FACULTY OF ENGINEERING AND ARCHITECTURE

Offline Depth Image Based Rendering for Immersive Experiences

Julie Artois
Student number: 01504096

Supervisors: Prof. dr. Peter Lambert, Prof. dr. ir. Glenn Van Wallendael
Counsellors: Niels Van Kets, Ir. Martijn Courteaux

Master's dissertation submitted in order to obtain the academic degree of
Master of Science in Computer Science Engineering

Academic year 2019-2020

GHENT UNIVERSITY

Table of contents

List of Figures

Acronyms

6-DoF six degrees of freedom. 6

AR augmented reality. 1

D dimensional, e.g. three-dimensional (3D). 2

DIBR depth-image-based rendering. 2, 6

FCC face-centred cubic. 37

FOV field of view. 1

GPU graphical processing unit. 3

HCP hexagonal close-packed. 37

Hz Hertz. 2

IBR image-based rendering. 5

LiDAR LIght Detection And Ranging of Laser Imaging Detection And Ranging. 16

MPEG the Moving Pictures Expert Group. 2

PPD pixels per degree. 13

PSNR peak signal-to-noise ratio. 40

ROI region of interest. 18

RVS Reference View Synthesizer. 2

VR virtual reality. 1

XR mixed reality. 1

Chapter 1

Introduction

When going through our daily lives, it might happen that we pass along a memorable event or a story that we want to share with others, and feel the need to capture that moment in time. This need to record, review and share the world around us is what made the camera become an integral part of our society. In the future, we might be able to look at our camera captures in a whole new way.

The more information about the scene that can be captured by a camera, the higher the level of immersion within the scene is achievable from the made image. Examples of this are panorama and 360° videos, which allow a viewer to see a much wider part of the scene that an ordinary camera would. On the other hand, an array of cameras can capture a scene from different positions, allowing the viewer to change where he/she is looking from what point in space.

This master dissertation is situated within a research field that aims to use the light captured by such camera arrays to reconstruct the original scenery digitally, allowing a viewer to freely move around in it. The created immersive experience can, for example, be accommodated through Virtual, Augmented or Mixed Reality (VR, AR, XR). With upcoming improvements to head-mounted displays, e.g. higher resolutions, refresh rates and field of view (FOV), even higher levels of immersion will be possible in the future.

The high specifications of VR present a challenging barrier for systems in this research field to overcome. Additionally, accurately capturing the desired scene is not a trivial task. These challenges and more will be discussed in the problem statement formulated in Section 1.2, but first, the context and constraints of this work are defined. Lastly, some key terms and concepts are explained in detail in Section 1.3.

1.1 Research context and constraints

The master dissertation operates within a certain context and a set of constraints. Figure 1.1 shows the envisioned scenario, where a viewer is wearing a VR headset to be immersed in a given scene. The area of the scene in which the viewer is allowed to move his/her head is predefined and fixed, for example, to the yellow cuboid in Figure 1.1. Within a certain distance around

the area, the displayed scene is completely empty. The viewer cannot interact with the scene. The scene is either static, meaning that all input images were taken in one moment in time, or dynamic, where the scene can change because the input cameras captured these changes over time, e.g. as videos. The scene can be a real-world scene, captured with physical cameras, implying that the achievable camera setups are limited by the physical size of the cameras, or the scene can be created using three-dimensional (3D) graphics. Lastly, it is assumed that the device to which the VR headset is connected is powerful enough to render the VR experience.

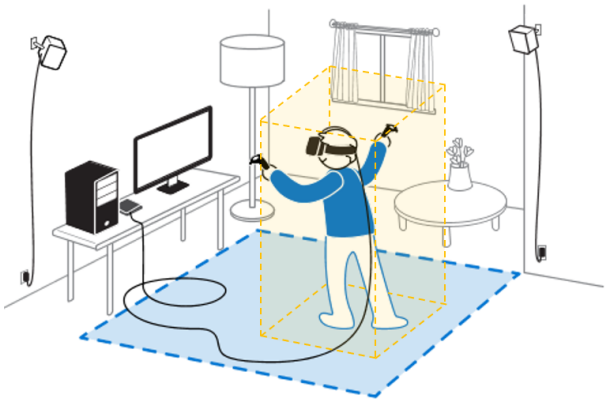


Figure 1.1: The envisioned VR experience, where the viewer movement is restricted to a certain area.

1.2 Problem statement

This master dissertation addresses two general challenges within the research field of rendering new views of a scene based on existing camera captures to achieve an immersive experience. The following paragraphs discuss these problems in detail, while defining the three major contributions made in this work.

1.2.1 Trade-off between quality and performance

The more cameras are placed in the scene, the more information is known and the higher the quality of the scene reconstruction will be. Additionally, the larger the area in which the viewer should be able to freely move around, the more light rays need to be captured and thus the more input cameras are generally necessary. However, there are two downsides to larger camera setups:

- More cameras lead to expensive camera setups if a real-world settings is used.
- The more images need to be processed to render a single output image, the more time this rendering takes. In real-time applications such as VR where the desired refresh rate is a minimum of 90 Hertz (Hz), the time this rendering is allowed to take is strictly limited.

This dissertation focuses on the second problem, i.e. the trade-off between quality (where a larger area of free movement is considered as "a better quality of experience") and performance. To get a general idea of the state-of-the-art achievable performance in a VR context, an implementation by the Moving Pictures Expert Group's (MPEG) called Reference View Synthesizer (RVS) is considered [8] [9]. RVS is a *depth image-based renderer* (DIBR), which means that it

is capable of generating new scene views from existing ones and some geometry information of the scene. Consider the context of a VR application, where two images need to be rendered every 1/90 of a second, one for each per-eye display a with resolution of 1600x1440 pixels and a 110° horizontal FOV. When tested with RVS on a powerful device with an Nvidia RTX 2080 TI Graphical Processing Unit (GPU), only up to four input images could be processed to render each frame in time at a speed of 90 frames per second. So if the number of input cameras goes up to increase the area in which the viewer can freely move around, or to increase the quality, the number of images that need to be processed per frame to get a high-quality result can stack up to much more than four, e.g. 50 or even 100. This implies that the achievable refresh rate can be drastically lower than what is required for VR now.

This dissertation proposes an end-to-end system that makes it so that the size of the area in which the viewer can freely move around is independent of the image quality and performance. On top of this, the proposed system to drastically reduces the number of input images that need to be processed to render a single frame while maintaining high-quality results. Therefore the system is capable of achieving a faster performance than the state-of-the-art, while also allowing more freedom for the viewer and keeping the image quality and realism high.

The end-to-end system leads to these two improvements by introducing a novel offline processing step. This step consists of using DIBR technology to generate the images for a large number of imaginary cameras, strategically placed within the area in which the viewer will have the freedom to move around. Since it is an offline step, no time limits hold and cloud computing can be used. The newly generated images are then added to the original collection of images. This makes it so that during the VR experience, where the real-time requirement does hold, at most four input images need to be processed per frame in time, i.e. the ones of the cameras closest to the head of the viewer and looking in the same direction.

To generate the new images during the new offline step, the system requires an implementation that is capable of delivering high-quality renders of the captured scenes. Since the step is offline, the implementation is not limited by hard time constraints. This work proposes a novel DIBR implementation aimed at maximising the quality when dense camera setups are used.

1.2.2 Challenging scene elements

Some objects consists of a *diffuse* material, meaning that incoming light is refracted in all directions equally with the same colour. In other words, each point on these objects look the same regardless of which angle they are viewed from. DIBR can easily reconstruct scenes with diffuse objects, since it suffices that at least one camera observes each object. On the other hand, points on *view dependent* objects appear differently when looked at from different angles. Such objects are challenging for DIBR to reconstruct if the angle at which the output image is looking at an object differs from the angles at which the input cameras captured it.

Examples of view dependent elements are objects with reflections or highlights, objects or fluids that break incoming light and special effects like lens flares, as illustrated in Figure 1.2. Chapter

3.6.1 gives a more thorough overview of view dependent elements.

Figure 1.2: The left image contains reflective surfaces like a mirror, glass and sink. The curved glass and water in the middle image break incoming light. The right image is an example of a lens flare.

In general real-world settings, many of these challenges will occur. In order to evaluate how well DIBR implementations or other algorithms that process 3D content would perform for realistic scenes, the data sets on which they are tested should reflect the same level of difficulty. However, most current data sets being used for quality testing lack a combination of different kinds of challenges, which leads to an incomplete image of the achievable realism. This work proposes a novel computer generated scene "Temple" that includes a whole spectrum of challenging scene elements, which are discussed in Chapter 3. Data sets generated from the scene can be static or dynamic, i.e. the scene can be fixed in time or it can change over time.

Additionally, in order for the proof-of-concept DIBR implementation to perform well on such an intricate scene, it will take special measures to try to deliver high-quality, realistic results. The camera setup used to capture the scene plays an important role in the achievable quality of the scene reconstruction. Therefore, Chapter 3 covers the properties of a good camera setup.

1.3 Terminology and concepts

Plenoptic function

The *plenoptic function* is a representation of the light that travels through a certain 3D space. According to Levoy et al.[1], the plenoptic function models the radiance L as a function of five parameters $L(x, y, z, \theta, \phi)$, where L follows a ray that goes through point (x, y, z) and has angles (θ, ϕ), as shown in Figure 1.3.

The 5D plenoptic function can be reduced to four dimensions when the radiance along each ray remains constant, which is the case in an empty space. The remaining 4D function is denoted as a *4D light field*. At each moment of time, an eye is capable of perceiving the plenoptic function for a limited range of (x, y, z, θ, ϕ), with other limitations such as maximum intensity per frequency. The same goes for light sensing devices such as cameras. This means that, if the pixels in an image can be associated with light rays characterised by (x, y, z, θ, ϕ), then the plenoptic function can be reconstructed for these light rays with a certain level of precision.

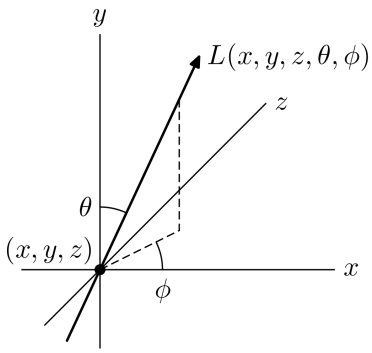


Figure 1.3: The plenoptic function is the radiance along a light ray through point (x, y, z) with angles (θ, ϕ) [1]

Image-based rendering

With *Image-Based Rendering* (IBR), a new image of a scene is generated from other images taken from that same scene. Figure 1.4 illustrates how this image-rendering technique works. One camera samples a small part of the plenoptic function in a given empty part of a scene. If enough cameras are placed in the empty part of the scene, the captured images can be used to make a rough reconstruction of the plenoptic function that would be seen by a new virtual camera. Thus, the image for the virtual camera, in this dissertation denoted as the *virtual image*, is produced.

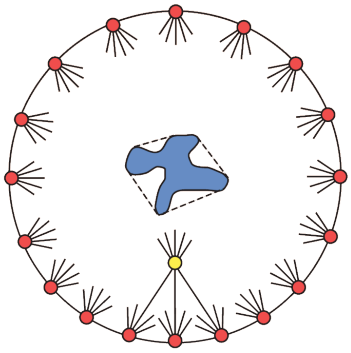


Figure 1.4: Illustration of cameras capturing the light along certain directions in a scene with an object in the middle. The light rays that would be captured by the virtual camera (yellow) can be reconstructed from the other captured rays. The virtual camera should not be placed within the boundary around the object, since this area contains rays that are not captured by any of the cameras [1].

This approach is also called *light field rendering*, where the light field is interpreted to be the collection of captured 2D *input images* [1]. The cameras placed in the empty part of the scene are denoted as the *light field cameras* or *input cameras* in this dissertation.

IBR is different from the traditional image rendering through 3D geometric models in two ways:

- The scenes on which it is applicable are not limited to computer-generated scenes, since real cameras can be placed in real-world settings.
- In real-time applications where rendering of the virtual camera needs to happen at a high

refresh rate, the quality and complexity that is achievable through 3D geometric models is limited. In contrast, the time to generate the virtual image using image-based rendering is independent of the scene complexity, implying that the allowable scene complexity and virtual image quality is often much larger.

Depth image-based rendering

This master dissertation focuses on *Depth-Image-Based Rendering* (DIBR), which is a subclass of IBR techniques that require as input, on top of the input images, one depth map per input image. The depth map contains depth information about each pixel of the corresponding input image. Figure 1.5 shows a camera, its *viewing vector*, which is aligned with the direction the camera is looking in, and its *viewing plane*, which has the viewing vector as normal and goes through the focal point of the camera. Since a pixel originates from a 3D point in the scene captured by the camera, its depth information can be the Euclidean distance from that point to (the focal point of) the camera. Alternatively, it can be the distance to the viewing plane, denoted as the *Z-depth*, since it can be seen as the z-coordinate with respect to the axial system of the camera, where the Z-axis is the same as the viewing vector. Both are illustrated in Figure 1.6.

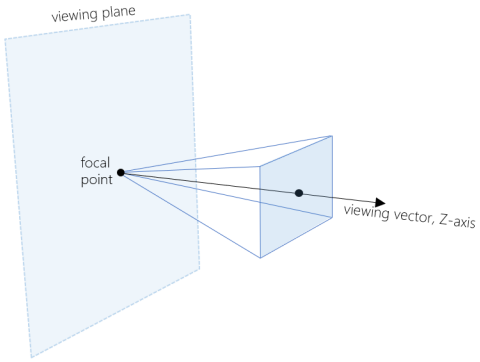


Figure 1.5: Viewing plane and vector.

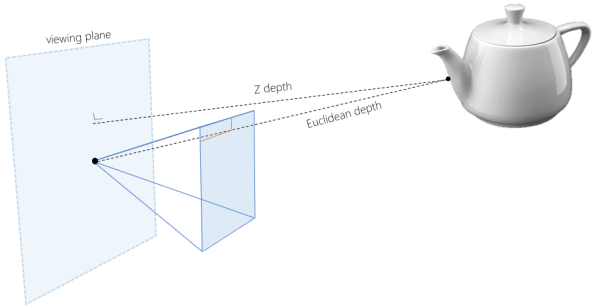


Figure 1.6: Euclidean depth versus Z-depth.

Six degrees of freedom

IBR takes a light field as input and outputs the image for a new, virtual camera. The rotation and position of the virtual camera within the empty space of the scene can be chosen freely. In other words, the virtual camera can move with *six degrees of freedom* (6-DoF). An example where 6-DoF is desired, is in VR, where the viewer can move his/her head freely, with the per-eye displays showing the viewed parts of the scene accordingly. In most applications, however, this freedom of movement will be restricted so that no parts of the scene which were not sufficiently captured by the input cameras become visible.

1.4 Chapters overview

Chapter 2 gives an overview of the state-of-the-art related to this master dissertation.
For the offline DIBR step, an implementation that is capable of generating close-to-truth results is necessary. The camera setup that is used to capture the scene plays an important role in the achievable quality, so Chapter 3 goes in depth into what makes a setup good. Chapter 3 also

covers the three computer-generate scenes, one of which is "Temple", that were created in this work, and the light fields data sets that were captured to evaluate the quality of new DIBR implementations or existing ones. Additionally, Chapter 3 goes over the challenges that DIBR faces regarding certain real-world scenes and which challenging aspects were incorporated in each scene.
Chapter 4 explains how a high-quality, close-to-truth reconstruction using DIBR can be achieved, as well as the workings of the proof-of-concept DIBR implementation that uses these ideas. Chapter 5 then covers the offline extension of the light field step in detail, which mainly consists of deciding where to place the virtual cameras. Chapter 6 contains an evaluation of the proof-of-concept DIBR implementation, using the data sets created in Chapter 3. Lastly, Chapter 7: Discussion and Chapter 8: Conclusion bring the dissertation to a conclusion.

Chapter 2

State-of-the-art

DIBR is a technique that reconstructs a scene from camera and depth sensor captures, so that the reconstruction can be rendered from a new viewpoint [10]. This technique allows for a wide variety of applications within the domain of 3D media. An example implementation is MPEG's RVS [8] [9]. RVS works well when between one and four input images whose cameras are close to the virtual camera are used. This is because RVS blends the pixels of the input images, which can lead to blurry results if too many visually different input images are processed. One disadvantage of RVS is that the area in which the virtual camera can have 6-DoF is limited to a small volume around the input camera setup. Another disadvantage is that RVS has difficulty rendering reflective objects, e.g. a mirror, because each input camera sees a different reflection, which get blended into something far from the groundtruth.

This paper proposes a novel DIBR implementation that avoids these disadvantages by combining two ideas: the amount of blending between input image pixels is strongly reduced and priority is given to input cameras that see objects in the scene from the same angle as the virtual camera, leading to better rendering of reflective objects. The DIBR implementation by Overbeck et al. [7] also uses these ideas. The disadvantage of their implementation is the use of meshes created from the depth maps. The meshes are simplified to achieve real-time rendering, which leads to the use of inaccurate depth information in detailed scene areas. The proposed implementation can use the full resolution of the depth maps since it does not require real-time execution.

Other research groups work on real-time DIBR, but the data sets on which they evaluate their implementations are mostly diffuse. A diffuse object reflects incoming light in all directions uniformly with the same colour, making it easy for DIBR to reconstruct. One example is the paper by Jens Ogniewski [2], which proposes a system that delivers high-quality DIBR at around 90 frames per second. He considers the diffuse data sets shown in Figure 2.1, which originate from the short film Sintel [11]. Another example is the DIBR proposed by Yanzhe Li et al. [3], which runs at 45 frames per second at 1080p. Figure 2.2 shows their used data sets, again consisting of diffuse objects only. The "Temple" scene proposed in this dissertation is a valuable addition to the state-of-the-art data sets of 3D content, since its challenging aspects lead to a better insight in the quality achievable for real-world scenarios.

On the other hand, the systems proposed by Sinha et al. [4] and Lischinski et al. [12] focus on the correct rendering of reflective scene objects through DIBR. Both systems work by separating diffuse and reflective pixels, processing them and recombining the results. Figures 2.3 and 2.4 show their considered data sets. However, the first system uses the input cameras that mere millimetres apart by extracting frames from a video. In contrast, the camera setups considered in this dissertation assume that the input cameras are much farther apart and remain still, making light field video a possibility. The second system only performs well for scenes with simple depth information, making real-world scenarios difficult to render in a performant way. In contrast, the novel DIBR implementation can render diffuse and reflective scene elements within realistic settings.

Within the research field of DIBR, some research groups focus on sparse camera setups and thus have to overcome large parts of the scene being occluded from the input cameras. For example, Guibo Luo et al. propose an inpainting framework that attempts to separate the occluded background from foreground and intelligently restore the background [6]. Figure 2.5 shows three example results of their technique. The camera setup guidelines presented in this dissertation can help minimise the impact of occluded areas for a desired number of input cameras, avoiding the inaccuracies of an inpainter and thus of errors or ghosting and flickering artefacts.

This paper also proposes an end-to-end system that is capable of delivering high-quality rendering in real-time for a high resolution and refresh rate context such as VR. The system by Overbeck et al. delivers high-quality rendering at 90 Hertz and a per-eye resolution of 1080 $\times$ 1200. However, their implementation will not be able to meet the real-time requirements if a larger area of 6-DoF is desired, because more input images will need to be processed per frame. In contrast, the proposed end-to-end system makes the size of the area of 6-DoF independent of the performance of the real-time rendering, allowing the expansion to more challenging 6-DoF area shapes and sizes.

Recently, a system like the proposed end-to-end system and the system by Overbeck et al. was developed by Dinechin and Paljic [5]. In other words, the system uses DIBR to achieve immersive VR experiences. In fact, it uses the same mechanism as the DIBR proposed in this work and as the one of Overbeck et al. to achieve realistic results for non-diffuse scene elements. However, the end-to-end system proposed in this work is designed to have a better performance compared to the mentioned state-of-the-art systems while maintaining quality. On top of that, the DIBR by Dinechin use simplified 3D meshes of the scene, just like Overbeck, which leads to lower quality results than the DIBR proposed in this dissertation, as explained above.

Lastly, the systems by Overbeck and Dinechin are limited in the amount of input images they can process for each frame that needs to be rendered, leading to a trade-off between quality, freedom of movement and performance. The novel end-to-end system is not limited by this trade-off and, as will be seen.

Figure 2.1: Data set used by Jens Ogniewski[2].

Figure 2.2: Data set used by Yanzhe Li et al.[3].

Figure 2.3: Data set used by Sinha et al.[4].

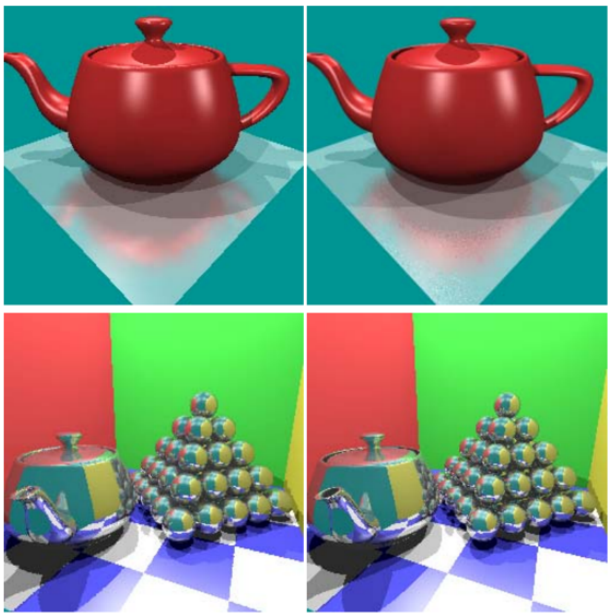


Figure 2.4: Data set used by Lischinski et al.[5].

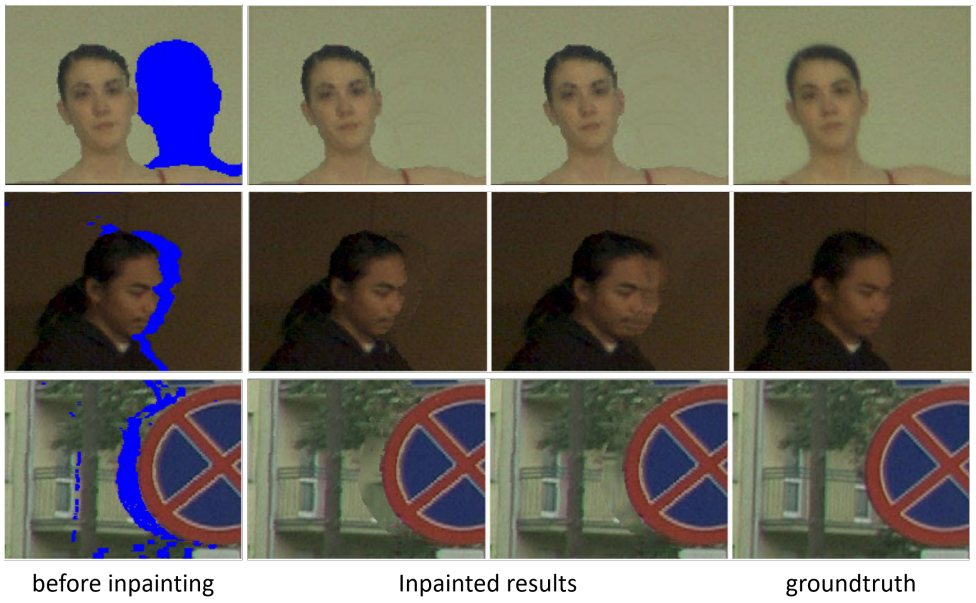


Figure 2.5: Data set used by Guibo Luo et al.[6].

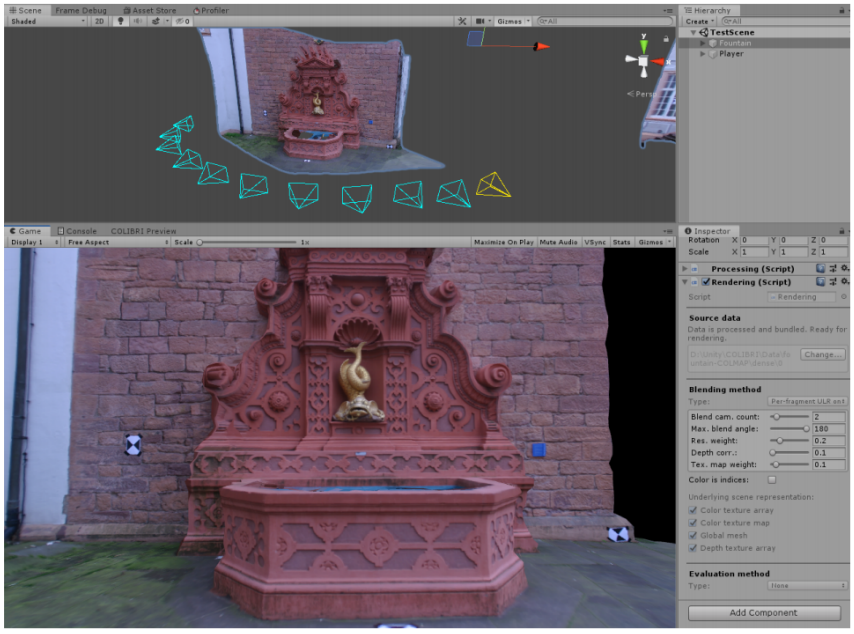


Figure 2.6: Screenshot of the system by Dinechin and Paljic [5].

Chapter 3

Camera setup

In practice, the light ray directions (x, y, z, θ, ϕ) for which the plenoptic function can be measured is limited by for example the physical dimensions of the used cameras, the cost and the stability of the setup. Luckily in most scenes, the redundancy of the plenoptic function is high because the scene contains many objects with material properties that are largely *diffuse*. A diffuse material interacts with light by refracting it in all outward directions with the same colour, resulting in an object on which the points on the surface keep their colour when viewed from different angles. For these materials, it suffices that at least one camera captures its surface. This observation can be exploited to drastically reduce the number of input cameras without losing DIBR result quality.

This chapter investigates the minimal camera placement and intrinsics (resolution and FOV) for scenes that are mostly diffuse. After that, it is reasoned about what needs to change when the scene contains non-diffuse elements, also denoted as *view dependent elements*. The role of the depth maps are discussed, after which three basic camera setups are analysed.

After that, the three scenes that will be used to test the DIBR proof-of-concept proposed in this dissertation will be reviewed in detail. It will be discussed how each scene imposes its own challenge upon the DIBR implementation, making to so that conclusions about the achievable quality in general scenes can be made. Lastly, the camera setup that was used in this work to extract light field data sets from the scenes is explained.

3.1 Best practices for diffuse scenes

First, an area in the scene denoted as the 6-DoF volume needs to be defined, i.e. the volume in which it is desired that the virtual camera has 6-DoF. If IBR is used for a VR experience, this 6-DoF volume corresponds to the area in which the head of the viewer is allowed to move, for example the yellow cuboid from Figure 1.1. The minimum collection of light rays to be captured can be derived from the 6-DoF volume. In other words, for each point on each object in the scene, it can be determined whether or not the point will be visible for a given virtual camera in the 6-DoF volume. If a point is not captured by any of the input cameras, it is denoted as a *hole*. Scene holes can have different causes and solutions, which are discussed as best practices

in the following paragraph.

- The virtual camera should not see any part of the scene that is not observed by any of the input cameras. If this condition is violated, the DIBR has no idea what this part of the scene should look like. In this dissertation, uncaptured parts of the scene are referred to as *large holes*. The best practice can thus be formulated as follows: the input cameras should capture enough of the scene so that no virtual camera in the 6-DoF volume can see any large holes.
- The virtual camera should not see any part of the scene in more detail than any of the input cameras. If this condition is violated, DIBR does not have the required information to fill in all the details, resulting in some pixels of the virtual image for which the real colour is unknown, leading to what is referred to as *small holes* in this dissertation. Scenarios in which the virtual camera might perceive a higher level of detail than any input camera are:
  - The pixels per degrees (PPD) of the virtual camera is higher than some of the input cameras.
  - The level of detail with which a camera captures an object depends on the angle between the surface normal of that object and the viewing angle. Figure 3.1 illustrates this, where a camera looks at a spherical object on the left and a plane on the right. Here, α is chosen to be the fixed angle between two neighbouring pixels. On the right, the x and y are both distances between the points corresponding to two neighbouring pixels. Since the angle between the surface normal and the line from the points to the camera is larger near y than near x, y is larger than x. This implies that the level of detail with which the camera captures the plane is higher near x than near y. As for the spherical object on the left, the surface normal and the line from a point on it to the camera always overlap, so here x and y are equal.

  Special attention needs to be paid to cameras that have a non-uniform PPD across the image that they produce. This is for example the case with some wide-angle lenses such as a fisheye camera, which has a higher PPD in the centre of the images than near the edges.

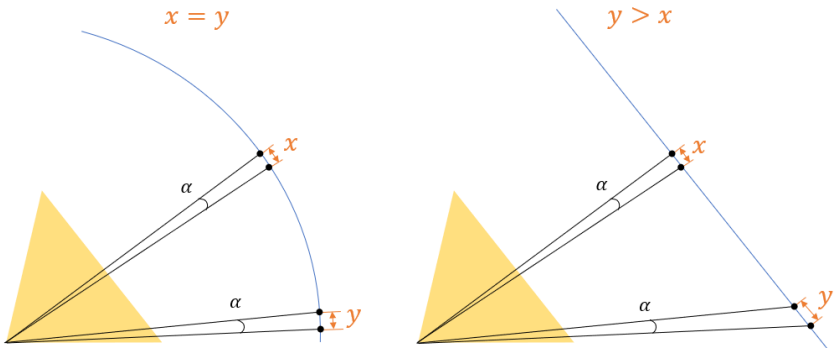


Figure 3.1: Illustration showing that the straighter a camera looks at a surface, the smaller the distance between the two points that are unprojections of two neighbouring pixels is.

Although a good camera setup tries to limit the occurrence of these holes, especially large holes, it is often not possible to eradicate all occurrences. Luckily, if the size of the holes is small or

the real contents of the holes are almost identical to neighbouring pixels, software can be used to fill the holes in such a way that they become unnoticeable. Section 4.3 elaborates on the use of inpainters for this specific task.

The remaining question now is, what are the guidelines for the distance between cameras, their resolution and their FOV? Figure 3.2 shows an example scene with three objects A, B and C, and a desired 6-DoF volume. For simplicity, in this example it is assumed that the virtual camera is always facing the same direction as Camera 2. Assuming that Cameras 2 and 4 were already placed, are more cameras necessary? Since objects B and C block cameras 2 and 4 from viewing object A, resulting in a large hole, Camera 3 should be added. Its FOV should be large enough to cover the large hole and its resolution should be at minimum that of the virtual camera, since in the 6-DoF volume, a virtual camera can be placed right behind it. Cameras 1 and 5 should be added to take care of the small holes on the left of B and on the right of C. For example, if the virtual camera was placed in the top left corner of the 6-DoF volume, the angle between its viewing angle and the surface normal of the left side of B would be smaller when compared to Camera 2 but larger when compared to 1. Again, the FOV of 1 and 5 should be enough to see B and C, and the resolution should be at minimum that of the virtual camera.

It would have also been possible to place the cameras behind (i.e. at the bottom of Figure 3.2) the 6-DoF volume. A camera that is placed behind the volume sees a larger part of the scene, but in less detail, which can lead to small holes. In the VR use case, the virtual camera will have a large resolution and FOV, so unless the input cameras have higher specifications than that, it is best to place them in between the 6-DoF volume and the scene, looking outward to the scene instead of inward at the 6-DoF volume.

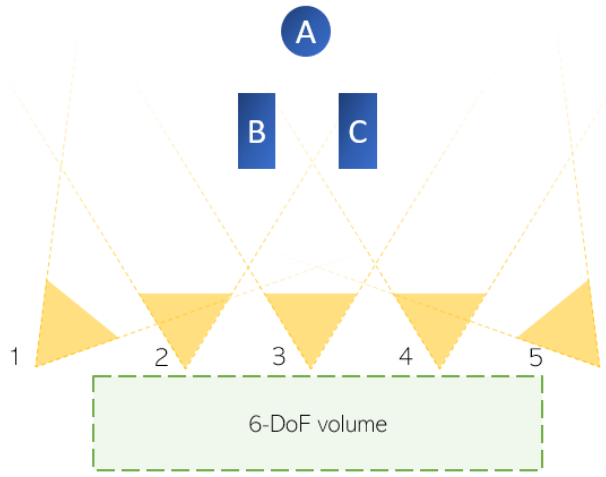


Figure 3.2: An example scene and 6-DoF volume with a good camera setup.

The last thing to note is that there exists a trade-off between small and large FOVs: large FOVs can avoid large holes, but for a fixed resolution, the PPD is lower, so the captured level of detail is lower.

In conclusion, there is no setup that works for every scene, and sometimes, a balance between large FOVs or high PPD needs to be made in order to reduce the visual impact of holes.

3.2 View dependent scene elements

IBR becomes significantly harder when the scene contains elements whose exterior is view-dependent. View-dependent elements are things in the scene that appear differently when looked at from different angles. This happens when a surface does not refract the light that hits it in all directions with a uniform colour. Most real-world objects have some degree of reflectiveness to them. View dependency also occurs when an object breaks/bends the light in a certain way. Section 3.6.1 gives a detailed list of example view dependent elements.

In theory, when dealing with general view dependent scene elements, an infinite amount of cameras needs to be placed in the scene to capture every light ray bouncing off of/going through its surface, otherwise the plenoptic function cannot be reconstructed perfectly.

In practice, such a high number of cameras is infeasible, so it will come down to adding as many cameras as possible and hoping that the quality of the reconstruction of the plenoptic function is adequate for the intended purpose.

Additionally, view dependent scene elements impact the required FOV of the input cameras, as illustrated in Figure 3.3. Object A is seen only by the right input camera, but the angle from which the virtual camera looks at A more closely resembles that of the left camera, because $\alpha < \beta$. So, if the FOV of the left camera would be increased to make the object visible, a better reconstruction of the virtual image would be possible.

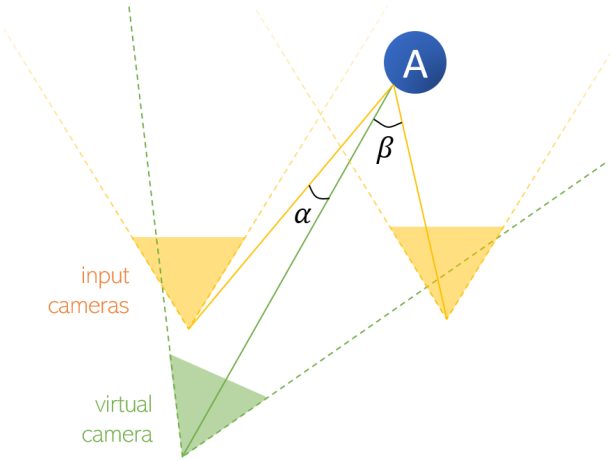


Figure 3.3: An illustration of how increasing the FOV of the input cameras can help improve DIBR of view dependent elements.

3.3 Depth maps or 3D meshes

How do we know which pixels of the input images correspond to which pixels of the virtual image? In a computer-generated scene, the original 3D mesh is known. In that case, it is easy to relate each pixel to a 3D point on the mesh, and thus to get an output image pixel via the corresponding input image pixel. The same logic holds if for each input camera, its depth map is known. A depth map can be generated using a 3D mesh, disparity information, or through a depth sensor, e.g. LIght Detection And Ranging of Laser Imaging Detection And Ranging (LiDAR) technology. Since acquiring and working with depth information is not straightforward, Section 3.6.2 covers these challenging aspects.

3.4 Basic camera setups

This section goes over some basic camera setups and variations. They can be used as building blocks when more complicated 6-DoF volumes are sought.

Plane

In this first setup, the cameras are positioned on a plane, where multiple arrangements are possible, e.g. a grid or a hexagonal pattern, as in Figures 3.4 and 3.5. The advantage of the grid arrangement is that rectangular-shaped cameras can be positioned closest to one another. In the most basic setup, the viewing vectors of all cameras are parallel to the normal of the plane, and they all face the same way. Variations on this setup can be made by rotating individual cameras, or by slightly curving the surface while the viewing vectors stay parallel to the surface normal.

As discussed earlier, the cameras can either all face the empty volume (facing inwards) or look at the other side (facing outwards). This planar setup is good at capturing one part of the scene, i.e. if the cameras stay within certain bounds, the number of large and small holes will be limited. This dissertation focuses on 6-DoF though, and since only one angle of the scene is captured, the next two camera setups extend upon the idea of the plane and the curved surface to capture the whole 360° view.

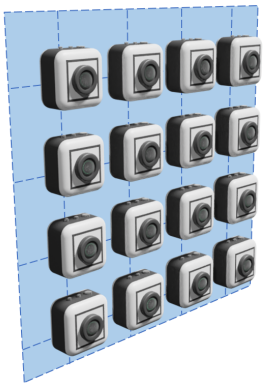


Figure 3.4: Grid pattern.

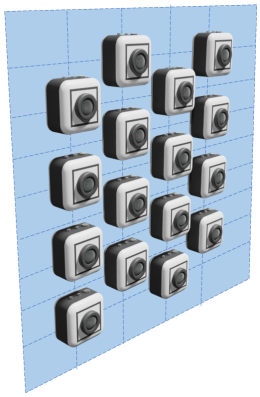


Figure 3.5: Hexagonal pattern.

Cube

Six planes can be combined to make a cube, as shown in Figure 3.6. One variant of this setup puts cameras on the edges and corners, with the rotation of the cameras adjusted to face outward at different angles, for example as shown in Figure 3.7. If this approach is not used, then the FOV of the cameras near the edges will need to be large enough to see enough of the scene. The benefit of this setup is that it is easily scaled in height, width and depth.

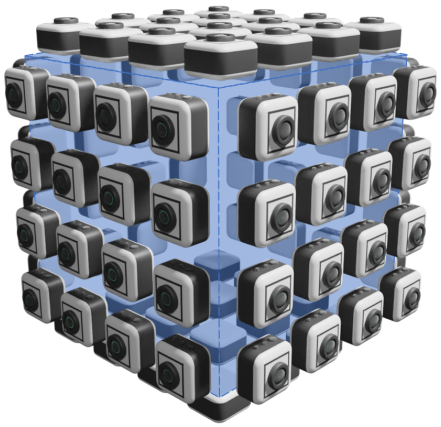


Figure 3.6: Regular cube.

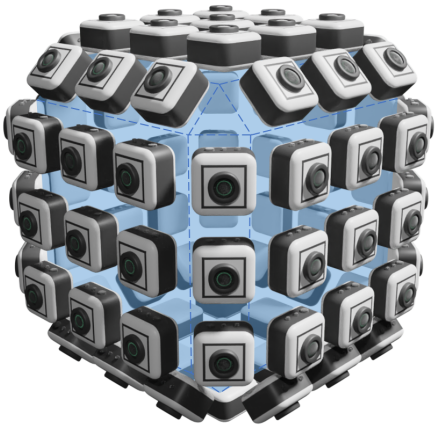


Figure 3.7: Cube with rounded edges.

Sphere

The cameras can be placed on the surface of a sphere, facing inward or outward, aligned with the surface normals. On Image 3.9, the cameras are placed at the vertices of a geodesic polyhedron (*icosphere* in Blender), which allows for a fairly uniform distribution, i.e. the distance to neighbour cameras is approximately the same everywhere. Image 3.8 shows the cameras being placed on the vertices of a *UV sphere*, where the number of cameras is reduced near the poles to avoid overlap. A sphere can be scaled up, of course, but if a stretched-out ellipsoid is not desirable, a cylindrical configuration (horizontal or vertical) can be considered.

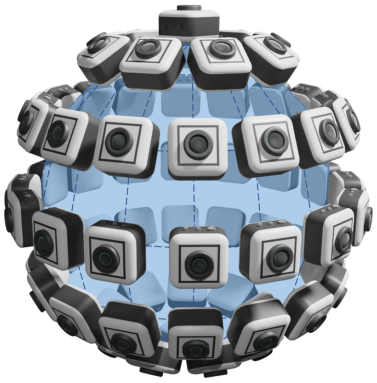


Figure 3.8: UV sphere.

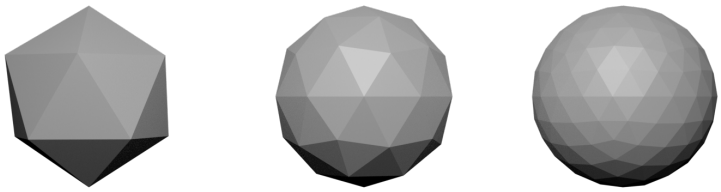


Figure 3.9: Icospheres with increasing number of subdivisions.

3.5 Scenes used for evaluation

There are many real-world phenomena that are challenging for DIBR to accurately reconstruct. In the following Sections, a (non-exhaustive) list of such challenging elements is given. Two main categories can be considered: view dependent elements and depth related challenges.

Based on this list, during the dissertation a scene "Temple" was made in the 3D-graphics tool Blender (version 2.80). The goal was to create a realistic scene of which challenging data sets could be rendered that would be used for evaluation of 3D-related algorithms. Its value as a data set source comes from combining a large amount of diverse challenging scene elements, as well as providing dynamic data sets (in which the scene changes over time).

In order to evaluate the DIBR proof-of-concept in Chapter 6, the dissertation uses "Temple", as well as two others scenes referred to as "Regular classroom" and "Mirror of classroom". The first one is unique in the sense that it consists mainly of diffuse objects that are relatively far away from the 6-DoF volume that will be defined for it. The second one is the same as the first, but with a large mirror in the middle of the classroom. The mirror forms the ultimate view dependent element. Together, these scenes cover the whole spectrum of difficulty for DIBR to accurately construct. In other words, the "Regular classroom" is almost trivial to render, "Temple" has many challenging but also some easy elements in it, while the "Mirror of classroom" consists of one challenging element.

The following Sections give an more detailed overview of the scenes. The last section discusses the camera setup that was used to create the data sets from the scenes for the evaluation in Chapter 6.

3.6 Temple

The Temple scene was created in Blender version 2.80. The goal of the scene is to generate data sets with a high level of realism, so that not only DIBR implementations but also other algorithms that process 3D content can be evaluated. Currently, there is an overall shortage of 3D content, especially of particularly challenging scenes. The scene contains a large variety of materials, different layers of depth and highly detailed areas. Additionally, the scene changes over time, which can be for example useful to evaluate compression or Region Of Interest (ROI) related algorithms.

Blender itself is a useful tool for generating data sets, since it can not only render colour images using ray tracing, but it can also produce depth maps, normal maps and maps that separate diffuse colour (albedo) and highlight colours.

Figure 3.10 shows parts of the model for the "Temple" scene, accompanied by three renders to illustrate the textures and materials. The next paragraphs cover a list of challenges related to DIBR. For most of the list items, the scene contains multiple occurrences, as illustrated by

Figure 3.11.

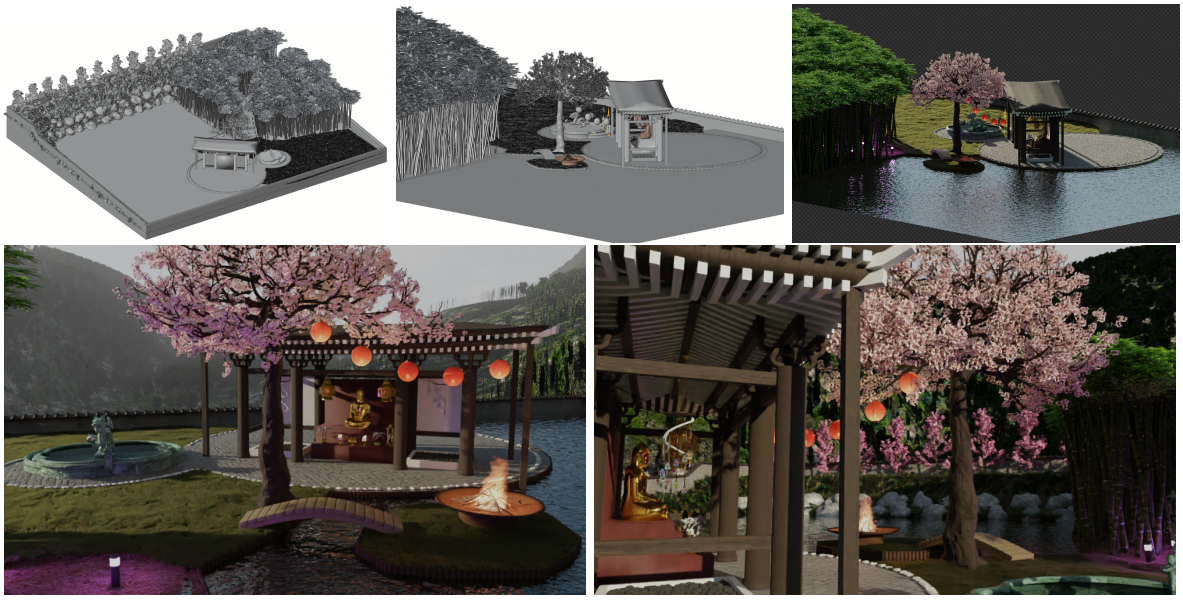


Figure 3.10: An illustration of parts of the model of "Temple", with three renders.

3.6.1 View-dependency-related challenges

- Breaking of light happens when the light enters a medium with a different density, e.g. fluids like water, hot air, translucent materials like glass and plastic, ...
- The Fresnel effect is the effect where a reflection on an object becomes more prevalent the flatter the viewing angle is. An example can be found on the body of water of the scene. When looking down into the water, the viewer can see the pebbles at the bottom of the pond. The water further away appears more reflective.
- A lens flare is an effect that can occur when a camera looks at a bright light source directly. Effects like god rays can also be considered in this category.
- There exist different types and degrees of reflections:
  - Most real-world objects show some degree of highlights, so this is also the case for most objects in the scene. In case of a perfect mirror, the incoming light rays always get reflected along their angle of incidence. Other materials reflect light rays non-uniformly, preferring some angles because of anisotropy. An example of this can be seen on the back of a cooking pan.
  - Highlights on metallic objects are a special case, since these highlights receive a slight colour tint dependent on the metallic composition, e.g. a golden tint for gold.

3.6.2 Depth-related challenges

In general, the accuracy of the depth maps plays a deciding role in the quality that can be obtained by a DIBR implementation. Even with high precision and resolution however, there exist plenty of challenges related to gathering adequate depth information.

- Near edges, the depth suddenly leaps from what is in the back to what is in front. Due to inaccuracies, it is possible that pixels in the front get assigned the depth of the object behind it, or vice versa. Since this depth error can be quite large around edges, it may become clearly noticeable.
- Partly see-through objects are hard to deal with: neither the depth of the object itself, nor the depth of what is behind it suffice on their own. Such surfaces can be seen as view dependent elements if the depth of the object itself is used. The scene contains partly see-through elements like water, fire, paper lanterns and glass. For the "Temple" scene, the depth of the object itself is used since in real-world scenarios, this is likely the only depth that can be easily measured.
- Physics simulation for water, fire, smoke and other fluids are unlikely to have an adequate depth because of their complicated shape and because they are partly see-through. For the "Temple" scene, the water is not simulated. It is simply a flat plane on which a trick is used to make it look like realistic waves. Therefore, the depth is that of a flat plane. The fire is simulated and therefore does not appear on the depth map, i.e. the depth sensor sees through it.
- When the range of values the depth can take is limited, for example by a lower and upper boundary, there should not be any scene elements outside of this range in order to have a complete depth map. For a computer generated scene, this is no problem, but for other situations the maximum depth limits the scenes that can be captured.

3.7 Regular classroom

The "Regular classroom" scene consists mostly out of diffuse objects instead out of view dependent elements. The objects are relatively far-away from the center of the room, where the camera setup will be placed, as indicated by an orange camera on Figure 3.12. the scene contains highly detailed textures, e.g. on the floor. When testing the DIBR implementation on this scene, conclusions can be made about how well it performs with details and diffuse elements.

3.8 Mirror of classroom

"Mirror of classroom" uses the same 3D model as the previous scene, but a mirror is added in the centre of the classroom. In Figure 3.13, the mirror is clearly visible inside the yellow frame. During the evaluation in Chapter 6, only the mirror itself will be used to evaluate the DIBR implementation, since the rest of the scene is the same as "Regular classroom". This isolation of a view dependent element serves to test how the DIBR performs on one of the most challenging types of view dependent elements.

3.9 Camera setup for evaluation

The camera setup to generate the light field data sets upon which the DIBR proof-of-concept can be evaluated is as follows. The icosphere camera setup discussed in Section 3.4 is used.

Figure 3.11: An illustration of the challenging elements combined within the "Temple" scene.

The cameras were placed on the vertices of an icosphere, rotated so that their viewing vector is aligned with the normal of the vertex they are on, facing outward. The icosphere setup is preferred because the cameras are approximately uniformly spread across a spherical surface, capturing each direction of the scene with a minimum amount of cameras. A second incentive to use a spherical setup is because Overbeck et al. use a spherical one for their system [7]. This creates the possibility to compare results and findings between the two DIBRs.

The radius of the icosphere is 40 centimetres (cm). The icosphere has 162 vertices, where a vertex has an average distance of 12 cm to its five or six neighbours. The radius is the same as chosen by Overbeck et al., but their distance between neighbouring cameras lies around 3 cm horizontally and 7 cm vertically, which leads to the expectation that they will be able to achieve better quality results. For the data sets in this dissertation, each camera has a resolution of

Figure 3.12: An overview of the "Regular classroom" scene's (textured) 3D model and two renders.

Figure 3.13: An overview of the "Mirror of classroom" scene's (textured) 3D model and two renders.

1024×1024 pixels, with a FOV of 110° and a 36 millimetre sensor width.

Figure 3.14 shows the top view within the three scenes, as well as a yellow circle and triangle.

The circle indicates the position of the icosphere used to generate the data sets. Every scene has a main ROI, which is in the direction indicated by the camera in the middle of the icosphere, i.e. the yellow triangle. For example, the ROI for "Mirror of classroom" is the mirror, while for "Temple" it is the buddha shrine.

3.10 Conclusion

How well a DIBR is able to reconstruct a scene from within a predifined 6-DoF depends on whether or not the camera setup succeeds in capturing the scene information necessary for the reconstruction. This Chapter formulated what it means to measure sufficient information as a list of best practices. It was found that the requirements for the camera setup depend on the geometry of the scene and the materials of the objects in the scene. Therefore, no camera setup works for every scene and finding a favorable setup requires doing many calculations.

The plane setup allows to capture one direction of the scene well, while the cuboid and spherical setups make it possible to reconstruct the scene in every direction. In general, the icosphere setup uses less input cameras than cuboid or other spherical setups, making it the preferred option in this work.

Lastly, the three proposed scenes tackle a wide range of difficulty for the DIBR to reconstruct, allowing for a better analysis of the achievable quality.

Figure 3.14: A top view of the three scenes ("Temple", "Regular classroom" and "Mirror of classroom" respectively) indicating the position of the icosphere and the default rotation of the virtual camera.

Chapter 4

Offline DIBR implementation

This Section explains in depth the proposed DIBR implementation that can be used for the offline extension of the light field. The synthesizer focuses on producing high quality, i.e. close to groundtruth results, while no hard time limits are set. First, the expected input and output for the DIBR is discussed, then a basic DIBR implementation is explained. The last sections elaborate on how to improve the basic implementation in order to enhance the quality of the result.

The basic DIBR implementation and the inpainter of Section 4.3 are well-defined aspects in this research field, meaning that the techniques used by the proof-of-concept are not new. For example, RVS was built upon the same unproject-project and inpainting steps that will be discussed below. The purpose of the basic implementation is to build an efficient and effective foundation on which improvements can be made. The rounding algorithm of Section 4.2.3 used by the proposed proof-of-concept is novel, though.

The pre-processing edges step of Section 4.4 is novel, while the disk-based blending approach of Section 4.6 originates from the paper by Overbeck et al., as will be explained.

4.1 Input and output

The synthesizer takes as input a light field, i.e. a collection of images, a set of parameters related to the cameras that captured these images, and a depth map per image. The depth map will be used to calculate for each pixel the corresponding point in 3D space. In this chapter, it is assumed that the depth map and images have the same resolution, but it is possible to have a higher resolution, so that sub-pixel depth information is present, or a lower one, so that neighbour pixels share the same depth information.

The set of parameters for the cameras consists of:

- the camera intrinsics: focal length in width (f_x) and height (f_y), principal point in width (p_x) and height (p_y) and distortion coefficients, all in pixel units,
- the camera extrinsics: position and rotation with respect to a predefined global axial system

in the scene,

- its resolution and FOV in width and height,
- the depth range, i.e. the minimum and maximum values for the depth information, for example in metres, that is present in the depth maps. If a pixel's real depth information falls outside of this range, it can either be interpreted as being on the closest depth range limit, or its depth can be set to zero or infinitely far away.

The synthesizer outputs an image and its depth map, for a virtual camera with desired parameters.

4.2 Basic DIBR implementation

This Section explains the basic implementation, after which the next sections address some improvements to it. Every DIBR implementation contains a *3D warping* step, where pixels from the input images are associated to 3D points in the scene, which can again be linked to pixels in the virtual image. This general 3D warping step is therefore adopted in this basic DIBR implementation and split up into an unprojection, project and rounding step.

4.2.1 Unproject pixels to points

Only images that see parts of the scene that will also be visible for the virtual camera are used. Each image is then *unprojected* into 3D space, i.e. for each pixel, its coordinates in 3D space with respect to the global axial system of the scene is calculated. For this, the depth maps and camera parameters mentioned in the previous section are necessary. This step can be interpreted as reconstructing the original scene as a coloured point cloud. Equation 4.1 shows how to calculate the 3D coordinates (x, y, z) with respect to the global right-handed axial system XYZ of a pixel on location $(row, column)$ with Z depth z_{depth} from the depth map.

$$\vec{R}\begin{pmatrix}(col - p_y)z_{depth}/f_y\\ -(row - p_x)z_{depth}/f_x\\ -z_{depth}\end{pmatrix} + \vec{t} = \begin{pmatrix}x\\ y\\ z\end{pmatrix} \tag{4.1}$$

Parameters f_x, f_y, p_x, p_y belong to the input camera, $\vec{R}$ is its 3x3 rotation matrix and $\vec{t}$ is its translation with respect to the origin of XYZ. In this case, it is assumed that the rotation and translation of the camera are zero if the camera is placed in the origin of XYZ and the viewing vector follows the negative Z-axis. The top left corner of an image is chosen to have $(row, col) = (0, 0)$. Figure 4.1 contains a camera with rotation and translation zero, which shows that the columns follow the X-axis and the rows the negative Y-axis.

4.2.2 Project points to pixels

To reconstruct the virtual image, the inverse operation is done: the coloured 3D points are projected onto the pixels of the virtual image. Equation 4.2 shows the performed calculations,

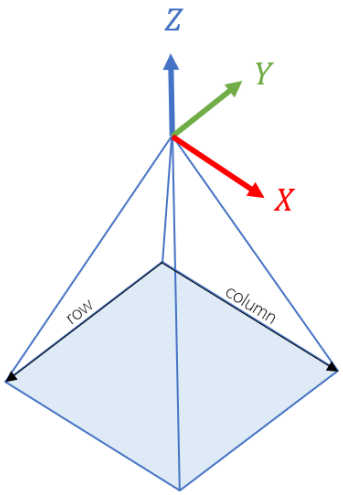


Figure 4.1: A camera with default rotation and translation.

this time with the parameters of the virtual camera.

$$(\begin{pmatrix} x \\ y \\ z \end{pmatrix} - \vec{t})\vec{R}^{-1} = \begin{pmatrix} a \\ b \\ c \end{pmatrix} \qquad \begin{cases} col = p_x + f_x b/c \\ row = p_y + f_y a/c \\ z_{depth} = -c \end{cases} \tag{4.2}$$

The result is an image and its depth map of which most pixels were assigned a colour and depth information. The remaining pixels for which no corresponding 3D points were captured make up the holes. Section 4.3 explains how inpainting can be used to fill these holes in such a way that the final result is close to the groundtruth.

4.2.3 Rounding to pixels

In Equation 4.2, the values of the row and column (row, col) of the pixel to which the 3D point is projected are continuous values, so these need to be rounded to discrete rows and columns. There two basic approaches. The first one rounds to the closest (row, column) pair, i.e. $([row], [col])$. The second one rounds to the four closest (row, column) pairs, which means the pixels with coordinates $(\lceil row \rceil, \lceil col \rceil), (\lfloor row \rfloor, \lceil col \rceil), (\lceil row \rceil, \lfloor col \rfloor)$ and $(\lfloor row \rfloor, \lfloor col \rfloor)$. The advantage of the first method over the second one is that it can sometimes lead to less blurred-out areas. However, with the first approach it is possible that large patterns of small holes appear. Figure 4.2 illustrates this, where the left image is the input image, the middle one is a virtual image generated through the first approach and the right one through the second approach. The bright red pixels represent holes, making it easy to spot the pattern of small holes in the middle figure. These patterns appear because even though there are points that are projected (through Equation 4.2) onto a continuous (row, column) pair close to these pixels, all these pairs where rounded to other neighbouring pixels by coincidence. The image on the right shows the same virtual image but with the use of the second approach. Figure 6.5 also shows the same effect.
This paper proposes a hybrid rounding approach: if the four surrounding pixels do not have a colour yet, they are coloured according to the newly projected pixel. Additionally, the closest pixel is always given the newest colour, even if this overwrites a previously assigned one. This

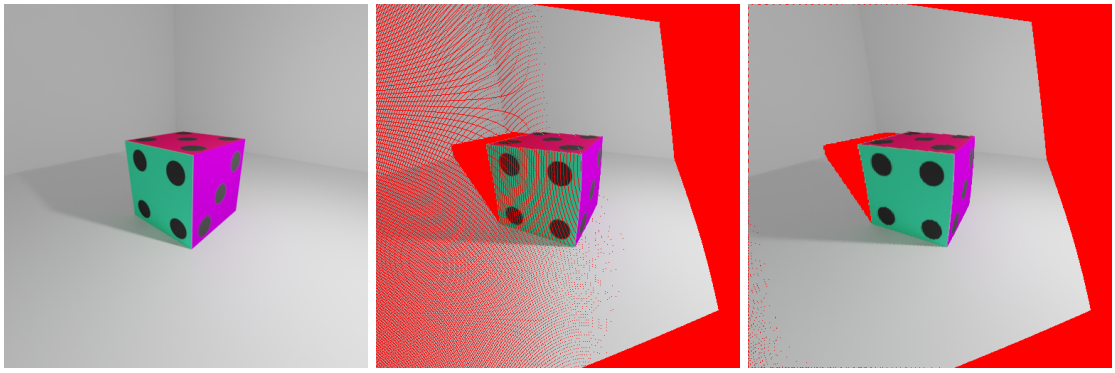


Figure 4.2: On the left is the input image on which DIBR was applied to produce the two virtual images on the right. The left and right virtual image were made using the first and second rounding approach respectively.

method results in the highest quality for the proof-of-concept because it gives priority to closer pixels, but by also colouring the other three neighbours, the number of holes is reduced.

Handling projection of multiple points onto the same pixel

It will often occur that multiple coloured 3D points are projected onto the same pixel, something that will happen even more if approach one for rounding pixels is not used. Again, different approaches can lead to good results. The pixel can take the colour of the point with the lowest depth information to avoid that background objects are suddenly appearing in front of foreground objects. If multiple points have a depth that is very similar to the one with the lowest depth, a weighted average of their colours can be used as the final colour for the pixel. A more advanced technique could be to give higher weights to certain cameras. An example where this might be beneficial is when certain cameras see certain areas with a higher level of detail, e.g. because they are closer or have a higher PPD. It is important to note that points that belong to view dependent scene elements can take diverse colours depending on the viewing angle. In this case, using the mentioned approaches will produce a result that is far from the groundtruth. Section 4.6 addresses the problem of view dependency in scenes.

4.3 Inpainting

When the virtual image contains small or large holes, inpainting software can be used as a post-processing step to colour these holes using information from neighbouring pixels. Implementations with different levels of complexity exist, from copying neighbouring pixels to using machine learning [13]. For the offline generation step, no hard time limits are imposed, so more advanced non-real-time inpainters can be considered. Even though inpainters can sometimes deliver good results, in general they cannot guarantee a close-to-truth result when many large and small holes are to be filled. Luckily, a good camera setup, as discussed in Chapter 3, and the DIBR can be chosen to reduce occurrence of large and small holes.

4.4 Pre-processing edges

A common problem in DIBR are depth map measurement errors around edges of objects, i.e. near groups of pixels that have a high depth difference. Around the edge of an object in an input image it can happen that a pixel has the colour of the foreground object but has the depth of what is behind it. The inverse is also possible: a pixel that has the colour of the background but the depth of the foreground object. In general, the latter case is preferred, since the background is more likely to have a uniform colour.

Figure 4.3 illustrates the problem. The images each show a virtual image of a white lamp. The red area behind the lamp is a hole, since the used input camera did not capture this area behind the lamp. The artefact consists of the edge of white pixels around the hole. It is created because the input image thinks that these pixels belong to the lamp, i.e. they are white, but the depth map believes they belong to the ceiling behind the lamp.

This dissertation proposes a novel pre-processing step that takes place before the unprojection step. In this new step, the colour images are modified as follows: for each group of neighbouring pixels, it is checked if there is an edge, i.e. a big depth gap. If so, it is checked whether or not the foreground pixels' colours are more similar to those of other neighbouring foreground pixels than to those of the background. If this is not the case, then the depth or the colour is likely incorrect. So, the colour of these incorrect foreground pixels is changed to the background colour. The same check is done for the background pixels. On the left of Figure 4.3, the virtual image if the DIBR implementation uses the pre-processing step is displayed.

Alternatively, the depth map could be changed in the exact same way, but with depth information instead of colours. Since changing the colour images is slightly more efficient and gives the same result, that approach is preferred over changing the depth maps.

Figure 4.3: An example virtual image created without pre-processing step (left) versus with the pre-processing step (right).

4.5 Weighted sum of points

Camera setups with many densely placed cameras that have a high resolution and FOV result in massive points clouds when unprojecting. In the projection step, it is often the case that many coloured 3D points are projected onto the same pixel of the virtual image. The final colour for the pixel can be calculated as a weighted sum of these colours. This section covers four factors to be considered when assigning a weight to a point. The next section will then cover an example DIBR implementation that takes some of these factors into account to allocate the weights.

1. As mentioned in Section 4.2.3, the quality of the end result can be increased by giving priority to certain points. There, it was already said that points with lower depth should be given priority over those with higher depth.
2. A pixel whose position in 3D space or colour is somehow believed to be more accurate than the other points can be given priority. This low precision can happen with wide-angle cameras, depth sensors, or during lossy compression, ... If a form of inaccuracy cannot be avoided, a possibility is to pre-process the light field images and depth maps with an advanced algorithm that would be capable of intelligently correcting sensor errors. Section 4.4 already covered an example of such a pre-processing step that can help improve accuracy.
3. Points that correspond to the camera that sees them with the highest level of detail are preferred. Section 3.1 already covered which factors play a role in capturing details.
4. When the area contains view dependent elements, points which lead to the smallest angle $\widehat{IPV}$, (I = input camera, P = 3D point, V = virtual camera) are preferred.

4.6 Disk-based blending

The proof-of-concept uses the *disk-based blending* technique proposed by Overbeck et al. [7]. This is because their approach is a simple and efficient implementation to assign larger weights to pixels with smaller $\widehat{IPV}$ angles. The technique works as follows. On the viewing plane, an imaginary disk with its centre on the input camera centre and a certain disk radius is defined, as shown in Figure 4.4. The disk can be seen as a window through which the virtual camera is allowed to see the scene as seen by that input camera. In other words, per input image, only the unprojected 3D points that are visible by the virtual camera through that disk-shaped window are used to construct the virtual image, i.e. have a weight higher than zero. This is illustrated on the left side of Figure 4.5, while the two right-side Figures show variations of the camera positions. Figure 4.6 from the paper of Overbeck et al. shows the disks in a spherical camera setup (the second setup from Section 3.4) in image (a) and (b).

What remains is to determine the weights of the points that are visible through the disk. Figure 4.4 shows point P that corresponds to a pixel p of input camera I. Point P' is the intersection of the line between the virtual camera V and P, and the viewing plane of I. The weight of pixel

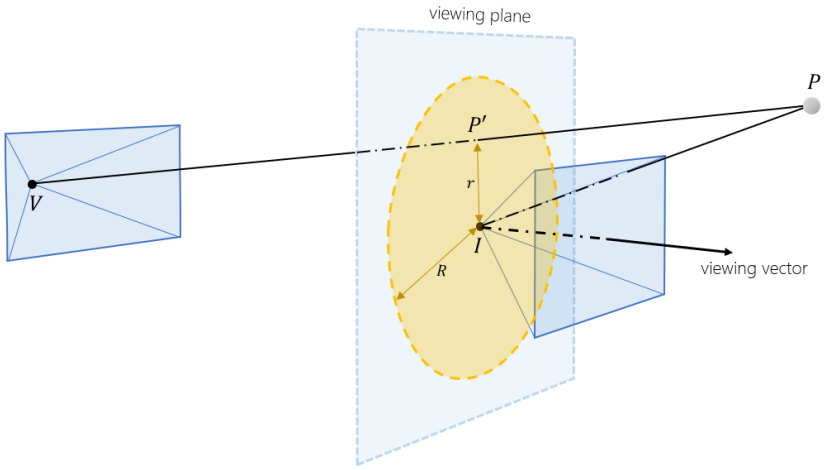


Figure 4.4: The disk lies in the viewing plane and is centred around the input camera.

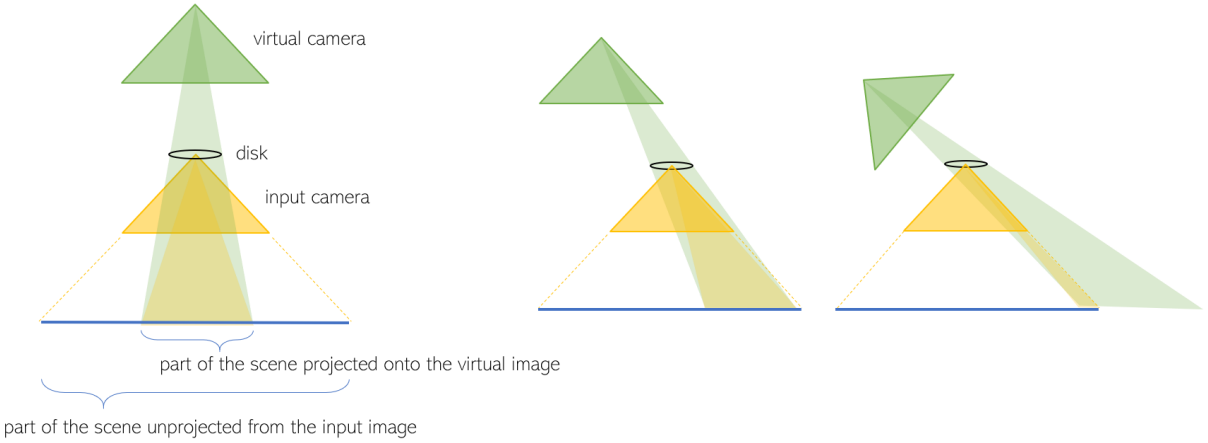


Figure 4.5: The virtual cameras only see the part of the scene unprojected from the input image through the disk-shaped window.

p can then be calculated through Equation 4.3 or 4.4, which are denoted as the *linear fall-off function* and the *gaussian fall-off function* respectively. Both functions are 1.0 if P' is in the middle of the disk and 0.0 if P' is outside of the disk with radius R, as shown in Figure 4.7.

$$linear\_weight(P) = max(1 - \frac{|IP'|}{R}, 0) \tag{4.3}$$

$$gaussian\_weight(P) = \begin{cases} e^{-\frac{|IP'|^2}{2(\frac{R}{3})^2}} & |\mathrm{IP'}| \leq \mathrm{R} \\ 0 & |\mathrm{IP'}| > \mathrm{R}, \end{cases} \tag{4.4}$$

The technique to blend input pixels through a linear function of $|IP'|$ originates from the paper by Overbeck et al. The proof-of-concept differs from their implementation by using the gaussian function rather than the linear one and using less large, symmetrical circles in the viewing plane. It was found that these parameters resulted in higher quality images for the used camera setup, as will be discussed in Section 6.1.

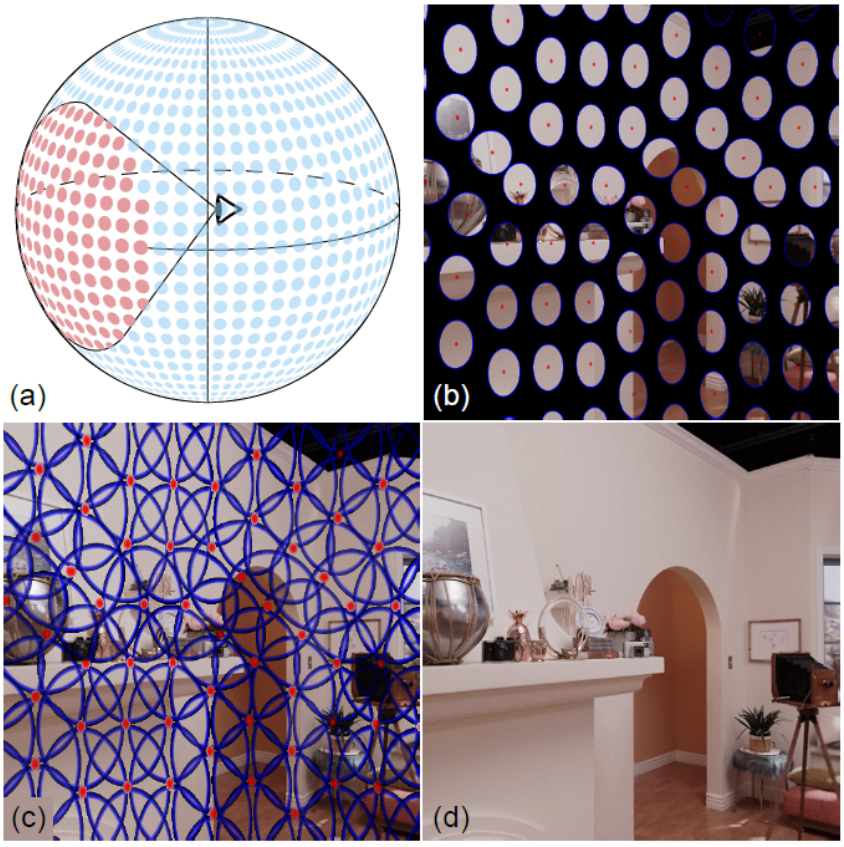


Figure 4.6: An illustration of the positioning of the disks across the spherical camera setup used by Overbeck et al. [7] Image (b) is the same as (c) but with smaller disk dimensions.

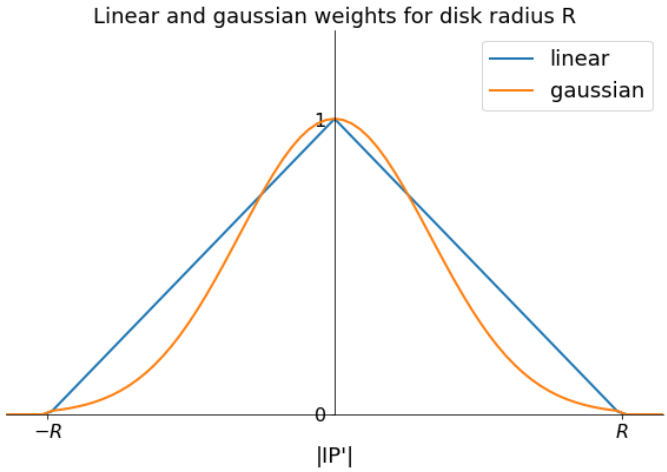


Figure 4.7: The linear and gaussian fall-off function plotted together.

To conclude, if multiple coloured points are projected onto the same pixel, the colour of this pixel is the normalised weighted sum of these colours, where the weights are as before, but the weighted sum is normalised, meaning that it is divided by the sum of the weights. The normalisation makes it so the weights in the weighted average always sum up to 1.

This disk-based approach gives priority to smaller $\widehat{IPV}$, (I = input camera, P = 3D point, V = virtual camera) angles, although it is not the same as only allowing angles lower than a certain threshold. It does, however, mean that smaller disk dimensions lead to only using smaller angles. Additionally, this approach somewhat neglects the third factor from Section 4.5, but by picking scenes that have a spherical layout as shown on the left side of Figure 3.1 (which also stimulates the user to rotate their head instead of moving large distances), and by limiting the 6-DoF volume, areas of potential lower quality are avoided.

4.7 Shrinking the disks shrinks the 6-DoF volume

A disadvantage of small disk radii in the disk-based approach from the previous Section is that they can cause parts of the scene to not be visible through any disk/window for certain virtual camera positions. This is illustrated in Figure 4.8, where the input cameras and their disk-shaped windows are placed in a circle, facing outwards. On the right image, it is indicated that there will be a gap on the virtual image. This is because the virtual camera cannot see this gap through input camera A, since A's FOV is not wide enough, and also not through B, since B's disk radius is too small.

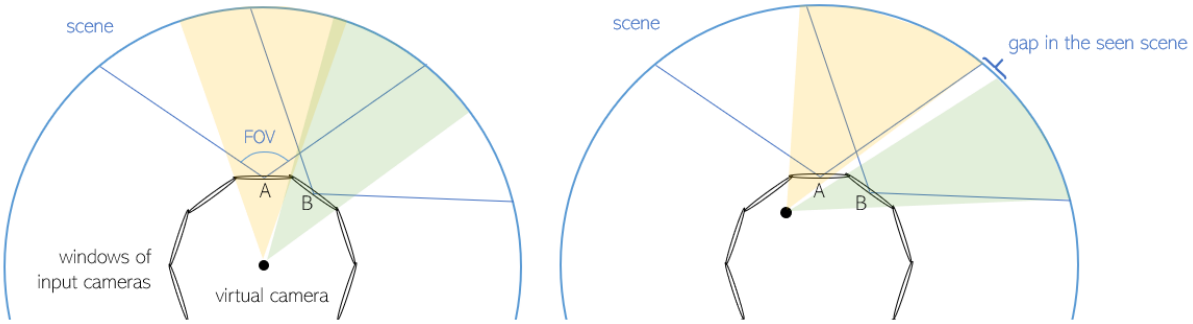


Figure 4.8: Illustration of small disk radii causing a gap in what is seen by the virtual camera.

Figure 4.9 demonstrates that there will be a gap in the virtual image if the virtual camera is in the red zone. The red zone is determined by drawing a line through point $A_{maxright}$, which is the right-most point seen by camera A, and the left-most edge of B. On the right, all such lines are drawn, which results in a volume in the middle where there will be no gaps no matter the rotation of the virtual camera, which is indicated as the 6-DoF volume. Outside of this volume, there might be gaps in the virtual image, but this depends on the rotation of the camera. In the examples above, it was assumed that the scene was perfectly spherical. However, for general scenes, the position of points like $A_{maxright}$ will change and so will the 6-DoF volume.

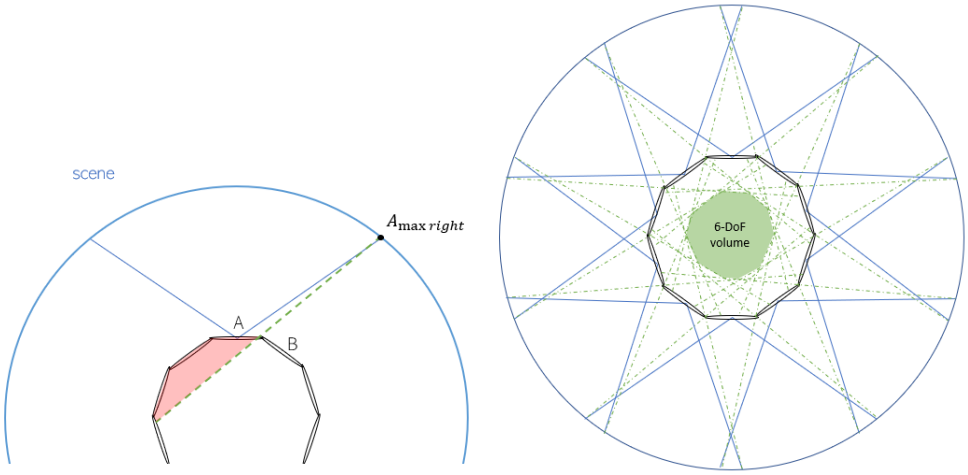


Figure 4.9: Illustration of how to determine the 6-DoF volume.

4.8 Conclusion

The pipeline of the proof-of-concept DIBR implementation is as follows:

1. Pre-process edges
2. Unproject pixels to 3D points and project the points back to pixels of the virtual image.
3. Use disk-based blending to form the final virtual image from an average weight of the input image pixels.
4. Inpaint holes.

It can be concluded that the unproject-project approach is capable of accurately associating pixels on the input images with those of the virtual image, for example as demonstrated by Figure 4.2. The novel rounding approach is capable of reducing the number of holes that would otherwise require inpainting, while maintaining fine details.

The fact that the unproject-project method leads to holes that need inpainting is interpreted as an advantage, because holes give a clear indication of flaws in the implementation or camera setup. On top of that, it is possible that an inpainter achieves a higher quality than the alternative of textured 3D meshes, where the textures are stretched out across elongated triangles.

When testing on the "Temple" scene, it became clear that the pre-processing of the edges was not necessary. In other words, for some scenes it can be helpful, but it is best to compare the results with and without the pre-processing when a new scene is visited.

The disk-based blending approach forms the main deviation from RVS and is therefore the primary explanation for the improvement in quality of the proposed proof-of-concept over RVS. It addresses RVS's problem of blending to many input pixels and not giving priority to smaller $\widehat{IPV}$ angles (I = input camera, P = 3D point, V = virtual camera), which is important when the scene contains view dependent elements.

Lastly, Section 4.7 discussed how to determine the 6-DoF when a certain camera setup is used in a given scene. In general, the disk-based approach leads to smaller 6-DoF volumes than when it is not applied, for example in RVS.

Chapter 5

Offline extension of the light field

The goal of the end-to-end system proposed in this master dissertation is to achieve real-time DIBR within a given scene. The use case that is considered is a VR experience where the viewer can freely move around in a predefined 6-DoF volume within the displayed scene. Chapter 3 covered how to setup the cameras, referred to as input cameras, in order to capture a desired part of the scene. Chapter 4 then discussed a DIBR implementation with no hard time limits that focuses on producing close-to-groundtruth results. This chapter explains how an offline DIBR step can use these building blocks to achieve real-time DIBR.

No proof-of-concept of the proposed end-to-end system was implemented, as this was considered as out-of-scope for this master dissertation. Section 5.4 opens a discussion of the feasibility and scope of proposed theory.

5.1 Motivation

As mentioned in Section 1.2, if a realistic camera setup is used, for some positions in a general 6-DoF volume, many input images will need to be processed by the chosen DIBR implementation for every frame in order to get a high-quality result. To meet the real-time requirements, e.g. a minimum of 90 frames per second for VR, the number of pixels that need to be processed per frame will have to be reduced. The offline extension step takes care of this. Due to the offline extension step, the number of input images that will need to be processed by the real-time DIBR during the last step of the end-to-end system will be fixed and independent of the size and shape of the 6-DoF volume. This allows for larger and differently shaped 6-DoF volumes without losing performance or quality.

5.2 Extension in 2D

Figure 5.1 shows a 2D example of a collection of input cameras capturing a part of the scene, as well as a desired 6-DoF volume. The input cameras produce a light field. During the offline extension step, DIBR will be used to add new virtual images to this light field, which can then be used as input for the real-time DIBR step.

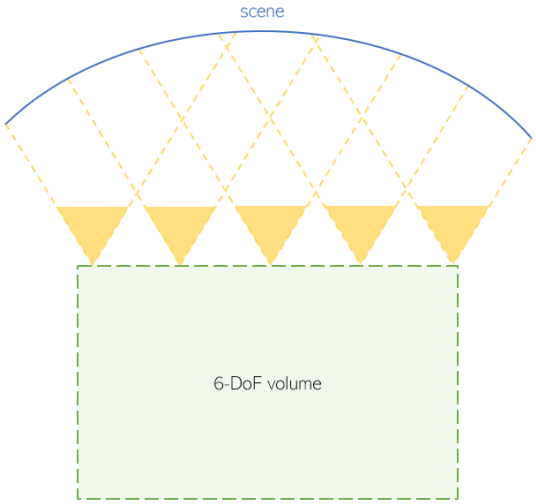


Figure 5.1: Input cameras that capture the scene.

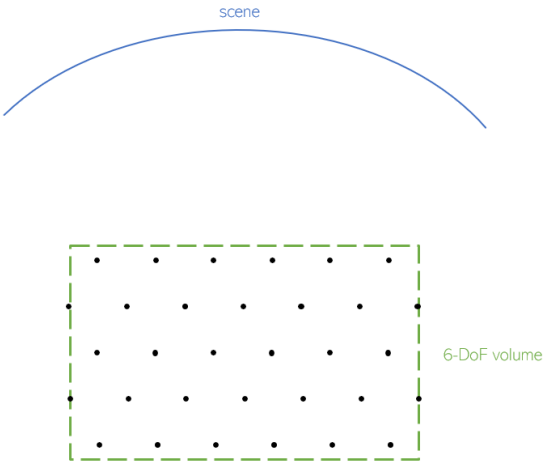


Figure 5.2: The positions of the extension cameras.

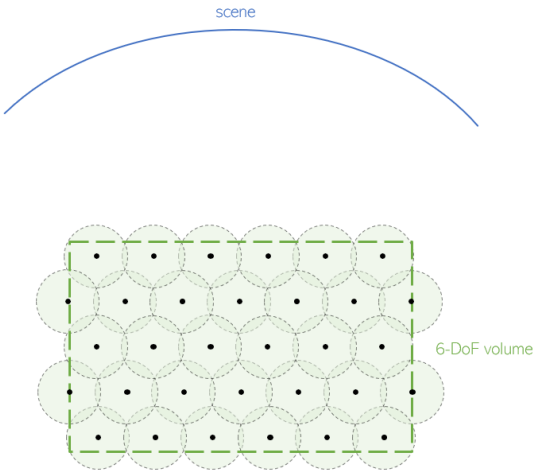


Figure 5.3: The circles around the extension cameras have radius MAX_DIST, indicating that any output camera in the 6-DoF volume is within this distance of at least one extension camera.

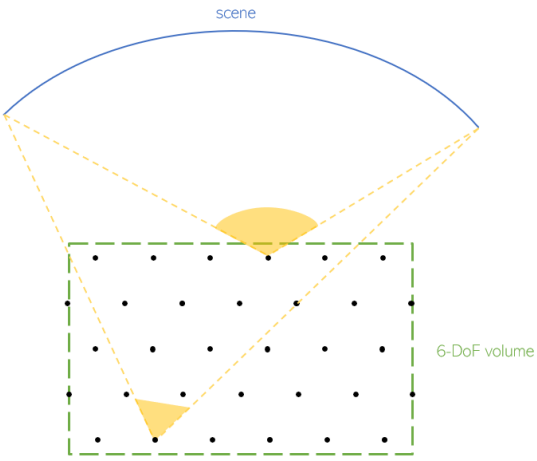


Figure 5.4: The FOV of two of the extension cameras, seeing the entire scene.

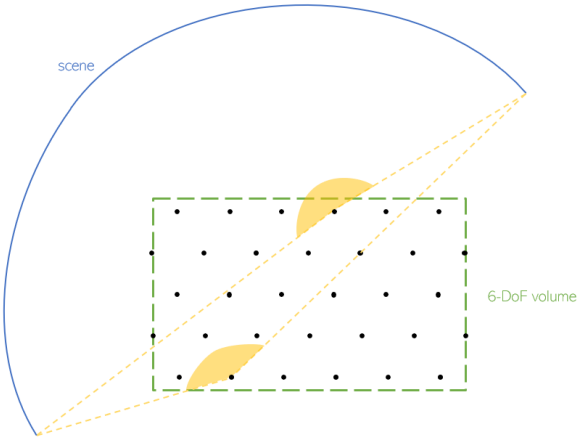


Figure 5.5: The FOV of two extension cameras when a larger part of the scene was originally captured by the input cameras.

For this, the 6-DoF volume is filled with virtual cameras, indicated by black dots on Figure 5.2. In this example, these *extension cameras* are positioned according to a hexagonal pattern, which was mentioned in Section 3.4. The reason for this is as follows.

For the real-time DIBR step, the one, two or three extension cameras that are closest to the virtual camera, e.g. the eye of the viewer in the VR use case, will be used as input cameras. In general, the quality of a DIBR result increases if the distance between the processed input cameras and the virtual camera decreases. With this in mind, a desired maximum distance MAX_DIST between any virtual camera within the 6-DoF and the up to three closest extension cameras can be defined. Figure 5.3 shows a sphere around each extension camera, with radius MAX_DIST. The cameras must be placed in such a way that every point of the 6-DoF volume lies in at least one of these spheres. In this case, the hexagonal pattern takes care of this, while minimising the number of necessary cameras. Similarly, if two or three extension cameras will be used during the real-time DIBR step, each point of the 6-DoF volume should be within at least two or three spheres respectively.

The FOV in width and height of each extension camera can be chosen so that the entire captured scene is within view, as illustrated by Figures 5.4 and 5.5. If the cuboid or spherical camera setups of Section 3.4 are used to capture the scene, each extension camera can have a FOV of up to 360°. Luckily, it is not necessary for the DIBR in the final real-time step to process every pixel of these 360° images, only for those belonging to parts seen by the virtual camera.

5.3 Extension in 3D

For 3D 6-DoF volumes, it becomes trickier to find an arrangement of the extension cameras within this volume so that the number of cameras is minimised for some MAX_DIST. One approach is to start from a *close-packing of equal sphere* [14], which fills the 6-DoF volume with non-overlapping spheres of equal size, and increase the radius until every point in the volume lies in at least one sphere. It can be shown that if any point in the 6-DoF volume should be within a maximum radius MAX_DIST of at least one extension camera, the positions of the extension cameras can be chosen as the centres of the spheres in a close-packing, where the spheres had a radius of $\frac{MAX\_DIST\sqrt{3}}{2}$. Note that near edge of the 6-DoF, this is extension cameras will need to be slightly adjusted. Figure 5.6 shows two example arrangements, i.e. the hexagonal close-packed (HCP) and face-centred cubic (FCC) lattice.

Assuming that the extended light field was made as explained above, the choice of the input extension cameras for the real-time DIBR step is as follows. It suffices to use the up to three closest extension cameras in front of the virtual camera. In conclusion: the extension of the original light field has enabled the DIBR for the real-time step to limit itself to using up to three input images per frame. The quality of the end result depends on the used MAX_DIST and the quality that the DIBR implementation can produce during the offline extension step.

Chapter 4 already analysed such an implementation.

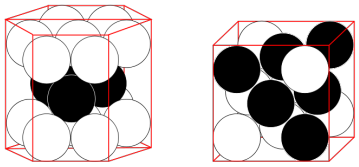


Figure 5.6: Two common high-density equal sphere packing arrangements: the HCP lattice on the left and the FCC lattice on the right.

5.4 Discussion

The main challenge of the proposed end-to-end system is that the maximum achievable quality is determined by the the quality attainable by the DIBR implementations used during the offline light field extension step and the real-time VR step. Unfortunately, most DIBR implementations do not focus on generating accurate depth maps, implying that more novel research on this will need to be performed. For these reasons, it does not make sense to implement the system until accurate DIBR implementations for these roles exist.

During the dissertation, a test was done to determine the MAX_DIST necessary for RVS to deliver high-quality results. The exact values depend on the scene geometry and how challenging the scene is, but it can be expected around the magnitude of 5 centimetres. For a cubic 6-DoF that is 1m $\times$ 1m $\times$ 1m, this comes down to around 2200 extension cameras, since the cube in Figure 5.7 misses some spheres around the edges of the cube.

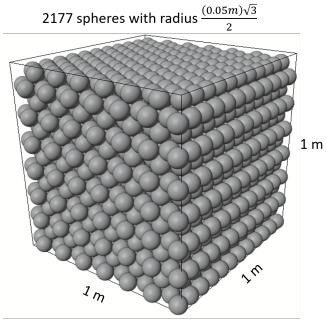


Figure 5.7: Close-packing of spheres in a cube.

On can reason about the size of the extended light field. For a given 6-DoF volume, the extension cameras are spread uniformly throughout the 6-DoF volume, for example via a close-packing of equal spheres algorithm. Doubling the 6-DoF volume will in general result in have to use a little less than double the number of extension cameras as before. For a fully immersive experience, 360° of the scene will be captured by the original camera setup. Therefore, each extension is chosen to be a 360° camera, because during the real-time DIBR step, the virtual camera will rely on its neighbour extension cameras to get the information for whichever direction the virtual camera is looking in. If the per-eye display of the head-mounted device has resolution $x \times y$, the resolution of each extension camera is, for simplicity, chosen to be six times $x \times y$. In other

words, each extension image is a cubemap with six sides, each the resolution of the per-eye display. Note that this resolution is just chosen for simplicity in this example, other projection maps might lead to different values.

For a VR headset with per-eye resolution 1440×1600, with the cube 6-DoF from earlier, the extension light field consists of around 2200 images consisting of $6 \times 1440 \times 1600$ pixels. Without compression and for 8-bit images, this comes down to approximately 28.324 gigabytes of rendered data. Moreover, the time to generate the extended light field can be calculated by multiplying 2200 images by the average time it takes to render one image.

Chapter 6

Evaluation of proof-of-concept DIBR

This Chapter contains an analysis of the output generated by the DIBR implementation from Chapter 4 for different parameters and scenes. The three scenes that can be used for evaluation as well as their camera setups were discussed in Chapter 3. Section 6.1 analyses the use of the linear versus the gaussian fall-off functions for the blending of the disks are applied, as well as the disk radii size. The tests are done on the "Regular classroom" and "Mirror of classroom" data sets, so that the results would be representative for general scenes with diffuse and view dependent elements. To asses the quality, the *Peak Signal-to-Noise Ratio* (PSNR) of the virtual images with respect to their groundtruths is calculated. On top of this, this work displays some representative result images to allow for subjective visual inspection. For the PSNR calculations, an image bit depth of 8 bit is used.

Section 6.2 uses the best fall-off function and disk radius from Section 6.1 to let the proof-of-concept render the images for virtual cameras spread throughout the 6-DoF volume. The goal is to analyse the quality (again from PSNR values and visual inspection) when the virtual camera is closer or further away from the processed input cameras. This gives an idea of how well the DIBR would perform for larger 6-DoF volumes, where more input cameras need to be processed per frame. This time, the "Regular classroom" and "Temple" scenes are used.

Section 6.3 compares the quality of the proof-of-concept to that of MPEG's RVS, in order to better position the PSNR values within the state-of-the-art.

6.1 Gaussian versus linear fall-off

Section 4.6 covered two possible fall-off functions for the blending of the disks: the linear and the gaussian fall-off functions. Different disk radii for both fall-offs were tested for two scenes: "Regular classroom" scene" and "Mirror of classroom". As seen on Figure 6.9, only the mirror is used for the PSNR and histogram calculations.

PSNR

Table 6.1 gives the results of a PSNR analysis for the two scenes, where the PSNR between a groundtruth and a virtual image generated by the DIBR implementation is calculated. The

virtual camera is placed at the centre of the icosphere. For the "Regular classroom" scene, disk radius 0.075 cm is the minimum radius, i.e. a smaller radius would result in some disks not overlapping with their neighbours. On the other hand, the "Mirror of classroom" scene uses an denser icosphere camera setup where neighouring cameras are placed an approximate distance of 3 cm apart. In this case, 0.04 cm is the minimal disk radius. The virtual images contain small holes, i.e. pixels for which the DIBR could not assign any colour. These pixels are excluded from the PSNR calculations, so the PSNR in the table is the maximum PSNR, i.e. the PSNR if an inpainter where to perfectly fill these holes. The holes are not taken into account here so that the mentioned PSNR values are independent of whichever inpainter would be used.

Scene	Fall-off function	Disk radius (m)	Maximum PSNR (holes not included)
Regular classroom	Linear	0.075	29.06
		0.08	29.3
		0.1	29.35
		0.12	29.97
	Gaussian	0.09	29.31
		0.12	28.88
		0.15	28.55
		0.18	27.92
Mirror of classroom	Linear	0.04	29.26
		0.05	29.51
		0.06	29.33
		0.07	28.87
		0.08	28.33
	Gaussian	0.045	29.27
		0.06	29.20
		0.09	28.24
		0.12	26.82

Table 6.1: PSNR measurements for different disk fall-off functions and radii.

Figures 6.7, 6.8, 6.9 and 6.10 show some of the DIBR results, the silhouette of the disks as seen from the virtual camera (denoted as *disk pattern*), and the groundtruth. The bright red pixels represent holes that require inpainting.

Mirror of classroom scene

The behaviour of the "Mirror of classroom" in Figure 6.9 can be explained by considering what happens when only three input images are shown, like in Figure 6.1. The three left images show the virtual images if each time only one of the input images were to be processed. The blending step of the DIBR implementation takes these three and blends them using the disk-based fall-off function into the fourth image. Since the images are looking at a mirror from slightly different angles, their perceived reflections are close, but different. This explain why on the combined image, it looks as if three ghost chalk boards overlap. The image on the right shows the disk pattern.

Figure 6.1: The "Mirror of classroom" scene if three input images are used to generate the virtual image.

Histograms of errors

Additionally, the histograms in Figure 6.2 show the distribution of the size of the errors with respect to the groundtruth, when the images are converted to greyscale. From the histograms, it can be concluded that most pixels in the scene differ only slightly from their groundtruth counterpart. This explains why through subjective visual inspection, the quality might appear to be higher than the PSNR suggests.

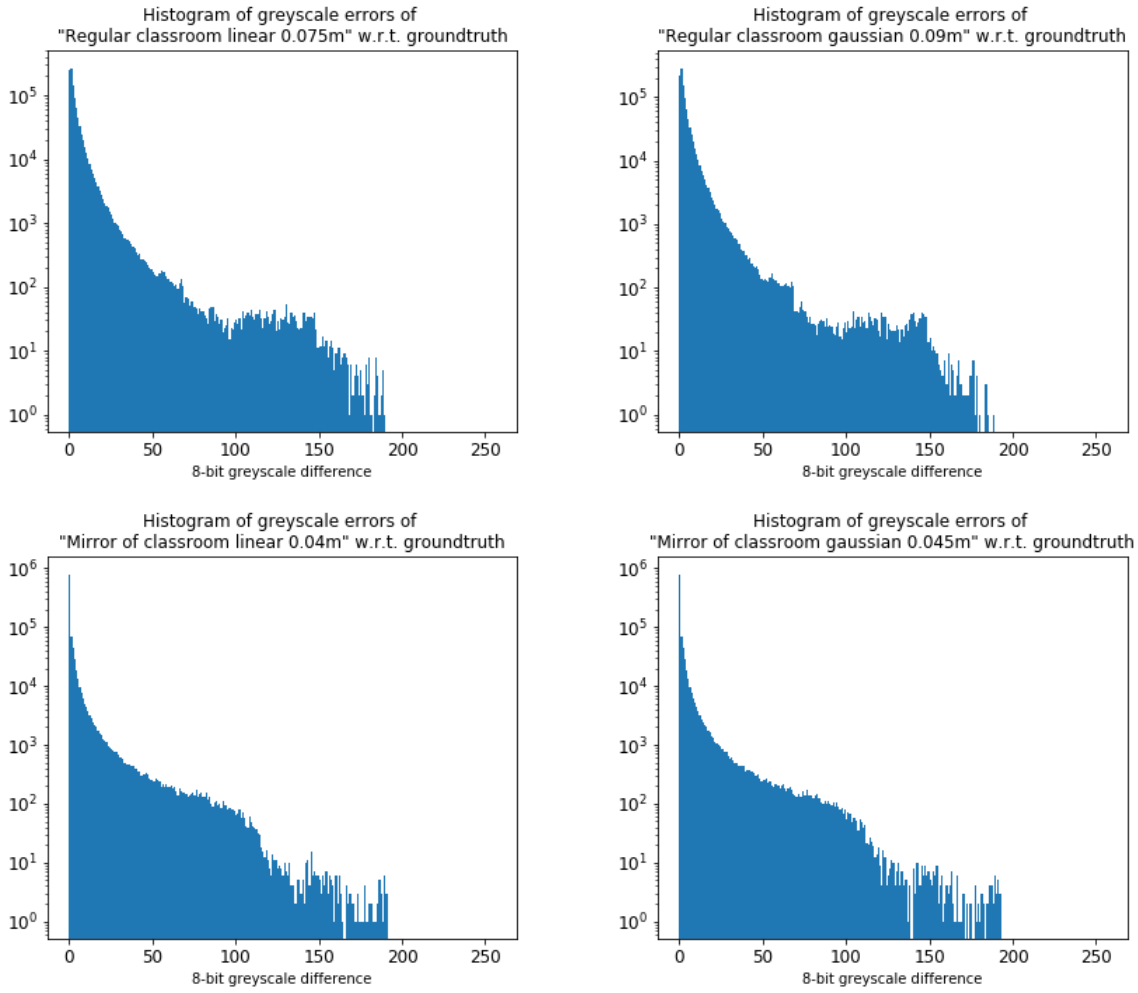


Figure 6.2: Histograms of the absolute differences between the DIBR results and their groundtruth, when they are converted to greyscale.

Conclusion

From the PSNR values and visual comparison of the DIBR results, it can be concluded that the smallest disk radii produce higher quality results. This is because the smaller the radii, the less blending there is between input images, leading to less blurriness in the virtual image. On the other hand, smaller disk radii increase the number of holes. In this sense, the use of the gaussian fall-off is preferable: larger disk radii than with linear fall-off can be used to achieve the same quality, since the blending near the disk edges is much less prevalent than with the linear fall-off. That is why in Table 6.1, the gaussian disk radii where generally larger than the linear ones, for approximately the same PSNR values.

6.2 Varying the virtual camera placement

Again, two scenes are used: "Regular classroom" and "Temple" mentioned in Chapter 3, the second scene contains more view dependent elements than the first. The DIBR implementation was used to get the virtual images for different camera positions within the icosphere camera setup. Figure 6.3 shows the selected positions as points, where the rings are 10 cm, 20 cm and 30 cm away from the center, while the radius of the icosphere is 40 cm. The virtual camera cannot be placed too close to the outer edge of the icosphere, as explained in Section 4.7. Since the icosphere is semi-symmetrical, only these 34 positions in one quarter of the icosphere are used instead of the whole sphere. To evaluate the quality of the DIBR proof-of-concept, a virtual image for each part of the scene from each of these 34 points could be considered. However, the rotation angle of the virtual camera is kept fixed, so that only one part of the scene is used for evaluation, i.e. the most challenging part of the scene.

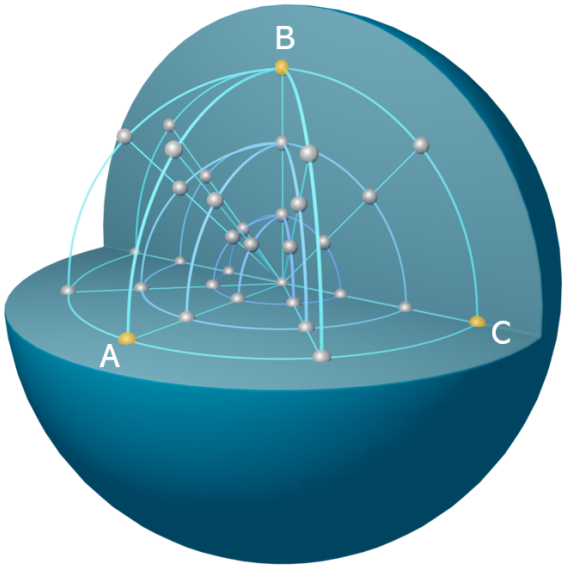


Figure 6.3: The 34 points in the icosphere at which the virtual camera was placed.

PSNR

For the "Regular classroom", the mean PSNR is 29.85 ± 0.2456. The "Temple" has a mean PSNR of 28.72 ± 0.6570. Again, the holes, represented as bright red pixels in this dissertation, are excluded from the PSNR calculations, so these values represent the maximum PSNR. Figures 6.11 and 6.12 show the results for three of the 34 points per scene, corresponding with points A, B and C on Figure 6.3.

Conclusion

From the PSNR and visual inspection, it can be concluded that the quality of the virtual images does not change much for different camera positions, if the camera rotation stays fixed. Inspection of the depth maps for the "Temple" scene indicate that their accuracy is lower than that of the other scene, which explains why the quality for these virtual images is lower. This is because the range of possible depth values for the "Temple" scene is ten times larger than that of the "Regular classroom" scene, while only 16 bits can be used per depth value. This problem and possible solutions are further discussed in Section 7.2.

The goal of varying the virtual camera position within the icosphere was to analyse the quality when the virtual camera is closer or further away from the processed input cameras. Since the quality remained approximately the same for the different positions, it can be concluded that the proposed DIBR will perform well for larger 6-DoF volumes, where more input cameras need to be processed per frame.

6.3 Comparison with an existing DIBR

In this Section, two DIBR implementations will be used to generate four virtual images within the "Temple" scene. To be precise, the virtual cameras are all positioned at the centre of the icosphere, as in Section 6.1, but with four different rotations: front, left, back and right. The first implementation is the proof-of-concept DIBR from Chapter 4. The second implementation is MPEG's RVS, which was introduced in Section 1.2.

Tuning RVS

Attention needs to be payed to the choice of the input images to be processed by RVS per frame. For the proof-of-concept, using as many input images as possible is generally the best approach. However, RVS does not have a mechanic that limits blending between the input images, like the proof-of-concept has thanks to the small disk radii and fall-off function. Therefore, different collections of input images for the four "Temple" views were tested and the best configurations are used in this dissertation. To be precise, the RVS results are best when the minimal collection of input images that reduces the impact of large holes as much as possible is processed.

PSNR

Table 6.2 displays the PSNR results, for which 8 bit images were used. Again, for the proof-of-concept, the holes that need inpainting are not included in the PSNR calculations, so their mentioned PSNR values are the maximum achievable PSNR. For RVS, the numbers of inpainted pixels is so low that excluding them from the calculations makes no significant difference. The

images generated by the two implementations, as well as their groundtruth, are shown in Figures 6.13 to 6.16.

PSNR	Proof-of-concept DIBR	RVS
Front	29.08	25.43
Left	23.36	17.65
Back	24.15	17.4
Right	26.93	23.26

Table 6.2: Comparison of PSNR measurements on four "Temple" views between the proof-of-concept DIBR and RVS.

Conclusion

As mentioned in Chapter 4, the difference with the most impact between the proof-of-concept and RVS, is the use of the disks and fall-off function to limit the blending between different input images. The PSNR values and visual inspections confirm that using the disks to limit blending indeed greatly improves the resulting quality. For example on the RVS generated image in Figure 6.14, there are artefacts around the branches of the tree that are clearly visible, as well as a prominent blurring of the stone tiles at the bottom of the image. The blurring of details and edges are caused as follows: if only one input image would be processed, the virtual image would slightly deviate from the groundtruth. Using multiple input images comes down to blending all their resulting virtual images, which in the end leads to a blurry result.

The disk based approach is good for limiting blending, but also for giving priority to pixels of which the $\widehat{IPV}$ angle (I = input camera, P = 3D point, V = virtual camera) is smaller. This results in a better rendering of view dependent elements, as illustrated Figure 6.4, which shows a close-up of some view dependent elements in the "Temple" scene.

Figure 6.4: Close-up of front view of "Temple", from left to right: the RVS result, the DIBR proof-of-concept result and their groundtruth.

6.4 Conclusion

The three performed experiments bring the strengths and remaining weaknesses of the proposed DIBR to the light. The reason why the quality perceived through subjective visual inspection appears higher than the PSNR suggests, is because the rendered virtual images have many pixels that slightly differ from the groundtruth. The combination of the camera setup and DIBR implementation make it so that detailed textures are well translated from the input images to the virtual image, as illustrated by the ground of the "Regular classroom". The disk-based blending approach renders view dependent elements (sometimes significantly) closer to the ground truth than RVS, for example in Figure 6.4.

On the other hand, there is still room for improvements. The quality of the proof-of-concept is high in the centres of the input image disks, since the light rays that go through the centres of the disks are almost identical for the input camera and the virtual camera. However, near the edges of the disks this is not the case. If near the disk edges, the depths or colour pixels that are being blended are noticeably different, the result becomes blurry. This is illustrated by the candles on the middle image of Figure 6.4

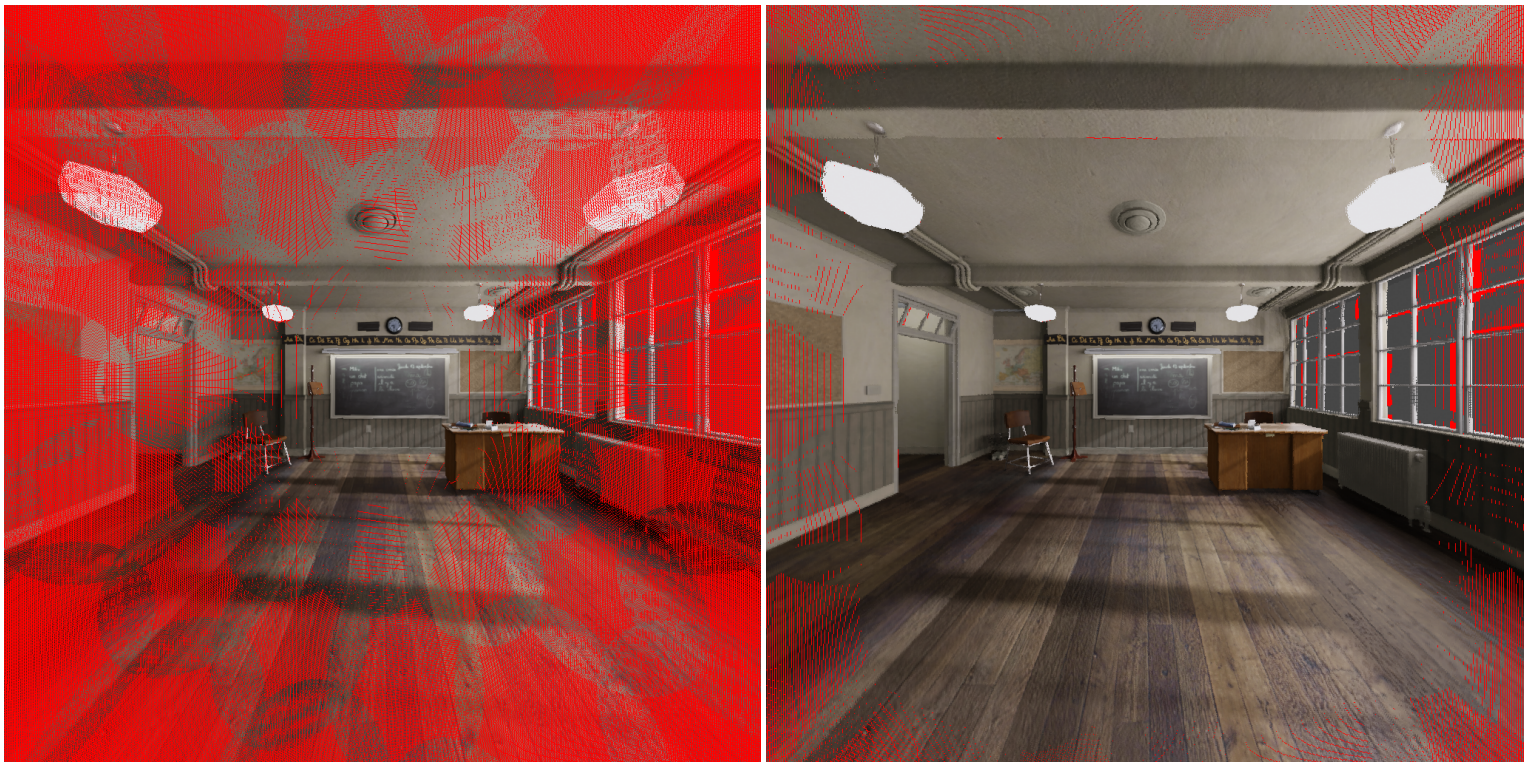
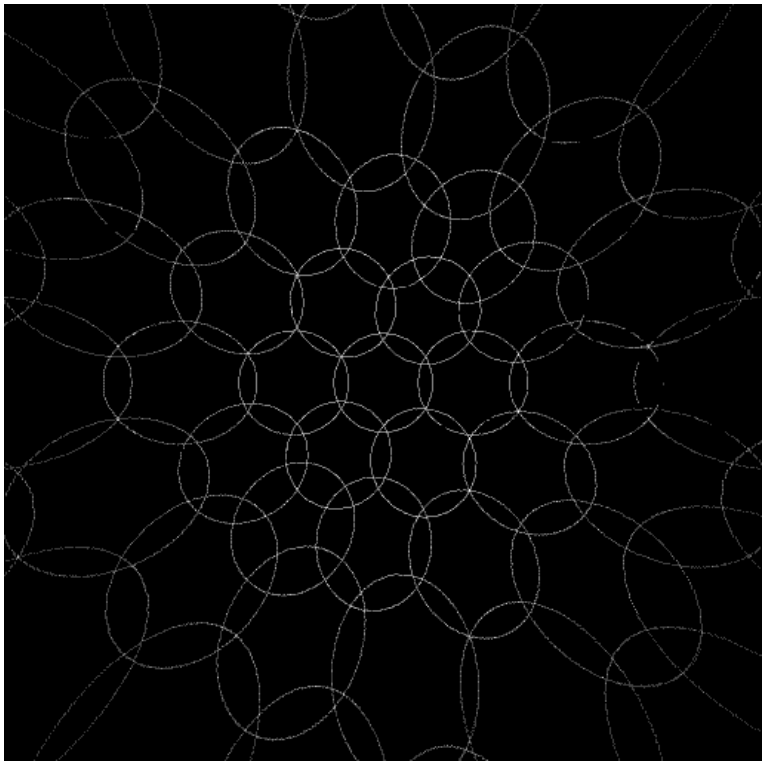


Figure 6.5: On the left is the DIBR result when the first rounding approach from Section 4.2.3 is used, i.e. when a 3D point is projected onto the virtual image, it is assigned to the closest $(row, column)$ pair. On the right, the 3D point is projected onto the four closest pairs/pixels.

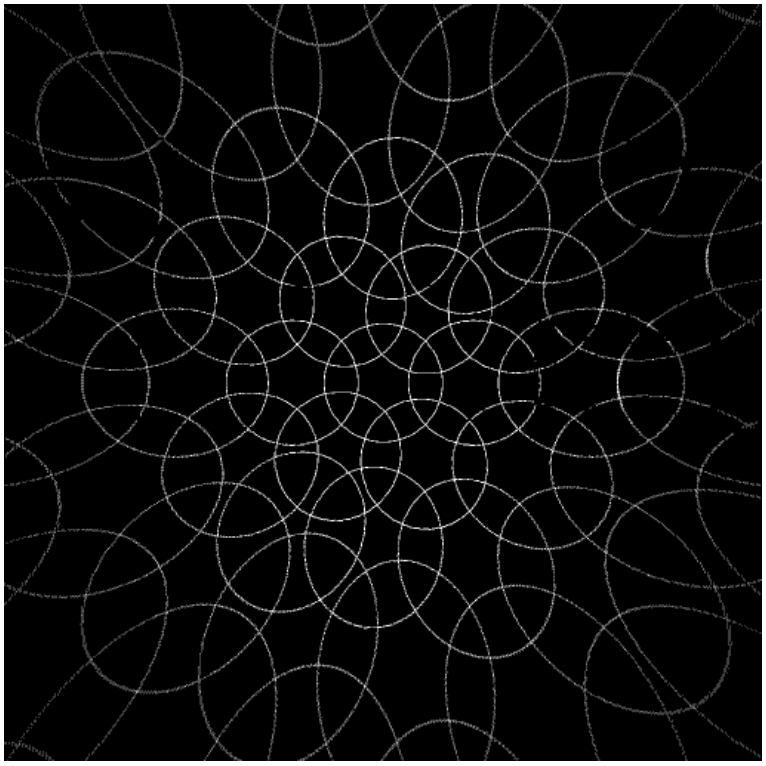
Figure 6.6: DIBR result when using one input image, on the left without the preprocessing step from Section 4.4 and on the right with. The red areas behind the lamps were occluded from the perspective of the input camera, but are visible for the virtual camera. The white edge artefacts on the left image (indicated by blue arrows) are caused by a difference between the depth map and the input image. To be precise, these white artefact pixels are white on the input image because they belong to the lamps, but the depth map gave them a depth value of the ceiling behind the lamp, resulting in these pixels being placed on the ceiling in the virtual image rather than on the lamp.



Disk pattern

Groundtruth

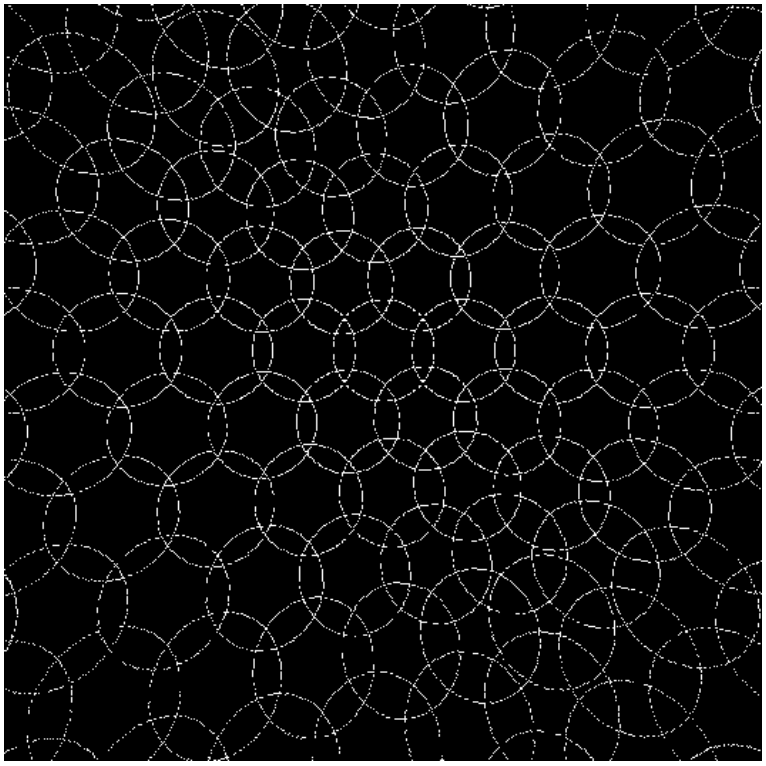
Figure 6.7: Regular classroom, linear fall-off, radius 0.075m



Disk pattern

Groundtruth

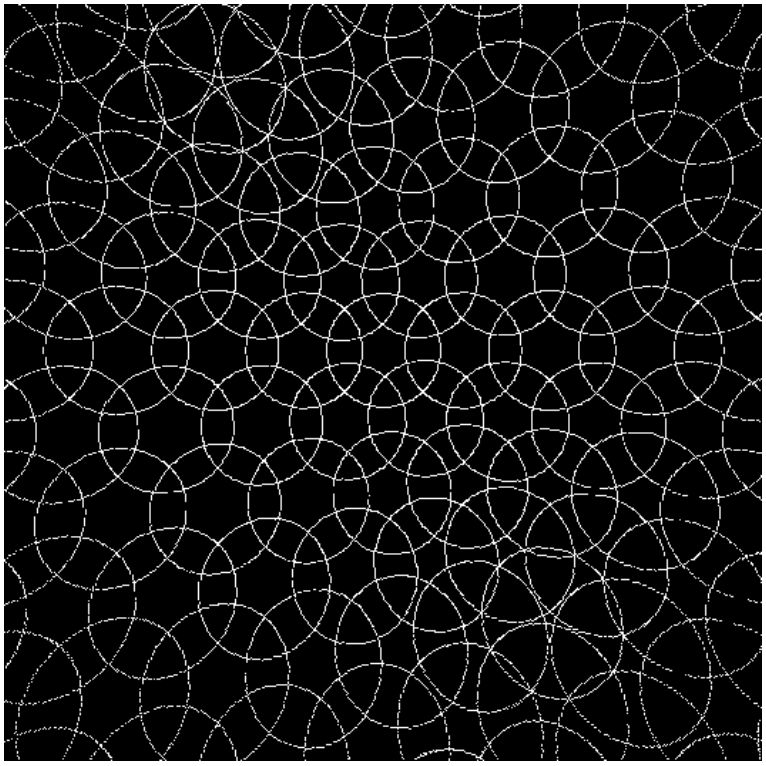
Figure 6.8: Regular classroom, gaussian fall-off, radius 0.09m



Disk pattern

Groundtruth

Figure 6.9: Mirror of classroom, linear fall-off, radius 0.04m



Disk pattern

Groundtruth

Figure 6.10: Mirror of classroom, gaussian fall-off, disk radii of 0.045m

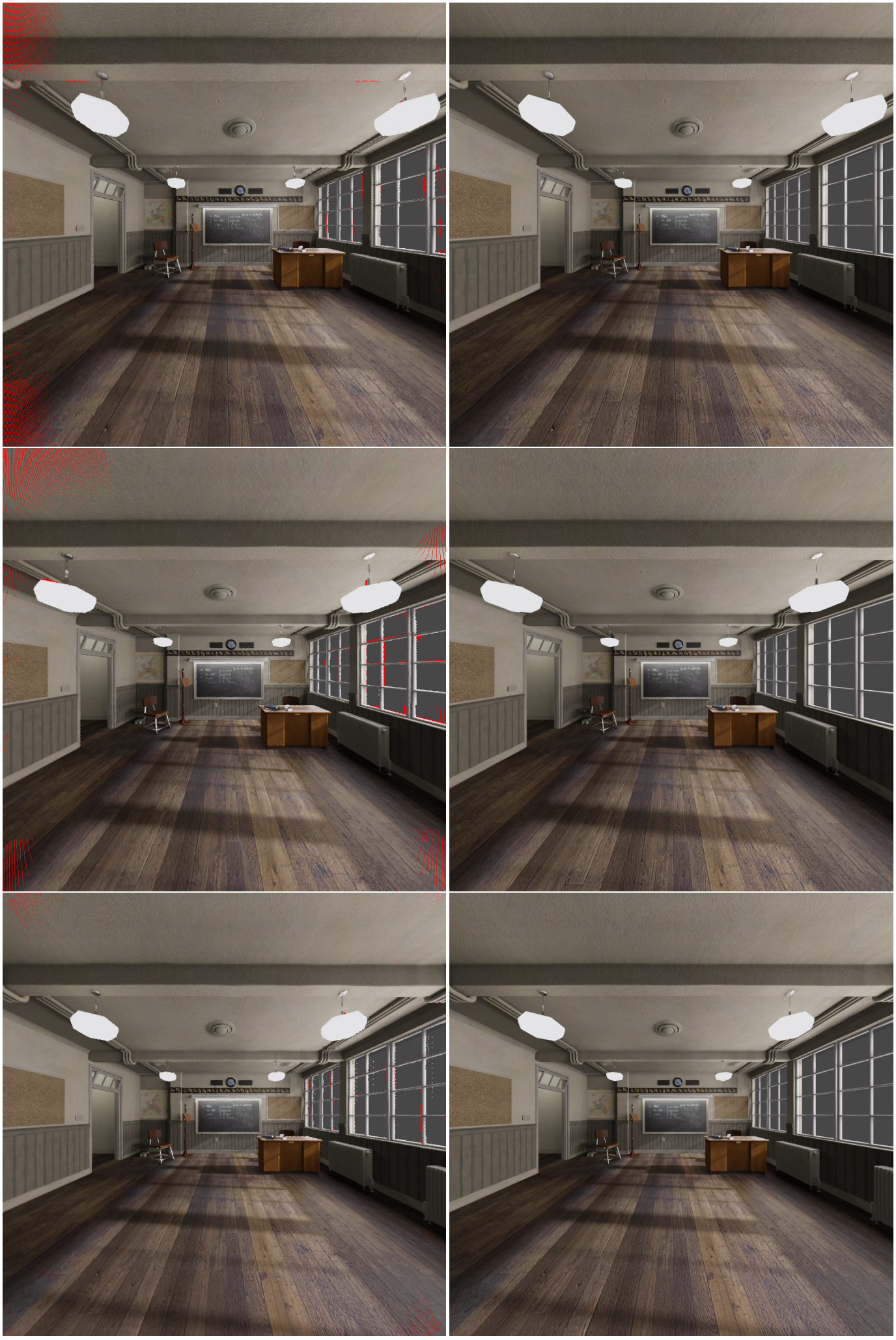


Figure 6.11: The three images on the left belong to a virtual camera on position A, B and C of the icopshere in Figure 6.3. On the right is their groundtruth.

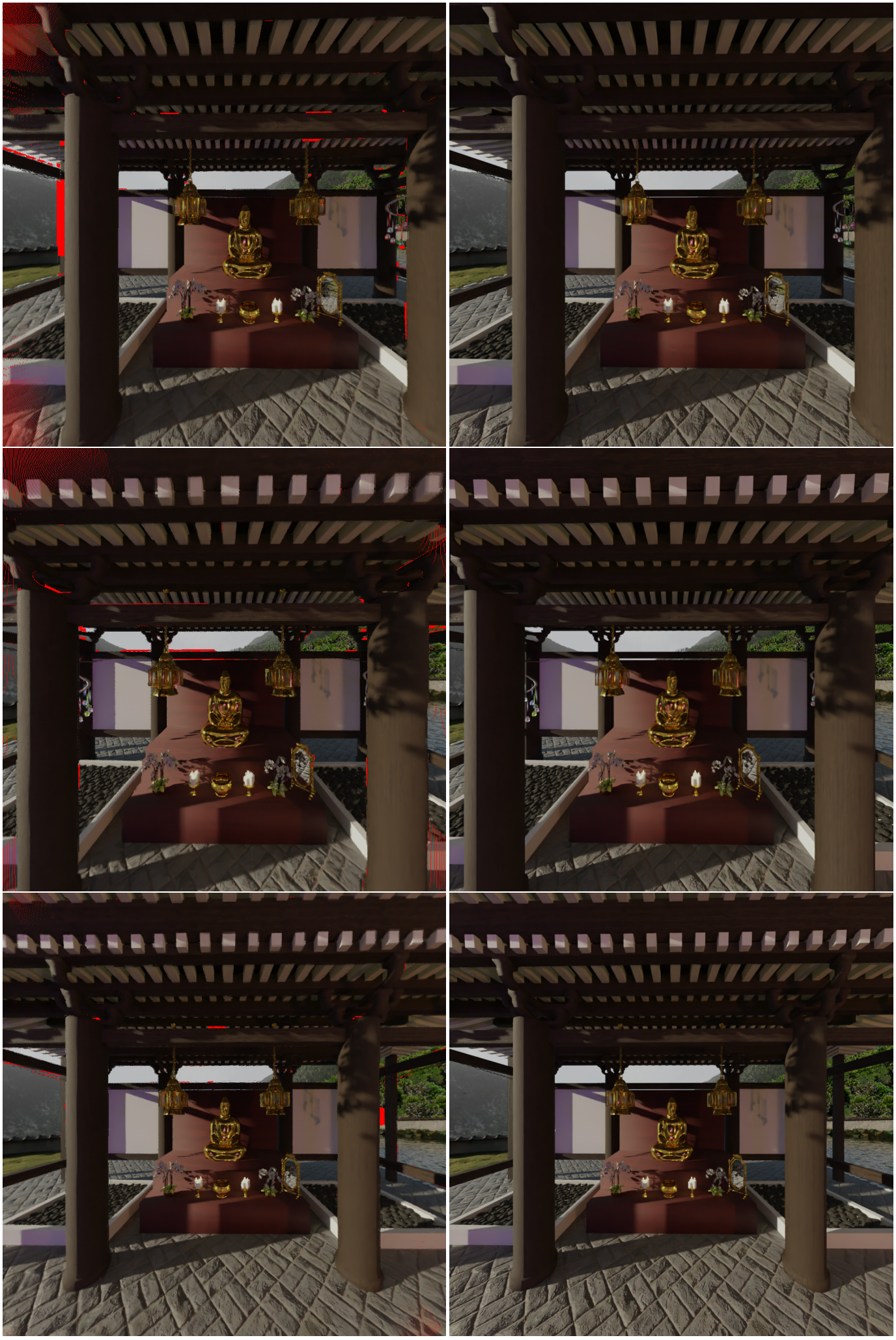


Figure 6.12: The three images on the left belong to a virtual camera 30cm to the left, above and behind the centre respectively. On the right is their groundtruth.

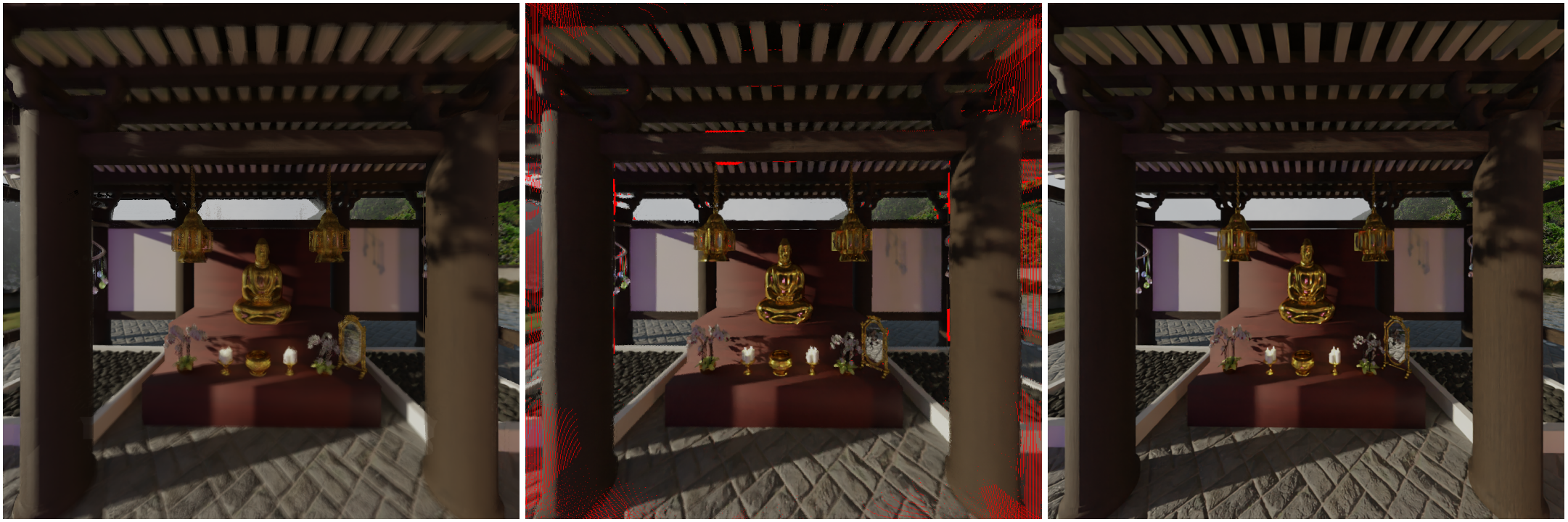


Figure 6.13: Front view of "Temple", from left to right: the RVS result, the DIBR proof-of-concept result and their groundtruth.

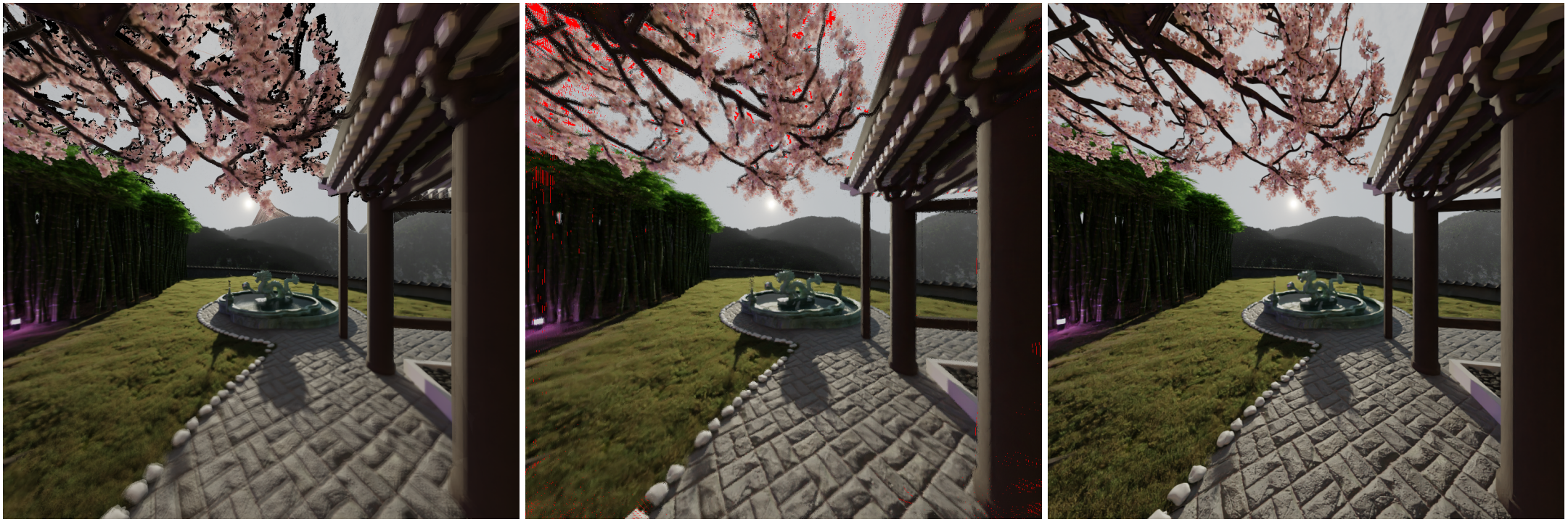


Figure 6.14: Left view of "Temple", from left to right: the RVS result, the DIBR proof-of-concept result and their groundtruth.

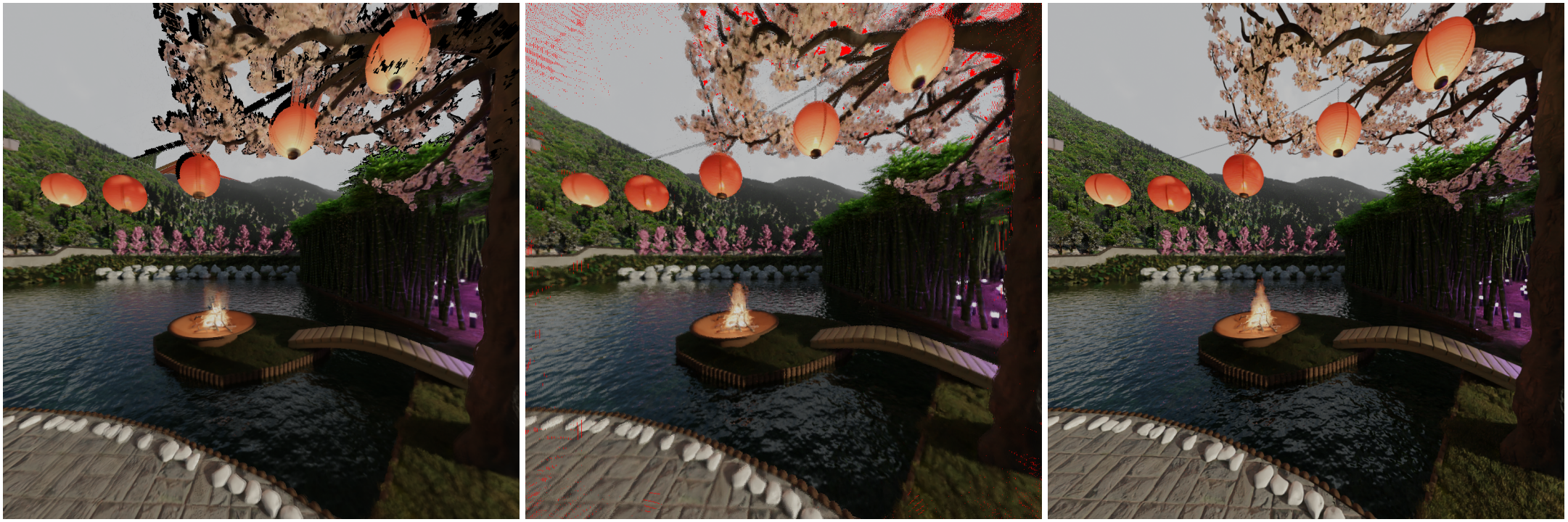


Figure 6.15: Back view of "Temple", from left to right: the RVS result, the DIBR proof-of-concept result and their groundtruth.

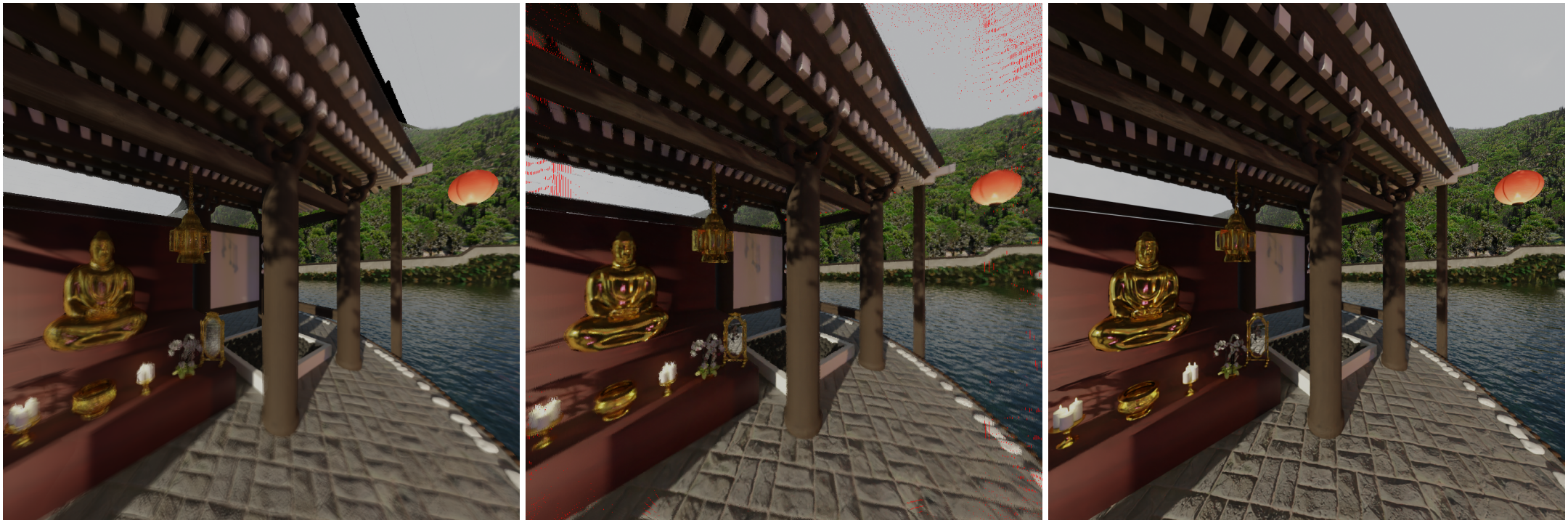


Figure 6.16: Right view of "Temple", from left to right: the RVS result, the DIBR proof-of-concept result and their groundtruth.

Chapter 7

Discussion

In Section 7.1, the contributions made in this work to the state-of-the-art are reviewed, i.e. they are compared to the goals formulated in the problem statement at the beginning of the master dissertation. Then Section 7.2 summarizes the remaining challenges which could not be addressed in this work and discusses potential solutions or improvements for these problems.

7.1 Review of made contributions

This work presented three main contributions to the state-of-the-art. The first is the offline DIBR implementation, of which a proof-of-concept was built. The design goals for the implementation were:

- Achieving close-to-groundtruth results for the rendered virtual images when a dense light field of the scene is given as input. Due to the disk-based approach of the DIBR, it can be concluded that in the centres of the input image disks, the quality of the proof-of-concept is high, since the light rays that go through the centres of the disks are almost identical for the input camera and the virtual camera. However, near the edges of the disks this is not the case. The blending near the edges of the disks can lead to low quality results when the depth maps and/or input images that are being blended differ noticeably. For the depth maps, this can be due to inaccuracies, for the images this can be due to view dependent elements.
  In conclusion, the DIBR implementation has the potential to deliver high-quality results, but a solution will have to be found for the problem regarding view dependent elements and differences between neighbouring depth maps.

- Being able to produce high-quality results for large 6-DoF volumes. In this work, large 6-DoF volumes could not be tested directly since the large camera setup would take too long to render. However, whether or not this requirement was met can be investigated as follows. During the evaluation of the proof-of-concept on 34 positions within an icosphere in Section 6.2, it was shown that the achieved quality, decided through the PSNR and visual inspection, stays approximately the same whether 5 input images or 111 are used. In other words, the quality remained stable independent of how far the virtual camera is from the processed input cameras.This supports the hypothesis that, due to the disk-based

approach, the DIBR implementation is able to process a large number of input images for away from the virtual camera while maintaining quality.
In conclusion, the challenge of large 6-DoF consists of having to process many images corresponding to input cameras far away from the virtual camera. The mentioned results support the hypothesis that the DIBR is indeed capable of maintaining quality even in larger 6-DoF volumes.

Secondly, the goal of the proposed end-to-end system was to make the performance and quality independent of the size of the 6-DoF volume, while also improving the overall achievable performance. Building a proof-of-concept of such a system was out-of-scope for this master dissertation, so the extent in which this requirement is met can only be discussed in theory. Section 5.3 explained how the offline extension of the light field results in the addition of a large number of cameras spread throughout the 6-DoF volume. The input cameras are packed as closely as possible, leading to a fixed number of closest neighbours and a maximum distance to the closest neighbours. This implies that during the real-time DIBR step, the virtual camera will always have a set number of neighbours n within a close distance d. In theory, this means that for each frame, only up to n images will need to be processed, where n is a generally small number. Since the n images are close to the virtual image (less than d apart), the quality will be high.
In conclusion, the end-to-end system is designed to meet the imposed requirements, in theory.

The last major contribution is the "Temple" scene. Section 3.6 illustrated how the scene fulfills the requirement of combining a wide spectrum of challenging scene elements. However, it can still be improved by adding more and larger reflective surfaces and by further developing the dynamic aspect of the scene.

7.2 Remaining challenges and possible improvements

- The depth maps used in this dissertation store the depth information per pixel as 16 bits, where each value lies between the minimum and maximum depth. The "Regular classroom" and "Temple" scene from Chapter 3 have the same minimum depth of 0.1 metre, but their maximum depth is 100 metre versus 1000 metre respectively. Therefore, the precision of the depth information is about ten times less for the "Temple" scene.
  A possible solution would be to give a higher depth precision to foreground objects and thus a lower precision for the background.

- The virtual images generated by the proof-of-concept DIBR in Chapter 6 contain notable amount of bright red pixels, i.e. holes for which no colour information was assigned. Expanding the dimensions of the disks in the disk-based blending approach will be able to fill most of these holes.
  It is therefore an option to design a novel inpainter that not only bases its decisions on the pixels in the virtual image, but also takes the coloured 3D scene points that were projected onto the virtual image but that fell outside of the disks into account. The downside to this

is that view dependent elements will not be inpainted well. However, in all other cases, the novel inpainter might perform better than existing implementations.

- The end-to-end system requires the offline DIBR implementation to not only generate close-to-groundtruth images, but also their depth maps. The proof-of-concept does produce depth maps, but it was considered to be out-of-scope for this dissertation to get them to be accurate.
  To improve the accuracy of the found depth, it might be useful to apply machine learning techniques, where a model learns to decrease the error between the generated depth map and its groundtruth.

- The scenes used in this work are computer generated and therefore produce relatively accurate depth maps. In a scenario where this is not the case, an additional pre-processing step of the depth maps can be included to the DIBR implementation. The pre-processing serves to detect and correct depth information. An example implementation is proposed in the paper by Quang H. Nguyen, Minh N. Do and Sanjay J. Patel on DIBR with low resolution depth [15].

Chapter 8

Conclusion

In this work, an end-to-end system was proposed to realise real-time, high-quality immersive experiences, where DIBR is used to take care of the realism and VR can accommodate the immersive experience. The system relies on a novel offline step where DIBR is used to extend the light field that was captured by the camera setup in a given scene. This dissertation proposed a novel DIBR implementation to be used in such an offline rendering step. Of the novel DIBR implementation a proof-of-concept was built and evaluated on three scenes.

It was shown that the proposed DIBR implementation is able to reconstruct diffuse objects with smooth geometry in high detail. The proof-of-concept is capable of reconstructing view dependent elements and elements that appear incorrectly on the depth maps better than a state-of-the-art DIBR alternative such as MPEG's RVS. Additionally, the quality remained approximately the same when processing a few versus a large amount of input images. This supports the hypothesis that the end-to-end system would be capable of working with larger areas in which the viewer has freedom of movement than state-of-the-art systems, while maintaining quality.

The computer-generates scenes cover a wide range of difficulty in order to challenge DIBR and allow making conclusions about the achievable quality in real-world settings. This work discussed a variety of scene elements that are challenging for DIBR to accurately reconstruct. Additionally, the characteristics of a good camera setup were covered, since the camera setup can be directly linked to the quality that a DIBR implementation would be able to accomplish.

One of the scenes "Temple" was especially designed to combine different real-world challenges. Data sets generated from this scene can be static or dynamic, i.e. the scene can be fixed in time or it can change over time. Its versatility makes it a valuable addition to state-of-the-art data set sources.

Bibliography

[1] M. Levoy, "Light fields and computational imaging," *Computer*, vol. 39, no. 8, pp. 46–55, 2006.

[2] J. Ogniewski, "High-Quality Real-Time Depth-Image-Based-Rendering," in *Linköping Electronic Conference Proceedings*, vol. 143. Linköping University Electronic Press, 2017, pp. 1–8.

[3] Y. Li, L. Claesen, K. Huang, and M. Zhao, "A real-time high-quality complete system for depth image-based rendering on fpga," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 4, pp. 1179–1193, 2019.

[4] S. N. Sinha, J. Kopf, M. Goesele, D. Scharstein, and R. Szeliski, "Image-based rendering for scenes with reflections," *ACM Trans. Graph.*, vol. 31, no. 4, Jul. 2012. [Online]. Available: https://doi.org/10.1145/2185520.2185596

[5] G. D. d. Dinechin and A. Paljic, "From real to virtual: An image-based rendering toolkit to help bring the world around us into virtual reality," in *2020 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, 2020, pp. 348–353.

[6] G. Luo, Y. Zhu, Z. Weng, and Z. Li, "A disocclusion inpainting framework for depth-based view synthesis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 6, pp. 1289–1302, 2020.

[7] R. S. Overbeck, D. Erickson, D. Evangelakos, M. Pharr, and P. Debevec, "A system for acquiring, processing, and rendering panoramic light field stills for virtual reality," *ACM Trans. Graph.*, vol. 37, no. 6, Dec. 2018. [Online]. Available: https://doi.org/10.1145/3272127.3275031

[8] S. Fachada, D. Bonatto, A. Schenkel, and G. Lafruit, "Depth image based view synthesis with multiple reference views for virtual reality," in *2018 - 3DTV-Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON)*, 2018, pp. 1–4.

[9] MPEG. (2018) Reference view synthesizer(rvs) manual. [Online]. Available: https://mpeg.chiariglione.org/standards/exploration/immersive-video/reference-view-synthesizer-rvs-manual

[10] C. Fehn, "Depth-image-based rendering (DIBR), compression, and transmission for a new approach on 3D-TV," in *Stereoscopic Displays and Virtual Reality Systems XI*, M. T.

Bolas, A. J. Woods, J. O. Merritt, and S. A. Benton, Eds., vol. 5291, International Society for Optics and Photonics. SPIE, 2004, pp. 93 – 104. [Online]. Available: https://doi.org/10.1117/12.524762

[11] S. G. B. M. BUTLER D., WULFF J., "A naturalistic open source movie for optical flow evaluation." in *In Proceedings of European Conference on Computer Vision (2012)*, 2012.

[12] D. Lischinski and A. Rappoport, "Image-based rendering for non-diffuse synthetic scenes," in *Rendering Techniques '98*, G. Drettakis and N. Max, Eds. Vienna: Springer Vienna, 1998, pp. 301–314.

[13] Nvidia. (2018) Nvidia ai inpainting. [Online]. Available: https://www.nvidia.com/research/inpainting/

[14] N. Sloane, "The sphere packing problem," *Documenta Mathematika, Vol. III (1998)*, pp. 387–396, Aug. 1998.

[15] Q. H. Nguyen, M. N. Do, and S. J. Patel, "Depth image-based rendering with low resolution depth," in *2009 16th IEEE International Conference on Image Processing (ICIP)*, 2009, pp. 553–556.

FACULTY OF ENGINEERING AND ARCHITECTURE

Offline Depth Image Based Rendering for Immersive Experiences

Julie Artois
Student number: 01504096

Supervisors: Prof. dr. Peter Lambert, Prof. dr. ir. Glenn Van Wallendael
Counsellors: Niels Van Kets, Ir. Martijn Courteaux

Master's dissertation submitted in order to obtain the academic degree of
Master of Science in Computer Science Engineering

Academic year 2019-2020

GHENT UNIVERSITY