Computers & Graphics 55 (2016) 21–32

Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

Technical Section

Depth of field synthesis from sparse views ☆

Xin Liu, Jon G. Rokne *

Department of Computer Science, The University of Calgary, Calgary, Alberta, Canada T2N 1N4

ARTICLE INFO

Article history:
Received 13 August 2015
Received in revised form
7 October 2015
Accepted 26 October 2015
Available online 19 November 2015

Keywords:
Image synthesis
Depth of field
Sparse view
Visibility

ABSTRACT

Computer generated images are most easily generated as pinhole images whereas images obtained with optical lenses exhibit a Depth-of-Field (DOF) effect. This is due to the fact that optical lenses gather light across finite apertures whereas the simulation of a pinhole lens means that the light is gathered through an infinitesimal small aperture, thus producing sharp images at any depth. Simulating the physical process of gathering light across a finite aperture can be done for example with distributed ray tracing, but it is computationally much more expensive than the simulation through an infinitesimal aperture. The usual way of simulating lens effects is therefore to produce a pinhole image and then post processes the image to approximate the DOF. Post processing algorithms are fast but suffer from incorrect visibilities. In this paper, we propose a novel algorithm that tackles the visibility issue with a sparse set of views rendered through the optical center of the lens and several peripheral viewpoints distributed on the lens. All peripheral images are warped towards the central view to create a Layered-Depth-Image (LDI), so that all observed 3D points located on the same central view-ray are stacked on the same pixel of the LDI. Then, each pixel in the LDI is conceptually scattered into a Point-Spread-Function (PSF) and blended in depth order. While the scatter method is very inefficient on a GPU, we propose a selective-gather method for DOF synthesis, which scans the neighborhood of a pixel and blends the colors from the PSFs covering the pixel. Experiments show that the proposed algorithm can synthesize high-quality DOF effects close to the results of distributed ray tracing but at a much higher speed.

1. Introduction

An optical lens collects light across a finite aperture and produces images with *Depth-of-Field* (DOF), a phenomenon that results in in-focus objects being sharply imaged and out-of-focus objects are blurred to different degrees. The DOF effect plays a crucial role in an image's photorealism and depth feeling [35], and it is widely used in cinematography to guide the audiences' attention and to create a special artistic atmosphere. It is thus important to simulate DOF effects when there is a requirement for photorealism in computer graphics.

A DOF simulation system for computer graphics is logically composed of a 3D renderer and an image processor. We classify existing algorithms according to the combination of the two components. High quality DOF effects can be synthesized by *object space algorithms* [2], which faithfully simulate the physical imaging process of an optical lens. However, those algorithms are very costly in terms of computing time. Therefore, a number of *post-processing algorithms* [3] has been devised to synthesize DOF effects from a pinhole image rendered through the lens' optical center. The post-processing algorithms are generally fast, but the quality of synthesized images is limited due to annoying artifacts. These artifacts are especially conspicuous when an in-focus object is partially occluded by a foreground object. The fundamental reason is that the pinhole image only covers a subset of an optical lens's visibility, and the visibility missing from the pinhole image cannot be recovered by image processing in 2D space. The term *visibility* is used in this paper to denote the set of points in a 3D scene that participates in the formation of a 2D image.

In this paper, we propose a novel algorithm that tackles the visibility issue by using a sparse set of views which are rendered through the optical center of the lens and a some peripheral points distributed on the perimeter of the simulated lens. The sparse views essentially quantize the continuous *light field* [31] across the aperture of the lens. To remove the disparity between the continuous distribution over the simulated lens and the discrete points on the lens we scatter pixels at the correct locations and warp all peripheral views towards the central view. By this we create a *Layered-Depth-Image* (LDI) [47], so that all 3D points located on the same central view-ray are stacked on the same pixel of the LDI. To save memory space and processing time, the depth-pixels at each discrete location of the LDI are clustered into a fixed number of layers according to their *perceptive depths*. Then,

☆This article was recommended for publication by Y. Gingold.

* Corresponding author.
E-mail addresses: xinliu@live.ca (X. Liu), rokne@ucalgary.ca (J.G. Rokne).

http://dx.doi.org/10.1016/j.cag.2015.10.015
0097-8493/

each pixel of the LDI is conceptually scattered into a Point-Spread-Function (PSF) and blended in depth order. While the scatter method is very inefficient on a GPU due to severe write-conflicts, we propose a *selective-gather* method for DOF synthesis, which scans the neighborhood of a pixel and blends the colors from the PSFs covering the pixel. To further speed up image synthesis, the PSFs are clustered into a fixed number of layers at each destination pixel according to the perceptive depths.

The proposed algorithm works equally well with both ray tracing and rasterization rendering algorithms. The peripheral views cover the lens' visibility missing from the central view to a satisfactory extent, which is important for high-quality DOF synthesis. They do not introduce *extra vision* not belonging to the lens, which would otherwise cause very objectionable see-through artifacts in the synthesized image.

Experiments show that with 5–9 views the proposed algorithm can render high quality DOF images close to the results of distributed ray tracing, but at a much higher speed.

The contributions of this paper are:

1. a new taxonomy of DOF synthesis algorithms according to the combination of 3D renderer and image processor,
2. a novel DOF synthesis strategy combining a multi-view renderer and a scatter image processor,
3. a GPU friendly algorithm for image warping and LDI creation, and
4. a GPU friendly selective-gather algorithm for DOF synthesis.

2. Literature review

DOF simulation has been intensively studied since 1980s. Barsky et al. [2,3,6] and Demers [13] made comprehensive surveys of this research field. From our viewpoint, every DOF simulation algorithm is logically composed of two components, i.e., a *3D renderer* and an *image processor*, and can be reasonably classified according to the combination of the two components. In this section we review some representative algorithms falling into each class, and thereby clarify the distinct features of the proposed algorithm.

2.1. 3D renderer

Each DOF simulation algorithm employs a 3D renderer to compute the light captured by a lens in a 3D scene. The 3D renderer provides the *raw data* for the image processor. According to their output, the 3D renderers can be classified into (1) multi-view (MTVW), (2) single-view-single-layer (SVSL), and (3) single-view-multi-layer (SVML) renderers.

A *multi-view* (MTVW) renderer [11,17] computes a set of the light samples across the lens which are hitting the image sensor (film). This constitutes a *lens light field*, as shown in Fig. 1(a). When the sampling is dense enough, the output can be used to synthesize physically correct DOF effects but at a very high computational cost.

A *single-view-single-layer* (SVSL) renderer computes a set of the light samples through the optical center of the lens which hit the image sensor (film) forming a *pinhole image*, as shown in Fig. 1(b). The pinhole image covers only a subset of the visibility of the lens. This means that some objects that are visible through a lens, as exemplified by the blue rectangle, are invisible through the optical center of the lens. The missing visibility cannot be recovered by any image processor and will cause unavoidable artifacts in the synthesized DOF image.

To address the visibility issue while keeping the computational cost fairly low, the *single-view-multi-layer* (SVML) renderer has been exploited, which captures a multi-layer pinhole image through the optical center of the lens. The SVML renderer can be implemented by either *scene stratification* [28,32], which stratifies the scene with parallel planes, as shown in Fig. 1(c), or *depth peeling* [30,29,14,33], which repeatedly peels off the exposed surfaces. The SVML renderer has some inherent drawbacks. First, decomposing a 3D scene destroys its structure and exposes objects invisible through the lens, as exemplified by the red disk in Fig. 1(c). The image processor cannot correctly solve the visibility in 2D space and may further expose the hidden objects in the synthesized image, causing the very objectionable *see-through* artifacts. Second, the surfaces parallel to the view-rays are missed out, as exemplified by the lateral edges of the green pentagon in Fig. 1(c).

2.2. Image processor

An image processor can be used to synthesize a DOF image from the raw data provided by the 3D renderer. Image processors

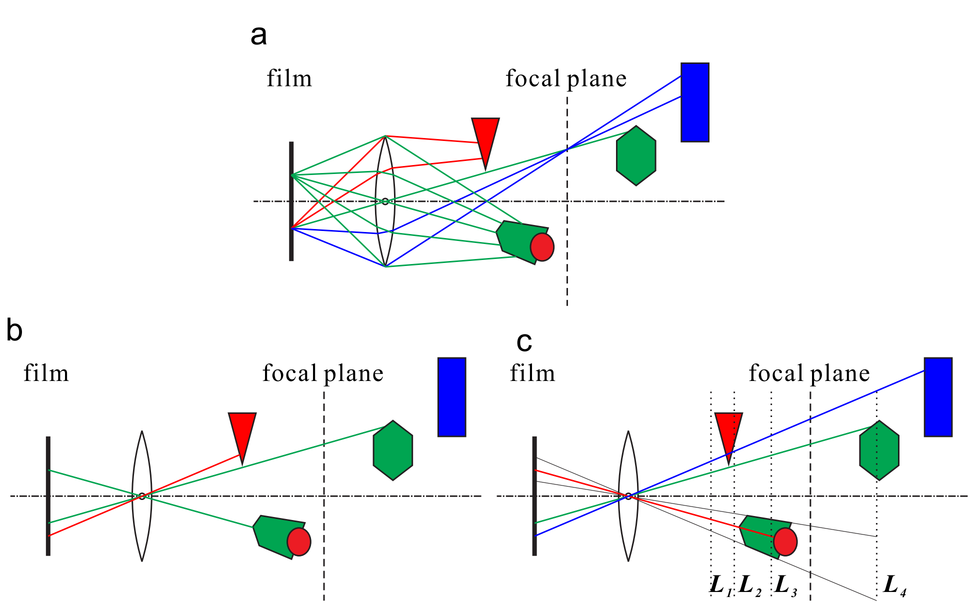

Fig. 1. Three types of 3D renderers: (a) multi-view, (b) single-view-single-layer, and (c) single-view-multi-layer. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)