EFFICIENT SIMULATION OF HALOS FOR COMPUTER GRAPHICS

JEAN-CHRISTOPHE GONZATO, SYLVAIN MARCHAND

LaBRI, Université Bordeaux 1, 351 cours de la Libération, F-33405 Talence cedex, France

ABSTRACT

We present a technique for efficiently generating photo-realistic pictures of halos. First, we describe an algorithm for producing images of halos from physical parameters supplied by the user. The problem with the resulting images is that they are sampled in a non-uniform way. Then, we propose a specific algorithm for reconstructing the uniform version of these images from their non-uniform sampling. With this technique, we obtain computer-generated halos looking as if they had been natural scenes captured by a camera.

Keywords: halo, irregular sampling, natural phenomena, signal reconstruction

INTRODUCTION

In this paper, we aim at generating photo-realistic images of halos. Natural halos usually appear on sunny winter days when there is enough moisture in the air. The sun light path is then deviated by ice crystals falling to the ground. Simulating halos can provide us with a very convenient way to enhance existing photographs with nice but scarce light phenomena. More than only describing the crystals, simulating the halos, and integrating the resulting simulations into real photographs, the main problem is to find a way to compute the value of light passing through the crystals for the whole view frustum. Indeed, the value of the computer-generated halo images is undefined for many pixels. The problem addressed here is how to recover as accurately as possible the missing data.

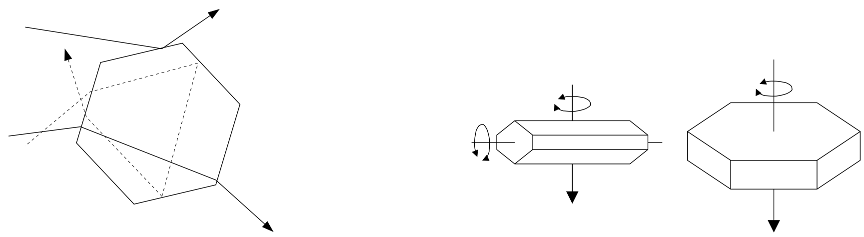
In the *Natural Phenomenon* section, we briefly introduce the physics of halos and the way they are usually modeled by physicists. The *Simulating Halos* section presents the main models for simulating halos using a computer. They all produce images in a non-uniform way: the value of the halo is not know for the whole image. So, in the *Complete Halo Image Reconstruction* key section, we present an efficient algorithm to reconstruct irregularly-sampled signals (sounds, images...). This algorithm proposes to approximate the original signal step by step, each iteration decreasing the reconstruction error.

Fig. 1: *Photograph of a natural 22-degree halo.*

NATURAL PHENOMENON

The well-known rainbows are created by the deviation of the sun rays by droplets when the sun shines whereas the rain falls elsewhere. During winter days, droplets are transformed into small ice crystals which can be regarded as regular hexagons forming a three-dimensional cloud of particles, slowly falling to the earth. The sun rays are also deviated – by series of reflections and refractions – when they travel through the crystals, thus giving birth to solar halos. Generally, one can see a – either partial or complete – lighted circle located at 22 degrees away from the sun (see Fig. 1). Many other kinds of halo exist, although they are more unlikely to occur in practice. The kinds of halos which can be observed depend not only on the kind of crystals within the cloud and the turbulence of the atmosphere, but also on the location of both the light source – usually the sun – and the observer.

Crystals are usually modeled by regular hexagons. When the light source hits the crystals, several interactions occurs, depending on whether or not the sun ray is directly reflected at the surface or refracted into the volume. Crystals with length-to-radius ratios above or below 2 are called pencils or plates, respectively. In a non-turbulent atmosphere, the ice crystals fall in such as way that the surface friction is maximal (see Fig. 2).



(a) Light-crystal interactions (b) Pencil and plate falling

Fig. 2: *Classic models of ice crystals.*

SIMULATING HALOS

Simulation algorithms allow physicists to verify the consistency of their theories. They also produce realistic pictures of great interest for people involved in computer graphics. Perhaps surprisingly, very few efficient simulation techniques have been proposed so far. Greenler (1980) provided the physical foundations of almost all halo simulation algorithms. He proposed to consider the case of a ray of sun being deviated by one single ice crystal during its travel through the iced cloud. The problem was now to compute the quantity of light passing through each pixel of the image representing the simulated halo. From the complexity point of view, using a standard ray tracing algorithm is not realistic. Indeed, for each ray leaving the eye and passing through the pixel, the algorithm has to test all the orientations in order to find the ones which deviate the ray into the direction of the light source. Greenler proposes to use an inverse ray-tracing algorithm instead. Knowing the direction of the ray of light leaving the crystal is equivalent to knowing where to look in the sky for light coming to your eye from the crystal with that particular orientation. Thus the presented algorithm follows 4 steps: In step 1, the algorithm chooses one crystal – knowing the distribution of the various types in the cloud of crystals – with a random orientation in the range of possible orientations. Step 2 consists in casting a ray from the light source (the sun for instance) to the crystal and compute the deviation. In Step 3, place the crystal in the atmosphere in order to see the light leaving the crystal. This positioning is simply done by plotting a light point on a fish-eye view image of the atmosphere. Finally, repeat the previous steps for an user-predefined number of crystals and orientations.

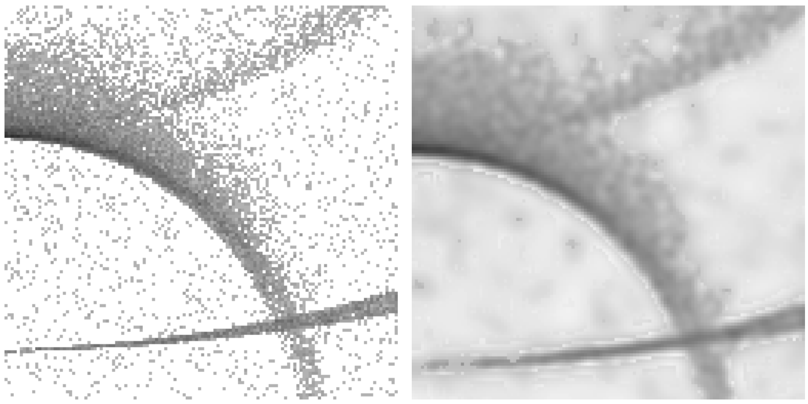
In 1996, Glassner (1996a;b) enhanced the Greenler method and proposed the first halo simulation algorithm for computer graphics. Moreover Glassner proposes to generate colored halos by splitting the visible spectrum into few wavelengths and by casting rays from the light source for each wavelength.

We present an efficient implementation of this algorithm, where we take advantage of the classic Fresnel and Snell-Descartes laws to compute the real intensity of light coming from the light source.

Unfortunately, as for the Glassner algorithm, the value of this intensity is not known for the whole image. No matter the – possibly huge – number of rays to cast from the light source to simulate the halo phenomena as accurately as possible, the intensity of the light coming from the halo cannot be known for the whole image.

COMPLETE HALO IMAGE RECONSTRUCTION

The algorithm for halo simulation described above produces images sampled in a non-uniform way (see Fig. 3(a)), each dot representing the intensity pattern of the halo. The problem is then, from this irregular sampling, to recover the missing information in order to reconstruct the whole images (see Fig. 3(b)) from their samples taken in an irregular way.



(a) Irregular Sampling (b) Uniform Reconstruction

Fig. 3: *Original (a) and reconstruction (b) of a portion of a halo image.*

Glassner (1996a) proposes to fill the missing information with black pixels and to use then a superposition of the same image smoothed by a series of Gaussian blurs at different scales. This technique is tricky, rather empirical though. Its results are satisfactory because blurring and sub-sampling are in fact the basic operations for uniform reconstruction, provided that the blurring is done using a (low-pass) reconstruction filter close to the one given by the theory of signal sampling and reconstruction. The problem experienced by Glassner is that small blurs do not get the dots to join up and form a smooth field, whereas large blurs make the whole picture go fuzzy. That is quite normal, since Glassner basically tries to apply an uniform reconstruction technique for non-uniform sampling.

IRREGULAR SAMPLING AND RECONSTRUCTION

We propose to refer to the irregular sampling theory in order to design an original and more efficient algorithm for halo image reconstruction. Let us first briefly explain the theory for one-dimensional signals (such as sounds), then we will easily generalize it to two-dimensional signals (such as images).

Irregular Sampling Theory

Let s be a real-valued one-dimensional signal, band-limited in frequency. This means that s has spectrum in some interval $[-\Omega_s, +\Omega_s]$, which is the case iff all the coefficients of its Fourier transform S corresponding to parts of the frequency domain outside this interval are zero. More formally:

$$\exists \Omega_s > 0,\ \mathrm{support}(S) \subseteq [-\Omega_s, +\Omega_s] \quad \text{where } S(\Omega) = \int_{-\infty}^{+\infty} s(t)\, e^{-j\Omega t}\, dt \tag{1}$$

It is possible to reconstruct the original signal s from samples taken in a non-uniform (irregular) way if the maximal distance between two consecutive sampling times does not exceed the Nyquist period $T_s = \pi/\Omega_s$.

Reconstruction Algorithm

Feichtinger *et al.* (1991) explain that most irregular reconstruction algorithms are iterative in nature. Starting from some initial guess, typically based on the given sampling values, further approximations of s are obtained step by step, using the available (assumed) knowledge about Ω_s.

This is the case of the Allebach algorithm, which is made of 3 steps. Step 1 consists of the interpolation of the sampling values. Many interpolation techniques can be used, such as the Voronoi interpolation: nearest neighborhood interpolation, using the arithmetic mean for equally-spaced neighbors. The interpolated signal contains many high frequencies outside of $[-\Omega_s, +\Omega_s]$. The information concerning Ω_s can be used next. In step 2 the interpolated signal is low-pass filtered with a cutoff frequency slightly greater than Ω_s. Let s_1 denote the first signal resulting from steps 1 and 2, then look at the difference signal $s - s_1$. According to the construction, s_1 has its spectrum within the same range $[-\Omega_s, +\Omega_s]$, and for obvious reasons we know its coordinates at the given sampling positions. Therefore, the estimate indicated above can be applied. Step 3 is the recursive reconstruction of the error – if significant – so that we can again recover a certain portion of the remaining signal by repeating the first two steps, now starting with the sampled coordinates of $s - s_1$. Continuing to use the difference between the given sampling values of s and those of the n-th approximation we generate additive corrections which lead stepwise to improved approximations.

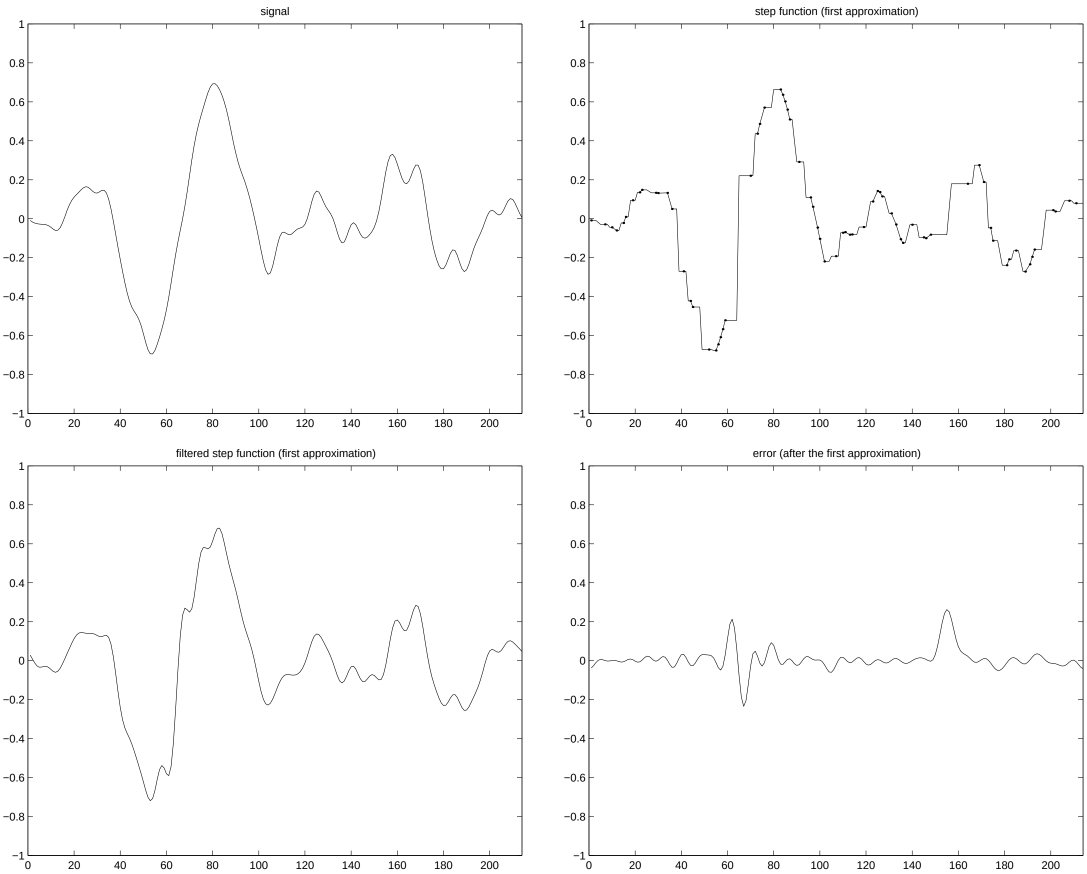


Fig. 4: *The first iteration of the Voronoi-Allebach algorithm. The Voronoi interpolation produces a step function which is low-pass filtered, then the reconstruction error is measured (for known sampling points) and reconstructed, recursively. The reconstruction error converges to 0 if the Nyquist criterion was respected during the sampling stage.*

Fig. 4 illustrates the Voronoi-Allebach algorithm, that is the Allebach algorithm with Voronoi interpolation as step 1. Given the sampling values at known positions we form a step function which is constant from midpoint to midpoint of the given sampling sequence. Because of the assumed smoothness of the signal, this step function will not be too far away from the original signal, in the mean-squared sense. The filtering process in step 2 destroys the point-wise interpolation property of the Voronoi method, but theoretical considerations show that if the maximal gap is smaller than the Nyquist period then the iterative scheme will converge to the original signal s at a geometric rate (the faster the smaller the actual maximal gap is compared to the Nyquist rate).

RECONSTRUCTION OF HALO IMAGES

Strohmer (1993; 1997) has studied the reconstruction of images from irregular sampling. Most reconstruction algorithms fail in the case of halo images. The main problem with pictures coming from halo simulations is that they show areas with accumulations of samples together with close-to-empty areas. Since the classic reconstruction methods need a Nyquist period – proportional to the distance between two sample points – constant all over the picture, they are bound to fail in the case of halo pictures. In fact, choosing a too small Nyquist period does not get the dots to join up and form a smooth field, whereas a too large value makes the picture go fuzzy.

Algorithm Overview

To address this problem, we generalize the Allebach reconstruction algorithm for pictures issued from halo simulations. We consider an iterative reconstruction algorithm: Starting from the given sampling values, further approximations of the complete picture are obtained step by step.

First, the unknown pixels are interpolated from the known sampling values. Again different interpolation methods can be used. We use the Marvasti method, where all the unknown values are simply set equal to zero. The iterative scheme of the Allebach algorithm would have converged faster with the Voronoi interpolation but the computation of Voronoi diagrams was too slow to suit our needs. Second, the interpolated picture is low-pass filtered. This time we perform a local non-uniform reconstruction by adaptively choosing for every pixel the best size for the reconstruction filter (see below). Finally, we consider the difference between the reconstructed picture and the original one for the known sampling values. If this error is significant then we iteratively repeat the first two steps. Continuing to use the difference between the given sampling values and those of the n-th approximation we generate additive corrections which lead stepwise to improved approximations.

Reconstruction Filter

The interpolation technique being chosen, let us now focus on the filtering operation of the Allebach algorithm. It can be performed by a simple convolution, provided that the impulse response of the filter is known. The problem is that the theoretical (ideal) reconstruction filter is a box in the frequency domain, corresponding to a sinc function in the time domain, and more precisely to $\mathrm{sinc}(t/T_s)$ where:

$$\mathrm{sinc}(t) = \frac{\sin(\pi t)}{\pi t} \quad \text{(and } \mathrm{sinc}(0) = 1\text{)} \tag{2}$$

Unfortunately this function has an infinite support. For practical use, it has to be truncated. To avoid aliasing phenomena, one may tapper the truncated version of the sinc function by multiplying it with a bell-shaped window, such as the Hann window (well-known in signal processing):

$$w_N(t) = \frac{1}{2}\left(1 - \cos\left(\frac{2\pi t}{N-1} + \pi\right)\right) \tag{3}$$

for $|t| \leq N/2$, where N is the size of the window (here in pixels). We then store the impulse response of the reconstruction filter in an odd-length square matrix defined by:

$$M_N(x,y) = w_N(t(x,y))\ \mathrm{sinc}(t(x,y)/T_s) \quad \text{where } |x| \leq N/2,\ |y| \leq N/2 \text{ and } t(x,y) = \sqrt{x^2+y^2} \tag{4}$$

Adaptive Filter Size

The problem with the images resulting from the halo simulation algorithm is that the Nyquist criterion is not respected: The maximal distance between two samples is much greater than the Nyquist period. Applying a non-uniform reconstruction scheme on the whole image would lead to the same problems as the ones experienced by Glassner. We propose instead to perform a local non-uniform reconstruction by adaptively choosing the appropriate neighborhood – and thus Nyquist period and reconstruction filter size – for every pixel. For that purpose, for each pixel we make the square area centered at this pixel grow until it contains a sufficient number of samples. The relation between the square neighborhood size N and the Nyquist period T_s is given by the following equation (k being an arbitrary constant, although values around 1 give satisfactory results):

$$N = 2kT_s + 1 \tag{5}$$

Then we consider $\Omega_s = 2\pi/T_s$ and we use the reconstruction filter of size $N \times N$. For the purposes of efficiency, these sizes are pre-computed once and stored prior to the reconstruction algorithm itself, together with the associated filters of various sizes.

CONCLUSION

In this article, we propose a combination of two algorithms. The first one, issued from the physics, allows us to simulate various kinds of halo phenomena. However, it generates an incomplete – irregularly sampled – image. The second algorithm, issued from signal theory, allows us to completely reconstruct the halo image from the result of the previous algorithm. The results are not only physically valid, but also visually realistic. For the mountain photograph (Fig. 5), we cast 100,000 rays to simulate the halo. The insertion in the photograph takes about 30 seconds (18 s for the simulation itself and 16 s for the image reconstruction) on a PC with a Pentium III processor at 550 MHz.

Fig. 5: *Real mountains together with a virtual halo.*

REFERENCES

Feichtinger HG, Cenker C, Steier H (1991). Fast Iterative and Non-Iterative Reconstruction Methods in Irregular Sampling. In Proceedings IEEE ICASSP, 1773–1776.

Feichtinger HG, Gröchenig K, Strohmer T (1995). Efficient Numerical Methods in Non-Uniform Sampling Theory. Numerische Mathematik, 69:423–440.

Feichtinger HG, Strohmer T (1993). Fast Iterative Reconstruction of Band-limited Images from Non-Uniform Sampling Values. In Proceedings of the Computer Analysis of Images and Patterns (CAIP) Conference, 82–91.

Glassner A (1996a). Computer-generated solar halos and sun dogs. IEEE Computer Graphics and Applications, 77–81.

Glassner A (1996b). Solar halos and sun dogs. IEEE Computer Graphics and Applications, 83–87.

Greenler, R (1980). Rainbows, Halos, and Glories. Cambridge University Press.

Jackèl D, Walter B (1998). Simulation and visualization of halos. In ANIGRAPH.

Lynch DK, Livingston W (1985). Color and Light in Nature. Cambridge University Press.

Strohmer T (1993). Efficient Methods for Digital Signal and Image Reconstruction from Nonuniform Samples. PhD thesis, University of Vienna, Austria.

Strohmer T (1997). Computationally Attractive Reconstruction of Band-Limited Images from Irregular Samples. IEEE Transactions on Image Processing, 6(4):540–548.