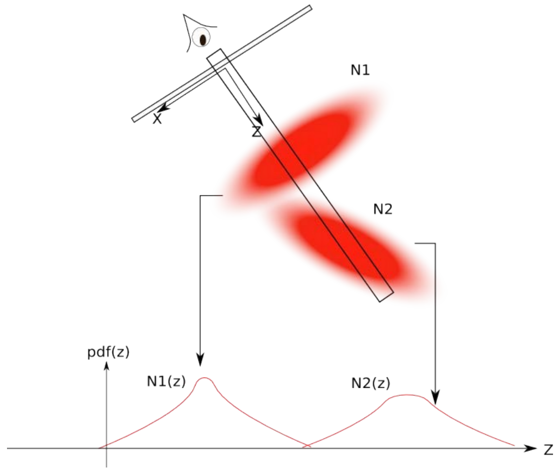

Figure 1: *Illustration of our method. The influence of the trivariate normal distributions over the final occlusion is evaluated for a view ray. First, the distributions are expressed in the ray's local frame, which allows us to build univariate gaussians by conditioning to 0 two dimensions (here, x and y). We then keep the maximum values of these distributions, typically the values of the functions at their respective mean positions. This process is applied to the set of distributions held in the nodes intersected by the view ray in the KD-tree.*

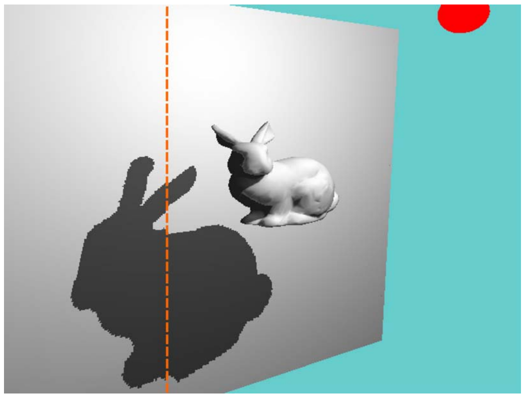

Figure 2: *Comparison of generated shadows on a "virtual" plane. The left part of the shadow was generated using our method and the right part using the mesh reconstruction algorithm from [MRB09]. The red sphere represents the light source position.*

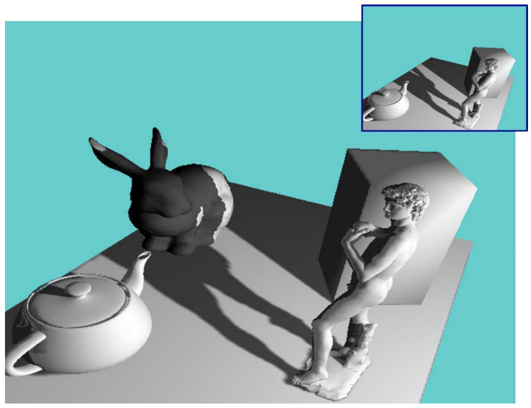

Figure 3: *Shadows in a virtual object integration context. The Stanford bunny is our "virtual object", and is inserted into the environment. The shadow produced by the David object on the bunny is evaluated using our method.*

4. Results

We evaluated our algorithm by range sampling the Stanford bunny (a "real" object) and computing the shadow cast on a plane which would have been integrated in a real scene. The scene is illuminated by a uniform and punctual source. The camera and light may move in time. The point cloud is incrementally refined with 713 range images extracted from successive viewpoints. Computation times are respectively 110 ms for KD Tree update, 40 ms for the multivariate normal distribution parameters update, and shadow map rendering times depend on the number of distributions to evaluate (about 500 ms for a 5000-distributions set). We compare our approach to a meshing strategy [MRB09] in Figure 2. Results on a more complex scene are provided in Figure 3.

5. Future Work

We proposed a statistical approach to visibility determination, built from a point cloud. The shadows obtained with our method are convincing but generally over-estimate the real occlusion. Results similar to [Paj03] may be achieved by rasterizing isosurface ellipsoids of each distribution. We plan to try this in future work. Also, occlusion is only evaluated using the leaf distributions. It would be interesting to use the coarser approximations from parent nodes of the KD tree. We believe that such optimizations would make our method suitable for interactive environment scanning in real-time augmented reality applications.

References

[KTB07] Katz S., Tal A., Basri R.: Direct Visibility of Point Sets. *ACM Transactions on Graphics (TOG) 26*, 3 (2007). 1

[MRB09] Marton Z. C., Rusu R. B., Beetz M.: On Fast Surface Reconstruction Methods for Large and Noisy Datasets. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)* (Kobe, Japan, 2009). 2

[Paj03] Pajarola R.: Efficient Level-of-Details for Point Based Rendering. In *Proceedings IASTED International Conference on Computer Graphics and Imaging (CGIM)* (2003). 1, 2