Computers & Graphics 34 (2010) 361–369

Contents lists available at ScienceDirect

Computers & Graphics

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

Technical Section

Hybrid visibility compositing and masking for illustrative rendering ☆

Stefan Bruckner [a,*], Peter Rautek [a], Ivan Viola [b], Mike Roberts [c], Mario Costa Sousa [c], M. Eduard Gröller [a]

[a] Institute of Computer Graphics and Algorithms, Vienna University of Technology, Austria
[b] Department of Informatics, University of Bergen, Norway
[c] Department of Computer Science, University of Calgary, Canada

A R T I C L E I N F O

Keywords:
Compositing
Masking
Illustration

A B S T R A C T

In this paper, we introduce a novel framework for the compositing of interactively rendered 3D layers tailored to the needs of scientific illustration. Currently, traditional scientific illustrations are produced in a series of composition stages, combining different pictorial elements using 2D digital layering. Our approach extends the layer metaphor into 3D without giving up the advantages of 2D methods. The new compositing approach allows for effects such as selective transparency, occlusion overrides, and soft depth buffering. Furthermore, we show how common manipulation techniques such as masking can be integrated into this concept. These tools behave just like in 2D, but their influence extends beyond a single viewpoint. Since the presented approach makes no assumptions about the underlying rendering algorithms, layers can be generated based on polygonal geometry, volumetric data, point-based representations, or others. Our implementation exploits current graphics hardware and permits real-time interaction and rendering.

1. Introduction

Digital compositing was arguably one of computer graphics' first mainstream commercial applications. Areas such as motion picture post-production greatly benefited from automated processing. The ability to flexibly combine multiple sources ultimately lead to the widespread adoption of digital special effects which are now ubiquitous in the film industry. Today, virtually every image editing software package has the ability to arrange elements in layers, modify alpha channels, control blending, and apply effects to individual layers.

In this paper, we focus on the compositing of dynamic 3D content. Instead of combining static elements such as images, movie sequences, or pre-rendered 3D animations, an interactive environment allows the modification of properties such as the viewpoint for individual layers which are rendered on-the-fly. Common software tools such as Adobe Photoshop have recently introduced the ability to embed dynamically generated layers based on 3D models. However, for the purpose of composing, the layer content is still treated as a 2D image even though additional information would be available. One reason for this choice is the fact that the conventional layered compositing approach, which assumes 2D layers, is deeply incorporated into these software packages and the workflow of their users. In this paper, we present a concept for the integration of 3D layers which preserves this intuitive notion, but allows artists to take advantage of 3D information by extending the operator set of traditional compositing approaches. We deliberately make minimal assumptions about the algorithms and data structures used to generate layer content to facilitate non-invasive integration into conventional image manipulation software.

One of our target applications is the generation of technical, medical, and scientific illustrations which frequently make use of selective occlusion overrides and blending in order to produce an expressive image. Using our approach, an illustrator can modify 3D properties of the content layers without having to go through the entire compositing process again. However, the presented approach is general and not restricted to this particular scenario. While recent work in illustrative visualization has put special emphasis on the role of methods employed by graphics artists and illustrators, many approaches are limited to specific types of scientific data (e.g., scalar volume data) and/or application domains. Although our approach can handle dynamically changing 3D layers, it does not require knowledge about the underlying rendering algorithms. This enables the flexible integration of different data representations such as polygonal meshes, volumetric-, and point-based data as well as different rendering algorithms such as painterly, photorealistic, or illustrative rendering.

The remainder of this paper is structured as follows: Section 2 reviews related work. In Section 3, we discuss the concepts behind our hybrid visibility compositing approach. Section 4 presents a technique for masking of dynamic 3D layers. Section 5 details our implementation and Section 6 presents further results. We discuss our approach in Section 7 and conclude the paper in Section 8.

☆ Funded by: FWF.

* Corresponding author. Tel.: +43 1 58801 18643; fax: +43 1 58801 18698.
E-mail address: bruckner@cg.tuwien.ac.at (S. Bruckner).

0097-8493/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cag.2010.04.003

2. Related work

The work presented in this paper is related to several fields. Our approach is based on the large body of research in the area of digital compositing. Masking of 3D layers is related to image-based rendering techniques which attempt to generate novel views of 3D objects based on partial information. Furthermore, we were also inspired by illustrative and non-photorealistic rendering techniques which aim to reproduce the aesthetic quality of manually generated artwork using computer graphics.

Digital compositing: Digital compositing dates back to the early days of computer graphics as a scientific discipline. Alvy Ray Smith and Ed Catmull combined two images using a third image of coverage values which subsequently lead to the notion of the alpha channel [1]. Wallace [2] extended the approach to recursive blending allowing layers to be composited in any order that obeys associativity. Porter and Duff [3] introduced the concept of premultiplied alpha and formulated the compositing algebra which is in widespread use today. For the purpose of anti-aliased combination of 3D rendering results, Duff [4] proposed the rgbαz representation which, in addition to color and alpha channels, also includes a depth value for every pixel. Recent work by McCann and Pollard [5] extends the flexibility of traditional compositing by enabling stacking decisions on a per-overlap basis. However, their approach is designed to provide more control over the compositing of 2D layers, while the concept presented in this paper aims at facilitating the integration of 3D content in a consistent manner.

Image-based rendering: The idea of avoiding expensive rendering passes by using compositing to combine parts of a scene gave rise to the area of image-based rendering. Image-based rendering approaches attempt to synthesize novel views which closely approximate correct visibility from information captured during the generation of a single image. Lengyel and Snyder [6] proposed a factorization of 3D scenes into independent 2D sprites which could have different update rates. While their approach attempts to identify independent 2D layers, our method makes use of the available depth information and therefore allows full control over intersecting objects. Nailboards additionally store a depth value for every pixel of a sprite to enable the rendering of interpenetrating 3D objects with correct visibility [7]. Layered depth images contain multiple pixels along each line of sight to enable the generation of novel views with higher fidelity [8]. These approaches use layers with additional spatial information to combine parts of a scene with correct visibility. However, the focus of our work is different: we want to provide the user with the ability to selectively override occlusion relationships as it is common in technical, medical, and scientific illustrations.

Illustrative rendering: Several systems for the generation of illustrations using computer graphics have been developed. Dooley and Cohen [9,10] presented approaches for the automatic generation of semi-transparent line and surface illustrations from 3D models. Pioneering work by Seligman and Feiner [11–13] first treated the topic of visibility constraints. Their work employed cutaways and ghosting to resolve visibility conflicts. Preim et al. [14] presented Zoom Illustrator, a semi-interactive tool for illustrating anatomic models. Their approach focuses on the integration of three-dimensional graphics and textual representations. Diepstraten et al. [15,16] proposed rendering algorithms for ghosting and cutaway effects. Owada et al. [17,18] developed a system for modeling and illustrating volumetric objects. They semi-automatically generate artificial cutting textures based on surface models. Viola et al. [19] introduced the notion of importance-based rendering for improved visualization of features in volume data. Extending this approach, Bruckner and Gröller [20] presented a flexible interactive direct volume illustration system. Rautek et al. [21,22] proposed the use of semantic layers defined using a fuzzy logic rule base. Cole et al. [23] proposed a technique for generating architectural illustrations featuring a stylized focus area through local variations in shading effects and line qualities. Kalkofen et al. [24] used stylized overlays for focus+context visualization in augmented reality applications. Li et al. [25,26] presented geometric methods for generating high-quality cutaway and exploded view diagrams. Raman et al. [27] discussed a system which uses layer-based effects to enhance the visualization of volume data. Similar to our approach, the ClearView system presented by Krüger et al. [28] uses layered rendering to generate a number of different transparency effects inspired by traditional illustrations. However, their approach relies on a globally defined layer order, for instance nested isosurfaces of a volume dataset. Furthermore, their method only allows the use of a single spherical focus region.

In this paper we contribute with a new approach to combining interactively rendered 3D output based on the communication goals and stylization requirements of technical, medical and scientific illustrations. We introduce the notion of hybrid visibility compositing which allows integration of layered 2D compositing with 3D visibility operations in a flexible and intuitive manner. Additionally, we propose a new method for performing common masking operations based on this concept. The resulting framework enables the interactive generation of 3D illustrations featuring effects and techniques typically only available in 2D compositing software.

3. Compositing

Duff [4] was the first to propose the rgbαz representation for compositing 3D rendered images. In such a representation each pixel stores, in addition to its color and alpha value, a depth value. In a way, such an rgbαz image is a generalization of a 2D sprite [29]—points with color, transparency, and depth, but without any thickness information. Even though rgbαz layers are not a complete description of a general 3D object, they are a useful extension of conventional 2D layers. One of the main reasons why we choose this representation is that it requires minimal information about the actual data structures and algorithms used to provide layer content. A layer may be generated through ray tracing, rasterization of polygonal models, point-based rendering, or virtually any other technique capable of producing color and depth information.

We therefore choose this representation as one of the basic building blocks of our compositing framework. Each 3D layer is bound to a renderer instance and captures its output as an rgbαz image at any time. The content of such a layer may change dynamically, e.g., due to user interaction or animation. The compositing engine then decides how these layers are combined to form the final image. Since current graphics hardware allows us to easily access its color and depth buffers, one advantage of employing an rgbαz representation is that no modifications to the rendering stage are required. This means that the compositing engine can be used to combine layers produced by a variety of different rendering algorithms.

3.1. Implicit visibility

In contrast to 2D compositing where the stacking order of layers is solely specified by the user, rgbαz layers have an *implicit visibility* defined by the relative depth values of their pixels. The general technique for compositing multiple rgbαz layers with correct visibility is through a per-pixel application of the painter's