

Renormalized Rendering of Unresolved Objects

J Tessendorf
Cinesite Digital Studios

February 23, 1998

1 Introduction

The rendering of hair strands involves simultaneous shading, hiding, and compositing of multiple strands, each of which is not fully resolvable. Within each pixel, many hair strands can contribute to the total light level. In practical testing, each pixel a hair strand passes through has only a small fraction occupied, approximately 0.25%. This tiny fraction is important to achieve the soft, fine appearance of hair. Thick hairs that individually occupy most of a pixel will not convey softness or furry patches. Rendered hair is therefore a very subtle process, and individual strands are generally visible only weakly or with special lighting conditions.

In the natural world, the number of hairs present in animal fur on a body can approach millions per square inch. In terms of practical computational resources, the total number of hairs rendered on a body may not exceed a few million total. The transition between a practically-achieved sparse-hair distribution and a thick fur must be accomplished by some method other than a brute-force increase in the number of hairs on the surface.

This paper presents an algorithm for performing this extension. It is based on the set of equations used to composite-hide-buffer many hairs together, with an extrapolation of the render results based on some statistical properties of the hair. The validity of this approach depends on several assumptions concerning the statistical properties of the body of hair, each of which are stated and examined a little.

A key feature of hair strands that is exploited in the JAHASA renderer is that each hair is “thin”, i.e. its width in the image plane is less than a single pixel. The hiding of one hair strand by another is accomplished by a mixture of A Buffer and Z Buffer criteria, which has well-behaved and correct limit properties as the strands become thick or infinitesimally thin; and which has a statistically nice behavior if each hair is regarded as a random event partially filling a pixel. In fact, these algorithms are directly transferable to the problem of rendering large numbers of any type of objects that are not resolvable by the camera (e.g. particle systems).

2 Composite Rendering of Unresolved Objects

The JAHASA renderer implements a compositing-style hiding mechanism based on the assumption that the objects being rendered are small and/or there are so many of them that they may be considered statistically uniformly placed across the entire area of a pixel. Such an approach leads to the following mathematical statement of the rendering process:

The objects are numbered, but not ordered in any way. If object n intersects a particular pixel, the fraction of pixel area occupied by that object is $\delta\alpha_n$, and the amount of light it can contribute to the pixel is δi_n .

The pixel is characterized by red, green, and blue channels, and by several alpha channels which track the visibility of the object in from of other objects, etc. Included in the set of alpha channels is one which tracks the fraction of the pixel occupied by hair. After $n - 1$ hairs, the pixel has the rgb triplet i_{n-1} and hair alpha α_{n-1} . In adding the n -th hair, these quantities are altered in an interactive way:

$$\begin{aligned} i_n &= i_{n-1} (1 - \delta\alpha_n) + \delta i_n \delta\alpha_n \\ \alpha_n &= \alpha_{n-1} (1 - \delta\alpha_n) + \delta\alpha_n \end{aligned}$$

Even though these compositing equations are motivated by arguments about the size of the objects and their random behavior, these equations actually work well in the limit that the objects *are* resolvable, when $\delta\alpha_n = 1$. Additional tests are needed to only insure that an object is composited only when it is visible (e.g. a z-buffer check).

In the case of hair rendering, many strands (on the order of millions) are rendered, with quasi-random placement and properties. If we treat the quantities coming into rendering as random fluctuations of a statistical process, and assume that enough hairs are rendered to generate an ensemble of fluctuations, then we can determine the actual statistical properties found in the rendering data. Two quantities of particular interest are the $\delta\alpha$ for each hair in each pixel, and the number N of hairs actually intersecting each pixel.

In a typical rendering situation, a histogram of $\delta\alpha$ in figure 1 reveals that it is approximately gaussian, with a mean of 0.0027 (i.e. on average 0.27% of a pixel is occupied by an individual hair strand), and a standard deviation of about 0.0006.

Similarly, a histogram of N shows that it is nearly a poisson or log-normal random variable, with a mean of 23 and a standard deviation of 9.96 (figure 2). It is interesting to note that other approaches¹ enforce a poisson statistical distribution in their hair/fur rendering algorithms. Goldman's work uses the statistically averaged properties of tiny objects that are rendered. In JAHASA, the poisson distribution and other statistical properties are direct consequences of computed geometrical properties. We can think of the algorithms in JAHASA

¹Dan B. Goldman, "Fake Fur Rendering", *SIGGRAPH Computer Graphics Proceedings, Annual Conference Series*, 1997, pg 127

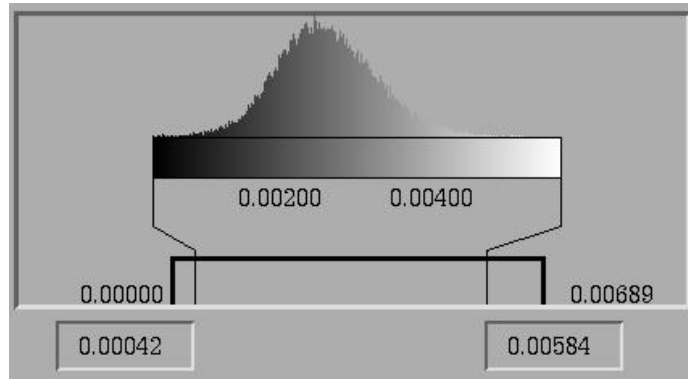


Figure 1: Histogram of $\delta\alpha$ generated by JAHASA.

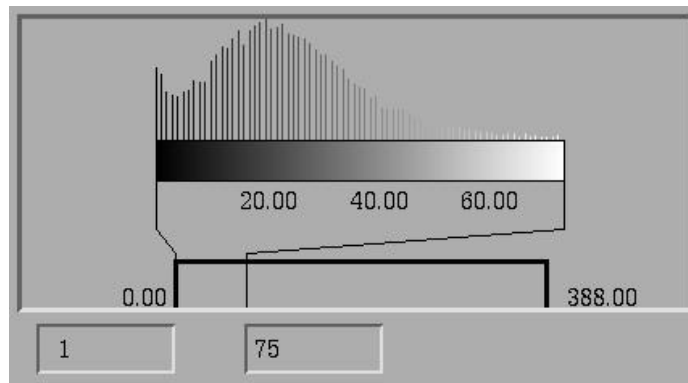


Figure 2: Histogram of N generated by JAHASA.

as more fundamental than Goldman's Fake Fur method, because Goldman's method is derivable as a statistically-averaged limit.

In the next section this compositing formula is combined with additional statistical analysis to arrive at a "renormalization" algorithm. The purpose of the algorithm is to accelerate the rendering of millions of small unresolved objects. In a statistical sense, there is little difference between rendering a few million tiny objects, and many million tiny objects. This assumes of course that all of the tiny objects are members of a statistical ensemble, and that the few million that have been rendered are a representative sample of that ensemble. Making this leap of assumption, the compositing formula above is used to extrapolate the few million rendered objects to any larger number.

In order to accomplish this renormalization extension, it is useful to rewrite the implicit finite-difference equations as an explicit solution. The solution of the compositing equations, after n objects have been collected in the pixel, is

$$i_n = \sum_{i=1}^n \delta i_i \delta \alpha_i \prod_{l=i+1}^n (1 - \delta \alpha_l) \quad (1)$$

$$\alpha_n = \sum_{i=1}^n \delta \alpha_i \prod_{l=i+1}^n (1 - \delta \alpha_l) \quad (2)$$

In this form, the solution is mathematically explicit, and allows us to rephrase the renormalization problem as follows: Given that we compute i_n and α_n for some sufficiently large n , can we estimate i_m and α_m for $m \gg n$, and explicitly as

$$i_m \approx i_n T_{nm} \quad (3)$$

Part of the problem is to determine the "transfer function" T_{nm} . This expression gives the solution its renormalization character.

3 Statistically Extrapolating a Few Objects to Many

In order to perform an extrapolation and renormalization, knowledge or assumptions about the statistical properties of the tiny objects must be used. In this process, we will assume properties that are the simplest to exploit:

1. The set of $\delta \alpha_i$ form a complete statistical ensemble.
2. The set of $\delta \alpha_i$ are statistically independent, i.e.

$$\langle \delta \alpha_i \delta \alpha_j \rangle = \langle \delta \alpha_i \rangle \langle \delta \alpha_j \rangle . \quad (4)$$

3. The set of $\delta \alpha_i$ are statistically homogeneous, i.e. $\langle \delta \alpha_i \rangle \equiv \delta \bar{\alpha}$ is a constant.

This set of assumptions is sufficient to arrive at a transfer function. Notice that we have *not* made any assumptions about the probability distribution that the pixel fractions $\delta\alpha_i$ satisfy, apart from homogeneity. In fact, these assumptions are testable within JAHASA.

With these few statistical assumptions, we can compute the average of α_n as

$$\langle\alpha_n\rangle = 1 - (1 - \delta\bar{\alpha})^n \quad (5)$$

This expression is arrived at by averaging the explicit equation 2, then evaluating the finite summations explicitly.

Now suppose a set of n values of $\delta\alpha_k$ and δi_k have been generated for a pixel, and that this has been done for a sufficiently large number of pixels that to entire set of $\delta\alpha_k$ over the many pixels constitutes a good approximation to an ensemble. In general, going beyond n to a value $m \gg n$ requires the set of additional intensity values δi_k ($k = n + 1, \dots, m$). However, if n is reasonably large, we can make the approximation that the δi_k $k \leq n$ are reasonably representative of the values for $k > n$. In the extrapolation then, we will make the assertion

$$\delta i_k = \delta i_{k \bmod(n)} . \quad (6)$$

With this assertion, the averaged intensity follows from explicit summation as

$$\langle i_m \rangle = \langle i_n \rangle \frac{\langle \alpha_m \rangle}{\langle \alpha_n \rangle} . \quad (7)$$

From which we make the identification

$$T_{nm} = \frac{\langle \alpha_m \rangle}{\langle \alpha_n \rangle} . \quad (8)$$

4 Updated Rendering Algorithm

The rendering algorithm in JAHASA has been modified in two ways to support renormalization. The first modification is the addition of two diagnostic channels that compute $\delta\bar{\alpha}$ and track the number of hairs n that wind up in each pixel. The second modification is an additional step at the end of rendering to compute the transfer function and apply it to each pixel. The transfer function is applied to the image channels and to the alpha channels. The critical user input to this renormalization operation is a factor called `RenormalizationFactor`. This parameter is used to increase the effective number of hairs in a pixel from n to $n \times \text{RenormalizationFactor}$

As an example of the practical achievement of the renormalization process, figure 3 shows hair strands rendered on a NURBS surface. In this render, 120,000 hairs were generated and no renormalization was applied. The hair appears to have gaps and thin areas. There is also an apparent “backlighting” effect on the edges of the surface, stemming from the limited number of hairs each pixel on the interior sees. Interior pixels hold 30 - 70 hairs each, whereas on the edge of the surface, the number of hairs is 200 - 470.



Figure 3: Hairy surface image generated by JAHASA. A total of 120,000 hairs were generated for the surface. No renormalization has been applied.



Figure 4: Hairy surface image generated by JAHASA. A total of 120,000 hairs were generated for the surface. Renormalization applied with a RenormalizationFactor = 15

In contrast, figure 4 shows the same identical surface and hairs, but after renormalization has been applied. The hair appears much denser, while the edges of the surface are still antialiased. There is no “backlighting” also, because the renormalization brings up the intensity of interior pixels relative to edge pixels.