Thirteenth Eurographics Workshop on Rendering (2002)
P. Debevec and S. Gibson (Editors)

Entropy-based Adaptive Supersampling

Jaume Rigau[†] and Miquel Feixas[‡] and Mateu Sbert[§]

Institut d'Informàtica i Aplicacions, Universitat de Girona, Spain

Abstract

Ray tracing usually needs supersampling to reduce aliasing or noise in the final image. Not all the pixels in the image require the same quantity of rays, thus adaptive supersampling *is implemented by adaptive subdivision of the sampling region, resulting in a refinement tree. We present here a theoretically sound adaptive supersampling method based on entropy, an information theory approach with strong analogies to the decision tree problem where entropy is frequently used as a decision criterion. Our adaptive supersampling algorithm is implemented within a path tracing method and we show that our results compare well to the ones obtained by a classic strategy.*

Categories and Subject Descriptors (according to ACM CCS): I.3.3; I.3.7 [Picture/Image Generation; Three–Dimensional Graphics and Realism]: Display algorithms; Colour, shading, shadowing, and texture / Raytracing.

1. Introduction

In stochastic ray tracing, the integral that gives the flux through a given pixel is computed by Monte Carlo. Rays are traced in a stochastic way through the pixel and the radiance of the hit point on the scene is returned (radiance usually being computed by a random walk method as in path-tracing [24]). Many rays per pixel are frequently required to eliminate aliasing or noise in the final image. However, not all the pixels in a ray-traced image need the same quantity of rays. In order to account for reliable data, the edge of an object, the contour of a shadow, and a high illumination gradient demands a much better treatment than a region with almost uniform illumination . This way of sampling is called *adaptive supersampling*[8, 9].

Stochastic adaptive supersampling is implemented by adaptive subdivision of the sampling region. This subdivision generally corresponds to a binary tree or a quadtree [11, 15]. Subdivision is triggered by the result of a refinement test based on a given measure. New samples are then added into the newly created subregions.

Several supersampling refinement measures have been defined. These measures are based on colour intensities and/or geometry [23, 12, 17]. Other supersampling approaches use also quasi-Monte Carlo [14], signal theory [6], characteristics of the human eye (contrast) [13], human perception and limitations of display devices (tone–operator) [25], and important characteristics (irregular and importance sampling, uncorrelation between the dimensions to sample, complete stratification at each refinement level, efficient reconstruction) [20].

In this paper we introduce a new refinement criterion, complementary to the one defined in [19], with the very important feature that it is based on the recursive expression of the entropy, i.e. its grouping property [3]. The idea behind the new criterion is to obtain sufficient *information* in the refinement tree which results from the recursive decomposition of a pixel in subpixels. The natural way to represent information is by *entropy*, which in our case can be interpreted as a measure of the degree of homogeneity of a pixel or subpixel. Thus, using the entropy criterion means to obtain enough information or homogeneity on a pixel or subpixel.

The fundamental novelty of this approach compared to others classic techniques is that it uses a sound theoretical framework, namely *information theory*, to obtain the refinement process. Also, although we use path tracing in our tests as the ray tracing technique, our approach can be applied to any other stochastic or quasi-MC gathering algorithm.

The organization of this paper is as follows: in section 2 we present some previous work, in section 3 we introduce the setting for an adaptive supersampling algorithm based

[†] jaume.rigau@udg.es
[‡] miquel.feixas@udg.es
[§] mateu.sbert@udg.es

on entropy and present the new algorithm. In section 4 we discuss our results, comparing them with the ones obtained with a classic contrast measures and uniform sampling, and finally we present our conclusions in section 5.

2. Previous Work

In this section we present previous work on the areas of supersampling refinement criteria, entropy and entropy based contrast measures, and decision tree.

2.1. Supersampling Refinement Criteria

The refinement measures used in adaptive supersampling are based on intensities and geometry. They are also useful for an adaptive subdivision of image space for progressive refinement [15]. Some of them have been recently applied in the image based rendering field for weighting pixel colour for reconstruction [16] and adaptive sampling strategies [4,5], and creating a priority schema for sampling in interactive rendering [23].

For the purpose of this paper we review here two widely used measures: contrast and depth difference. In [13], Mitchell presents one the most used intensity measures [2], the *contrast*, defined by

$$C = \frac{I_{\max} - I_{\min}}{I_{\max} + I_{\min}} \tag{1}$$

where $I_{\min}$ and $I_{\max}$ are the minimum and maximum light intensities respectively. As each sample value consists of three separate intensities for red, green, and blue, Mitchell computes a separate contrast for each one and supersampling is done if any contrast is higher than a given threshold. Red, green, and blue thresholds are set to 0.4, 0.3 and 0.6 respectively, based on the relative sensitivity of the visual system.

In [23], Simmons uses a *priority* value p_c based on the above concepts (contrast and perception) [13,9] defined by

$$p_c = .4\frac{r_{\max} - r_{\min}}{r_{\max} + r_{\min}}\overline{r} + .3\frac{g_{\max} - g_{\min}}{g_{\max} + g_{\min}}\overline{g} + .6\frac{b_{\max} - b_{\min}}{b_{\max} + b_{\min}}\overline{b} \tag{2}$$

where max, min, and the overline represent the maximum, minimum, and average values respectively for r, g, and b colour channels.

On the other hand, a useful and simple geometric measure for refinement is *depth difference*, used recently in image based rendering [4,5,16] and interactive rendering [23]. Depth difference is given by

$$p_d = 1 - \frac{d_{\min}}{d_{\max}} \tag{3}$$

where $d_{\max}$ and $d_{\min}$ represent maximum and minimum distance. In [23], p_c and p_d measures are combined with weights of 90% and 10% respectively.

2.2. Entropy

In [22], Shannon defined the *entropy* $H(X)$ of a discrete random variable X with values in the non empty set $\mathcal{X} = \{x_1, \ldots, x_n\}$ as

$$H(X) = -\sum_{i=1}^{n} p_i \log p_i \tag{4}$$

where $n = |\mathcal{X}|$, $p_i = Pr[X = x_i]$ for $i \in \{1 \ldots n\}$, and the convention that $0 \log 0 = 0$ is used by continuity [22]. As $-\log p_i$ represents the *information* associated with the result x_i, the entropy gives the average information or the *uncertainty* of a random variable. Entropy is the most basic information theory measure. As we take the logarithms in base 2, the entropy is expressed in *bits*.

Some relevant properties [22,3] of the entropy are:

1. $0 \leq H(X) \leq \log n$
   - $H = 0$ if and only if all the probabilities except one are zero, this one having the value unity, i.e., when we are certain of the outcome: $p_j = 1$ and $p_i = 0$ for $i \in \{1 \ldots j-1, j+1 \ldots n\}$.
   - $H(X) = \log n$ when all the probabilities are equal. This is the most uncertain situation: $p_i = \frac{1}{n}$ for $i \in \{1 \ldots n\}$.
2. If we equalize the probabilities, entropy increases.
3. Recursivity (grouping): if a choice is broken down into two successive choices, the original H should be the weighted sum of the individual values of H (see figure 1).

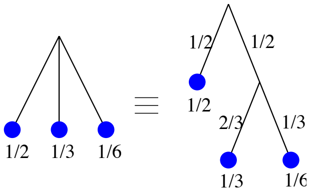


Figure 1: *Recursive or grouping property of the entropy:* $H(\frac{1}{2}, \frac{1}{3}, \frac{1}{6}) = H(\frac{1}{2}, \frac{1}{2}) + \frac{1}{2}H(\frac{2}{3}, \frac{1}{3})$.

It is worth mentioning the case $n = 2$ with $\mathcal{X} = \{x_1, x_2\}$:

$$X = \begin{cases} x_1 & \text{with probability } p, \\ x_2 & \text{with probability } 1-p. \end{cases} \tag{5}$$

The entropy of this probability distribution is called *binary entropy*, H_2. Then $H_2(X) = -p \log p - (1-p) \log(1-p)$. The behaviour of this binary entropy H_2 is shown in figure 2.

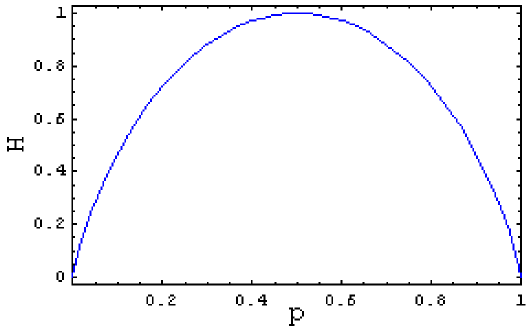


Figure 2: *Binary entropy corresponding to the probability distribution* $(p, 1-p)$ *of random variable* X*. The maximum* $H_2(X) = \log 2 = 1$ *is obtained when* $p = \frac{1}{2}$ *and two minimums* $H_2(X) = 0$ *are obtained when* $p = 0$ *and* $p = 1$.

2.3. Entropy-based Contrast Measures

In [19], the *pixel channel entropy* was introduced. This measure is defined by

$$H^c = -\sum_{i=1}^{n_s} p_i \log p_i \qquad (6)$$

where p_i represents the channel colour fraction of ray i with respect to the sum of the colours of the same channel of all the rays passing through a pixel, and n_s is the number of rays traversing a pixel. Pixel channel entropy can be interpreted as the channel colour homogeneity of the rays passing through the pixel. It can also be considered as the pixel quality.

In order to give a pixel contrast measure between 0 and 1, the pixel channel entropy is normalized with $\log n_s$. Thus, the *pixel channel contrast* can be defined by

$$C^c = 1 - \frac{H^c}{\log n_s} \qquad (7)$$

and represents the channel colour unhomogeneity or contrast of a pixel. Considering all the colour channels, we introduce the global *pixel colour contrast*[19] as

$$\mathbf{C}^c = \frac{\sum_{i=1}^{n_c} w_i C_i^c \overline{c_i}}{\sum_{i=1}^{n_c} w_i} \qquad (8)$$

where the channel contrasts are weighted by perception coefficients w_i and by $\overline{c_i}$, the colour average of channel i of all the pixel rays, which expresses the "importance" for each channel. In an RGB system, accordingly to [18], we can take $w_r = 0.299$, $w_g = 0.587$ and $w_b = 0.114$. These weights try to capture the sensitivity of human colour perception.

A similar measure for the "geometry" of a pixel, the *pixel geometric entropy*[19], is defined by

$$H^g = -\sum_{i=1}^{n_s} p_i \log p_i \qquad (9)$$

where p_i represents the geometric fraction of ray i, $\frac{\cos\theta_i}{d_i^2}$, with respect to the sum of the geometric factors of all the rays traversing a pixel. The geometric information of each ray is given by the angle θ_i which the normal forms at the hit point with the ray, and also by the distance d_i between this point and the eye. Similarly to the colour case, the geometric entropy represents the pixel geometric homogeneity or quality.

From this measure, the *pixel geometric contrast* is defined by

$$C^g = 1 - \frac{H^g}{\log n_s} \qquad (10)$$

which represents the geometric unhomogeneity of a pixel.

Alternative colour and geometric contrast measures can be obtained by substituting the pixel entropy by the binary pixel entropy, which is computed by only considering the maximum and minimum values captured by the pixel.

A combination of colour and geometric contrasts can be considered. This combination enables us to graduate, with a coefficient δ between 0 and 1, the influence of both measures and is given by

$$\mathcal{C} = \delta \mathbf{C}^c + (1-\delta) C^g \qquad (11)$$

2.4. Decision Tree

A decision tree is a representation of a decision procedure for determining a class label to associate with a given example, this is, a classifying decision. At each internal node of the tree, there is a test (question), and a branch corresponding to each of the possible outcomes of the test. At each leaf node there is a class label (answer).

The need to partition a data space into subsets arises frequently in machine learning schemes. This task appears both as a preprocessing step preceding the learning phase and as a step integrated into the induction algorithms. An area where the partioning task is heavily applied is the induction of symbolic classifiers, such as decision trees or classification rules. Typically, the required partitions are binary, but multi-way partitions are used in so-called multisplitting decision trees.

Given a series of examples, a learning algorithm can build a decision tree that will be able to classify new examples. If the new examples are handled correctly, nothing is done. Otherwise, the structure of the tree is modified until the correct results are displayed. The challenge is getting the algorithm to perform well on very large sets of data, handling errors in values (noise), and determining the optimal fit of the tree to the training and test data. Information theory has long been used in decision tree problems [21, 26].

Classification is, in essence, a partioning task: the objective is to find a function that divides the instance space cleanly into class uniform regions by decision boundaries.

The class–coherence of partitions is typically measured by impurity functions [7, 1]. They evaluate both the internal class–coherence of the subsets and the overall complexity of the partition, for example, the number of subsets in the partition or the simplicity of the splitting function. The intent is to find coherent subsets with a low complexity partition. The design of evaluation functions that keep these two effects in good balance is a delicate and still not very well understood issue. One of the most widely utilized impurity measures is the class entropy.

3. Adaptive Supersampling Algorithm Based on Entropy

We present in this section an adaptive supersampling algorithm based on the entropy of the refinement tree. The approach to be used in refinement "will be to make sure that all the samples in a given region are similar in some specified way, so we can feel that we have captured what is happening in a region of the signal. If a region is non uniform, or heterogeneous, then we will typically want to refine our regions until each subregion is uniform"[9]. This process is by nature a recursive process, giving rise to a refinement tree.

3.1. Recursive Entropy Tree

As we have seen in the previous section *entropy* gives us the information content of a random variable, and this information can be recursively expressed. In this section, the recursive property of the entropy (section 2.2) is generalized in the following way:

Consider X a discrete random variable with distribution $p = \{p_1, \ldots, p_n\}$, $G = \{G_1, \ldots, G_m\}$ an m-partition of p with $1 \leq m \leq n, G_i = \{p_{i_1}, \ldots, p_{i_{n_i}}\}, |G_i| = n_i, q_i = \sum_{j=1}^{n_i} p_{i_j}$ and $r_{i_j} = \frac{p_{i_j}}{q_i}$ ($\forall i \in \{1 \ldots m\}$ and $\forall j \in \{1 \ldots n_i\}$).

If $Y, Y_1, \ldots, Y_m$ are the random variables associated to $G, G_1, \ldots, G_m$ with probability distributions $q, r_1, \ldots, r_m$ respectively, then

$$H(X) = \sum_{i=1}^{m} q_i H(Y_i) - \sum_{i=1}^{m} q_i \log q_i = \sum_{i=1}^{m} q_i H(Y_i) + H(Y) \tag{12}$$

This formula can be written as $H(X) = H_{in} + H_{out}$ where $H_{in} = \sum_{i=1}^{m} q_i H(Y_i)$ and $H_{out} = H(Y)$ represent respectively the hidden information (pending to be discovered) and the information already acquired in the descent of the recursion tree. An example of an entropy quadtree is shown in figure 3. The total entropy of the tree is written within the root node.

In our case, formula (12) can be interpreted (for one colour channel) as

- $H(X)$ represents the entropy of all the image.
- $H(Y_i)$ represents the entropy of each root pixel.
- Probability q_i is the colour of pixel i divided by the sum of the colours of all the pixels. It can be considered as the "importance" of pixel i.

This interpretation can be recursively extended to the subpixel levels. Similarly, this framework can be applied to geometric entropy.

3.2. Algorithm

In this section we show how a practical adaptive supersampling algorithm can be obtained from the entropy tree. Although the full algorithm will take into account both colour and geometry, in the following analysis we only consider the colour information of a channel.

The adaptive supersampling algorithm is as follows: On an image plane of n_p pixels we cast first n_s rays per pixel to capture the colour of the hit points and so evaluate the information content (entropy) of each pixel from the colour probability distribution. If the information of a pixel is high enough, i.e. the rays give us sufficient colour homogeneity on that pixel, refinement is not made, and the colour of this pixel is given by the average of all the colours returned by the rays cast. In the contrary case (the contrast is high enough), this pixel is subdivided into n_r regions and we proceed in the same way for each region (subpixel). Finally we obtain a n_r-tree (see figure 3 for $n_r = 4$).

For each root pixel, the algorithm consists of two phases:

- Refinement of a pixel (tree descent)
- Final colour computation of a pixel (tree ascent)

The refinement process is represented in figure 4 where we show a fixed path $R = (r_0, \ldots, r_{N-1})$ with length $N > 0$ arriving at a determined node, which represents the successive subregions (subpixels) selected at each level, where $r_n \in \{1, \ldots, n_r\}$ for $0 \leq n < N$. Level $n = 0$ corresponds to the pixel root. The weight or importance of a node at level n in this path is q^n and can be expressed by

$$q^n = \begin{cases} \frac{\overline{c}^0}{\sum_{k=1}^{n_p} \overline{c}_k^0} & \text{if } n = 0 \\ q^{n-1} p^{n-1} & \text{if } 0 < n \leq N \end{cases} \tag{13}$$

where $\overline{c}^n$ is the colour of n-level subpixel obtained from the rays cast over it and $p^n = \frac{\overline{c}_{r_n}^n}{\sum_{j=1}^{n_r} \overline{c}_j^n}$ is the probability of r_n-subregion. $\overline{c}^n$ can also be expressed by the average of the colours of its respective subregions $\frac{\sum_{j=1}^{n_r} \overline{c}_j^n}{n_r}$.

This q^n value will weight our colour contrasts. In our algorithm, $\overline{c}^n$ and q^n do not have to be necessarily normalized, thus we drop $\sum_{k=1}^{n_p} \overline{c}_k^0$. Equivalently, we have

$$q^n = \overline{c}^0 \prod_{\ell=1}^{n} p^{\ell-1} \tag{14}$$

Its accuracy can be improved by taking more accurate colour

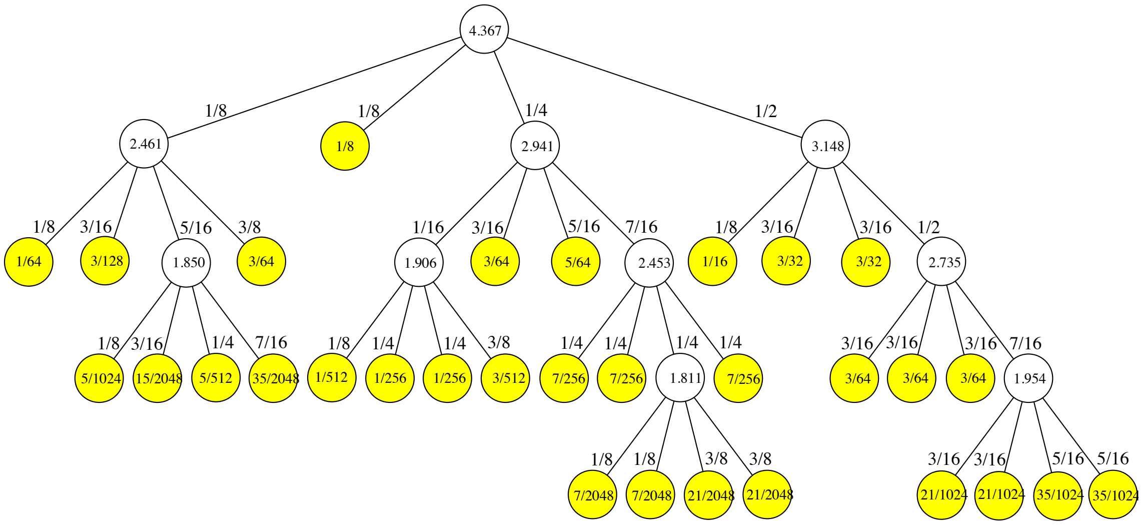


Figure 3: *Entropy of a quadtree. Each internal node (white) stores the entropy of its respective subtree ($H_{in} + H_{out}$). The total entropy is written within the root node. Each leaf (yellow) stores its probability.*

values, $\overline{c}^n \cong \overline{c}^{n-1}_{r_{n-1}}$, and thus we obtain an optimal and less costly expression to compute q^n:

$$q^n = \frac{\overline{c}^n}{n_r{}^n} \tag{15}$$

Proof: We proceed by induction. When $n = 0$,

$$q^n = \overline{c}^0 \cong \frac{\overline{c}^n}{n_r{}^n} \tag{16}$$

Hypothesis: $\forall \ell < n. q^\ell = \frac{\overline{c}^\ell}{n_r{}^\ell}$. Then, for n

$$\begin{aligned} q^n &= q^{n-1} p^{n-1} = \frac{\overline{c}^{n-1}}{n_r{}^{n-1}} \frac{\overline{c}^{n-1}_{r_{n-1}}}{\sum_{j=1}^{n_r} \overline{c}_j^{n-1}} \\ &\cong \frac{\frac{(\sum_{j=1}^{n_r} \overline{c}_j^{n-1}) - \overline{c}^{n-1}_{r_{n-1}} + \overline{c}^n}{n_r}}{n_r{}^{n-1}} \frac{\overline{c}^n}{(\sum_{j=1}^{n_r} \overline{c}_j^{n-1}) - \overline{c}^{n-1}_{r_{n-1}} + \overline{c}^n} \\ &= \frac{\overline{c}^n}{n_r{}^n} \end{aligned} \tag{17}$$

The contrast measure we use is obtained from (8) and (11), but now the "importances" q weighting the different channels come from the entropy of the refinement tree. Thus the contrast at the n-level is given by

$$\mathbf{C}^{c^n} = \frac{\sum_{i=1}^{n_c} w_i C_i^{c^n} q_i^n}{\sum_{i=1}^{n_c} w_i} \tag{18}$$

and the colour and geometric combination is

$$\mathcal{C}^n = \delta \mathbf{C}^{c^n} + (1-\delta) C^{g^n} \tag{19}$$

After this computation, we decide to subdivide whether the contrast or unhomogeneity of a pixel or subpixel, is greater than a given threshold ε. In the refinement test, two different contrasts are possible: normalized and binary. The latter is a bit cheaper than the first because it uses only two values for its computation.

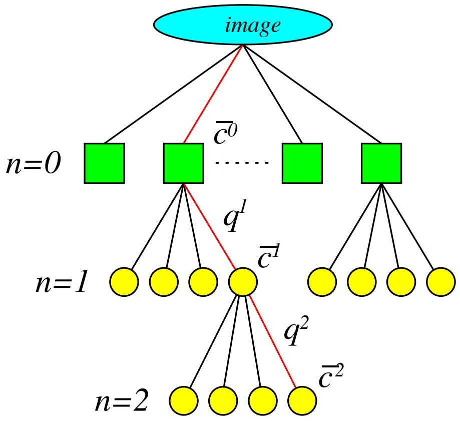


Figure 4: *A path in the refinement tree.*

The descent in the refinement tree can be interpreted as a progressive information gain. The information acquired at each level is additively combined so that, at the end of the refinement process, the total information of the tree is the sum of the information obtained over all the branches.

Importance sampling is also naturally integrated in the process. Following importance sampling criterion a function should be sampled proportional to its value which is what we get with our adaptive descent.

If the test fails, the final colour computation process (ascent) starts. The value of c^n is recursively obtained from the final colours returned by each subpixel which are already weighted by their respective areas (included in q).

Finally, in figure 5 we show the algorithm. Some informations about it are

- i is the pixel, s is the set of samples of this subpixel already sampled at the parent level ($\emptyset$ for the root), ℓ is the depth level (0 for the root).
- We sample the pixel to obtain the colours of each channel and geometric data using S (s is reused for efficiency). Any sampling procedure can be used, stochastic or quasi-MC. We use here a stratified stochastic sampling.
- A is the average of colour and geometric information.
- $Q = (Q_1, Q_2)$ are the unnormalized probabilities for the colour and geometric respectively. Q_2 is used in contrast combination.
- In the refinement test, either a maximum depth level or a minimum division area is included.
- $\oplus$ is the colour addition (colour already arrives correctly weighted).

```
Adaptive($i,s,\ell$)
  1. $S =$ Sample($i,s$)
  2. $A =$ Average($S$)
  3. $Q = \frac{A}{n_r{}^{\ell}}$
  4. if Contrast($Q,S$) $> \varepsilon$ then
     a. $J =$ Partition($i$)
     b. $c =$ Clear()
     c. For each pixel $j \in J$ do
        i. $c = c \oplus$ Adaptive($j, S_j, \ell+1$)
  5. else
     a. $c = Q_1$
  6. return($c$)
```

Figure 5: *Entropy-based adaptive supersampling division algorithm.*

4. Results

We present here results for the *Cornell box* scene. In figure 6 our approach is compared with a classic contrast, similarly to [23], which is a combination of colour contrast (2) and geometric contrast (3). Perceptual coefficients in (2) are taken equal to 0.299, 0.587, and 0.114 (see section 2.3). In both cases we use a combination as in (11), where $\delta = 0.9$, and the tree depth level bound is set to 4.

Four rays are cast stratifiedly at each tree node (pixel and subpixel) to compute the contrast measures for the refinement decision. These rays are reused at the next levels in the tree, whenever necessary. An implementation of classic path-tracing with next event estimator [10] has been used to compute all images. In all the tests we compare with the same average number of rays per pixel as the cost of both our algorithm and the classic adaptive approach is similar. All presented images are unfiltered.

The average number of rays per pixel is the same (60) for all the images, with a minimum of 4 rays per pixel. We show in figures 6(a,b) the results for our approach ((a) entropy contrast and (b) binary–entropy contrast), and in figure 6(c) the result obtained with the classic approach. Supersampling temperature maps of figures 6(I) are shown in figures 6(II) (the red colour corresponds to the highest supersampling, the blue colour to the lowest).

We see from comparing the images in figures 6(a,b) and figures 6(c) that the entropy contrast is much better than the classic contrast used here. Observe for instance the ceiling, the shadows and the mirroring wall. A drawback of our approach are the peaks of high radiance that we observe at the right wall because this region is undersampled in our method. However, this effect can be easily eliminated by a filtering technique. Finally, the comparison of the supersampling temperature maps in figures 6(II) shows a better discrimination of complex regions of the scene in the entropy case, figures 6(a,b)(II), against the classic contrast case, figure 6(c)(II). This explains the better results obtained with our approach.

5. Conclusions

We have presented a new adaptive supersampling algorithm for stochastic raytracing based on the recursive expression of the entropy of a pixel, computed from the sampled intensity and the distance and orientation of the first hit object in the scene. Thus, in contrast to previous heuristic techniques, our approach uses a sound theoretical framework (information theory) in order to establish the refinement criterion.

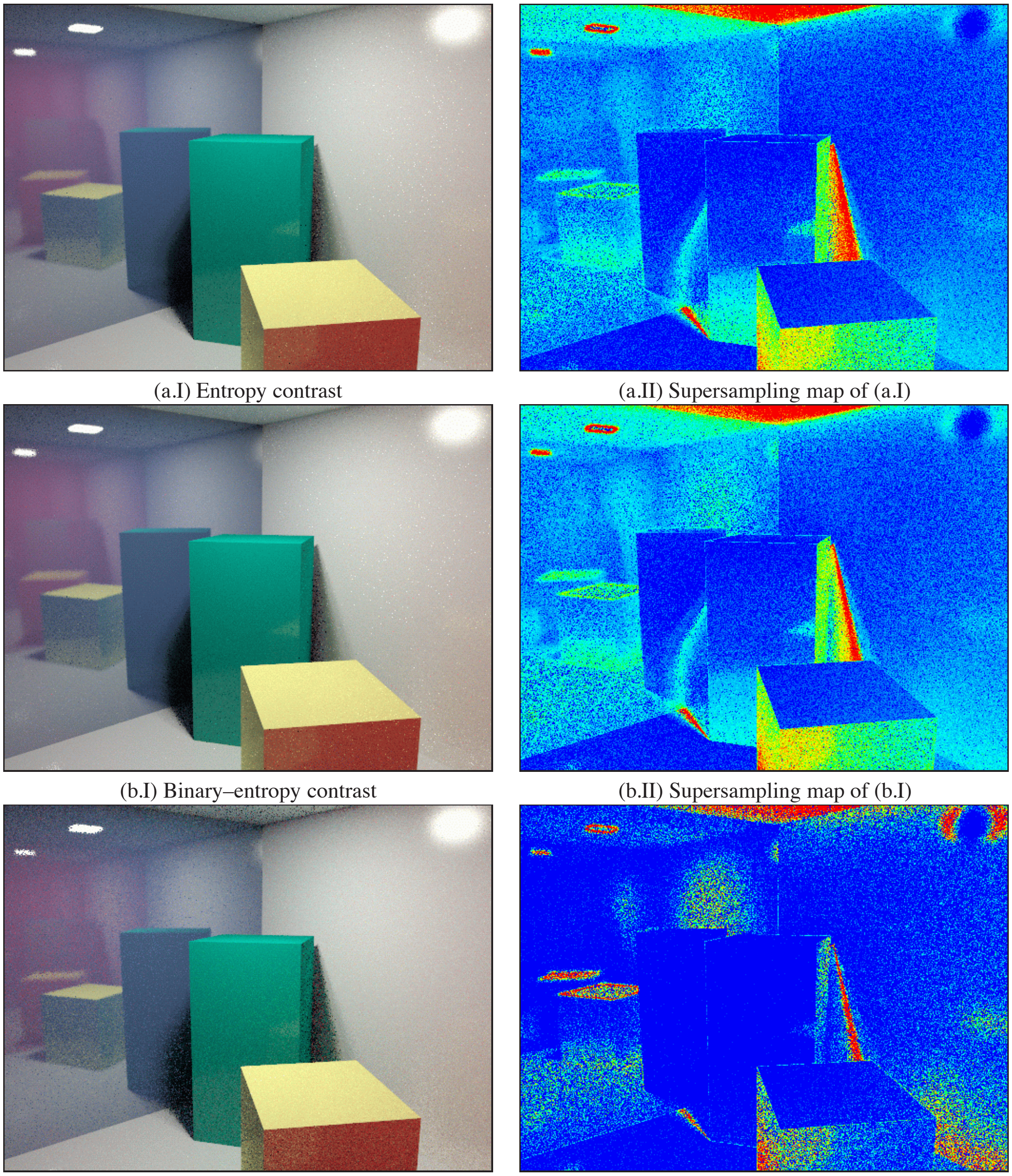
Our method, demonstrated here for path tracing, can be used in all raytracing algorithms. The results obtained show that the new refinement algorithm improves substantially over a classic adaptive refinement technique. Future work will be addressed towards finding automatic criteria for the threshold used in the refinement test.

Acknowledgements

To the anonymous reviewers for their comments. This project has been funded in part with grant numbers TIC-2001-2416-C03-01 of the Spanish Government, HU2000-0011 (joint action) of the Spanish–Austrian Governments, and 2001-SGR-00296 of the Catalan Government. All the images have been obtained with the *RenderPark*[10] software. The *Cornell box* reference scene is from Cornell University.

References

1. Leo Breiman, Jerome H. Friedman, Richard A. Olshen, and Charles J. Stone. *Classification and Regression Trees*. Statistics/Probability. Wadsworth Publishing Company, 1984.



(a.I) Entropy contrast

(a.II) Supersampling map of (a.I)

(b.I) Binary–entropy contrast

(b.II) Supersampling map of (b.I)

(c.I) Classic contrast

(c.II) Supersampling map of (c.II)

Figure 6: Cornell box *scene. All images are obtained with an average of 60 rays per pixel with a minimum of 4 rays per pixel. Entropy contrast (a), binary-entropy contrast (b) and classic contrast (c) are shown with the final image (I) and supersampling colour maps (II).*

2. Terrence M. Caelli. *Visual Perception: Theory and Practice*. Pergamon Press, Oxford, UK, 1981.

3. Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley Series in Telecommunications, 1991.

4. Lucia Darsa and Bruno Costa. Multi-resolution representation and reconstruction of adaptively sampled images. In *Proceedings of IX Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI'96)*, pages 321–328, October 1996.

5. Lucia Darsa, Bruno Costa, and Amitabh Varshney. Navigating static environments using image-space simplification and morphing. *1997 Symposium on Interactive 3D Graphics*, pages 25–34, April 1997.

6. Mark A.Z. Dippé and Erling H. Wold. Antialiasing through stochastic sampling. *Computer Graphics (Proceedings of SIGGRAPH'85)*, 19(3):69–78, July 1985. Held in San Francisco (CA), USA.

7. Usama M. Fayyad and Keki B. Irani. The attribute selection problem in decision tree generation. In *Proceedings of the 10th National Conference on Artifical Intelligence'92*, pages 104–110. MIT Press, 1992.

8. Andrew S. Glassner, editor. *An Introduction to Ray Tracing*, chapter An Overview of Ray Tracing. Academic Press, San Diego (CA), USA, 1989.

9. Andrew S. Glassner. *Principles of Digital Image Synthesis*. Morgan Kaufmann Publishers, San Francisco (CA), USA, 1995.

10. Computer Graphics Research Group. *RenderPark: A Photorealistic Rendering Tool*. Katholieke Universiteit Leuven, Leuven, Belgium, November 2000.

11. James T. Kajiya. The rendering equation. *Computer Graphics (Proceedings of SIGGRAPH'86)*, 20(4):143–150, August 1986. Held in Dallas (TX), USA.

12. Mark E. Lee, Richard A. Redner, and Samuel P. Uselton. Statiscally optimized sampling for distributed ray tracing. *Computer Graphics (Proceedings of SIGGRAPH'85)*, 19(3):61–67, July 1985. Held in San Francisco (CA), USA.

13. Don P. Mitchell. Generating antialiased images at low sampling densities. *Computer Graphics (Proceedings of SIGGRAPH'87)*, 21(4):65–72, July 1987. Held in Anaheim (CA), USA.

14. Ryutarou Ohbuchi and Masaki Aono. Quasi–monte carlo rendering with adaptive sampling. Technical report, Tokyo Research Laboratory, IBM Japan Ltd., Tokyo, Japan, 1996.

15. James Painter and Kenneth Sloan. Antialiased ray tracing by adaptive progressive refinement. *Computer Graphics (Proceedings of SIGGRAPH'89)*, 23(3):281–288, July 1989. Held in Boston (MA), USA.

16. Kari Pulli, Michael F. Cohen, Tom Duchamp, Hugues Hoppe, Linda Shapiro, and Werner Stuetzle. View-based rendering: Visualizing real objects from scanned range and color data. In Julie Dorsey and Philipp Slusallek, editors, *Rendering Techniques'97 (Proceedings of the 8th Eurographics Workshop on Rendering)*, pages 23–34, New York (NY), USA, June 1997. Springer-Verlag Vienna-New York. Held in St. Etienne, France.

17. Werner Purgathofer. A statistical method for adaptive stochastic sampling. *Eurographics'86: Proceedings of the European Conference and Exhibition*, 11(2):157–162, August 1986. Held in Lisbon, Portugal.

18. Comité Consultatif International Radio. Encoding parameters of digital television for studios. In *Recommendations of the CCIR (also resolutions and options)*, volume XI, chapter Part 1 - CCIR Rec. 601-2, pages 95–104. Broadcasting Service (TV), International Telecommunication Union, Geneva, Switzerland, 1990.

19. Jaume Rigau, Miquel Feixas, and Mateu Sbert. New contrast measures for pixel supersampling. In *Proceedings of Computer Graphics International'02*, July 2002. Held in Bradford, UK. Available in *http://ima.udg.es/ rigau*.

20. Christophe Schlick. An adaptive sampling technique for multidimensional ray tracing. In *Proceedings of the 2nd Eurographics Workshop on Rendering*, pages 48–56, May 1991. Held in Barcelona, Spain.

21. Ishwar K. Sethi and G.P.R. Sarvarayudu. Hierarchical classifier design using mutual information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 4(4):441–445, July 1982.

22. Claude E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27:379–423,623–656, July,October 1948.

23. Maryann Simmons and Carlo H. Séquin. Tapestry: A dynamic mesh-based display representation for interactive rendering. In Bernard Péroche and Holly Rushmeier, editors, *Rendering Techniques 2000 (Proceedings of the 11th Eurographics Workshop on Rendering)*, pages 329–340, New York (NY), USA, June 2000. Springer-Verlag Vienna-New York.

24. László Szirmay-Kalos. *Monte-Carlo Methods in Global Illumination*. Institute of Computer Graphics, Vienna University of Technology, Vienna, Austria, 1999.

25. Rasmus Tamstorf and Henrik W. Jensen. Adaptive sampling and bias estimation in path tracing. In Julie Dorsey and Philipp Slusallek, editors, *Rendering Techniques'97 (Proceedings of the 8th Eurographics Workshop on Rendering)*, pages 285–295, New York (NY), USA, June 1997. Springer-Verlag Vienna-New York. Held in St. Etienne, France.

26. Satoshi Watanabe. Pattern recognition as a quest for minimum entropy. *Pattern Recognition*, 13(5):381–387, 1981.