Photo-Realistic Simulation and Rendering of Halos

Jean-Christophe Gonzato
Sylvain Marchand

LaBRI[1]
Université Bordeaux 1
351, cours de la Libération, F-33405 Talence cedex, France
[gonzato|sm]@labri.u-bordeaux.fr

ABSTRACT

We present a technique for efficiently generating photo-realistic pictures of halos and inserting them into existing photographs. First, we describe an algorithm for producing images of halos from physical parameters either coming from existing photographs or 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. Finally, we explain the complete algorithm for effectively including computer-generated halos into real photographs, thus leading to new pictures with halos looking as if they had been part of the natural scenes captured by the camera.

Keywords: halo, natural phenomena, irregular sampling & reconstruction, augmented reality

1 Introduction

In this paper we aim at generating photo-realistic images of halos that could be inserted into real photographs. Halos could be compared to rainbows, although in halos the sunbeams are not deviated by droplets but by ice crystals. In order to insert halos in real photographs, we first have to compute the optical phenomenon, depending on the position of the light source – usually the sun – together with the locations and shapes of the crystals. The implemented algorithm, issued from physics, produces pictures sampled in a non-uniform way. More precisely, the value of the halo is not known for the whole picture. We propose an efficient reconstruction algorithm in order to recover the missing part of the halo data. Finally, a real photograph and the reconstructed halo are combined together into a new picture now taking the physical phenomenon into account.

In Section 2, we briefly introduce the physics of halos and the way these halos can be modeled and generated using a computer. As mentioned above, the resulting halos are not completely defined. We present in Section 3 an efficient way to reconstruct the missing data, based on the non-uniform sampling theory. We show then in Section 4 how we manage to retrieve essential information from real existing photographs in order to generate realistic halos, which can be successfully combined to the initial photographs after the reconstruction step for the halos. Finally, we present some pictures where halos have been inserted according to our method at the end of this paper.

2 Generating Halos

Solar halos are visible during sunny winter days when there is moisture in the air. Generally, one can see a – either partial or complete – lighted circle located at 22 degrees away from the sun. Many other kinds of halo exist, although they are less unlikely to occur in practice. Figure 1 shows an upper-tangent arc together with sun dogs.

2.1 Physical Phenomena

To easily explain these phenomena, let us make a comparison with the well-known rainbows. A rainbow is 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. The sun rays are also deviated when they travel through the crystals. Nice photographs of various types of halos can be found in [6, 9].

[1]Laboratoire Bordelais de Recherche en Informatique (*Université Bordeaux 1* and *Centre National de la Recherche Scientifique*). The present work is also supported by the *Conseil Régional d'Aquitaine*.

Figure 1: Photograph of a 22-degree halo with an upper-tangent arc and sun dogs.

Ice Crystals. Generally, these ice crystals have approximately the same shape. More precisely, they can be regarded as regular hexagons as shown in Figure 3. The crystals can look like pencils (a) or plates (b) depending on the value of their ratio of length to radius.

When a sunbeam hits one of these crystals, series of reflection and refraction phenomena generally occur. Figure 2 illustrates this. More precisely, when the ray first hits the crystal, it may either be reflected at its surface (a) – thus staying outside – or be refracted into its volume (b). When the sun ray enters the crystal, further reflections may occur on the inner faces until a final refraction finally allows the light to exit the crystal.

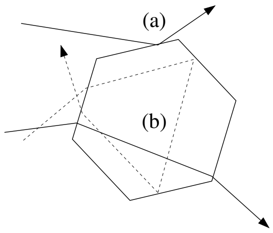


Figure 2: Several interactions between sun rays and ice crystals, depending on whether or not the sun ray is directly reflected at the crystal surface (a) or refracted into the crystal volume (b).

All of these ice crystals – both pencils and plates – constitute an invisible cloud of particles slowly falling across the atmosphere. What matters then is the radii and lengths of these particles as well as the way the particles are organized within the cloud. In a non-turbulent atmosphere, the crystals fall in such a way that the surface friction is maximal, as shown in Figure 3. When the atmosphere is turbulent, the crystals are oriented randomly.

Various Halos. 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

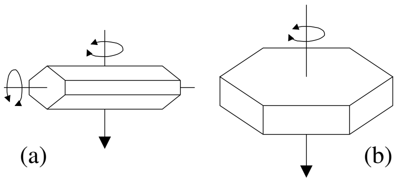


Figure 3: Shapes of standard ice crystals. Crystals with length-to-radius ratios above or below 2 are called pencils (a) or plates (b), respectively. In a non-turbulent atmosphere, the ice crystals fall in such as way that the surface friction is maximal.

also on the location of both the light source – usually the sun – and the observer. Figure 4 shows all the different kinds of halos. Although these halos have different names – depending mainly on their position in the sky – they are all produced by the same physical light-crystal interaction described above.

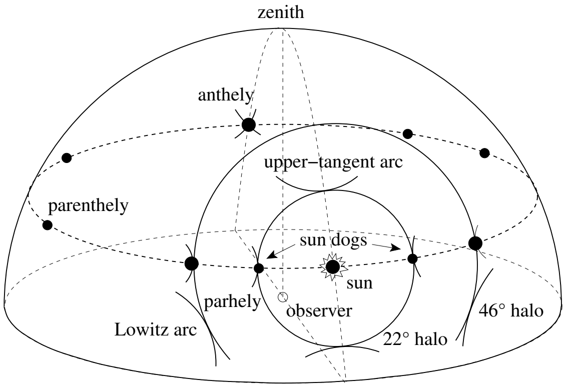


Figure 4: The most common halos shown on a schematic view of the celestial sphere.

2.2 Computer Simulation

Initially designed in order to verify the consistency of physical theories, simulation algorithms turn out to improve the realism of computer images. Moreover, they can advantageously enhance existing images by taking physical phenomena into account.

2.2.1 Previous Works

Although the very first simulation algorithms were generating only black and white pictures of halos, recent techniques can generate grey-scale or even colored halos. As far as we know, only two main works have been presented in computer graphics: the first one by Glassner [5, 4] in 1996, the second one by Jackèl and Walter [8] in 1998. Let us first present the basis of all these halo simulation algorithms, proposed by Greenler [6] in 1980.

Greenler Algorithm. Greenler's basic hypothesis [6] is that a ray of sun is deviated by one single ice crystal during its travel through the iced cloud. The different types of natural crystals are approximated by regular hexagonal crystals as we have seen above.

Computing, as in standard ray tracing, the quantity of light passing through each pixel of an image representing halos is not realistic with a simple computer. 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. By using inverse ray-tracing, knowing the direction of the light ray 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 these steps:

1. Choose one crystal – knowing the distribution of the various types in the cloud – with a random orientation in the range of possible orientations;
2. Cast a ray from the light source (the sun for instance) to the crystal and compute the deviation;
3. Place the crystal in the atmosphere in order to see the light leaving the crystal (see Figure 5). This positioning is simply done by plotting a light point on a fish-eye view image of the atmosphere.
4. Repeat the previous steps for an user-predefined number of crystals and orientations.

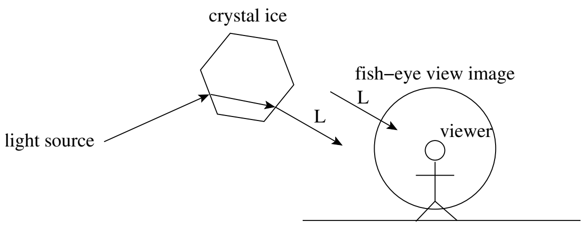


Figure 5: The basic halo simulation technique.

The resulting image simulating the halo phenomenon is in black and white. Moreover, the value of the halo is not known for the whole image. More precisely, the resulting spot diagram represents the intensity pattern in the sky that we should see from light passing through a cloud of ice crystals (similar to Figure 7(a), but in black and white).

Colorless Halos. In 1996, Glassner [5] enhanced the Greenler algorithm by taking into account the real part of energy coming from the sun and reaching the eye of the viewer. The resulting image – also a spot image – represents the halo phenomenon in grey scale. In fact, the author stores for each pixel the quantity of light hitting the pixel. This quantity comes from possibly different light-crystal interactions and by the loss of energy during reflection-refraction phenomena. The range of energy for each pixel of the whole image is spread in the classic range of grey-scale pictures (e.g. 256 levels). Figure 7(a) shows such a picture, resulting from our implementation.

Colored Halos. Colored halo phenomena are mainly visible near the upper-tangent arc and near the sun dogs (see Figure 4). The simulation algorithms for colored halos are derived from the grey-scale techniques, by casting several rays of light with different wavelengths. Two works present how to integrate color in halo simulation. Glassner, in 1996 [4], first proposed to take into account the different paths covered by the different wavelengths – of color spectrum – in the crystal. The color spectrum is divided into 7 wavelengths. Jackèl and Walter, in 1998 [8], propose to decompose the color spectrum into 21 wavelengths. For each pixel, a spectrum is stored and the final color is reconstructed after the simulation has ended. To convert a color spectrum into the classic RGB colorspace, the method proposed by Hall [7] in 1989 using XYZ colorspace is used.

Iterative or Parallel Algorithms. The main problem with the use of an inverse ray-tracing is the huge number of rays to cast from the light source to simulate the halo phenomena as accurately as possible. Jackèl and Walter [8] propose to use a parallel algorithm in order to cast about 500,000 rays per halo generation, 100 orientations per ray, and 21 wavelengths per orientation. Their resulting image is a dot matrix smoothed by the number of light impacts. Glassner [5] proposes to cast a much smaller number of rays. As a consequence, the intensity of the light coming from the halo is not known for the whole image. In order to reconstruct the missing data, he proposes to use different types of filters. This method is explained in the next section.

2.2.2 Implemented Algorithm

The algorithm for halo generation we have implemented is not really new. In fact, we have implemented the Greenler algorithm enhanced by Glassner for the real intensity of light coming from the light source. The parts of energy reflected (K_r) and refracted (K_t) in function of the incident angle (θ_i) and of the refraction index of the ice ($n_2 = 1.31$) are computed by the classic Fresnel laws:

$$\begin{cases} K_{r1} = \frac{n_1 \cos\theta_i - n_2 \cos\theta_t}{n_1 \cos\theta_i + n_2 \cos\theta_t} \\ K_{r2} = \frac{n_2 \cos\theta_i - n_1 \cos\theta_t}{n_2 \cos\theta_i + n_1 \cos\theta_t} \\ K_r = (K_{r1}^2 + K_{r2}^2)/2 \\ K_t = 1 - K_r \end{cases} \quad (1)$$

where n_1 the refraction index of the air ($n_1 = 1$), and θ_t is the angle of refraction computed by the well-known Snell-Descartes law.

We choose to simulate halos by an iterative algorithm where the user has the possibility to define various types of parameters: the types of the crystals, the range of possible crystal orientation, the number of rays traced, the number of levels of grey (for a colorless halo) and the number of wavelengths composing the full color spectrum (for colored halos). Figure 7(a) and its enlargement Figure 8(a) show images produced using our algorithm. As for the Glassner algorithm, the intensity of the halo is not known for the whole image.

3 Recovering the Complete Halo

The algorithm for halo simulation described above produces images sampled in a non-uniform way (see Figures 7(a) and 8(a)), each dot representing the intensity pattern of the halo. The problem is now, from this irregular sampling, to recover the missing information in order to reconstruct the whole images (see Figures 7(b) and 8(b)) from their samples taken in a irregular way.

3.1 Glassner Method for Reconstruction

Glassner proposes in [4] 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 use an uniform reconstruction technique for non-uniform sampling.

3.2 Irregular Sampling Theory

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).

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:

$$\exists \Omega_s > 0,\ support(S) \subseteq [-\Omega_s, +\Omega_s] \quad (2)$$

$$\text{where } S(\Omega) = \int_{-\infty}^{+\infty} s(t)\ e^{-j\Omega t}\ dt \quad (3)$$

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 so-called Nyquist period $T_s = \pi/\Omega_s$.

3.3 Reconstruction Algorithm

Most irregular reconstruction algorithms are iterative in nature [1]. 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. 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 cut-off 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.

Strohmer has studied in [10, 3, 2, 11] the reconstruction of images from irregular sampling. In step 1, different interpolations can be used. We use the Marvasti method, that is the Allebach algorithm together with the trivial interpolation (all the unknown values are set equal to zero). The iterative scheme of the Allebach algorithm would have converged faster with the Voronoi interpolation (nearest neighborhood interpolation, using the arithmetic mean for equally-spaced neighbors), but the computation of Voronoi diagrams was too slow to suit our needs.

3.3.1 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, more precisely to $\mathrm{sinc}(t/T_s)$ where:

$$\mathrm{sinc}(t) = \frac{\sin(\pi t)}{\pi t} \quad (\text{and } \mathrm{sinc}(0) = 1) \tag{4}$$

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):

$$w_N(t) = \frac{1}{2}\left(1 - \cos\left(\frac{2\pi t}{N-1} + \pi\right)\right) \tag{5}$$

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:

$$M_N(x,y) = w_N(t(x,y))\ \mathrm{sinc}(t(x,y)/T_s) \tag{6}$$
$$\text{where } |x| \leq N/2,\ |y| \leq N/2 \tag{7}$$
$$\text{and } t(x,y) = \sqrt{x^2+y^2} \tag{8}$$

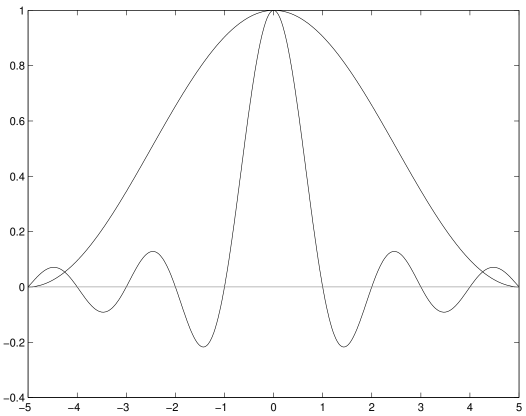


Figure 6: The impulse response of the reconstruction filter – here in the one-dimensional case for $N = 11$ (i.e. $k = 5$) and $T_s = 1$ – is the product of the *sinc* function with the bell-shaped Hann window.

3.3.2 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):

$$N = 2kT_s + 1 \tag{9}$$

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.

4 Combining Halos and Real Pictures

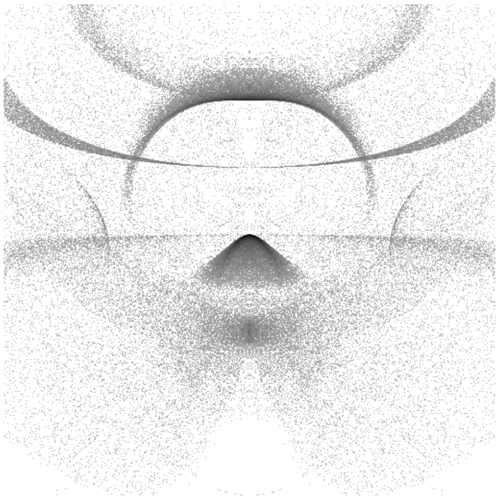
In this section, we present how we include halos in real photographs once they have been generated and reconstructed. This implies to retrieve essential information from a photograph like sun elevation, camera parameters, etc. Then, the way of inserting computer-generated halos into existing photographs is discussed.

4.1 Recovery of Physical Parameters

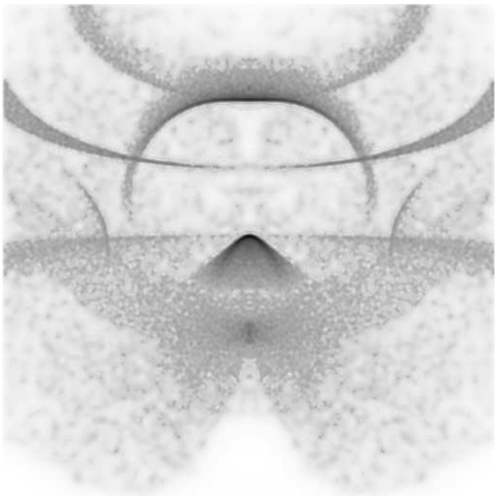
In order to generate a halo, the user must supply various data. Some data concern the snapshot (focal length used by the camera) whereas others concern the sun itself (sun color and elevation relatively to the horizon).

First, the user is asked to define the focal length of the camera which has taken the snapshot. If the user cannot determine or evaluate this length, the program chooses a standard 50-mm lens. To determine the elevation of the sun in the atmosphere, the user has then to draw on the photograph the horizon line and has to indicate the position of the sun. The elevation of the sun is automatically determined in function of the focal length used for the photograph (see above). If the sun or the horizon is not visible in the photograph, the user has to evaluate the elevation of the sun relatively to the horizon. Finally, the color of the sun is recovered from the picture.

When the horizon line and the sun position are user-defined, the elevation of the sun is automatically com-

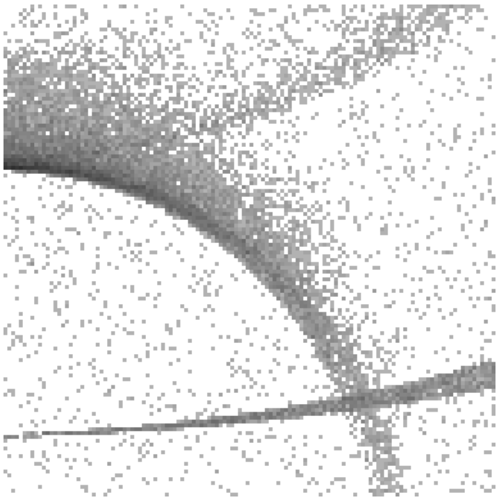


(a) Irregular Sampling

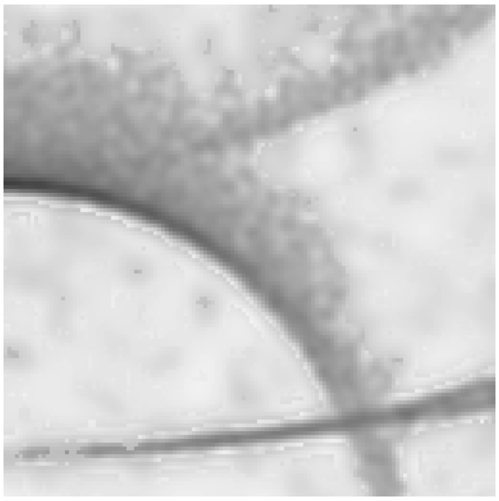


(b) Uniform Reconstruction

Figure 7: Halo reconstruction.



(a) Irregular Sampling



(b) Uniform Reconstruction

Figure 8: Halo reconstruction (zoom).

puted by the generalized Pythagoras formula (see Figure 9). First, the length between the horizon and the sun (noted b) is computed. Second, we compute the length between the focus point and the sun (c), as well as between the focus point and the horizon (a). To compute the different lengths and to know the height and width of a pixel, we assume that the original photograph was taken with a standard 24×36 mm (i.e. a $2/3$ ratio). If the height-to-width ratio of the manipulated image is not equal to $2/3$, the algorithm resizes automatically – by adding by virtual black pixels – the image in order to have the good ratio. As a consequence, the pixels are always square. The elevation of the sun from the horizon line is equal to:

$$\alpha = \arcsin\left(\frac{a^2 - b^2 + c^2}{2ac}\right) \tag{10}$$

After these steps of image data analysis, our algorithm can simulate the halo by using the techniques presented in Sections 2 and 3. The view frustum of both the computer-generated halo image and the real photograph are the same thanks to the length of focus and the horizon line defined by the user.

4.2 Modification of the Original Picture

After halo simulation using the different used-defined parameters, the final step of our algorithm is to inlay the halo in the photograph. Three cases must be considered, depending on the distance of the user to the cloud of ice crystals. More precisely, if the cloud is

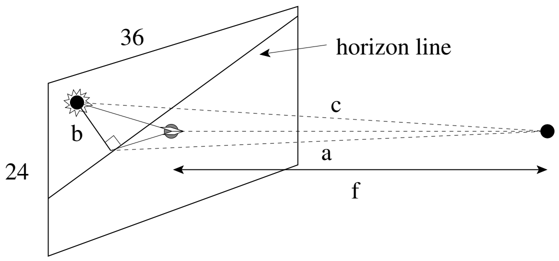


Figure 9: Computation of the sun elevation.

far from the user, then the generated halo has only to be applied above the horizon line. In a second case, if the user is inside the cloud, he can view the phenomenon in its whole view field, so the halo has to be applied to the whole image. The last case is an intermediate one. The halo can be seen below the horizon line but not on the whole image. In this last case, the user has to define the zone where the halo is visible. This zone is stored in a black and white layer – white color meaning that the halo is visible.

Finally, we have three layers for making the insertion of the halo in the photograph: The first one is the photograph itself, the second is the halo simulation, and the third one is the mask of visible halo. The final image is computed by combining the original photograph and the color of the sun using the halo data as an alpha channel. Of course these data have been multiplied by the visibility mask of the halo prior to the combination.

4.3 Results

The plates presented in the last page of this article present two original photographs together with their halo-enhanced versions. On the original photograph, the sun position and horizon line are plotted. In the first one, the viewer is inside the cloud of ice crystals. In the second one, the cloud is far from the viewer, thus the visible halo is only above the horizon.

For the mountain photograph, 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.

5 Conclusions

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 two algorithms are part of a general method used to include halos in real photographs. The elevation of the sun in the real photo is computed, and the image of the generated halo is superimposed to the photograph by using three layers. The results are not only physically valid, but also visually realistic. Of course, our technique works not only for existing photographs but also for computer-generated images as well, all the physical parameters being then part of the scene model.

REFERENCES

[1] FEICHTINGER, H. G., CENKER, C., AND STEIER, H. Fast Iterative and Non-Iterative Reconstruction Methods in Irregular Sampling. In *Proceedings IEEE ICASSP* (Toronto, May 1991), pp. 1773–1776.

[2] FEICHTINGER, H. G., GRCHENIG, K., AND STROHMER, T. Efficient Numerical Methods in Non-Uniform Sampling Theory. *Numerische Mathematik 69* (1995), 423–440.

[3] FEICHTINGER, H. G., AND STROHMER, T. Fast Iterative Reconstruction of Band-limited Images from Non-Uniform Sampling Values. In *Proceedings of the Computer Analysis of Images and Patterns (CAIP) Conference* (Budapest, 1993), pp. 82–91.

[4] GLASSNER, A. Computer-generated solar halos and sun dogs. *IEEE Computer Graphics and Applications* (March 1996), 77–81.

[5] GLASSNER, A. Solar halos and sun dogs. *IEEE Computer Graphics and Applications* (January 1996), 83–87.

[6] GREENLER, R. *Rainbows, Halos, and Glories*. Cambridge University Press, 1980.

[7] HALL, R. *Illumination and Color for Computer Generate Imagery*. Springer-Verlag, 1989.

[8] JACKÈL, D., AND WALTER, B. Simulation and visualization of halos. In *ANIGRAPH* (1998).

[9] LYNCH, D. K., AND LIVINGSTON, W. *Color and Light in Nature*. Cambridge University Press, 1985.

[10] STROHMER, T. *Efficient Methods for Digital Signal and Image Reconstruction from Nonuniform Samples*. PhD thesis, University of Vienna, Austria, 1993.

[11] STROHMER, T. Computationally Attractive Reconstruction of Band-Limited Images from Irregular Samples. *IEEE Transactions on Image Processing 6*, 4 (1997), 540–548.

(a) Original Photograph

(b) Extraction of Physical Parameters

(c) Final Picture

Plate 1: Mountains with an ambient iced cloud.

(a) Original Photograph

(b) Extraction of Physical Parameters

(c) Final Picture

Plate 2: Sunset on sea with and without halo.