LIGHTWEIGHT AND SUFFICIENT TWO VIEWPOINT CONNECTIONS FOR AUGMENTED REALITY

A Dissertation

Submitted to the Faculty

of

Purdue University

by

Chengyuan Lin

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

May 2020

Purdue University

West Lafayette, Indiana

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF DISSERTATION APPROVAL

Dr. Voicu S. Popescu, Chair

Department of Computer Science

Dr. Daniel G. Aliaga

Department of Computer Science

Dr. Christoph M. Hoffmann

Department of Computer Science

Dr. Xavier M. Tricoche

Department of Computer Science

Approved by:

Dr. Christopher W. Clifton

Head of the School Graduate Program

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

Figure Page

Figure Page

ABBREVIATIONS

AI	artificial intelligence
ANOVA	analysis of variance
API	application programming interface
AR	augmented reality
CPU	central processing unit
DR	diminished reality
FOV	field of view
HD	high definition
HMD	head-mounted display
IQR	interquartile range
ISO	international organization for standardization
LCD	liquid-crystal display
LIDAR	light detection and ranging
NASA	national aeronautics and space administration
OLED	organic light-emitting diode
SLAM	simultaneous localization and mapping
RGB	red, green and blue
SSQ	simulator sickness questionnare
TLX	task load index
VR	virtual reality

ABSTRACT

Lin, Chengyuan Ph.D., Purdue University, May 2020. Lightweight and Sufficient Two Viewpoint Connectionsfor Augmented Reality. Major Professor: Voicu S. Popescu.

Augmented Reality (AR) is a powerful computer to human visual interface that displays data overlaid onto the user's view of the real world. Compared to conventional visualization on a computer display, AR has the advantage of saving the user the cognitive effort of mapping the visualization to the real world. For example, a user wearing AR glasses can find a destination in an urban setting by following a virtual green line drawn by the AR system on the sidewalk, which is easier to do than having to rely on navigational directions displayed on a phone. Similarly, a surgeon looking at an operating field through an AR display can see graphical annotations authored by a remote mentor as if the mentor actually drew on the patient's body.

However, several challenges remain to be addressed before AR can reach its full potential. This research contributes solutions to four such challenges. A first challenge is achieving visualization continuity for AR displays. Since truly transparent displays are not feasible, AR relies on simulating transparency by showing a live video on a conventional display. For correct transparency, the display should show exactly what the user would see if the display were not there. Since the video is not captured from the user viewpoint, simply displaying each frame as acquired results in visualization discontinuity and redundancy. A second challenge is providing the remote mentor with an effective visualization of the mentee's workspace in AR telementoring. Acquiring the workspace with a camera built into the mentee's AR headset is appealing since it captures the workspace from the mentee's viewpoint, and since it does not require external hardware. However, the workspace visualization is unstable as it changes frequently, abruptly, and substantially with each mentee head motion.

A third challenge is occluder removal in diminished reality. Whereas in conventional AR the user's visualization of a real world scene is augmented with graphical annotations, diminished reality aims to aid the user's understanding of complex real world scenes by removing objects from the visualization. The challenge is to paint over occluder pixels using auxiliary videos acquired from different viewpoints, in real time, and with good visual quality. A fourth challenge is to acquire scene geometry from the user viewpoint, as needed in AR, for example, to integrate virtual annotations seamlessly into the real world scene through accurate depth compositing, and shadow and reflection casting and receiving.

Our solutions are based on the thesis that images acquired from different viewpoints should not always be connected by computing a dense, per-pixel set of correspondences, but rather by devising custom, lightweight, yet sufficient connections between them, for each unique context. We have developed a self-contained phone-based AR display that aligns the phone camera and the user by views, reducing visualization discontinuity to less than 5% for scene distances beyond 5 m. We have developed and validated in user studies an effective workspace visualization method by stabilizing the mentee first-person video feed through reprojection on a planar proxy of the workspace. We have developed a real-time occluder in-painting method for diminished reality based on a two-stage coarse-then-fine mapping between the user and the auxiliary view. The mapping is established in time linear with occluder contour length, and it achieves good continuity across the occluder boundary. We have developed a method for 3D scene acquisition from the user viewpoint based on single-image triangulation of correspondences between left and right eye corneal reflections. The method relies on a subpixel accurate calibration of the catadioptric imaging system defined by two corneas and a camera, which enables the extension of conventional epipolar geometry for a fast connection between corneal reflections.

1 INTRODUCTION

Augmented reality (AR) technologies aim to improve a user's visualization of the real world by overlaying 3D computer graphics annotations over the user's field of view. These annotations are superimposed onto real world objects in order to help the user's understanding of the scene [1].

AR interfaces can be classified into two main categories based on the type of display they employ. Hand-held AR uses a small display such as a computer tablet or phone that acquires the real world with the back-facing video camera and shows it to the user together with annotations. Head-mounted display (HMD) AR uses a headset worn by the user. In the case of optical see-through AR HMD's, the user sees the real world directly, through a transparent glass. In the case of video see-through AR HMD's, the user sees a live video stream of the real world.

AR technology has proven to be useful in many scenarios. One such scenario is surgical telementoring [2, 3], which is a promising approach for transmitting surgical expertise over large geographic distances promptly and efficiently, allowing a local general surgeon to provide urgent care without the delay of transporting the patient or the expert surgeon. The conventional approach in surgical telementoring is based on telestrators that allow the remote mentor to annotate a live video feed of the surgery, and the annotated video feed is shown to the mentee on a nearby display. This requires the mentee to continually shift focus away from the surgical field in order to consult the nearby display, to memorize the instructions provided by the mentor, and to transfer them, from memory, in the context of the actual surgical field, which can lead to surgery delays and even errors. By contrast, AR allows integrating the mentor-authored annotations directly into the field of view of the mentee. The mentee sees the annotations as if the mentor actually drew them onto the surgical

field, avoiding focus shifts and alleviating the high cognitive load of having to map annotations to the surgical field, avoiding surgery delays and errors.

AR is not only capable of adding virtual objects to the scene, but in some use cases, AR can be also called upon to remove some elements from the real world to facilitate its understanding by the user. This specialization of AR technology is called diminished reality (DR) [4]. DR covers real objects with images of their occluded background to make the objects virtually invisible to the user. One goal of DR applications is the removal of an occluder that hides a part of the scene that is of interest to the user. Most DR displays are video see-through for their advantage of providing perfect opacity compared to optical see-through displays. Indeed, an optical see-through display cannot erase a bright surface when the object it occludes is dark.

One example DR application is the visualization of an object that moves through a crowded scene, such as a pedestrian or a car moving on a busy street. As the target moves, the target can become temporarily occluded by other objects in the scene. The conventional solution is to acquire the scene with multiple surveillance cameras and to switch between video feeds to keep sight of the target. However, constantly switching between different viewpoints requires the additional cognitive load of establishing spatial mappings between consecutive viewpoints. For example, one can easily misjudge the trajectory of the target when switching from one camera to a second camera with a considerably different view direction. Furthermore, the remote visualization of the scene requires transmitting all the feeds to the user, which strains the network connection. On the other hand, the DR approach avoids these shortcomings by erasing the occluders from the main user view, keeping the target visible at all times. The DR visualization demands a lower cognitive load than constantly switching between views, and the continuity of the target trajectory is preserved. Furthermore, the DR approach effectively integrates the many video feeds into a single non-redundant video feed, lowering the bandwidth requirement considerably.

Figure 1.1.: An example of dual-view perceptual issue. Discontinuity and redundancy occur between a hand-held AR display and the scene.

1.1 AR Challenges

In order for AR to reach its full potential in applications, several challenges have to be overcome.

A first AR challenge is to remove the discontinuity at the boundary of hand-held AR displays. As the augmented video on the display is not adapted to the user's viewpoint, this leads to a discontinuity and a redundancy between the parts of the scene viewed directly by the user and the parts viewed on the display (Figure 1.1). The problem, also known as the *dual-view perceptual issue* [5], places an additional cognitive load on the user who has to map the information given, from the context of the display, to the context of the scene observed directly. Also, the user has to switch back and forth between the display and the scene to translate the information received on the display to the real world. Furthermore, relying on the device-perspective view of the scene shown on the display can lead to incorrect depth interpretation and an inability to properly estimate distances.

The challenge will go away once there will be truly transparent hand-held displays. However, for the foreseeable future, phone and tablet components such as the battery, the CPU, and the GPU will remain opaque and of considerable size. The shortest path towards a truly transparent hand-held display might be a tablet design with a transparent screen with all opaque components moved away from the screen, but no such tablet exists so far.

In order to make conventional tablets and phones work effectively as an AR platform, what is needed is to simulate the transparency of the display by showing on the screen exactly what the user would see in the absence of the display. This would turn the display into a frame through which the user can freely observe the scene, with the benefit of AR guidance rendered on top. Tablets and phones do have a back-facing video camera that acquires the real world the user sees. In order to warp the frames acquired by the device camera to the user viewpoint, two pieces of information are required: the position of the user head, and the geometry of the scene. This way the scene can be rendered from the user's viewpoint, projectively texture mapped with the video frame, to simulate a transparency effect.

A second AR challenge is to convey the workspace of one user to a remote collaborator. For example, in the context of telementoring, the mentor relies on high quality visualization of the mentee's workspace in order to provide effective guidance (Figure 1.2).

One approach is to acquire the workspace with a static auxiliary video camera and to send its video feed to the mentor. This approach requires additional hardware, and furthermore, the auxiliary camera captures the workspace from a view that is substantially different from the mentee's view. This view discrepancy reduces telementoring effectiveness, as the mentor can best guide the mentee when the mentor sees what the mentee sees. Furthermore, the mentor annotations are most effective when they are drawn in the mentee's frame of reference. For example, the mentee might need help with a part of the workspace that is not visible to the mentor due to

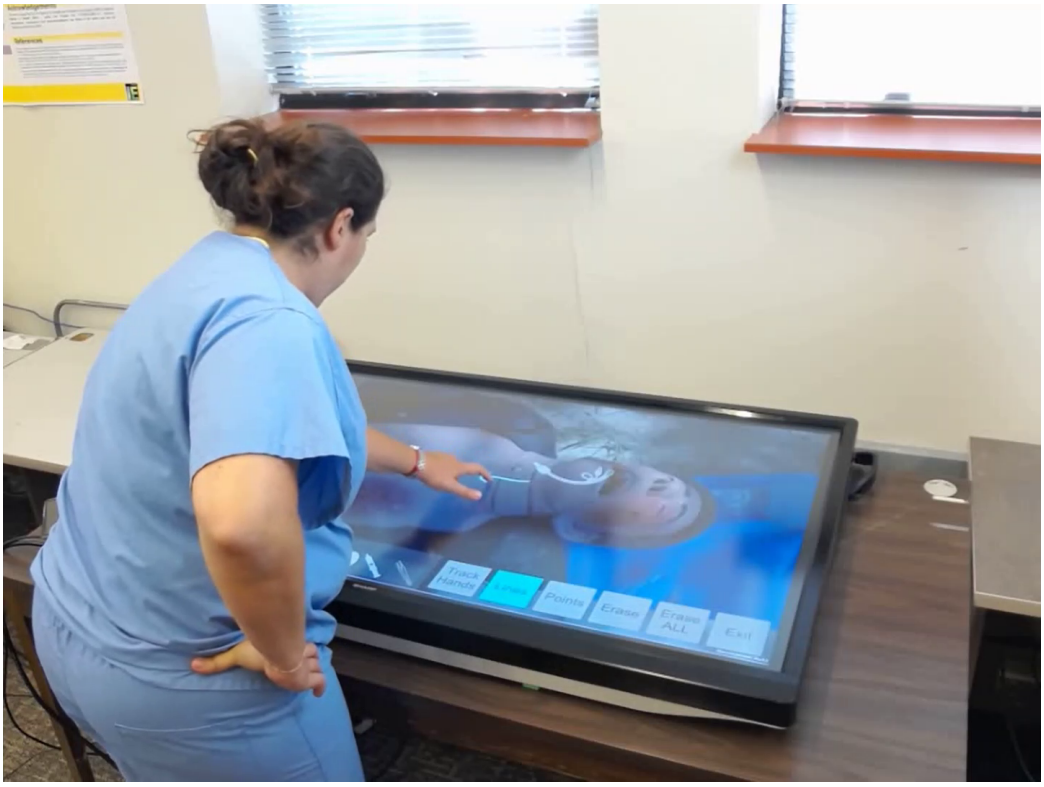


Figure 1.2.: Mentor authorizing annotations on video feed of mentee's workspace. Effective guidance can only be provided when high quality visualization of the workspace is available.

occlusions, or, conversely, the mentor might annotate a part of the workspace that is not visible to the mentee.

Another approach is to use a camera mounted on the mentee head. With the advancement of technology, self-contained HMDs typically incorporate an on-board scene-looking camera, which can capture the workspace from a viewpoint close to the mentee's viewpoint. Such a camera provides the mentor with a visualization of the workspace that matches the mentee's visualization and therefore has the potential to facilitate effective telementoring. However, simply providing the mentee first-person video directly to the mentor is inadequate. As the mentee changes head position and view direction, the mentor's visualization of the workspace changes frequently and

substantially, which adversely affects the mentor's understanding of the scene, the instructional quality of the mentor annotations, and ultimately the performance of the mentee.

By combining the advantages of the approaches above, another approach is to first capture the geometry of the workspace, and then to render the workspace geometry from the mentor viewpoint, projectively texture mapped with the mentee's first-person video frame. If the mentor viewpoint matches the mentee's viewpoint, the method has the advantage of a good view agreement between mentor and mentee, a prerequisite for effective telementoring. Furthermore, since the mentor viewpoint is stationary, or under the mentor's control, the approach effectively decouples the mentor view from the mentee view, avoiding the sudden view changes that plague using the mentee first person directly at the mentor. However, high-quality real time depth acquisition and geometric modeling of a dynamic scene with complex geometric and reflectance properties remains a challenging problem. Furthermore, the workspace has to be acquired from multiple viewpoints to avoid the objectionable artifacts caused by disocclusion errors, which increases the complexity of the system, making it inappropriate for deployment outside laboratories, in austere conditions.

A third AR challenge is to improve the user's visualization of an object of interest, i.e. a target, in a complex scene with occluders. The goal is to keep the target always visible, free of occlusions. Occlusion removal is a difficult problem, which entails finding the footprint of the occluder in the user's view, finding the target footprint in an auxiliary view, and transferring the auxiliary view target pixels to the user view in order to effectively erase the occluder. One approach is again 3D scene acquisition. Once the geometry of the scene is known, segmenting the occluder becomes easier, and rendering the scene without the occluder from the user's viewpoint provides exactly what the user would see without the occluder. However, as discussed above, the real time acquisition of a dynamic scene with complex geometry is challenging, especially when the equipment cost has to be minimized. Figure 1.3 illustrates a scenario where an occluder hinders keeping track of a moving target across the scene.

Figure 1.3.: An object person on a scooter (highlighted by red circles) moving across the scene. Tracing the target is difficult when the target is occluded by other objects in the scene.

1.2 Thesis Statement

Our thesis is based on an insight that promises to address the three challenges above. A dense, per-pixel connection between two images, acquired from different viewpoints, does allow approximating the scene geometry, and the geometry does provide a bidirectional mapping between the two images. However, such a dense connection is difficult to establish, typically requires extra hardware, and not suitable for austere settings. Furthermore, it is not sufficient to address adequately the AR challenges above.

Regarding the simulated transparent display challenge, one could use a hand-held display with two on-board back-facing cameras to acquire the scene geometry with conventional stereo. Similarly, one could use frames acquired from different locations by a single back-facing camera of the hand-held display, and reconstruct the scene geometry with conventional structure from motion. However, this requires establishing a dense set of correspondences between two images, i.e. $O(wh)$ correspondences for images of resolution $w \times h$. Even at this high computational cost, the resulting correspondence map will only cover the intersection of the two images. Furthermore, mapping camera's view using this correspondence map to the user's viewpoint will leave holes where the user sees parts of the scene that are not seen by the camera. Figure 1.4 illustrates the problem: O_1 is the user viewpoint, O_2 is the display's back-facing camera. FG is the display, around which the discontinuity needs to be removed. Even if camera O_2 is enhanced with a perfect per-pixel depth acquisition, it will not sample the green object between B and D, leaving a gap, i.e. a disocclusion error, between B and C. What is needed is a mapping from the on-board camera to the user viewpoint that can be computed quickly and that preserves the quality of the image.

Similarly, regarding the challenge of providing to the mentor an effective visualization of the mentee's workspace, an incomplete acquisition of the workspace geometry will result in the same disocclusion artifacts. What is needed is a mapping from

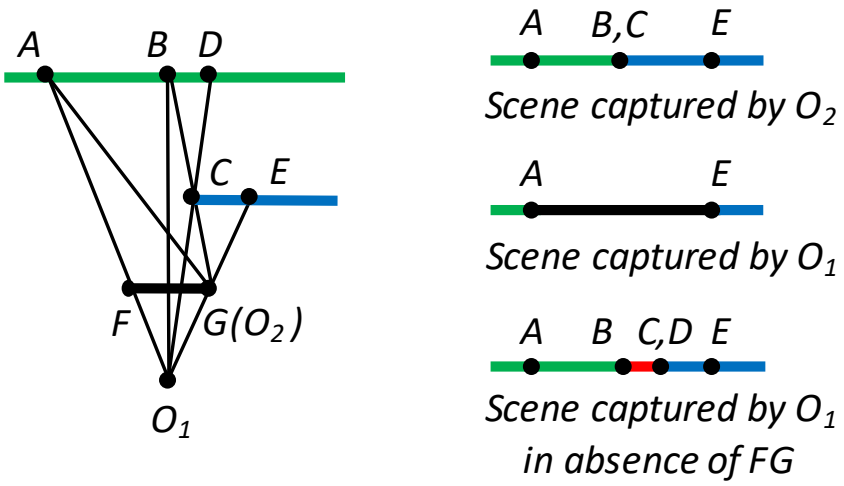


Figure 1.4.: Scene geometry (left) and views captured from two different viewpoints (right) (e.g. a user and a camera). O_2 samples the green object from the left until B, and then the blue object from C towards the right (right, top). O_1 is affected by the occluder FG, it sees the green object from the left until A, then the occluder, and then the blue object from E to the right (right, middle). Even if the view from O_2 comes with perfect per-pixel depth, projecting the 3D samples onto O_1 will leave a gap between the projection of B and the projection of D, due to the occlusion from the blue object.

the mentee first person camera feed to a stable mentor viewpoint that results in a high-quality visualization of the workspace for effective telementoring. Figure 1.5 illustrates the problem, where O_1 is the mentor viewpoint, and O_2 is the mentee viewpoint, also the head-mounted camera's viewpoint. The disparity between two viewpoints leads to part of the scene BD not sampled by the camera, translates to artifacts because of disocclusion errors in BC. Indeed, a mapping from the moving mentee viewpoint to a static viewpoint for the mentor computed in real-time will serve better the purpose of conveying the workspace effectively.

Finally, regarding the challenge of painting over an occluder to reveal the target behind it, even if the feed of the auxiliary camera is enhanced with per-pixel depth,

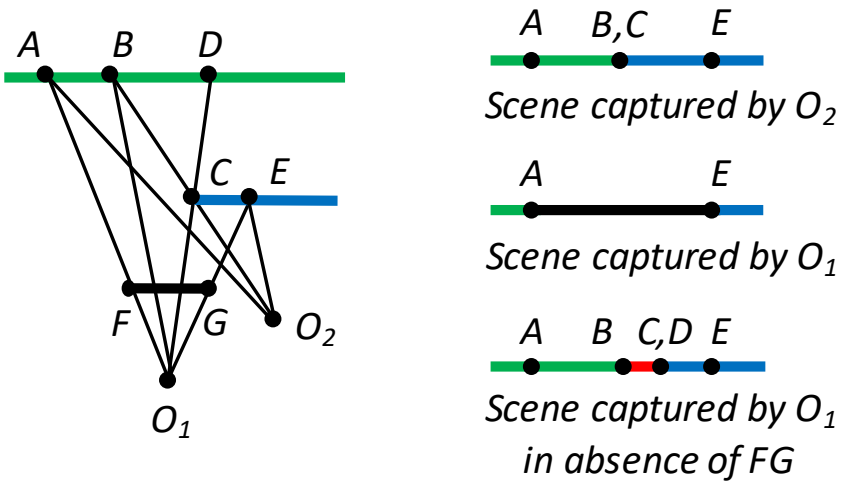


Figure 1.5.: Scene geometry (left) and views captured from two different viewpoints (right) (e.g. two cameras at different locations).

which comes at the high cost of passive (i.e. stereo or structure from motion) or active depth acquisition (i.e. LIDAR), reprojecting the feed to the user's viewpoint can leave gaps where the occluder is not erased. Again, referring to Figure 1.5, with the user's viewpoint at O_1 and the auxiliary camera at O_2, the disocclusion error in BC cannot be avoided by methods based on geometry acquisition. What is needed is a mapping from the auxiliary camera view to the user view that allows erasing the entire occluder, with good continuity between the inpainted target and the surrounding background.

Thesis Statement

For some Augmented Reality problems that require establishing a connection between two images with a different viewpoint, the traditional approach of computing a dense, per-pixel set of correspondences between the images is both challenging and insufficient; instead, a custom connection can be designed to provide an inexpensive yet effective solution to each problem.

Figure 1.6.: Our prototype of the transparent display. There is no discontinuity around the display and no redundancy between the AR display and the scene.

Our thesis advocates abandoning the pursuit of $O(wh)$ mappings between images of resolution $w \times h$ and instead designing $O(w)$ or even $O(1)$ mappings that preserve image quality and therefore solve each problem well.

To remove the discontinuity at the boundary of a video see-through display, we assume the scene geometry is infinitely far away. This allows us to establish the connection between the user viewpoint and the viewpoint of the on-board back-facing camera in constant time $O(1)$, a negligible overhead for the regular rendering pipeline. We prove that the achieved transparency effect error is below 5% when the scene is farther than 6 m, and we demonstrate a compelling simulated transparency effect on a self-contained and compact mobile phone, without any additional devices. Figure 1.6 shows one example frame of our transparent display, where the discontinuity around the contour of the display is removed, and there is no redundancy between the scene viewed on the display and viewed directly.

To improve the effectiveness of telementoring, we propose to use a plane to approximate the workspace geometry. This also allows us to establish the connection

Figure 1.7.: Our telementoring system in operation. Left column showcases third-person views, right column shows the corresponding stabilized view and the unstabilized view (raw camera feed).

between mentor viewpoint and mentee viewpoint in constant time $O(1)$. We then projectively texture map the mentee video feed onto the planar approximation of the workspace geometry, providing an effective real-time visualization of the workspace to the mentor. The visualization is of high quality, i.e. without distortions due to inadequate geometric approximation, and without tears due to disocclusion errors. All scene lines project to lines in the visualization. All these properties contribute to the effectiveness of telementoring. Figure 1.7 shows our system for AR telementoring implementing this method.

To remove an occluder from a video feed that captures a real world scene in real time, we propose an $O(w)$ mapping from the view of an auxiliary camera to the user view, which is sufficient to paint the target over the occluder. The mapping is based on a global rotation and a contour pixel correspondence refinement. Given a primary, i.e. user, view and a secondary, i.e. auxiliary camera, view of a target, our method first computes an approximate global mapping between views, then splices in the occluder footprint with pixels from the secondary video feed. The result is a multi-perspective visualization, where the scene surrounding the occluder is shown conventionally, from the user viewpoint, and the scene behind the occluder is shown from the second camera viewpoint. Switching abruptly from one perspective to the other at the occluder contour would create a discontinuity. Instead, our method connects the two perspectives seamlessly with a local mapping that achieves a gradual transition from one viewpoint to the other (Figure 1.8).

Of course, our thesis does not advocate that there are no applications where depth acquisition of the 3D scene based on a dense set of correspondences between two or more images with different viewpoints is the best approach. Even in the context of AR, a measure of scene geometry is important, for example when attempting to integrate the computer graphics annotations of the real world scene in a way that is as convincing as possible. More specifically, in the AR surgical telementoring context, the annotations have to be placed at an appropriate depth, on the operating

Figure 1.8.: An object person on a scooter (highlighted by red circles) moving across the scene. Tracing the target is made easy with the occluder rendered semi-transparently using our method. The target is free from occluder (pedestrians, cars) in the scene.

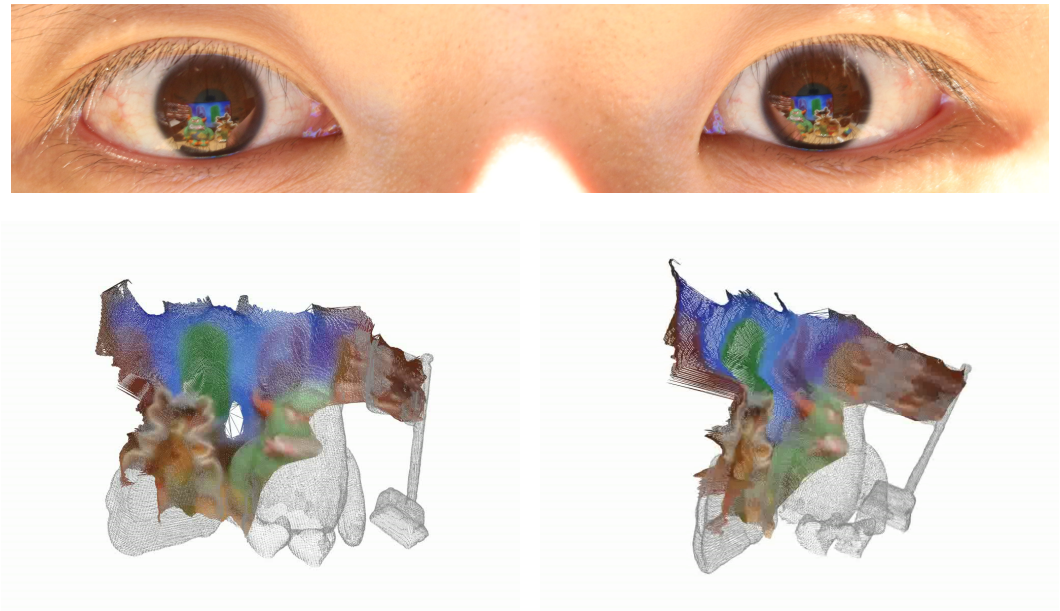


Figure 1.9.: An example of scene reconstruction using corneal reflections. Top row shows the input image, cropped to eye region, bottom row shows the output reconstruction (colored wireframe) from two corneal reflections, aligned with truth geometry (gray points).

field surfaces, and they need to be occluded appropriately by instruments or surgeon hands.

An appealing solution is to acquire a geometry and color model of the scene, from the user's viewpoint, with minimal hardware. One possibility that we have investigated is to use the catadioptric system defined by a camera and the user's corneas. The interpupillary distance provides a baseline that introduces disparity between the left and right eye reflection, which could be exploited to recover the geometry and color of the scene, from the user viewpoint, with a single camera. The approach brings important challenges such as modeling the corneas-plus-camera catadioptric system, establishing correspondences between the two reflections, as well as acquiring the cornea reflections with sufficient resolution.

Reflections on convex surfaces, such as corneas, are particularly rich in information, as the divergent reflected rays sample the scene with a large field of view. From a single image of the user's eyes, we are able to recover the 3D geometry of the scene. We designed a procedure for calibrating the catadioptric model defined by two corneal spheres and a camera. We managed to calibrate precisely the position of the eyes with respect to the camera with subpixel accuracy. We then use the corneal imaging model to recover dense depth through stereo matching and further generate a 3D model of the scene geometry, captured from the user perspective. This user perspective 3D reconstruction could benefit AR applications by reducing artifacts due to different viewpoints. Figure 1.9 shows the reconstruction result using our method of a desktop scene using a corneal image.

2 SIMULATED DISPLAY TRANSPARENCY ON HAND-HELD, SELF-CONTAINED MOBILE DEVICES

2.1 Introduction

The advancement on the computational power of hand-held smartphones and tablets has opened the door to augmented reality (AR) applications, which overlay information onto the real world without the need for a dedicated device. For example, a technician can receive graphical guidance from a table in front of a car engine. A tourist can use their smartphone to receive navigational guidance overlaid directly onto the view of the streets. In the case of telementoring, a general surgeon can received guidance from a tablet placed above an operating field streaming annotations authorized from an expert surgeon that is physically thousands of miles away.

Nowadays, hand-held mobile devices video camera captures the real world in high resolution and show to the user on a high-quality display. Overlaying textual and graphical annotations directly on this video feed provides the user with information about the scene with more situational awareness. In this way, this additional information is easy to parse, as each piece of information is right on top of the real world element that it annotates. However, the part of the scene captured by the camera, and shown on the display, usually has a larger field of view than that should be seen if the display was not there. In other words, the video shown on the display is not adapted to the user viewpoint, resulting in visual discontinuity and redundancy between the parts of the scene viewed directly and the parts viewed on the display. In this case, the user has to map the guidance from the context of the display, to the context of the real world scene observed directly, placing an additional cognitive load on the user. This is especially true when the real world scene is constantly changing (for example, as user move) and that the information need to keep updating accord-

ingly. The user has to constantly switch back and forth between the world seen on the display and seen directly, maintaining a mental mapping between the two worlds. The user needs to translate the information received on the display to the real world. Furthermore, since the device's camera is closer than the user's eye, directly using camera's view without any processing can lead to incorrect depth perception and difficulty to properly estimate distances [5].

What is needed is a transparent display that lets the user see the real world as if they are looking through a window. There is a perfect alignment with no discontinuity and redundancy between the parts of the scene viewed on the display and viewed directly by the user when using such a transparent display. This enables integrating the AR annotations seamlessly into the field of view of the user.

One approach is to develop physically transparent displays. Large OLEDs of 40% transparency have been developed [6], but are far away from full transparency, and further improving on the transparency to let more light pass through is a substantial challenge. Furthermore, to make the system remain in a self-contained and compact form, the technology of making other parts of a smartphone or a tablet, such as the battery, the CPU, and the GPU to be also transparent, or at least minimized. This remains elusive in a foreseeable future.

Another approach, and the approach we take, is to *simulate* the transparency by reprojecting the video feed acquired by the camera to the user's viewpoint, before showing them on the display. With this approach, the world displayed on screen appears aligned with the real-world viewed directly outside the screen's borders, making the display appear virtually transparent.

In this paper, we describe the algorithm of hand-held self-contained simulated transparent display, as well as our prototype implementation. Under the assumption that the geometry is far away, simulating a transparent display requires acquiring the color of the scene, tracking the user's head position, and rendering the color on the display from the user's viewpoint. Color acquisition, 3D rendering, and display are solved problems since modern tablets and smartphones have capable back-facing

cameras, GPUs, and LCDs. User head tracking technology is beginning to appear in hand-held devices recently.

Our prototype is based on a smartphone with built-in head tracking support. It is compact and fully self-contained, that all the acquisition and computation is performed on board, with no need for tethered connection, which is an essential requirement in an austere setting. It adapts to the user's viewpoint, that we validated the transparency error is less than 5% once the scene is beyond 6 m. It does not require geometry acquisition capability, nor passive or active depth acquisition, making it more suitable for outdoor scenes. Our prototype shows its usability for mobile AR applications (Figure 2.1). All the first person view of the transparent display effect shown in this paper were captured by wearing a head mounted camera (i.e. Google Glass). Wearing the camera is not needed during the normal use of our prototype, but is only needed to capture the first-person illustrative footage.

2.2 Prior Work

There exists prior work on simulated transparent display systems, one of the early attempts is ARScope [7]. The user holds an opaque surface as the display like they would hold a magnifying glass. The surface is made transparent to the user by projecting an image on it that contains what the user would see in the absence of the surface, using a head mounted projector. The system works by acquiring the scene with two cameras, one head-mounted, and one attached to the hand-held surface that captures the scene. A homography is computed between acquired images from the two cameras based on correspondences, the homography is then used to warp the scene-looking camera's video feed to the user's viewpoint, and the warped image is projected onto the hand-held surface using the projector. This system demonstrates the simulated transparency, but it suffers from a few important limitations such as the reliance on additional hardware, such as head-mounted cameras and projectors, and the reliance on tethering to a nearby workstation.

Figure 2.1.: Actual first-person photographs of our transparent display prototype, which is compact and it adapts to the user's viewpoint, the transparency effect is accurate for scenes that are far away .

Systems using passive surfaces requiring external projector to make it transparent, and was later preceded by systems using an LCD that can display the image to simulate transparency without the need of the projector. We summary prior simulated transparent display systems based on LCD according to how they track the user's head, to how they acquire the scene geometry, and to whether or not they are tethered or not.

Some systems track the user's head with a head-mounted sensor [8,9]. The encumbrance of a head-mounted device can be avoided by tracking the user with a camera attached to the display. Some prior works assume that the scene is planar, so scene geometry acquisition is simplified to the problem of registration of display pose with respect to the scene proxy plane. Registration is done using either manually [10], based on markers placed in the scene [9, 11, 12], or using features detected in the

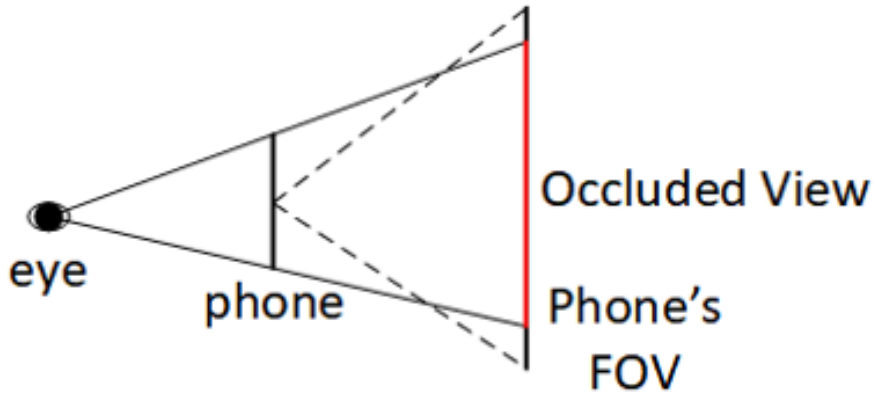
scene image [13]. Other systems reconstruct the scene geometry either actively using on-board depth cameras [14], or passively from the scene video frames [15].

Take the advantage of the general-purpose and graphics computing capabilities of modern mobile devices, recent systems attempt to untether the smartphones and tablets from a nearby workspace and to perform all computation in situ [9,10], making the systems to be more self-contained and compact.

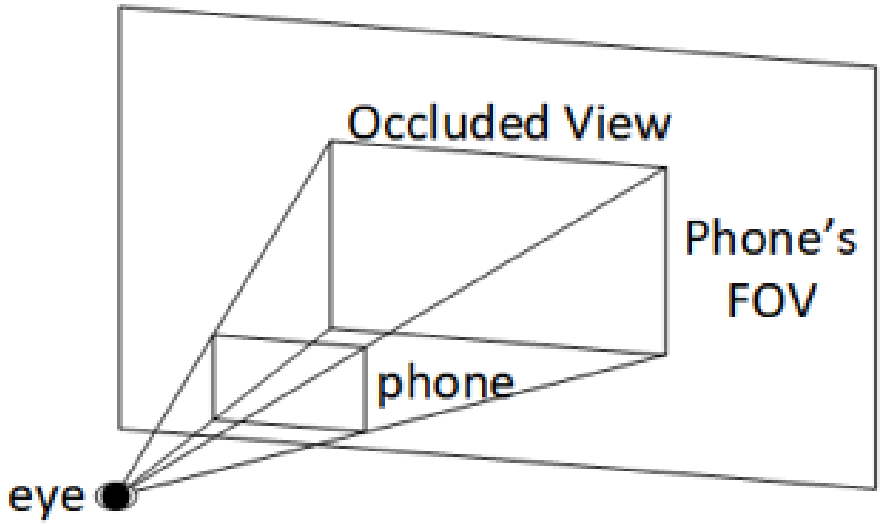
Our prototype advances the state of the art in simulated transparent displays as follows. It is the first untethered transparent display system that uses integrated multi-camera head position tracking; the user's head position is triangulated using multiple front-facing cameras which improves z-tracking accuracy compared to prior systems that use a single camera [16].

2.3 Simulated Transparent Display

To simulate a transparent display using a conventional LCD, one has to display the image that the user would see in the absence of the display. The part of the scene obstructed by the LCD has to be captured with a camera. Placing the camera at the user's viewpoint is not beneficial because the camera's view would also be obstructed by the LCD, in addition to the disadvantage of the user having to wear the camera. Consequently, the camera has to be placed at a different viewpoint, beyond the LCD, such that the scene is captured without occlusions. The frame captured by the camera has to be reprojected to the user's viewpoint, with the assumption of geometry is far away. This is essentially a texture resampling, which is fast on GPUs. In Figure 2.2, the parts of the scene in the display occlusion shadow are acquired with a color camera. The user's viewpoint is acquired with a tracker that triangulates the position of the user's head. The acquired color data is rendered from the user's viewpoint to simulate transparency.



(a)



(b)

Figure 2.2.: Overview of our simulated transparent display.

2.4 Implementation and Results

We pursue the implementation of the simulated transparent display pipeline in a form that is as compact as possible, without wires, and without the need for an auxiliary workstation. All tablet and smartphone platforms now have high resolution video cameras and display. We have implemented the prototype which takes advantage of a smartphone with integrated head tracking capability.

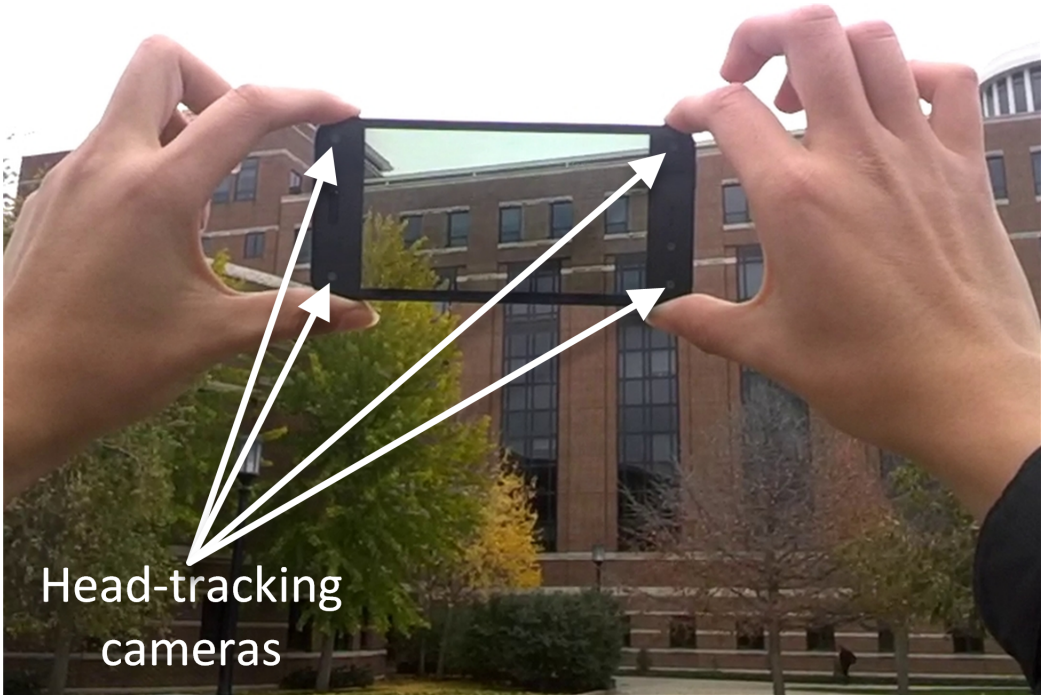


Figure 2.3.: Our prototype.

2.4.1 Prototype Implementation

The prototype leverages Amazon's Fire Phone [17], a 4.7-inch smartphone with four front-facing cameras dedicated to tracking the user's head (Figure 2.3). The device has four cameras to increase the chance that at least two of them have a good view of the user's head, free of finger occlusion. The head position is triangulated from the frames of the two cameras that provide the best view of the user's head. Compared to tracking the user's head with a single camera, triangulation has the advantage of better z tracking accuracy. The Fire Phone API provides a tracking frame rate of up to 100Hz. The Fire Phone does not have depth acquisition capability. We compute the transparency effect under the assumption that the scene is infinitely far away, an assumption that is reasonable for outdoor scenes. As discussed in Section 2.4.2, the transparency effect error is below 5% when the scene is farther than 6 m. As can be seen in Figures 2.1 and 2.3, our prototype is well suited for outdoor

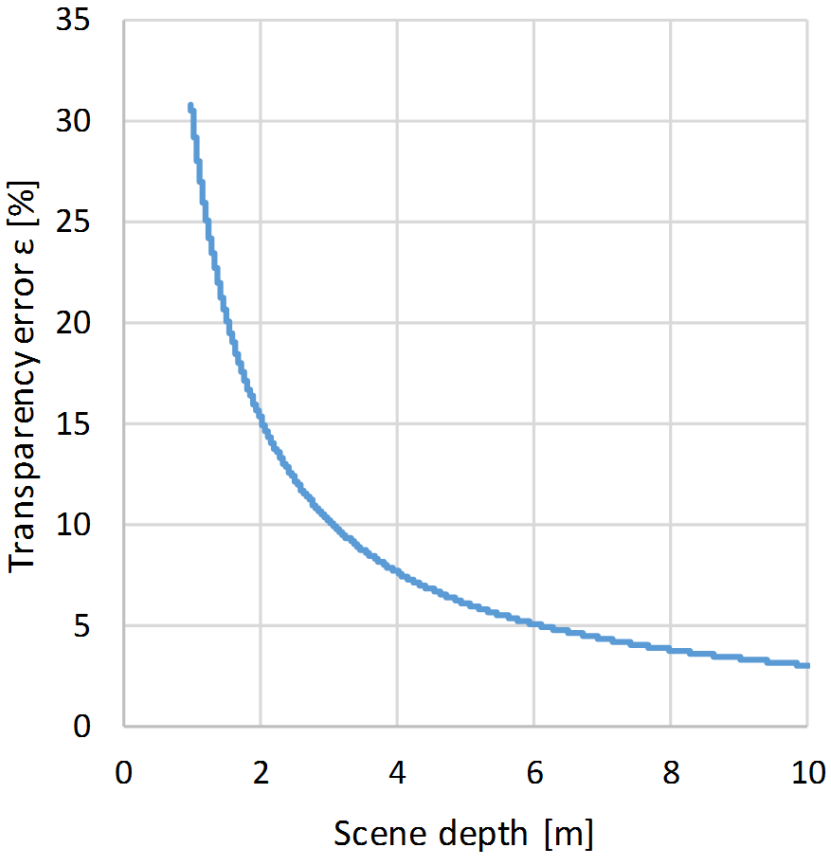


Figure 2.4.: Maximum transparency error of our prototype due to the infinite scene depth assumption, as a function of actual scene depth.

scenes. It is very compact and portable, which readily supports driving and walking navigation assistance applications. Although active depth acquisition outdoors is not yet practical for smartphone-like devices, the benefit from depth acquisition would be small since the scene is typically away from the user.

2.4.2 Quality of transparent display effect

Perfect transparency requires displaying exactly what the user would see if the display were not there. We analyze the quality of the transparent display effect achieved

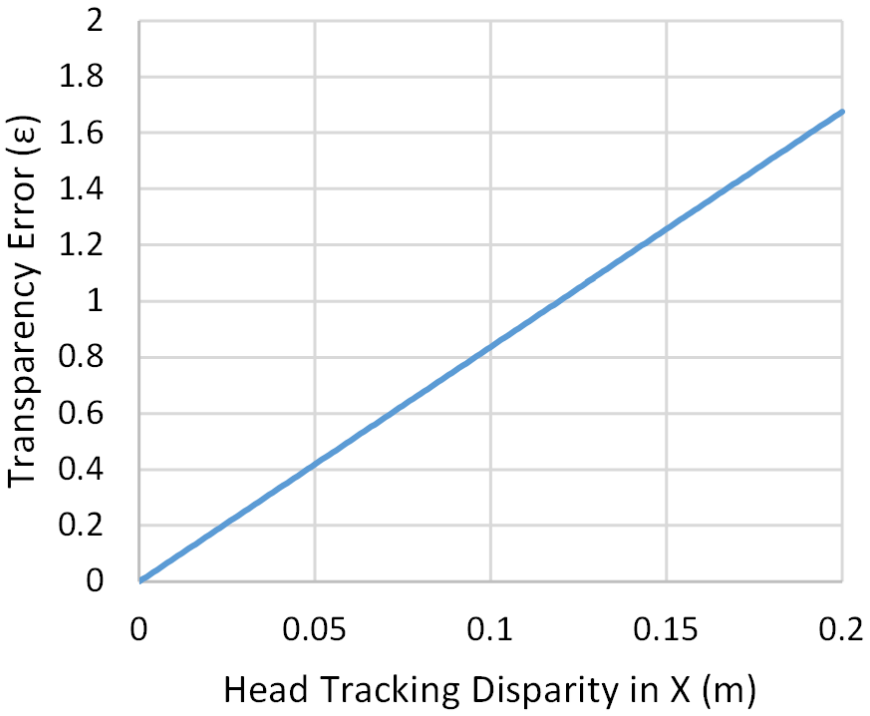


Figure 2.5.: Maximum transparency error of our prototype due to head tracking error in x- and z-direction, respectively.

by our prototypes both theoretically and empirically. We define the transparency error ϵ at a point p on the simulated transparent display as

$$\epsilon = \left\| p - p^0 \right\| / d \tag{2.1}$$

The numerator is the distance in pixels between the actual position p and the correct position p^0 of the scene 3D point imaged at p, and d is the length of the diagonal of the display in pixels.

Figure 2.4 shows the maximum transparency error for our prototype as a function of the depth of the scene. The maximum error is defined as the largest error over the four corners of the display. Because we assume that scene geometry is infinitely far away from the display, the transparency error is only low when the scene geometry is far from the display (e.g. $\epsilon < 5\%$ beyond 6 m).

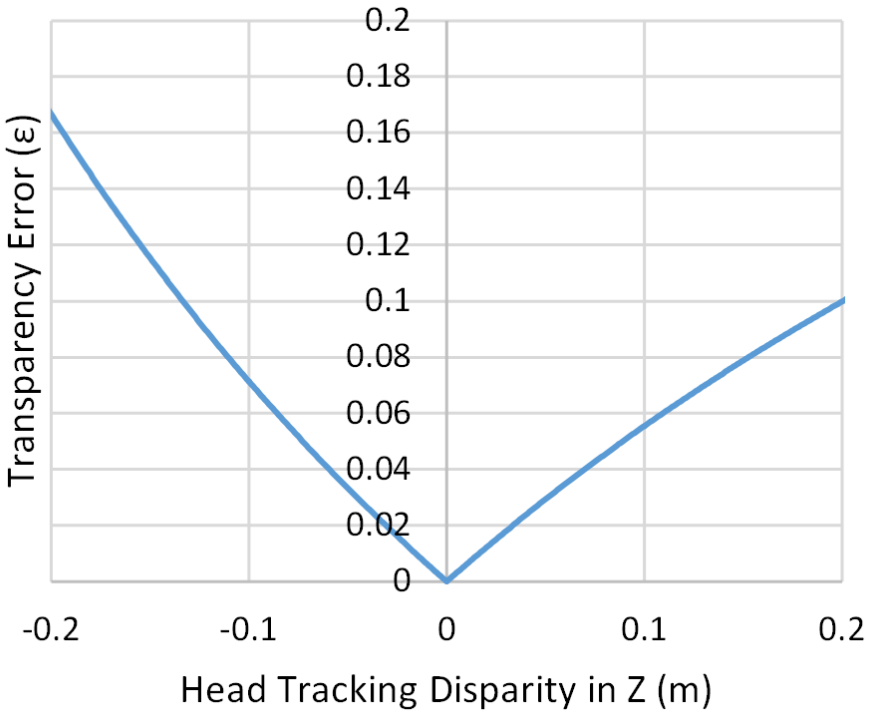
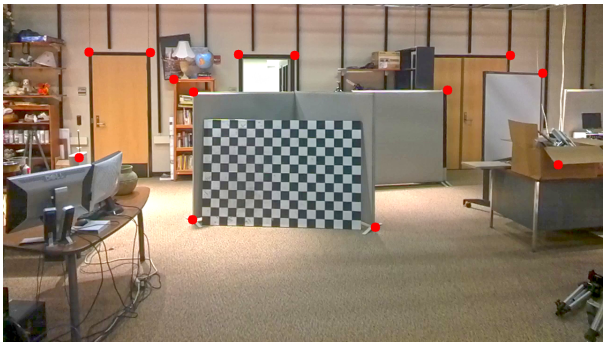


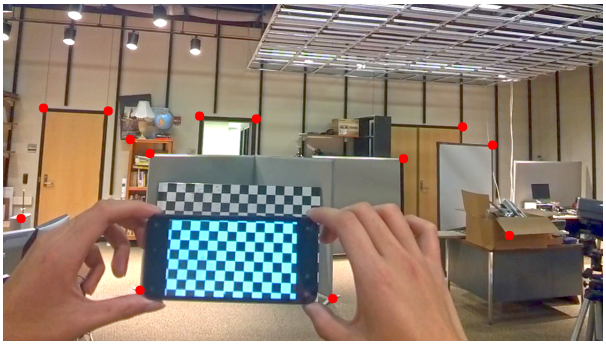
Figure 2.6.: Maximum transparency error of our prototype due to head tracking error in x- and z-direction, respectively.

Figure 2.5 shows the maximum transparency error for our prototype as a function of user head position tracking error in x (similar for y). Figure 2.6 shows the maximum transparency error as a function of the head tracking error in z. Negative head tracking errors in z indicate that the true head position is farther from the display than tracked, while positive errors indicate that the true head position is closer to the display than tracked. The user's head is assumed to be 0.5 m away from the displays; the scene is assumed to be 10 m away, which are typical use cases. The transparency error depends more on the x than the z head tracking error. For our prototype, head tracking is typically accurate to less than 10 mm in x and 30 mm in z, which translates to maximum transparency errors of 8.4%.

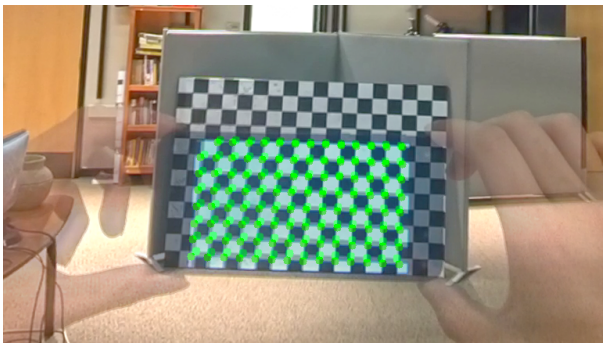
All first person images were taken by having the user wear the Google Glass head mounted camera [18]. In addition to their illustrative purpose, we also use



(a) Reference image of the scene taken by Google Glass.



(b) Image took by Google Glass while using the transparent display.



(c) Overlay image.

Figure 2.7.: Empirical transparency error measurement. The red dots illustrate manually selected salient features in the region outside of the transparent display, which are used to align the two images. Overlay image is where the actual transparency error is measured, using manually selected correspondences (green dots) in the region covered by the transparent display.

Table 2.1.: Empirical transparency errors for our simulated transparent display prototype.

Scene	Figure 2.3	Figure 2.1
Transparency error ϵ [%]	1.2	1.7

these first person images to estimate the transparency error empirically, as shown in Figure 2.7. First, the user acquires an image I_1 of the scene using the Google Glass camera (Figure 2.7, left). Next, the user acquires a second image I_2 of the scene while holding up the simulated transparent display, which has been calibrated to generate a transparent effect for the viewpoint of the Google Class camera (Figure 2.7, middle). Since the user is likely to tilt their head slightly as they acquire the two images, I_1 and I_2 have to be first aligned using the region outside the transparent display. We align the two images by computing a homography between I_1 and I_2 using manually selected corresponding salient features in the region outside the display. The homography is used to compute an overlaid image I_3 (Figure 2.7, right). The transparency error is then computed by measuring the distance between manually selected corresponding features in I_3 that are within the transparent display region. Table 2.1 gives actual transparency error values for our prototypes. These empirical results show that our prototypes achieve a good transparency effect. The small error values indicate that the actual head tracking errors are smaller than the upper bounds used in the theoretical analysis above.

2.4.3 Frame rate and latency

As objects in the scene move and as the user's head moves with respect to the display, the transparent effect has to be recomputed to match the current configuration. There is a delay, or latency, between when the change in the scene or in the head position occurs and when the transparent effect is reestablished. The latency

is due to delays accumulated in the color acquisition, depth acquisition, depth hole filling, triangulation, head tracking, head tracking communication, and rendering. Color is acquired by the on-board color camera at 30 Hz our prototype. The user head tracking on Fire Phone operates at 100 Hz. Rendering only takes 3 ms.

The average latency for our simulated transparent display prototypes is 120 ms. The latency was measured using the Google Glass first person video feed by counting the number of frames it takes to the transparency effect to converge after a change in the scene or in the user's head position occurs. We have also measured the latency of displaying a video frame as it is acquired, without any processing. For the Fire Phone, the device that underlies our prototype, this acquire-and-display latency is 114 ms. Consequently, most of the latency of our prototypes comes from the latency with which the devices display the images they acquire. The additional latency due to the processing required to achieve the transparent effect is less than a third of the total latency. For our prototype, this additional latency is negligible (i.e. 6 ms out of the total 120 ms), as the computation is simple. This indicates that, in addition to integrating the user head tracking capability into next generation tablets and smartphones, portable device manufacturers should pursue improving support for AR applications by also reducing the acquire-and-display latency of their devices.

2.4.4 Limitations

As discussed, our prototype does not acquire scene geometry, so an accurate transparent effect requires the scene to be far away. Also, displays exhibit latency. An important limitation arises from the fact that the transparent display caters only to a single viewpoint. Our transparent displays can cater to one of the eyes, or to the midpoint of the interpupillary segment. The lack of disparity between the images shown to the user's eyes hinders depth perception. Furthermore, when the scene is far away from the display and the display is close to the user, the user cannot focus both on the scene and on the display.

2.5 Conclusions and Future Work

We have demonstrated the feasibility of a simulated transparent display that is completely self-contained, i.e. untethered. The user does not have to wear any sensors and the scene does not have to be enhanced with markers. We have developed a prototype that takes advantage of user tracking capability on the emerging mobile platforms. We believe that this function will be commonplace for the next smartphone and tablet generations, in support of powerful AR applications.

In addition to improving our simulated transparent displays to alleviate the limitations discussed above, future work also includes using the transparent displays in actual AR applications such as car and pedestrian navigation assistance and surgical telementoring.

3 EFFECTIVE WORKSPACE VISUALIZATION IN AR TELEMENTORING

In this chapter, we describe our research towards achieving an effective visualization of the workspace for the mentor in AR-based surgical telementoring. The fundamental challenge is to capture the workspace from an inherently different viewpoint than that of the beneficiary of the visualization, i.e. than that of the mentor, and then making this visualization as useful as effective as possible by connecting the acquisition viewpoint to the mentor's viewpoint. We have investigated two approaches. The first approach captures the workspace, i.e. the operating field, using an overhead camera (Section 3.1). The second approach investigates capturing the workspace using a video camera built into the mentee's AR headset, which poses the main challenge of stabilizing the visualization by attenuating the abrupt changes caused by the mentee's head motions (Section 3.2).

3.1 A First-Person Mentee Second-Person Mentor AR Interface for Surgical Telementoring

3.1.1 Introduction

As surgery continues to specialize more narrowly and deeply, it becomes more and more challenging to provide all needed surgical expertise at all points of care. Surgical telementoring is a promising approach for transmitting surgical expertise over large distances promptly and efficiently. Consider a rural surgery center staffed with only a general surgeon. An expert surgeon from a major urban hospital could "virtually scrub in" to assist with a procedure that the general surgeon is not entirely comfortable performing alone. Consider the scenario of a critical patient who cannot be urgently transported to a facility where the required surgical expertise is

available. This could be the case, for example, in a combat zone where a compartment syndrome relieving fasciotomy procedure has to be performed urgently at a forward operating base to save a patient's leg, and evacuating the patient is too slow or too dangerous. An orthopaedic trauma surgeon from a major military hospital could assist from thousands of miles away via telementoring. As a third example, a novel surgical procedure can be rapidly disseminated through surgical telementoring. Finally, telementoring could also benefit surgical training, with a single instructor working in parallel with multiple surgical residents, providing assistance on demand, to the trainees who need it.

The conventional approach for surgical telementoring is based on a telestrator that allows a remote mentor to annotate graphically a video feed of the surgery, which is then shown to the mentee on a nearby display. This requires the mentee to shift focus away from the surgery, and to map mentally the instructions from the nearby display to the surgical field, which can lead to surgery delays and even errors. Augmented Reality (AR) is a promising alternative for surgical telementoring because it allows to integrate the mentor-authored annotations directly into the field of view of the mentee. The mentee sees the annotations as if the mentor actually drew them onto the surgical field, which avoids focus shifts and the high cognitive load of having to map annotations to the surgical field.

One possible AR interface for surgical telementoring is a transparent display that is placed between the mentee and the patient and that shows the mentor annotations overlaid onto the surgical field. However, truly transparent displays are not yet available. Video see-through transparent displays simulate transparency by showing the real world scene with the help of a video camera. Such a display supports only monoscopic viewing of the surgical field, which reduces depth perception and can decrease surgical performance. Furthermore, the transparent display approach poses the challenge of work-space encumbrance, as the surgeon has to reach around the display. An alternative interface is an optical see-through AR head-mounted display

(HMD). The AR HMD avoids workspace encumbrance and it allows the mentee to see the surgical field directly, with natural depth perception.

We are a group of computer science and industrial engineering researchers, trauma and orthopaedic trauma surgeons, and surgery educators. In this application paper we describe a novel system for surgical telementoring based on an AR HMD, as well as an initial evaluation in a study where surgery residents performed lower-leg fasciotomies on cadaver patient models.

Figures 3.1 and 3.2 gives an overview of our system. The surgical field is acquired with an overhead camera whose feed is sent to the remote mentor site where it is displayed on a custom full-size interaction table. The mentor annotates the surgical field using touch-based gestures. The annotations are sent to the mentee site where they are integrated into the mentee's view of the surgical field using an AR HMD. The annotations are converted from 2D to 3D by projection from the overhead camera view to the 3D geometry of the surgical field acquired by the AR HMD. In this way, the remote mentor can annotate the surgical field in real time, and the annotations are shown to the mentee anchored to the surgical field, with correct depth perception. Our AR interface provides a first-person view to the mentee, who sees the annotations from their own viewpoint, and a second-person view for the mentor, who sees the surgical field and authors annotations from the overhead camera viewpoint.

We have conducted a user study to test our system with fourteen surgery residents and six medical students, who were asked to perform a lower-leg fasciotomy on a cadaver patient model. The participants were assigned to two groups: a control group (CG), which performed the fasciotomy after studying the procedure from printed surgery course materials, and an experiment group (EG), which performed the fasciotomy under telementoring guidance using our system. Participant performance was rated by an expert surgeon who witnessed the procedure and quantified performance using an Individual Procedure Score (IPS) metric. The EG participants received an IPS score 16% higher than the CG participants. The two groups were also evaluated using a system usability questionnaire. The answers to all eight questions

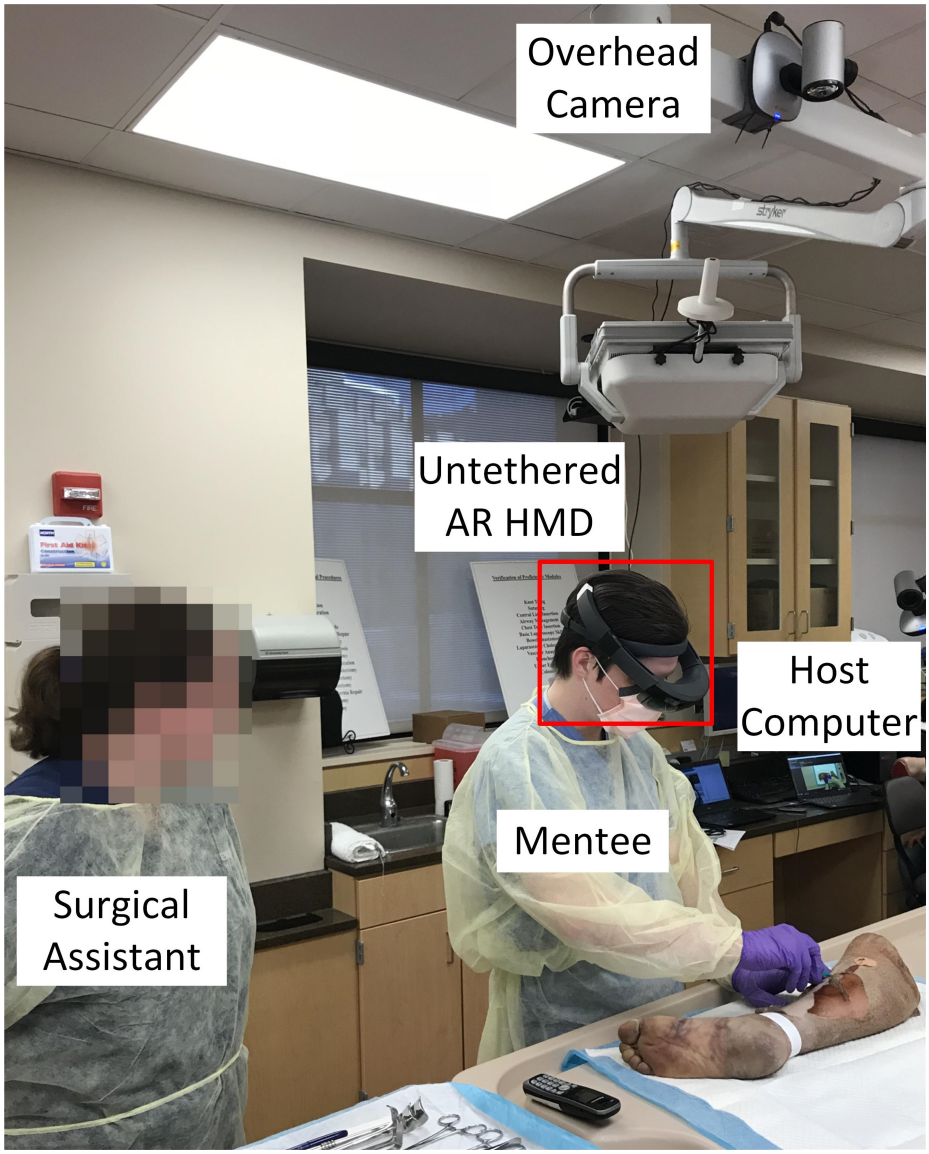


Figure 3.1.: Mentee subsystem of our telementoring system, based on an AR HMD.

indicate a usability advantage for our system, and for four of the questions the advantage was statistically significant. Finally, the two groups were also evaluated based on self-reported confidence in the knowledge of the fasciotomy procedure, before and after the study. The *EG* group showed statistically significant growth for all four confidence metric questions, and they ended up with a higher confidence level than the *CG* group.

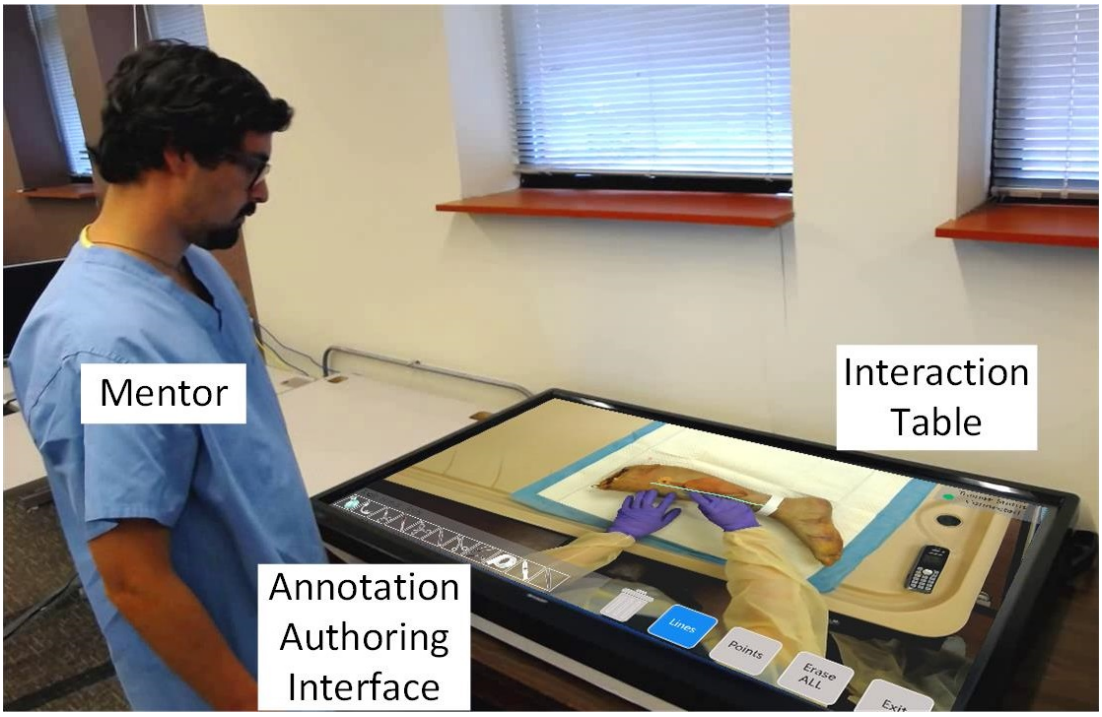


Figure 3.2.: Mentor subsystem of our telementoring system, based on a full-size touch-based interaction table.

3.1.2 Prior Work

The conventional approach for surgical telementoring is based on a telestrator. The live video feed of the surgical field is transmitted to the remote mentor, who annotates it, the annotations are sent back to the mentee, and the annotated video is shown to the mentee on a nearby display [19]. Such annotations are not naturally seen by the mentee due to the lack of depth perception, due to the lack of parallax, and due to occasional occlusions. Another shortcoming is the need for the trainee to shift focus repeatedly from the surgical field to the nearby display. Each time, the mentee has to remember the position and type of individual annotations, and then to map them from memory onto the actual surgical field. These focus shifts increase the cognitive load of the mentee, which can translate to surgery delays or even surgical errors [19].

AR interfaces can provide a natural approach for overlaying annotations into mentee's field of view, as if the mentor actually drew them there, thus eliminating focus shifts. This potential of AR in surgery has been noted for a long time [20]. The recent leap forward of AR technology has intensified anew research efforts aimed at bringing AR into the operating room.

There are two major options for designing the AR interface: based on a transparent display interposed between the mentee and the patient, and based on an AR HMD [21]. In previous work we have explored the transparent display option [3]. A video-see through display, implemented by a computer tablet, was suspended above the surgical field. The camera built into the tablet acquires the surgical field, the video feed is sent to the mentor, and the mentor uses a touch-based interface to annotate the surgical field. The annotations are sent back to the trainee site, shown on the tablet, and superimposed onto the live view of the surgical field. The trainee can then follow the instructions from the mentor to complete the surgery, without having to switch focus away from the surgical field. Compared to a conventional telestrator system, a user study revealed that our system led to 57% smaller surgical port and instrument placement errors, and to 65% fewer focus shifts. One of the shortcomings of such a tablet-based AR interface is the lack of depth perception that ensues from the monoscopic visualization of the surgical field. A second important shortcoming is the workspace encumbrance brought by the tablet, which can require the mentee to deviate from their preferred arm and hand poses and motions during surgery.

In this paper we investigate the use of an optical see-through AR HMD interface, which has the potential to address these shortcomings. The mentee sees the surgical field directly, with natural depth perception. The annotations are drawn in 3D, with correct parallax between the left and right eyes, so the annotations are seen with depth perception as well. Furthermore, the HMD does not interfere with the mentee's arm motions. Prior work investigation of the use of AR HMD interfaces in the operating room have found benefits in the context of overlaying a static image or model onto

the patient [22, 23], and of overlaying a visualization of patient specific data acquired with an imaging system [24].

3.1.3 Surgical Telementoring through Head-Mounted Display Augmented Reality

The goal of surgical telementoring is to allow the mentee to see the mentor-authored annotations naturally, as if the mentor actually drew them on the patient. We have developed a system that allows the mentor to see and annotate the surgical field, and that integrates the annotations into the mentee's field of view of the surgical field. We first discuss the design of the AR interface at the mentor and mentee that enables telementoring, and then we give an overview of the calibration and operation of our system that implements the AR interface.

AR Interface Design

We developed the AR interface of our surgical telementoring system based on the following considerations. First, we wanted the mentee to see the annotations directly overlaid onto the surgical field. This was satisfied by using an AR interface. The second consideration was to provide the mentee with depth perception for the surgical field and the annotations. This was satisfied by resorting to an optical see-through AR HMD, through which the surgical field can be seen directly, and which visualizes the annotations stereoscopically. The third consideration was to avoid encumbering the mentee workspace, which reinforced our choice for an HMD AR interface, as opposed to interposing a display in between the mentee and the patient.

The fourth consideration was to provide the mentor with an appropriate visualization of the surgical field. Our first attempt was to use the on-board camera already built into the AR HMD. However, in our preliminary tests, such a visualization proved to be ineffective, as it changes frequently, abruptly, and substantially as the mentee moves their head. This unstable visualization of the surgical field is particularly disconcerting to the mentor when trying to draw an annotation. Furthermore, directly

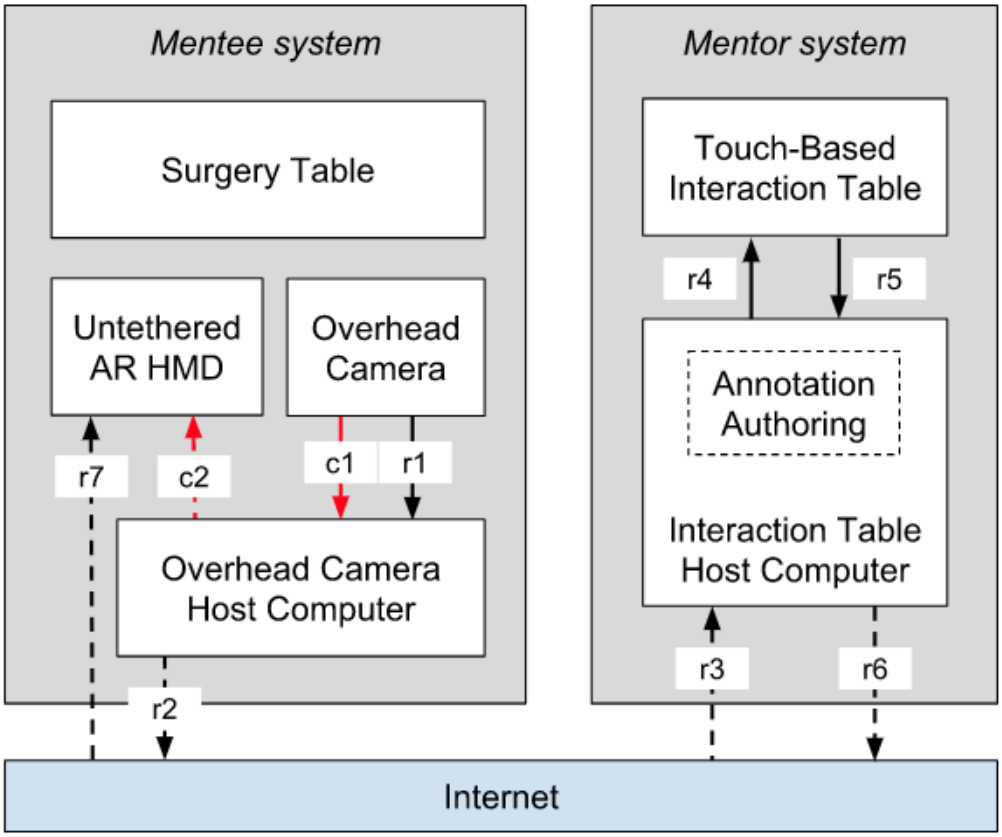


Figure 3.3.: System diagram. Solid and dotted arrows correspond to wired and wireless communication, respectively. Red illustrates system calibration, and black illustrates system operation.

inheriting another user's first person view can be disorienting and it can even induce nausea [25]. To avoid these problems, we decided to deploy an external overhead camera that captures the surgical field from a stationary position above the surgical field. In conclusion, our interface uses a first person view for the mentee and a second person view for the mentor.

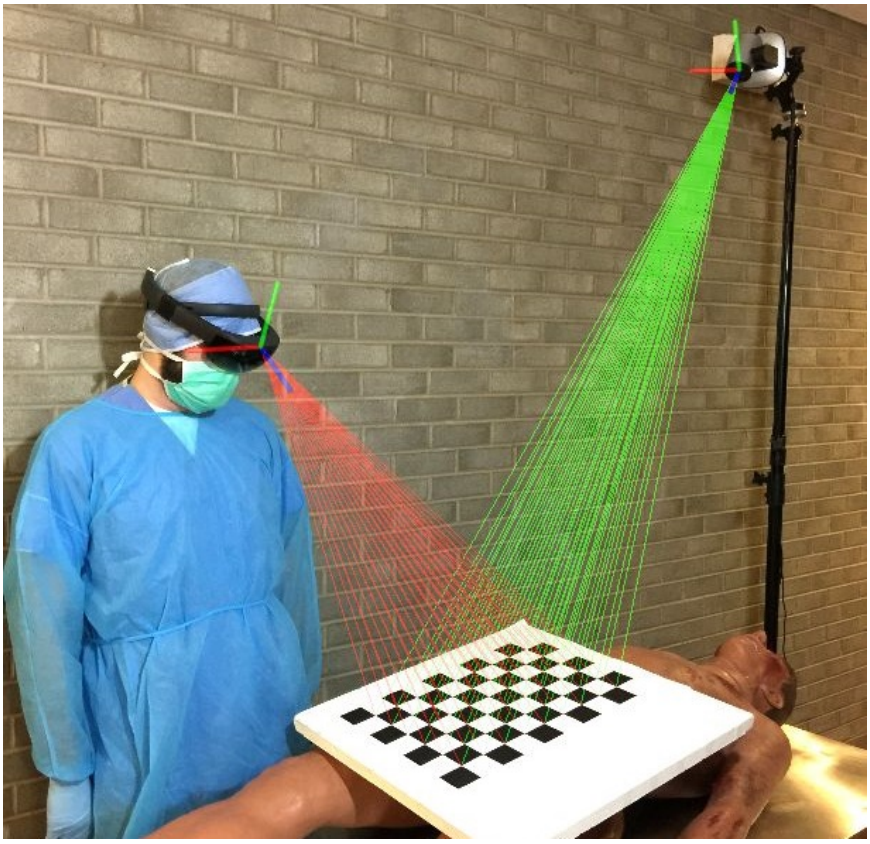


Figure 3.4.: Calibration process. The overhead camera (green ray visualization) is registered with respect to the camera built into the AR HMD (red rays) using a calibration checkerboard.

System Calibration

Figure 3.3 gives an overview of our surgical telementoring system (Figures 3.1 and 3.2). We describe our system using the $\xi_{A,B}$ notation for the $SE(3)$ transformation between coordinate systems A and B.

There is a one-time calibration process after which the system becomes operational. We use an untethered, self-tracking AR HMD, which, for every frame, provides the position and orientation of the HMD with respect to the world. The goal of the calibration stage is to determine the pose $\xi_{oc,w}$ of the overhead camera (OC) in the world coordinate system (W) of the AR HMD. Our AR HMD has a built-in video

camera which we leverage for this calibration process. We use a standard calibration procedure [26] that first calibrates the intrinsics of the overhead and built-in cameras. Then the overhead and built-in camera extrinsics are found by showing a calibration checkerboard to both cameras simultaneously (Figure 3.4). The overhead camera sends its image to the host computer (c1 in Figure 3.3), where the checker corners are detected and the pose $\xi_{oc,cp}$ relative to the checkerboard pattern (CP) is computed by solving a perspective-n-point problem [27]. The pose of the AR HMD relative to the checkerboard pattern $\xi_{hmd,cp}$ is computed similarly. $\xi_{oc,cp}$ is sent to the AR HMD (c2), where the pose of the overhead camera $\xi_{oc,w}$ is finally computed with the following concatenation of transformations (Equation (3.1)), where $\xi_{hmd,w}$ is the HMD pose tracked for the frame that captures the checkerboard pattern. $\xi_{oc,w}$ is stored on the AR HMD and used during operation to visualize the mentor annotations.

$$\xi_{oc,w} = \xi_{oc,cp} \cdot \xi_{hmd,cp}^{-1} \cdot \xi_{hmd,w} \tag{3.1}$$

System Operation

The overhead camera captures a live video feed of the surgical field (r1 in Figure 3.3), which is sent to the remote mentor via the Internet (r2). The feed is received at the mentor subsystem (r3), where it is displayed on the touch-based interaction table (r4). The mentor examines the surgical field, zooms in and pans the view digitally, and authors annotations as needed using touch-based gestures. The annotation authoring commands are collected (r5) and sent to the mentee subsystem via the Internet (r6). The AR HMD is connected to the Internet and directly receives the annotation commands (r7), which it uses to draw the annotations for the mentee as follows.

Given a 2D annotation point p in the overhead camera image plane, its 3D position P is computed by unprojection to the overhead camera ray r_{oc}, by transforming the ray to world coordinates $r_W = \xi_{oc,w} r_{oc}$, and by intersecting the ray with the surgical

Figure 3.5.: Annotation projection. The incision line, the scalpel tip, and the textual label stem tip are projected from the overhead camera perspective onto the geometry of the surgical field. The incision line lies on the patient, whereas the scalpel and the label annotations float above the patient.

field geometry G, i.e. $P = r_w \cap G$. We approximate G with the coarse geometric model of the scene acquired by our AR HMD. Figure 3.5 illustrates the process of mapping 2D authored annotations to 3D by projection onto surgical field geometry along overhead camera rays.

3.1.4 Results and Discussion

We implemented our system using a Microsoft HoloLens AR HMD which has the advantages of being untethered, allowing the mentee to move freely, of having a built-in video camera, allowing for overhead camera calibration, of self-tracking, allowing annotation anchoring as the mentee moves, and of acquiring a geometric proxy of the scene, allowing for annotation projection. The AR HoloLens display has a $1,280 \times 720$ resolution and a refresh rate of 60Hz. An important shortcoming of the HoloLens is the small field of view of the AR display (i.e. about 30 by 17.5 degrees), which restricts annotation display to the center of the field of view of the mentee.

The overhead camera is a Logitech PTZ Pro 2, acquiring 1920×1080 pixel frames at 30fps. Audio communication between the mentor and the mentee was provided with a conventional phone in speaker mode. The interaction table at the mentor was built from a multi-touch interaction Sharp LCD (1920×1080 resolution, 60 fps, physical size of 52.3×29.4 inches), connected to a PC.

We first discuss system performance based on technical metrics, then we describe a user study where we tested our system in the context of fasciotomy telementoring, and we end the section with a discussion of the limitations of our system.

System Performance

One important aspect of our real-time visual communication system is latency. One latency is the delay with which the overhead camera video feed is transmitted from the mentee site to the mentor site. We have measured ping times from 50ms within our Purdue servers, to over a second from Purdue to universities in South-East Asia and Australia. The encoding and decoding of the video stream are done with negligible delay. In our experiments network bandwidth was not a concern as it was sufficient to transmit the overhead camera feed at full resolution with levels of compression that did not affect video quality. Another latency is the delay between the mentee head movement and the required repositioning of annotations, which for our AR HMD is an almost unnoticeable 16ms. In other words, when the mentee moves their head, the annotations appear stationary in the 3D world, and do not "follow" the mentee's view direction.

The annotation display error is the cumulative effect of camera calibration, mentee head tracking, surgical field geometry, and HMD fitting errors. We have measured the annotation display error empirically, by placing a physical marker A in the surgical field, asking the mentor to annotate the position of the marker in the overhead camera feed, and then by asking the mentee to place a second physical marker B at the location where they see the annotation drawn. The annotation display error is the

distance between markers A to B. By marking the entire surgical field, we measured a maximum and average annotation display error of 1.60cm and 1.22cm, respectively.

As the direction and length of the AB segment is consistent over the surgical field, we have devised an optional additional calibration procedure that improves annotation display accuracy under the assumption that most of the systematic error is due to an consistent overestimation of scene geometry by the HoloLens. Indeed, using the built-in Kinect-like depth camera, the HoloLens builds an approximate geometric model of the scene that consistently overestimates scene geometry, by wrapping a coarse geometric mesh over the actual detailed geometry. The additional calibration procedure is based on interaction between mentor and mentee. The mentor places an annotation and then asks the mentee to place and hold their index where they see the virtual annotation. The annotation display error is apparent to the mentor in their overhead camera view as a distance between the mentee's finger tip and where the mentor drew the annotation. Using this visualization, the mentor shifts the approximate geometric model of the surgical field to reduce the annotation display error.

User Study

We have conducted a user study at the Indiana University School of Medicine with $n = 20$ participants: 14 surgery residents and 6 medical students. The *task* was a four-compartment release by dissecting lower-leg fascia on cadaver models. Such a fasciotomy intervention is an emergency procedure for treating compartment syndrome, which is a lack of blood circulation to the limb due to excessive swelling as the result of blunt trauma. If left untreated, compartment syndrome leads to the loss of the affected limb. Fasciotomies remain challenging surgical procedures. In a recent systematic review on the surgical management of chronic exertional compartment syndrome, the overall success rate was reported at 66%, the satisfaction rate was 84%, and the rate of return to previous or full activity was 75% [28]. Furthermore, symptom

Figure 3.6.: *EG* participant in the fasciotomy user study. The virtual incision line and instruments are only seen by the participant, and they were added here for illustration purposes.

recurrence was up to 44.7%, reoperation rate up to 19%, and overall complication rate was 13%.

Participants were randomly assigned to one of two groups: a control group (CG), which received instruction on how to perform the fasciotomy from an illustrated brochure, i.e. the Advanced Surgical Skills for Exposure in Trauma [29] course material on fasciotomies, and an experiment group (EG), which received real-time guidance with our telementoring system. The *EG* group did not receive any fasciotomy instruction prior to actually performing the procedure. Figure 3.6 and Figure 3.7 show a participant in the experiment group and control group, respectively. The additional interactive calibration procedure was performed by the mentor with each mentee, as the procedure depends on the actual surgical field geometry, and the cadaver lower leg models had great shape and size variability.

Figure 3.7.: *CG* participant in the fasciotomy user study.

The two groups were compared based (1) on expert rating, (2) on self-reported usability, (3) on self-reported confidence in procedure knowledge, and (4) on procedure completion time. To analyze the data, we first check the data normality assumption using the Shapiro-Wilks test [30] and in our case no data was normal. For the unpaired (between subject) data (1, 2 and 4), we use the Mann-Whitney U test [31] to test for statistical significance. For the paired (i.e. within subject) data (3), statistical significance is tested with the Wilcoxon signed-rank test [32].

(1) An expert surgeon evaluated the performance of each participant during and after the experiment using the Individual Procedure Score metric [33], which we adapted to fasciotomy. IPS is a test that assesses whether a training course is being

Table 3.1.: Self-reported support method usability. P-values with an asterisk (*) represent a statistically significant difference between the two groups. For questions 6 and 8, a lower score is indicates a higher preference.

Question	EG	CG	p-value
[1] Sufficient information provided	5.0 ±1.00	4.0 ± 0.50	0.024*
[2] Instructions easy to follow	5.0 ± 1.00	4.0 ± 1.25	0.018*
[3] Instructions conveyed effectively	4.0 ± 1.25	4.0 ± 1.00	0.415
[4] Cleared procedure doubts	4.0 ± 1.25	3.0 ± 1.50	0.063
[5] Expedited procedure completion	5.0 ± 2.25	3.5 ± 2.25	0.111
[6] Generated frustration	2.0 ± 1.25	3.0 ± 2.00	0.037*
[7] Better than side-by-side mentoring	2.0 ± 2.00	2.0 ± 1.00	0.139
[8] Worse than side-by-side mentoring	2.5 ± 2.25	4.0 ± 2.00	0.028*

effective on improving the overall surgical expertise of a participant. The test includes an objective analysis of the participant's execution of the required procedural steps, as well as a subjective analysis to identify any errors that occur during procedure execution. *EG* participants received a median IPS of 81.15 with an interquartile range of ± 23.25, which was 16% higher than for *CG* participants (69.55 ± 33.40). The interquartile range is defined by the score received by the 25th percentile participant and the 75th percentile participant, and was used here as the data pointed to non-normality. However, the greater *EG* IPS scores were not statistically significant ($p = 0.26$).

(2) The two groups were compared based on self-reported usability through a five-level Likert scale questionnaire (Table 3.1). *EG* participants reported a higher preference for their condition than *CG* participants. For four out of the eight questions, the difference was statistically significant.

Table 3.2.: *EG* participant self-reported confidence scores. All p-values report a significant improvement.

Confidence Assessment Aspect	Self-Reported Confidence Difference	p-value
Identify anatomical landmarks	1.0 ± 1.25	0.014*
Knowledge of procedural steps	1.0 ± 1.00	0.006*
Instrument handling technique	1.0 ± 1.25	0.014*
Perform procedure alone	1.5 ± 1.00	0.006*

(3) The two groups were also compared in terms of self-reported confidence in performing a fasciotomy procedure. Table 3.2 and Table 3.3 report the increase in participant confidence level from before to after the experiment, for *EG* and *CG* participants, respectively. The confidence scores are assigned on a scale from 1 to 5. *EG* participants reported a statistically significant improvement in all four confidence categories, whereas *CG* participants reported statistically significant improvements in only half of the categories. Table 3.4 and Table 3.5 provide the initial and final confidence levels, for the two participant groups. The *CG* participants were more confident than the *EG* participants in their knowledge of the procedure before the task, but *EG* participants were more confident after the task.

(4) *EG* participants completed the procedure marginally faster (i.e. 4% faster, 1,379s median completion time with a ± 380s interquartile range) than *CG* participants (1,444s ± 685s).

This first study indicates that our AR surgical telementoring has the potential to provide surgical expertise remotely in an effective way. Not all advantages detected are statistically significant. One reason is the great variability and low number of participants. Another reason is that the remote mentor was a faculty member overseeing the surgery residency program, who was known to the participants, which added sig-

Table 3.3.: *CG* participant self-reported confidence scores. p-values with an asterisk (*) represent a statistically significant improvement.

Confidence Assessment Aspect	Self-Reported Confidence Difference	p-value
Identify anatomical landmarks	1 ± 1.00	0.022*
Knowledge of procedural steps	1 ± 2.00	0.036*
Instrument handling technique	0 ± 1.00	0.225
Perform procedure alone	1 ± 0.25	0.11

Table 3.4.: Participants' self-reported confidence before the experiment.

Confidence Assessment Aspect	EG	CG
Identify anatomical landmarks	3.00 ± 1.25	3.50 ± 1.00
Knowledge of procedural steps	3.00 ± 0.50	2.50 ± 2.00
Instrument handling technique	3.00 ± 2.00	4.00 ± 1.50
Perform procedure alone	2.00 ± 1.25	3.00 ± 1.25

Table 3.5.: Participants' self-reported confidence after the experiment.

Confidence Assessment Aspect	EG	CG
Identify anatomical landmarks	4.00 ± 1.25	4.00 ± 1.00
Knowledge of procedural steps	4.00 ± 0.00	3.50 ± 1.25
Instrument handling technique	4.00 ± 2.00	4.00 ± 2.00
Perform procedure alone	3.50 ± 1.00	3.50 ± 1.50

nificant performance pressure on *EG* participants, whereas *CG* participants worked without the pressure of being evaluated by one of their professors. Furthermore, the

telementoring sessions turned into practical lessons of surgery, which included revisiting of fundamental concepts in anatomy and in surgical procedures. This was of course not the case for *CG* participants. Not counting the tangential teaching mixed in with fasciotomy telementoring is difficult to do objectively, but it is likely to reduce the overall procedure completion times considerably for *EG* participants.

Limitations

Both the mentee and the mentor complained occasionally that the annotation showing the incision line would obstruct the view of the actual incision, as the incision progressed as it was executed. A possible solution for this problem that we will explore in a future study is to ask the mentee to transfer the annotation on the actual skin of the patient with a surgical marker before actually performing the incision.

Another limitation of our system is that the AR HMD is not very bright, and annotations appear faint when the background is brightly lit, as it is the case of surgical fields illuminated by surgical lights. A video see-through AR HMD is able to have opaque annotation pixels that completely erase the real world pixels, but an optical see-through AR HMDs can only draw semi-transparent annotations on top of the user's view of the real world.

Our system inherits additional limitations of the AR HMD, such as a small field of view of the active part of the display, which confines annotation display to the center of the mentee's field of view. Another limitation is the poor ergonomics of operating with a heavy and sometimes poorly fitting contraption attached to one's head. Several participants reported back and neck strain, especially the ones with little surgical experience who would tilt their head forward, moving the weight of their head and of the display away from their body.

3.1.5 Conclusions and Future Work

In this application paper we have presented the design and implementation of a surgical telementoring AR interface, and we have validated our system in a user study where participants performed a cadaver-leg fasciotomy under telementoring. Our system promises surgical telementoring benefits, although not all benefits measured were statistically significant in this initial study.

Another direction of future work is to improve the mentor's sense of presence in the operating room. One option is to directly use the video feed acquired by the AR HMD from the mentee's viewpoint. As discussed in Section 3.1.3, the challenge is to stabilize this first-person view. This not only simplifies the system, but also potentially increases the accuracy of the annotations, by authoring annotations in a view similar to the one from where they will be seen. Another option is to not only provide a video feed of the surgery, but actually an RGBZ stream of frames with per pixel depth, which allows the mentor to choose his viewpoint interactively, to draw annotations more accurately in 3D (e.g. a non-planar incision curve), and even to visualize the surgical field immersively, e.g. with a Virtual Reality headset.

Telementoring could also benefit from extending the types of annotations supported with the ability to send a visual depiction of the mentor's hands, as surgical instruction includes mid-air gestures that sketch, for example, the use of an instrument. We foresee that the quickest path to achieving this is to capture the mentor hands with a video stream, to segment them, and to display them at the mentee.

Our current surgical telementoring system relies on a high-quality network, which is not always available in the case of austere environments. For this, the system should be enhanced with AI mentoring capabilities that can provide basic assistance to the mentee when the network connection is failing, or is not available at all. One of the major challenges is to recognize automatically the current state of the surgery, a difficult case for computer vision algorithms as surfaces are fragmented, with view-dependent reflective properties, with complex occlusions, and deforming rapidly.

Beyond system refinements, additional user studies are needed to specialize the interface and to optimize the surgical telementoring benefits of our system in the context of many other types of surgical procedures.

3.1.6 Acknowledgments

We thank our Augmented Reality Tea group for insightful comments and suggestions. This work was supported by the Office of the Assistant Secretary of Defense for Health Affairs under Award No. W81XWH-14-1-0042, and by the NSF under Grant DGE-1333468. The views expressed in article, reflect the results of research conducted by the author(s) and do not necessarily reflect the official policy or position of the funders, including but not limited to the Department of the Navy, Department of Defense, or the United States Government.

3.2 Robust High-Level Video Stabilization for Effective AR Telementoring

3.2.1 Introduction

As science and technology specialize ever more deeply, it is more and more challenging to gather in one place the many experts needed to perform a complex task. Telecollaboration can transmit expertise over large geographic distances promptly and effectively [34].

A special case of telecollaboration is telementoring, where a mentee performs a task under the guidance of a remote mentor. One approach is to rely only on an audio channel for the communication between mentor and mentee. Telestrators add a visual channel—the mentor annotates a video feed of the workspace, which is then shown to the mentee on a nearby display [35]. The challenge is that the mentee has to switch focus repeatedly away from the workspace, and to remap the instructions from the nearby display to the actual workspace, which can lead to a high cognitive load on the mentee, and ultimately to task completion delays and even errors [19].

Augmented Reality (AR) technology can solve this problem by directly integrating the annotations into the mentee's field of view. The mentee sees the annotations as if the mentor actually drew them on the 3D geometry of the workspace, eliminating focus shifts [36].

A problem less studied but nonetheless of great significance is conveying the workspace to the remote mentor effectively [21, 37]. One approach is to acquire the workspace with an auxiliary video camera, and to send its video feed to the mentor [38]. The approach requires additional hardware, and the auxiliary camera captures the workspace from a different view than that of the mentee. Effective telementoring requires the mentor to see what the mentee sees for the instructions to be as relevant and easy to understand as possible [39]. For example, the mentor might annotate a part of the workspace that is not visible to the mentee due to occlusions, or, conversely, the mentor might not see the part the mentee is working on.

With the advancement of AR, self-contained optical see-through head mounted displays (HMDs) are now available. Such HMDs typically incorporate a camera, which can capture the workspace from a viewpoint close to the mentee's viewpoint. However, simply providing the mentee first-person video to the mentor is insufficient for effective telementoring [40]. As the mentee changes head position and view direction, the mentor's visualization of the workspace changes frequently and substantially, which adversely affects the mentor's understanding of the scene. This in turn degrades the quality of the guidance provided by the mentor, and ultimately the mentee's performance. For example, when the mentee looks to the left, the workspace visualization shifts by hundreds of pixels to the right; when the mentee moves to the other side of the workspace as might be needed for best access during task performance, the visualization rolls 180°, which results in an upside-down visualization that is frustratingly difficult to parse. What is needed is a robust stabilization of the mentee first-person video, such that it can provide an effective visualization of the workspace to the mentor. The needed *high-level* stabilization has to neutralize the effects of substantial rotations and translations of the acquisition camera, and

cannot be provided by prior work *low-level* stabilization techniques that remove jitter in hand-held acquired video.

In this paper we present the design, implementation, and evaluation of a method for robust high-level stabilization of a video feed acquired from a mentee's first-person view, in order to provide a remote mentor with an effective visualization of the mentee's workspace. The output visualization has to be (1) stable, i.e. to show the static parts of the scene at a constant image location, (2) real-time, i.e. to keep up with the input feed, and (3) of high quality, i.e. without distortions, tears and other artifacts. In addition to conveying the workspace to the mentor, the output visualization should also be a (4) suitable canvas on which the mentor can author annotations to provide guidance. The paper investigates three approaches and adopts projective video texture-mapping onto a planar proxy of the workspace geometry, as the approach that best satisfies the design requirements. Figure 3.8 illustrates the robustness of our stabilization method on a variety of challenging workspaces.

We evaluated the effectiveness of our stabilization method in two controlled within-subject user studies. One study (n = 30) investigated workspace visualization quality by asking participants to find matching numbers in a video of a workspace annotated with numbers. The study used three workspaces: a *Sandbox*, a *Workbench*, and an *Engine* (the *Workbench* and the *Engine* are shown in Figure 3.8 without the numbers). In the control condition, participants watched the original (unstabilized) video acquired with the HMD camera; in the experimental condition, the video was stabilized with our method, which showed significant advantages in terms of task performance and participant workload. For the sandbox workspace we compared our method to a perfectly stable video acquired from a tripod, and there were no significant differences in performance. The second study tested our method in the context of surgical telementoring, where participants (n = 20) practiced cricothyroidotomy (cric) procedures on patient simulators (Figure 3.9). The study was conducted in an austere setting of an empty room, with the patient simulator on the floor, with poor visibility achieved with a fog machine, and with loud combat-like noises. Compared

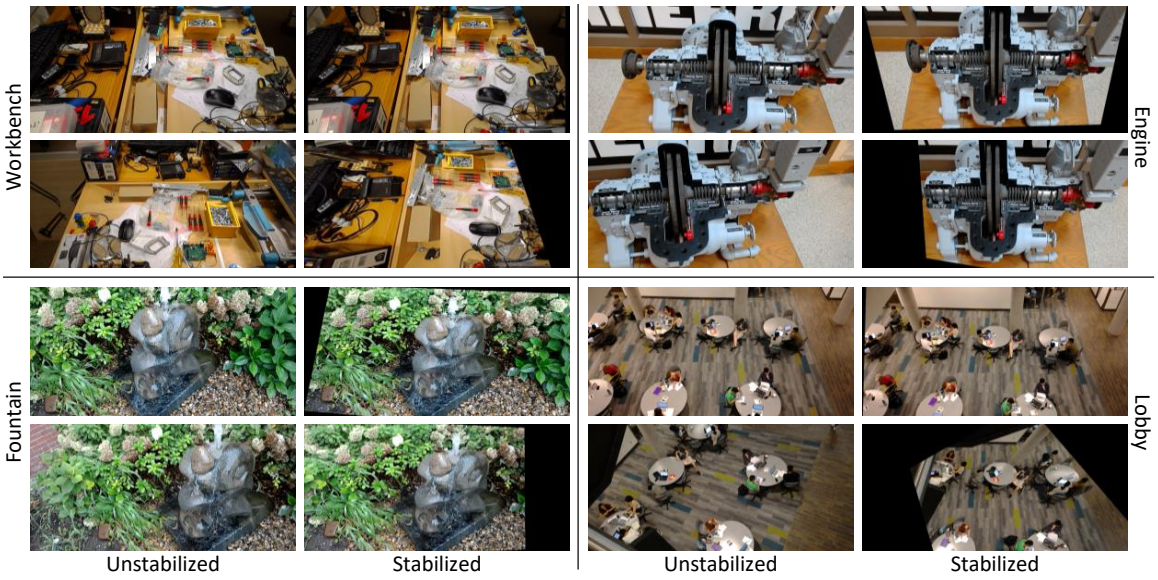


Figure 3.8.: Original (unstabilized) and stabilized video frame pairs for four sample workspaces. The videos are acquired with the camera built in an AR HMD worn by a user who walks around and rotates their head. Our method alleviates the view changes in the original first-person videos, which results in a stable visualization of the workspace, suitable for a remote collaborator, e.g. a mentor. Our method can handle complex 3D geometry (all examples), large view changes (*Workbench, Lobby*), large depths (*Lobby*), and dynamic geometry, complex reflectance properties, and outdoor scenes (running *Fountain*).

to audio-based telementoring, the stabilized video telementoring improved surgical performance significantly.

3.2.2 Prior Work

The widespread availability of digital cameras and of broadband internet connectivity enable telecollaboration by acquiring the local workspace with a video camera whose feed is transmitted to a remote site. An important design decision is where to place the camera in order to provide an effective remote visualization of the workspace.

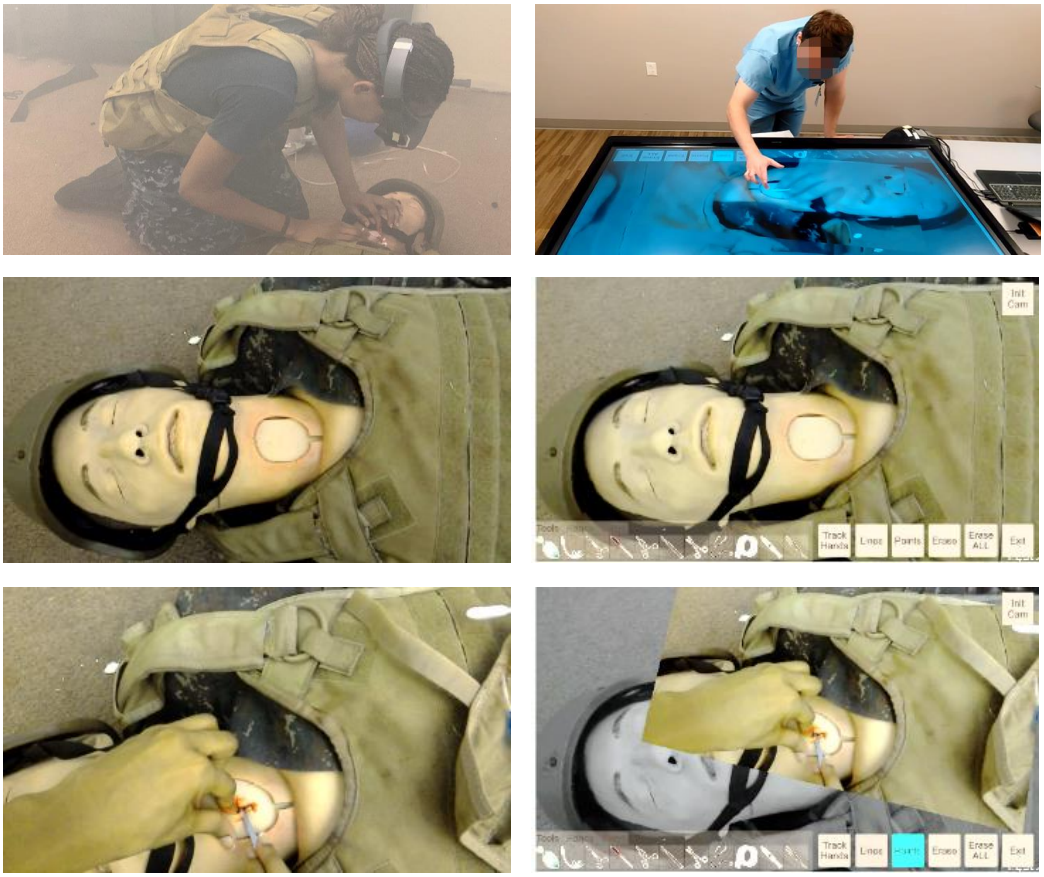


Figure 3.9.: Cricothyroidotomy training in austere environment using video feed stabilized with our method. The mentee wears an AR HMD that acquires the surgical field (top left), the video feed is sent to the mentor where it is stabilized (rows 2-3, raw left, stabilized right), the mentor annotates the stabilized feed (top right), and the annotations are sent to the mentee where they are displayed with the AR HMD. The first frame (grayscale) is used for context.

One approach is to mount the camera on a tripod. This approach was used to build a surgical telementoring system where the operating field was acquired with a ceiling-mounted overhead camera [38]. The top view is substantially different from the mentee's view, which reduces telementoring effectiveness, as a mentor can best guide a mentee when the mentor sees what the mentee sees, and when the mentor issues instructions in the mentee's frame of reference. Another surgical telementoring

system acquires the operating field with the back-facing camera of a computer tablet mounted with a bracket between the mentee and the patient [3]. The operating field is acquired from a view similar to that of the mentee, but the tablet creates workspace encumbrance. A shortcoming common to both systems is that the operating field is acquired from a fixed view. A second approach is to rely on the local site collaborator to acquire the workspace with a hand-held video camera, changing camera pose continually for a good visualization for the remote collaborator [41]. The problem is that the local collaborator becomes a cameraman, which hinders collaboration.

A third approach is to rely on a head mounted camera [42]. This brings freedom to the local collaborator, who can focus more on the task. A 360° video camera captures more of the environment and provides the remote collaborator with more awareness of the local space [43]. One disadvantage is having to wear the head mounted camera. The disadvantage has been alleviated as internet-connected cameras have been miniaturized, e.g. telecollaboration using Google Glass [44]. We have adopted this third approach. In our context, having to wear the head-mounted camera is not an additional concern since the mentee already has to wear an AR HMD.

The fundamental challenge of acquiring the workspace with a head-mounted camera is that the visualization of the workspace provided to the mentor changes abruptly, substantially, and frequently as the local collaborator moves their head during task performance. Such a visualization can lead to a loss of situational awareness, to a high cognitive load, to task performance delays and errors, and to cybersickness. Researchers have investigated addressing this challenge by attempting to stabilize the video such that it does not change as the local collaborator moves their head.

One approach of stabilization is to use optical flow to track features over the sequence of frames, to define homographies between consecutive frames using the tracked features, to register all frames in a common coordinate system, and to stabilize each frame by 2D morphing it to the common coordinate system [42]. A second approach is to acquire a 3D geometric model of the workspace, to track the video camera, and to projectively texture map the model with the video frames, from a

constant view. One option for acquiring the model is SLAM [41], another option is to use real-time active depth sensing. As we designed our stabilization technique, we investigated both of these approaches, as discussed in Section 3.2.3.

Researchers have developed low-level video stabilization techniques designed to remove small, high-frequency camera pose changes, such as the jitter of a hand-held camera [45, 46], or of a bicycle helmet mounted camera [47]. However, the large amplitude camera pose changes remain. If a hand-held camera is rolled 30°, low-level stabilization preserves the 30° roll, striving for a smooth angle change from 0° to 30°. In contrast, high-level stabilization aims to remove the 30° roll altogether.

Beyond technical challenges, researchers have also investigated video telecollaboration design from a user perspective, to optimize collaboration effectiveness. The problem of obtaining a good view of the workspace has been studied extensively in the context of telemedical consultation [48], where fixed, head-mounted, or hand-held cameras, 2D (view dependent) or 3D (view independent) interfaces each have advantages and disadvantages. A recent study finds that giving remote collaborators independent views is more beneficial than letting the local participant choose the view for the remote participant [49]. The benefit of view independence were also noted in the context of shared live panorama viewing [50], and remote instruction of cockpit operation [51]. Another study found that a scene camera was preferred in video telecollaboration over a head-mounted camera, not just by the remote helper who enjoyed the stable, comprehensive view of the workspace, but also by the worker who preferred not having to wear the camera [40].

Researchers have also demonstrated the acquisition of a complex environment with simple hardware, such as a tablet and its camera [41], to allow a remote collaborator a view-independent exploration of the environment; however, such systems are limited to static environments. Some systems allow the remote collaborator to suggest placement of objects in the workspace [52], again, under the assumption of an otherwise static environment. Complex dynamic scenes are handled by doing away with geometry acquisition, under the assumption that the entire scene is sufficiently

far away, which enables panorama acquisition and rendering [53], but this precludes nearby workspaces. Finally, dynamic geometry has been handled through the volumetric fusion of data acquired with multiple off-the-shelf depth cameras, which affords a remote collaborator an independent visualization of the workspace [54]; however, this comes at the cost of additional hardware, intractable in austere environments, and it is limited to the outside-looking-in scenario.

3.2.3 High-Level Stabilization of First-Person Video

Consider the AR telementoring scenario with a mentee wearing an optical see through AR HMD. The HMD has a built-in back-facing video camera that captures what the mentee sees. The goal is to use this video feed to inform a remote mentor of the current state of the workspace. In addition to audio instructions, the mentor also provides guidance through graphical annotations of the workspace. Therefore, the video feed should also serve as a canvas on which the mentor authors annotations of the workspace.

Effective Mentor-Side Visualization Requirements

An effective mentor-side workspace visualization has to satisfy the following requirements:

Stability. The visualization of the workspace should not move, to allow the mentor to examine it in detail. Complex tasks require for the mentor to concentrate on the workspace, and unexpected changes in the visualization are particularly frustrating, forcing the mentor to abandon the AR-enabled graphical communication channel, and to take refuge in the trusted audio communication.

View agreement. The mentor's view of the workspace should be similar to that of the mentee, for the mentor to provide guidance directly in the mentee's context, avoiding any remapping that could confuse the mentee. Furthermore, different viewpoints could show different parts of the workspace to the mentor and mentee, which

impedes communication when one party refers to workspace elements not visible to the other party.

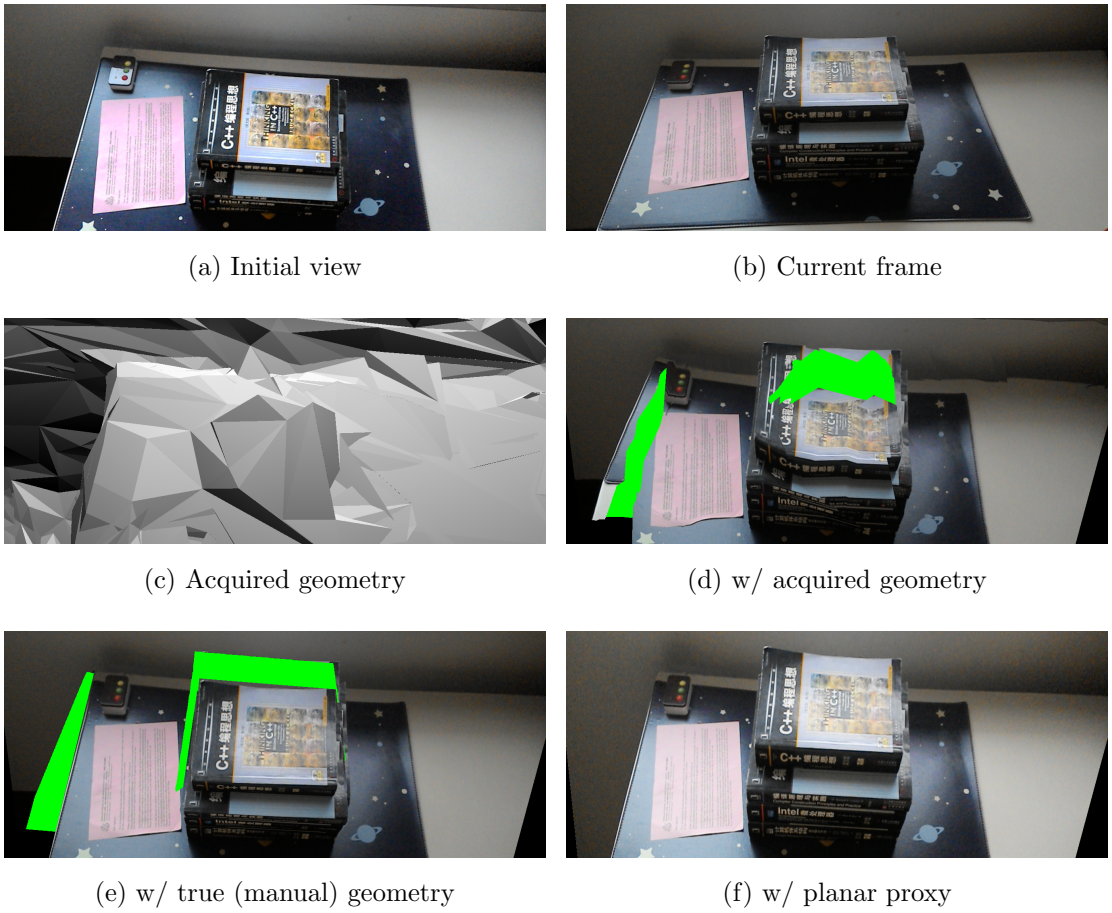
Real time. The visualization of the workspace should be up to date, as latency leads to workspace inconsistencies between mentor and mentee, complicating communication.

High visual quality. The visualization should be free of static and temporal artifacts such as tears, holes, and distortions. Of particular importance are scene lines, which should project to lines in the visualization. This is essential for the mentor's ability to understand and annotate the workspace.

Approaches Considered

Acquiring the workspace with a fixed camera satisfies the stability requirement, but not the view agreement one. A mentee-acquired first-person video satisfies the view agreement requirement, and it is well suited for austere environments since it does not require additional equipment. However, meeting the stability requirement is challenging. As the mentee looks away from the workspace, e.g. to grab a tool, the mentor's visualization changes abruptly and significantly.

The first is a 2D stabilization approach similar to the one described by Lee and Höllerer [42], based on tracking and stabilizing 2D video features. The approach lacked robustness in our context, with occasional incorrect feature tracking causing unacceptable stabilization artifacts. The second approach is based on the acquisition of workspace geometry (Figure 3.10). Real-time acquisition of complex 3D scenes is imperfect, resulting in stabilized frame distortions (Figure 3.10d); furthermore, the workspace has to be acquired from multiple viewpoints to avoid disocclusion errors (Figure 3.10e).



(a) Initial view (b) Current frame

(c) Acquired geometry (d) w/ acquired geometry

(e) w/ true (manual) geometry (f) w/ planar proxy

Figure 3.10.: Stabilization of current frame (b) to initial view (a) by projective texture-mapping onto acquired (c, d), truth (e), or proxy geometry (f). Disocclusion errors are highlighted in green.

Stabilization by Projection on Planar Proxy

The third approach investigated, which we adopted, is to projectively texture map the tracked video feed onto a planar approximation of the workspace geometry. The planar proxy is defined once per session. Rendering the textured planar proxy takes negligible time, even on the thinnest of mentor platforms, such as a computer tablet or a smartphone, so the visualization is real time. The visualization is of high quality (Figure 3.10f), i.e. without distortions due to inadequate geometric approximation,

and without tears due to disocclusion errors. All scene lines project to lines in the visualization. The effect is similar to a photograph of a photograph of a 3D scene. The concatenation of an additional projection does not make the visualization confusing, the same way a visualization makes sense to two or more users seeing it on a display, with no one assuming the true viewpoint from where it was rendered.

3.2.4 Theoretical Visualization Stability Analysis

The two possible sources of visualization instability are workspace geometry approximation error, and video camera tracking error. In this section we provide a theoretical analysis of the impact of these two errors on visualization stability. In the next section we provide empirical measurements of visualization stability.

Visualization instability definition

Given a 3D workspace point P, an initial frame F_0 with view V_0, and a current frame F_i with view V_i, we define the reprojection error of P as the distance $e_i(P)$ between where P should be seen from V_0 and where it is actually seen in the stabilized F_i. In Equation (3.2), the actual location of P in the stabilized frame is denoted with $\chi(P, V_i, V_0)$, and the correct location $\pi(P, V_0)$ is obtained by projecting P with V_0. The approximate projection function χ depends on the stabilization approximation errors. $e_i(P)$ is relative to the frame's diagonal d to obtain an adimensional, image resolution independent measure of reprojection error.

$$e_i(P) = \frac{\|\chi(P, V_i, V_0) - \pi(P, V_0)\|}{d} \tag{3.2}$$

Given a point P and two consecutive frames F_i and F_{i+1}, we define visualization instability at P as the absolute change in reprojection error from F_i to F_{i+1}, as given by Equation (3.3).

$$\epsilon_i(P) = |e_{i+1}(P) - e_i(P)| \tag{3.3}$$

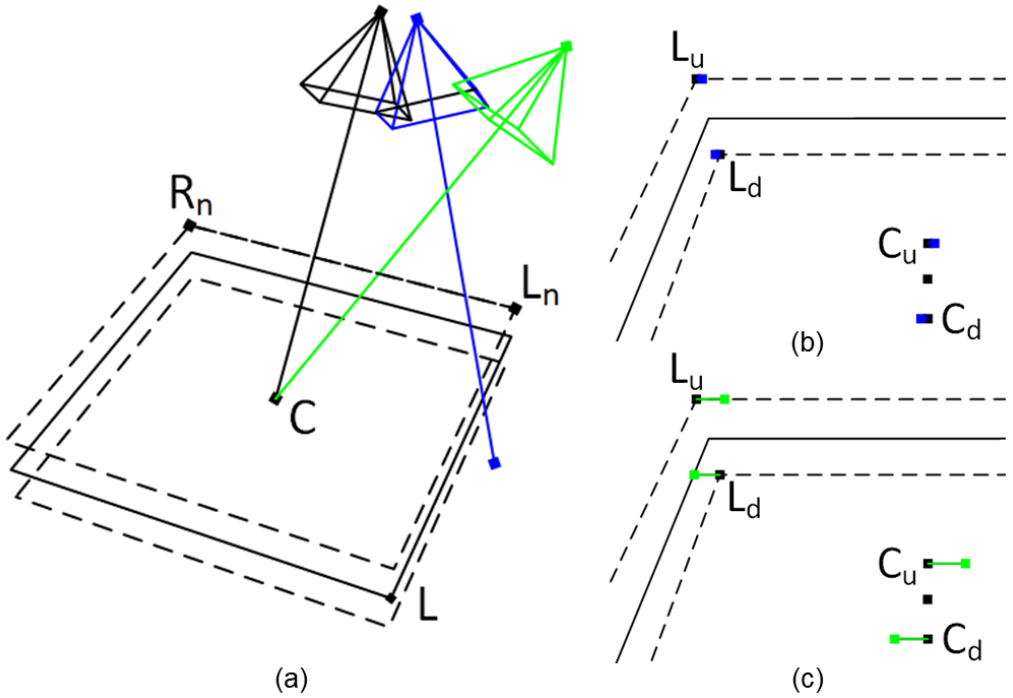


Figure 3.11.: Visualization stability analysis through simulation.

Simulation scenario

We analyze visualization instability in a typical telementoring scenario. The workspace is 1m×1m wide, and it is 1m above the floor (Figure 3.11a). This is the largest workspace size for which the mentee can work in the outside looking in scenario—for larger workspaces the mentee would have to travel from one area to another, and stabilizing the mentor view to a single view is not applicable. The actual workspace geometry is in between two planes (dotted lines) that are 20cm apart. This height variation is sufficient to model a workbench with tools on it. The workspace geometry is approximated with the solid line rectangle. The mentee is 1.8m tall, and their default view, to which the video is stabilized, is shown with the black frustum.

We consider two typical mentee view sequences. The first sequence is a 25° pan to the left (blue frustum in Figure 3.11a), as needed, for example, to reach for a tool placed just outside the workspace. The panning sequence also has a small lateral

translation of 10cm, to account for the translation of the eyes when someone turns their head to the side. The second sequence corresponds to the mentee moving to the corner of the workspace to see it diagonally (green frustum in Figure 3.11a), which implies a 50cm lateral translation from the initial position, while looking at the center of the workspace. Instability depends on frame to frame view changes. We assume the sequence is completed in 1s, which implies 30 frames at 30Hz. This is a conservative upper bound for the view change speed. For abrupt focal point changes, the mentee does not want to and cannot focus on the workspace during the transition, so any instability will not be perceived, as also noted in walking redirection research that takes advantage of saccadic eye movement to manipulate the visualization [55].

Dependence on Geometry Approximation Error

In Figure 3.12, point P is acquired by video frame V_i and projected onto the proxy plane w at P^G. P and P^G project at different locations onto the stabilized view V_0, which results in the reprojection error $e_i^G(P)$. The dependence of visualization instability on geometry approximation error is obtained by plugging into Equation (3.3) the expression for χ given in Equation (3.4), where $V_iP \cap w$ is P^G in Figure 3.12.

$$\chi(P, V_i, V_0) = \pi(V_iP \cap w, V_0) \tag{3.4}$$

The instability induced by geometry approximation error is largest where the true location of a workspace point is farthest from the proxy plane, i.e. on the dotted rectangles in Figure 3.11. Figure 3.11 illustrates the reprojection errors at the center C and corner L of the workspace proxy, for the last frames of the panning (Figure 3.11b) and translation (Figure 3.11c) sequences. The correct projections of L_u, L_d, C_u, and C_d are shown with black dots. The actual projections are shown with blue dots for the panning sequence, and with green dots for the translation sequence. As expected, the reprojection error is tiny for the panning sequence since the viewpoint translation is minimal. Pure panning would have a zero reprojection error.

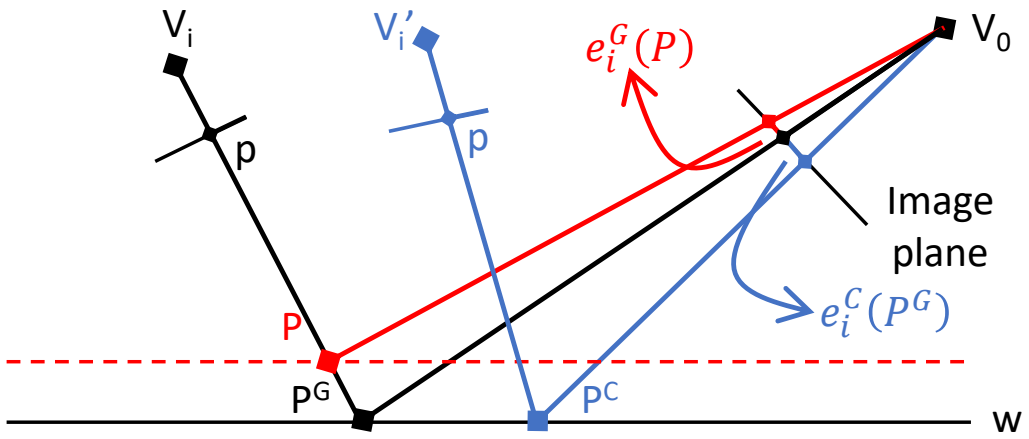


Figure 3.12.: Reprojection error $e_i^G(P)$ due to workspace geometry approximation error, and $e_i^C(P^G)$ due to camera tracking error.

Table 3.6.: Visualization instability due to geometric approximation error for two mentee sequences.

	Panning	**Translation**
Center	0.03%	0.17%
Max	0.05%	0.25%

Table 3.6 gives the visualization instability for each of the two sequences. The maximum instability at the center of the workspace (i.e. C in Figure 3.11) is 0.03% and 0.17% for the panning and translation sequences, respectively. The maximum is reached for the last frame of the sequence, where the viewpoint translation is largest. For an HDTV display with a diagonal of 2,200 pixels and 1m in length, the instability figures translate to 1.1pix and 0.5mm for the panning sequence, and 5.5pix and 2.5mm for the translation sequence. We computed the maximum instability over the entire workspace to be 0.05% and 0.25% for the two sequences, respectively, which occurs at the workspace corners, i.e. L_n and R_n in Figure 3.11a, for the last frame.

An important advantage of our method is that the geometric approximation is constant, i.e. the proxy plane does not change. This means that, when the mentee translates their viewpoint, the instability is not only small, but also smooth, and when the mentee pauses to focus on a part of the workspace, the instability is 0. For a method that uses a geometric model acquired in real time, the instability is noisy, even when the mentee does not move.

Dependence on Camera Tracking Error

The second source of visualization instability is the error in tracking the video camera which acquires the workspace. Using Figure 3.12 again, let us now assume that proxy plane point P^G is an actual workspace point to factor out all geometry approximation error. P^G is captured at pixel p by the frame with true viewpoint V_i. If V_i is incorrectly tracked at V_i', then p is incorrectly projected onto the proxy at point P^C, which generates reprojection error $e_i^C(P^G)$. The dependence of visualization instability on camera tracking error is obtained by plugging into Equation (3.3) the expression for χ given by Equation (3.5), where w is the workspace proxy.

$$\chi(P, V_i, V_0) = \pi(V_i' p \cap w, V_0) \tag{3.5}$$

Unlike for the instability due to the workspace geometry approximation, tracking inaccuracy affects the entire frame uniformly. We have measured tracking accuracy to be 2 degrees for rotations and 2cm for translations. In the scenario above, these maximum tracking errors translate to a 2.68% and a 1.45% instability, figures that dwarf the instability caused by geometric approximation error (Section 3.2.4). Even assuming tracking that is an order of magnitude more accurate than what our AR HMD provides, instability due to geometry approximation will still be smaller than instability due to tracking.

In conclusion, we have defined instability metrics to be used in the empirical validation, and we have established that instability due to geometric error is dwarfed by that due to camera tracking error, which validates, at principle level, our approach.

3.2.5 User Study I: Number Matching

We developed a method for stabilizing the video of a workspace captured by a head mounted camera. The stabilized video serves as a visualization of the workspace for a remote collaborator. In a first controlled user study, we tested the effectiveness of workspace visualization by asking participants to find matching numbers in the original and the stabilized videos, for three workspaces.

Experimental Design

Participants. We recruited participants (n = 30, 8 female) from the graduate student population of our university, in the 24–30 age group. We opted for a within-subject design, with each participant performing the task in all conditions.

Task. A participant was seated 2m away from an LCD monitor with a 165cm diagonal. The monitor displays a video of a workspace annotated with numbers, and the participant is asked to find pairs of matching numbers. When a participant spots a matching pair, they call out the number, and an experimenter tallies the number of matches found. All numbers called out by participants were correct matches, i.e. they were not just reading out numbers at random.

Workspace 1: Sandbox. The first workspace is a sandbox in our lab (Figure 3.13). The sandbox is approximately 1m×1m in size, and it is placed about 1m off the floor. The sand had a depth variation of about 20cm, so this corresponds to the scenario investigated by the theoretical instability analysis in Section 3.2.4. An overhead projector displays a matrix of 4×4 numbers on the sandbox. The workspace was acquired with the back-facing camera of an AR HMD (i.e. Microsoft's HoloLens [56]) worn by an experimenter who walked around the sandbox while looking at its

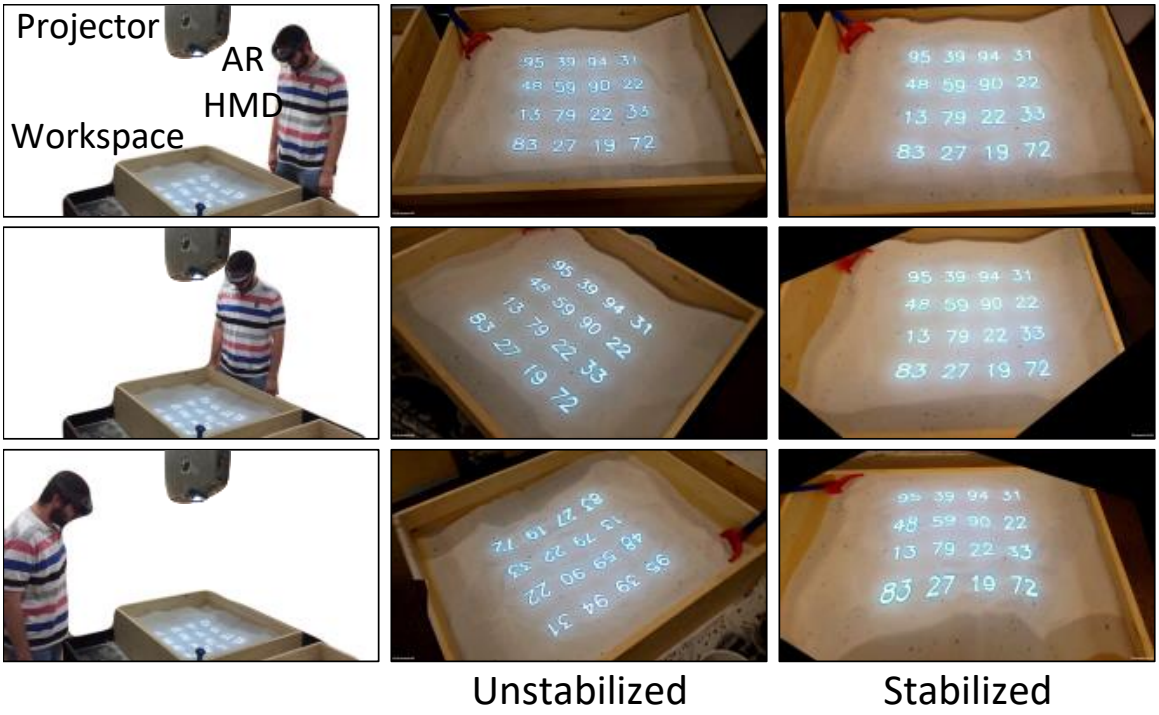


Figure 3.13.: Sandbox workspace with overhead projected numbers acquired with video-camera built into an AR HMD (left column), original, unstabilized video frame (middle), and stabilized video frame (right).

center. The experimenter starts out at the default position, where the numbers are correctly oriented (first row of Figure 3.13). This is also the view to which the video was stabilized. The experimenter occasionally pans the view to the side. Then the experimenter walks to the corner of the sandbox (second row of Figure 3.13), and even on the other side, which makes the numbers appear upside down in the video (third row of Figure 3.13). This results in a video sequence where the matrix of numbers moves considerably. The video shows 21 matrices, and each matrix was shown for 5s, for a total video length of 105s. 18 of the 21 matrices had exactly one pair of matching numbers, and 3 of the matrices had no matching numbers. Half the numbers of two consecutive matrices are the same, which means that when the video switches from one matrix to the next, exactly 8 of the 16 numbers change. All 8 numbers change

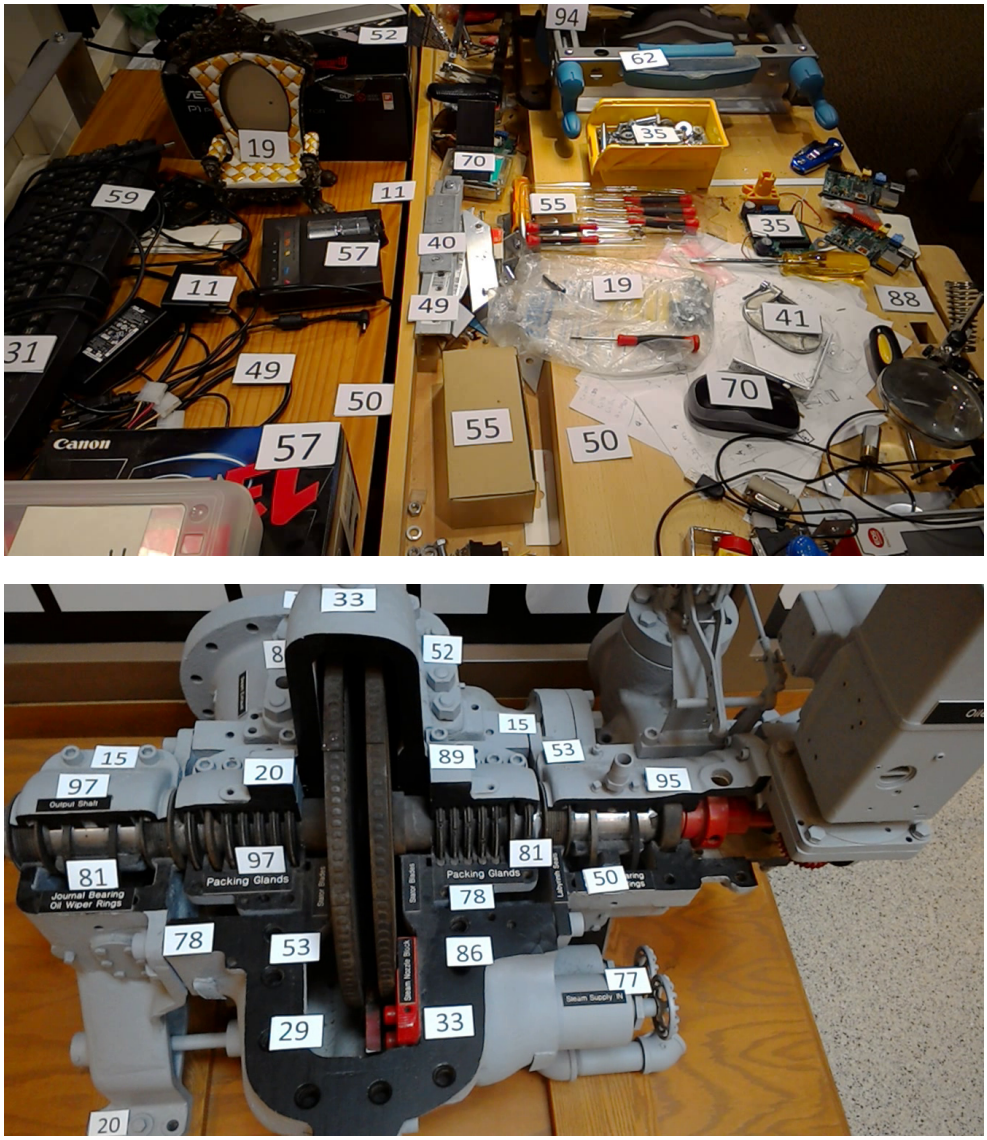


Figure 3.14.: *Workbench* (top) and *Engine* workspaces used in study I.

simultaneously at the end of the 5s. When a matrix had a matching pair, at least one of the numbers in the pair was replaced for the next matrix, such that a matching pair would not persist longer than the 5s that each matrix is displayed.

Workspace 2: Workbench. The second workspace is an actual workbench cluttered with tools (Figure 3.8 and Figure 3.14). The acquisition path was similar to that for the *Sandbox* workspace. The tallest tool reached 30cm above the workbench plane.

The experimenter wearing the AR HMD impersonating a mentee started out at the default position, then panned the view, and then finally moved to the side of the workbench to see it from a direction rotated by 90°. The numbers were added to the workspace using pieces of paper, all facing the mentee in the initial position. There were 24 numbers, 8 of which appeared twice, so 8 numbers were unique. Although the numbers on paper did not change, the mentee moved tools on the workbench covering and uncovering a few numbers. Furthermore, as the mentee viewpoint translated, some of the numbers would appear and disappear due to occlusions.

Workspace 3: Engine. The third workspace is an Engine mounted on the floor, 80cm high (Figure 3.8 and Figure 3.14). The *Engine* was decorated with numbers and was acquired similarly to the *Workbench*.

Conditions. Each participant performed the number matching task for the *Sandbox* workspace in each of three conditions, in randomized order. In one control condition, the participant was shown the raw video with no stabilization (NS). In a second control condition, the participant was shown a perfectly stable (PS) video that was acquired by placing the AR HMD on a mannequin head mounted on a tripod at the default position. In the experimental condition, the participant was shown the video stabilized with our method (S). The hypotheses related to the *Sandbox* were that (1) participants will perform better in the S condition compared to the NS condition, and that (2) participants will not perform better in the PS condition compared to the S condition. A subgroup of 20 participants were tested for each of the *Workbench* and the *Engine* workspaces, for each of two conditions. Participants were shown the original, unstabilized video in the control condition, and the stabilized video in the experimental condition.

Metrics. We measured participant task performance as the number of pairs found. We also measured participant workload using the NASA Task Load Index (NASA-TLX) questionnaire [57], and participant simulator sickness using the Simulator Sickness Questionnaire (SSQ) [58]. Better performance means more matching pairs found, lower cognitive load, and absence of simulator sickness.

Results and Discussion

A within-subject statistical analysis compared the three *Sandbox* conditions, with three data points for each metric and for each participant. The participants and the order of the trials were treated as blocks in the statistical design. The data normality assumption was confirmed with the Shapiro-Wilk test [30]. In addition, the data equal-variance assumption was confirmed with the Levene test [59], so no data transformation was needed. We ran a repeated measures ANOVA [60] with Bonferroni correction [61] for each condition pair, i.e. PS vs NS, PS vs S, and S vs NS. The two conditions for the *Workbench* and *Engine* were similarly compared, except that no Bonferroni correction is needed.

Figure 3.15 gives the box and whisker plot [62] of the number of pairs found, and of the six NASA-TLX subscales, for each of the three *Sandbox* conditions. The six subscales are: mental demand, physical demand, temporal demand, performance, effort, and frustration. All seven metrics are normalized. The plot indicates the inter-quartile range (IQR) with a box, the average value with an x, the median value with a horizontal line, farthest data points that are not outliers with whiskers, and outliers with dots. Outliers are data points "outside the fences", i.e. more than 1.5 times the IQR from the end of the box. NS participants found on average 28% or 5.1 of the 18 matching pairs. S participants found on average 36% or 6.5. PS participants found on average 34% or 6.3. The differences between S and NS, and PS and NS are significant, while the difference between PS and S is not. The best performing participant found 12 of the 18 matching pairs for both the S and PS conditions, performance levels that are within the fence and therefore not outliers; this participant only found 8 matching pairs in the NS condition.

S and PS participants reported significantly lower cognitive load than those in NS on all six NASA-TLX subscales, and there was no significant difference between PS and S. For NS, the upper fence exceeded the maximum possible value of 1.0, and it was therefore capped at 1.0, for all six NASA-TLX subscales. This indicates the high

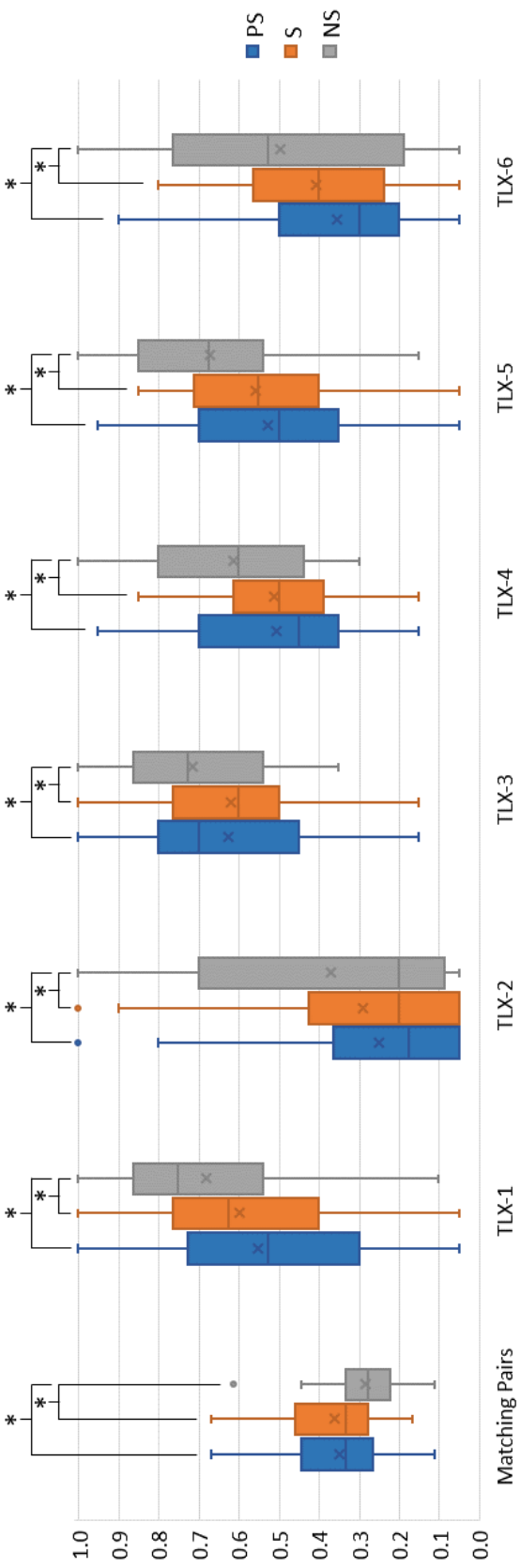


Figure 3.15.: Normalized box and whisker plot of pairs found, and of NASA-TLX subscores, for each of the three *Sandbox* conditions: perfectly stabilized (PS), stabilized (S), and not stabilized (NS). The star indicates significance ($p \leq 0.05$). No S to PS difference was significant.

Table 3.7.: Comparison between the number of pairs found in the no stabilization (NS) and stabilization (S) conditions.

Workspace	NS	S	S - NS	p-value
Workbench	5.45±0.83	5.95±1.19	0.50±0.28	0.043*
Engine	5.05±1.57	6.10±1.29	1.05±0.31	0.002*

cognitive load in the NS condition, and it eliminates the possibility of outliers. For S and PS, two of the scales had the upper fence at 1.0, which leaves the possibility of outliers for the other four scales. However, there was only one outlier for each of the PS and S conditions, both for the TLX-2 scale, which increases the confidence that PS and S place less demand on the participant.

Table 3.7 gives the number of pairs found for the *Workbench* and the *Engine* workspaces, for each of the unstabilized (NS) and the stabilized (S) conditions. S has a significant advantage for both workspaces. Table 3.8 compares the NASA TLX scores between the S and NS conditions (i.e. NS-S, as lower NASA TLX scores indicate less demand on the participant). Most S advantages are significant. For the *Sandbox* workspace, the analysis of the Total Severity score derived from the SSQ answers indicates the absence of simulator sickness in all three conditions. Furthermore, there are no significant differences for any of the three differences PS-NS, S-NS, and PS-S, for any SSQ subscore. While this suggests that our stabilization might not induce simulator sickness, and that discomfort levels are similar to those for a perfectly stabilized video, the absence of differences between PS and NS indicates that the exposure might have been too short and the workspace too simple for a revealing simulator sickness comparison between the three conditions.

The SSQ provided more insight in the case of the more visually complex *Workbench* and *Engine* workspaces (Table 3.9). S had a significant advantage over NS in terms of Total Severity score, for both workspaces. The S advantage was due to less

Table 3.8.: p-values of NASA TLX subscore differences between no stabilization (NS) and stabilization (S) conditions (i.e. NS-S).

Workspace	Mental Demand	Physical Demand	Temporal Demand	Perfor-mance	Effort	Frustration
Workbench	0.000*	0.000*	0.001*	0.188	0.356	0.001*
Engine	0.005*	0.050*	0.000*	0.034	0.002*	0.001*

Table 3.9.: p-values of SSQ Total Severity score differences between no stabilization (NS) and stabilization (S) conditions (i.e. NS-S).

Workspace	Nausea	Oculomotor	Disorientation	Total Severity
Workbench	0.019*	0.001*	0.116	0.004*
Engine	0.053	0.060	0.019*	0.021*

nausea and oculomotor effort for the flatter but more cluttered *Workbench*, and due to disorientation for the more occlusion/disocclusion prone *Engine*. Although the differences between conditions were significant, for no workspace and no condition did the Total Severity score increase from pre- to post- exposure above the threshold of 70, which would indicate the presence of simulator sickness.

Empirical Visualization Stability Analysis

Section 3.2.4 defined visualization instability and analyzed its dependence on the workspace geometry approximation error and on the camera tracking error. Here we measure the actual instability in the raw video and in the stabilized video by tracking nine salient feature points over the entire *Sandbox* sequence. The features are dark particles mixed in with the white sand, and they cover the matrix area uniformly. The frame trajectories of the tracked features are shown in Figure 3.16, where the

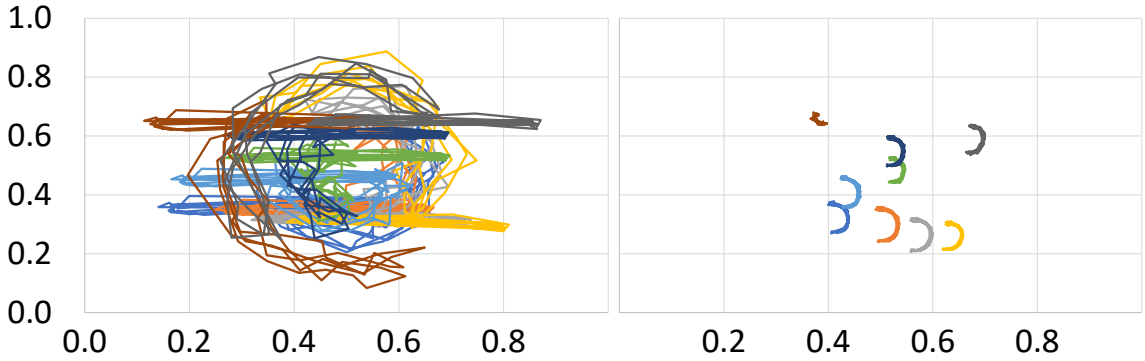


Figure 3.16.: Trajectories of 9 tracked feature points, in normalized pixel coordinates, for the NS (left) and S (right) *Sandbox* conditions.

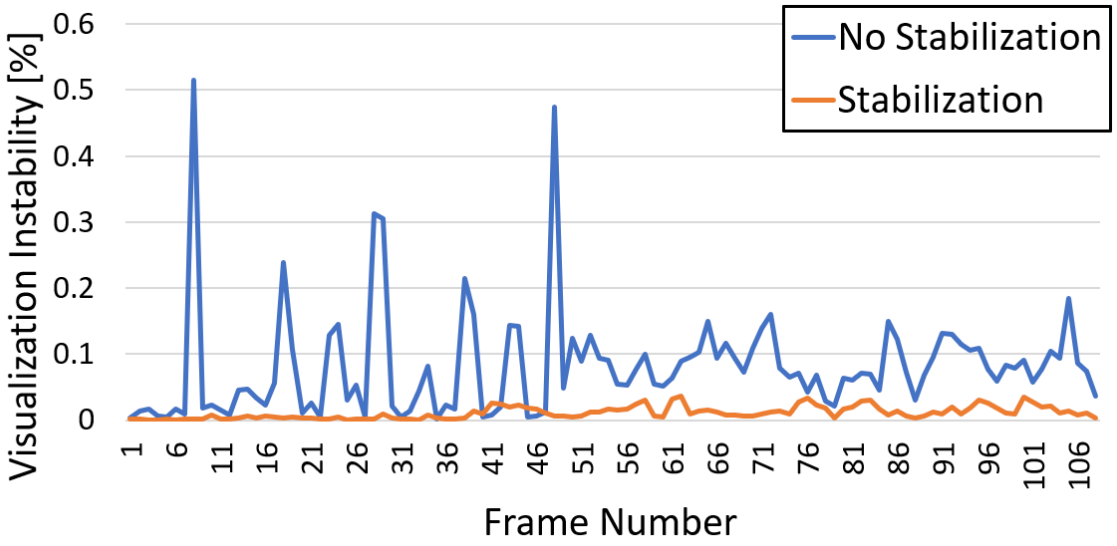


Figure 3.17.: Empirical visualization instability measured by tracking feature points over the video sequences.

coordinates in the 1,280×720 video frame were normalized. Whereas the tracked points move considerably in the NS video, their trajectory is short and smooth in the S video. The average reprojection error (Equation (3.2)) over all feature points and all frames is 13.5%±7.9% for NS and 2.0%±1.8% for S; the maximum reprojection error is 37.5% for NS and 5.8% for S.

The average visualization instability (Equation (3.3)) over all 9 feature points is given in Figure 3.17 for both the unstabilized and the stabilized sequences. These instability values are based on empirical values for the $\chi(P, V_i, V_0)$ and $\pi(P, V_0)$ from the definition of reprojection error Equation (3.2). Instability is large for NS, and it is largest for the first part of the sequence, when the mentee panned their head left and right repeatedly. This is expected since, for a non-stabilized sequence, panning motions change the frame coordinates of workspace features quickly and substantially. Instability is low for our stabilized sequence, and it is lower for the first part of the sequence when workspace geometry approximation error has little influence on instability. For the first part of the sequence, the instability is very low most of the time, with the exception of some small spikes which we attribute to camera tracking latency. The average instability is $0.081\% \pm 0.082\%$ for the NS sequence, and about eight times lower for the S sequence at $0.011\% \pm 0.0093\%$.

3.2.6 User Study II: Austere Surgical Telementoring

We conducted a second user study, which tests the benefits of stabilization in the context of a complete surgical telementoring system. The mentee acquires the surgical field with a back-facing video camera built into their AR HMD, the video is transmitted to the remote mentor site, the video is stabilized, the stabilized video is shown to the mentor, the mentor provides guidance by annotating the stabilized video, and the annotations are sent to the mentee site, where they are overlaid onto the surgical field using the AR HMD. The study evaluates the benefit of stabilization indirectly: the hypothesis is that the stabilized video leads to a better mentor understanding of the operating field, to better guidance for the mentee, and ultimately to better mentee performance.

Experimental Design

Participants. The participants served as mentees in the study. We recruited participants ($n = 20$) from the corpsmen of a naval medical center who were training for performing surgical procedures in austere settings. The participant age range was 18–43, and 3 participants were female. The study used two mentors that are teaching faculty at a surgery residency program. The mentor site was 900km away from the mentee site. We opted for a within-subject design, with each participant performing a task in both conditions.

Task. The participants performed a practice cric on a synthetic patient simulator in an austere setting (Figure 3.9). The cric is an emergency procedure performed when a patient is not able to breathe due to airway obstruction. The procedure entails performing precise incisions through multiple layers of neck tissue, opening up the cricoid cartilage, inserting and securing a breathing tube, and connecting a breathing bag to the tube. Since emergent, the procedure stands to benefit greatly from telementoring.

Conditions. In the experimental condition (EC), the mentee benefited from visual and verbal guidance from the mentor. The visual guidance was provided through the AR HMD, which overlaid mentor-authored annotations onto the operating field, such as freehand sketched incision lines, or dragged-and-dropped instrument icons. The mentor monitored the operating field and authored annotations based on a first-person video of the operating field acquired by the mentee, which was stabilized with our method. In the control condition (CC), the mentee benefited from verbal mentor guidance.

Metrics. The mentee performance was evaluated by two expert surgeons located at the mentee site. The experts used the cric evaluation sheet typically used at the naval center to score the performance of the mentees. The evaluation sheet contains 10 subscales based on procedure steps, which are scored with a 5-level Likert Scale. The subscales evaluate aspects related to anatomical landmark identification, incision

performance, and patient airway acquisition. The overall mentee performance score was computed as the average of the 10 subscale scores.

Results and Discussion

A within-subject statistical analysis was run to compare both conditions, with two data points for each metric and for each participant. The condition was treated as an independent variable, while each of the expert evaluation scores were treated as dependent variables. The participants and the order of the trials were treated as blocks in the statistical design. The data normality and equal variance assumptions were confirmed with the Shapiro-Wilk [30] and the Levene test [59], respectively, and a repeated measures ANOVA was run [60].

The results are shown in Figure 3.18, which gives means and standard deviations. The total performance score (EE-T) was significantly higher ($p = 0.04$) for EC than for CC. The means for each of the ten subscale scores (i.e. EE-1 to EE-10) favor EC over CC, but only two of the differences are significant, i.e. for EE-8 ($p = 0.03$) and for EE-9 ($p = 0.01$). We attribute the lack of significance for the score differences for the other subscales to the low number of participants. EE-8 verifies that the cuff of the Melker canula was inflated with 10ml of air, which indicates that there is air circulating through the tube. EE-9 verifies that the air actually makes it into the lungs of the patient (simulator) as indicated by a bilateral rise and fall of the chest. On the other hand, EE-10 verifies that the cannula is properly secured with tape for patient transport, so it concerns a step beyond the end of the actual cric, and participants could score highly on EE-10 even if the procedure actually failed. Thus, EE-8 and EE-9 are important scores that depend on the success of all previous steps, and they validate the entire procedure.

The mentee moves their head considerably as they reach for surgical instruments, which causes numerous, substantial, and abrupt changes in the input video. In one typical instance, a mentee translated their head for a total of 7.66m over a 3min and

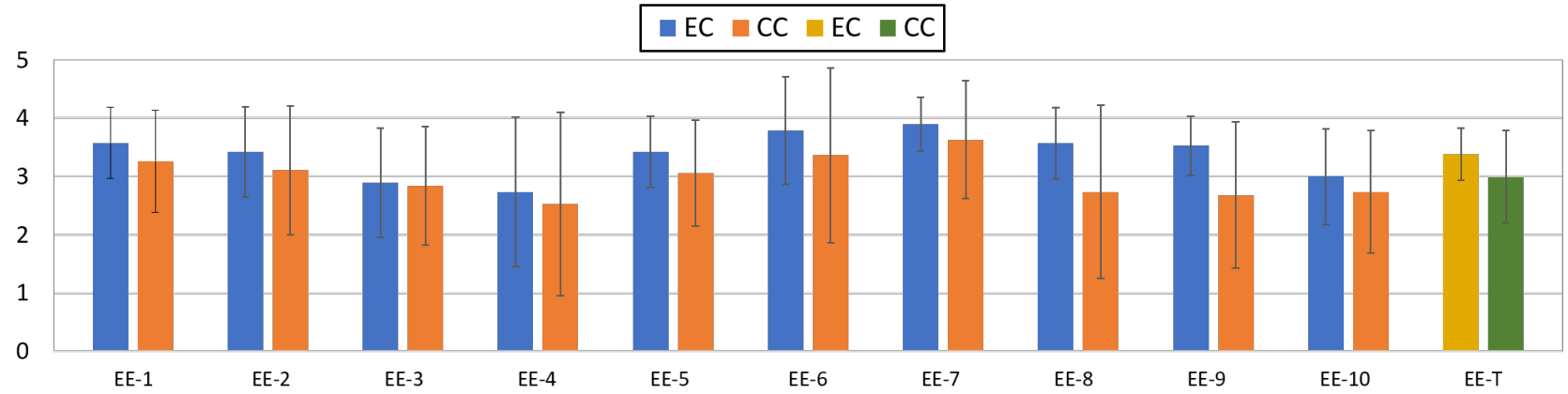


Figure 3.18.: Procedure subscale (EE-1 to EE-10) and overall (EE-T) cric performance. EC has an advantage over CC for each metric. The star indicates a significant advantage ($p \leq 0.05$).

12s sequence, with spikes of over 20cm per second. In the same sequence, the mentee rotates the view direction by over 1,500°, which is more than four full rotations. These large view changes make the raw video unusable at the mentor, and our stabilization is essential to the success of the AR telementoring system.

The workspace in the surgical telementoring study is highly dynamic, with the mentee's hands and instruments moving in the video feed. While such dynamic environments are challenging for approaches that rely on real time geometry acquisition, the dynamic workspace does not pose any additional challenge to our approach. Note that our definition of instability (Equation (3.3)) does apply to dynamic environments since it does not simply measure how far the projection of a 3D point moves from one frame to the next, which would penalize the moving elements of the environment even in a perfectly stabilized visualization; instead, our definition is based on how far away the 3D point is in the visualization from where it should be in a perfectly stabilized visualization.

3.2.7 Conclusion, Limitations, and Future Work

We have presented the design and evaluation of a method for stabilizing a first-person video of a workspace, such that it can effectively convey the workspace to a remote collaborator. We investigated three approaches and we chose an approach that projectively texture maps the registered video feed onto a planar proxy of the workspace. The approach has the advantages of stability, view agreement, real time performance, lack of distortions, lack of disocclusion errors, good temporal continuity, and robustness with workspace geometric, reflectance property, and motion complexity.

The stabilized video doesn't always contain all the pixels in the input mentee video. This happens when the mentee view frustum is not a subset of the mentor view frustum. For example, in Figure 3.8, for the *Engine* workspace, the top left unstabilized frame captures more of the text on the wall than its stabilized counter-

part. This is due to the fact that the mentor view frustum was chosen to encompass tightly the workspace, i.e. the engine. A wider mentor field of view would have kept the entire back wall pixels in the stabilized frame. Certainly, this would come at the cost of a lower resolution on the workspace, and each application should decide what works best in its own context. Another possibility to be explored as future work, is to not insist on a fixed mentor view, but rather a view that slowly keeps up with the mentee view in order to show the mentor everything the mentee sees. For example, if the the mentee chooses to focus on a completely different area of the workspace, the mentor view should gradually focus on that area as well.

When the mentee looks away from the workspace, the mentor's live visualization of the workspace is truncated, or even interrupted if the mentee view frustum is completely disjoint from the mentor view frustum. One solution for mitigating this problem is to rely on previous frame pixels to maintain workspace visualization continuity. Of course, these are not live pixels so they can only be used for orientation purposes, and not for up to date situational awareness. We took this approach in the cric study, where the a previous frame is used to provide context (see frame in Figure 3.9, row 3, right). The background frame is shown in grayscale to make it clear to the mentor that it is not a live shot. Future work could explore updating the background frame to keep up with a dynamic workspace, i.e. to be more recent and less obsolete. Another direction of future work is to rely on a series of background images and to rely on an approach similar to projective texture mapping to choose the most suitable background image for the current frame. Suitability can be quantified as the number of missing mentor frame pixels that are filled in, which requires view direction similarity, and as the continuity of the transition from live to background pixels, which requires viewpoint similarity.

One limitation to address in future work is that our first study does not provide a sufficiently long exposure to measure simulator sickness. Another direction of future work is to examine conveying the workspace to the remote collaborator through a Virtual Reality (VR) HMD, where simulator sickness is likely to be a bigger factor.

The current method aims to project the video acquired from one view to a stationary default second view. Future work could examine projecting the acquired video to a stable but changing second view. Indeed, if the mentee moves to a different part of the workspace, e.g. they move from the head to the legs of the patient, the mentor visualization should not remain stuck on the head, but rather gradually catch up with the mentee view. Such gradual view change should discern between abrupt and short term view changes, as needed, for example, to change focal point on a workspace that is captured by the default view, or to grab an instrument, and a long term view change implied by a location change within a large workspace. View stabilization should eliminate the former, and gradually adopt the latter. Finally, our work could be extended to transfer one user's first-person view to the first-person view of a second user, allowing the second user to change the view on the workspace interactively.

The second user study compared AR telementoring based on our stabilization to a control condition where the mentor and mentee could only communicate through audio. One reason for this is that audio communication is the most frequently used means of communication between mentor and mentee. The second reason is that the unstabilized video was judged by the expert surgeon mentors as unusable in the context of the emergent cric and of the austere conditions. In other words, it was not possible to run a user study where one of the conditions was AR telementoring with the raw, unstabilized video. Future studies could attempt to isolate the stability factor in settings where the surgical intervention and the environment are less stressful to make the unstabilized video acceptable, at least for the purpose of a user study.

Our work tests AR surgical telementoring with actual health care practitioners, in a real training exercise, in a highly demanding austere setting, towards placing AR technology into societal service.

3.2.8 Acknowledgments

We thank our Augmented Reality Tea group for insightful comments and suggestions. This work was supported by the United States Office of the Assistant Secretary of Defense for Health Affairs under Award No. W81XWH-14-1-0042, and by the United States National Science Foundation under Grant DGE-1333468. The views expressed in article, reflect the results of research conducted by the author(s) and do not necessarily reflect the official policy or position of the funders, including but not limited to the Department of the Navy, Department of Defense, or the United States Government.

4 FAST INTRA-FRAME VIDEO SPLICING FOR OCCLUSION REMOVAL IN DIMINISHED REALITY

4.1 Introduction

Augmented reality improves a user's view of the real world by overlaying graphical annotations that provide information about the real world objects to which they are anchored. Sometimes, however, improving the user's view of the real world calls for the removal of some objects from the user's view. Motivations for such diminished reality visualizations include eliminating distracting clutter, investigating changes to a real world scene efficiently without actually modifying the scene, and giving the user line of sight to parts of the scene occluded by objects of little interest.

Removing an occluder from the user's view of the real world requires finding the footprint of the occluder in the user's view, computing what the user should see in the absence of the occluder, and transferring it to the occluder footprint. One approach is 3D scene acquisition. Once the geometry of the scene is known, segmenting the occluder is easier, and rendering the scene without the occluder from the user's viewpoint provides exactly what the user would see in the absence of the occluder. This works well, for example, when one wants to remove an object from a corner of a room, since the color and the geometry of the three planes defining the room corner is simple and can be easily acquired.

A challenging case for this geometry acquisition approach is when the parts of the scene hidden by the occluder have intricate and dynamic appearance and geometry. Acquiring such a scene in real time, and with minimal equipment remains challenging. The scene geometry has to be acquired high fidelity to support high quality 3D rendering from the user viewpoint. Furthermore, acquisition from a single viewpoint

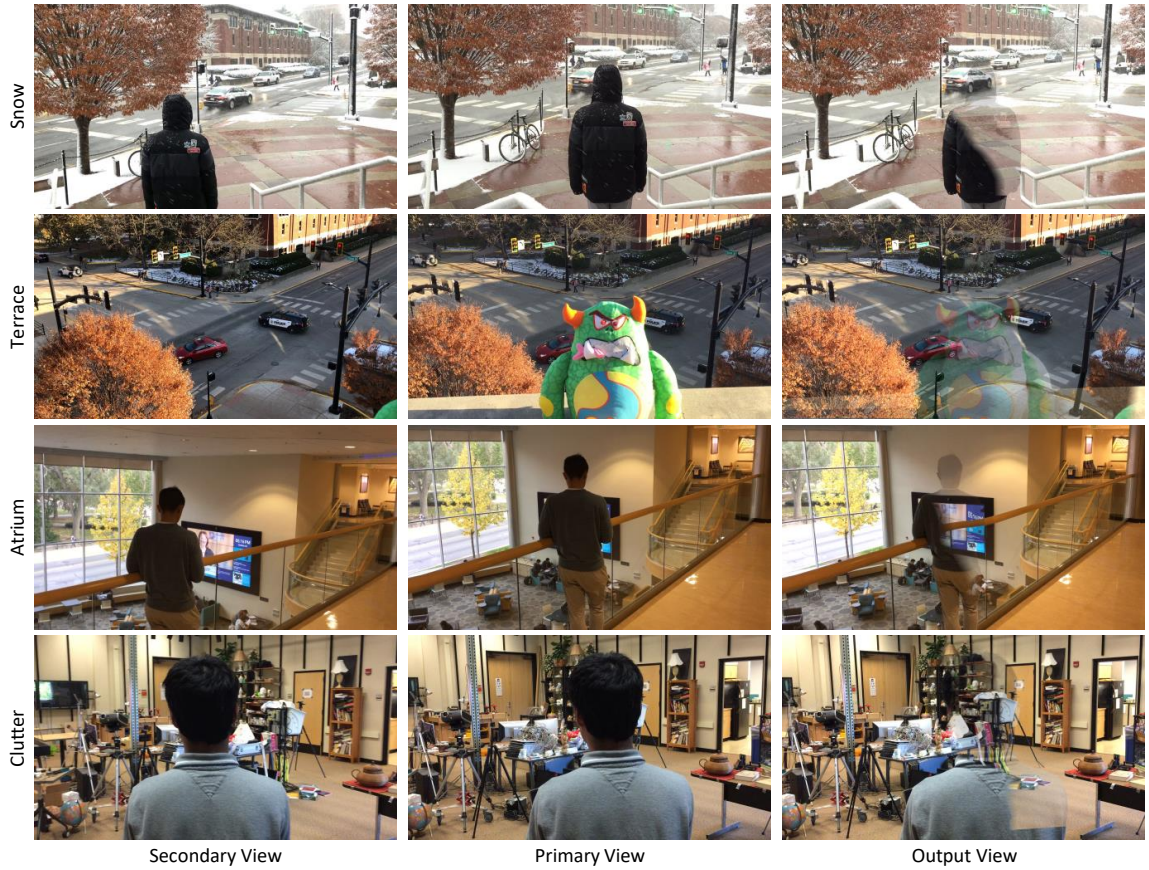


Figure 4.1.: Results of our fast video occlusion removal method. The scene is acquired from the user viewpoint with a primary video and from a translated viewpoint with a secondary video. The output frames are obtained by blending the occluder pixels in the primary frame with background pixels from the secondary frames, achieving good continuity at the occluder contour. Our method supports intricate dynamic scenes at a frame rate between 40 and 60 frames per second.

is not enough since, in the absence of the occluder, the user could see parts of the scene to which there is no line of sight from the single acquisition viewpoint.

In this paper we propose a method to remove an occluder in a primary video acquired from the user viewpoint, using pixels from a secondary video acquired from a translated viewpoint. For each pair of primary and secondary frames, our method replaces the primary frame occluder pixels with pixels from secondary frame. The

secondary frame pixels are integrated seamlessly into the primary frame, with good continuity across the occluder contour. The result is a multiperspective frame, which shows most of the scene from the user viewpoint, except for the parts hidden by the occluder, which are shown from the secondary viewpoint. Due to the visual continuity achieved along the occluder contour, the effect is a good approximation of the transparency effect needed to remove the occluder, which comes without the high cost of color and depth acquisition. As shown in Figure 4.1, our method supports complex, dynamic scenes. Our method removes the occluder from the user video by splicing in a secondary video. Our method is fast, as it bypasses geometry acquisition.

4.2 Prior Work

The removal of objects that hinder the user's view of a region of interest is a typical diminished reality problem [63]. By covering up a real object with the image of the background it occludes, one can make the object virtually invisible by creating a "see-through" effect [4]. Achieving the effect requires the following steps: (1) acquiring the occluder background (2) modifying the acquired background to fit the occluder footprint as seen from the user's viewpoint, and (3) compositing the modified background into the user's view.

In order to remove an object from the user's view, information on the occluded background is required in order to swap the object with its background. The background information can be captured with multiple images acquired by the user in advance [64]. When the hidden background is composed of known hidden objects, such as a specific person's face, a pre-captured dataset with angle-dependent images is sufficient [65]. An internet photo collection can also be used to delete a person in a video sequence, especially when the scene is at a popular sightseeing spot that is frequently photographed [66]. However, due to the large time interval between the pre-acquisition of the images and the occluder removal, there is a sizeable photometric difference between the pre-acquired images and the image to be processed. Another

method is to search for the best matching disoccluded view of the target in an earlier frame. The method was used to disocclude walking human [67]. The same method was used to remove a person by stitching the current frame with the background from a previous frame [68]. The method fails when the current background configuration has not been observed in any of the earlier frames. Both the pre-acquisition and the temporal resampling methods cannot handle highly dynamic scenes, such as a football game.

To deal with dynamic scenes, the scene has to be acquired in parallel, from additional viewpoints. Surveillance cameras have been used to see through walls [69, 70]. Multiple users each with their own hand-held camera can capture the background for each other [71]. A remotely controlled robot equipped with a camera can be deployed to acquire background information [72]. Like these prior methods, we acquire the background information with a secondary camera.

Once the background image is acquired, the image needs to be first transferred to primary camera viewpoint. By assuming the scene only consists of large planes, homography matrices have been used to warp the background to the user view [73]. Homographies have also been used to transfer the best matching background image from an internet image collection to the user view [66]. Another category of methods is to explicitly extract the 3D geometry of the background scene. Using stereo vision, the background has been approximated as a set of small planes [63]; stereo vision has also been used to generate a dense depth map that allows 3D warping the background to the primary viewpoint [74]. The background geometry has also been acquired with the help of RGB-D cameras [75]. These methods are usually computationally heavy, or they require additional hardware such as a depth camera. Reconstructing an accurate model of a complex 3D scene in real time remains an open research question, and inaccurate geometry leads to output image artifacts, such as holes and tears. We bypass 3D geometry acquisition by combining a global alignment based on a rotation-only assumption and a local alignment for residual mapping, to achieve in

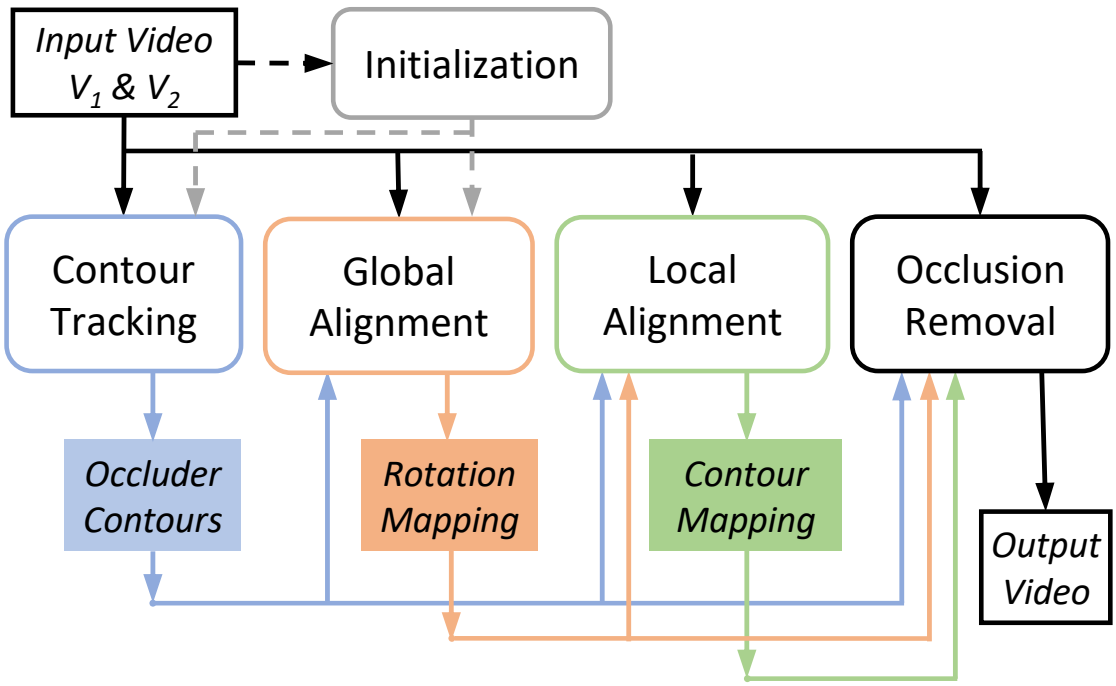


Figure 4.2.: System pipeline.

a computationally efficient way a good estimate of the mapping between the primary and secondary view.

4.3 Approach

We first give an overview of our occlusion removal pipeline, and then we describe the algorithm it implements.

4.3.1 Pipeline Overview

Given a primary input video stream, acquired from the user's viewpoint, and a secondary input video stream, acquired from a translated viewpoint, our method removes an occluder from the primary video using pixels from the secondary video, according to the pipeline shown in Figure 4.2. First, an initialization stage defines the occluder contour in the first frame of each video, and computes an approximate

Algorithm 1 Contour adjustment (also see Figure 4.3)

Input: Image I_1, contour C_1 in I_1, image I_2, contour C_2^* in I_2

Output: Adjusted contour C_2

for each vertex pair (p_2^*, q_2^*) in C_2^* **do**
    $s_{max} = -\infty$
    for each pixel center q in neighborhood S of q_2^* **do**
        $Q_1 = I_1$ patch centered at q_1
        $Q_2 = I_2$ patch centered at q
        $s_q = \text{sim}(Q_1, Q_2) + \lambda \exp(-|q - q_2^*|^2/(2\sigma^2))$
        if $s_q > s_{max}$ **then**
            $q_2 = q$, $s_{max} = s_q$
        end if
    end for
    $p_2 = p_2^* + q_2 - q_2^*$
end for
RemoveSelfIntersections(C_2)

mapping between the two first frames. Then, pairs of primary and secondary video frames are processed in four stages: the contour of the occluder is updated in each of the two frames; an initial mapping between the pair frames is computed as a rotation, by minimizing color differences outside the occluder; the initial mapping is locally refined at the occluder contour to enable splicing in the pixels from the secondary frame with good continuity to the surrounding primary frame pixels; finally, the occluder is removed from the primary frame by looking up its pixels in the secondary frame, using a concatenation of the global and local mappings.

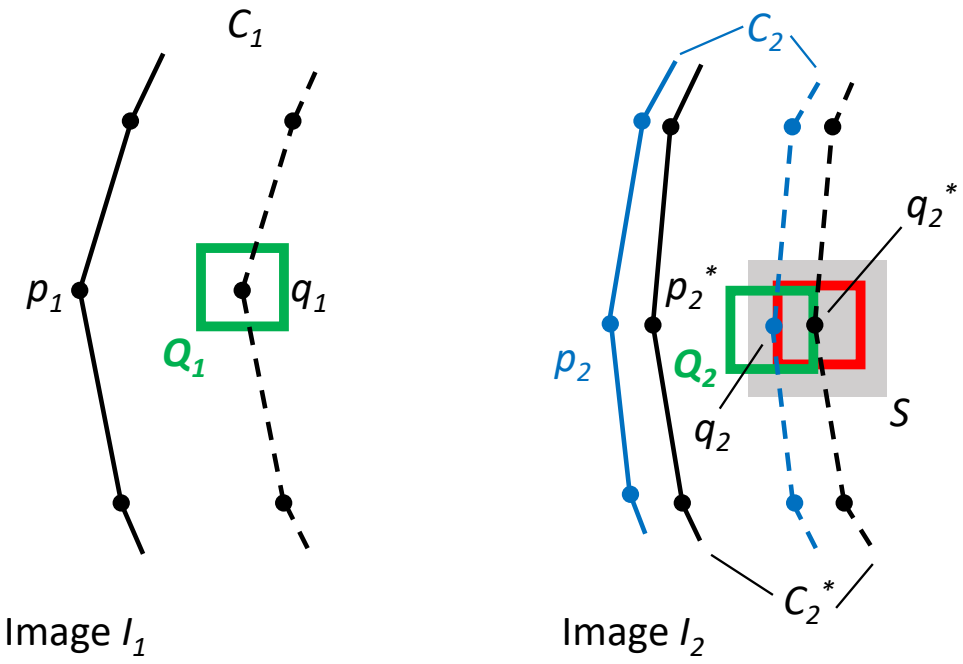


Figure 4.3.: Adjustment of approximate contour C_2^* to C_2 in image I_2, given the corresponding contour C_1 in image I_1 (Algorithm 1). The algorithm searches for a better position for each inner contour vertex q_2^* over its neighborhood S; a good position q_2 yields a high color similarity between I_2 at q_2 and I_1 at q_1; q_1 is the inner contour vertex of C_1 corresponding to q_2^*. Once q_2^* is adjusted, the corresponding outer contour vertex p_2^* is adjusted to p_2 with the same offset.

4.3.2 Contour adjustment algorithm

For the purpose of this paper, a *contour* is a pair of polylines that model the inner and outer boundaries of an object (at least partially) visible in an image. The inner contour is on the object and the outer contour is on the background surrounding the object. The inner and the outer contours have the same number of 2D vertices, the inner contour is inside the outer contour, the segments of a contour do not intersect, each contour has disk topology, and contours do not have to be convex. The outer and inner contours are needed to restrict color comparisons to the occluder object, in

the case when the inner contour is used, or to the background around the occluder object, in the case when the outer contour is used.

Our pipeline relies several times on a contour adjustment algorithm, which we describe first (Algorithm 1). The algorithm takes as input a first image I_1, a known contour C_1 in I_1, a second image I_2, and an estimate C_2^* of C_1 in I_2. The algorithm output is a contour C_2 obtained by adjusting C_2^*. The algorithm adjusts C_2^* one pair of vertices (p_2^*, q_2^*) at the time, where p_2^* and q_2^* are corresponding vertices on the inner and outer contours (line 1). We describe the algorithm for the case when the adjustment proceeds along the inner contour, as shown in Figure 4.3. Adjustment along the outer contour is similar.

The algorithm adjusts the position of q_2^* by searching its neighborhood S for a better location (lines 3-10). For each candidate location q, the algorithm computes color similarity between I_2 at q and I_1 at q_1. Color similarity is evaluated over square image patches Q_1 and Q_2 (lines 4-6). The inner contour vertex q_2 is adjusted every time image similarity improves (lines 7-8). Once the entire neighborhood of q_2^* has been searched, the contour vertex p_2 is adjusted by the same offset $q_2 - q_2^*$ as q_2^* (line 11). Once the inner and outer contours have been adjusted, C_2 is returned after any self intersection is removed (line 13). Our algorithm checks and removes self-intersections by traversing the outer contour; if two outer contour segments (p_2^i, p_2^{i+1}) and (p_2^j, p_2^{j+1}) intersect, where $i < j$, all outer contour vertices from p_2^{i+1} to p_2^j are removed, together with their corresponding inner contour vertices.

In line 6, the similarity between image patches Q_1 and Q_2 is computed differently based on whether images I_1 and I_2 are frames of the same video, i.e. both primary or both secondary, or not, i.e. one primary and one secondary. When I_1 and I_2 are from the same video, we compute similarity using the inverse of the sum of squared per pixel color differences:

$$sim_{intra}(Q_1, Q_2) = -\sum_p (Q_1[p] - Q_2[p])^2. \tag{4.1}$$

When I_1 and I_2 are from different videos, we use a cosine similarity [76], in order to compensate for any large exposure and white balance differences between the two videos. Cosine similarity is computed by treating each patch as a vector, and by computing the cosine of the angle between the two vectors:

$$sim_{inter}(Q_1, Q_2) = \frac{\sum_p Q_1[p] \cdot Q_2[p]}{\sqrt{\sum_p Q_1[p]^2}\sqrt{\sum_p Q_2[p]^2}}. \tag{4.2}$$

In addition to the color similarity value sim, the aggregate similarity score s_q (line 6) also includes a displacement term $\lambda \exp(-|\delta p|^2/(2\sigma^2))$, which favors small contour adjustments when sim values are similar. This displacement term aims to avoid large adjustments for marginal color similarity improvements, as is the case, for example, when a patch capturing an object edge could slide up and down the edge without meaningful color similarity changes.

4.3.3 Main algorithm

Our pipeline implements Algorithm 2. The algorithm takes as input the primary V_1 and secondary V_2 videos and removes a user specified occluder from V_1 using pixels from V_2.

Initialization. The algorithm first performs a once per session initialization (lines 1-5). The user draws in the first frame V_1^0 of the primary video an approximate piecewise linear outer boundary B_1^0 of the occluder to be removed (line 1, red line in Figure 4.4). An approximate boundary that overestimates the occluder is sufficient, as the occluder will be removed with safety margins. Whereas other applications of segmentation have to recover an object contour with high-fidelity, as needed to paste it inconspicuously into a destination image, our application simply has to make sure that the entire occluder is discarded.

The outer boundary B_1^0 is refined to define the initial contour C_1 in frame V_1^0 (line 2), as follows: B_1^0 is rasterized to obtain a pixel mask M_1; M_1 is eroded to pixel

Algorithm 2 Intra-frame video splicing for occlusion removal

Input: Primary video V_1, secondary video V_2

Output: Disoccluded primary video V_d

// Initialization

$B_1^0 = \text{UserInputContour}(V_1^0)$

$C_1 = \text{RefineContour}(B_1^0)$

$C_2^* = \text{HomographyMapping}(C_1, V_1^0, V_2^0)$

$C_2 = \text{AdjustContour}(C_1, V_1^0, C_2^*, V_2^0)$

$R_1^0 = I$; $R_2^0 = \text{InitializeRotation}(C_1, V_1^0, V_2^0)$

for each frame i **do**

// Contour tracking

$C_1 = \text{AdjustContour}(C_1, V_1^{i-1}, C_1, V_1^i)$

$C_2 = \text{AdjustContour}(C_2, V_2^{i-1}, C_2, V_2^i)$

// Global alignment

$j = k \times \lfloor i/k \rfloor$

$R_1^i = R_1^j \times \text{RotationMapping}(C_1, V_1^j, V_1^i, R_1^{i-1})$

$R_2^i = R_2^j \times \text{RotationMapping}(C_2, V_2^j, V_2^i, R_2^{i-1})$

$R = (R_2^i)^{-1} \times R_1^i$

// Local alignment

$A_1 = \text{SalientContourPoints}(C_1)$

$A_2 = \text{AdjustContour2}(A_1, V_1^i, R \times A_1, V_2^i, R)$

// Occlusion removal

$V_d^i = V_1^i$

for each pixel $p \in C_1$ **do**

$p' = \text{LookUp}(p, R, A_1, A_2, C_2)$

$V_d^i[p] = \text{Blend}(V_1^i[p], V_2^i[p'])$

end for

end for

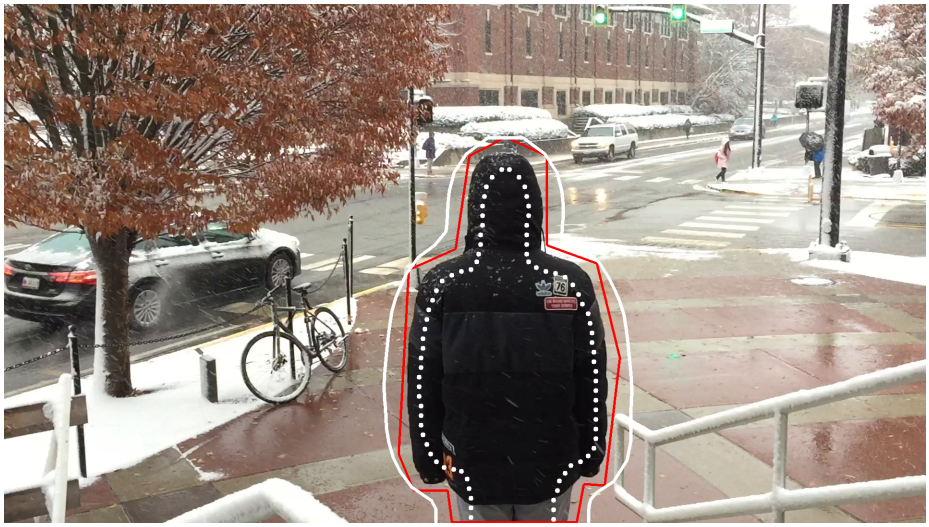


Figure 4.4.: Contour initialization in first frame of the primary video: user drawn outer contour (red), initial inner (white dots) and outer (white line) contours.

mask M_1'; the inner contour of C_1 is defined as a subset of the outer pixels of M_1', i.e. pixels who have at least one of their eight neighbors not part of M_1' (white dots in Figure 4.4); every inner contour vertex is moved outwards along its normal to define its outer contour vertex pair (white line in Figure 4.4).

C_1 is used to initialize the occluder contour C_2 in the secondary video. C_1 is transferred to V_2^0 in two steps.

First, C_1 is taken almost all the way to its correct location in V_2^0 with a homography mapping from the C_1 region of V_1^0 to V_2^0; this provides an estimate C_2^* of the occluder contour in V_2^0 (line 3). The homography assumes that the occluder is a 3D plane, which is imaged by the two cameras with known intrinsic parameters. The homography is computed by detecting SURF features [77] inside the occluder region C_1 in V_1^0, and over the entire frame V_2^0. Each V_1^0 feature is matched to a V_2^0 feature with similar descriptor using FLANN [78]. The homography is determined by minimizing the reprojection error of corresponding features, using a RANSAC approach [79], which provides robustness to outlier feature correspondences. Figure 4.5 shows the outer and inner contours of C_2^* with white solid and dotted lines, respectively; C_2^*

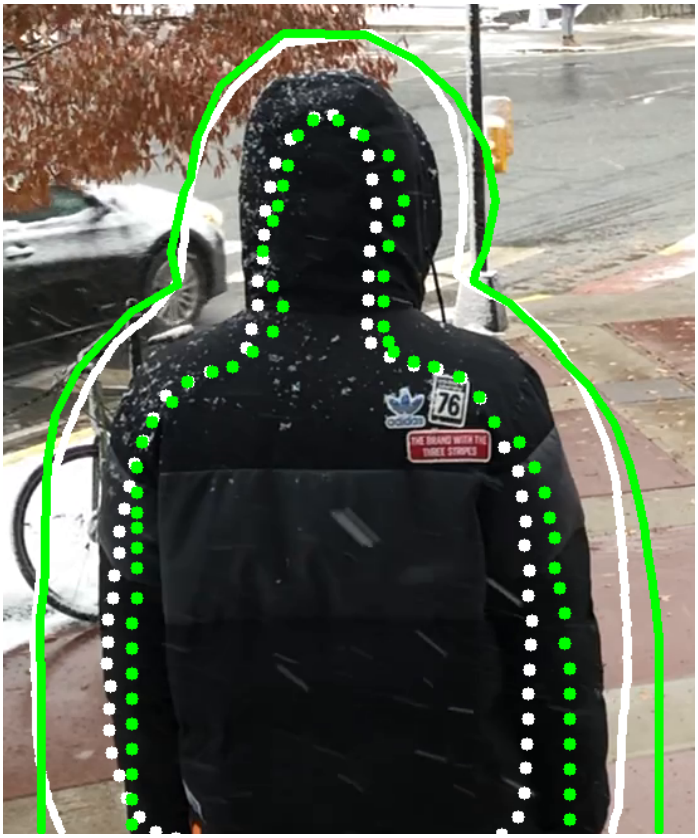


Figure 4.5.: Contour initialization in first frame of the secondary video: initial contour transferred from first frame of primary video with a homography (white), and adjusted contour (green).

does not capture the occluder quite perfectly as the outer contour crosses into, and the inner contour crosses out of the occluder.

Second, C_2^* is adjusted to C_2, using our contour adjustment Algorithm 1 (line 4). Since the frames provided to Algorithm 1 belong to different videos, the cosine similarity metric is used. Figure 4.5 shows the adjusted contour C_2 with green solid and dotted lines.

The algorithm maintains two arrays of 3D rotations, R_1 and R_2, one for each video. R_1^i rotates frame i of the primary video to frame V_1^0. R_2^i rotates frame i of the secondary video to V_1^0 as well, as the first frame of the primary video serves as a common reference. The last step of the initialization sets R_1^0 and R_2^0 (line 5). R_1^0

Figure 4.6.: Initialization of rotation from the first frame of the secondary video V_2^0 (top right) to the first frame of the primary video V_1^0 (top left). The rotation is visualized by averaging V_1^0 with the rotated V_2^0 (bottom). The rotation is recovered robustly, as indicated by the alignment of the distant parts of the scene, despite the considerable disparity between the two frames, indicated by the ghosting on the near parts of the scene.

is the identity matrix. R_2^0 is computed by minimizing feature reprojection error, as described in Section 4.3.4. Figure 4.6 illustrates R_2^0 by blending the rotated V_2^0 on top of V_1^0.

After initialization, each pair of primary and secondary video frames is processed with the four main stages of our pipeline.

Figure 4.7.: Contour tracking: old contour (blue) is adjusted to the occluder (red). The algorithm adjusts the contour from the previous frame $i-1$; for illustration clarity, the blue contour shown here is from an older frame, $i-5$.

Contour tracking. Contours C_1 and C_2 are updated in the current frames V_1^i and V_2^i, using the known contours in the previous frames V_1^{i-1} and V_2^{i-1} (lines 7-8). We use Algorithm 1 again; the frame with the known contour is the previous frame, the frame where to adjust the contour is the current frame, and the estimate of the contour in the current frame is given by the contour in the previous frame. The no-motion contour prediction is sufficient because of the high frame rate of the videos compared to camera and occluder motion and velocity. The frames are part of the same video, so similarity is computed using color difference. Figure 4.7 illustrates the result of our contour tracking stage that snaps the contour (blue) into place (red).

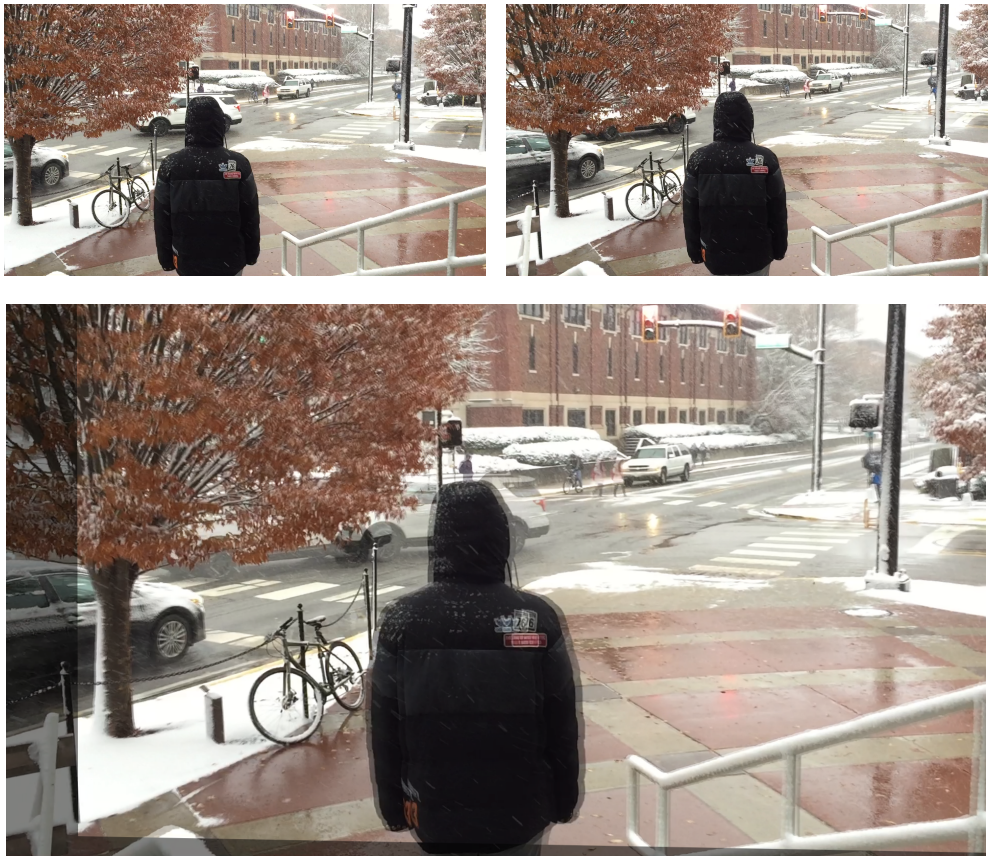


Figure 4.8.: Global alignment of two frames of the primary video (top). The frames differ in view direction, see different relative location of light post at right of image, and in time, see moving car turning in intersection. The blended visualization (bottom) reveals that the global alignment recovers the accurate rotation between the two camera poses, as indicated by the good alignment of the distant stationary parts of the scene; the alignment is robust to the motion in the scene (i.e. moving car), and to the disparity between the frames induced by objects near the camera, such as the person and the handrail.

Global alignment. The algorithm has to compute a mapping from the primary video frame V_1^i to the secondary video frame V_2^i. For this, the algorithm first computes an approximate mapping. The approximate mapping is found by computing, for each video, the rotation of the current frame i to an earlier frame j *of that same video* (lines

9-11). Once the two rotations R_1^i and R_2^i are known, the approximate mapping R from V_1^i to V_2^i is easily obtained by leveraging the common reference V_1^0 of all rotations (line 12).

To find the rotation of the current frame i with respect to its earlier frames, we use a more distant key frame j, and not the previous frame $i-1$, as consecutive frames would be too similar, and the alignment would drift. The key frames are spaced k frames apart. Unlike for initial rotation computation (Section 4.3.4), where the rotation had to connect two frames acquired from a different viewpoint, by two different cameras, here the rotation only has to connect two video frames acquired by the same camera, from a similar viewpoint. Consequently, the global alignment can be computed by directly minimizing color difference between the two frames i and j, which bypasses the slower feature detection and matching. However, global alignment has to avoid the inconsistencies introduced by parts of the scene near the camera, which create frame disparity even for small camera translations, and by parts of the scene that move, which appear at different locations in the two frames. Our global alignment computation is described in Section 4.3.5. Figure 4.8 illustrates the accuracy and robustness of our global alignment stage.

Local alignment. The mapping R between frames V_1^i and V_2^i will be used to replace the occluder pixels in frame V_1^i with pixels from V_2^i. The mapping is approximate when the occluded scene is near and it has to be refined. The inaccuracy of the mapping is noticeable only at the occluder contour C_1, where the V_2^i pixels are spliced into V_1^i (Figure 4.9, left). The algorithm computes a local alignment that alleviates color differences on each side of the occluder contour (lines 13-14). First, the outer contour of C_1 is sampled to gather a set of points A_1 with large color changes (line 13, and red dots in Figure 4.9, middle). These points are better suited for computing the local alignment than the outer contour vertices because they do not sample wastefully regions of uniform color, and because they sample most regions with large color changes.

Figure 4.9.: Local mapping need (left), implementation (middle), and result (right). Left: disoccluding using only the global mapping results in discontinuities where near objects cross the occluder contour, e.g. where the sidewalk and handrail cross the red line in the left image. Middle: the local mapping connects primary frame salient contour points (red points) to their correspondence in the secondary frame (green points); the local alignment offset is larger for near objects. Right: disocclusion with continuity at occluder contour.

The newly defined outer contour A_1 is adjusted with an algorithm similar to Algorithm 1, with two differences. The first difference is that the adjustment now proceeds following the outer contour, and not the inner one. Using Figure 4.3 again, adjustment based on the outer contour is not concerned with the outer contour vertices and directly moves p_2' to its better position p_2 that minimizes the color difference between I_2 at p_2 and I_1 at p_1. The second difference is that the adjustment now compares Q_1 to a rotated image patch Q_2, and not an axis aligned one (line 6 in Algorithm 1). The rotated Q_2 is computed using rotation R. This more accurate comparison is now needed because the contour adjustment for the local alignment crosses between videos, and axis aligned patches do not match. Furthermore, adjustment is performed at the output frame cut line between the two video sources, so an inaccurate alignment would be readily visible. Figure 4.9, middle, visualizes the displacement of the points of A_1 (red dots) to their correct locations A_2 (green dots). Figure 4.9, right,

shows the continuity achieved at contour boundary in the disoccluded frame using our local alignment.

Occlusion removal. Finally, the algorithm removes the occluder in the primary frame V_1^i (lines 15-19). The disoccluded frame V_d^i starts out as a copy of V_1^i (line 15), and then pixels p inside the contour are looked up in V_2^i. A pixel p is first rotated to p_r using R, and then p_r is offset with a weighted sum of offsets $a_2 - R \times a_1$, for all a_1 points in the vicinity of p. We support several disocclusion visualization modes, such as cutaway, where p' completely replaces p (Figure 4.9, right), transparency, where p and p' are blended together (Figure 4.1), with and without showing the contour of the occluder. Our disocclusion visualization supports transitioning gradually from the background of the primary frame to the occluder shadow (Figure 4.11), and from the occluder shadow to the residual occluder (third column in Figure 4.1).

4.3.4 Rotation Initialization

The videos V_1 and V_2 are acquired from different viewpoints, so computing the rotation R_2^0 of frame V_2^0 to V_1^0 is challenging, as it does not benefit from frame to frame coherence. Indeed, the gap between V_1 and V_2 only has to be bridged for the first frame of V_2, as subsequent V_2^0 frames only have to be registered to their previous frame, whose rotation to V_1^0 is already known.

R_2^0 is computed by finding SURF features [77] in V_1^0, outside of C_1, and in V_2^0. V_1^0 features are matched to V_2^0 features using FLANN [78]. A pair of corresponding features is given a weight commensurate to the confidence in its correctness. The weight w_{ij} of a correspondence between a feature f_{1i} in frame V_1^0 and the most similar feature f_{2j} in frame V_2^0 is computed with:

$$w_{ij} = |f_{1i} - f_{2k}|/|f_{1i} - f_{2j}| \tag{4.3}$$

where f_{2k} is the feature second most similar to f_{1i}, and $|f_a - f_b|$ is the difference between the descriptors of two features f_a and f_b. The smallest possible weight is 1, when f_{1i} is equally similar to its best two matches, indicating the possibility of an ambiguous correspondence. When the second most similar feature f_{2k} is considerably less similar to f_{1i} than f_{2j} is, the correspondence is less likely to be incorrect, hence the larger weight. The reprojection error of corresponding features is minimized using a Gauss-Newton non-linear optimization [80], while also leveraging a RANSAC [79] approach to mitigate possible incorrect correspondences. The selection of the best rotation out of the multiple RANSAC tries is not done by merely choosing the try with the highest number of inlier correspondences. Instead, we choose the try with the highest sum of inlier correspondence weights.

4.3.5 Global Alignment

The global alignment computes the rotation of the current frame i to a previous key frame j, independently, for each of the two videos. We globally align two frames with a rotation because it provides a good approximation of the mapping between the frames without the prerequisite of scene geometry. We use the Gauss-Newton method [80] to find the three rotational degrees of freedom that minimize color difference.

Given a current frame V^i, a key frame V^j, the camera intrinsic matrix M, and a candidate rotation R from V^i to V^i, the color residual r_p at pixel p is given by:

$$r_p(R) = V^j[p] - V^i[M^{-1}(RM) \cdot p] \tag{4.4}$$

In Equation (4.4), p is first unprojected from V_i, then rotated, and then projected to V_j. The stacked color residual vector over the entire frame is given by $\vec{r} = (r_1, \ldots, r_n)^T$, and the color error $E(R)$ is the L_2 norm $|\vec{r}|$ of the residual vector. We use a left-compositional formulation. Starting with an initial estimate R^* given by the rotation of the previous frame V_{i-1} to V_j, we compute an increment δR for each iteration:

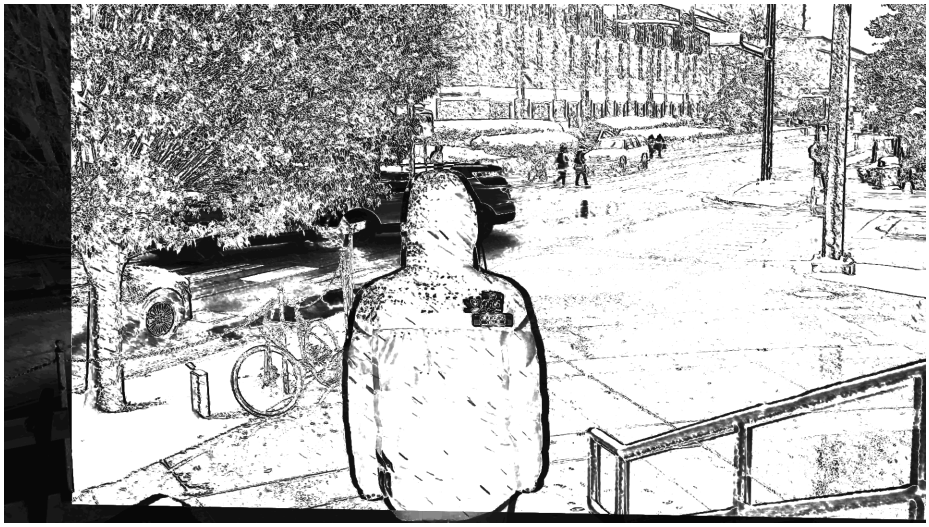


Figure 4.10.: Weights used in global alignment from Figure 4.8. Moving objects, such as the car and the pedestrians, and regions with high disparity, such as the contour of the person near to the camera, are assigned low weights, to reduce noise in the rotation computation.

$$\delta R = -(J^T J)^{-1} J^T \vec{r}(R), \text{ where } J = \frac{\partial \vec{r}(\epsilon \oplus R)}{\partial \epsilon}|_{\epsilon=0} \tag{4.5}$$

J is the derivative of the residual vector $\vec{r}$ with respect to an increment ϵ, and $J^T J$ is the Gauss-Newton approximation of the Hessian matrix of E. We then update the current rotation estimate by multiplying it with the iteration's increment:

$$R = \delta R \oplus R \tag{4.6}$$

In order to gain robustness with outliers caused by moving objects, by the disparity of near objects, and by view dependent effects (e.g. reflections), the minimization is done in an iteratively reweighted fashion [81]. The weight of a pixel p equals the inverse $1/r_p$ of its residual. The weight is capped to avoid infinite weights when a pixel residual is very small. Figure 4.10 visualizes the pixel weights for the global alignment from Figure 4.8). The weighted rotation increment is given by

Figure 4.11.: Abrupt (left) and progressive (right) transition from background to occluder shadow.

$$\delta R = -(J^T W J)^{-1} J^T W \vec{r}(R), \text{ where } W = \text{diag}(1/r_1, \ldots, 1/r_n) \tag{4.7}$$

For speed, we perform this color residual minimization with a coarse-to-fine approach, that works at different levels of the image resolution pyramid. We start from the coarsest level of 30×17, as our frames have a 16:9 aspect ratio, and we stop at for levels deeper, i.e. at 480×270 . The minimization converges at each level in between 2 and 4 iterations.

4.4 Results and Discussion

We have tested our occlusion removal method on several scenes, including the *Snow*, *Terrace*, *Atrium*, and *Clutter* scenes shown in Figure 4.1, and the *Crossing* scene shown in Figure 4.11. All scenes were abundantly dynamic, except for the *Clutter* scene, which was stationary. Each scene was acquired with two videos, captured with separate handheld phone and tripod mounted tablet cameras, from differ-

Table 4.1.: Average running times [ms] for the stages of our pipeline, and overall frame rate [fps].

Stage	Global alignment	Contour tracking	Local alignment	Occlusion removal	FPS
Clutter	2.9	3.8	8.8	5.2	48.3
Atrium	2.9	3.2	8.2	3.5	56.0
Crossing	2.9	2.6	10.9	2.5	52.8
Terrace	2.9	4.1	9.8	6.7	42.7
Snow	2.9	2.5	9.0	2.7	58.5

ent viewpoints, matching the scenario described in the paper; the *Clutter* scene was acquired with a single handheld camera that revolved around the occluder, and the later frames were used to disocclude the earlier frames. Our method worked well with all scenes, alleviating occlusions by creating a convincing transparency effect.

4.4.1 Time

We ran our disocclusion method for each pair of videos on an Intel E5-1620 workstation with a 3.5 GHz CPU clock. Our implementation only uses the CPU of the workstation, and not the GPU, and it is strictly serial. The videos were played back at the original 30 Hz frame rate, and our method was fast enough to comfortably process the frames in real time, with no precomputation.

Table 4.1 gives the average times for each of the four stages of our pipeline, as well as the average frame rate, which is at least 40 fps. Global alignment performance depends on the number of pixels in the resolution pyramid level used, contour tracking performance depends on the number of contour vertices, local alignment performance depends on the number of salient contour points, and occlusion removal depends on the occluder footprint. The slowest stage is the local alignment stage, which evaluates

color differences with rotated and not axis aligned patches (parameter R of line 14 in Algorithm 2). In addition to the cost of the rotation itself, comparing color between a rotated patch and an axis aligned patch introduces a bilinear interpolation per color comparison. There is very little frame rate variability as the workload is nearly constant from frame to frame.

4.4.2 Quality

Our method handles well a variety of scenes, replacing the occluder pixels with pixels from the secondary video, with good continuity. The limitations of our method are discussed in the next section. Our method relies on a weak connection between the primary and the secondary frames: the frames are connected by an approximate mapping inside the occluder contour, and by a more rigorous mapping along the occluder contour. The weaker connection is faster to compute than the per-pixel correspondences used in structure from motion. Moreover, the weaker connection has the advantage of avoiding disocclusion errors.

Comparison to depth acquisition. Even if both videos are replaced with perfect RGBD streams, disocclusion errors can occur when the occluder is removed and the primary viewpoint gains line of sight to a part of the scene not visible from the secondary viewpoint. In Figure 4.12, the primary viewpoint is O_1 and the secondary viewpoint is O_2. The secondary frame samples the green object from the left until B, and then the blue object from C towards the right. The primary frame is affected is affected by the occluder FG. The primary frame sees the green object from the left until A, then the occluder, and then the blue object from E to the right. Our occluder removal method replaces in the primary frame the occluder pixels FG with the secondary frame pixels from A to E. Our local alignment makes sure that the primary and secondary frames are aligned at A and E. A 3D occluder removal method leverages the perfect depth available at each secondary frame pixel to project the secondary frame pixels to their correct location in the primary frame. However,

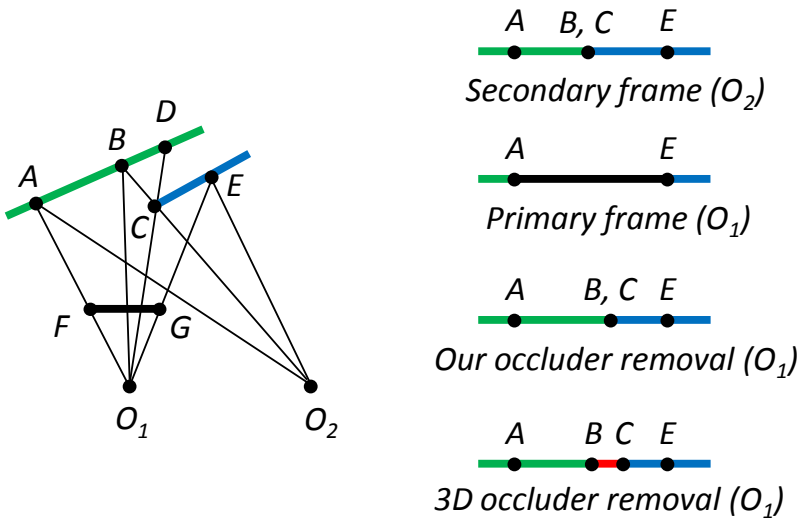


Figure 4.12.: Disocclusion error caused by 3D occluder removal. The secondary frame with viewpoint O_2 does not capture the green object between B and D. Even if the secondary frame has perfect depth per pixel, projecting the 3D samples of the secondary frame onto the primary frame will leave a gap between the projection of B and the projection of C. Our occluder removal method does not suffer from such a disocclusion error, as the mapping it uses does not allow B and C to separate in the primary frame.

since the secondary frame does not sample the green object between B and D, the 3D occluder removal method leaves a gap, i.e. a disocclusion error, between B and C.

Comparison to ground truth. We have also compared our method to a ground truth transparency effect. For this, we have recorded primary and secondary video feeds with the occluder obstructing both views (Figure 4.13a), we extracted the occluder from both views (Figure 4.13b), we recorded primary and secondary video feeds without the occluder (Figure 4.13c), we inserted the extracted occluder in each video feed (Figure 4.13d), and we ran our algorithm on the two video feeds with the inserted occluder. Our algorithm produces results Figure 4.12f that are substantially

(a)

(b)

(c)

(d)

(e)

(f)

Figure 4.12.: Comparison of our disocclusion method to a ground truth transparency effect: (a) primary and secondary frames with occluder, (b) extracted occluder, (c) primary and secondary frames without occluder, (d) extracted occluder b inserted into frames c, (e) ground truth transparency effect, (f) output of our algorithm.

similar to the ground truth transparency effect (Figure 4.12e), which was acquired by the primary view camera without the occluder being present. A sliver of the occluder

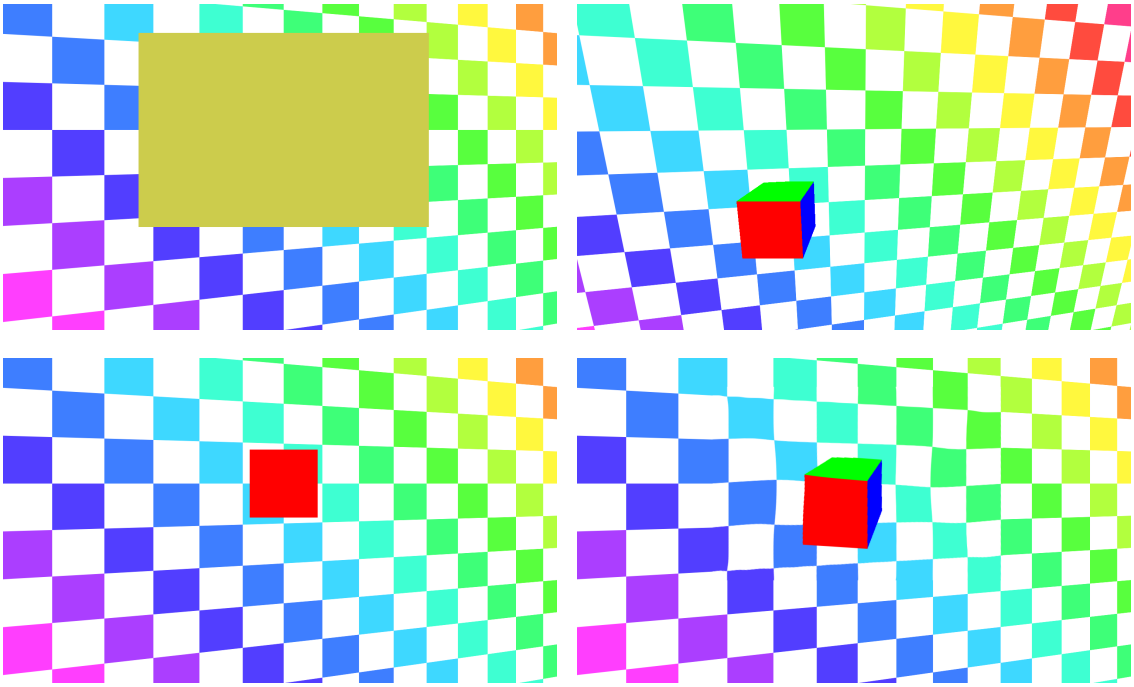


Figure 4.13.: Illustration of multiperspective effect achieved by our disocclusion method: primary and secondary view frames (top), ground truth transparency effect (bottom left), and output of our algorithm (bottom right).

remains in our output since the secondary view direction is tilted up and it does not cover that part of the occluder.

Multiperspective occlusion removal. As discussed above, our method does not recreate the primary view for the disoccluded part of the scene. Instead, the pixels used to fill in the occluder shadow in the primary view come from the secondary view which has a different viewpoint, i.e. a different perspective on the disoccluded scene. The different perspective on the disoccluded scene is maintained, since the global alignment of our method is a rotation and not a 3D warp, i.e. it does not alleviate the viewpoint difference, and since the local alignment of our method alleviates the viewpoint difference only at the boundary between the disoccluded region and the background.

Figure 4.13 illustrates on a synthetic scene the multiperspective nature of the disocclusion effect achieved by our method. The top images show two frames from the primary and secondary views. In the primary view, the yellow rectangle occludes a box with red, green, and blue (RGB) faces. In the secondary view, the RGB box is seen from a translated viewpoint, which reveals the blue and green faces. The bottom images compare the output of our algorithm to the ground truth transparency effect. Whereas the ground truth transparency effect only shows the red front face of the RGB box, our visualization shows the RGB box from the secondary perspective, revealing the the red, green, and blue faces of the box. Our visualization changes perspective continuously based on the local alignment step which splices in the pixels from the secondary view.

4.5 Conclusions. Limitations. Future Work

We have presented a method for removing an occluder from a video, by transferring pixels from a second video that captures what the first video should show if the occluder were not present. The method is fast, with a minimum frame rate of 40 fps, and it achieves good results on a variety of scenes with intricate and dynamic geometry. The pixels from the second video are spliced in with good continuity across the occluder contour. The method is based on the insight that a convincing transparency effect can be obtained without knowledge of 3D scene geometry. The method computes an approximate mapping from the first video to the second video. The approximate mapping orients the second camera the same way as the first camera, but it does not attempt to translate the second camera viewpoint to the first viewpoint. The approximate mapping is sufficient to fill in the occluder footprint with plausible pixels from the second video. The result is a multiperspective visualization, where the scene surrounding the occluder is shown from the first viewpoint, and the scene behind the occluder is shown from the second camera viewpoint. Switching abruptly from one perspective to the other at the occluder contour would create a discontinuity.

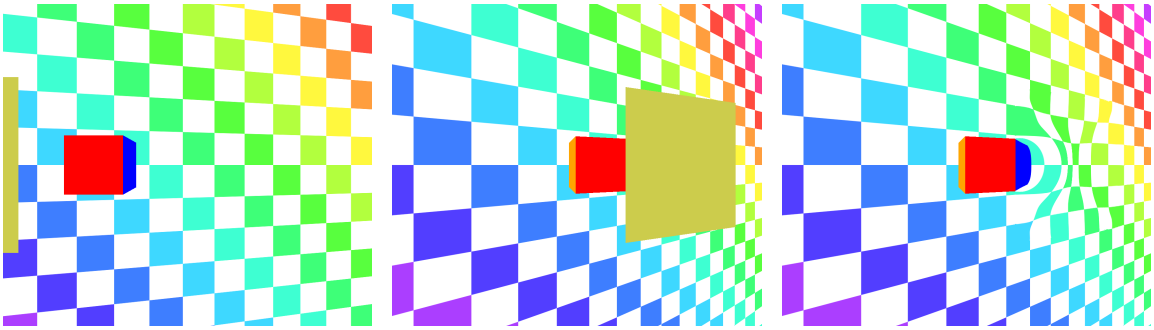


Figure 4.14.: Illustration of multiperspective effect achieved by our disocclusion method: secondary view frame (left) primary view frame (middle) and output of our algorithm (right). Both left and right face of the box is visible in our output.

Instead, our method connects the two perspectives seamlessly with a local mapping that achieves a gradual transition from one viewpoint to the other.

One limitation of our method pertains to near objects that cross the occluder contour. A near object is imaged from different directions by the two cameras, and therefore it has a different appearance in the two frames, a difference that cannot be alleviated by the global mapping rotation. This poses no problems when the near object is completely hidden behind the occluder. However, when the object crosses the occluder contour, the local mapping helps switch from one perspective to to the other continuously, but the object appears distorted as it starts out in one perspective and ends in the other, the same way Picasso's cubism portraits distort the subject, blending in a single image views perpendicular to each side of the face. In Figure 4.14, the switch from the primary to the secondary perspective occurs over the RGB box, with the resulting visualization integrating both perspectives of the box. In Figure 4.15, the handrail crosses the occluder contour in region A, where it switches from the primary perspective, outside the occluder, to the secondary perspective, inside the occluder. The switch is continuous, but the handrail is distorted as it is shown with two perspectives.

Figure 4.15.: Method limitation due to near object crossing the occluder contour: perspective switch deformation (A) and extrapolation discontinuity (B).

Another problem posed by near objects that cross the occluder contour arises when the secondary frame does not see everything the occluder hides in the primary frame. In such a case, a piece of the object is missing from both frames, and the local mapping cannot fill in the missing piece. In Figure 4.15, the visualization appears discontinuous to a a human observer in region B, who knows that the scene has one straight, uninterrupted handrail, and therefore expects that the disoccluded hand rail be aligned with the handrail reemerging to the left of the occluder. We call this an extrapolation discontinuity.

Both the perspective switch deformation and the extrapolation discontinuity problems discussed above are inherent to our method, in the sense that they occur even though our algorithms work as intended. The *Atrium* scene is a worst case scenario for these problems as the long, straight handrail makes them conspicuous. future work could aim to reduce the perspective switch deformation by widening the area over which the switch between perspectives occurs; future work could also aim to reduce the extrapolation discontinuity, by leveraging or even pursuing a high-level

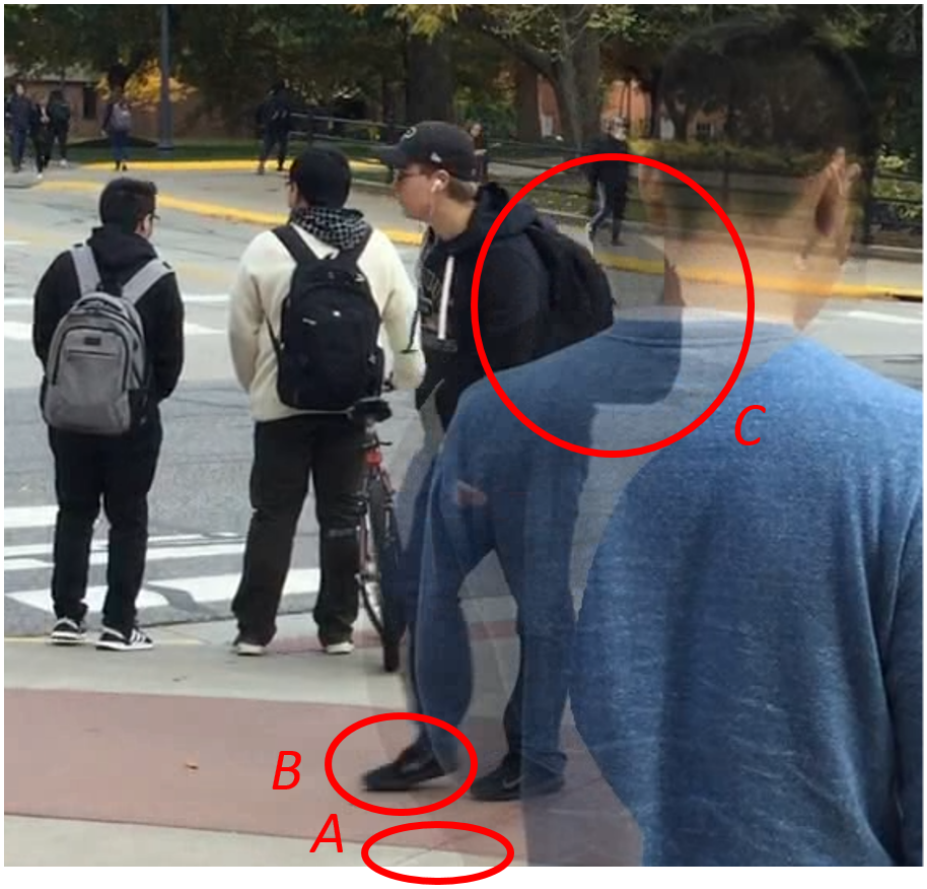


Figure 4.16.: Method limitation due to near object crossing the occluder contour: the local mapping achieves continuity across the occluder contour for the pavement line (A), for the moving foot (B), but fails for the backpack (C)

understanding of the scene that connects the two parts of the handrail even though the occlusion shadows of the two frames have a non-zero intersection.

A third problem posed by near objects that cross the occluder contour is that the local mapping fails occasionally (Figure 4.16). For near objects, correcting the global mapping requires large offsets, which requires a large search neighborhood in the local adjustment computation, which is time consuming. The problem is exacerbated when the object moves quickly, and when the object does not have much texture, as is the case of the backpack and jacket in region C of Figure 4.16. Using a small search neighborhood in the interest of performance reduces local mapping robustness. Future

work could examine increasing the robustness of the local mapping computation with a strategy that leverages the image resolution pyramid to search over large distances to gain robustness without a significant performance trade-off.

Another limitation of the current implementation is that the visualization is not always perfectly stable. Presently, the set of salient points used by the local mapping is computed from scratch for every frame. Future implementations could limit the number of points replaced at every frame, in the interest of stability. Finally, the current implementation computes the global alignment with respect to a fairly recent key frame of the same video, and global alignment is computed across videos only once, for the first pair of frames. This works well for our sequences of 30 s, but for longer sequences, global alignment drift could be a concern, which will have to be addressed by occasionally recomputing the global alignment between the current frames of the two videos.

We have shown that our method runs fast enough on a workstation, using only its CPU, to keep up with prerecorded videos. Future work should deploy our pipeline to portable computers, such as phones or tablets, leveraging their GPUs to process the videos in real time, as they are being acquired. Future work could focus on absorbing into the local adjustment algorithm the latency of transmitting the secondary video to the user device where the disocclusion effect is computed. Another possible direction of future work is to increase the number of secondary video streams to handle complex occlusions.

Our work describes a multiperspective framework for the continuous and non-redundant integration of multiple images, which, compared to traditional structure from motion, comes at the lower cost of having to establish only $O(w)$ and not $O(wh)$ correspondences between pairs of images of $w \times h$ resolution. This framework might find other applications, in augmented reality and beyond.

5 SUBPIXEL CATADIOPTRIC MODELING OF HIGH RESOLUTION CORNEAL REFLECTIONS

5.1 Introduction

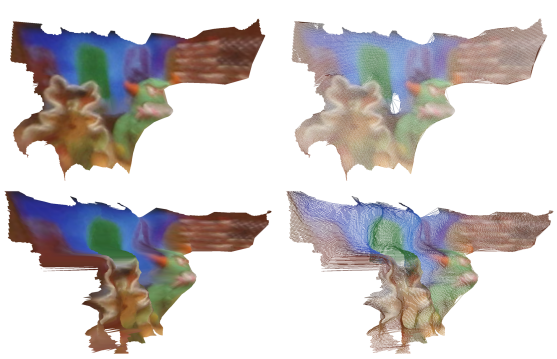
Digital cameras now capture images with a resolution that far exceeds conventional displays. Whereas a display cannot show simultaneously all the pixels of the image, the underlying high resolution is useful for digital zoom-in operations or for large format printing. Another important benefit of high resolution is increasing the quality of 3D scene reconstructions derived from images.

Many real world scenes contain reflective objects, and high resolution images capture a wealth of scene information in fortuitous reflections. Reflections on convex surfaces are particularly rich in information, as the divergent reflected rays sample the scene comprehensively, with a large field of view. Furthermore, reflections introduce additional sampling viewpoints, which allow measuring disparity and triangulating 3D positions from a single image.

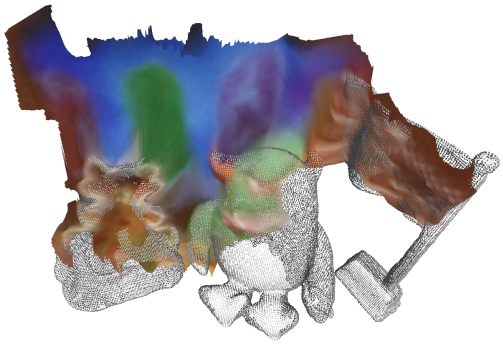
The human eyes are convex reflectors, and researchers have long speculated on the possibility of using corneal reflections to infer 3D scene structure. One challenge is the small baseline, i.e. a typical interpupillary distance is 63 mm [82], which translates to low depth accuracy at distances of 0.5 m and beyond. Another challenge is the low resolution of the corneal reflections. Both challenges are alleviated by increases in the overall image resolution. A third challenge is accurate calibration of the catadioptric system defined by the two eyes and a camera. An accurate catadioptric model is needed to limit the search for correspondences between corneal reflections to 1D epipolar curves, and for accurate triangulation of 3D scene points.

In this chapter we present a procedure for calibrating the catadioptric model defined by two corneal spheres and a camera. The input is a high resolution im-

(a) Input image cropped to eye region.



(b) 3D reconstruction visualized in filled and wireframe mode.



(c) 3D reconstruction (shaded) aligned with truth geometry (grey).

Figure 5.1.: 3D scene reconstruction with our catadioptric modeling approach.

age of a person looking at a 3D scene. In our experiments, the image resolution is $5,472 \times 3,648$ and each corneal reflection has a resolution of approximately 600×600. First, a preliminary corneal catadioptric model is inferred from the projection of the limbus circles in the corneal reflections. Then, the model is refined iteratively using a custom RANSAC approach that relies on bundle adjustment to minimize feature reprojection error. We obtain an error between 0.16 and 0.58 pixels. We use the corneal catadioptric model to recover dense depth through stereo matching with the support of epipolar-like constraints (Figure 5.1). The truth geometry used for comparison (grey points in Figure 5.1c) was obtained by scanning the toys with an active depth sensing camera.

5.2 Prior Work

We first give an overview of prior efforts on acquiring scenes using catadioptric imaging systems, and then we review prior work in modeling the catadioptric imaging system defined by a camera and the two human eyes.

Researchers have long noticed the benefits of devising acquisition systems that combine refractive and reflective elements. One such benefit is an increased field of view. Debevec used a chrome ball as a light probe to capture the complex illumination of a real world scene with a single shot, and to apply it to synthetic objects integrated into the scene [83]. Nayar has developed omnidirectional cameras using paraboloidal mirrors with a single viewpoint, so their images can be resampled to conventional images [84]. A second scene acquisition benefit of catadioptric systems is the ability to integrate multiple perspectives in the same image. The additional perspectives encode depth disparity, which enables single-shot depth from stereo [85]. The additional perspectives are also useful for devising acquisition systems that are robust to occlusions, by guiding the scanning laser beam towards hard-to-reach places [86].

Human eyes are often captured in images, and leveraging corneal reflections to infer information about the scene is appealing and has been carefully studied [87]. The corneal reflections are readily available, without the challenge of augmenting the camera with reflective elements. Furthermore, the corneal reflections introduce additional viewpoints that capture parts of the scene missed from the camera viewpoint. The additional viewpoints not only provide a comprehensive image of the scene, but also allow measuring disparity to extract depth. The catadioptric system defined by a camera and two eyes requires modeling the cornea's reflective surface. Prior work models this surface as a sphere cap, which is part of the corneal sphere, and delimited by the sclera sphere [88]. We use the same cornea surface model. Another challenge is that, unlike for catadioptric imaging devices where the reflective elements have a fixed, pre-calibrated position and orientation with respect to the camera, in the case

of corneal reflections the eyes are free to move with respect to the camera, and their position has to be recovered in every image.

One use of corneal reflections is to capture a panoramic image of the scene, leveraging the large field of view sampled by the reflected rays [88]. The information in the corneal reflection can be used to extract gaze direction in camera-display systems [89], and also to reconstruct a super resolution image of the environment reflected in the user's eyes [90]. Corneal reflections have also been proposed as a way of gaining insight into a crime scene, demonstrating that camera resolution is now sufficient for identifying humans present in such reflections [91].

We discuss in detail the two prior art papers most relevant to our work. One describes a system that does not recover 3D scene structure from corneal reflections, but rather from parabolic metal mirrors [92]. Metal mirrors greatly simplify catadioptric scene reconstruction by providing a precisely known reflective surface shape, and by generating clear and high contrast reflections. Furthermore, metal mirrors are perfectly stationary which avoids the blurriness that results from the slight user head motion as the picture is taken. Moreover, the metal mirrors used are about three times larger than the corneal sphere, and about four times larger than the limbus circle, which delimits the reflection in our case. Consequently, the prior work reflections have a resolution of 2M pixels, compared to the 0.1M pixels for our work, which aids significantly with reconstruction quality. The earlier system refines calibration without a preliminary RANSAC step to weed out mismatched features. The reprojection error achieved by the earlier system is about five times larger than ours, most likely due to the simpler calibration refinement step, as discussed above. Finally, the earlier work does not report any quantitative measure of the 3D reconstruction error. Our work validates the 3D reconstruction quality in an absolute sense by reconstructing objects of known size, as discussed in the following sections.

The other paper highly relevant to our work is the only prior art paper that actually recovers any 3D structure from corneal reflections [93]. The paper proposes the idea of finding correspondences between a pair of corneal reflections and of tri-

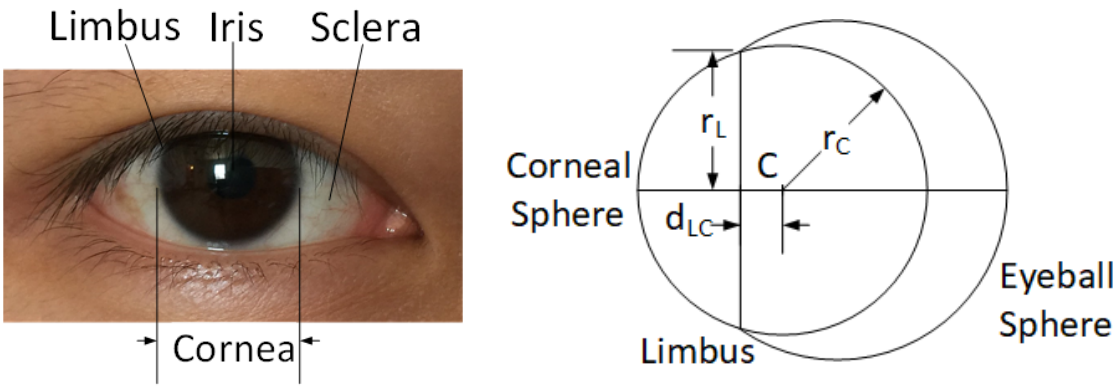
angulating them into depth. We extend this work in the following ways. First, the earlier system calibration stops at our precalibration phase. The earlier system is crudely calibrated by inferring the position of the corneal spheres from the limbus circles, whereas our system refines this initial calibration with our custom RANSAC and bundle adjustment approach, which reduces the reprojection error substantially. We achieve sub-pixel accuracy, whereas the previous paper doesn't report calibration accuracy, which we estimate as being orders of magnitude lower based on the accuracy achieved by our similar precalibration stage. Second, the earlier system requires establishing correspondences between the two corneal reflections manually, by clicking corresponding points. Our system detects, matches, and validates correspondences automatically. Third, the earlier system does not perform dense stereo reconstruction, whereas our system does. Finally, the only scene where 3D reconstruction is demonstrated is that of a large cube with uniformly colored faces. Inspired by their pioneering work, with the help of our subpixel catadioptric modeling framework, we demonstrate 3D scene structure recovery from corneal reflections.

5.3 Catadioptric Model of Corneal Reflections

Many scenes of interest to computer vision applications contain humans, and corneal reflections present the opportunity for catadioptric stereo scene reconstruction. Before scene reconstruction can begin, one has to model the catadioptric system defined by two eyes and a camera.

5.3.1 Eye Model

Figure 5.2a shows an outer view of the human eye. The most distinctive components are the color-textured iris and the surrounding white sclera. The cornea is the transparent outer layer of the eye that covers the iris. The cornea has an internal pressure higher than that of the atmosphere, which maintains the cornea's convex



(a) Outer view. (b) Geometric model.

Figure 5.2.: Eye model.

shape. The cornea surface is coated with a thin film of tear fluid which makes it smooth, with mirror-like reflective characteristics [94].

Geometrically, the eye is well approximated by two intersecting spherical segments of different radii: a smaller, anterior corneal segment, and a larger, posterior scleral segment (Figure 5.2b). The intersection of the two segments defines the limbus circle, i.e. the perimeter of the iris. In the field of anatomy, extensive measurements of the shape and dimensions of the cornea have been conducted [95]. The corneal segment covers about one-sixth of the eye, and has a radius of curvature r_C of 7.8 mm. The radius of the limbus circle r_L is 5.5 mm. The displacement d_{LC} between the center of the limbus circle and the center of the corneal sphere can be obtained as

$$d_{LC} = \sqrt{r_C^2 - r_L^2} \approx 5.53\,\text{mm} \ . \tag{5.1}$$

5.3.2 Catadioptric Model

We model the catadioptric system defined by a camera and two eyes with the following parameters: (1) the intrinsic parameters of the camera, (2) the limbus circle radius r_L, (3) the corneal sphere radius r_C, and (4) the 3D positions of the centers of each of the two corneal spheres in the camera coordinate system.

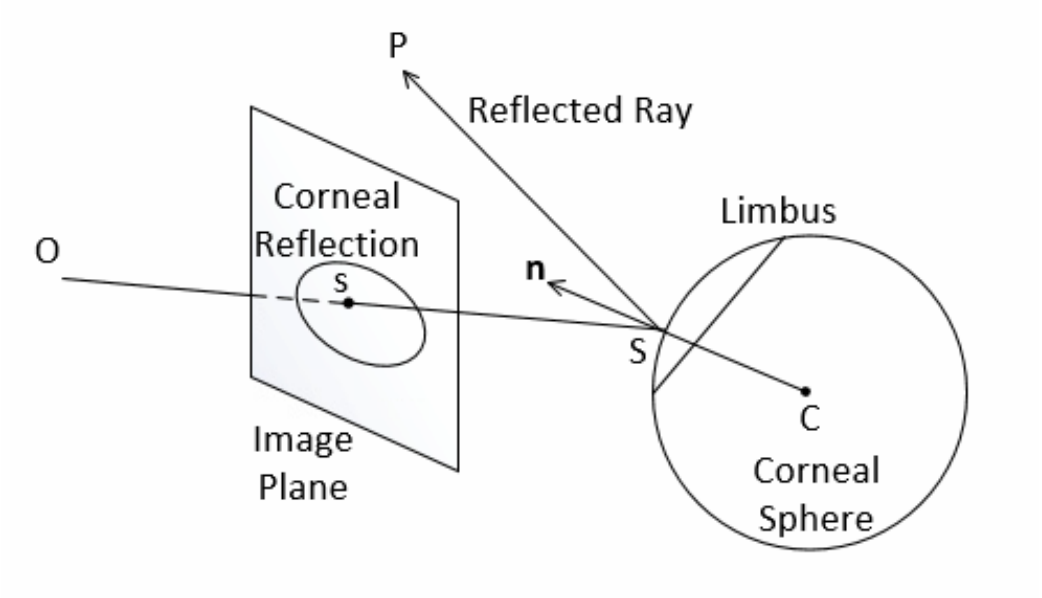


Figure 5.3.: Corneal catadioptric imaging system.

We measure the camera intrinsic parameters with a standard calibration process [96]. We assume that both eyes have the same limbus circle radius, and we use the average value of 5.5 mm. We assume both corneal spheres have the same radius, and we use the average value of 7.8 mm. We confirm the validity of these assumptions in Section 5.5.3. The 3D positions of the corneal sphere centers are found for each image as described in the next section.

Using the catadioptric model (Figure 5.3), given a pixel s in the corneal reflection, one can compute the corresponding reflected ray SP by reflecting the camera ray OS off the corneal sphere. The converse, projection operation is more challenging. Given a scene 3D point P, we compute its corneal reflection projection $s = \pi(C, P)$ by first finding its reflection point S with a fourth order equation [97]. Then s is computed by projecting S on the image plane.

5.3.3 Epipolar Geometry

Epipolar geometry is used in stereo matching to reduce the dimensionality of the correspondence search space from two to one. In our case the rays reflected by the corneal sphere are not concurrent, so the epipole is ill-defined, and traditional epipolar geometry does not apply. However, we derive epipolar-like constraints as follows. Given a pixel s_1 in the left corneal reflection (Figure 5.4), we compute its left corneal sphere reflected ray $\vec{r}$, we sample $\vec{r}$ with 3D points, and we project each 3D point P onto the image plane using the right corneal sphere, leveraging the projection operation described above. The projected points define an epipolar curve in the right corneal reflection which is known to contain the correspondence s_2 of s_1, if such a correspondence exists. Like in traditional stereo, the search for correspondences is confined to a 1D subset of the image pixels. We note that the epipolar curve can be described analytically with a quartic [98]. However, we have opted to sample the epipolar curve by sampling the 3D ray for a better control of the sampling rate, as it is challenging to sample a high-order parametric curve with steps of equal Euclidean length.

5.4 System Pipeline

Figure 5.5 shows the stages of our system pipeline.

5.4.1 Eye Region Extraction

The first stage crops the input image to only contain the eyes region. We use a Haar feature-based cascade classifier specialized for eye detection, proposed by Viola [99] and improved by Lienhart [100]. Previous approaches for extracting the eye regions proceed with a preliminary step of finding the faces in the input image. In our case, a single face dominates the input image, and it can even happen that

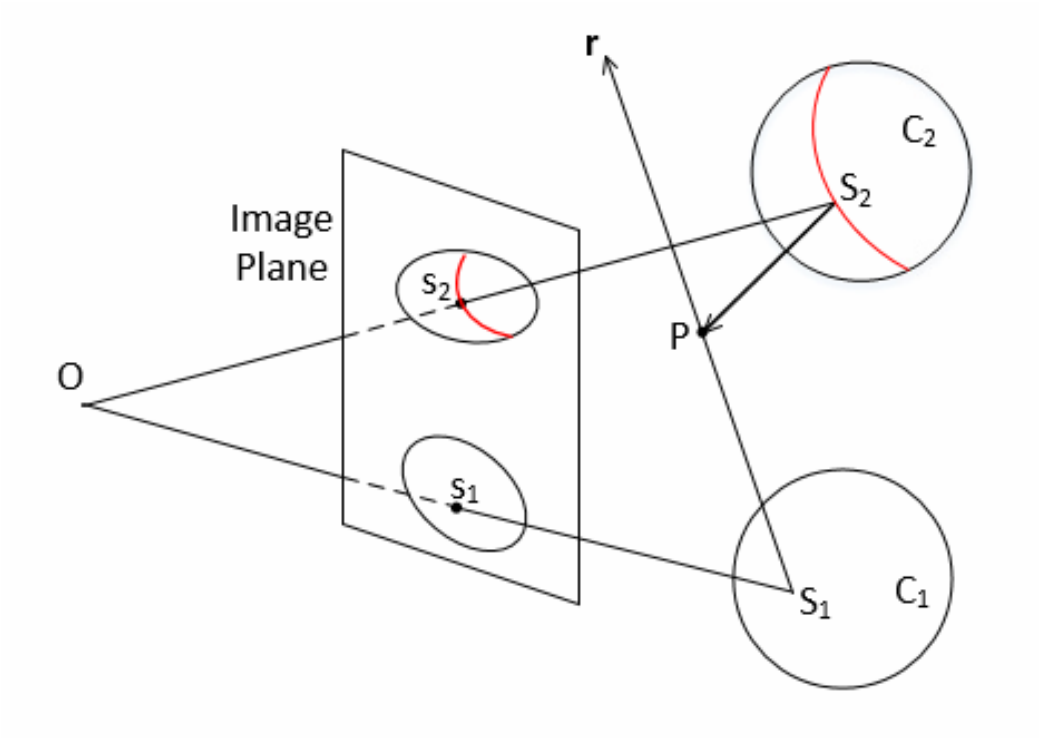


Figure 5.4.: Epipolar geometry of corneal catadioptric system.

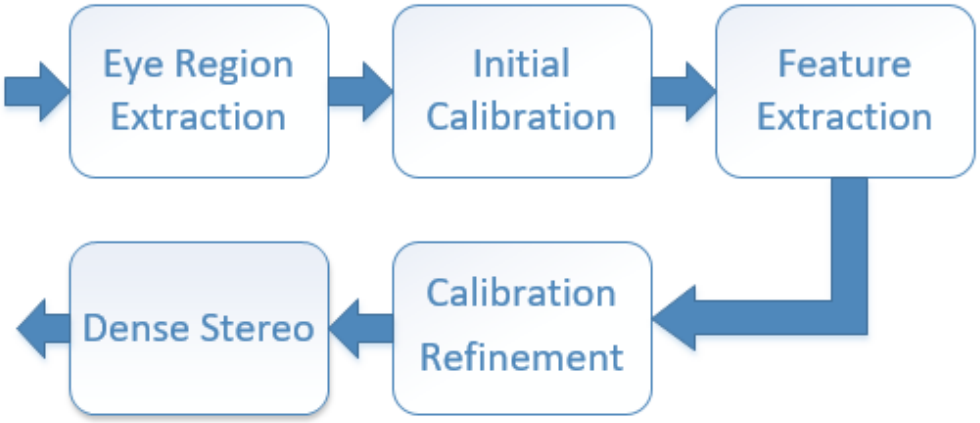
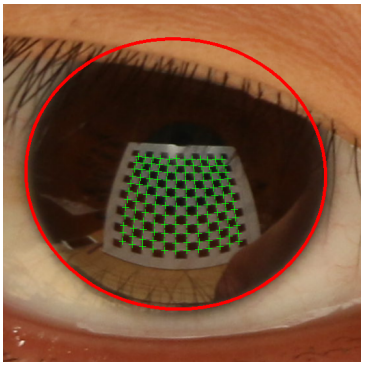


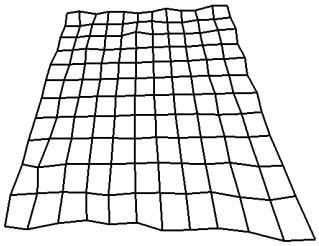
Figure 5.5.: System pipeline overview.

an image does not capture the entire face, so face detection is not necessary, and sometimes not even possible.

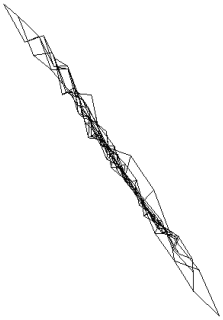
(a) Eye region of input image.



(b) Limbus and feature detection.



(c) Reconstructed checkerboard.



(d) Side view of checkerboard.

Figure 5.6.: 3D reconstruction of checkerboard. The average out of plane displacement for the checker corners is 7.3 mm.

5.4.2 Initial Calibration

The second stage of the pipeline derives an estimate of the position of the corneal spheres in the camera coordinate system. This is achieved with a method similar to the one described before in the context of achieving super-resolution of corneal reflections [90]. We summarize the procedure here for completeness.

The limbus projection is detected in each eye region using a weak perspective assumption. Prior art has also developed methods for recovering the limbus under full-perspective projection assumption [101]. However, the weak-perspective assumption is justified by the small limbus diameter relative to the distance to the camera, and by the fact that at this stage we are only deriving an initial estimate that is then refined in the subsequent pipeline stages.

The ellipse corresponding to the limbus projection is found in a downsampled eye region image using a Canny edge detector. Edge segments are assembled from edge map pixels and the ellipse is assembled from edge segments with a combinatorial search [102]. The downsampling of the eye region not only helps accelerate ellipse detection, but also serves as a low-pass filter that improves robustness. In particular, the downsampling suppresses the corneal reflections, which are an important source of noise for this stage of the pipeline. Note that the limbus circle is never entirely visible, as it is occluded by eyelids and eyelashes. Our edge detection/combinatorial search method handles well the variable occlusion of the limbus. Figure 5.6b shows a limbus detection example. Once the ellipse is determined, using the known radius of the limbus circle, the 3D position of the center and the orientation of the limbus circle are computed leveraging the known camera intrinsics. Since the radius of the corneal sphere is known, the corneal sphere center is computed using the 3D position of the center and the normal of the limbus plane [101].

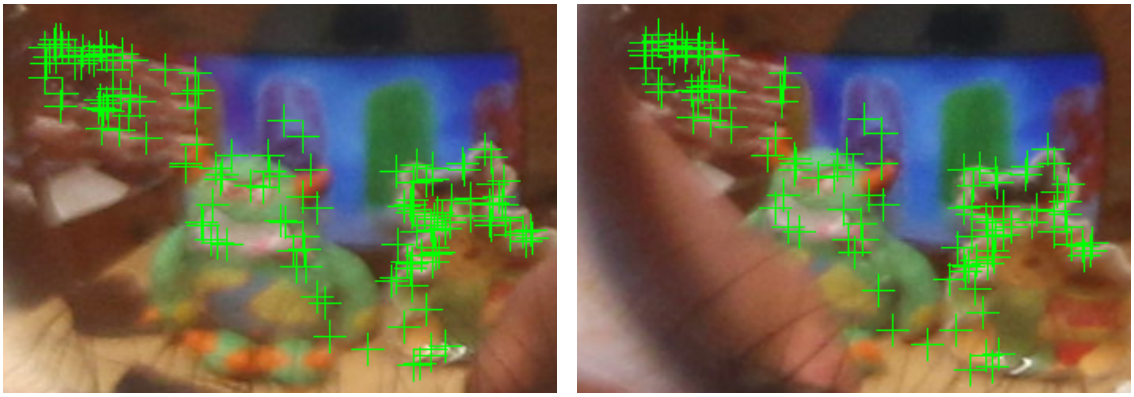


Figure 5.7.: Corneal reflection feature points for Figure 5.1.

5.4.3 Feature Extraction

The third stage of the pipeline extracts features in the reflections within the two limbus ellipses. We detect features using the FAST algorithm [103] (Figure 5.7). In anticipation of feature matching, the features are described with the BRIEF [104] algorithm. Feature scale and orientation will not vary much between the reflection in the left eye and the reflection in the right eye. Therefore, the additional memory and processing costs of scale and orientation invariant descriptors such as SIFT [105] or SURF [77] are not justified in our context. The BRIEF descriptor is binary, so the hamming distance between two descriptors can be found quickly using XOR and counting bit operations.

5.4.4 Calibration Refinement

The fourth stage of the pipeline refines the catadioptric model with a RANSAC approach we have developed (Algorithm 3). The algorithm takes as input the initial catadioptric model C_0 estimated from the limbus circle projections in the second stage of the pipeline; the set of features F_L and F_R detected in the left and right corneal reflections in the third stage of the pipeline; and the number of RANSAC iterations k over which to refine the catadioptric model.

Algorithm 3 Refinement of catadiotpric model

Input: Initial catadioptric model C_0, features F_L and F_R, number of iterations k

Output: Feature matching M, and refined catadioptric model C

1: $M_0 = \text{InitialMatching}(F_L, F_R)$
2: **for** each iteration i of k **do**
3: $\quad hex_i = \{(f_{L1}, f_{R1}), \ldots, (f_{L6}, f_{R6})\} \subset M_0$
4: $\quad C_i = \text{BundleAdjustment}(C_0, hex_i)$
5: $\quad$ **for** each (l_j, r_j) in M_0 **do**
6: $\quad\quad e_{ij} = \text{ReprojectionError}((l_j, r_j), C_i)$
7: $\quad\quad$ **if** $e_{ij} < \epsilon$ **then** // inlier correspondence
8: $\quad\quad\quad M_i \mathrel{+}= (l_j, r_j)$, $n_i{+}{+}$
9: $\quad\quad$ **end if**
10: $\quad$ **end for**
11: $\quad$ **if** $n_i > n_{best}$ **then**
12: $\quad\quad n_{best} = n_i$, $M = M_i$, $C_{best} = C_i$
13: $\quad$ **end if**
14: **end for**
15: $C = \text{BundleAdjustment}(C_{best}, M)$

An initial matching of features M_0 is computed (line 1) with an all-pairs approach that considers each feature f_L in F_L and matches it to the F_R feature with the smallest distance to f_L. However, in the case of scenes with repetitive texture, a feature could have several matches with similar quality, which can lead to matching ambiguity. We reject such features using the ratio test [105], which only keeps a feature if its second best match is significantly worse.

Based on this initial matching M_0, each iteration i of the RANSAC approach computes a possible refined catadioptric model C_i, and retains the best refinement (lines 2–14). The refined model C_i is computed with a bundle adjustment approach

from a set of six correspondences hex_i that are drawn at random from M_0 (line 3). The bundle adjustment uses a trust-region optimization [106] to find the two corneal centers C_L and C_R ($2 \times 3 = 6$ parameters), and the 3D positions P_j of the six scene features ($6 \times 3 = 18$ parameters). The optimization minimizes the sum of correspondence reprojection errors. For correspondence (f_{Lj}, f_{Rj}) the reprojection error is:

$$\left\|\pi\left(C_L, P_j\right) - f_{Lj}\right\|^2 + \left\|\pi\left(C_R, P_j\right) - f_{Rj}\right\|^2 \ , \tag{5.2}$$

where π is the projection function of the corneal catadioptric system (Section 5.3.2). An initial guess of a feature's 3D position P_j is computed by triangulation, as the midpoint of the common perpendicular segment of the two reflected rays at f_{Lj} and f_{Rj}. The six correspondences are sufficient to determine the $6 + 18 = 24$ parameters, since each of the six correspondences contributes two 2D corneal projection equations, for a total of four scalar equations:

$$\begin{aligned} \pi(C_L, P_j)_x = f_{Ljx} \ , \ \pi(C_L, P_j)_y = f_{Ljy} \ , \\ \pi(C_R, P_j)_x = f_{Rjx} \ , \ \pi(C_R, P_j)_y = f_{Rjy} \ . \end{aligned} \tag{5.3}$$

Then, using the model C_i, the correspondences in M_0 are partitioned in inlier and outlier correspondences (lines 5–10). A correspondence is considered an inlier if its reprojection error e_{ij} (Equation (5.2)) is smaller than a threshold ϵ. Inlier correspondences are counted by n_i, and are collected in set M_i. The model C_{best} with the most inlier correspondences is found over all k RANSAC iterations (lines 11–13). In a last step, C_{best} is refined over all inlier correspondences M with the bundle adjustment procedure described above for line 4), to generate the final catadioptric model C. The catadioptric model refinement reduces the average reprojection error to subpixel levels (Figure 5.8).

In conventional structure from motion, bundle adjustment is used over multiple frames, which results in a large but sparse feature correspondence matrix. This sparsity is exploited by specific optimization methods (e.g. Sparse Bundle Adjustment

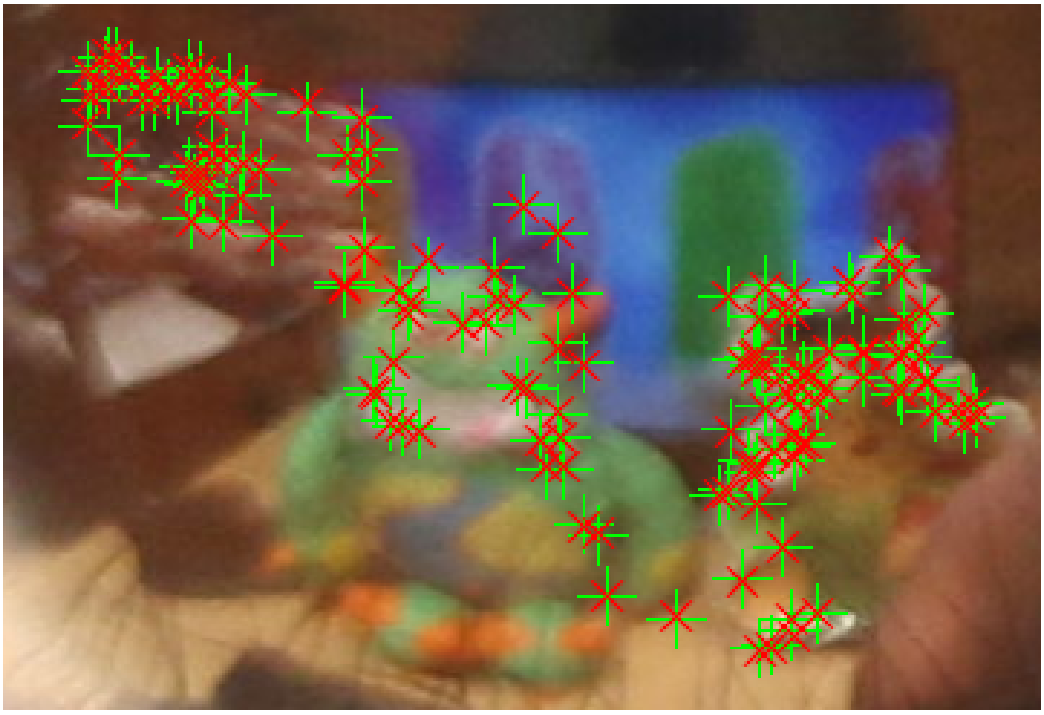


Figure 5.8.: Detected features (green) and reprojected features (red). The average reprojection error is 0.54 pixels.

based on Levenberg-Marquardt [107]). In our case, we only rely on the two images provided by the two corneal reflections, so our correspondence matrix is always full and small, hence our choice of the trust-region optimization.

5.4.5 Dense Stereo

The catadioptric model refinement stage produces a sparse reconstruction of scene geometry by computing the 3D positions of corresponding features. Scene reconstruction fidelity is increased in a final stage that attempts to compute a correspondence, and thereby a 3D point, for each corneal reflection pixel. For every pixel in the left corneal reflection we search for a correspondence p_R in the right corneal reflection along $p'_L s$ epipolar curve (Figure 5.9, top). The epipolar curve (blue) is truncated to a short arc (red) based on a depth range estimate inferred from the sparse recon-

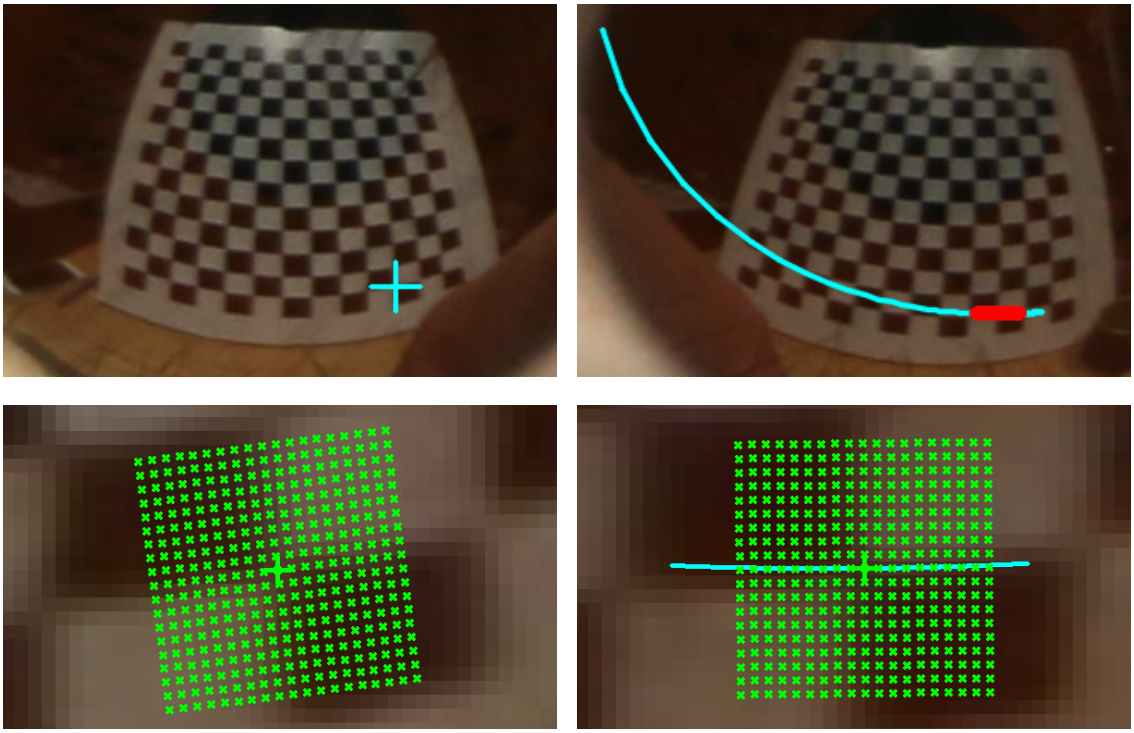


Figure 5.9.: Correspondence search on epipolar curve (top), and rotation of corresponding patches (bottom).

struction. The smaller search space accelerates correspondence finding, and increases robustness by removing from consideration parts of the image with similar texture.

Given a candidate corresponding point p_R on the epipolar curve, the matching error $E(p_L, p_R)$ is the sum of squared color differences between square patches R_{p_L} and R_{p_R} centered at p_L and p_R in the left and right reflections:

$$E(p_L, p_R) = \sum_{p_i \subset R_{p_L}} \|R_{p_L}(p_i) - R_{p_R}(F(p_i))\|^2 \ . \tag{5.4}$$

Whereas in standard stereo configuration the mapping F from R_{p_L} to R_{p_R} can be approximated with the identity, in our case there is significant rotation between R_{p_L} and R_{p_R}. We use a mapping that rotates each patch to become aligned with the epipolar curve tangent (Figure 5.9, bottom).

Figure 5.10.: Experiment setup.

5.5 Results and Discussion

Figure 5.10 shows our experimental setup. All the pictures were taken with a Canon E70D camera, which has a resolution of 5,472 x 3,648, and with a 135 mm lens. Aperture, ISO and shutter time were chosen to best capture the corneal reflections. Focus bracketing was used to obtain sharp corneal reflections, which is also aided by the fact that the reflection in a small convex surface is "shallow", forming close to the reflective surface, and focusing close to the surface will capture the entire reflection in focus, even for a small depth of field. We have tested our pipeline on several scenes: *Checkerboard* (Figure 5.6), *Toys* (Figure 5.1), *Presents* (Figure 5.11), and *Workbench* (Figure 5.12).

5.5.1 Quality

The automatically detected ellipse has an average Hausdorff distance of 1.51 pixels to a truth ellipse fitted through manually chosen points [108].

Table 5.1.: Reprojection errors [pixel].

	Checkerboard	*Toys*	*Presents*	*Workbench*
Initial	2.44	7.93	13.88	62.26
Refined	0.16	0.54	0.57	0.58

We extract features with OpenCV's FAST feature detector [103]. The initial feature matching (line 1 in Algorithm 3) has a low outlier rate, e.g. 8 out of 106 for the *Toys* scene. Consequently, a small number of RANSAC iterations (i.e. $k = 10$) are sufficient to converge to an accurate catadioptric model since the randomly selected sets of six correspondences are unlikely to contain outliers. The refinement stage reduces the average reprojection error (Equation (5.2)) substantially, as shown in Table 5.1. For the *Workbench* scene the limbus is heavily occluded in the input image, so limbus detection is approximate, which leads to a coarse initial calibration. However, even for this case, model refinement converges, reducing the reprojection error below one pixel.

For the *Checkerboard* scene, the average out of plane displacement for the 144 3D points recovered at the 12×12 checker corners is 7.3 mm. For the dense-stereo reconstructed points, the average out of plane displacement is also 7.3 mm. The length of the reconstructed diagonal of the checkerboard is 0.61 m, whereas the true diagonal is 0.59 m, which corresponds to a 2.7% error. For a qualitative assessment of our depth maps, we scanned the *Toys* and the *Presents* scenes with a depth camera (i.e. a *Structure* sensor). The truth geometry aligns with the geometry reconstructed from corneal reflections (Figures 5.1c and 5.11). For the *Presents* scene we fitted planes to the box faces, with an average error of 15.3 mm. The normals of parallel faces had an average angle error of 6.2°.

Figure 5.11.: *Presents* scene: reflection, and reconstruction aligned with truth geometry (grey points), for comparison.

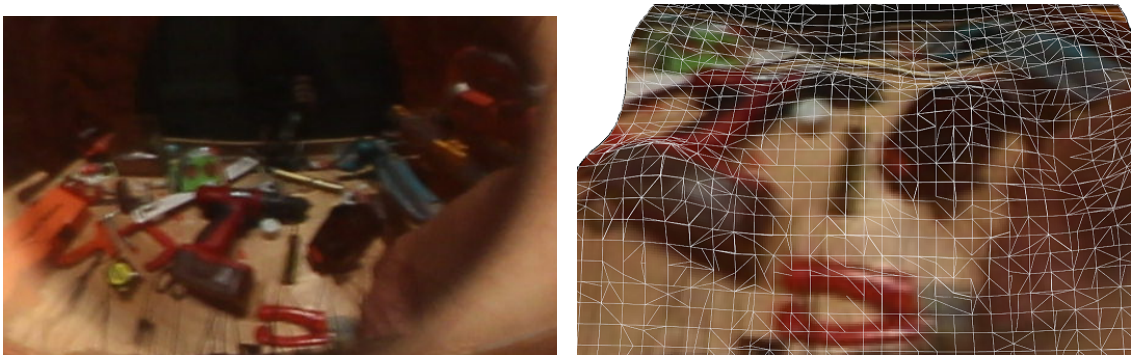


Figure 5.12.: *Workbench* scene: reflection and reconstruction.

5.5.2 Speed

We measured performance on an Intel(R) Core(TM) i5-7600K 3.8 GHz workstation. The running times of each stage of our pipeline are given in Table 5.2. For eye region extraction, we use the Haar cascade classifier provided in OpenCV. A minimum eye region size is set to avoid false detections. For the limbus detection in the

Table 5.2.: Typical running times for our pipeline.

Pipeline stage	Time [ms]
Eye region extraction	53
Initial calibration	82
Feature extraction	50
Calibration refinement (Algorithm 3)	
Initial feature matching (line 1)	2
RANSAC iterations (lines 2–14)	20
Final bundle adjustment (line 15)	1,053
Dense Stereo	287,327

initial calibration, we start the search at the center of the eye region. The bulk of the limbus detection time goes to downsampling the image. The dense stereo stage is by far the slowest, but also the best candidate for parallelization.

5.5.3 Error Analysis

Like any depth from stereo system, our depth accuracy depends on the baseline, on the image resolution, and on the correspondence detection error. There isn't much flexibility for the baseline, which is fixed to the interpupillary distance. In terms of resolution, we use one of the off-the-shelf highest resolution cameras. Due to the high curvature of the corneal sphere, correspondence detection errors result in larger depth errors than in the case of conventional stereo, as reflected rays are more divergent. The detection error is commensurate to the feature reprojection error, which in our experiments is consistently below one pixel. For our system, a one pixel detection error translates to an average depth error of 20 mm at 0.5 m. This error is larger closer to the limbus circle, where reflected rays are more divergent.

We use a catadioptric model that assumes known and equal limbus circle radii. The limbus circle radius is only used in the initial calibration stage, which provides an initial guess for the model refinement stage. In all our experiments this initial guess was good enough for the model refinement stage to converge, which indicates that one can safely use the known and equal limbus circle radii assumption. Our catadioptric model also assumes that the corneal surfaces are spherical, and that the corneal sphere radii are known and equal. We have investigated the reconstruction error sensitivity to deviations from these two assumptions analytically. The reconstruction error is computed for a 3D point P at a typical distance from the eyes of 0.5 m. The projections p_L and p_R of P in the corneal reflections are computed with our ideal catadioptric model C. Then, for a given imperfect catadioptric model C', we compute a deviated position P' of P as follows. First, the camera rays at p_L and p_R are reflected according to C', and then the reflected rays are triangulated to obtain P'. The reconstruction error is defined as the Euclidean distance between P and P'.

Figure 5.13 shows the reconstruction error dependence on cornea eccentricity and on left/right eye asymmetry. The same 0 to 0.2 range is used for both independent variables. Cornea eccentricity is modeled by assuming the true cornea is in fact an ellipsoid. For an eccentricity of 0.2, which corresponds to a small/large ellipse axis ratio of 0.98, the reconstruction error is 38 mm. The eye asymmetry is quantified as the ratio of the radii of the left and right eye corneal spheres. For an eye asymmetry of 10%, the error is 40 mm. This analysis indicates that the reconstruction error is quite sensitive to these two parameters.

In our anatomy research review we did not find a human population range for these parameters. We experimented with extending our bundle adjustment to optimize for eye asymmetry as well, but the reprojection errors did not decrease significantly. Furthermore, we have also investigated the validity of our assumptions empirically, by reconstructing our scenes from reflections captured from two high-grade steel bearing balls of similar size to the human corneal spheres (Figure 5.14). The bearing balls are truly spherical and of equal size, so the bearing balls catadioptric system satis-

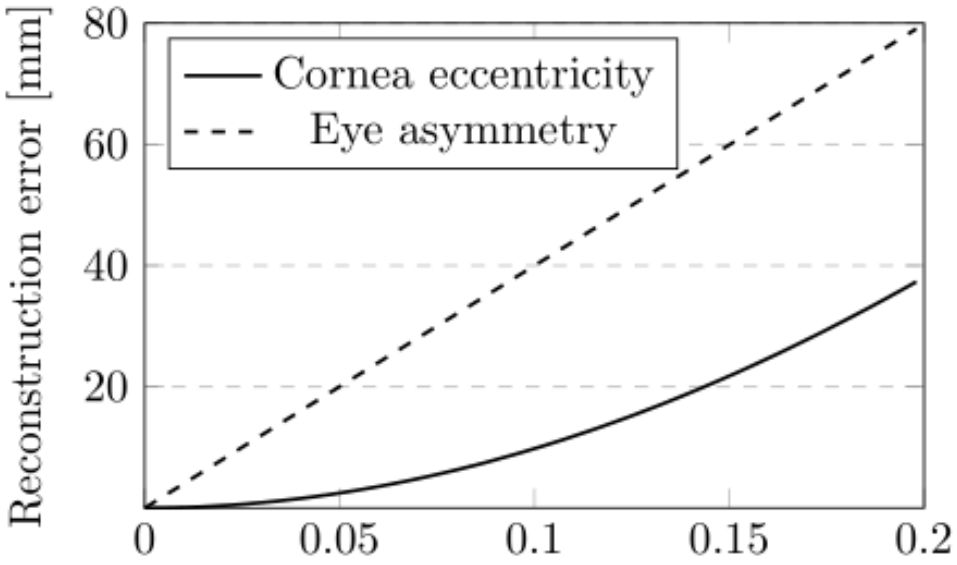


Figure 5.13.: Reconstruction error analysis.

Figure 5.14.: Steel ball catadioptric system, for comparison.

fies all our assumptions. The reconstructed scene accuracy was comparable to the reconstructions from corneal reflections for the *Checkerboard* scene, which indicates indirectly that our corneal catadioptric system assumptions are valid.

5.6 Conclusions and Future Work

We described a pipeline for extracting 3D scene structure from high resolution corneal reflections. The system first calibrates the position of the eyes with respect to the camera with subpixel accuracy, and then uses the resulting catadioptric model to triangulate corresponding corneal reflection features and pixels.

One limitation of the system stems from the assumption that the input image provides a perfect corneal reflection. Future work should take into account the iris texture, which is a considerable source of noise for light colored eyes. Methods for separating the local from the global illumination [109] could be used to this effect. Another limitation of the current pipeline implementation is that the dense stereo stage relies on a naive patch color matching algorithm, which reduces the quality of the 3D scene reconstruction. Our paper contributes a subpixel accurate calibration of the corneal catadioptric imaging system, which can be readily used with more sophisticated stereo matching algorithms, such as for example those that exploit scene geometry coherence [110], [111], [112].

Another direction of future work is to accelerate the pipeline to interactive performance, which allows accumulating scene 3D structure over several frames, or even from a video stream. A first step is to implement the dense stereo stage on a GPU. For a stationary camera, the 3D points contributed by each frame are already in a common coordinate system and can be readily merged, without alignment.

Future work to extend our method beyond the lab setting is challenging. Our work already reduces the calibration error of the catadioptric system below one pixel, which is an order of magnitude improvement over prior art. But the inherent limitation that prevents the reconstruction of scenes outside the lab is the large distance from the eyes to the scene, relative to the interpupillary distance and to the corneal reflection pixel resolution. Indeed, even for a 0.1 pixel reprojection error, which is the standard for the calibration error of simple optical systems with one camera, corneal reflection reconstructions will incur errors of 6.33, 25.3, and 602mm at scene distances of 1,

2, and 10m, for a 5,472 x 3,648 resolution camera placed at 0.5m from the eyes. Our corneal catadioptric system calibration and scene reconstruction pipeline already achieves the best results afforded by the current resolution of commercial digital cameras, further improvements will have to come from increasing the resolution of the corneal reflections.

Although images now have sufficient resolution for direct display, giving the user the option to zoom in on regions of interest, such as faces, and extracting scene information from corneal and other fortuitous reflections will continue to benefit from further increases of image resolution. Many of these applications do not require high resolution throughout the image, and a promising direction of future work in imaging system design is to achieve a variable resolution over the field of view. Although consumer-level devices, such as phones, now have multiple cameras with various focal lengths, achieving a high resolution at application specified locations in the field of view remains intractable. A more promising approach is to rely on a high resolution sensor with a wide angle lens and to read and save only the pixels needed, resulting in a versatile imaging system that helps leveraging secondary rays for scene acquisition.

6 CONCLUSIONS

Our thesis advocates to design custom and specific solutions for connecting images acquired from different locations, solutions that are inexpensive yet effective for each AR problem. We presented solutions tailored for simulated transparent display continuity, effective mentor workspace visualization in AR surgical telementoring, and occlusion removal for effective diminished reality visualization. In essence, all these challenges have the fundamental underlying problem of establishing a connection between pairs of images captured from different viewpoints. Denoting the image resolution as $w \times h$, we argue that a dense, per-pixel connection, which comes at least at $O(wh)$ cost, is not only difficult to establish, but it is also insufficient due to disocclusion errors. We show that, instead, a lightweight connection of cost $O(w)$ or even $O(1)$ can be designed to address each problem effectively.

The first AR challenge that we focused on is the avoidance of the discontinuity at the boundary of a video see-through AR display. Assuming the scene geometry is sufficiently away from the user, we showed that an $O(1)$ mapping from the camera's viewpoint to the user viewpoint is sufficient to show the user what they would see if the display were not there, producing a convincing display transparency effect. The scene captured by the back-facing camera is warped to the user viewpoint by alleviating the view direction differences between the camera and the user. This way, the frame acquired by the camera is cropped to what the user would see through the display frame, removing visualization discontinuity across the display boundary. A theoretical analysis shows that the achieved transparency effect has an error of less than 5% when the scene is farther than 6 m. The effectiveness of simulated transparency is also validated empirically on a self-contained and compact simulated transparent display implemented on a mobile phone.

The second AR challenge is to effectively convey the workspace to the mentor in telementoring. The workspace visualization has to allow the mentor to understand the current state of the task performed by the mentee in order to give adequate guidance, leading to successful telementoring. We have shown that a simple planar approximation of workspace geometry can be computed quickly and that it is effective. The planar proxy establishes an $O(1)$ mapping between the mentor and mentee viewpoints. We project the mentee video feed onto the proxy, and then reproject the textured proxy to a static viewpoint for the mentor. This provides an effective real-time visualization of the workspace to the mentor. The visualization is of high quality, i.e. without distortions due to inadequate geometric approximation, and without tears due to disocclusion errors. All scene lines project to lines in the visualization. All these properties contribute to the effectiveness of the workspace visualization and therefore of the telementoring application.

The third challenge investigated by this thesis is the removal of an occluder from a video feed, in real time. We presented a method that establishes the mapping from the view of an auxiliary camera to the user view in $O(w)$ time. The mapping is based on a global rotation and a local refinement generated from contour pixel correspondences. The mapping is sufficient for painting over the occluder using pixels from the auxiliary camera, which amounts to a convincing transparency effect. The result is a multi-perspective visualization, where the scene surrounding the occluder is shown conventionally, from the user viewpoint, and the scene behind the occluder is shown from the second camera viewpoint. A gradual transition is implemented by our method to connect the two perspectives seamlessly from one viewpoint to the other.

One traditional approach applicable to all the challenges above is to acquire a complete geometry model of the scene, then to render the scene geometry from one viewpoint, with projective texture mapping of images acquired from the second viewpoint. However, in the AR contexts investigated by our thesis, this one-size-fits-all approach is not only time consuming, but also insufficient due to disocclusion errors.

Depth acquisition has a cost of at least $O(wh)$ since it requires a dense, per-pixel correspondence mapping to establish the connection between two images. Even with perfect scene geometry, problems like simulated transparent display continuity or mentor workspace visualization are not adequately solved due to persistent disocclusion errors. Our proposed custom solutions for such challenges are not only faster but they also produce better results.

In future work, the algorithms developed in the individual contexts of improving visualization continuity for simulated transparent displays, of surgical telementoring, of diminished reality, and of 3D scene reconstruction from corneal reflections can be combined. For example, one direction of future work could explore devising a simulated transparent that also handles the case of nearby geometry by acquiring the real world scene from the user viewpoint by capturing corneal reflections. Another possibility is to bypass user tracking and geometry acquisition altogether and to achieve transparent display visualization continuity with a user head mounted camera that relies on the display camera for the secondary video feed from which to borrow pixels to inpaint the occluding tablet, making it to appear transparent.

Our work takes a step towards realizing the potential of AR technology. Current video see-through displays suffer from the dual-view perceptual issue, i.e. part of the augmented video on the display is redundant with the scene viewed directly by the user. By removing the discontinuity at the boundary, our method helps to remove the additional cognitive load required from the user to translate the annotated visualization to the real world context. Truly transparent hand-held displays will probably remain elusive in the foreseeable future, so improving the visualization effectiveness of simulated transparent displays is likely to remain an infrastructure contribution that will continue to benefit many AR applications.

By conveying the mentee workspace to the mentor effectively, our method improves the mentor's scene understanding, which in turn increases the quality of the guidance they provide, and ultimately increases mentee performance. Our method has shown significantly better results than a raw, unstabilized first-person view of

the workspace. Our AR telementoring system has proven its effectiveness even in challenging scenarios such as that of practice cricothyroidotomies in austere settings.

Diminished reality is an important case of AR visualization that improves the user's visual perception of the scene by reducing scene clutter. One diminished reality approach is to paint over occluders, rendering them transparently. We have demonstrated the effectiveness of an occluder removal technique based on a fast connection between the user video and an auxiliary video, which opens the door to effective real time occlusion management by leveraging a multitude of video feeds, each with its own viewpoint. For example, a group of users watching an event can provide an occlusion-free visualization of the event to a control and command center, or to each of the users, using the multiple video sources.

Longer term, we see AR as the ultimate human computer interface, where the quality and the convenience of the integration of the visual enhancements into the user's view of the real world will make traditional displays obsolete.

REFERENCES

[1] DWF Van Krevelen and Ronald Poelman. A survey of augmented reality technologies, applications and limitations. *International journal of virtual reality*, 9(2):1–20, 2010.

[2] Max Talbot, EJ Harvey, GK Berry, R Reindl, H Tien, DJ Stinner, and G Slobogean. A pilot study of surgical telementoring for leg fasciotomy. *Journal of the Royal Army Medical Corps*, 164(2):83–86, 2018.

[3] Daniel Andersen, Voicu Popescu, Maria Eugenia Cabrera, Aditya Shanghavi, Gerardo Gomez, Sherri Marley, Brian Mullis, and Juan P Wachs. Medical telementoring using an augmented reality transparent display. *Surgery*, 159(6):1646–1653, 2016.

[4] Shohei Mori, Sei Ikeda, and Hideo Saito. A survey of diminished reality: Techniques for visually concealing, eliminating, and seeing through real objects. *IPSJ Transactions on Computer Vision and Applications*, 9(1):1–14, 2017.

[5] Ernst Kruijff, J Edward Swan, and Steven Feiner. Perceptual issues in augmented reality revisited. In *2010 IEEE International Symposium on Mixed and Augmented Reality*, pages 3–12. IEEE, 2010.

[6] Samsung display introduces first mirror and transparent oled display panels. `http://www.businesswire.com/news/home/20150609006775/en/Samsung-Display-Introduces-Mirror-Transparent-OLED-Display`. Accessed: 2015-12-14.

[7] Takumi Yoshida, Shinobu Kuroki, Hideaki Nii, Naoki Kawakami, and Susumu Tachi. Arscope. ACM SIGGRAPH 2008 new tech demos.

[8] Domagoj Baričević, Cha Lee, Matthew Turk, Tobias Höllerer, and Doug A Bowman. A hand-held ar magic lens with user-perspective rendering. In *2012 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 197–206. IEEE, 2012.

[9] Ali Samini and Karljohan Lundin Palmerius. A perspective geometry approach to user-perspective rendering in hand-held video see-through augmented reality. In *Proceedings of the 20th ACM Symposium on Virtual Reality Software and Technology*, pages 207–208, 2014.

[10] Emile Zhang, Hideo Saito, and Francois De Sorbier. From smartphone to virtual window. In *2013 IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*, pages 1–6. IEEE, 2013.

[11] Hikari Uchida and Takashi Komuro. Geometrically consistent mobile ar for 3d interaction. In *Proceedings of the 4th Augmented Human International Conference*, pages 229–230, 2013.

[12] Klen Čopič Pucihar, Paul Coulton, and Jason Alexander. Creating a stereoscopic magic-lens to improve depth perception in handheld augmented reality. In *Proceedings of the 15th international conference on Human-computer interaction with mobile devices and services*, pages 448–451, 2013.

[13] Makoto Tomioka, Sei Ikeda, and Kosuke Sato. Pseudo-transparent tablet based on 3d feature tracking. In *Proceedings of the 5th Augmented Human International Conference*, pages 1–2, 2014.

[14] Yuko Unuma, Takehiro Niikura, and Takashi Komuro. See-through mobile ar system for natural 3d interaction. In *Proceedings of the companion publication of the 19th international conference on Intelligent User Interfaces*, pages 17–20, 2014.

[15] Domagoj Baričević, Tobias Höllerer, Pradeep Sen, and Matthew Turk. User-perspective augmented reality magic lens from gradients. In *Proceedings of the 20th ACM Symposium on Virtual Reality Software and Technology*, pages 87–96, 2014.

[16] Jens Grubert, Hartmut Seichter, and Dieter Schmalstieg. Towards user perspective augmented reality for public displays. In *2014 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 339–340. IEEE, 2014.

[17] Amazon fire phone. `http://www.amazon.com/firephone`. 2015-12-14.

[18] Glass press. https://sites.google.com/site/glasscomms/. 2015-12-15.

[19] Etai M Bogen, Knut M Augestad, Hiten RH Patel, and Rolv-Ole Lindsetmo. Telementoring in education of laparoscopic surgeons: An emerging technology. *World journal of gastrointestinal endoscopy*, 6(5):148, 2014.

[20] Jeffrey H Shuhaiber. Augmented reality in surgery. *Archives of surgery*, 139(2):170–174, 2004.

[21] Long Chen, Thomas W Day, Wen Tang, and Nigel W John. Recent developments and future challenges in medical mixed reality. In *2017 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 123–135. IEEE, 2017.

[22] Ehsan Azimi, Jayfus Doswell, and Peter Kazanzides. Augmented reality goggles with an integrated tracking system for navigation in neurosurgery. In *Virtual Reality Short Papers and Posters (VRW), 2012 IEEE*, pages 123–124. IEEE, 2012.

[23] Huixiang Wang, Fang Wang, Anthony Peng Yew Leong, Lu Xu, Xiaojun Chen, and Qiugen Wang. Precision insertion of percutaneous sacroiliac screws using a novel augmented reality-based navigation system: a pilot study. *International orthopaedics*, 40(9):1941–1947, 2016.

[24] Goh Chuan Meng, A Shahzad, NM Saad, Aamir Saeed Malik, and Fabrice Meriaudeau. Prototype design for wearable veins localization system using near infrared imaging technique. In *Signal Processing & Its Applications (CSPA), 2015 IEEE 11th International Colloquium on*, pages 112–115. IEEE, 2015.

[25] Joseph J LaViola Jr. A discussion of cybersickness in virtual environments. *ACM SIGCHI Bulletin*, 32(1):47–56, 2000.

[26] Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Transactions on pattern analysis and machine intelligence*, 22(11):1330–1334, 2000.

[27] Xiao-Shan Gao, Xiao-Rong Hou, Jianliang Tang, and Hang-Fei Cheng. Complete solution classification for the perspective-three-point problem. *IEEE transactions on pattern analysis and machine intelligence*, 25(8):930–943, 2003.

[28] Dominic Campano, Jose A Robaina, Nicholas Kusnezov, John C Dunn, and Brian R Waterman. Surgical management for chronic exertional compartment syndrome of the leg: a systematic review of the literature. *Arthroscopy*, 32(7):1478–1486, 2016.

[29] American College of Surgeons Committee on Trauma, J. Fildes, J.W. Meredith, D.B. Hoyt, F.A. Luchette, M.W. Bowyer, P.A. Byers, E.E. Cornwell, J. Cuschieri, R.I. Gross, et al. *ASSET (Advanced Surgical Skills for Exposure in Trauma): Exposure Techniques When Time Matters*. American College of Surgeons, 2010.

[30] Samuel Sanford Shapiro and Martin B Wilk. An analysis of variance test for normality (complete samples). *Biometrika*, 52(3/4):591–611, 1965.

[31] Henry B Mann and Donald R Whitney. On a test of whether one of two random variables is stochastically larger than the other. *The annals of mathematical statistics*, pages 50–60, 1947.

[32] Frank Wilcoxon. Individual comparisons by ranking methods. *Biometrics bulletin*, 1(6):80–83, 1945.

[33] Colin F Mackenzie, Evan Garofalo, Stacy Shackelford, Valerie Shalin, Kristy Pugh, Hegang Chen, Adam Puche, Jason Pasley, Babak Sarani, Sharon Henry, et al. Using an individual procedure score before and after the advanced surgical skills exposure for trauma course training to benchmark a hemorrhage-control performance metric. *Journal of surgical education*, 72(6):1278–1289, 2015.

[34] H Sebajang, P Trudeau, A Dougall, S Hegge, C McKinley, and M Anvari. The role of telementoring and telerobotic assistance in the provision of laparoscopic colorectal surgery in rural areas. *Surgical Endoscopy and Other Interventional Techniques*, 20(9):1389–1393, 2006.

[35] Sarah Treter, Nancy Perrier, Julie Ann Sosa, and Sanziana Roman. Telementoring: a multi-institutional experience with the introduction of a novel surgical approach for adrenalectomy. *Annals of surgical oncology*, 20(8):2754–2758, 2013.

[36] Daniel Andersen, Voicu Popescu, Maria Eugenia Cabrera, Aditya Shanghavi, Gerardo Gómez, Sherri Marley, Brian Mullis, and Juan Pablo Wachs. Avoiding focus shifts in surgical telementoring using an augmented reality transparent display. In *MMVR*, volume 22, pages 9–14, 2016.

[37] Andrius Budrionis, Gunnar Hartvigsen, and Johan Gustav Bellika. Camera movement during telementoring and laparoscopic surgery: Challenges and innovative solutions. In *SHI 2015, Proceedings from The 13th Scandinavien Conference on Health Informatics, June 15-17, 2015, Tromsø, Norway*, number 115

in Linköping Electronic Conference Proceedings, pages 1–5. Linköping University Electronic Press, Linköpings universitet, 2015.

[38] Hideaki Kuzuoka. Spatial workspace collaboration: a sharedview video support system for remote collaboration capability. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 533–540. ACM, 1992.

[39] William W Gaver, Abigail Sellen, Christian Heath, and Paul Luff. One is not enough: Multiple views in a media space. In *Proceedings of the INTERACT'93 and CHI'93 conference on Human factors in computing systems*, pages 335–341, 1993.

[40] Susan R Fussell, Leslie D Setlock, and Robert E Kraut. Effects of head-mounted and scene-oriented video systems on remote collaboration on physical tasks. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 513–520. ACM, 2003.

[41] Steffen Gauglitz, Benjamin Nuernberger, Matthew Turk, and Tobias Höllerer. World-stabilized annotations and virtual scene navigation for remote collaboration. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*, pages 449–459. ACM, 2014.

[42] Taehee Lee and Tobias Höllerer. Viewpoint stabilization for live collaborative video augmentations. In *Mixed and Augmented Reality, 2006. ISMAR 2006. IEEE/ACM International Symposium on*, pages 241–242. IEEE, 2006.

[43] Gun A Lee, Theophilus Teo, Seungwon Kim, and Mark Billinghurst. Mixed reality collaboration through sharing a live panorama. In *SIGGRAPH Asia 2017 Mobile Graphics & Interactive Applications*, page 14. ACM, 2017.

[44] Brent A Ponce, Mariano E Menendez, Lasun O Oladeji, Charles T Fryberger, and Phani K Dantuluri. Emerging technology in surgical education: combining real-time augmented reality and wearable computing devices. *Orthopedics*, 37(11):751–757, 2014.

[45] Shuaicheng Liu, Lu Yuan, Ping Tan, and Jian Sun. Bundled camera paths for video stabilization. *ACM Transactions on Graphics (TOG)*, 32(4):78, 2013.

[46] Miao Wang, Guo-Ye Yang, Jin-Kun Lin, Song-Hai Zhang, Ariel Shamir, Shao-Ping Lu, and Shi-Min Hu. Deep online video stabilization with multi-grid warping transformation learning. *IEEE Transactions on Image Processing*, 28(5):2283–2292, 2019.

[47] Johannes Kopf, Michael F Cohen, and Richard Szeliski. First-person hyperlapse videos. *ACM Transactions on Graphics (TOG)*, 33(4):78, 2014.

[48] Greg Welch, Diane H Sonnenwald, Henry Fuchs, Bruce Cairns, Ketan Mayer-Patel, Ruigang Yang, Herman Towles, Adrian Ilie, Srinivas Krishnan, Hanna M Söderholm, et al. Remote 3d medical consultation. In *Virtual realities*, pages 139–159. Springer, 2011.

[49] Seungwon Kim, Mark Billinghurst, and Gun Lee. The effect of collaboration styles and view independence on video-mediated remote collaboration. *Computer Supported Cooperative Work (CSCW)*, 27(3-6):569–607, 2018.

[50] Gun A Lee, Theophilus Teo, Seungwon Kim, and Mark Billinghurst. A user study on mr remote collaboration using live 360 video. In *2018 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 153–164. IEEE, 2018.

[51] Steffen Gauglitz, Cha Lee, Matthew Turk, and Tobias Höllerer. Integrating the physical environment into mobile remote collaboration. In *Proceedings of the 14th international conference on Human-computer interaction with mobile devices and services*, pages 241–250. ACM, 2012.

[52] Jakob Zillner, Erick Mendez, and Daniel Wagner. Augmented reality remote collaboration with dense reconstruction. In *2018 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, pages 38–39. IEEE, 2018.

[53] Jörg Müller, Tobias Langlotz, and Holger Regenbrecht. Panovc: Pervasive telepresence using mobile phones. In *2016 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 1–10. IEEE, 2016.

[54] Matt Adcock, Stuart Anderson, and Bruce Thomas. Remotefusion: real time depth camera fusion for remote collaboration on physical tasks. In *Proceedings of the 12th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and Its Applications in Industry*, pages 235–242. ACM, 2013.

[55] Bruce Bridgeman, Derek Hendry, and Lawrence Stark. Failure to detect displacement of the visual world during saccadic eye movements. *Vision research*, 15(6):719–722, 1975.

[56] Microsoft hololens. `https://www.microsoft.com/en-us/hololens`. Accessed: 2019-02-14.

[57] Sandra G Hart and Lowell E Staveland. Development of nasa-tlx (task load index): Results of empirical and theoretical research. In *Advances in psychology*, volume 52, pages 139–183. Elsevier, 1988.

[58] Robert S Kennedy, Norman E Lane, Kevin S Berbaum, and Michael G Lilienthal. Simulator sickness questionnaire: An enhanced method for quantifying simulator sickness. *The international journal of aviation psychology*, 3(3):203–220, 1993.

[59] Howard Levene. Robust tests for equality of variances. *Contributions to probability and statistics. Essays in honor of Harold Hotelling*, pages 279–292, 1961.

[60] Ronald Aylmer Fisher. Statistical methods for research workers. In *Breakthroughs in statistics*, pages 66–70. Springer, 1992.

[61] Carlo Bonferroni. Teoria statistica delle classi e calcolo delle probabilita. *Pubblicazioni del R Istituto Superiore di Scienze Economiche e Commericiali di Firenze*, 8:3–62, 1936.

[62] Michael Frigge, David C Hoaglin, and Boris Iglewicz. Some implementations of the boxplot. *The American Statistician*, 43(1):50–54, 1989.

[63] Siavash Zokai, Julien Esteve, Yakup Genc, and Nassir Navab. Multiview paraperspective projection model for diminished reality. In *Proceedings of the 2nd IEEE/ACM International Symposium on Mixed and Augmented Reality*, page 217. IEEE Computer Society, 2003.

[64] Shohei Mori, Fumihisa Shibata, Asako Kimura, and Hideyuki Tamura. Efficient use of textured 3d model for pre-observation-based diminished reality. In *2015 IEEE International Symposium on Mixed and Augmented Reality Workshops*, pages 32–39. IEEE, 2015.

[65] Masayuki Takemura and Yuichi Ohta. Diminishing head-mounted display for shared mixed reality. In *Proceedings of the 1st International Symposium on Mixed and Augmented Reality*, page 149. IEEE Computer Society, 2002.

[66] Zhuwen Li, Yuxi Wang, Jiaming Guo, Loong-Fah Cheong, and Steven ZhiYing Zhou. Diminished reality using appearance and 3d geometry of internet photo collections. In *2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 11–19. IEEE, 2013.

[67] Meng-Lin Wu and Voicu Popescu. Rgbd temporal resampling for real-time occlusion removal. In *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, page 7. ACM, 2019.

[68] Kunihiro Hasegawa and Hideo Saito. Diminished reality for hiding a pedestrian using hand-held camera. In *2015 IEEE International Symposium on Mixed and Augmented Reality Workshops*, pages 47–52. IEEE, 2015.

[69] Christopher Mei, Eric Sommerlade, Gabe Sibley, Paul Newman, and Ian Reid. Hidden view synthesis using real-time visual slam for simplifying video surveillance analysis. In *2011 IEEE International Conference on Robotics and Automation*, pages 4240–4245. IEEE, 2011.

[70] Yoshinari Kameda, Taisuke Takemasa, and Yuichi Ohta. Outdoor see-through vision utilizing surveillance cameras. In *Proceedings of the 3rd IEEE/ACM International Symposium on Mixed and Augmented Reality*, pages 151–160. IEEE Computer Society, 2004.

[71] Akihito Enomoto and Hideo Saito. Diminished reality using multiple handheld cameras. In *Proc. ACCV*, volume 7, pages 130–135, 2007.

[72] Benjamin Avery, Wayne Piekarski, and Bruce H Thomas. Visualizing occluded physical objects in unfamiliar outdoor augmented reality environments. In *Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 1–2. IEEE Computer Society, 2007.

[73] Peter Barnum, Yaser Sheikh, Ankur Datta, and Takeo Kanade. Dynamic seethroughs: Synthesizing hidden views of moving objects. In *2009 8th IEEE International Symposium on Mixed and Augmented Reality*, pages 111–114. IEEE, 2009.

[74] François Rameau, Hyowon Ha, Kyungdon Joo, Jinsoo Choi, Kibaek Park, and In So Kweon. A real-time augmented reality system to see-through cars. *IEEE transactions on visualization and computer graphics*, 22(11):2395–2404, 2016.

[75] Siim Meerits and Hideo Saito. Real-time diminished reality for dynamic scenes. In *2015 IEEE International Symposium on Mixed and Augmented Reality Workshops*, pages 53–59. IEEE, 2015.

[76] Amit Singhal et al. Modern information retrieval: A brief overview. *IEEE Data Eng. Bull.*, 24(4):35–43, 2001.

[77] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *European conference on computer vision*, pages 404–417. Springer, 2006.

[78] Marius Muja and David G Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. *VISAPP (1)*, 2(331-340):2, 2009.

[79] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[80] Jorge Nocedal and Stephen Wright. *Numerical optimization*. Springer Science & Business Media, 2006.

[81] Paul W Holland and Roy E Welsch. Robust regression using iteratively reweighted least-squares. *Communications in Statistics-theory and Methods*, 6(9):813–827, 1977.

[82] Neil A Dodgson. Variation and extrema of human interpupillary distance. In *Electronic imaging 2004*, pages 36–46. International Society for Optics and Photonics, 2004.

[83] Paul Debevec. Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography. In *ACM SIGGRAPH 2008 classes*, page 32. ACM, 2008.

[84] Shree K Nayar. Catadioptric omnidirectional camera. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 482–488. IEEE, 1997.

[85] Sujit Kuthirummal and Shree K. Nayar. Multiview radial catadioptric imaging for scene capture. In *ACM SIGGRAPH 2006 Papers*, SIGGRAPH '06, page 916–923, New York, NY, USA, 2006. Association for Computing Machinery.

[86] Andrea Fasano, Marco Callieri, Paolo Cignoni, and Roberto Scopigno. Exploiting mirrors for laser stripe 3d scanning. In *3-D Digital Imaging and Modeling, 2003. 3DIM 2003. Proceedings. Fourth International Conference on*, pages 243–250. IEEE, 2003.

[87] Christian Nitschke, Atsushi Nakazawa, and Haruo Takemura. Corneal imaging revisited: An overview of corneal reflection analysis and applications. *IPSJ Transactions on Computer Vision and Applications*, 5:1–18, 2013.

[88] Ko Nishino and Shree K Nayar. Corneal imaging system: Environment from eyes. *International Journal of Computer Vision*, 70(1):23–40, 2006.

[89] Christian Nitschke, Atsushi Nakazawa, and Haruo Takemura. Display-camera calibration using eye reflections and geometry constraints. *Computer Vision and Image Understanding*, 115(6):835–853, 2011.

[90] Christian Nitschke and Atsushi Nakazawa. Super-resolution from corneal images. In *BMVC*, pages 1–12, 2012.

[91] Rob Jenkins and Christie Kerr. Identifiable images of bystanders extracted from corneal reflections. *PloS one*, 8(12):e83325, 2013.

[92] Amit Agrawal, Yuichi Taguchi, and Srikumar Ramalingam. Beyond alhazen's problem: Analytical projection model for non-central catadioptric cameras with quadric mirrors. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 2993–3000. IEEE, 2011.

[93] Ko Nishino and Shree K Nayar. The world in an eye [eye image interpretation]. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 1, pages I–I. IEEE, 2004.

[94] Christian Nitschke, Atsushi Nakazawa, and Haruo Takemura. Image-based eye pose and reflection analysis for advanced interaction techniques and scene understanding. *Computer Vision and Image Media (CVIM)(Doctoral Theses Session)*, pages 1–16, 2011.

[95] KP Mashige. A review of corneal diameter, curvature and thickness values and influencing factors. *African Vision and Eye Health*, 72(4):185–194, 2013.

[96] Zhengyou Zhang. Flexible camera calibration by viewing a plane from unknown orientations. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 1, pages 666–673. Ieee, 1999.

[97] David Eberly. Computing a point of reflection on a sphere, 2008.

[98] Amit Agrawal, Yuichi Taguchi, and Srikumar Ramalingam. Analytical forward projection for axial non-central dioptric and catadioptric cameras. *Computer Vision–ECCV 2010*, pages 129–143, 2010.

[99] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–I. IEEE, 2001.

[100] Rainer Lienhart and Jochen Maydt. An extended set of haar-like features for rapid object detection. In *Image Processing. 2002. Proceedings. 2002 International Conference on*, volume 1, pages I–I. IEEE, 2002.

[101] Dirk Schnieders, Xingdou Fu, and Kwan-Yee K Wong. Reconstruction of display and eyes from a single image. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1442–1449. IEEE, 2010.

[102] Moritz Kassner, William Patera, and Andreas Bulling. Pupil: an open source platform for pervasive eye tracking and mobile gaze-based interaction. In *Proceedings of the 2014 ACM international joint conference on pervasive and ubiquitous computing: Adjunct publication*, pages 1151–1160. ACM, 2014.

[103] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *European conference on computer vision*, pages 430–443. Springer, 2006.

[104] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. Brief: Binary robust independent elementary features. In *European conference on computer vision*, pages 778–792. Springer, 2010.

[105] David G Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee, 1999.

[106] Andrew R Conn, Nicholas IM Gould, and Philippe L Toint. *Trust region methods*. SIAM, 2000.

[107] M.I. A. Lourakis and A.A. Argyros. SBA: A Software Package for Generic Sparse Bundle Adjustment. *ACM Trans. Math. Software*, 36(1):1–30, 2009.

[108] R Tyrrell Rockafellar and Roger J-B Wets. *Variational analysis*, volume 317. Springer Science & Business Media, 2009.

[109] Shree K. Nayar, Gurunandan Krishnan, Michael D. Grossberg, and Ramesh Raskar. Fast separation of direct and global components of a scene using high frequency illumination. In *ACM SIGGRAPH 2006 Papers*, SIGGRAPH '06, page 935–944, New York, NY, USA, 2006. Association for Computing Machinery.

[110] Y. Ohta and T. Kanade. Stereo by intra- and inter-scanline search using dynamic programming. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-7(2):139–154, March 1985.

[111] Jian Sun, Nan-Ning Zheng, and Heung-Yeung Shum. Stereo matching using belief propagation. *IEEE Transactions on pattern analysis and machine intelligence*, 25(7):787–800, 2003.

[112] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International journal of computer vision*, 47(1-3):7–42, 2002.

VITA

Chengyuan Lin is a Ph.D. student at Purdue University, and a member of Computer Graphics and Visualization Lab (CGVLAB). He received his B.S. from Zhejiang University, China.

His research interest lies in techniques and applications of augmented reality and diminished reality.