Refinement Criteria for High Fidelity Interactive Walkthroughs*

António Oliveira†
Departamento de Informática
Universidade do Minho

Luís Paulo Santos‡
Departamento de Informática
Universidade do Minho

Alberto Proença§
Departamento de Informática
Universidade do Minho

Abstract

Physically based global illumination rendering at interactive frame rates would enable users to navigate within complex virtual environments, such as archaeological models. These algorithms, however, are computationally too demanding to allow interactive navigation on current PCs. A technique based on image subsampling and spatiotemporal coherence among successive frames is exploited, while resorting to progressive refinement whenever there is available computing power. A physically based ray tracer (Radiance) is used to compute reflected radiance at the model's triangles vertices. Progressive refinement is achieved increasing the sampling frequency by subdividing certain triangles and requesting shading information for the resulting vertices. This paper proposes and evaluates different criteria for selecting which triangles to subdivide. A random criterium and two criteria based on Normalized Luminance Differences are evaluated: one operating on image space, the other on object space. Results, obtained with a model of an old roman town, show that the object space criterium is able to locate and represent visual discontinuities, such as shadows, and does so requiring less triangle subdivisions than the other two.

CR Categories: I.3.7 [Computer Graphics]: Ray Tracing—; I.3.2 [Computer Graphics]: Distributed Network Graphics—; J.2 [Computer Applications]: Archaeology—;

Keywords: interactive walkthrough, asynchronous rendering, coherence, ray tracing, refinement criteria

1 Introduction

Physically based global illumination rendering at interactive frame rates is a major challenge for the computer graphics research community but it is also a major requirement to allow walkthroughs within virtual environments, since it would enable users to navigate within such models while visualizing high quality images that can be used on a predictive manner. This capability is essential for many applications, such as architectural design, lighting engineering and virtual reconstruction of long disappeared archaeological sites.

The latter is the main motivation for the current work. Archeologists working with a model of the ancient roman town of Bracara Augusta often complain about the odd feeling they experience in these interactive tours: lack of realism due to local illumination models that do not take into account light interactions among objects. To take advantage of walkthrough facilities, archaeologists require correct illumination to study architectural building options and the urban development of the town. This includes to observe how the behaviour of the sun influences the size and shape of windows, the allocation of the inner space and streets' orientation; to understand the quantity of daylight in every house compartment, enhancing the perception of realism in the inner space of the building; to infer if a room needs artificial lighting (oil lamps) and simulate these artificial lighting conditions.

To provide accurate visualizations of the underlying virtual world, physically based global illumination models must be used to render the scenes. Global illumination algorithms, however, are computationally too demanding to allow interactive navigation on large virtual models using current PCs. Brute force rendering of each frame on a virtual walkthrough is impractical and ingenious techniques are required to support a smooth navigation while presenting tolerable degradation of the image quality. To support smooth virtual walkthroughs a technique based on parallel processing, asynchronous rendering and spatiotemporal coherence among successive frames is exploited, while resorting to progressive refinement to converge to high fidelity images whenever there is available computing power. It is structured as a three-tier architecture: a parallel high quality renderer (the physically based ray tracer Radiance [Ward and Shakespeare 1998; Ward 1994]), a Shading Management Agent (SMA), which caches previously computed shading values, and a visualization client. Spatial coherence is exploited by accurately shading only at visible triangles vertices, thus spasely sampling the image plane, and interpolating among them when displaying the image [Santos et al. 2005; Tole et al. 2002; Walter et al. 1999; Walter et al. 2002]; interpolation is performed by the visualization client, using the graphics hardware. Temporal coherence is exploited by reusing previously shaded points and reprojecting them according to the current view point (thus currently restricting the illumination models to Lambertian ones). By limiting the shading points (or samples) to the original geometry triangles' vertices, the resulting images are inaccurate wherever the shading frequency should be higher that the actual sampling frequency, such as at shadows boundaries. To overcome this limitation a hierarchical subdivision mesh is associated with each original triangle on the model; a new level is added to the hierarchy whenever a triangle is selected for subdivision - resulting on several children triangles - thus increasing the sampling frequency over that region. Image quality will progressively converge to a fully ray traced solution as the projected area of each subdivision triangle approaches one pixel. The policy used to select which triangles to subdivide at a given instant is referred to as the refinement criterium. This paper's main contribution is to propose and evaluate three different refinement criteria: a random one and two criteria based on Normalized Luminance Differences, one working on image-space and the other on object-space. These will be evaluated with respect to convergence rate, resources' consumption and empirical image quality.

Bracara Augusta, currently Braga, Portugal, was one of the three major towns founded by Emperor Augustus in the Iberian Peninsula, at the end of the Cantabrian wars (around 16 BC). The archae-

*This work was partially supported by portuguese FCT research grant POSI/CHS/42041/2001

†e-mail: antaroli@gmail.com
‡e-mail:psantos@di.uminho.pt
§e-mail:aproenca@di.uminho.pt

ological rescue project of Bracara Augusta started in 1976: some relevant buildings and infrastructures, regarding urban development and architecture, have been found, studied and interpreted. Based on these archaeological data and on their interpretation, a first challenge to virtually reconstruct Bracara Augusta resulted on a virtual model, which shows the urban development of the city with some accurate reconstructions of the Alto da Cividade *thermae* (the only public health-resort totally excavated in Braga), the *insulae* of the Carvalheiras (a wealthy private house), the defensive wall and the orthogonal streets. The virtual model of Bracara Augusta was created in 2000, using some available commercial tools. It has a complexity of about 1,140,000 faces. One of the main goals of Bracara Augusta's virtual model was to make an educational movie to promote archaeological research. A 20 minutes long movie was initially produced in 2000, each frame taking about 2.5 hours to render on a SGI Octane, single processor [Martins and Bernardes 2000].

The next section contains an overview of the three-tier architecture and the framework used to support smooth high fidelity virtual walkthroughs. The following section presents related work. Section 4 discusses the refinement criteria proposed on this paper. To evaluate system performance and the quality of the rendered images, section 5 describes the experimental methodology and setup, while section 6 critically analyzes the obtained results. The conclusion section closes the paper.

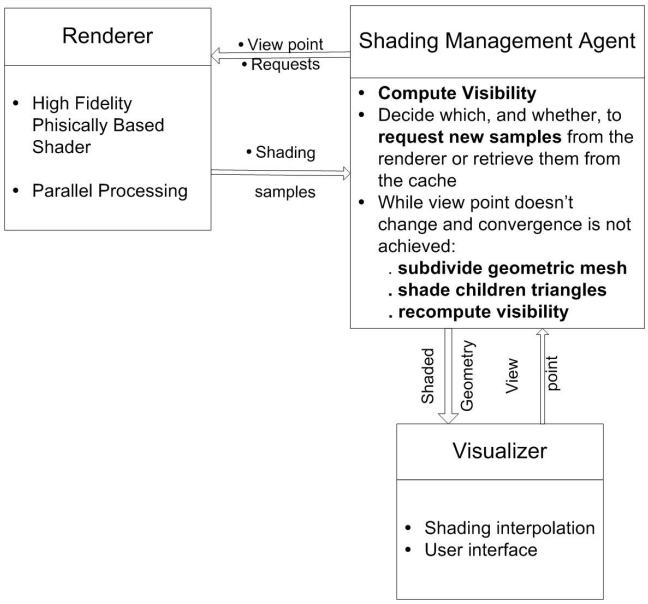


Figure 1: System structure.

2 System Overview

To support high frame rates both spatial and temporal coherence are exploited. This approach relies on image space subsampling and object space caching of shading values; progressive refinement is proposed as the mean to converge to high quality images, whenever computing power surplus is available. Two basic functionalities of a navigation system are identified and decoupled: image visualization and rendering run asynchronously and at their own pace. An additional process is identified and inserted between the renderer and the visualizer: a "Shading Management Agent" (SMA) computes visibility and decides which, and whether, shading samples are requested to the renderer or retrieved from a local cache of previously evaluated shading information (see figure 1).

Spatial coherence is exploited by subsampling the image space, requesting accurate shading samples only at visible triangles vertices and interpolating among samples when displaying the image [Tole et al. 2002; Walter et al. 1999; Walter et al. 2002]; interpolation is performed by the visualization client, using the graphics hardware. The SMA determines which triangles are visible for a given view point and either requests vertices' shading values from the renderer or retrieves these values from the local cache, if available. By supplying the visualizer with only the visible portion of the geometry, the traffic volume between the SMA and the visualizer is reduced and the workload imposed upon the visualizer hardware is kept to a minimum. Furthermore, since the visualizer is supplied with the geometry, this is always geometrically accurately reproduced, even if the view point changes.

Temporal coherence is exploited by caching previously computed shading samples and reusing them whenever the geometry becomes visible again. Caching occurs on object space to avoid reprojection artifacts. The SMA is responsible for maintaining this local cache. Illumination artifacts occur when the view point changes due to specular phenomenons. The cached samples must therefore be periodically recomputed (refreshed); this capability is not implemented on the prototype version under evaluation. Temporal coherence is also exploited to reduce the communication traffic between the SMA and the visualizer by resorting to an incremental communication protocol. On an ordinary walkthrough most triangles visible on a given frame will also be visible on the consecutive frame; instead of sending all the geometry for each frame, the SMA indicates to the visualizer which new triangles must be added to the visible set and which ones are no longer visible. This results on three sets of data to be sent: the list of triangles to add that were found on the SMA cache, the list of triangles that are no longer visible and can be deleted from the visualizer data structure to reduce memory requirements and finally the list of newly visible triangles whose shading information had to be requested to the renderer. This ordering results in the fastest updates on the image displayed by the visualizer.

Note that when the view point changes smoothly, as in ordinary navigations, the set of visible triangles also changes smoothly from frame to frame. Thus very fast feedback is given to the user by reusing triangles that are already stored on the visualizer or sending triangles from the SMA cache that are already shaded (because they have been used on past frames). Only triangles that have never been shaded must be requested to the renderer. The user can thus navigate on lower quality images, visualizing only the subset of shaded triangles, and converge to higher quality by temporarily stopping at a given view point. The second time a portion of the geometry becomes visible feedback will be much faster since all shaded information is stored on the cache.

Image reconstruction from a sparse set of samples is inaccurate on those regions where high spatial frequencies occur, such as at shadows boundaries, specular materials (e.g., glass, mirrors) and on texture mapped polygons. This can be overcome by progressively requesting more samples to the renderer. The geometry is represented on the SMA using a hierarchical subdivision mesh, which allows original triangles to be arbitrarily subdivided. Whenever there is available computing power surplus (e.g., when the user stops on a given view point for some time), the SMA selects some triangles for subdivision according to a refinement criterium and requests shading values from the renderer. These subdivisions and respective shading information are stored on the hierarchical mesh and sent to the visualizer. The image will thus converge to a high qual-

ity image, given enough computing time; no interpolation will take place when each fine grain triangle maps onto a single pixel. The assessment of refinement criteria is this paper's main contribution.

The first time a view point is reached the SMA computes visibility for the initial geometry and requests shading information for all visible triangles' vertices; shaded triangles are then sent to the visualizer. Once all initial triangles are shaded the refinement process steps in: triangles are selected for subdivision and shading information is requested to the renderer. This is an iterative process, triangles being subdivided multiple times until the termination criterium is satisfied. Whenever a triangle's children are shaded, information is sent to the visualizer to display the new triangles and delete the original one from its local data structure. The SMA recomputes triangle visibility when it receives a chunk of recently shaded children triangle to make sure that only useful information is sent to the visualizer and that non-visible children triangles are not further subdivided; this is required because some children of a visible triangle may not be visible themselves.

The renderer, the SMA and the visualizer can be mapped onto different sets of machines, favouring remote visualization and parallel computing.

A point based renderer, such as path tracing or ray tracing, must be used to accept requests arbitrarily distributed across the image plane or the object space. A modified version of Radiance [Ward 1994; Ward and Shakespeare 1998] is used, which accepts input over POSIX sockets and launches several processes across a cluster of workstations to increase rendering throughput. Radiance is a high quality physically based ray tracer extended with the irradiance cache to account for diffuse interreflections [Ward and Heckbert 1992].

The SMA is a multithreaded program that connects to the renderer and the visualizer. The current evaluation version supports a single visualization client, while future versions will be able to accept connections from multiple clients, which share the same geometric model and cached samples, thus reducing memory and workload requirements. Special care must be taken, however, with shared shading values resulting from different view points, since the specular components depend on these. The system is conceived as a "Rendering Service Provider", in the sense that visualization clients can connect to the SMA from remote locations and can run on light, affordable machines. The SMA and the rendering processes can run on powerful machines, or even on a cluster of workstations, hosted on the service provider facilities and eventually shared by multiple clients.

The visualization client is a light-weight program with modest memory, computing and communication requirements. Care has been taken to assure that only the visible geometry is stored on the client's memory, thus reducing memory demands. Ordinary graphics hardware is required to project and interpolate over the geometry. An incremental communication protocol is used to minimize bandwidth requirements. With current developments on mobile devices technology it is conceivable that in the near future the client can run on portable devices, such as PDAs or mobile phones.

3 Related Work

The Render Cache system [Walter et al. 1999; Walter et al. 2002] pioneered the integration of asynchronous rendering, exploitation of spatiotemporal coherence and parallel processing. The renderer is decoupled from the visualizer; the latter requests shading values for specific 3D points (shading samples) and builds an image by interpolating over whatever shading data there is at each moment. Thus the image plane is subsampled, exploiting spatial coherence. Shading samples are cached on image space and reprojected whenever the view point changes, thus exploiting temporal coherence. Progressive refinement is achieved by requesting further shading samples to the renderer, whenever there is computing power surplus. Reprojection of image space cached samples results in geometric errors. To avoid this problem the Shading Cache [Tole et al. 2002] uses a similar approach but shading samples are cached as a cloud of points in object space, thus eliminating the need for image space reprojection and associated errors. The system used for the current work is based on the Shading Cache architecture.

The refinement criterium guides the process convergence by selecting which geometric primitives should be further subdivided. More shading samples will be requested, increasing the sampling frequency on those locations. The Render Cache builds a refinement priority map based on the age of the sample that maps onto a pixel - age being defined as the number of frames elapsed since that sample was computed. For interpolated pixels - those where no sample maps directly - priority is computed as the inverse of number of neighbouring pixels that have samples mapped to them. Higher priority is given to older samples and to image space regions that have less shading samples. An error diffusion dithering algorithm is then applied to the priority map to obtain a good spatial distribution. If the view point is changing from frame to frame, then further samples are requested by predicting several frames ahead where regions without data are likely to become visible. This is an image space based refinement criterium that doesn't use any information from the shading values themselves and that can be a good indicator of whether or not there is a feature of interest (such as a shadow) on an image region. The Shading Cache uses a mixed object space and image space refinement criterium. Each pixel on the image is assigned a priority based on its predicted interpolation error value. This is computed as the luminance difference between the maximum and minimum luminances of the vertices of the triangle that maps onto each pixel. The priority map is then normalized, a pixel is randomly selected and an additional random number is generated. If the pixel normalized priority is larger than this random number, then the triangle that projects onto that pixel is selected for subdivision. The probability of a triangle being selected for refinement is proportional both to its predicted interpolation error and to the number of pixels it projects to. Triangles projecting on small areas are very unlikely to be selected for subdivision by this random process. However, since the refinement scheme is trying to direct samples into regions of high gradients, such regions inevitably contain a large number of small triangles. In order to improve the rate of convergence in these cases, a second image plane selection phase is applied that tests the neighboring locations of the already selected high priority samples and accepts them if their priority is greater than the current sample. This can be seen as a neighbourhood propagation strategy. Kavita et al. [Bala et al. 2003; Velázquez-Armendáriz et al. 2006] propose a technique to improve both interpolation and refinement based on edge and point images. Edges, corresponding to geometric discontinuities or shadow boundaries, are located on the image plane at subpixel precision by applying image processing operators. This information is used to interpolate between samples using an edge-respecting invariant and to direct the refinement process to regions containing such discontinuities. This technique requires complex image processing operators.

The Holodeck system [Ward and Simmons 1999] caches data on ray space in order to enable interactive walkthroughs on models rendered using the Radiance system. Although it supports some kinds of view-dependent phenomena, such as glossy reflections, it does not support object motion. Approaches based on the Shad-

ing Cache support dynamic geometry by refreshing samples that become invalid [Tole et al. 2002].

4 Refinement Criteria

Most refinement criteria used on previous work operate on image space and require both an image preview at the SMA level based on currently available data and image processing operations that can be too demanding for interactive settings. Our main goal is to propose and evaluate new refinement criteria which are more intuitive and do not consume a significant percentage of resources.

The refinement criteria can work either on image space or object space. Image space criteria use information at the pixel level obtained by projecting the visible geometry onto the image plane. The selection of the refinement point is based on features such as segmented edges or image regions characteristics. Our image based approach selects such a point based on region characteristics, identifies which triangle maps onto that pixel and subdivides it. An orthogonal approach is to operate on object space and directly identify which triangles to subdivide based on some object space metric. Rays can then be traced through those triangles subdivision points to collect higher frequency shading information.

The refinement criteria must select regions where the available shading information is probably not enough to accurately characterize lighting distribution over those regions; the selected regions should be the ones that exhibit colour variations along their areas. It is highly probable that those regions exhibiting reflected radiance variations over their surfaces are the ones that also exhibit variations among some boundary points. Our approach is to, on image space, consider quadrangular regions and use their four corners as the regions defining points; similarly, on object space we use the triangles' vertices. To quantify variations among these points we propose a metric based on normalized luminance differences (NLD), which represents quite well the luminance variations among a set of n points. This metric is inspired on the standard deviation but instead of the samples' average uses differences among all pairs of samples. Luminance is used because it is available on our system due to the used tone mapping operator [Drago et al. 2003] together with the maximum luminance value present on an image, L_{max}, used to normalize our metric. The metric, denoted by S, is given by

$$S = \sqrt{\frac{\sum_{i=1}^{n-1}\sum_{i+1}^{n}(L(x_i)-L(x_j))^2}{(n^2 \text{ div } 4)L_{max}^2}} \quad (1)$$

We evaluate two criteria using the same metric, one operating on image space, the other on object space. The image space criterium, labelled as NLD_IS, divides the image plane into quadrants and for each quadrant computes S using its four corners as the defining points. The luminance value used for each corner usually will not correspond to an exact sample computed by the renderer; rather it will be a value interpolated by the SMA graphics hardware from the vertices belonging to the triangle that projects onto that pixel. The quadrant which presents the larger S value is selected for subdivision, meaning that the triangle that projects into its center is subdivided according to the subdivision criterium; in fact, a triangle subdivision only occurs if S is above a given threshold, S_{req}, and if the triangle's projected area on the image plane is larger or equal than 6 pixels. This process is continued hierarchically: each quadrant at level l is subdivided into 4 quadrants (level $l+1$) and the same procedure is recursively applied. The process terminates either when the quadrant dimension is equivalent to one pixel or when a given fraction of the total number of pixels has been requested to the renderer.

NLD_OS operates on object space. S is computed for each visible triangle and they are selected for subdivision if S is above a given threshold, S_{req}, and if the triangle's projected area on the image plane is larger or equal than 6 pixels. Once a triangle is subdivided and shading requests for the children's vertices are satisfied, these new triangles become also candidates for subdivision, going through the same classification process as the others. The process terminates when all S are below S_{req}.

To assess the results achieved with both these criteria, we compare them with a baseline technique, which uses no information at all. The random refinement criterium (RND) randomly selects points on the image plane, subdividing the triangles that project on those points. It terminates after a given number of selected points, specified as a fraction of the total number of pixels on the image.

4.1 Optimizations

The above described refinement criteria require some optimizations related either to system performance or to a more accurate selection of triangles. While evaluating the refinement criterium the SMA is focused on this task and locks the cache data structure, thus is unable to perform other tasks such as new view point reception (if the user interacts with the visualization client) and/or insertion of new shaded samples onto the cache data structure. This would result on reduced interactivity, which is not acceptable since the system's main goal is short response times to user actions. To overcome this problem the refinement process is interrupted after a given number of triangles, R, has been selected for subdivision and control is returned to the main loop. On the main loop the new shading requests are sent to the renderer and the occurrence of any potential events is checked. If view point has not changed then control is returned to the refinement process to select yet another set of R triangles. R impacts on system performance and must be carefully selected.

As the refinement process progresses new information becomes available, such as more dense shading information over the image plane and a finer triangle mesh. The refinement criterium processes only triangles tagged as visible and these are determined by projecting them onto the image plane using the graphics hardware. Maintaining this set of tagged triangles as close to the real set of visible triangles as possible is a major requirement to assure that both the refinement process workload and the volume of data sent to the visualizer are kept to a minimum. However, when a visible triangle is subdivided onto several children, some of these might not be visible. Thus, it is necessary to regularly reproject the new, finer, mesh in order to determine which children triangles are visible. Also, since NLD_IS uses the image plane projected luminance to decide whether or not to subdivide, this must be recomputed regularly in order to use the latest data for decision making. Visible triangles' luminance is thus reprojected onto the image plane after a given number of new shading samples has been received.

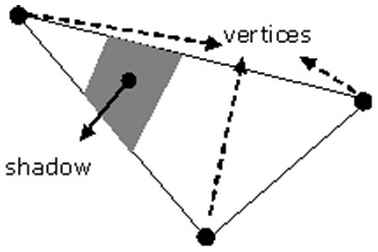
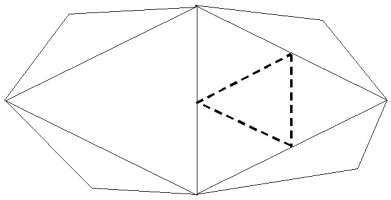


Figure 2: Shadow crossing a triangle without affecting its vertices shading values.

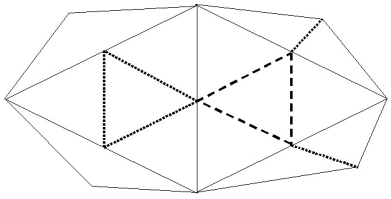
The main drawback of NLD_OS is that significant shading features, such has shadows, may appear on a triangle's surface and yet not affect its vertices, as shown on figure 2. On such cases, the normalized luminance differences metric, S, will report a minimal value and the triangle will not be selected for refinement, thus missing important information. To minimize this error two further optimizations are used. If a triangle projects onto a number of pixels larger than a given fraction of the image size, then it is always selected for refinement, independently of S. Large triangles are thus always subdivided; if a subdivision point falls onto the feature of interest, then S will correctly report it on the next refinement level and the children triangles will be normally selected for subdivision. The second optimization relates to the fact that a shadow crossing a triangle will probably also cross its neighbours. Therefore, when a triangle is subdivided its neighbours are also selected for subdivision, as described on section 4.2, assuring that whenever a feature of interest is found all the neighbourhood is processed in order not to miss it.

4.2 Subdivision Criterium

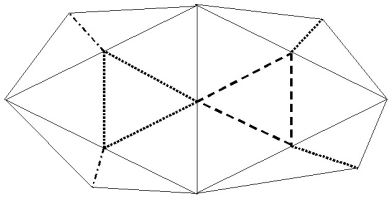
Each triangle selected for subdivision is divided onto 4 children triangles. These are defined by three subdivision points, corresponding to the middle point of each edge, as shown in figure 3(a). This subdivision approach is based on Loop's subdivision algorithm [Loop 1987]; it results on a well balanced refined mesh and will detect a feature of interest that crosses the middle of any edge.



(a) Selected triangle subdivision into 4 children.



(b) Subdivision propagation into neighbours. Those that have an height to width ratio less than a given threshold are divided into two children and do not propagate into their own neighbours.



(c) Propagation termination.

Figure 3: Selected triangle subdivision and propagation to neighbours.

As stated before subdivision is propagated to the neighbours. This helps avoiding T-vertices and contributes to find the real limits of a feature of interest. Since subdivision propagation is a recursive process, it would propagate through all connected triangles. To avoid this situation, if a neighbour's height to width ratio is less than a given value, then the triangle is subdivided only into two children and propagation terminates; else, the triangle is normally divided into four children and propagation continues. Figure 3 illustrates this process. It is implemented on our prototype and presented very good results, both on avoiding T-vertices and on finding and following features of interest, such as shadows, without resorting to too much useless subdivisions.

5 Experimental Setup

Since our goal is to assess refinement criteria, all experiments consist on fixed view point measurements, rather than navigation within the model. We will empirically assess the quality of the obtained images with respect to the quality of visual discontinuities found by the refinement process, specifically the quality of shadows. We also report the time, in seconds, required to obtain these images, T, as well as the total number of triangles being displayed, Δ_{img}, and the total number of triangles that had to be sent to the visualizer to reach that state, Δ_{total} – these include triangles that were subdivided and thus deleted at some time, but reflects the number of triangles that had to be rendered and communicated through the network. Results are presented for three view points, corresponding to images where shadows are important and have peculiar shapes: a view of the city wall (CW) where the number of triangles belonging to the view frustrum is large, a view of a street (ST) where columns and the buildings' roofs cast shadows on the floor and a view of a temple (TP) where a column casts a large shadow that is not captured without the refinement process.

Images and metrics are taken at two different stages of the progressive refinement process: stage 1 corresponds to the instant where all initial visible geometry has been shaded and triangle subdivision is just starting, stage 2 is the final image obtained when the refinement process terminates. Metrics corresponding to each stage are superscripted with the stage number; Δ^1_{total}, for example, corresponds to the total number of triangles sent to the visualizer up to stage 2.

The working environment uses two machines, interconnected through 100 Mbits/s Ethernet:

- the visualization client is a PC with an AMD Athlon CPU and a NVidia GeForce 5200 GPU;
- the SMA and the renderer run on the same machine, a dual-Xeon 3.2 GHz processor, with 2 GB RAM and a NVidia 6800GT GPU.

For results concerning interactive navigation and parallel rendering with Radiance using this framework (without triangle subdivision) refer to [Santos et al. 2005].

6 Results' Analysis

Figures 4 to 6 display the resulting images for the different view points mentioned before - CW, ST and TP - starting with the non refinement stage (S1), followed by RND, NLD_IS and NLD_OS. Table 1 complements the captions on these figures with quantitative data.

As expected in CW, where shadows are not perceptually relevant, the differences in the quality of shadowing among the three ap-

	View	CW		ST		TP	
	Stage	S1	S2	S1	S2	S1	S2
RND	T (secs)	15,9	34,1	3,5	15,5	4,8	18,5
	Δ	2511	47909	603	26426	1100	28111
	Δ_{total}	2511	77528	603	42896	1100	45354
NLD_IS	T (secs)	15,9	104,2	3,5	50,2	4,8	8,5
	Δ	2511	45909	603	34989	1100	3378
	Δ_{total}	2511	69493	603	53804	1100	4383
NLD_OS	T (secs)	15,9	53,3	3,5	20,5	4,8	22,5
	Δ	2511	19474	603	7701	1100	8007
	Δ_{total}	2511	29657	603	11923	1100	15072

Table 1: Results for different refinement criterium.

proaches is not significant. However, the results achieved with NLD_OS show better defined shadow boundaries, less than half processed triangles (Δ_{total}) and is twice as fast as NLD_IS.

The street scene has a richer and more complex set of shadows, such as the ones cast by the columns and the boundaries of the ones cast by the roof on the middle of the street. Only NLD_OS shows non-fuzzy shadows' boundaries; its overall quality is close to Radiance results, requiring only 40% rendering time.

Impressive enhancements are achieved with the TP scene, where NLD_OS shows a clearly defined shadow when compared to NLD_IS and requiring less triangles than the random approach. As before, the overall quality is close to Radiance results with similar gains in execution times.

Globally the images refined with NLD_OS show clearly defined shadow contours, require less triangles to be processed and with acceptable execution times (significantly faster than Radiance). The better quality of shadows obtained with NLD_OS with a smaller set of triangles demonstrate that this refinement criterium is superior than the other two approaches. Intermediate images, visible during the refinement process, show that, once a potential discontinuity is identified, this approach starts refining triangles on its neighbourhood and is able to match the refined triangle mesh with the discontinuity boundary.

All the reported timings refer to the first visit to each view point, thus requiring all triangles to be shaded by the renderer. On a second visit, these triangles are stored on the SMA cache, therefore, the response time is much faster. Results obtained for these view points and for the refined geometry show that each image is completely displayed on the visualizer on a few hundred milliseconds.

7 Conclusion

Three different refinement criteria, used to select which triangles to subdivide within a subsampling rendering approach, were analyzed: a random criterium (RND), an image space criterium (NLD_IS) and an object space one (NLD_OS). The last two are based on Normalized Luminance Differences (NLD). Results were obtained using a model of the old roman town of Bracara Augusta.

The analysis of these results showed that the object space criterium clearly identifies visual discontinuities, such as shadows, and adapts the triangle mesh to their boundaries. The other two approaches achieve only fuzzy visual results, while requiring more triangle subdivisions. NLD_OS achieves better results faster than NLD_IS, but is slower than random. This added execution time is, nevertheless, compensated by the increased quality of the resulting image; optimizing NLD_OS is probably worth to reduce its overheads on the selection of triangles.

As future work we intend to merge NLD_OS with a random pass. This may help finding discontinuities that are not found by exploring only regions with luminance differences. Tone mapping must still be improved to display perceptually more pleasant images. Porting the visualizer to a mobile device and exploring wireless communication protocols is a major goal. The main R&D issues to tackle are support for specular phenomena, textured polygons and moving objects.

References

BALA, K., WALTER, B., AND GREENBERG, D. 2003. Combining Edges and Points for Interactive High-Quality Rendering. *ACM Transactions on Graphics - Proc. of SIGGRAPH 2003 22*, 3, 631–640.

DRAGO, F., MYZKOWSKI, K., ANNEN, T., AND CHIBA, N. 2003. Adaptive Logarithmic Mapping For Displaying High Contrast Scenes. *Computer Graphics Forum - Proceedings of Eurographics'03 22*, 3.

LOOP, C. 1987. *Smooth subdivision surfaces based on triangles*. Master's thesis, University of Utah.

MARTINS, M., AND BERNARDES, P. 2000. A Multi-disciplinary Approach for Research and Presentation of Bracara Augusta's Archaeological Heritage. *Archeologia e Calcolatori 11*, 347–357.

SANTOS, L. P., COELHO, V., BERNARDES, P., AND PROENÇA, A. 2005. High Fidelity Walkthroughs in Archaeology Sites. In *6th International Symposium on Virtual Reality, Archaeology and Cultural Heritage - Short Papers*, Eurographics Association, Pisa, Italy, R. S. M. Mudge, N. Ryan, Ed.

TOLE, P., PELLACINI, F., WALTER, B., AND GREENBERG, D. 2002. Interactive Global Illumination in Dynamic Scenes. *ACM Transactions on Graphics 21*, 3, 537–546.

VELÁZQUEZ-ARMENDÁRIZ, E., LEE, E., WALTER, B., AND BALA, K. 2006. Implementing the Render Cache and the Edge-and-Point Image on Graphics Hardware. In *Graphics Interface*.

WALTER, B., DRETTAKIS, G., AND PARKER, P. 1999. Interactive Rendering Using the Render Cache. In *Proceeding of the Eurographics Workshop on Rendering Techniques*, 235–246.

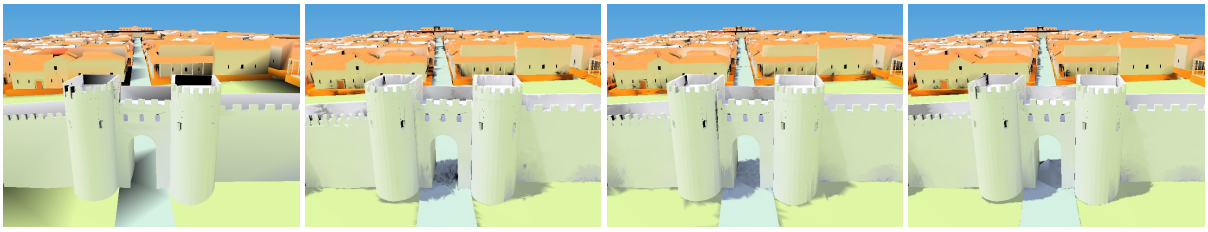
WALTER, B., DRETTAKIS, G., AND GREENBERG, D. 2002. Enhancing and Optimizing the Render Cache. In *Proceeding of the Eurographics Workshop on Rendering Techniques*, 37–42.

WARD, G., AND HECKBERT, P. 1992. Irradiance Gradients. In *3rd Annual Eurographics Workshop on Rendering*.

WARD, G., AND SHAKESPEARE, R. 1998. *Rendering with Radiance: the art and science of lighting visualization*. Morgan Kaufmann.

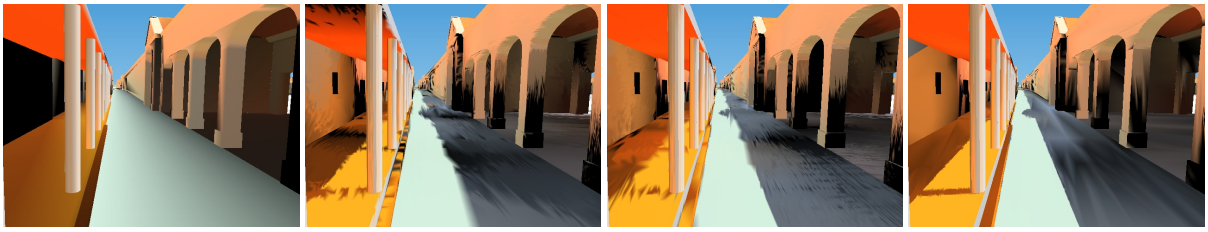
WARD, G., AND SIMMONS, M. 1999. The Holodeck Ray Cache: an Interactive Rendering System for Global Illumination in Non-diffuse Environments. *ACM Transactions on Graphics 18*, 4 (October), 361–398.

WARD, G. 1994. The RADIANCE Lighting Simulation and Rendering System. In *SIGGRAPH'94 - 21st International Conference on Computer Graphics and Interactive Techniques*, ACM Press, 459–472.



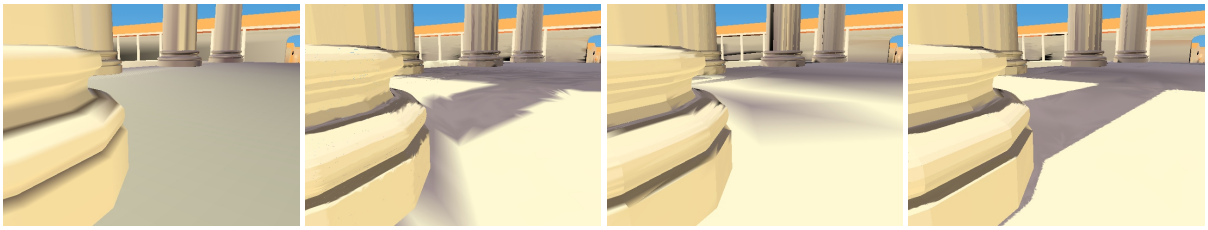
(a) No refinement - 2511 triangles; 15,9 secs (b) RND - 47909 triangles; 31,1 secs (c) NLD_IS - 45909 triangles; 104,2 secs (d) NLD_OS - 19474 triangles; 53,2 secs

Figure 4: Bracara Augusta city wall (CW) - comparison of different refinement criteria



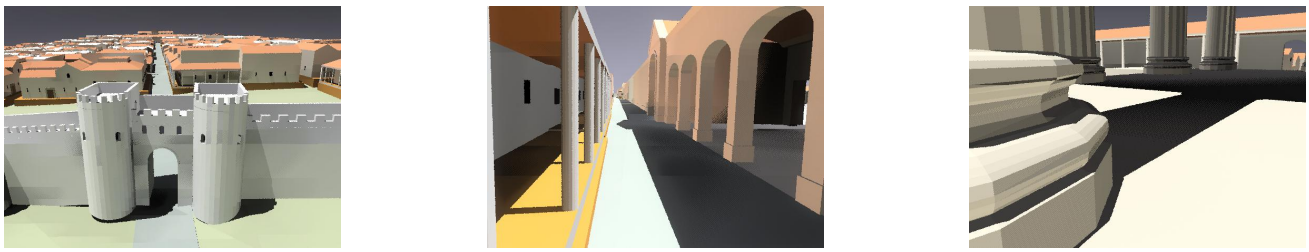
(a) No refinement - 603 triangles; 3,5 secs (b) RND - 26426 triangles; 15,5 secs (c) NLD_IS - 34989 triangles; 50,2 secs (d) NLD_OS - 7701 triangles; 20,5 secs

Figure 5: Bracara Augusta street (ST) - comparison of different refinement criteria



(a) No refinement - 1100 triangles; 4,8 secs (b) RND - 28111 triangles; 18,5 secs (c) NLD_IS - 3378 triangles; 8,5 secs (d) NLD_OS - 8007 triangles; 22,5 secs

Figure 6: Bracara Augusta temple (TP) - comparison of different refinement criteria



(a) City wall - 66 seconds (b) Street - 49,4 seconds (c) Temple - 50,3 seconds

Figure 7: Radiance results