Photographic Depth of Field Blur Rendering

Pichard Cyril
DUBOI
221 bis Bld Jean Jaurès
92100, Boulogne, France
cyril@duboi.com

Michelin Sylvain
SISAR Team
University of Marne-la-Vallée
6, cours du Danube
77700, Serris, France
michelin@univ-mlv.fr

Tubach Olivier
DUBOI
221 bis Bld Jean Jaurès
92100, Boulogne, France
tubach@duboi.com

ABSTRACT

The purpose of our study is to introduce a new method for depth of field blur simulation used for compositing real and synthetic images in a realistic way. Existing methods do not simulate accurately the depth of field blur produced by photographic equipment. The problem is twofold: photometric and geometric. Existing methods based on convolution do not produce a realistic photometric rendering compared to real photograph and the photographic blur contains patterns associated with the diaphragm shape, neither handled by existing methods. Our method, based on image processing, addresses these two issues by following the photographic image creation process. It can be used on synthetic or real images, essentially for special effects compositing.

Keywords

Depth of field blur, Image compositing, Image processing, Special effects, Boke

1. INTRODUCTION

Depth of field is an inherent phenomenon of optical systems. Cinema special effects intensively mix and compose images from different sources: computer graphic images, photographs, matte-painting, stock shot and real shot. The goal is to obtain the maximum of realism in the composite. One aspect of realism is the depth of field blurs matching in composited images, so frequently the depth of field blur has to be simulated on sharp images. The depth of field blur has some visual characteristics that existing methods do not handle. Photographers know that defocused lights tend to take on the characteristic shape of the camera's aperture. This phenomenon is not simulated by any existing method. Moreover, the lighter values tend to swallow dark values on defocused images, and methods based on convolution do not deal with this issue.

After introducing previous work, there follows an explanation as to how we simulate the lens, and a presentation of a method that takes into account visual photographic defocus.

Conference proceedings ISBN 80-903100-7-9
WSCG'2005, January 31-February 4, 2005
Plzen, Czech Republic.

2. PREVIOUS WORK

Image processing

Potmesil and *al.* [Pot81] were the first to present a method able to generate depth of field blur. This method is based on convolution with a varying kernel size dependant on the point depth on the pixel. Kernel matrices are computed using characteristics of the confusion circles created by a thin lens. Shinia [Shi94] improved Potmesil's method by adding ray-tracing to compute masked pixels.

Computer graphics

Cook et *al.* [Coo84] describe an anti-aliasing ray-tracing method that can be used to simulate depth of field blur and motion blur. The idea here is to use anti-aliasing rays to simulate light beams. This method is named distributed ray-tracing. Kolb, Mitchel and Hanrahan [Kol95] describe a camera model based on physical properties of lenses. Their principle is to follow light rays through every optical component of the lens from film to scene. Interactive depth of field blur [Hae90] [Nei99] can be achieved with OpenGL using an accumulation buffer. This method is based on accumulating different images taken from near viewpoints. There are no physical properties behind this method, so the simulation is not accurate. Heidrich, Slussallek, Seidel [Hei97] extend this principle to improve the quality of the rendering by computing the characteristics of the various accumulated views.

3. LENS SIMULATION

A good review on lenses can be found in [Ray02]. To simulate the lens, we use the thin lens model

described by Potmesil [Pot81]. We first add a diaphragm shape to this model.

Diaphragm

The diaphragm is an aperture of varying size generally placed inside the lens. Each aperture diaphragm has a specific shape due to its conception. Diaphragms are generally made with blades' leaves that form an iris. The more blades there are the more circular the shape is. The side effect is the modeling of light beams' shapes, which affects the form of the light stains projected on the film. To simulate the form of the aperture, we use a Bezier curve. These kind of curves can handle every form of existing diaphragm. Rounded corners of diaphragms can be simulated accurately.

4. FILM SIMULATION

Physical phenomenon

Physically, the film records exposure values (in lux/s-1). The film is chemically processed and generally scanned to obtain a digital image. The final image I is in pixel values, with no physical unit. There is a non-linear mapping between pixel values and exposure values [Deb97].

On the film, each disk of light mixes with the others and produces the image, blurry or not. Physically, we can consider that a sort of convolution is done with exposure values. Methods like Potmesil's use convolution of pixel values. We can notice in Figure 2 that the convolution of pixel values does not produce the same visual result like defocused image does in Figure 3. The convolution of pixel values appears darker. Our idea is to simulate this phenomenon using inverse mapping of the film. We want to mix pixel values in a different space value, ideally in the exposure space. Unfortunately the conversion curves between exposure values and pixel values are unknown and impractical to obtain for most of the source images we use in compositing environment: film, photographs, matte-painting, stock-shots... Moreover, some exposure values cannot be recorded on the film because they are too high, for instance, light peaks. Therefore, they are value-saturated and we cannot obtain their exposure information without high dynamic range information [Deb97]. We propose here a simple method to simulate this conversion without using conversion curves and high dynamic range information.

Simple algorithm for film rendering

Our first idea is based on the fact that the bright values tend to swallow darker values, as with dilation. This phenomenon shows the importance of the apparent brightness. The brighter the pixel is, the bigger the effect. In order to characterise the brightness of a pixel $I(I_r, I_g, I_b)$ we use the digital luminance commonly defined as follows:

$$L(I) = 0.3 \times I_R + 0.59 \times I_G + 0.11 \times I_B$$

Figure 1. Original image

Figure 2. Gaussian blur

Figure 3. Defocused image

Note that this luminance does not have any physical property. The conversion we propose is simple. We apply a conversion function F to the initial pixel value. This function F is used to control the swallowing of the dark values and the boosting of light values. Simple conversion functions can be used for F like gamma function, or look-up table handled by Bezier curves. Processed pixels values are then pre-multiplied by the digital luminance of the source pixel. Then, an optical effect can be simulated, as occurs in exposure values: motion blur,

depth of field blur, flare effects, etc. There follows an explanation of how to make realistic depth of field blur using this conversion. Note that we have to store the pixel luminance contribution during operation, therefore, we use a special channel. To convert back in pixel values, we have to divide by the luminance contribution, and apply the inverse function F^{-1} (see Fig.4).

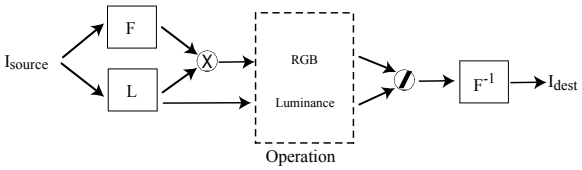

Figure 4. Film rendering principle

5. DOF BLUR ALGORITHM

Principle

Our algorithm is a mixture between Potmesil's method and OpenGL methods [Hae90][Hei97]. Like Potmesil's method we use an array of matrices and the method is based on image processing. As with OpenGL methods, we use the principle of the accumulation buffer. The algorithm works in 3 stages:

- Matrices creations
- Accumulation of light stains
- Final rendering of accumulation buffer into a frame buffer.

Mask matrices creation

The mask matrices M_i are made by rasterizing the Bezier diaphragm shapes. Figure 5 shows different rasterized masks matrices. For each depth, we have to create a mask matrix. The size of the mask calculated in pixels depends on the image format size and the size of the stain. For the current application we only store 256 matrices of different sizes, with the same shape, each associated with an alpha value. A curve is used to convert pixel depth to alpha values. Computer artists can manipulate this curve. To fill the mask matrices, we do not use intensity distribution as with Potmesil's method; we only use black or white values. This choice has been made for several reasons: light distribution for circular shapes is known but for unknown shapes the light distribution is undefined.

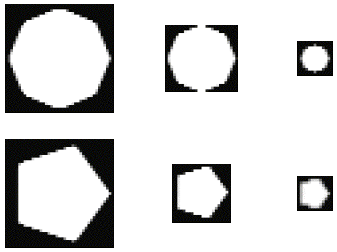

Figure 5. Rasterized Bezier diaphragms

Light distribution depends on light waves, which are difficult to obtain with pixel values. Moreover, the use of intensity distribution would increase computing time and the visual benefit would be nominal when taking the time into account.

First stage

Our algorithm follows the photographic image creation process. The processing consists of accumulating each light stain produced by each theoretical point of the scene. Theoretical points are represented by the source image pixels. This accumulation is done in an accumulation buffer (see Fig.6). The algorithm works as follows:

1. For each pixel $I(p')$ of the source image
    1. Compute luminance $L(I)$,
    2. Compute value $I_F = F(I_{source})$,
    3. Multiply L with I_F: $I_{premult} = L(I) \times I_F$
    4. Accumulate the resulting value $I_{premult}$ in the accumulation buffer through the appropriate mask M_i.
    5. Also accumulate luminance contribution L through the mask in a special channel.

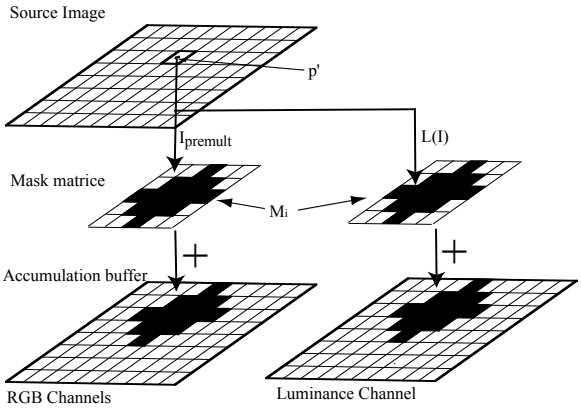

Figure 6. Accumulation principle

Second stage

The second stage is necessary to return to pixel values. This can be seen as the "chemical process". When the accumulation of confusion stain values is finished, the result is copied back from the accumulation buffer into the frame buffer, dividing by the contribution of luminance stored in the special channel.

6. RESULTS

Figure 7 shows the real defocused image of the scene and Figure 8 the original image processed with our algorithm. We can see that the method produces the same kind of visual effect. But we have to correct the colour of the original image to obtain similar values and to digitally remove the bridge fences in order to obtain this image. We predict the result would be

better in terms of colour and visual effects if we were to use high dynamic range images.

Figure 7. Defocused image

Figure 8. Defocus simulated

The second issue is that we generally do not have an image depth map. So for Figure 8 we used a constant value of mask. We can notice that the effect seems bigger at the bottom of Figure 8 than in Figure 7. This is because of the constant depth mask used.

For large size of defocus, we have to use multi-resolution techniques because computing time can be too important: it is possible to reach half an hour for a 3000x2000 16bit/channel image on a 1Ghz Pentium 4.

7. CONCLUSIONS

Our method can produce visual characteristics of photographic defocus by simulating diaphragm shape and film response. It can be used on real and synthetic images to produce a photo-realistic depth of field blur. This method tends to produce satisfactory results in terms of user satisfaction and has been intensively used on several French movies: “Immortel” by director Enki Bilal, “Un long dimanche de fiançailles” by director Jean Pierre Jeunet, and many others.

Nevertheless, we have made many assumptions: the shape does not turn with the depth, distribution of intensity is constant, the size of the image does not vary with focusing, etc. None of these assumptions provide enough visual improvement compared to the time increase. In any case, this method is included in our image compositing software DUTRUC, so computer artists can pre-process original images in order to obtain a satisfactory result.

8. REFERENCES

[Coo84] Cook, R., and Porter, T., and Carpenter, L. Distributed Ray Tracing. In SIGGRAPH 84 conf.proc., Computer Graphics, pp.137-144, 1984.

[Deb97] Debevec, P.E., and Malik, J. Recovering High dynamic range radiance maps from photographs. In Computer Graphics and Interactive Techniques conf.proc., ACM Press, pp.369-378, 1997.

[Hae90] Haeberly, P., and Akeley, K. The accumulation buffer : Hardware support for high-quality rendering. In SIGGRAPH 90 conf.proc., Computer Graphics, ACM Press, pp.309-318, 1990.

[Hei97] Heidrich, W., and Slusallek, P., and Seidel, H. An image-based model for realistic lens systems in interactive computer graphics. Graphic Interface 97, pp.68-75, 1997.

[Kol95] Kolb, C., and Mitchell, D., and Hanrahan, P. A realistic camera model for computer graphics. In SIGGRAPH 95 conf.proc., Computer Graphics, ACM Press, pp.317-324, 1995.

[Nei99] Neider, J., and Davis, T., and Woo, M. OpenGL Programming Guide : the official guide to learning OpenGL, version 1.2. Addison-Wesley, 1999.

[Pot81] Potmesil, M., and Chakravarty, I. A lens aperture camera model for synthetic image generation. In SIGGRAPH 81 conf.proc., Computer Graphics, ACM Press, pp.297-305, 1981.

[Ray02] Ray, S.F., Applied Photographic Optics, third edition. Focal Press, 2002.

[Shi94] Shinia, M. Post-filtering for depth of field simulation with ray distribution buffer. In Graphic Interface 94 conf.proc., Canadian Computer Communications Society, pp.39-66, 1994.