EUROGRAPHICS 2004 / M.-P. Cani and M. Slater
(Guest Editors)

Volume 23 (2004), Number 3

A Progressive Rendering Algorithm Using an Adaptive Perceptually Based Image Metric

Jean Philippe Farrugia and Bernard Péroche
LIRIS, CNRS, INSA de lyon,Université Claude Bernard Lyon 1
Université Lyon2, Ecole centrale de Lyon
8, boulevard Niels Bohr,
69622, Villeurbanne, France
email:
jean-philippe.farrugia@liris.cnrs.fr
bernard.peroche@liris.cnrs.fr

Abstract

In this paper, we propose to solve the global illumination problem through a progressive rendering method relying on an adaptive sampling of the image space. The refinement of this sample scheme is driven by an image metric based on a powerful vision model. A Delaunay triangulation of the sampled points is followed by a classification of these triangles into three classes. By interpolating each triangle according to the class it belongs to, we can obtain a high quality image by computing only a fraction of all the pixels and thus saving computation time.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Perceptual Rendering, Global illumination

1. Introduction

Photorealism is a major goal to reach in computer graphics: the accurate simulation of complex environments needs to compute complete light transport between all parts of a virtual scene. This is possible by solving the rendering equation described by Kajiya in [Kaj86], and many solutions have already been proposed. Among all known methods, we may quote hierarchical radiosity [HSA91], simple or bidirectional path tracing [LW93], radiance interpolation [War94b], photon maps [Jen96], light vectors [ZSP98], Metropolis light transport [VG97], density estimation [WHSG97], ... All these methods give realistic pictures, but despite the increase in computer performance, computation times remain really long. Thus, a challenging problem is to dramatically reduce the computation time in order to be able to solve the global illumination problem at interactive rate. One way to evolve in this direction is to use a progressive rendering method: by carefully choosing a sample set, an approximate picture is generated. If the desired quality of this picture is not reached, new samples may be selected to produce a new image and the process can be repeated until reaching the desired quality. The main problem with commonly used progressive methods is to decide where to cast these new samples. This usually relies on empirical criteria based more or less on statistics. We propose to replace these criteria with perceptual ones as they are more general and natural than statistics based ones. Usually, statistical criteria take into account only one important picture characteristics at once: some of them care about object discontinuities, some others test local contrast, frequencies, color dispersion,... With a perceptual criterion, all these characteristics might be taken into account with only one process.

Thus, in this paper, we suggest a progressive perceptually-based rendering method. The refining criterion is given by a perceptual adaptive metric based on a vision model. To avoid unnecessary computations, this metric is computed for the pixels who have been already ray-traced by the rendering process. The final goal is to refine the approximate picture in such a way that the human eye would not see any difference with the reference picture computed with a classical global illumination method.

This article will be divided into six sections including this

Figure 1: *A dark scene*

introduction. We will review previous work on perceptual rendering in Section 2, and we will introduce our adaptive metric in Section 3. Our perceptual rendering algorithm will be presented in Section 4 and some results will be shown in Section 5. Finally, Section 6 will open some perspectives for the future.

2. Previous Work

One might wonder what visual perception knowledge may provide in rendering. Figure 1 gives prominence of taking perceptual criteria into account. This scene is very dark, we clearly see the sword in the center, but the columns on the sides are barely visible. Now suppose that the geometric model for these columns is very complex and detailed: since these details cannot be seen, it would be useless to compute them accurately like a Monte-Carlo method would. Using a vision model, it could be guessed that the columns do not require so much computation. More information about the usefulness of knowledge on human vision in computer graphics may be found in [Fer01]. In this section, we will review some vision models and then some rendering methods using them.

2.1. Vision models

There is a great variety of vision models and image metrics designed in the image processing field (cf. [Ahu93], [FZvdBLS97]), but very few of them were reused in computer graphics. In this section, we will only deal with the most commonly used in image synthesis.

Daly's Visual Differences Predictor [Dal93] was one of the first models tested with synthetic images. It is a three steps model, each step simulating specific vision characteristics through a complex mathematical function. The first step simulates light adaptation, the second one takes care of contrast sensitivity and the last one merges all detection mechanisms, including multi-scale decomposition, contrast masking and a psychometric function. This model is efficient but complex. Besides, it requires a quite important calibration phase to be effective.

In [RWP*95], the authors used a metric inspired by existing compression algorithms to compare radiosity pictures. This was the first attempt to apply perceptual knowledge in computer graphics in order to obtain better pictures and faster computations.

The algorithm in [Lub95] is an attempt to simulate the visual pathway from the retina to the mammalian cortex. Lubin's model features much more steps and requires more memory space than Daly's, but is faster. It is basically composed of a Laplacian pyramidal picture decomposition, local contrast computations and a psychometric function.

The authors of [PFFG98] improved Lubin's model to take color perception into account. They used their model in a tone mapping operator, but its design is generic so it could be included in an image metric as well. The global structure is approximately the same as in Lubin's model, but is extended to four channels: one for the rods, and three for the short, medium and long cones receptors. This model provides impressive results due to a complete simulation of retina and cortex treatments in the human visual system. It correctly handles adaptation, color appearence and masking, but it requires a lot of computational resources (memory and cpu time). We chose this model for our adaptive metric.

2.2. Vision-Based Rendering Algorithms

Mitchell [Mit87] was the first to use the visual perception of noise to optimize antialiasing for ray tracing. His main idea was to avoid unnecessary computations in the areas where noise is invisible to the human eye. To reach this goal, a simplifed vision model consisting of a mathematical formulation of local contrast perception was designed.

Meyer and Bolin [BM95] used a rather identical approach, but they decided to cast rays directly in the frequency domain, in order to only render frequencies that are significant to the human eye. In [BM98], they changed their approach and had a complete vision model, including color effects and masking, to monitor their Monte Carlo method. They showed that the computational cost of the model was low, regarding to the saving in rendering time.

Ramasubramanian *et al.* [RPG99] developed a threshold model which gives, for each pixel of the picture, the maximum luminance that can be added or subtracted at this point without noticing any difference. Therefore they were able to refine the picture in problematic areas.

In the hierarchical radiosity field, where an oracle is required to guide the subdivision of patches, vision models and metrics seem particularly adequate. A very interesting overview of perceptually driven radiosity methods is given in [PP99a]. Gibson and Hubbold [GH97] used a vision-based tone mapping operator and a Luv color space metric to subdivide or gather patches. Myszkowski also did a vast amount of research on perceptually-driven radiosity and ray tracing [MRT99], [MTAS01]. For example, he studied in [Mys98] the efficiency of Daly's VDP model [Dal93] to detect masked artefacts in radiosity scenes. Purgathofer and Prikryl [PP99b] did a perceptually-driven termination for stochastic radiosity.

Recently, Cater *et al.* [CCW03] presented a progressive rendering method taking advantage of the fact that visual attention can not focus on all the details in the scene at once. Their algorithm depends on the task assigned to the observer, for example "count the books on the shelve" or "memorize the drawing" and exploits it to refine the scene in areas the attention is mainly focused. A general framework for perceptually based rendering is extracted from this.

All these methods have their pros and cons. Some of them can hardly deal with specular objects (radiosity methods) or color scenes ([RPG99]). But the main problem is that they have a sophisticated vision model coupled with a slow rendering method (such as a Monte Carlo one). It would probably be interesting to substitute to this rendering method a progressive one which could render a faster sequence of approximate pictures.

3. Our Adaptive Metric

As already said, our purpose is to replace empirical criteria in progressive rendering with a perceptual one. Therefore, to manage our progressive algorithm (to be detailed in Section 4), we had to pick up an adequate vision model for computer graphics. We chose the Multiscale Model of Adaptation and Spatial Vision by Pattanaik *et al.* [PFFG98] for its excellent performance on various types of scenes. This model was originally designed for tone mapping, but its output is usable for distance computing. We also had to reduce the computation time for this model as we intend to use it in a progressive rendering algorithm. To do so, we designed an adaptive computation method to evaluate the metric. In this section, we will first describe how to use Pattanaik *et al*'s model for an image metric; then we will depict our adaptive method and discuss some results.

3.1. Using the vision model for an image metric

All details concerning the vision model may be found in [PFFG98]. The output of this model is a n-dimensional map of visible contrasts, each level corresponding to a spatial frequency band. Let us suppose that we want to compare image 1 with image 2. The outputs of the model for these two pictures are two n-dimensional maps. To compare the two pictures, we decided to calculate a pseudo-euclidian distance between these two n-dimensional maps.

Suppose that we want to compute the distance for pixel p. Let us denote p_1 and p_2 the vectors corresponding to the contrast values at p for pictures 1 and 2 across all the levels of the n-dimensional maps. The distance value for p will be given by the following formula, which is a $L^{2.4}$ metric:

$$d(p) = ||p_2 - p_1||_{2.4}$$

3.2. An Adaptive Method for Computing Distances

Our method intends to reduce computing time by processing only a part of the picture's pixels. We use an enhancement of the method by Albin *et al*, described in [ARPT02].

The main drawback of complex vision models is their computation time. The psychometric functions often involve logarithmic or exponential mathematical operations, which are very expensive. Since every pixel of every decomposition stage of the two images is processed through these functions, it is quite impossible to obtain the result at an interactive rate.

Albin *et al* noticed in [ARPT02] that, during a poll, only a small representative number of individuals is used to obtain a global result. They also made some tests showing that, above a certain threshold, this number no longer depends on the size of the whole population. Thus, in a similar way, only a small percentage of the pixels could be used to compute the distance between two images. Their method is based on a quad-tree partition of the image plane. For each cell, a fixed number of samples is randomly shot (the number of samples depends on the size of the cell), the distance is computed for these samples and an homogeneity test is performed. According to the result of this test, the cell will be subdivided or not. The mean distance value of the samples is affected to the cell. The final result is a "block map" of the distance between the two pictures composed of uniform distance value zones of various sizes. This map contains approximately the same information than a "complete" one, but it can be obtained up to ten times faster.

To apply this method, we need the vision model to be usable on a single pixel. This is not possible most of the time, because the first step on most models is a spatial decomposition. In Pattanaik *et al*'s model, it is a pyramidal difference of gaussians. This treatment can not be applied on a single pixel in the image space. Thus, we decided to apply this first step on the entire image, then to compute the following steps on individual pixels.

The image plane is subdivided with a quad tree. For each cell, we will shoot a fixed number N of samples depending on the size of the cell:

- 2000 if the cell has more than 10000 pixels;
- 500 if the cell has between 1000 and 10000 pixels;
- a third of the cell' size if the cell has less than 1000 pixels.

The distance is computed for each sample and the mean value is affected to the cell. Then we have to decide if this cell should be subdivided or not. We use the homogeneity test described in [ARPT02] as a subdivision criteria. We compute the following expression called "the homegeneity criteria":

$$\frac{\#\{x \in [\bar{x} - \varepsilon; \bar{x} + \varepsilon]\}}{|X|}$$

where ε is the threshold distance from the mean value $\bar{x}$ inside the cell and $|X|$ is the number of pixels in the cell.

If this quantity is greater than a fixed homogeneity percentage, then the cell will be subdivided. The algorithm stops when no cell needs to be subdivided anymore.

3.3. Summary

Here is a step by step summary of our adaptive method:

- Spatially decompose the two images;
- Draw randomly an initial sample batch (this step will be modified when we will integrate the metric into the progressive rendering algorithm);
- For each cell :
  - Compute the distance on each sample belonging to the cell and the mean distance value of the cell;
  - Compute the homogeneity criterion;
  - If this criterion is above the homogeneity percentage, then subdivide the cell;

until no cell needs to be subdivided anymore.

3.4. Some Results

In this section, we will present a simple result obtained with our metric. We designed four pictures to specifically test contrast masking. We computed the distance maps for the test scene with the following parameters: the homogeneity percentage is 80%, ε is set to 1. These values are empirically chosen but they are based on various experiments made on different scenes.

The test scene is a checkerboard with four spheres lying on it. The pictures on the left side of Figures 2 and 3 are computed by casting 64 rays per pixel, the right ones by casting 8 rays per pixel.The noise of the shadow projected on the ground is more or less visible depending on the texture of the floor. We test this feature with two different textures, a rough one (Figure 2) and a smooth one (Figure 3). The results on Table 1 show that the mean distance value is higher for the smooth texture scene. This confirms our expectations as we clearly see that the noise is more visible on the smooth texture floor. The distance maps (Figure 4) are the graphic translation of this result. On these maps, the brighter the area is, the greater is the perceptual difference on this area. The map on the left side corresponds to the rough texture floor and the map on the right side is obtained from the smooth texture scene. Table 1 gives computation times. These times may be compared to the 30 seconds needed to compute the vision model for every pixel on a 512x512 picture.

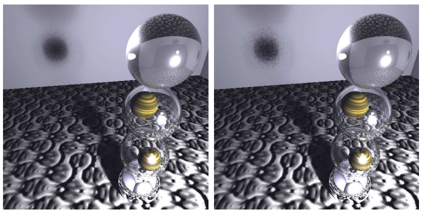

Figure 2: *Rough texture scene*

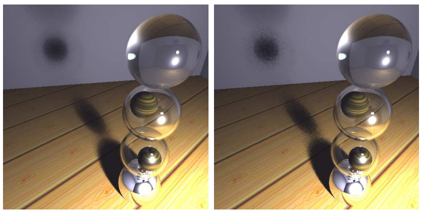

Figure 3: *Smooth texture scene*

4. Our Rendering Method

Our goal is to obtain an adaptive vision-based refining process coupled to a progressive ray tracing based algorithm. We chose to extend the directional coherence map from Guo [Guo98] by replacing its refining criteria with our metric. We will now explain how our extention to this method works.

In sections 4 and 5, we will call "reference picture" a scene computed with the light vectors method, described in [ZSP98] and "approximate picture" the same one computed with one or more steps of our rendering method. In this section, every pixel referred as "ray-traced" is in fact computed using the global illumination method from [ZSP98].

If we could ideally compare the reference picture with an approximate one, it would allow us to determine whether the picture requires to be refined or not. Since we don't know what the reference picture looks like, we chose to apply the metric on two successive frames rendered by the progressive algorithm and to use the distance map to assign a distance value to each triangle. The distance is initiated using the sample scheme from the rendering process. If the homogeneity criterion is not satisfied, the distance map is refined

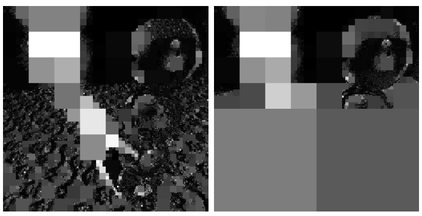

Figure 4: *Distance maps*

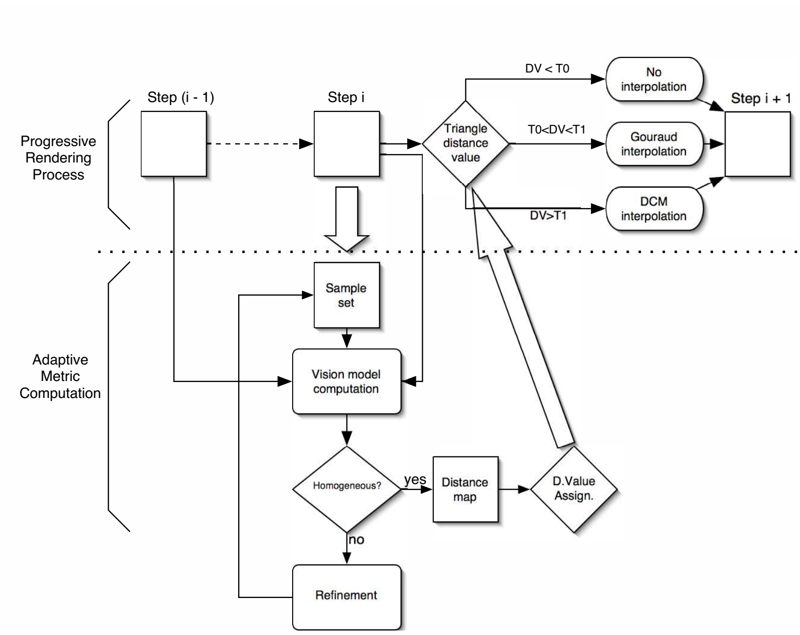

Figure 5: *Our method*

Scene	Time	Mean distance value
Rough texture scene	6.6s	3.5
Smooth texture scene	2.57s	5.32

Table 1: *Results*

as in Section 3.2. Figure 5 shows the whole rendering process. We will now detail each step in the following sections.

4.1. Scene Analysis and First Sample Set

Initial samples are very important as we could expect to converge faster toward the final approximate pictures if these samples are carefully chosen. We decided to sample the whole scene with a higher sample density around visible edges. This prevents missing small objects as their edges will be concerned by the sampling during the first step. Because of the CSG nature of our modeler, our algorithm has a precomputing phase: we must cast one ray at each pixel to obtain a object id map, then we derive a contour map from theses ids. To locate our samples around the contours, we process this contour map with a gaussian-like filter. It was specifically modified to place samples around the edges, but not on the edges. This is critical since our interpolation method seeks discontinuities in the image plane. By using this filter, we ensure that no object will be missed.

Then, our scene is sampled with the help of this map. For each pixel, a random number between 0 and 1 is drawn: if this number is less than the corresponding probability in the map, a ray is cast toward this pixel.When every pixel has been processed, we obtain a first sample set. The number of samples depends on the size of the picture and on the scene geometric complexity. It is around 1 percent of the whole picture's pixels for usual scenes.

4.2. Rendering and Refining Steps

To turn these samples into an approximate picture, a Delaunay triangulation is calculated. This is usually very quick since the number of samples is low. The first approximate picture is obtained by Gouraud shading this triangulation. But we need two pictures to apply our metric. The second approximate picture is generated by subdividing the Delaunay triangulation: for each triangle, a new sample is added in the proximity of its center and its value is computed by ray-tracing. By redoing a Gouraud interpolation, we obtain a second approximate picture.

We now use our adaptive metric to compute a distance map between these two pictures. To correctly initiate the distance process, we use the sample set given by the rendering algorithm and compute a distance block map as in Section 3.2. Please note that none of the new refining distance samples is ray-traced; the only use of these new samples is to obtain the distance map, making the rendering process and the distance process totally independent from each other. Finally, a distance value is affected to each triangle by taking the mean distance value of all the pixels inside the triangle.

From these values, we may classify our triangles into three categories inspired by [Guo98]. A "Final" triangle (with a distance value lower than a given threshold T_0) is supposed to be closely related to the corresponding triangle in the previous iteration; it is Gouraud shaded in the approximate picture and won't be subdivided during the next iteration. A "smooth" triangle (with a distance value between thresholds T_0 and T_1) should not have any "hard" discontinuity in its interior, but could contain soft shadows or caustics from indirect lighting; as for "final" triangles, it is Gouraud shaded in the approximate picture, but will be subdivided during the next iteration by placing one sample to be ray-traced near its center. Finally, a "complex" triangle (with a distance value greater than T_1) goes through a more sophisticated interpolation process. The first step is a contour evalution: all the pixels located on the triangle's contour are ray-traced. Such a triangle is supposed to have a complex content, so we are going to attempt to detect the least discrepancy direction. We try to locate two "rupture" pixels on the contour of the triangle. These pixels are defined as those with the greatest euclidian color distance with the previous pixel on the contour in trigonometric order. The segment joining these two pixels will be regarded as the least discrepancy direction. A linear interpolation is made along this direction to fill in the triangle. When the largest dimension of the triangle is less than five pixels, no interpolation is made and the whole triangle is ray-traced. A complex triangle is of course subdivided during the next iteration, unless all pixels have been computed.

After all triangles have been subdivided and interpolated, the new picture is compared to the previous frame via the same method as above, and the process goes back to the classification until all triangles have a distance value less than T_0. If this threshold is set to 0, the final approximate picture will be strictly identical to the reference picture as every pixel will be ray-traced.

5. Some Results

In this section, we will present some results of our rendering algorithm. We will compare a reference picture computed with the light vectors model [ZSP98] with our approximate pictures. Pictures' size is 512x512 pixels. For each test scene, we will present the reference picture and our approximate one. Please note that the vision model is calibrated for viewing pictures on a display with a 1280x854 pixels resolution, from a viewpoint located at least at 50 cm from the screen. The model has not been calibrated to view printed pictures. All these pictures have been computed on a 2.8Ghz Pentium computer with 512Mb of memory. The threshold T_0 has been empirically set to 1, which seems to be a correct value. From on our own experiences, a good estimation for the T_1 threshold can be obtained by measuring the mean distance value for all the pixels on the first iteration. This method is applied on the following examples.

5.1. Cellar scene

In this section we put our algorithm to the test with the scene shown on Figure 1. The reference picture, the approximate picture and the sample map are shown on figure 6. The computation times are 21 minutes for the ray-traced picture and 6 minutes for our method. 29 percent of the image's pixels have been computed.

We also tested our algorithm against Guo's method on Figure 7. The picture obtained with this method requires 7 minutes to compute and 35 percent of the picture's pixels are ray-traced. Unlike our method, Guo's algorithm computes all the details in the scene, wherever they are visible or not. Please note that Guo's algorithm has no stopping criteria, the user has to stop the rendering when he is satisfied with the result. We decided to stop it when the sampling rate becomes similar to the one obtained with our thresholds.

5.2. Cornell Box

These scene was chosen because it has both direct and indirect lighting. Figure 8 presents the reference image on the left, our approximate image on the right, and the sample map. This distance is computed with our metric. Computation times are 4 minutes 26 seconds for the reference image and 3 minutes 9 seconcs for our method. Sample rate is 33 percent.

5.3. Photo Laboratory

This scene represents a dark room in which light comes from the half-open door only. Everyday experience shows that this is enough to clearly see all the details in the room. The particularity of this scene is that it would be entirely black if global illumination was not taken into account; every lighting in the scene results from indirect lighting. For this test scene, we decided to emphasize the details that are not perceptible, so we can check that our metric is a good criteria. In Figure 9, the reference picture is compared with the approximate one. We can not perceive much difference between these two pictures. Now, let us examine Figure 10. The approximate picture is exactly the same as before, except that we have increased the gamma value by a factor of two. On this figure, we may see that many details that are in the dark are actually strongly interpolated. This was expected, as the model predicts that the scene's details located in the dark would not be visible for a human observer. The perceptually-based picture requires 20 minutes 23 seconds to compute, to compare with the 50 minutes required for the reference image. Sample rate is 26 percent.

6. Conclusion and Future Improvements

In this paper, we have proposed a new and efficient method to compute global illumination in a scene. This method

Figure 6: *Interpolated(left), reference(right) pictures for the cellar scene*

Figure 7: *our method (left) and Guo's one (right)*

mixes up a progressive approach based on a Delaunay triangulation of sample points with a perceptually-based algorithm using a metric defined by a vision model.

For an approximate picture which is almost indistinguishable from a picture computed with a global illumination algorithm, we only need between one third and one half of the computing time.

However, we are not yet able to produce pictures at an interactive rate. One of our goals is to optimize the computation time of our metric, in particular by much more tabulating the various functions used. We also plan to use conductance maps (see [Per03] for more information) for a better initial sampling scheme, taking shadows and reflection effects into account .

In the same way, we plan to enhance the vision model by taking the orientational selectivity of the human visual system into account and by adding depth of field and attention focus to our model.

References

[AB98] AHUMADA A., BEARD B. L.: A simple vision model for inhomogeneous image quality assessment. *SID Digest of Technical Papers 29* (1998).

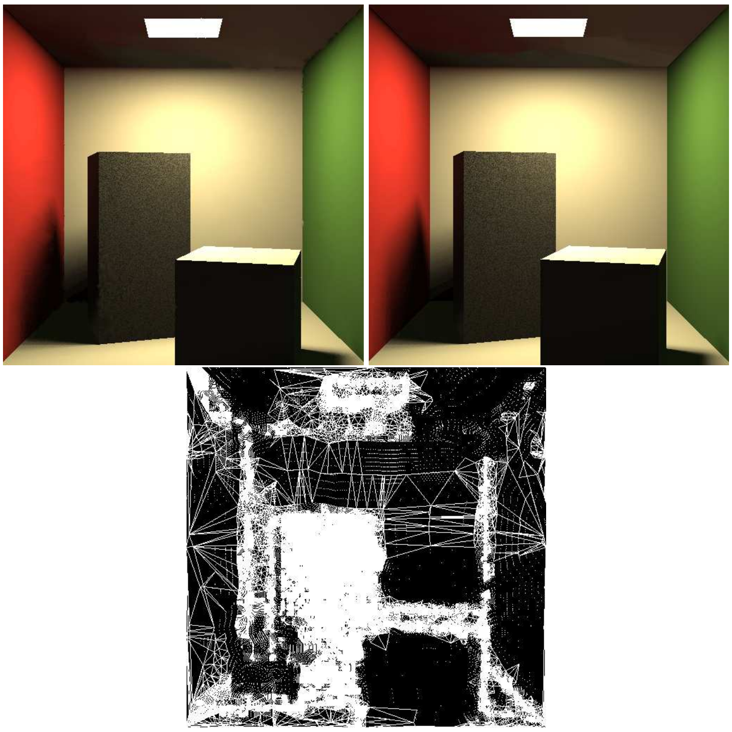

Figure 8: *Results for the Cornell Box scene*

[Ahu93] AHUMADA A. J.: Computational image quality metrics: A review. *Society for Information Display International Symposium Digest of Technical Papers 24* (1993), 305–308.

[ARPT02] ALBIN S., ROUGERON G., PEROCHE B., TREMEAU A.: Quality image metrics for synthetic images based on perceptual color differences. *IEEE Transactions on Image Processing 11*, 9 (September 2002), 961–971.

[BM95] BOLIN M. R., MEYER G. W.: A frequency based ray tracer. In *Computer Graphics (SIGGRAPH 95 Conference Proceedings)* (1995), vol. 29, pp. 409–418.

[BM98] BOLIN M. R., MEYER G. W.: A perceptually based adaptative sampling algorithm. In *Computer Graphics (SIGGRAPH 98 Conference Proceedings)* (July 1998), vol. 32, pp. 299–310.

[CCW03] CATER K., CHALMERS A., WARD G.: Detail to attention: Exploiting visual tasks for selective rendering. In *Proceedings of the 14th Eurographics Workshop on Rendering* (Aire-la-Ville, Switzerland, June 25–27 2003), Christensen P., Cohen-Or D., (Eds.), Eurographics Association, pp. 270–280.

[Dal93] DALY S.: The visual differences predictor : an algorithm for the assessment of image fidelity. In *Digital Images and Human Vision* (Cambridge, 1993), Watson A. B., (Ed.), MIT Press, pp. 179–206.

[DD00] DURAND F., DORSEY J.: Interactive tone mapping. In *Rendering Techniques 2000 (Proceedings of the Eleventh Eurographics Workshop on Rendering)* (New York, NY, 2000), Peroche B., Rushmeier H., (Eds.), Springer Wien, pp. 219–230.

[Fer01] FERWERDA J. A.: Elements of early vision for computer graphics. *IEEE Computer Graphics and Applications 21*, 5 (Sept./Oct. 2001), 22–33.

[FZvdBLS97] FARRELL J. E., ZHANG X., VAN DEN BRANDEN LAMBRECHT C. J., SILVERSTEIN D. A.: Image quality metrics based on single and multi-channel models of visual processing. In *Proceedings of the 42nd IEEE Computer Society International Conference (COMPCON'97)* (1997).

Figure 9: *Interpolated(left) and reference(right) pictures for the Photo lab scene*

Figure 10: *Same picture with increased gamma and sample map*

[GH97] GIBSON S., HUBBOLD R. J.: Perceptually-driven radiosity. *Computer Graphics Forum 16*, 2 (June 1997), 129–141.

[Guo98] GUO B.: Progressive radiance evaluation using directional coherence maps. In *Computer Graphcs (SIGGRAPH 98 Conference Proceedings)* (July 1998), vol. 32, Addison Wesley, pp. 255–266.

[HSA91] HANRAHAN P., SALZMAN D., AUPPERLE L.: A rapid hierarchical radiosity algorithm. In *Computer Graphics (SIGGRAPH '91 Proceedings)* (July 1991), vol. 25, pp. 197–206.

[Jen96] JENSEN H. W.: Global illumination using photon maps. In *Eurographics Rendering Workshop 1996* (New York City, NY, June 1996), Pueyo X., Schröder P., (Eds.), Eurographics, Springer Wien, pp. 21–30.

[Kaj86] KAJIYA J. T.: The rendering equation. In *Computer Graphics (SIGGRAPH '86 Proceedings)* (Aug. 1986), vol. 20, pp. 143–150.

[Lub95] LUBIN J.: A visual discrimination model for imaging system design and evaluation. *Vision Models for Target Detection and Recognition* (1995), 245–283.

[LW93] LAFORTUNE E. P., WILLEMS Y. D.: Bi-directional Path Tracing. In *Proceedings of Third International Conference on Computational Graphics and Visualization Techniques (Compugraphics '93)* (Alvor, Portugal, Dec. 1993), Santo H. P., (Ed.), pp. 145–153.

[Mey86] MEYER G. W.: Wavelength selection for synthetic image generation. *Computer Graphics and Image Processing II* (1986), 39–44.

[Mit87] MITCHELL D. P.: Generating antialiased images at low sampling densities. *Computer Graphics 21*, 4 (July 1987), 65–72.

[MRT99] MYSZKOWSKI K., ROKITA P., TAWARA T.: Perceptually-informed accelerated rendering of high quality walkthrough sequences. In *Rendering Techniques '99* (1999), Lischinski D., Larson G. W., (Eds.), Eurographics, Springer-Verlag Wien New York, pp. 5–18.

[MTAS01] MYSZKOWKSI K., TAWARA T., AKAMINE H., SEIDEL H.-P.: Perception-guided global illumination solution for animation. In *Computer Graphics (SIGGRAPH 2001 Conference Proceedings)* (Aug. 2001), vol. 35, pp. 221–230.

[Mul85] MULLEN K. T.: The contrast sensitivity of human colour vision to red-green and blue-yellow chromatic gratings. *J. Physiol.(Lond) 359* (1985), 381–400.

[Mys98] MYSZKOWSKI K.: The visible differences predictor: Applications to global illumination problems. In *Rendering Techniques '98 (Proceedings of Eurographics Rendering Workshop '98)* (New York, NY, 1998), Drettakis G., Max N., (Eds.), Springer Wien, pp. 233–236.

[NMP98] NEUMANN L., MATKOVIC K., PURGATHOFER W.: Perception based color image difference. *Computer Graphics Forum 17*, 3 (Sept. 1998), 233–241.

[Per03] PEREZ F.: *Global Illumination Techniques for the Computation of High Quality Images in General Environments*. PhD thesis, Universitat Politecnica de Catalunya, Barcelona, Spain, 2003.

[PFFG98] PATTANAIK S. N., FERWERDA J. A., FAIRCHILD M. D., GREENBERG D. P.: A multiscale model of adaptation and spatial vision for realistic image display. In *Computer Graphics (SIGGRAPH 98 Conference Proceedings)* (July 1998), vol. 32, pp. 287–298.

[PP99a] PRIKRYL J., PURGATHOFER W.: *Overview of Perceptually-Driven Radiosity Methods*. Tech. Rep. TR-186-2-99-26, Institute of Computer Graphics, Vienna University of Technology, Vienna, Austria, Dec. 1999. Available from http://www.cg.tuwien.ac.at/research/TR.

[PP99b] PRIKRYL J., PURGATHOFER W.: Perceptually-driven termination for stochastic radiosity. In *WSCG '99 (Seventh International Conference in Central Europe on Computer Graphics, Visualization and Interactive Digital Media)* (Plzen-Borey, Czech Republic, Feb. 1999), University of West Bohemia, pp. 418–425.

[RPG99] RAMASUBRAMANIAN M., PATTANAIK S. N., GREENBERG D. P.: A perceptually based physical error metric for realistic image synthesis. In *Computer Graphics (SIGGRAPH 99 Conference Proceedings)* (1999), vol. 33, pp. 73–82.

[RWP*95] RUSHMEIER H., WARD G., PIATKO C., SANDERS P., RUST B.: Comparing real and synthetic images : some ideas about metrics. In *6th Eurographics Workshop on Rendering, Dublin* (June 1995), pp. 213–222.

[VG97] VEACH E., GUIBAS L. J.: Metropolis light transport. In *Computer Graphics (SIGGRAPH '97 Proceedings)* (Aug. 1997), vol. 31, pp. 65–76.

[War94a] WARD G.: A contrast-based scalefactor for luminance display. In *Graphics Gems IV*, Heckbert P., (Ed.). Academic Press, Boston, 1994, pp. 415–421.

[War94b] WARD G. J.: The radiance lighting simulation and rendering system. In *Computer Graphics (SIGGRAPH '94 Proceedings)* (July 1994), vol. 28, pp. 459–472.

[WHSG97] WALTER B., HUBBARD P. M., SHIRLEY P., GREENBERG D. F.: Global illumination using local linear density estimation. *ACM Transactions on Graphics 16*, 3 (July 1997), 217–259.

[ZSP98] ZANINETTI J., SERPAGGI X., PÉROCHE B.: A vector approach for global illumination in ray tracing. In *Computer Graphics Forum* (1998), Duke D., Coquillart S.,, Howard T., (Eds.), vol. 17(3), pp. 149–158.