

ARSlice: Head-Mounted Display Augmented with Dynamic Tracking and Projection

Yu-Ping Wang
Tsinghua University
wyp@tsinghua.edu.cn

Sen-Wei Xie
Tsinghua University
jsw19@mails.tsinghua.edu.cn

Lihui Wang
Institute of Semiconductors
Guangdong Academy of Sciences
wanglihui@gdisit.com

Hongjin Xu
KDDI Cooperation
hongjinxu95@gmail.com

Satoshi Tabata
The University of Tokyo
tabata@ishikawa-vision.org

Masatoshi Ishikawa
The University of Tokyo
ishikawa@ishikawa-vision.org

Abstract

Computed tomography (CT) generates cross-sectional images of the body. Visualizing CT images has been a challenging problem. The emergence of augmented and virtual reality technology has provided promising solutions. However, existing solutions suffer from tethered display or wireless transmission latency. In this paper, we present *ARSlice*, a proof-of-concept prototype that can visualize CT images in an untethered manner without wireless transmission latency. Our ARSlice system consists of two parts, the user end and the projector end. By employing dynamic tracking and projection, the projector end can track the user-end equipment and project CT images onto it in real time. The user-end equipment is responsible for displaying these CT images into 3D space. Its main feature is that, the user-end equipment is a pure optical device with light weight, low cost, and no energy consumption. Our experiments demonstrate that our ARSlice system provides part of six degrees of freedom for the user, and a high frame rate. By interactively visualizing CT images into 3D space, our ARSlice system can help doctors better understand CT images.

1. Introduction

Computed tomography (CT) is a computerized X-ray imaging procedure, which generates cross-sectional images (or “slices”) of the body. Traditionally, doctors need to be trained to “imagine” the body from these slices. With the help of 3D reconstruction, we can build a 3D model based on these slices. However, the goal of doctors is usually not to look at the 3D model from the outside. Finding a way to visualize CT images to help doctors see inside the body is a challenging problem.

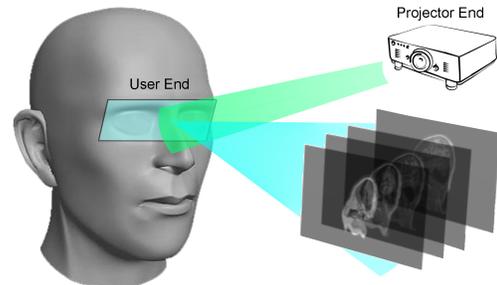


Figure 1: Overview of the proposed ARSlice system. The projector end generates virtual content and augments the user end; and the user end shows virtual content in 3D space. The user-end equipment is a pure optical device, and thus is light weight and untethered, and there is no need to worry about energy consumption.

Spatial Augmented Reality (SAR) technique provides a promising solution. SAR [29] is an interactive technique that can augment virtual appearance onto real-world objects. In a common SAR system, projectors are employed to project images onto white objects, and the objects can then be observed by users through their natural eyes. For example, augmenting a wall to show a furnished room [3], augmenting a model to show a human face [21, 22], or augmenting clothes to try different styles [27]. Wang et al. [32] designed a system to augment CT images onto a moving plane. Common disadvantages to SAR systems, however, include that they rely on real-world objects, and usually require a large space. These disadvantages limit the application scenarios of SAR techniques.

Head-Mounted Displays (HMDs) supported by Augmented Reality (AR) and Virtual Reality (VR) technology provides another promising solution [7]. AR/VR are techniques of integrating virtual content into 3D space in real

time [1]. They have been proven applicable in many scenarios, such as education [13], security [4], medical treatment [2], and industry [26]. The main issue when visualizing CT images using HMDs is virtual contents generation. CT images and the 3D model are usually large in size, and high computing power is required to generate virtual contents according to CT images. In terms of where the virtual contents are generated, existing AR/VR HMDs can be classified into two categories. The first kind of HMD (Discrete HMD) does not generate AR/VR contents, but relies on an external computer to do it. In this case, AR/VR contents have to be transmitted to the HMD. For most commercial products, such as HTC Vive, Oculus Rift, and PlayStation VR, a video cable can serve the purpose. Such a solution is also called Tethered HMD. The presence of video cable affects its user experience and limits its application scenarios. The second kind of HMD (Integrated HMD) generates AR/VR contents by itself, such as Google Glass, Microsoft HoloLens, and even Google Cardboard. This solution is limited by the need of balancing computing power, energy consumption, and weight. Higher computing power enables generating high quality virtual contents, but increases energy consumption and weight. CT images and the 3D model are usually large in size, and high computing power is required to visualize them. In order to lower the requirement of computing power, transmitting the virtual contents from the server to the HMD could be feasible [24]. For such a system, the transmission latency becomes the main issue, since a Motion-To-Photon (MTP) latency higher than 16ms can cause severe motion sickness to VR users [31]. Therefore, comprehensive per-application optimizations are required to achieve acceptable low MTP latency.

Overall, existing solutions are affected by the presence of extra real-world objects for SAR technique, the presence of video cable for discrete HMD, and the energy consumption or transmission latency for integrated HMD. The challenge that we are facing is to design an AR/VR architecture without tethered, with low energy consumption and transmission latency.

In this paper, we present a novel AR/VR architecture, called *ARSlice*. Its brief architecture is shown in Fig. 1. It employs a projector, much like SAR. However, the projector does not project images onto real-world objects, but to the HMD worn by the user. With *ARSlice*, the user can wear an HMD that is a pure optical device with light weight, low cost, and no energy consumption. With a properly designed dynamic tracking and projection algorithm, the projector can continue tracking and projecting images onto the HMD, while the HMD displays these images into 3D space with their optical devices. As a result, *ARSlice* is more applicable than SAR solutions, because it needs no extra real-world objects; it is more applicable than integrated HMD solutions, because the HMD is relieved from virtual con-

tent generation; it is more applicable than discrete HMD solutions, because it can achieve low MTP latency without tethered or energy consumption.

The main contributions of our work are as follows:

- We present a novel AR/VR architecture, called *ARSlice*. The HMD is a pure optical device with light weight, low cost, and no energy consumption. To our knowledge, none of the previous designs have augmented the HMD itself with a projector as our *ARSlice* system does.
- We introduce a hardware configuration and software implementation for our *ARSlice* prototype, along with a geometric calibration technique when real-world measurements are limited.
- We demonstrate our *ARSlice* prototype by interactively visualizing CT images. We also discuss the impact of some parameters during implementation and potential improvements.

The rest of this paper is organized as follows. Section 2 discusses related work. Section 3 introduces the design and implementation of our *ARSlice* prototype. Section 4 shows the experimental results. In Section 5, we discuss some limitations and possible extensions. Finally, we conclude the paper in Section 6.

2. Related Work

2.1. Spatial Augmented Reality

SAR technique employs projectors to project augmented virtual content onto real objects. Thus, multiple users can simultaneously see augmented virtual content with their natural eyes. Projecting onto different objects offers different application scenarios. Ehnes et al. [6] employed a projector that is capable of panning and tilting. By detecting and tracking a marked target, it can augment walls or turn a marked white board into a control panel. Schwerdtfeger et al. [30] presented two designs to realize SAR: a head-mounted projector, and a tripod-mounted projector. Lincoln et al. [21, 22] proposed a method of projecting onto a model of human head. The projected content was captured by a real person. Thus, they could provide a full-duplex tele-presence experience. Jones et al. [16] scanned target surfaces before projecting onto them. Thus, they could augment everyday objects. Narita et al. [27] further enabled projections onto dynamic deformed surfaces (e.g., clothes) by designing a Deformable Dot Cluster Marker (DDCM). These works have been inspirational. However, the common drawback of these SAR techniques is that they require real-world objects onto which to project. A traditional SAR system cannot show users a virtual object in an empty space. *ARSlice* employs the same methodology as SAR

techniques, but it can project virtual content on the HMD to enable the user to see virtual objects without relying on real-world surfaces.

Displays that combine HMDs and projectors have been proposed. Hua et al. [12] designed a head-mounted projective display (HMPD) prototype. They mounted micro-projectors on HMDs. Projected images were reflected back to the user by retro-reflective materials placed strategically in the environment. They also presented two collaborative applications for their HMPD prototype. Benko et al. [3] presented a system called FoveAR. A projection-based SAR technique was introduced to compensate for the limited FOV of the HMDs. Projectors projected background augmented content, and the HMDs showed this content with view dependence. Hamasaki et al. [9,10] presented a system named HySAR. They addressed the problem that previous SAR techniques could not render view-dependent components such as specular reflection. HMDs were introduced to present view-dependent components, while projectors were responsible for presenting view-independent components such as diffuse reflection. Although all of these works combined HMDs and projectors, different augmented content was generated for the HMDs and projectors respectively. Therefore, the content-generation burden for HMDs was not relieved. To our knowledge, none of the previous designs have augmented the HMD itself with projector as our ARSlice system does.

2.2. Head-Mounted Displays

HMD is one of the core AR/VR technology, and many novel HMDs have been described in the past two decades [20,37]. Most researches have focused on improving the performance of HMDs. Liu et al. [23] addressed the vergence accommodation conflict (VAC) problem and presented a novel design based on a variable focus lens. Chakravarthula et al. [5] presented a system that enables the user to adjust the focal length for both the real world and the virtual augmented content. Maimone et al. [25] presented novel designs based on phase-only holographic projection to provide a wide field of view (FOV). Jang et al. [14] introduced pupil-tracking to their HMD design and provided a large FOV and a large depth range. In terms of light sources, the HMDs above can be divided into three categories: HMDs based on a micro-display [5,23], spatial light modulator (SLM) [25], or scanned laser beams [14]. In all cases, these devices are tethered to a computer because they rely on its resources to generate virtual content. Shrinking the computer to the size of an HMD (e.g., Microsoft HoloLens) would increase the weight and energy consumption.

By contrast, our ARSlice architecture separates the content generation task to the projector end. Thus, the user end is a pure optical device with light weight, low cost, and no

energy consumption.

2.3. Visualizing CT images

Although reconstructing a 3D model of CT images has been a mature technique, visualizing CT images is still a challenging problem [19]. Because doctors not only need to see the 3D model from outside, but also its inside. To solve this problem, most researches rely on segmenting different kinds of body parts and visualizing them separately. Wu et al. [34] focused on visualizing bones. Hu et al. [11] visualized bones along with the skin. García-Berná et al. [7] focused on visualizing abdominal aorta. Jung et al. [17] designed a system that can interactively change the transparency of different parts. With the help of VR technique, Javan et al. [15] designed a framework that allows the user to explore the internal view. But it required very specialized medical knowledge, and thus mainly designed for medical education. Wang et al. [32] employed the SAR technique to visualize the CT images directly. But it shares the common disadvantage of SAR techniques, which require a large space.

Our ARSlice architecture can address the disadvantage of requiring a large space. The user can see different CT images by just moving his/her head.

3. Our ARSlice System

In this section, we introduce our ARSlice system in detail. To make it easier to understand, we first describe the hardware configuration of our ARSlice prototype and then explain the software implementation in detail.

3.1. Hardware Configuration

Our ARSlice system consists of two parts: the projector end, and the user end. The projector end is responsible for tracking the movement of the user end, generating virtual content, and projecting it to the proper location. The user end is responsible for displaying the projected images into 3D space.

3.1.1 Projector-End Equipment

The projector end of our ARSlice prototype mainly consists of a projector, a variable focus lens, a hot mirror (which only reflects infrared light), an infrared lamp, and a high-speed camera that captures infrared light. A photo of the projector-end equipment and a sketch of its corresponding configuration are shown in Fig. 2. Detailed hardware information is listed in Table 1.

The left-most infrared lamp emits infrared light to the user end. The infrared light is reflected by retro-reflective markers attached to the user-end equipment, reflected by the hot mirror, and then captured by the high-speed camera. The captured infrared images are analyzed by a PC. Thus,

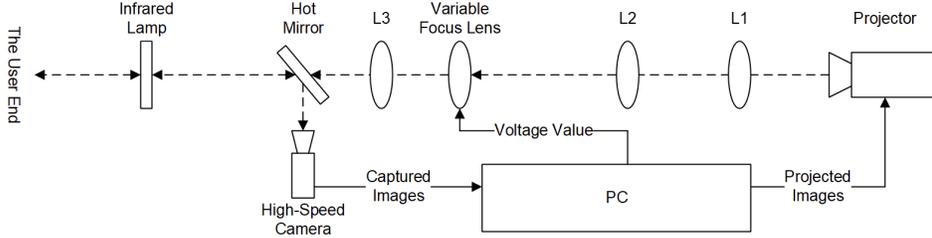
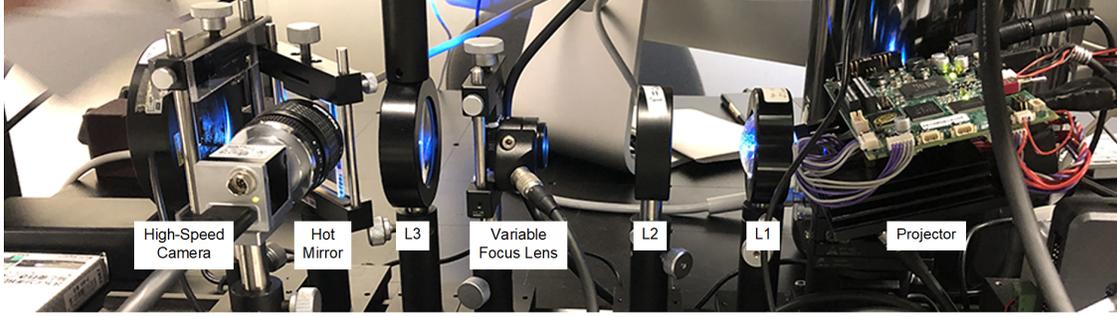


Figure 2: Projector-end equipment (above) and a sketch of its corresponding configuration (below).

Table 1: Hardware parameters for our ARSlice prototype.

Name	Model Number
High-speed projector	TI DLP, LightCrafter 4500
Variable focus lens	Optoune, EL-10-30
Lens controller	Gardasoft, TR-CL 180
DA board	Interface, LPC-361216
High-speed camera	Basler, acA640-750um

we can obtain the pose of the user end, including its location and distance.

The location is used to generate the images that are projected. In our ARSlice prototype, the scope of the projector is fixed. We realize dynamic projection by moving the location of actual content within the projected images. Actual content is projected to the location of the detected user-end equipment. All other parts of the projected images are filled with a black background to avoid unnecessary light leakage.

The distance is used to control the variable focus lens. The distance is first converted into a voltage value, and the voltage value is output with a DA adapter to adjust the focal length of the variable focus lens. By adjusting the variable focus lens, the projector projects images to the screen pad to present clear and real images onto it. With the advantage of the high-speed response of the variable focus lens, a clear image can always be projected on the screen pad, even when it moves back and forth. The presented real images are then refracted by the user-end equipment to present virtual images in 3D space.

As we can see, the key routine of the ARSlice system is

the analysis, and we explain this in Section 3.2.

3.1.2 User-end Equipment

The user-end equipment is responsible for showing the projected images in 3D space. We have designed two models of the user-end equipment.

The first model is inspired by a commercial HMD, namely, Vufine ARKit. The Vufine ARKit is an OST HMD designed for AR. It is very cheap in price (under USD \$15 on Amazon) because it is mainly built with hard papers. However, a mobile phone is required to generate images. In our ARSlice prototype, we modify the HMD by replacing the mobile phone with a screen pad accepting projected images. The light path of this model is shown in Fig. 3a. As shown in Fig. 3a, images are projected onto a screen pad. The screen pad is made of the same material as a rear-projection screen, which costs approximately USD \$35 per square meter. Images on the screen pad are then reflected by an ordinary mirror, refracted through a lens, reflected by a half mirror, and then seen by the user’s eyes. The lens in the middle acts as a magnifying lens, and thus a virtual image appears in front of the user. At the same time, the user can see the environment through the half mirror, and thus, a simple AR effect is obtained. On the screen pad, several markers made of retro-reflective material are attached to provide location information to the projector end. The cost of the screen pad and the markers is negligible. A photo of the prototype of the first model is shown in Fig. 3b, and the screen pad and the markers are labeled. In practice, the material of the screen pad has to be chosen carefully.

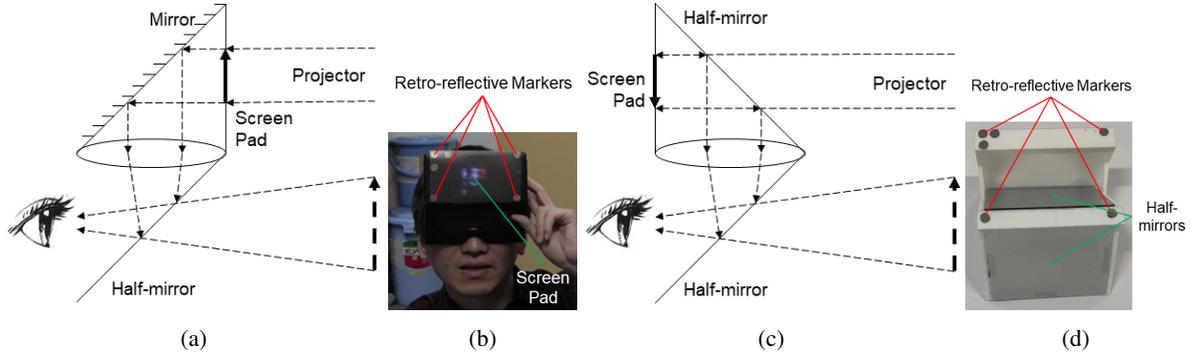


Figure 3: User-end equipment. (a & b) Model 1: a modified Vufine ARKit. (c & d) Model 2: a 3D-printed prototype. (a & c) The light path of different models. (b & d) Photos of the prototypes. Note that the projected images for model 1 and model 2 are up-down opposite.

We also design our second model to avoid using the rear-projection screen material. The light path of our second model is shown in Fig. 3c. As shown in Fig. 3c, images are projected through a half mirror onto a screen pad made of normal screen materials. Images on the screen pad are then magnified and seen by the user’s eyes. We build the prototype of the second model with 3D printing technique. A photo of it is shown in Fig. 3d, and the half mirrors and the markers are labeled.

Because the material we use to build the second model is stronger than hard papers, we can attach a camera to it to get the user-perspective view. Because the screen pad of the first model can be seen from outside, we can get a better outside view. Other than that, user experiences of the two models are the same.

3.2. Software Implementation

The software architecture of our ARSlice system is shown in Fig. 4. It consists three main modules: the *Tracker*, the *Converter*, and the *Generator*. The *Tracker* and the *Converter* module run at the same thread. Whenever an infrared image is captured by the camera, the *Tracker* module detects and tracks markers from the captured image, and then the *Converter* converts the coordinates of markers to the user pose and the voltage value. The *Generator* module run in a separated process. It obtains the user pose via inter-process communication channel, and generates the projected image. In this section, we explain these three modules.

3.2.1 Marker Tracking

The main task of the *Tracker* module is to detect and track markers in real time. Thanks to the combination of the infrared lamp and the retro-reflective markers, captured infrared images contain very little noise. Fig. 5 shows an example. The group of three points indicates the top-left cor-

ner of the screen pad. In our configuration, only one face of the screen pad can be captured, such that the positional correspondence of the other three points can be deduced. Therefore, tracking these markers is much simpler than a generic object tracking problem [28, 35].

We employed the OpenCV library to realize the *Tracker* module. The location of markers can be simply detected by the *findContours* function and by calculating the center of each contour. There could be points that misidentified as markers. We employed the Hungarian algorithm to solve the corresponding relations between markers of adjacent frames, and we used the distance between two points as the weight. In our prototype, we employ a high-speed camera with the frame rate of 500 frames per second (fps). Therefore, the movement of each marker between adjacent frames is relatively small, and the trajectory of each marker can be accurately tracked.

At run time, although the complexity of the Hungarian algorithm is $O(n^3)$, where n is the number of detected markers, the number of detected markers is small in practice. We simulated some inputs to test the elapsed time. Even when n is 120, the Hungarian algorithm can still finish in less than 1 ms. When including the time required for the contour detection algorithm, the total time is less than 2 ms, which means that the tracking routine can be sufficiently fast to process each frame at 500 Hz.

3.2.2 Calibration and Transformation

The *Converter* module converts the coordinates of the markers to the user pose and the voltage value. During this routine, several coordinate systems are involved: the camera coordinates (C), the projector coordinates (P), the coordinates for the camera images (CI), the coordinates for the projected images (PI), the coordinates for the screen pad (SP) and the coordinates for the displayed image (D). Further, several points are involved: four corner markers (M_i),

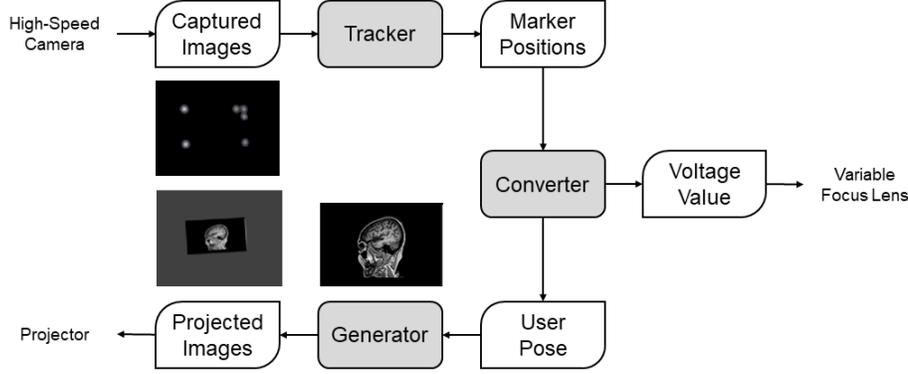


Figure 4: Software architecture of our ARSlice system.

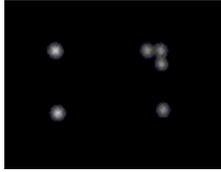


Figure 5: Example of a captured infrared image.

and four corner points of the displayed image (D_i). The relations between these coordinate systems and points are shown in Fig. 6. Note that M_i and D_i are different points, because the marker should not occlude the presented area. Especially in the second model, the markers and the screen pad are not at the same plane. In the explanation below, we use symbols such as $P(C)$ to indicate the coordinates of the point P in the C coordinate system.

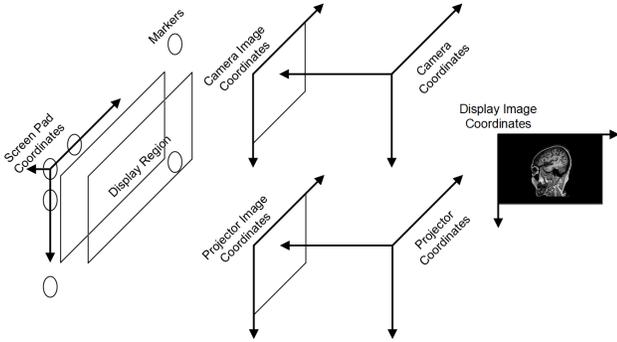


Figure 6: Relations of the involved coordinate systems and points.

At run time, the *Converter* module accepts an input of the markers under the camera coordinates $M_i(CI)$. The user pose is the displayed image under the projected coordinates $D_i(PI)$. The voltage value can be obtained from the distance from the markers to the camera, which is contained in $M_i(C)$. There are two known constants, the reso-

lution of the displayed image $D_i(D)$ and the markers under the screen pad coordinates $M_i(SP)$. To complete the above conversion, extra parameters are needed, and these are obtained with the calibration routine. The calibration routine can be divided into three stages.

Camera Calibration. The first stage is to calibrate the camera, which reveals the relationship between C and CI . We employ the classic chessboard camera-calibration method in OpenCV [36]. The resulting camera parameters are used to solve a Perspective-N-Point (PNP) problem at run time. The inputs of this PNP problem are $M_i(SP)$ and $M_i(CI)$. The *solvePnP* function of OpenCV is used, and the distance from the markers to the camera can be discovered from the result of *tvec*.

Camera-Projector Calibration. The second stage is to calibrate the relationship between CI and PI . There is an important difference between our way of doing so and previous camera-projector calibration methods [8, 18], owing to real-world measuring limitations. The main limitation is that $D_i(SP)$, which denotes the coordinates of the display region in the screen pad coordinate system, is hard to measure directly in real world. This is because we do not know the desired display region before we attach the screen pad to the user-end equipment, and it is difficult to measure after the screen pad has been attached because of the hardware configuration of the user-end equipment. In the first model, the display region is hollow and the screen pad is soft; in the second model, the display region is hidden behind a half mirror. Thus, measuring the display region generates more errors than measuring the distance normally on a rigid plane.

Thus, we designed an interactive calibration application for this stage. During this calibration stage, the user-end equipment is fixed, and $M_i(CI)$ is obtained from the *Tracker* module and recorded. Then, we manually adjust the projected image until it is coincident with the markers. Thus, we can obtain $M_i(PI)$. Similarly, we can obtain $D_i(PI)$ by manually adjusting the projected image to

the desired display region. The above procedure is repeated several times with different locations of the screen pad. At run time, we can obtain the actual coordinates of the markers $M'_i(CI)$ with the high-speed camera, and we can calculate the location of the projected image $D'_i(PI)$. The calculation is shown in Eq. 1. Here, $D_i(PI)$, $M_i(PI)$ and $M_i(CI)$ are recorded during calibration.

$$D'_i(PI) = W \cdot D_i(PI),$$

where :

$$W = M'_i(PI) \cdot M_i(PI)^{-1}, \quad (1)$$

$$M'_i(PI) = H \cdot M'_i(CI),$$

$$H = M_i(PI) \cdot M_i(CI)^{-1}$$

Distance–Voltage Calibration. The third stage is to calibrate the relationship between the user-camera distance and the control voltage of the variable focus lens. This is also done interactively. To derive a more accurate relation for practical usage, the user-camera distance is measured in the same way that the *Converter* module does so at run time. We fix the user-end equipment to multiple distances, and manually adjust the control voltage until the presented image looks clear. A series of distance–voltage pairs are thus recorded. We input these data into the curve fitting toolbox in Matlab, and this toolbox fits these data into a fitting function. The resulting fitting function is shown in Fig. 7 and the expression of it is given in Eq. 2. This equation is used by the *Converter* module to convert the distance between the camera and the markers into the voltage value.

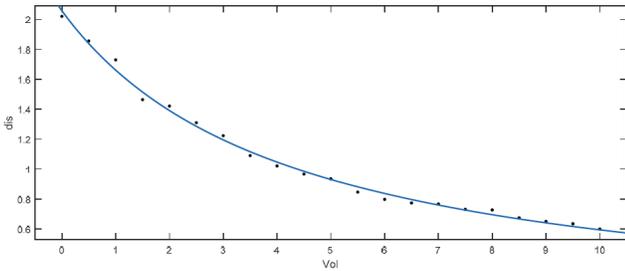


Figure 7: Relationship between the user-camera distance and the control voltage of the variable focus lens.

$$Voltage = \frac{8.89}{Distance + 0.02958} - 4.258 \quad (2)$$

3.3. Virtual Content Generation

After obtaining the user pose, we generate the projected image. This is achieved with two steps.

In the first step, we decide what image the user should see, or *user-view image*. This is achieved by designing an Unity VR application. Within the Unity VR application,

there should be a main camera that simulate the user’s view. We control the location and the orientation of the main camera based on the user pose obtained from the *Converter* module. The Unity framework will then generate the user-view image. In our motivation scenario, we design an application simply chooses a CT image based on the distance between the user and the camera. This application can be replaced to adapt to other application scenarios. In Section 4, in order to test the ability of our ARSlice system, we also design an application to project a 2D video onto the user-end equipment.

In the second step, we generate the projected image based on the user-view image. In other words, we need to apply the transformation from $D_i(D)$ to $D_i(PI)$ on the user-view image. This is achieved by modifying the Unity VR toolkit. We instrument the *Render* function within the Unity VR toolkit. Whenever the Unity VR toolkit attempts to render the user-view image, we apply the transformation on it and render the transformed image instead. When we project the transformed image, the part of image that is projected onto the screen pad will be shown to the user by the user-end equipment. The transformation will guarantee that the user can see the user-view image in the 3D space.

4. Experimental Results

In this section, we show the results brought by our AR-Slice system.

4.1. CT Visualization Application

4.1.1 AR Experience

Fig. 8 shows the AR experience results. The full video can be accessed in the supplementary video. In this experiment, the user moves forward or backward. We attach an ordinary smartphone to provide the user’s perspective result. In Fig. 8, the right column shows the user’s status, and the left column shows the user’s view at the corresponding position. We add some green lines to explicitly show that the user is moving forward or backward. As we can see, the user can see different CT images at different position. At the same time, real-world scenes can also be seen. Note that the CT images seem to be out of focus. This is because the user’s perspective camera is auto-focusing on real-world scenes, and CT images are visualized at positions different from real-world scenes.

4.1.2 VR Experience

In order to show the results more clearly, we slightly change the user-end equipment by replacing the bottom half-mirror (see Fig. 3a and Fig. 3c) with a total reflection mirror. Thus, the user can only see the projected images, and cannot see real-world scenes. This experience is similar with VR experiences.



Figure 8: Experiment of visualizing CT images with an AR experience. When the user moves forward or backward (the right column), the user can see different CT images on their corresponding position in the 3D space (the left column). At the same time, real-world scenes can also be seen. Note that the CT images seem to be out of focus, because the user’s perspective camera is auto-focusing on real-world scenes.

Fig. 9 shows the VR experience results. In this experiment, we employ the same experimental methodology, and the only change is the user-end equipment. In Fig. 9, the right column shows the user’s status, and the left column shows the user’s view at the corresponding position. We add some yellow lines to explicitly show that the user is moving forward or backward. Without real-world scenes, the user’s perspective camera can be better focused, and we can see CT images more clearly.

4.2. Numeric Metrics

In order to test the ability of our ARSlice system, we also design an application to project a 2D video onto the user-end equipment. In the following experiments, a dancing video of a famous virtual character is used to generate the projected images.

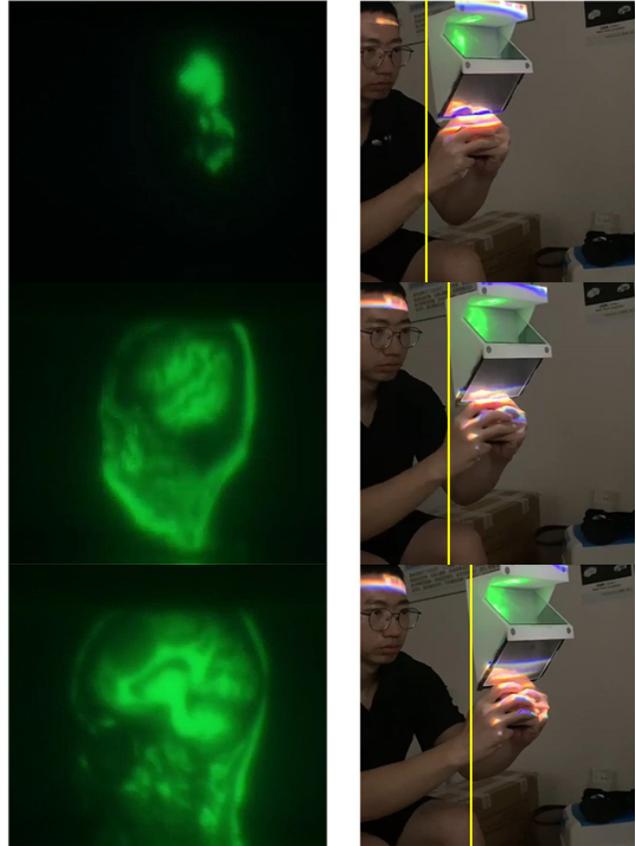


Figure 9: Experiment of visualizing CT images with a VR experience. When the user moves forward or backward, the user can see different CT images on their corresponding position in the 3D space. Without seeing real-world scenes, CT images can be seen more clearly and correctly focused.

4.2.1 Degrees of Freedom

The DoF (Degrees of Freedom) is an important metric of an AR or VR system. In order to test the DoF of our ARSlice system, we conduct three experiments.

In the first experiment, the user moves along the axis of the projector, and checks whether the video is clear. This experiment is designed to see whether the focal length can be properly changed based on the location of the user. The procedure is demonstrated in Fig. 10. As a result, the user can move from approximately 0.5 m to 2 m, with clear and continuous video. This range is owing to the capability of the variable focus lens.

In the second experiment, the user turns his head in place freely with his body facing front, and checks whether the video remains clear. This experiment is designed to see whether the projector could project images to the proper location based on the location of the user. The results are shown in Fig. 11. To better illustrate the experimental re-

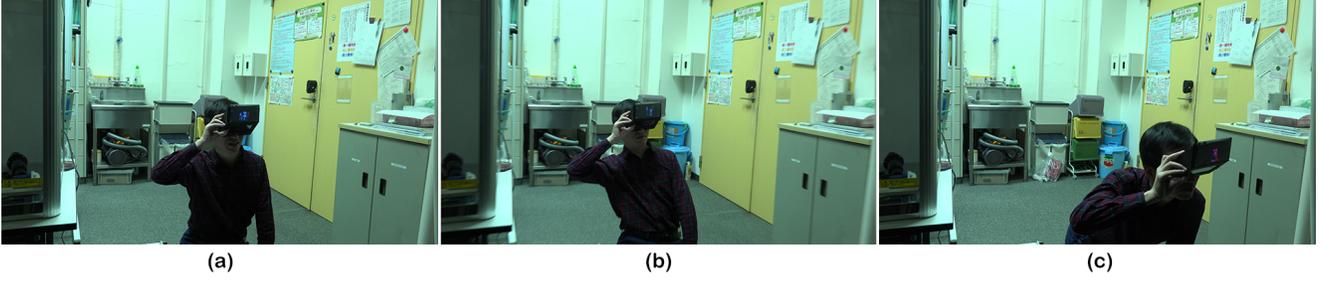


Figure 10: Experiment with varying focal length. (a) Original position, (b) distant position, and (c) near position.

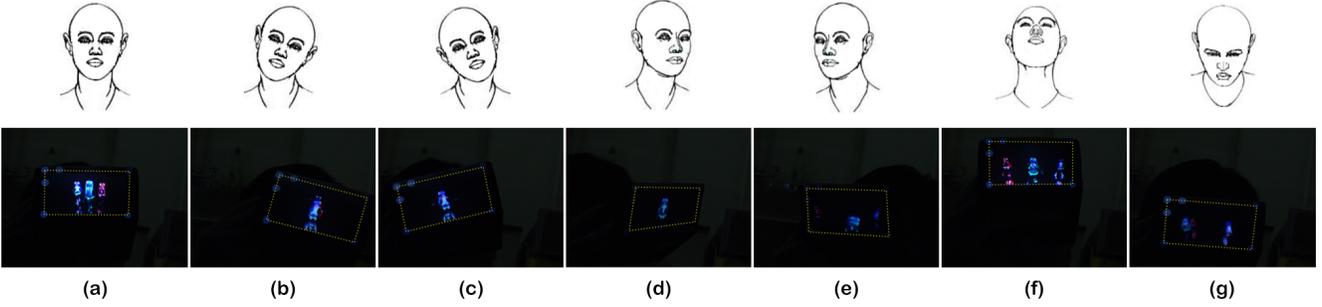


Figure 11: Experiment with user's in-place movement. The yellow dotted lines indicate the projected region. (a) Original position, (b) leaning to the left, (c) leaning to the right, (d) turning to the left, (e) turning to the right, (f) looking up, (g) looking down.

sults, we turn off the lights in the environment and record video from the direction parallel to the axis of the projector. In this way, we can see the images presented on the screen pad more clearly. Owing to the user-end equipment, these images can be seen by the user as though they are floating in front of the user at a distance of approximately 0.5 m. As a result, the user's head can move freely and the video remain clear and continuous. This result implies an angular range of at least $\pm 45^\circ$ in all directions.

In the third experiment, the user moves his head up, down, left and right, with his head facing front, and checks whether the video remains complete. This experiment is designed to see the *Field of Projection* (FOP). Here, FOP refers to the conical region that the projector covers. The projector is the vertex of the cone. FOP is determined by the configuration of the projector and the lenses. According to the equation for FOP (Eq. 3, where FOP_V indicates the vertical FOP, and FOP_H indicates the horizontal FOP), our configuration presents a FOP of $18^\circ \times 12^\circ$. This can be improved by changing the configuration of the lenses. We discuss this in Section 5.

$$\begin{aligned}
 FOP_V &= 2 \cdot \arctan\left(\frac{Height}{2 \cdot Distance}\right) \\
 FOP_H &= 2 \cdot \arctan\left(\frac{Width}{2 \cdot Distance}\right)
 \end{aligned}
 \tag{3}$$

Overall, the above experiments show that ARSlice can provide part of six degrees of freedom within the limitation of a human body.

4.2.2 Field of View

The FOV is determined by the HMD employed. According to the introduction on its homepage¹, the Vufine ARKit offers a FOV of 75° . There are alternatives to using commercial HMDs, and we discuss this in Section 5.

4.2.3 Frame Rate

During the above experiments, the video is played at 25 fps. The user can see the video clearly and continuously, indicating that the actual frame rate is higher than 25 Hz. The frame rate is mainly limited by the projector and the software routine. In our experiment, we use a projector whose refresh rate is 120 Hz. Time profiling has shown that the *Tracking* and the *Converter* module can complete their computation within 2 ms, along with the transmission latency between the *Converter* module and the Unity application. In our experiment, the Unity application can generate the projected images at more than 60 Hz. Further, there are no

¹<http://store.vufine.com/products/vufine-ar-kit>

observable “swimming” effects when the user moves. Thus, the actual frame rate can be considered above 50 Hz.

4.3. User Study

To testify our claims on user experiences, we recruited 14 participants for a user study. They all had experience with VR/AR devices. The instructor first explained that our ARSlice system is designed to help understand CT images. After experiencing our ARSlice system for 5-15 minutes, each participant was asked to fill out a questionnaire shown in Fig. 12. There were four questions. Each question was asked to be answered on a scale of 1-10, with 5 being the acceptable boundary, and higher rating indicated better experience.

Questionnaire

1. Please rate the resolution of the image you see.
2. Please rate the response speed.
3. Please rate the overall visual experience.
4. Please rate how helpful the system is in understanding CT images.

Figure 12: The questionnaire used in the user study. Each question was asked to be answered on a scale of 1-10, of which 5 is the acceptable boundary.

The user study results are organized into Fig. 13. It is a box diagram showing the minimum, first quartile, average, third quartile, and maximum for the ratings of each question. All participants gave ratings of 5 or more to the response speed (ranges from 6 to 8, with 7.6 being the average) and help in understanding CT images (ranges from 5 to 9, with 7.3 being the average). Most participants considered the resolution (12 of 14 participants) and visual experience (13 of 14 participants) acceptable. We will discuss potential improvements on resolution in Section 5.

5. Discussion and Future Work

In this section, we will discuss some of the alternative designs that might affect the resulting performance.

5.1. Alternative User-End Designs

In the model 1 of the user-end equipment, we replace the extra mobile phone with a screen pad. At first, we tried employing ordinary papers to make the screen pad, and failed. The user-end equipment practically amplifies the screen pad, and the user can clearly see the textures of ordinary papers. Then we tried removing the screen pad, and also failed. The lights from the projector directly shot into the user-end equipment and the user can only see a highlight spot. After the above attempts, we realized that the projector should present real images on the screen pad, and the

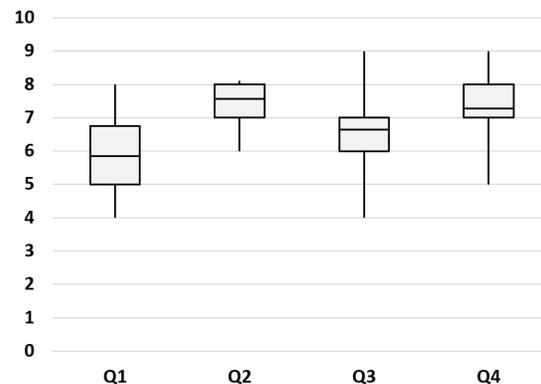


Figure 13: User study result. Participants gave high ratings to the response speed and help in understanding CT images. Most participants considered the resolution and visual experience acceptable.

lights have to penetrate through the screen pad and are further refracted into virtual images. Therefore, the screen pad is made of the same material with rear-projection screens. Furthermore, we find that the parameters of the screen pad may affect user experience. During prototyping, we experimented with two screens with different transparency. We find that the screen with its transparency of 58% performs better than the screen with its transparency of 72%. Since the light from the projector is bright enough, lower transparency does not affect the projected image much, but blocks more environmental light. As a result, better immersion is realized in the AR experience and higher contrast is realized in the VR experience. However, it is obvious that a transparency of 0% is not acceptable, because all lights including the presented real images are blocked. The optimal value of transparency needs to be further tested.

In the model 2 of the user-end equipment, the user is equivalent to looking at the screen pad in the opposite direction of the projection, and there are no need to look through the real image any more. In practice, we find that the 3D printing material can work fine as the screen pad. The main issue for the model 2 is the top half-mirror. Light from the projector first passes through the half-mirror, and then the light from the screen pad is reflected by it. Therefore, the transparency and the reflectivity of the half-mirror have to be choose more carefully.

Besides, in our ARSlice prototype, the projector is placed at the front of user, which implies that the marker and the screen pad should be oriented toward the front. The projector can be placed with other orientations, e.g., projecting from the front-above direction. It is not difficult to adjust the user-end equipment to adapt different configurations, as long as the user’s head does not block the screen pad.

5.2. Alternative Projector-End Designs

In our ARSlice prototype, the projector is fixed. Dynamic tracking and projection is achieved by software image transformation. This could be replaced with a steerable projector [33]. In such a system, the projector is steerable along with the camera. After dynamic tracking, the projector can pan and tilt to maintain the user-end equipment at the center of the FOP. Therefore, it enables a smaller FOP, and potentially higher resolution. In our current ARSlice system configuration, the projected images have a resolution of 1280x800, while the user can only see a part of it at a resolution of about 320x200. If we could replace the projector-end with a steerable or optical zoomable projector, the user could see almost the whole projected images, thus increasing the resolution to about 1280x800 or even higher. Its main issue is that the focal length cannot change fast, so it can only be applied in scenarios where the focal length is relatively fixed, or the out-of-focus issue does not affect the user experience. For our ARSlice prototype, moving the variable focus lens fast may damage it. How to design a steerable projector with extended depth-of-field is remained as our future work.

6. Conclusion

Visualizing CT images has been a challenging problem. In this paper, we have presented ARSlice, a proof-of-concept prototype that can visualize CT images in an untethered manner without wireless transmission latency. Our ARSlice system consists of two parts, the user end and the projector end. Its main feature is that, the user end is a pure optical device with light weight, low cost, and no energy consumption. By employing dynamic tracking and projection, the projector end can track the user-end equipment and project CT images onto it in real time. Our experiments demonstrate that our ARSlice system provides part of six degrees of freedom for the user, and a high frame rate. By interactively visualizing CT images into 3D space, our ARSlice system can help doctors better understand CT images.

We have also discussed further improvements that can be made. By further modifying the configuration of the user end or the projector end, the proposed ARSlice system can further provide better performance to adapt more real-world scenarios.

References

- [1] R. Azuma. A survey of augmented reality. *Presence*, 6(4):355–385, 1997. 2
- [2] Z. Bai, A. F. Blackwell, and G. Coulouris. Using augmented reality to elicit pretend play for children with autism. *IEEE Trans. Vis. Comput. Graph.*, 21(5):598–610, 2015. 2
- [3] H. Benko, E. Ofek, F. Zheng, and A. D. Wilson. Fovear: Combining an optically see-through near-eye display with projector-based spatial augmented reality. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology, UIST 2015, Charlotte, NC, USA, November 8-11, 2015*, pages 129–135, 2015. 1, 3
- [4] F. Buttussi and L. Chittaro. Effects of different types of virtual reality display on presence and learning in a safety training scenario. *IEEE Trans. Vis. Comput. Graph.*, 24(2):1063–1076, 2018. 2
- [5] P. Chakravarthula, D. Dunn, K. Aksit, and H. Fuchs. Focuser: Auto-focus augmented reality eyeglasses for both real world and virtual imagery. *IEEE Trans. Vis. Comput. Graph.*, 24(11):2906–2916, 2018. 3
- [6] J. Ehnes, K. Hirota, and M. Hirose. Projected augmentation - augmented reality using rotatable video projectors. In *3rd IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR 2004), 2-5 November 2004, Arlington, VA, USA*, pages 26–35, 2004. 2
- [7] J. A. García-Berná, J. M. Sanchez-Gomez, J. Hermanns, G. García-Mateos, and J. L. F. Alemán. Calcification detection of abdominal aorta in CT images and 3d visualization in VR devices. In *38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBC 2016, Orlando, FL, USA, August 16-20, 2016*, pages 4157–4160. IEEE, 2016. 1, 3
- [8] S. Gupta and C. O. Jaynes. Active pursuit tracking in a projector-camera system with application to augmented reality. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR Workshops 2005, San Diego, CA, USA, 21-23 September, 2005*, page 111, 2005. 6
- [9] T. Hamasaki, Y. Itoh, Y. Hiroi, D. Iwai, and M. Sugimoto. Hysar: Hybrid material rendering by an optical see-through head-mounted display with spatial augmented reality projection. *IEEE Trans. Vis. Comput. Graph.*, 24(4):1457–1466, 2018. 3
- [10] Y. Hiroi, Y. Itoh, T. Hamasaki, D. Iwai, and M. Sugimoto. Hysar: Hybrid material rendering by an optical see-through head-mounted display with spatial augmented reality projection. In *2017 IEEE Virtual Reality, VR 2017, Los Angeles, CA, USA, March 18-22, 2017*, pages 211–212, 2017. 3
- [11] Z. Hu, J. Zou, J. Gui, J. Rong, Y. Li, D. Xi, and H. Zheng. Real-time visualization and interaction of three-dimensional human CT images. *J. Comput.*, 5(9):1335–1342, 2010. 3
- [12] H. Hua, C. Gao, L. D. Brown, N. Ahuja, and J. P. Rolland. Using a head-mounted projective display in interactive augmented environments. In *4th International Symposium on Augmented Reality (ISAR 2001), 29-30 October 2001, New York, NY, USA*, page 217, 2001. 3
- [13] A. Ibrahim, B. Huynh, J. Downey, T. Höllerer, D. Chun, and J. O’Donovan. Arbis pictus: A study of vocabulary learning with augmented reality. *IEEE Trans. Vis. Comput. Graph.*, 24(11):2867–2874, 2018. 2
- [14] C. Jang, K. Bang, S. Moon, J. Kim, S. Lee, and B. Lee. Retinal 3d: augmented reality near-eye display via pupil-tracked light field projection on retina. *ACM Trans. Graph.*, 36(6):190:1–190:13, 2017. 3
- [15] R. Javan, A. Rao, B. S. Jeun, A. Herur-Raman, N. Singh, and P. Heidari. From CT to 3d printed models, serious gaming, and virtual reality: Framework for educational 3d visualization.

- tion of complex anatomical spaces from within - the pterygopalatine fossa. *J. Digit. Imaging*, 33(3):776–791, 2020. 3
- [16] B. R. Jones, R. Sodhi, R. H. Campbell, G. Garnett, and B. P. Bailey. Build your world and play in it: Interacting with surface particles on complex objects. In *9th IEEE International Symposium on Mixed and Augmented Reality, ISMAR 2010, Seoul, Korea, 13-16 October 2010*, pages 165–174, 2010. 2
- [17] Y. Jung, J. Kim, and D. D. Feng. Dual-modal visibility metrics for interactive PET-CT visualization. In *Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBC 2012, San Diego, CA, USA, August 28 - September 1, 2012*, pages 2696–2699. IEEE, 2012. 3
- [18] H. Kato and M. Billinghurst. Marker tracking and HMD calibration for a video-based augmented reality conferencing system. In *2nd IEEE and ACM International Workshop on Augmented Reality, IWAR '99, San Francisco, CA, USA, October 20-21, 1999*, pages 85–94, 1999. 6
- [19] R. Kikinis, M. E. Shenton, D. V. Iosifescu, R. W. McCarley, P. Saiviroonporn, H. H. Hokama, A. Robatino, D. Metcalf, C. Wible, C. M. Portas, R. M. Donnino, and F. A. Jolesz. A digital brain atlas for surgical planning, model-driven segmentation, and teaching. *IEEE Trans. Vis. Comput. Graph.*, 2(3):232–241, 1996. 3
- [20] K. Kim, M. Billinghurst, G. Bruder, H. B. Duh, and G. F. Welch. Revisiting trends in augmented reality research: A review of the 2nd decade of ISMAR (2008-2017). *IEEE Trans. Vis. Comput. Graph.*, 24(11):2947–2962, 2018. 3
- [21] P. Lincoln, G. Welch, A. Nashel, A. Ilie, A. State, and H. Fuchs. Animatronic shader lamps avatars. In *Science & Technology Proceedings, 8th IEEE International Symposium on Mixed and Augmented Reality 2009, ISMAR 2009, Orlando, Florida, USA, October 19-22, 2009*, pages 27–33, 2009. 1, 2
- [22] P. Lincoln, G. Welch, A. Nashel, A. State, A. Ilie, and H. Fuchs. Animatronic shader lamps avatars. *Virtual Reality*, 15(2-3):225–238, 2011. 1, 2
- [23] S. Liu, D. Cheng, and H. Hua. An optical see-through head mounted display with addressable focal planes. In *7th IEEE and ACM International Symposium on Mixed and Augmented Reality, ISMAR 2008, Cambridge, UK, 15-18th September 2008*, pages 33–42, 2008. 3
- [24] X. Liu, C. Vlachou, F. Qian, C. Wang, and K. Kim. Firefly: Untethered multi-user VR for commodity mobile devices. In A. Gavrilovska and E. Zadok, editors, *2020 USENIX Annual Technical Conference, USENIX ATC 2020, July 15-17, 2020*, pages 943–957. USENIX Association, 2020. 2
- [25] A. Maimone, A. Georgiou, and J. S. Kollin. Holographic near-eye displays for virtual and augmented reality. *ACM Trans. Graph.*, 36(4):85:1–85:16, 2017. 3
- [26] C. Menk and R. Koch. Truthful color reproduction in spatial augmented reality applications. *IEEE Trans. Vis. Comput. Graph.*, 19(2):236–248, 2013. 2
- [27] G. Narita, Y. Watanabe, and M. Ishikawa. Dynamic projection mapping onto deforming non-rigid surface using deformable dot cluster marker. *IEEE Trans. Vis. Comput. Graph.*, 23(3):1235–1248, 2017. 1, 2
- [28] H. Pirsiavash, D. Ramanan, and C. C. Fowlkes. Globally-optimal greedy algorithms for tracking a variable number of objects. In *The 24th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2011, Colorado Springs, CO, USA, 20-25 June 2011*, pages 1201–1208, 2011. 5
- [29] R. Raskar, G. Welch, K. Low, and D. Bandyopadhyay. Shader lamps: Animating real objects with image-based illumination. In *Proceedings of the 12th Eurographics Workshop on Rendering Techniques, London, UK, June 25-27, 2001*, pages 89–102, 2001. 1
- [30] B. Schwerdtfeger, D. Pustka, A. Hofhauser, and G. Klinker. Using laser projectors for augmented reality. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology, VRST 2008, Bordeaux, France, October 27-29, 2008*, pages 134–137, 2008. 2
- [31] J. Stauffert, F. Niebling, and M. E. Latoschik. Simultaneous run-time measurement of motion-to-photon latency and latency jitter. In *IEEE Conference on Virtual Reality and 3D User Interfaces, VR 2010, Atlanta, GA, USA, March 22-26, 2020*, pages 636–644. IEEE, 2020. 2
- [32] L. Wang, H. Xu, S. Tabata, Y. Hu, Y. Watanabe, and M. Ishikawa. High-speed focal tracking projection based on liquid lens. In *SIGGRAPH 2020: Special Interest Group on Computer Graphics and Interactive Techniques Conference, Emerging Technologies, Virtual Event, USA, August, 2020*, pages 15:1–15:2. ACM, 2020. 1, 3
- [33] A. Wilson, H. Benko, S. Izadi, and O. Hilliges. Steerable augmented reality with the beamatron. In R. Miller, H. Benko, and C. Latulipe, editors, *The 25th Annual ACM Symposium on User Interface Software and Technology, UIST '12, Cambridge, MA, USA, October 7-10, 2012*, pages 413–422. ACM, 2012. 11
- [34] J. Wu, A. Belle, R. H. Hargraves, C. Cockrell, Y. Tang, and K. Najarian. Bone segmentation and 3d visualization of CT images for traumatic pelvic injuries. *Int. J. Imaging Syst. Technol.*, 24(1):29–38, 2014. 3
- [35] J. Xing, H. Ai, and S. Lao. Multi-object tracking through occlusions by local tracklets filtering and global tracklets association with detection responses. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, pages 1200–1207, 2009. 5
- [36] Z. Zhang. Flexible camera calibration by viewing a plane from unknown orientations. In *ICCV*, pages 666–673, 1999. 6
- [37] F. Zhou, H. B. Duh, and M. Billinghurst. Trends in augmented reality tracking, interaction and display: A review of ten years of ISMAR. In *7th IEEE and ACM International Symposium on Mixed and Augmented Reality, ISMAR 2008, Cambridge, UK, 15-18th September 2008*, pages 193–202, 2008. 3