IMMERSIVE VIRTUAL REALITY FOR LIVE-ACTION VIDEO USING CAMERA ARRAYS

Matthias Ziegler, Joachim Keinert, Nina Holzer, Thorsten Wolf, Tobias Jaschke, Ron op het Veld, Faezeh Sadat Zakeri and Siegfried Foessel

Fraunhofer Institute for Integrated Circuits IIS, Germany,
Department Moving Pictures Technologies

ABSTRACT

The advent of high-resolution head-mounted displays (HMDs) enables new applications for virtual and augmented reality. The ability to freely move and look around in the scene with six degrees of freedom creates the feeling of being part of a fascinating story.

Based on today’s technology, content for six degrees of freedom is most easily built from computer-generated content. In order to make the story even more convincing, we propose to add photorealistic footage to the scene. Ultimately, this will allow integration of live-action elements with real actors and famous places for a more believable experience allowing the spectator to dive into the story.

In this work we present a corresponding lightfield workflow consisting of multi-camera capture, post-production and integration of live-action video into a VR-environment. Thanks to high quality depth maps, visual effects like camera movements, depth-guided colour correction and integration of CGI elements are easily achievable, allowing generating both 2D movies and VR content.

INTRODUCTION

Today, use of Computer Generated Imagery (CGI) is a key element in movie production. Techniques like match-moving allow the placement of actors in a computer-generated 3D environment. This requires the virtual camera to follow the path of the real camera in order to obtain a consistent sequence composed of live-action and CG-elements. Up to now, the final result of such compositing was always a fixed sequence of 2D or stereo images. A spectator had no option to change his point-of-view of the scene.

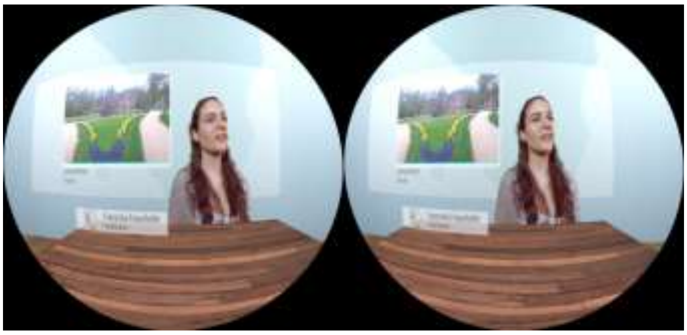

Figure 1 – Multi-View live action footage of a news-presenter is integrated in a CG-VR environment.

This situation changed when the first head-mounted displays (HMDs) appeared. For the first time it became possible for the audience to experience full 6 degrees of freedom (6-DOF) virtual reality (VR). A spectator can move freely through the scene with proper change in perspective. This effect is also known as motion-parallax. A VR compositing workflow needs to deliver content that allows for such 6-DOF. As before, live-action content and CG content need to be consistent.

For a long time, 6-DOF content that could be presented on such HMDs was limited to CGI. Against this background we propose a novel workflow that provides an immersive VR experience for live-action video. Our workflow incorporates a portable camera array capturing an actor. The obtained footage is processed using a set of specifically designed plug-ins for the compositing software *NUKE*. The processing thereby reconstructs a dense lightfield from a multi-camera input. Finally, we can import the 6-DOF live-action content into the *Unreal Engine* (UE) which is used as a playback platform. In combination with standard 3D elements we can create a 6-DOF VR experience that features significant head motion-parallax in the natural scene elements.

PREVIOUS WORK

In the past years, several authors and companies have presented first prototype like systems that can be used to capture live-action VR video.

In 2015, Anderson et al. [1] presented a circular camera rig consisting of 16 action cameras. The captured footage is processed by *Google* in a cloud system. After processing, the content can be played on a HMD providing a 360° stereo experience, but does not provide head-motion parallax. Furthermore, the processing pipeline is a black-box system, leaving no possibilities to compose different types of content in a creative way.

A similar system was presented in 2016 by *Facebook*[1]. In contrast to the system described by Anderson, the processing software has been published as an open-source project and can run on a standard computer. Users can capture with custom-built camera rigs and develop pipeline processes. As before, the system did not consider head-motion parallax. Very recently, Facebook announced a second-prototype system called x24. Compared to its predecessor, the new system is a closed-source processing pipeline. Moreover, the possible motion parallax is limited due to 'basketball' shaped camera array.

Lytro presented their new *IMMERGE* lightfield camera in early 2017 comprising about 90 cameras arranged on a plane in hexagonal shape. While the huge number of cameras promises good quality, it requires massive storage capabilities.

The technology of *8i* aims to record photorealistic human holograms with true volume and depth. Similarly, *TEN24-Media* uses a capture stage consisting of over 170 cameras to reconstruct the 3D shape of an actor using photogrammetry software.

Compared to these approaches, the following sections will present a system that does not need any mesh reconstruction, since it purely operates in the lightfield domain using a

[1] facebook360.fb.com/facebook-surround-360

planar camera array. Moreover, capture costs are reduced by converting a sparse lightfield into a dense one.

PROPOSED WORKFLOW

Our proposed system is presented, as an example, on a news-presentation scenario with the speaker located in front of a green-screen. Instead of a single camera as in classical 2D workflows we capture the actor using an array of cameras. This way, we can reconstruct depth maps of the actor allowing generation of content for a media experience with 6 degrees of freedom.

For the playback the spectator is assumed to be sitting on a chair with an HMD in front of his eyes. In this case, the maximum motion parallax that needs to be covered for a consistent immersive experience is defined by the movement of the spectator's head. Typically, this amounts to several centimetres. In this case, a small planar camera array, as the one proposed below, is sufficient.

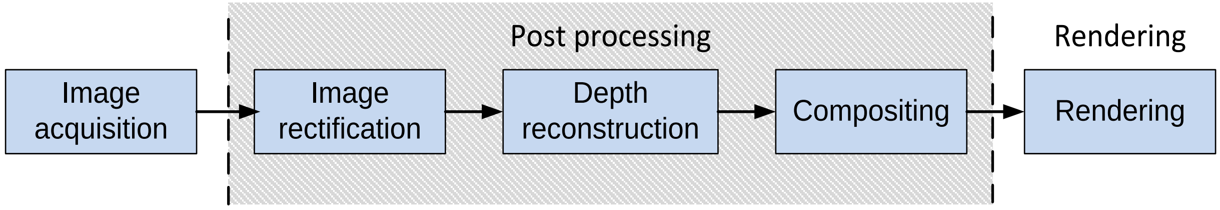

Figure 2 – Workflow overview: After the footage is captured, the images are rectified and the depth is reconstructed. The compositing comprises standard steps, like green-screen keying, but also combines live-action and CG-elements. Depth, colour and other 3D elements are subsequently imported in a rendering engine like Unreal Engine.

Specifically designed plug-ins for compositing software such as NUKE allow the integration of such multi-camera data into existing workflows, allowing easy generation of classical 2D output as well as providing 6-DOF content and a classical 2D output. The plug-ins are designed for various custom-built camera arrays.

The following sections will elaborate the steps of the workflow depicted in Figure 2 in more detail.

Camera array

Capturing dynamic scenes for playback in VR requires a set of cameras that capture the scene from different perspectives. For this purpose, we use a camera array. The 3x3 rectangular camera array used in this work is presented in Figure 3. It is built from *Black Magic Micro Studio 4K* cameras with horizontal and vertical spacing of 120mm and 78mm between adjacent cameras. The design allows it to be attached to normal tripods. Apart from the metallic frame, the system is built from off-the-shelf components typically used in media production

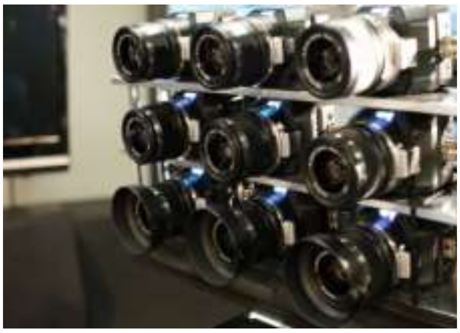

Figure 3 – 3x3 camera array built from *Black Magic Micro Studio 4K* cameras.

environments.

Scene configuration and image acquisition

Consider the situation as outlined above, with the news-speaker in the centre of the scene (Figure 4). Given such a setting, we need to setup the cameras and object distances such that the amount of parallax is sufficient for an immersive experience. Parallax is defined as the total amount of occlusion in the scene. It can be computed from the foremost and backmost object and is a measure of the change in perspective between two views.

Figure 4 schematically depicts the situation: Three equally spaced cameras are placed in front of the news-speaker and the green-screen. Here, the relevant depth of the scene is defined by the news-speaker and is about 1m. Equation (1) expresses the horizontal pixel coordinate *u* of a world point *M* projected on the image sensor. The total parallax between the foremost point M_F and backmost point M_B can be computed as in equation (2). The variable s_p denotes the pixel-size, *f* denotes the focal length and Δ_x denotes the distance between two cameras. In equations (1) and (2) it is assumed that the coordinate origin is located in the leftmost camera. Furthermore, it is assumed that the considered world points M_F and M_B are visible in both cameras.

$$u = \frac{1}{s_p} \cdot \frac{f \cdot M_x}{M_z} = C_0 \cdot \frac{M_x}{M_z} \text{ with } C_0 = \frac{f}{s_p} \qquad (1)$$

$$d = u_1 - u_2 = C_0 \cdot \Delta_x \cdot \frac{M_{F,z} - M_{B,z}}{M_{F,z} \cdot M_{B,z}} \qquad (2)$$

In order to obtain a high quality depth map for each stereo pair, we need to limit the parallax to about 4% of the image width. In this scenario, one obtains a minimum distance $M_{F,z}$ of 1.95m when the cameras are separated by Δ_x=120mm.

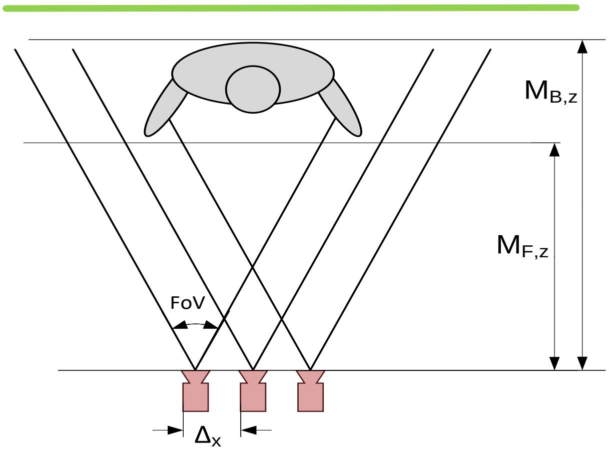

Figure 4 – The total amount of parallax can be computed from the scene-geometry and the camera setup. In this scenario, only the total depth of the actor is relevant for post-processing.

Post processing

Precise depth-reconstruction is a prerequisite for high-quality view-synthesis in a VR environment. Throughout this work, all required multi-camera image processing algorithms have been implemented as plugins for NUKE. In the first step, images are rectified such that pixels belonging to the same object points are situated in the same line or column of two camera images. Such calibration can be obtained using checkerboard methods as presented in [2]. Figure 6 shows the aligned images of the corner cameras in the array.

The precise alignment ensures that the subsequent depth-reconstruction can be executed with high efficiency and precision. Since the depth values for the green-screen in the back are of no interest for us, these areas are keyed in advance. Depth-reconstruction and subsequent filtering for planar camera setup have been presented in earlier works [3]. These algorithms have been integrated into plug-ins for NUKE. This allows for simple, scene-specific tuning of parameters. The right side of Figure 6 shows a resulting depth map using a colour representation. Blue areas are located in the back, while red areas correspond to regions which are closer to the camera array.

In the depth-reconstruction process such a map is obtained for every image in the array. These dense maps may now serve various purposes in classical 2D post-production tasks, like depth-based colour grading, relighting or refocusing. In addition, these maps can also be used to generate novel views. Depending on the application, such novel view synthesis can be done in NUKE to drive classical 2D compositing or it can be done in gaming engines like Unity or Unreal Engine, as will be explained in the ongoing part of this section.

Lightfield export & import

Integrating a video lightfield with a 3D gaming environment requires in a first stage, a suitable representation format for depth and colour information. In contrast to other approaches, we do not reconstruct a 3D model or a point-cloud. Instead, we directly export and import the colour information as a RGB image in addition to the per-pixel disparity.

Lightfield rendering

For the rendering, we follow the approach presented by Levoy and Hanrahan [4]. Their model represents the lightfield by the intersection of a ray with two parallel planes (UV- and ST-plane). It is also well known as a 4D-plenoptic function. A short summary (with respect to our considered scenario) will be given in the following:

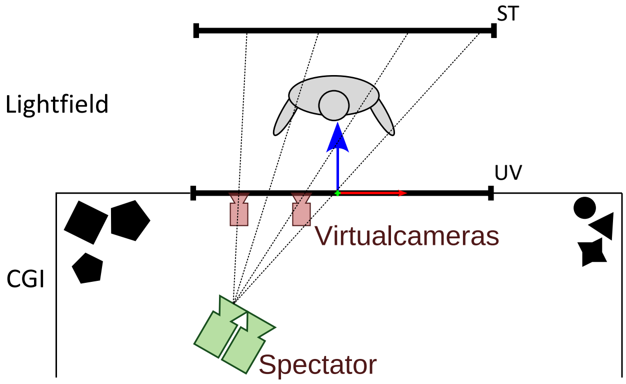

Figure 5 – Classical lightfield rendering as proposed by Levoy. The UV and ST planes are integrated in a 3D environment. The UV plane can be considered as window separating the 3D world and the lightfield.

In classical lightfield acquisition, a dense lightfield can be captured by sampling the *UV* plane at many different positions, as depicted in Figure 5. By these means we capture all possible rays traversing the UV plane. The spectator, depicted by a green pair of stereo cameras, is standing in the CG scene looking through an opening of the computer-generated world. This opening can be imagined as a window: Rays from the outside world pass through this surface and finally hit the camera (or the spectator's retina). In our scenario this outside world comprises the actor captured in front of the green-screen.

A ray can be represented by the sampling position (*u, v*), a pixel position (*s, t*) and the corresponding colour and brightness information. Given that we know all possible rays traversing the window, generation of a novel view at the location of the spectator is possible by finding all rays constituting this novel view.

In Figure 5, the *UV*-plane virtually separates the 3D domain and the lightfield domain. This model also shows an important property of such a lightfield: The area, wherein the lightfield can be observed is limited. I.e. it is not possible to go around the actor. In Figure 5, this is for example circumvented by the solid wall left and right of the window. Moreover, the spectator is not allowed to approach the news presenter closer than cameras capturing the lightfield. This allows the lightfield to be captured with a planar array, thus reducing capture costs while at the same time providing significant parallax.

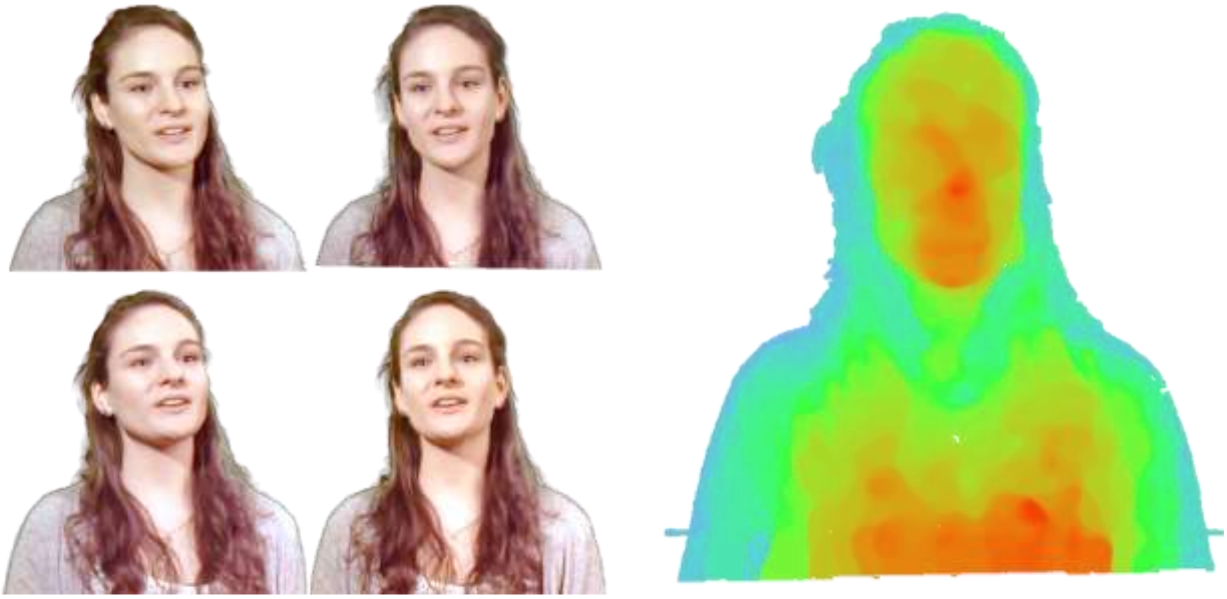

Figure 6 – **Left**: An exemplary 2x2 set of calibrated images. Image points corresponding to the same 3D points are in the same line or column.
Right: A depth-map of the news-presenter in colour representation. Red areas correspond to foreground pixels; blue areas correspond to background elements.

Another interpretation of the view rendering algorithm is as follows: In this case, the window is seen as a 2D canvas element. The colour of the rays constituting the novel view is painted on this canvas element. Then, each point on the canvas element can be identified by a 3D coordinate. The 3D coordinate can easily be projected onto the spectator’s view. In case of stereo images, the picture on the canvas element is of course different for the left and right eye.

Lightfield integration

The combination of the lightfield and the CG content can be performed using NUKE as 3D modelling and rendering software. Figure 7 shows some sample images from our discussed scenario. The final scene is mostly composed of virtual objects like a wooden floor, a desk and lights. Our news presenter is placed behind the desk and in front of a virtual screen. As explained above, the lightfield is integrated as a 2D canvas element. The top row in Figure 7 shows a bird's-eye view of the scene. A pair of cameras is positioned in the middle of the scene pointing towards the newsreader.

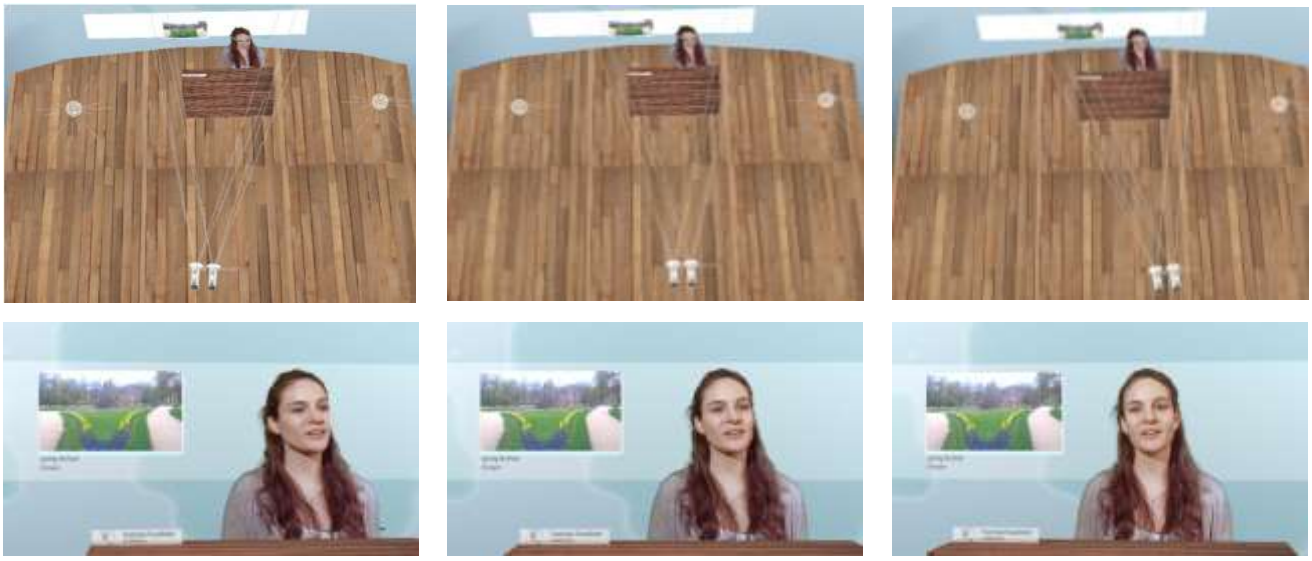

Figure 7 – The top row shows a bird's-eye view of the scene. A stereo camera travels through the scene from left to right. The bottom row shows the 2D output images corresponding to the left camera of the stereo pair. Motion parallax is visible i.e. in the CG desk and also in the face of our news-presenter.

In Figure 7 a stereo camera pair representing the spectator moves through the scene from left to right. The bottom row depicts the corresponding 2D output, as seen from the left camera. Depending on the position of each camera, the respective novel view (containing the news presenter) is rendered in order to provide stereo vision as well as motion-parallax. Looking at the right ear of our news-presenter, the perspective change from the leftmost image to the rightmost image is clearly visible.

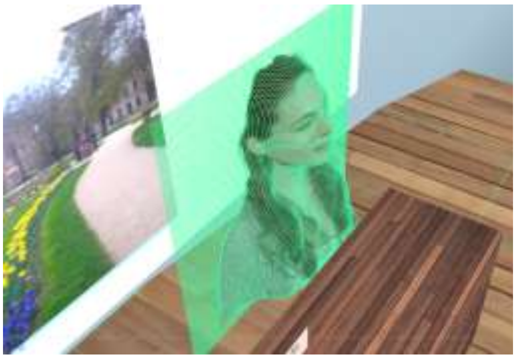

Figure 8 –.The lightfield as seen from an extreme position. Depth from the lightfield is mapped on the canvas element forming an approximate 3D hull. This hull can be used to recover approximate normal maps for correct handling of CG-lighting.

A drawback of integrating lightfields as a 2D canvas element is the lack of 3D information. Although the perspective is correct, the influence of synthetic lights is not handled properly. Optionally, this can be solved approximately by reconstructing a coarse hull of the elements encoded in the lightfield. Figure 8 illustrates such a 3D hull for our news presenter. However, since fine details like hair are not represented properly by this 3D hull, it can lead to visual artefacts. In our scenario, this is not an issue, since lights are not that close to the newsreader.

Unreal Engine integration

An immersive VR experience requires high performance rendering. The Oculus Rift HMD needs 75 fps, while the HTC Vive HMD even requires 90 fps. In both cases, two viewports need to be rendered for the stereo effect. In our scenario, the CG environment as well as the lightfield needs to be rendered at the given rate. For the CG world this is a solved problem as long as overall scene complexity does not exceed the GPU's capabilities.

For the lightfield rendering we implemented a 2D DIBR algorithm using *High Level Shading Language* (HLSL). These shaders are wrapped by a plug-in for the Unreal Engine. When running the VR experience, the Unreal Engine calls the rendering plug-in and provides a set of depth-maps to it. Two images are rendered, one for the left and one for the right eye. The resulting images are mapped onto the 2D canvas element which is placed in the 3D environment.

CONCLUSION

In this work we presented a system and a workflow for the integration of live-action video in VR environments. We use a 3x3 camera array, built from off-the-shelf, yet high quality, components to capture a sparsely sampled lightfield. From our disparity estimation, we reconstruct a dense lightfield using a set of specifically designed NUKE plugins. Once the lightfield is reconstructed, it is integrated in a CG environment, such as a virtual news studio.

In order to drive a classical 2D or 3D pipeline, such a CG environment can be created using NUKE. Alternatively – or in combination – the reconstructed lightfield can be integrated in a gaming engine like Unreal Engine. In both cases, the lightfield adds live-action footage with proper change of perspective, depending on the movement of the camera or player.

The plugin-suite used for lightfield reconstruction is designed to support different types of custom–built camera arrays. In the near future we would like to extend our software towards non-planar camera arrays enabling further promising use cases in the area of VR and also classical productions.

ACKNOWLEDGEMENT

The authors of this paper were funded from the European Union's Horizon 2020 research and innovation program under the Marie Sklodowska-Curie grant agreement No 676401, European Training Network on Full Parallax Imaging and by the Fraunhofer and Max Planck cooperation program within the German pact for research and innovation (PFI).

REFERENCES

1. Anderson, R., et al., *Jump: virtual reality video.* ACM Transactions on Graphics (TOG), 2016. **35**(6): p. 198.
2. Xu, Y., et al., *Camera array calibration for lightfield acquisition.* Frontiers of Computer Science, 2015. **9**(5): p. 691-702.
3. Ziegler, M., et al. *Multi-camera system for depth based visual effects and compositing*. in *Proceedings of the 12th European Conference on Visual Media Production*. 2015. ACM.
4. Levoy, M. and P. Hanrahan. *Lightfield rendering*. in *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. 1996. ACM.