Computer Science



# Optimizing Ray Tracing with Visual Coherence



Nathan Loofbourrow*
Steven A. Shafer

December 8, 1993
CMU-CS-93-209



Carnegie Mellon

# Optimizing Ray Tracing with Visual Coherence

Nathan Loofbourrow*
Steven A. Shafer

December 8, 1993
CMU-CS-93-209

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

**Abstract**

We present a method that directs sampling and reconstruction for image generation. This method reduces the cost of the sampling stage, allows a reconstruction of the scene to be progressively refined during the sampling process, and produces a representation of the scene that permits flexibility in the generation of images.

The method employs a set of *region estimators* to model and evaluate the samples that are taken of the image. These models describe coherent regions of the image plane. The region estimators are used to direct the sampling process and to provide a description of the scene for reconstruction.

The use of region estimators removes the dependence of the reconstruction process upon the sampling process by providing a compact and accurate representation of the sampled data. This representation permits the resolution and filter kernel to be chosen at reconstruction. An algorithm is presented to perform this reconstruction efficiently.

*Currently with the Department of Computer and Information Science, The Ohio State University.

This research was sponsored by the Avionics Laboratory, Wright Research and Development Center, Aeronautical Systems Division (AFSC), U.S. Air Force, Wright-Patterson AFB, Ohio 45433-6543 under Contract F33615-90-C-1465, ARPA Order No. 7597.

## 1. Introduction

Point sampling is a convenient and intuitive method for the generation of pixel-based images. Its generality permits complex imaging processes such as ray tracing to be evaluated. An image is described by a function whose value is the intensity of light reaching the viewer at a given point. The parameters to the function typically describe a two-dimensional point in the viewer's space.

However, the generation of images via point sampling is difficult, as the total light reaching a given region of an image is represented by an integral of the image function over a finite domain (or convolved with a finite filter).

We present a new method for the evaluation of this integral by means of point sampling. *Polynomial regression* is employed in order to provide a continuous model that estimates the image function within a bounded region. This model is used to evaluate the data received via sampling. During reconstruction, the model is employed to provide a closed-form solution to the image formation integral. This process is analogous to *polynomial interpolation*, which is a common approximation method used in numerical integration[11]. The resulting integral may be evaluated over any region in image space, providing a description of the image that is independent of pixel size, shape, or resolution.

In the following discussion, the image function shall be assumed to be two-dimensional, and further, that it evaluates to a single real value representing the intensity of light reaching the viewer. This restricts our examples to monochromatic scenes, although this restriction is not endemic to the technique. We shall be referring to surfaces throughout our discussion, but wish to emphasize that these are not intended to relate directly to the surfaces that define objects being rendered.

## 2. Previous approaches

Adaptive sampling methods have commonly been associated with ray tracing. These methods attempt to combat both the problem of missed features and later, the computational expense of ray tracing. Initial work was directed towards the supersampling of individual pixels[10, 5], though the technique was later extended to encompass subdivision of larger regions of image plane[6, 9, 1].

These approaches all use measures of variance versus the mean (within some measure of tolerable error) as a stopping condition for sampling. This low level constraint forces rendering to proceed fully for all regions of an image whose variance from the mean exceeds a perceptible threshold. Thus, in these methods regions of nonconstant intensity will be subdivided and approximated by a number of regions of constant intensity. This is computationally expensive and wasteful of information in regions of high intensity variation.

## 3. Sampling

Two assumptions form the basis of adaptive sampling: the first, that we may describe some areas of an image given a small number of samples taken in that area; and the second, that we may detect such areas by examining the samples taken.

The first assumption introduces a concept we shall call *visual coherence*. We define this concept as a lack of visual features, such as edges or texture, at the visible scale. For the purpose of intuition, we associate this concept with the notion of zero-order continuity; however, this is done cautiously, as noncontinuous surfaces often appear both visually and in implementation as continuous surfaces. We will instead use the assumption that if the positions of image samples and their intensities exhibit statistically significant correlation (given sufficient confidence that features have been located by the samples), then the area is coherent.

This suggests the manner in which we detect visual coherence: by the creation and evaluation of models,

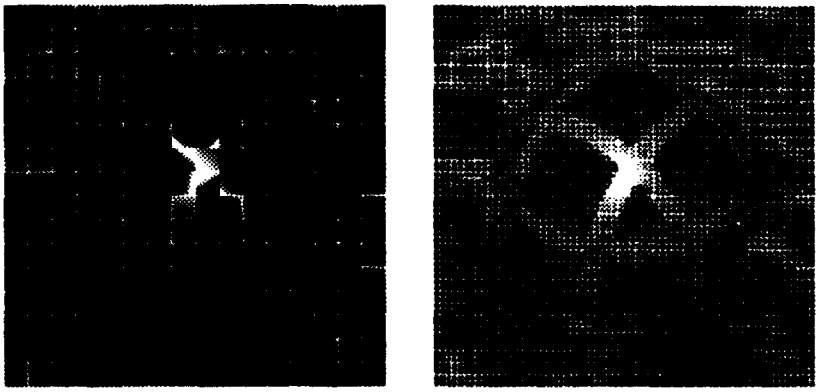

Figure 1: Adaptive subdivision of the image space. (a) Early subdivision; (b) later subdivison.

or *estimators*, that exhibit coherence. These estimators are surfaces that are conformed to the sampled data. Painter and Sloan use a plane of constant intensity as the model for the image in a given region[9]; this plane intersects a single sample point placed in that region.

We introduce the use of polynomials as region estimators. By fitting a low-order polynomial surface (such as a constant, bilinear, biquadratic, or bicubic surface) to sampled data, we may better describe the data collected by a small number of samples. This process is illustrated in one dimension in Figure 2. Surfaces are obtained by means of statistical regression[2]; in our case, this is done by a least squares method[1], which for bicubic surfaces reduces to a 10 by 10 set of linear equations given in Appendix 1. By the continuous nature of our estimates, we are guaranteed that if our models possess visual coherence, then a well-fit estimate is no less coherent than the data in the original scene. This guarantee is used to curtail sampling of regions that exhibit coherence.

Estimates are produced in rectangular regions of space formed by subdivision of the entire image by a k-D tree[3] (after Kajiya[5] and Painter and Sloan[9]). However, we place two constraints upon the subdivision thus performed: the first, that the subdivision must begin at a minimum level of resolution (that is, to ensure that the smallest visible object is detected), and the second, that subdivision must proceed in order of importance. This second constraint, in our context, is used simply to force nonsquare images to be subdivided more often along its long axis than its short axis. In order to derive maximum benefit from this subdivision strategy, samples are taken with uniformly random distribution across the region. This permits a subdivided region to have its samples distributed to the two resulting regions while maintaining this same distribution within each subregion. If some caching of information is permitted, we can enforce this same relationship for $n$-rook sampling patterns[7].

Once an estimate has been produced, we must determine if the estimate fits the sampled data well. If we choose to overconstrain the surfaces used as region estimators (by means of presenting more samples for surface fitting than the degrees of freedom the surface possesses), we retain the ability to check the fit of the surface against the sampled data. This is accomplished by evaluating the region estimate against the sampled data at the points at which the region was sampled. The difference between the estimated value at each point and the actual sampled value is the error. This error encompasses both the coherence of the data in the region and the fit of the model to that data; should either (or both) be unsatisfactory, the estimate will be rejected. While this may cause the false rejection of coherent regions, it will not accept a region whose sampled data is incoherent.

We are also required to constrain our resulting image description to exhibit coherence between regions,

[1] The method attempts to minimize the summed squared error of the estimate, which may also be conveniently perceived as the variance of the sampled data with respect to our estimate.

to prevent discontinuities from appearing at the border between two regions. This may be done by testing the proximity of two surfaces along the edge they share in the image. While the function describing the polynomial surface along its edge is trivially obtained, it is only slightly more costly to evaluate both surfaces at four points along their common edge. The maximum error between these four points will be the maximum at that edge for all surfaces of cubic degree or lower.

It should be noted that the regions we have described are not defined in terms of groups of pixels. We adopt the convention of choosing a *target resolution* at which the image is described. This resolution is the maximum density at which the image may be divided into regions. This constrains a region to be sufficiently small to contain a sufficient number of samples (as mentioned earlier) at a fixed number of samples per region, yet no smaller than the target resolution. This target resolution is presumed to approximate the maximum resolution at which reconstruction will be attempted (that is, the maximum sampling density of our output function), which implies that regions are approximately pixel-sized in the ideal case.

## 4. Reconstruction

An excellent discussion of the difficulties of reconstruction from nonuniform samples appears in a paper by Mitchell[6]; it is sufficient to state that we wish to use as much of the spatial information obtained during sampling as possible.

We choose therefore to use the k-D tree generated during sampling as a basis for reconstruction. This has particular advantage, as each leaf in this tree describes a region whose samples have been distributed with uniform randomness. Most importantly, each leaf contains a region estimate, which is a polynomial surface we have guaranteed to fit the sampled data within some error tolerance. The surface was also fit with even weight for all points in the region, as it was obtained from this set of uniformly random sample points; it is thus resistant to the effects of sample density fluctuation mentioned in Mitchell's paper. It is this surface that we shall employ for reconstruction of the image.

We are now faced with a much simpler problem than in previous work, as interpolation of our data has already been performed in advance of reconstruction. Our task is merely to convolve this data with an appropriate filter and integrate. We will discuss this process in terms of our bicubic surface, as all other surfaces employed in this work are degenerate forms of this surface.

Let us first employ a box filter for reconstruction of the image. The integral for a single pixel centered at $(u, v)$, for image function $f$ and filter $p$,

$$f(u,v) * p(u,v) = \int\int_{(-\infty,-\infty)}^{(\infty,\infty)} f(u,v)p(u-x, v-y)dxdy \tag{1}$$

reduces under filtering by a $2a \times 2b$ box filter:

$$f(u,v) * p(u,v) = \int\int_{(u-a,v-b)}^{(u+a,v+b)} f(x,y)dxdy \tag{2}$$

This simplified form permits easy substitution of our bicubic surface

$$f(u,v) = au^3 + bu^2v + cuv^2 + dv^3 + eu^2 + fuv + gv^2 + hu + jv + k \tag{3}$$

into Equation 2, which may then be explicitly integrated:

$$f(u,v) * p(u,v) = \left[ \frac{ax^4y}{4} + \frac{bx^3y^2}{6} + \frac{cx^2y^3}{6} + \frac{dxy^4}{4} + \frac{ex^3y}{3} + \frac{fx^2y^2}{4} + \frac{gxy^3}{3} + \frac{hx^2y}{2} + \frac{ixy^2}{2} + kxy \right]_{(u-a,v-b)}^{(u+a,v+b)} \tag{4}$$

Our lower order surfaces may be found as degenerate forms of equations 3 and 4.

We may thus convolve our image with a box filter by evaluating the above integral at the four corners of each pixel. If a multiple stage filter (such as those suggested by Heckbert[4] and Mitchell[6]) is desired, further integration of Equation 4 may be performed; alternately, the integral may simply be supersampled over multiple regions in each pixel, which will lead naturally into the second stage of Mitchell's multistage filter.

If these suggested filters are insufficient, any finite filter that can be approximated by polynomials may simply be substituted into Equation 1 along with the cubic in Equation 3 and integrated symbolically. The same technique may be attempted for filters that are easily integrated, although polynomial approximations are likely to be more efficient. Some possible examples are described by Mitchell and Netravali in [8], though the use of absolute-value symmetry in those filters may necessitate integration by parts.

It should be noted at this point that many pixels will be covered by more than one region, and thus require an integral encompassing more than one estimate. We thus perform integration by parts for each of these pixels, which consists of the sum of the contributions from each integral across the region of the pixel it covers. It is important to ensure that this coverage is adequately considered, meaning that the contribution of a given region in space is the convolution of the region with its filter; for a filter $n$ pixels wide, this means that a region of dimension $a \times b$ contributes to a region $(a + n) \times (b + n)$ pixels in size.

## 5. Algorithm

With the concept of region estimation for sampling and reconstruction, we may define a new algorithm for ray tracing and point sampled rendering that provides greater efficiency over coherent regions of the image, permits progressive refinement of an image during rendering, and produces a description that may be sampled for display at a range of resolutions. Sampling and reconstruction are described in two independent parts, as was done in previous discussion.

Our sampling algorithm can be described as follows:

1. Divide the image into a sufficient number of regions, as determined by the target resolution.
2. Iteratively
   (a) Choose a region that has not yet been well described by its estimate (as estimated by its error vs. the estimate).
   (b) If the region is not sufficiently sampled, throw a number of samples into the region.
   (c) Repeat
      i. Fit a given estimating surface to the data.
      ii. Determine the error of that estimate.

      ...until a good fit is reached.
   (d) If the region remains ill-described, and is no smaller than the target resolution,
      - Split it in half along its long axis and distribute its samples accordingly.

      Otherwise,
      - Output the computed estimate.

   ...until all regions have been well described.

Should incremental refinement be desired during rendering, the intermediate estimates may be shown to the user during the subdivision process. It is likely to be sufficient to merely sample the estimates at pixel

| *Target resolution* | *Sampling runtime* | *Regions generated* |
|---|---|---|
| 16×16 | 8.8 | 252 |
| 32×32 | 31.1 | 932 |
| 64×64 | 98.9 | 3010 |
| 128×128 | 276 | 8284 |
| 256×256 | 739 | 21950 |
| 512×512 | 2040 | 59782 |

Table 1: Performance of the sampling algorithm. The image is rendered at a target resolution, which controls the minimum and maximum levels of subdivision. Sampling runtime is shown as a function of target resolution; this runtime is dominated by the ray tracing procedures. All runtimes are in seconds.

| *Resolution of input data* | 16×16 | 32×32 | 64×64 | 128×128 | 256×256 | 512×512 |
|---|---|---|---|---|---|---|
| *Reconstruction resolution* | *Reconstruction runtime* | | | | | |
| 256×256 | 6.4 | 6.5 | 6.7 | 7.6 | 9.7 | 16.9 |
| 297×297 | 9.2 | 10.2 | 12.0 | 15.2 | 21.6 | 36.1 |
| 411×411 | 16.8 | 18.4 | 20.8 | 24.7 | 32.2 | 48.7 |
| 512×512 | 25.2 | 25.2 | 25.6 | 26.3 | 28.7 | 34.5 |
| 1024×1024 | 100.3 | 100.4 | 100.7 | 101.8 | 103.4 | 108.7 |

Table 2: Performance of the reconstruction algorithm. The image is reconstructed (with box filtering) from a particular representation, shown at bottom, at a given image resolution (in pixels), shown at left.

centers for this purpose, though the reconstruction algorithm should be employed as a final (antialiasing) pass.

Our reconstruction algorithm may be described as follows:

- For each region estimate output during sampling,
  - For all pixels whose bounds intersect the region,
    1. Determine the bounds of that intersection.
    2. Evaluate the integral over those bounds.
       * For rectangular bounded regions, this requires that the indefinite integral be evaluated at the four corners of the region (see Equation 4).
    3. Add this integral to the value of the pixel.

This algorithm assumes that overlapping regions are not present in the image description; this assumption is preserved by our sampling algorithm.

## 6. Results

Best results were obtained through the use of constant (average) and linear estimates for image sampling. This is in part due to the tendency for polynomials to extrapolate from given data, especially when higher order polynomials are employed. Linear estimates appear to balance the cost of sufficient samples for estimation versus the need to subdivide areas that may possess coherence.

Our goal in sampling is to adaptively subdivide only those regions which are incoherent. Those regions that exhibit coherence need not be sampled further, and so may be removed from consideration during sampling. This appears in Table 1 as a sublinear number of regions generated versus the target resolution. Sampling runtime is not significantly increased by adaptive sampling overhead.

In reconstruction, we would like runtime to exhibit some independence with respect to the size of the input data set. This may be observed in Table 2 for a wide range of resolutions. The result is greatest in the cases where the reconstructed resolution is approximately equal or greater than the target resolution used when sampling.

It is important to note that the choice of resolution has a strong effect on the reconstruction process—the number of "difficult" pixels whose value depends on more than one region increases. In Figure 9, a resolution is chosen that is a power of two, resulting in a low number of difficult pixels; Figure 10 illustates the computational difficulty for a resolution that is significantly distant from a power of two. This is a significant reason for the increased performance of the algorithm at 512×512 pixels versus smaller "difficult" sizes at the highest resolution (Table 2).

All timings were performed on a 25 MHz 68040 NeXTcube.

## 7. Conclusions

Researchers have achieved speedups in sample-based rendering by analysis of image contrast and variance. However, by assuming the image to be constant over a region of interest, they have overlooked the higher-level coherent properties of the image. By identifying coherence in the values of image samples, we have further reduced the cost of ray tracing and other sample-based rendering.

This insight is derived not from the traditions of computer graphics, but from computer vision. Image generation and image understanding share some parallel goals. In some sense, the rendering problem is an ideal problem in image understanding, as the renderer is permitted to generate greater amounts of data about the scene upon demand. It may be sufficient to perform just enough work to permit a rendered image to be computationally understood to be the scene that it describes.

## 8. Acknowledgements

Our thanks to all the members of the Calibrated Imaging Laboratory at Carnegie Mellon, particularly Reg Willson, Carol Novak, and Jim Moody. Special thanks are due to John Krumm for his assistance with curvefitting derivations. We would like to thank Dr. Richard Parent and Sheryl Rose for providing access to equipment at The Ohio State University for the completion of this work. Some of the image functions sampled for this report were the product of the *Rayshade 4.0* libraries by Craig Kolb and Rob Bogart.

## A Appendix: Polynomial Regression

The $10 \times 10$ array of linear equations to fit a bicubic surface to sampled data points $f_1 \ldots f_n$ for sample positions $(x_1, y_1) \ldots (x_n, y_n)$ are derived from the following equations:

$$v_i = \left[ \begin{array}{cccccccccc} x_i^3 & x_i^2 y_i & x_i y_i^2 & y_i^3 & x_i^2 & x_i y_i & y_i^2 & x_i & y_i & 1 \end{array} \right]$$

so that $v_{i1} = x_i^3$, $v_{i2} = x_i^2 y_i$, ..., and:

$$\sum_{i=1}^{n} \sum_{j=1}^{10} v_{ij} \cdot v_{ik} \cdot s_j = \sum_{i=1}^{n} f_i v_{ik}, \ \forall k \in \{1 \ldots 10\}.$$

This may be solved for the solution vector $s$, which will be a vector of coefficients such that $s \cdot v^T$ is a

cubic best fitting the given data. In this context, $v = \begin{bmatrix} x^3 & x^2y & \cdots & 1 \end{bmatrix}$ for the point $(x, y)$. Degenerate solutions to the above equations will produce the lower-order polynomials.

## References


[1] AKIMOTO, T., MASE, K., AND SUENAGA, Y. Pixel-selected ray tracing. *IEEE Computer Graphics & Applications 11*, 4 (July 1991), 14–22.

[2] ANDERSON, T. W., AND SCLOVE, S. L. *An Introduction to the Statistical Analysis of Data.* Houghton Mifflin, 1978.

[3] BENTLEY, J. L., AND FRIEDMAN, J. J. Data structures for range searching. *ACM Computing Surveys 11*, 4 (1979), 397–409.

[4] HECKBERT, P. S. Filtering by repeated integration. *Computer Graphics (Proceedings of SIGGRAPH '86) 20*, 4 (1986), 315–321.

[5] KAJIYA, J. T. The rendering equation. *Computer Graphics (Proceedings of SIGGRAPH '86) 20*, 4 (1986), 143–150.

[6] MITCHELL, D. P. Generating antialiased images at low sampling densities. *Computer Graphics (Proceedings of SIGGRAPH '87) 21*, 4 (1987), 65–69.

[7] MITCHELL, D. P. Spectrally optimal sampling for distribution ray tracing. *Computer Graphics (Proceedings of SIGGRAPH '91) 25*, 4 (1991), 157–164.

[8] MITCHELL, D. P., AND NETRAVALI, A. N. Reconstruction filters in computer graphics. *Computer Graphics (Proceedings of SIGGRAPH '88) 22*, 4 (1988), 221–228.

[9] PAINTER, J., AND SLOAN, K. Antialiased ray tracing by adaptive progressive refinement. *Computer Graphics (Proceedings of SIGGRAPH '89) 23*, 3 (1989), 281–288.

[10] WHITTED, T. An improved illumination model for shaded display. *Commun. ACM 23*, 6 (June 1980), 343–349.

[11] ZWILLINGER, D. *Handbook of Integration.* Jones and Bartlett Publishers, 1992.

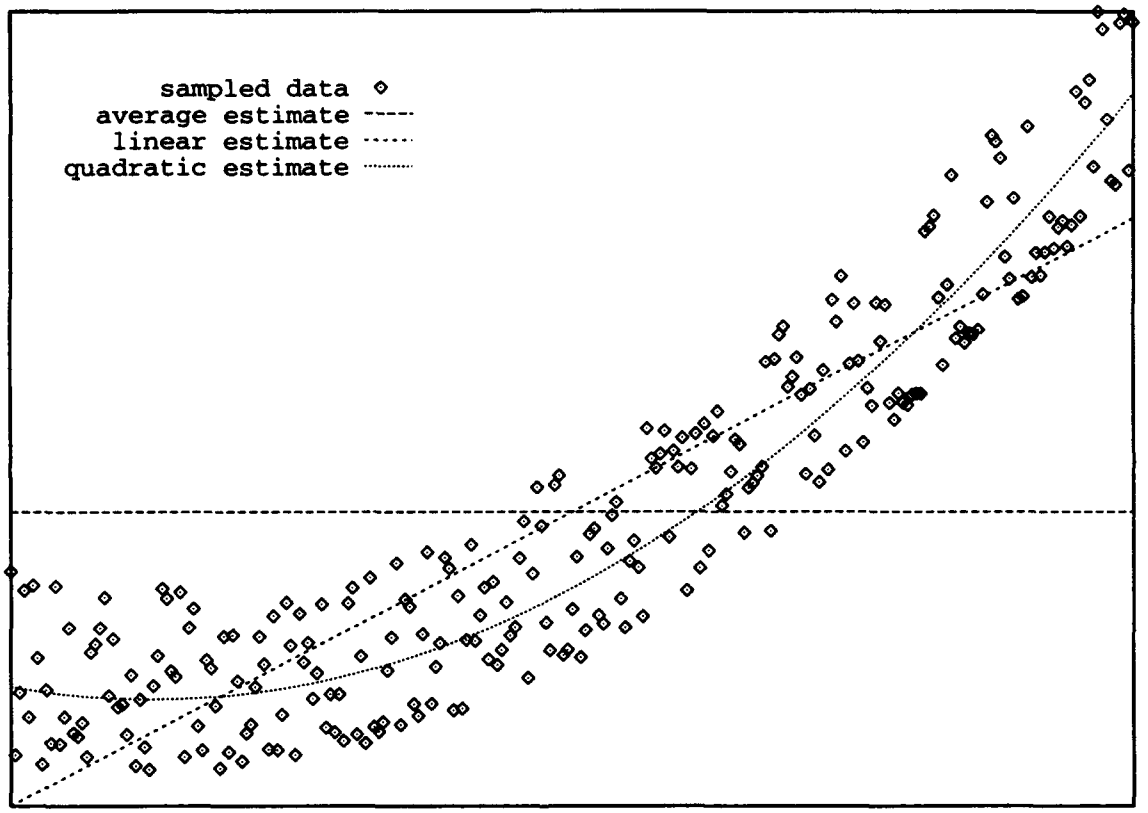


Figure 2: An example of polynomial regression: a set of sampled data and its average, linear and quadratic estimates.

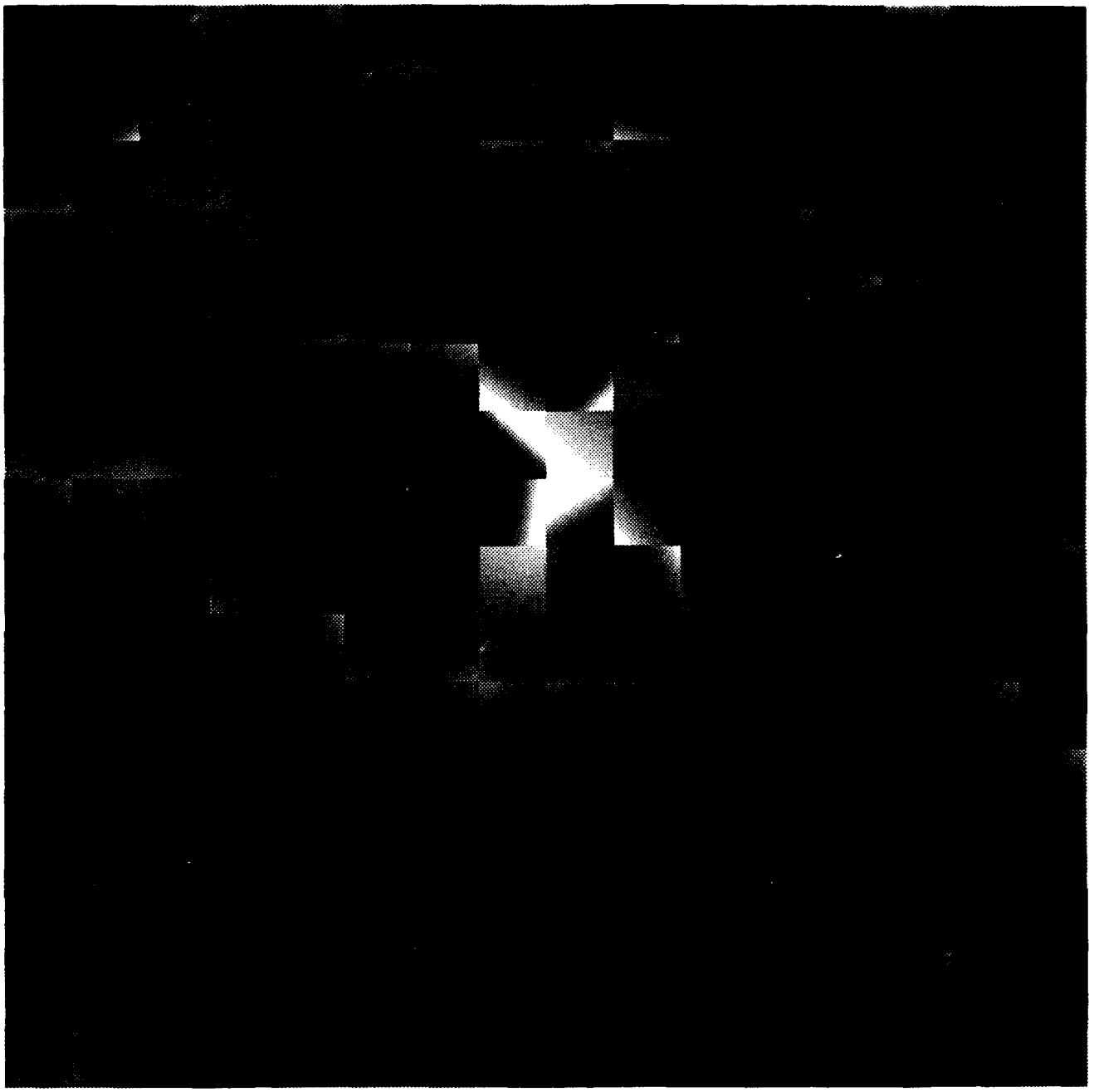

Figure 3: A test image, sampled at a target resolution of 16×16, and reconstructed at 512×512 pixels.

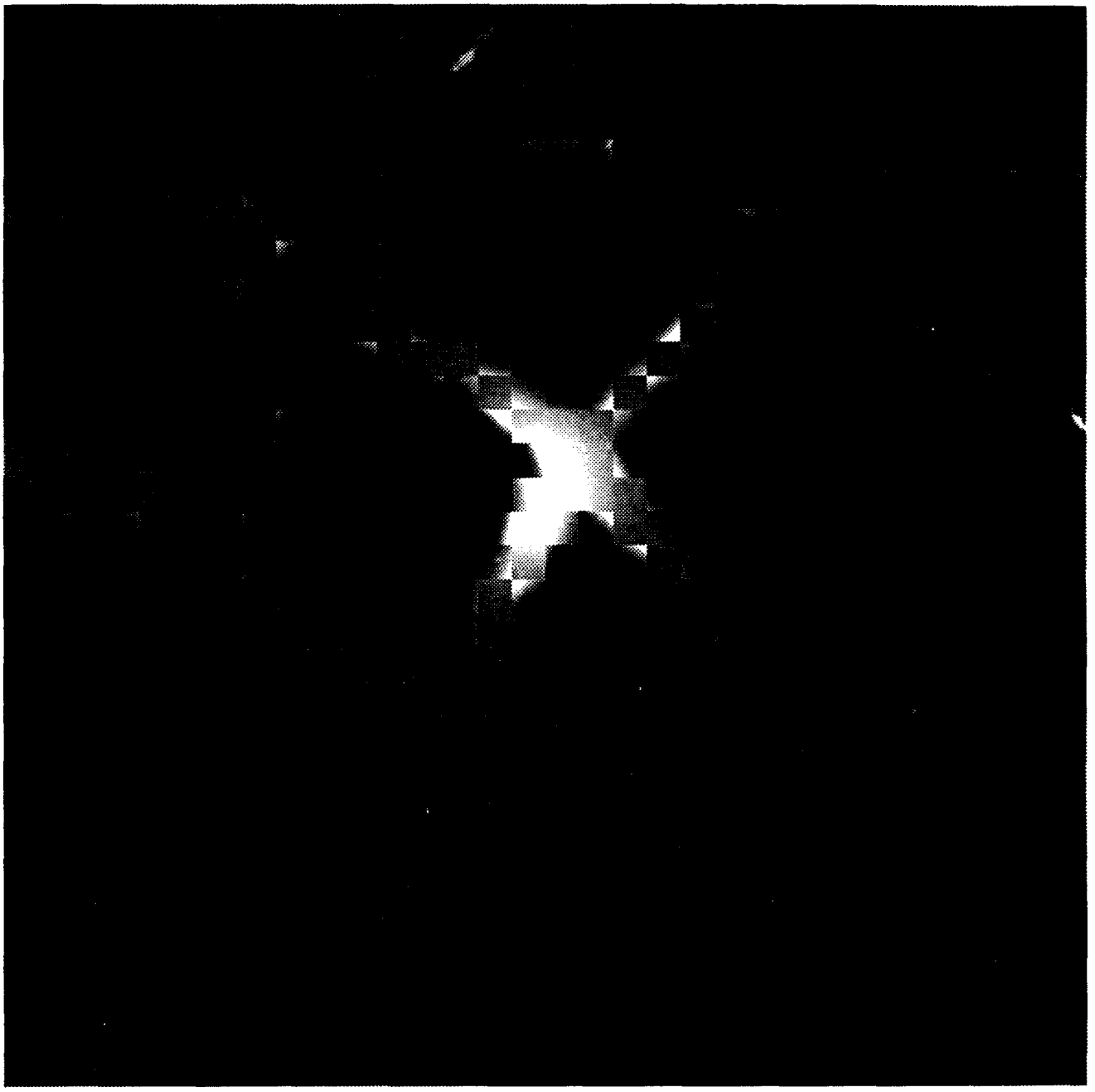

Figure 4: The test image sampled at a target resolution of 32×32.

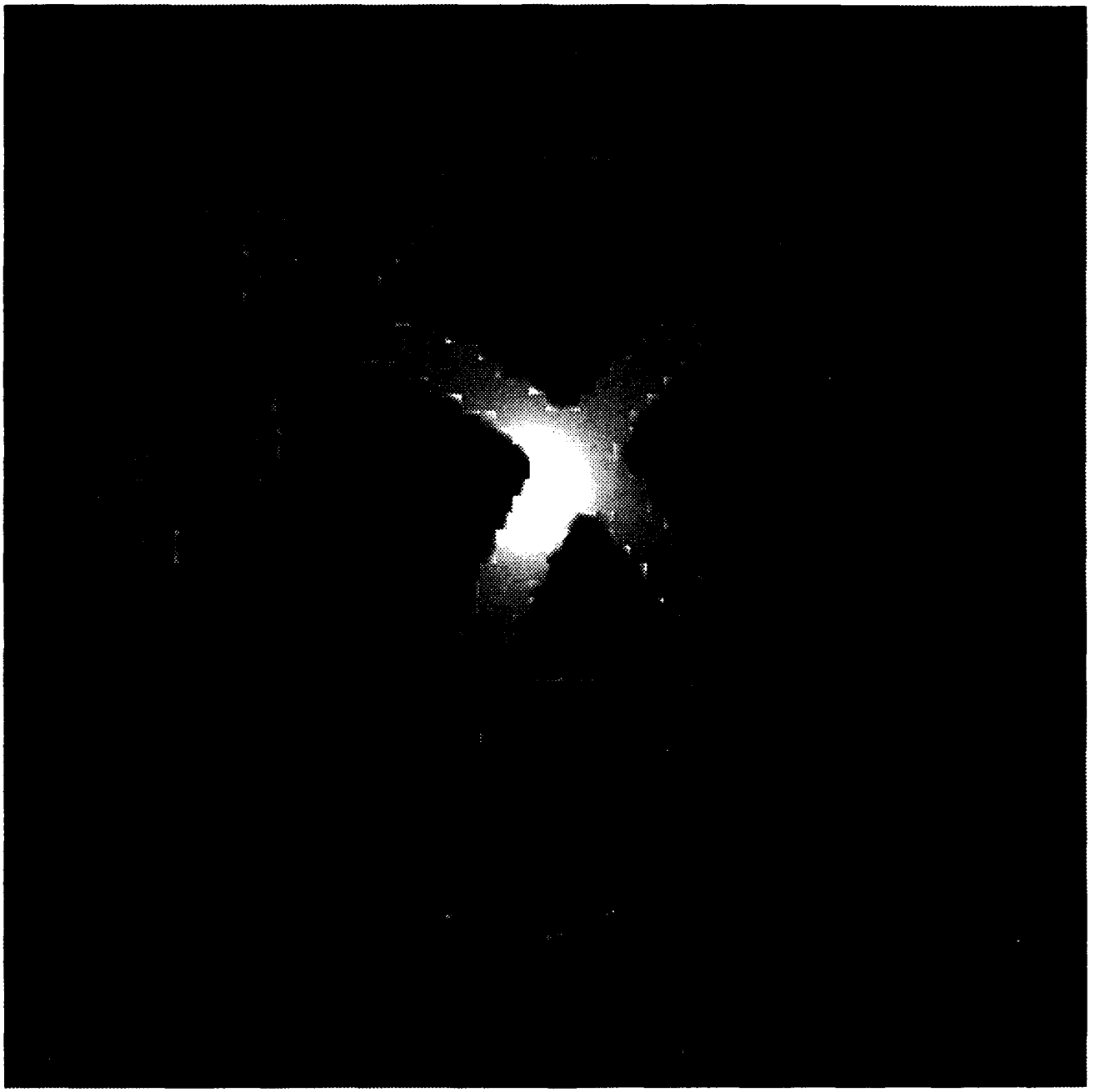

Figure 5: The test image sampled at a target resolution of 64×64.

Figure 6: The test image sampled at a target resolution of 128×128.

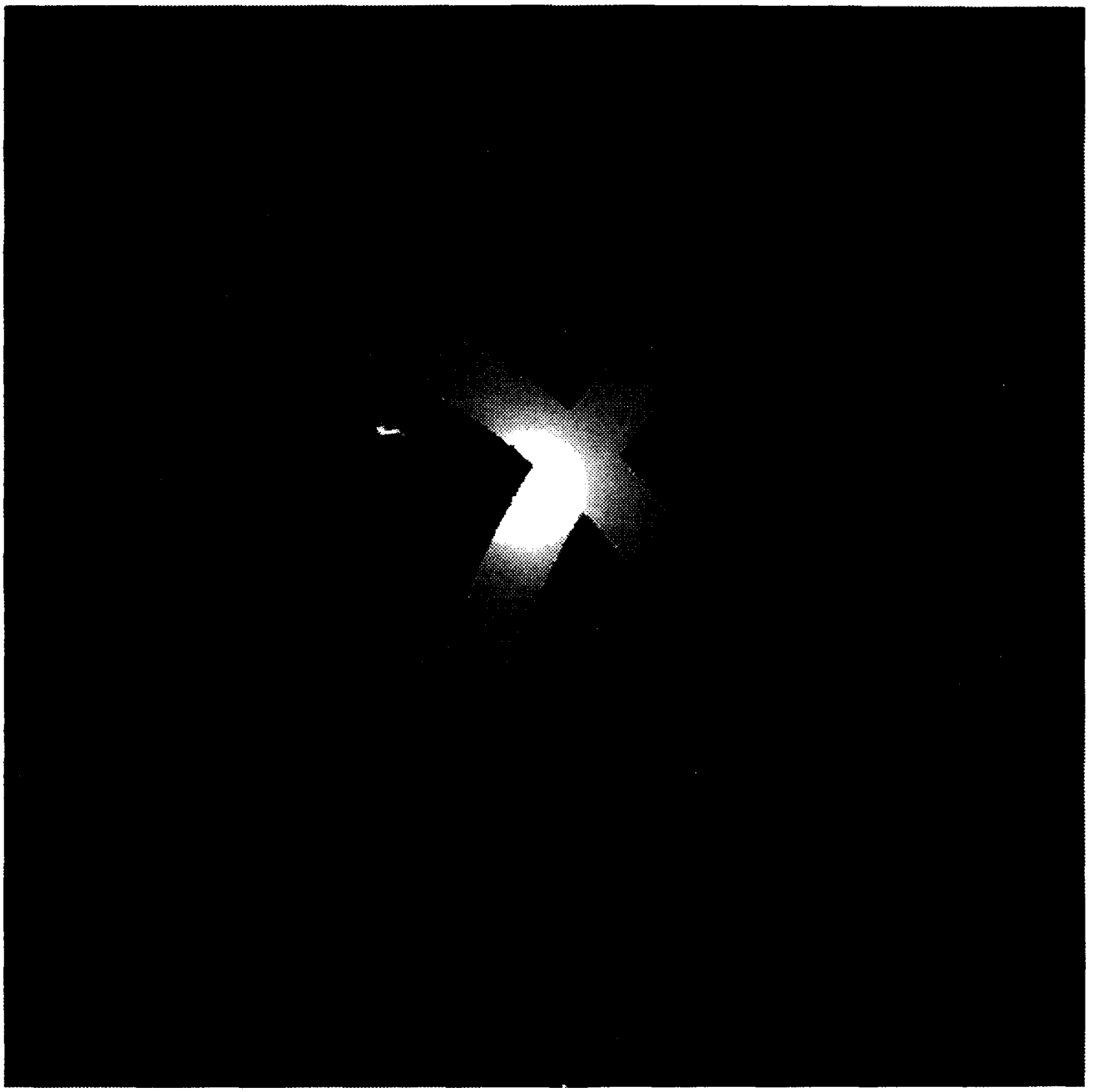

Figure 7: The test image sampled at a target resolution of 256×256.

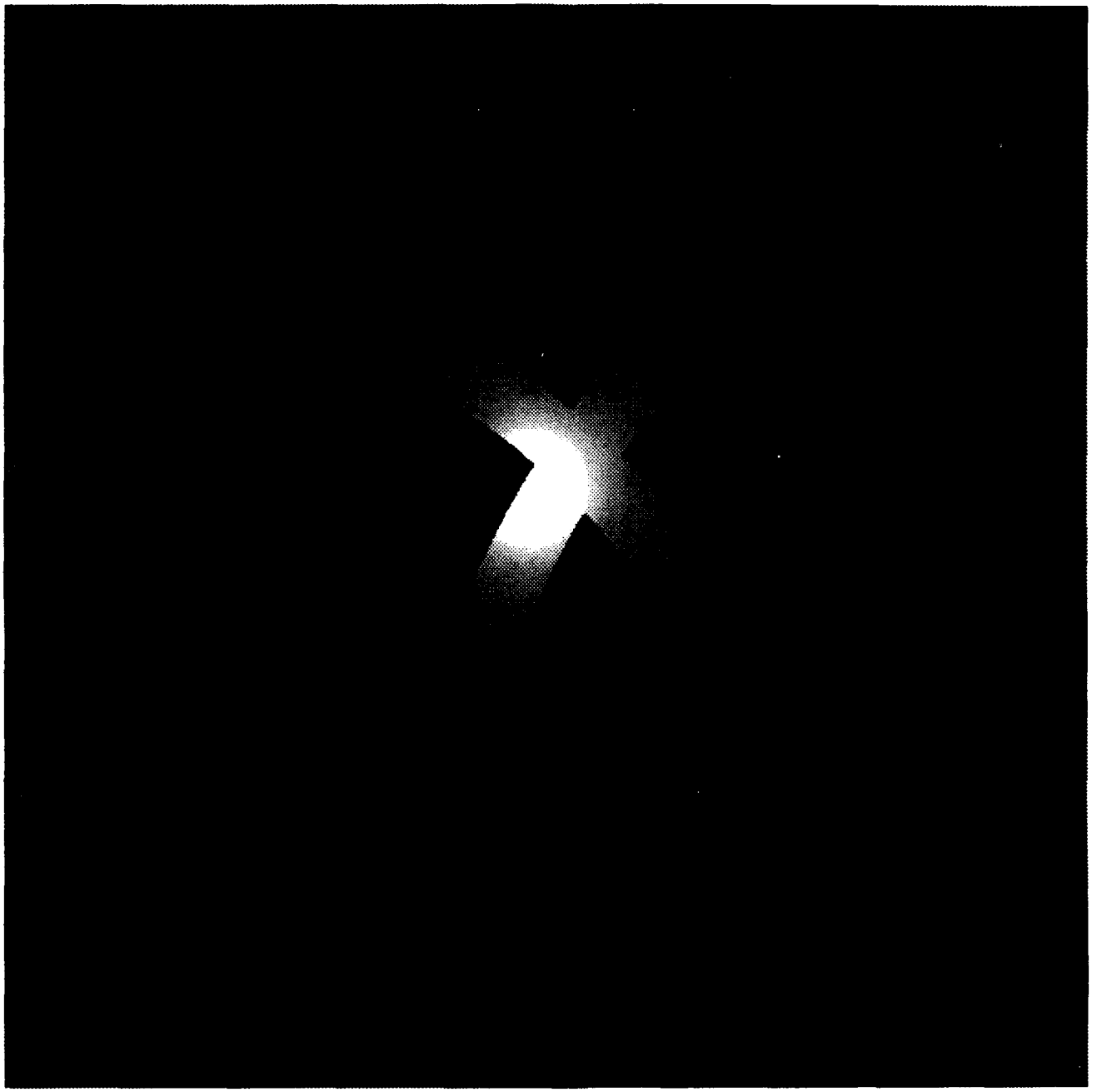

Figure 8: The test image sampled at a target resolution of 512×512.

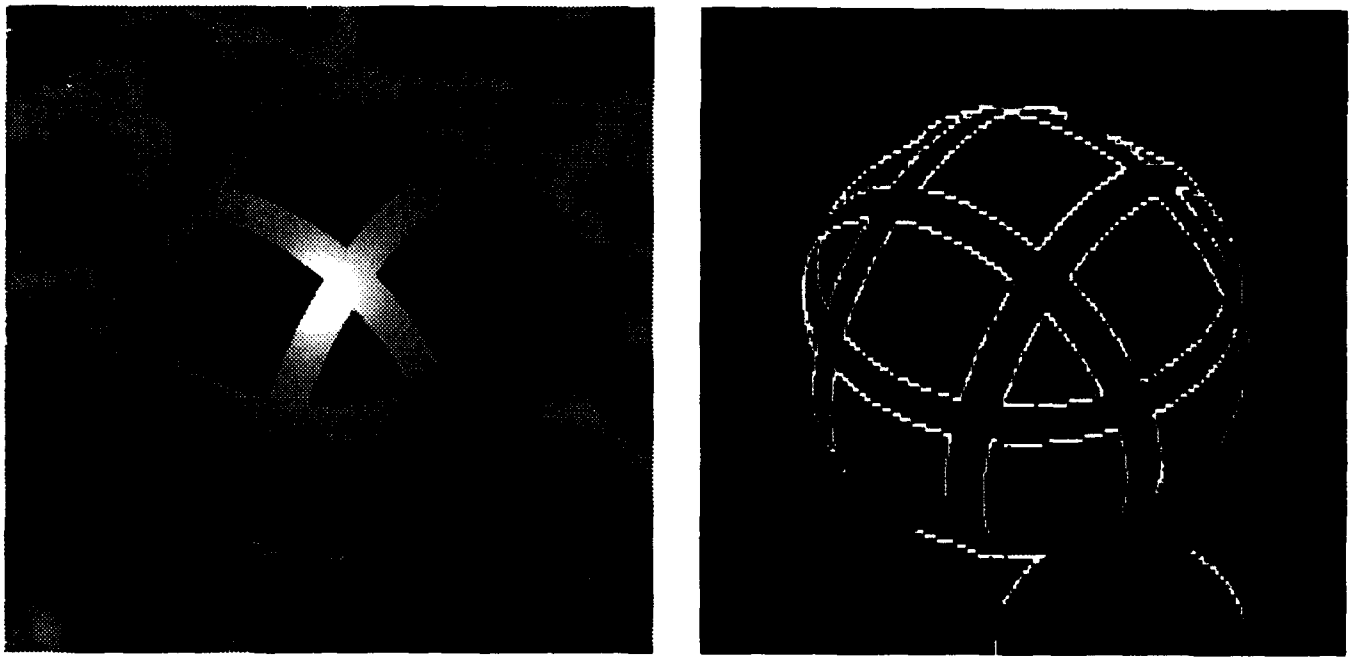

Figure 9: The 512×512 target is scaled to 256×256, and is shown along with an a gray scale image indicating the number of regions incident upon each pixel, where black is 1 region and white is 4 regions.

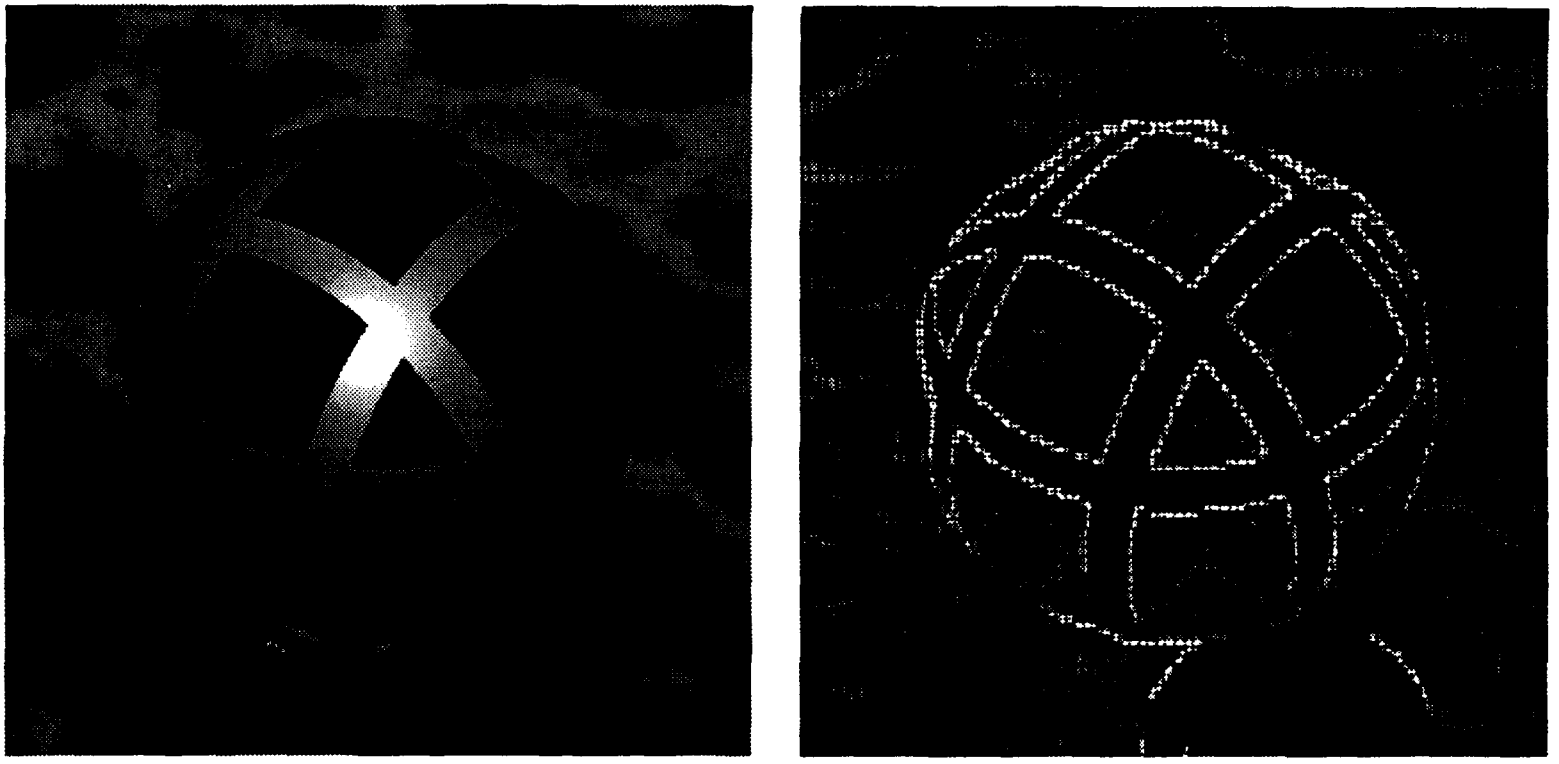

Figure 10: The 512×512 target is scaled to a pathological size (297×297), and is shown along with an a gray scale image indicating the number of regions incident upon each pixel, where black is 1 region and white is 8 regions.

Figure 11: A map is shown of the 512×512 image description. Black areas are described by average (constant) estimates; white areas are better described by linear estimates.

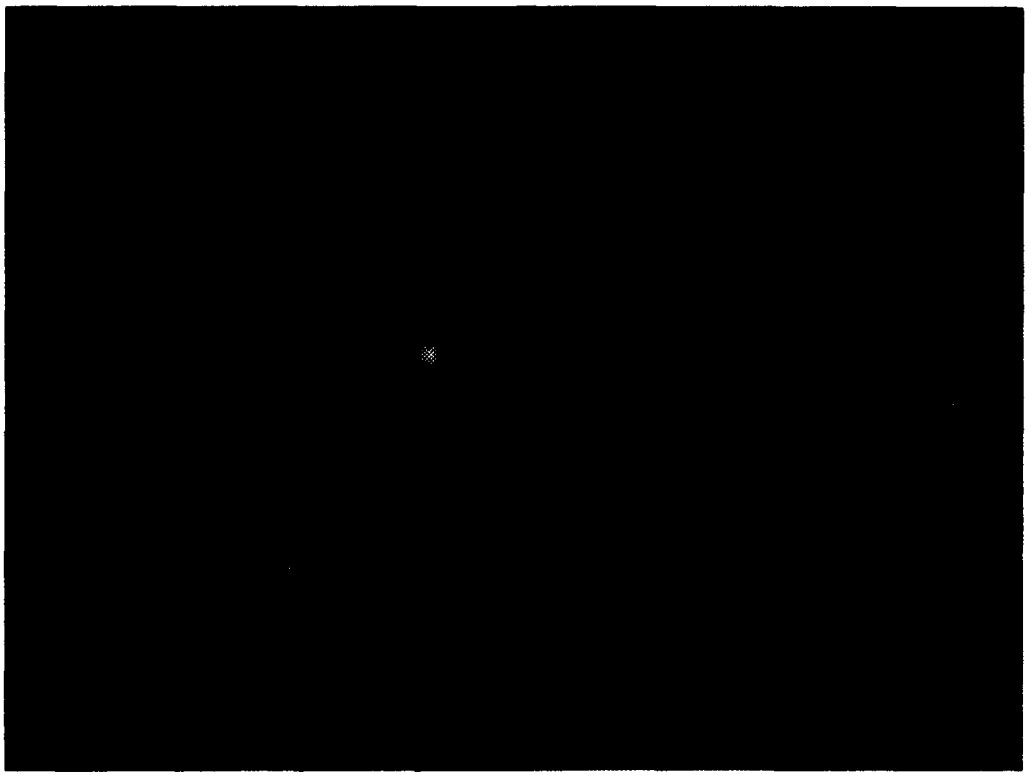

Figure 12: A different scene is sampled at 256×192, with runtime 1391 seconds, generating 33318 regions; it is reconstructed here at 400×300, with runtime of 29 seconds.

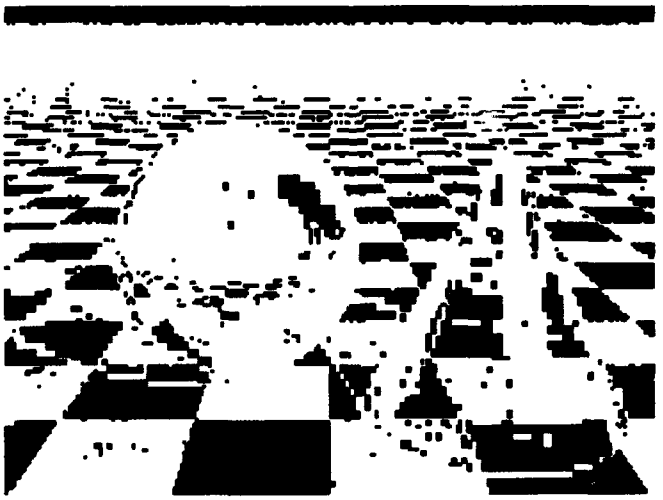

Figure 13: The estimate map derived from Figure 12 is shown. Black areas are adequately described by average estimates; white areas are better described by linear estimates.

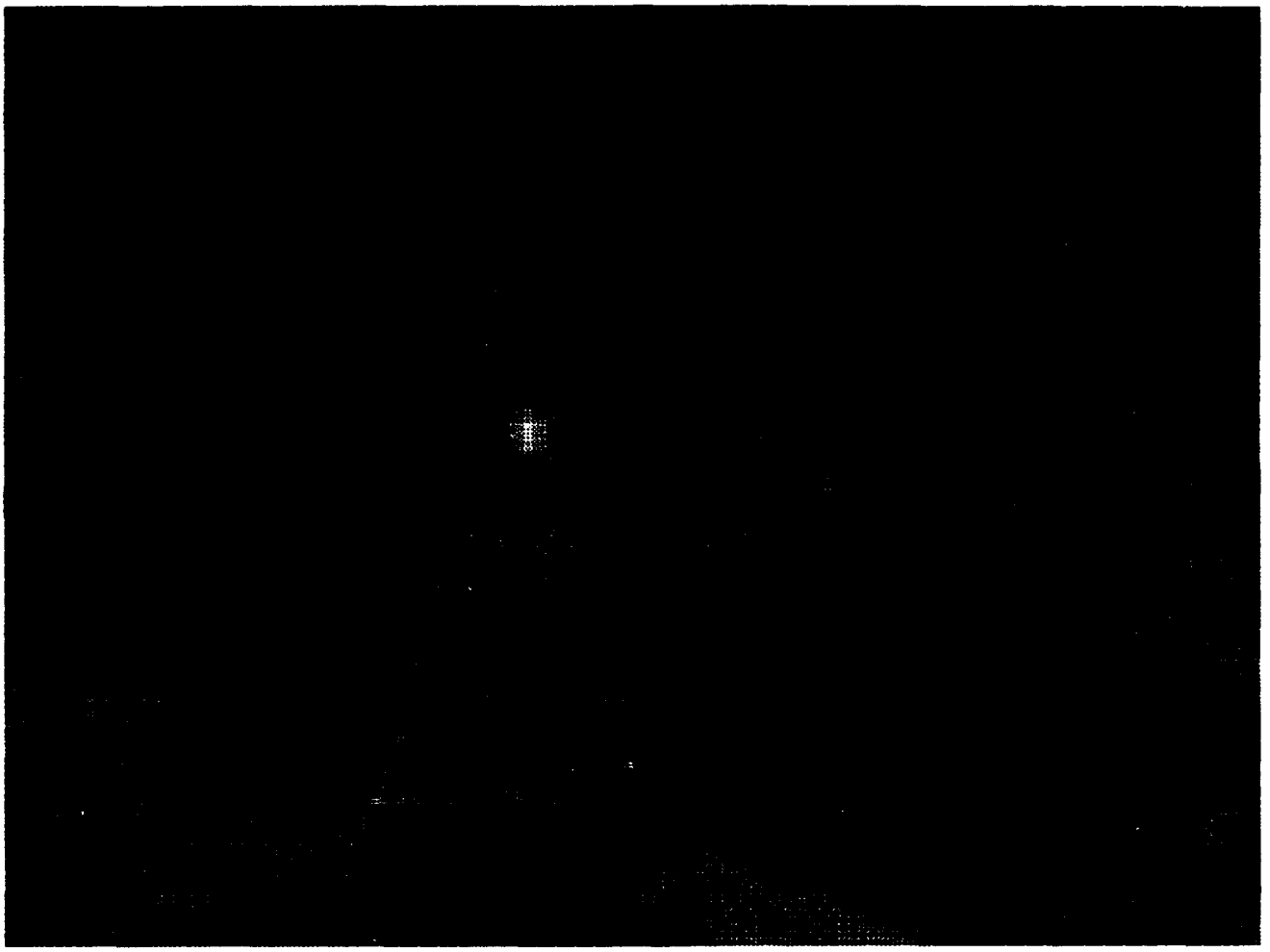

Figure 14: The k-D tree resulting from adaptive subdivision of Figure 12 is shown.