Universiteit Leiden

Master Computer Science

Optimizing Realism & Interaction in Augmented Reality Head-mounted Displays: A Study of Virtual-Physical Object Relationships using Projection Mapping methods

Name:	Raashid Khan
Student ID:	S2705745
Date:	January 2024
Specialisation:	Advanced Computing and Systems
1st supervisor:	Edwin van der Heide
2nd supervisor:	Daisuke Iwai

Master's Thesis in Computer Science

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Abstract

With recent advances in AR systems such as the Hololens 2 and Magic Leap 2, wearable and portable AR systems have become synonymous with immersive mixed-reality experiences. These systems however still find challenges in terms of real-time realistic and adaptive virtual content for Optical see-through head-mounted displays. In this research thesis, we developed a real-time rendering method to enhance virtual content quality by studying the computational algorithms used for Projection Mapping systems, and existing relationships with virtual objects, deriving new relationships and adapting a Deep Learning Convolutional Neural Network to estimate light direction and intensity. The methods used in our technique adapt to the environment's light and adjust the virtual content lighting and light intensity in an AR scene rendered using Hololens 2.

Contents

1 Introduction **4**

2 Related Research **6**

3 Fundamentals **8**
3.1 Optical See-through Head Mounted Displays (OST-HMD) 8
3.2 Projection Mapping and methods used for image processing 8
3.3 Deep learning Convolutional Neural Networks 9
3.4 Data Augmentation 9

4 Methodology **10**
4.1 Relationships review 10
4.2 Proposed relationships 11
4.3 Virtual Image Enhancement Method 11
4.3.1 Adapting radiometric compensation for OST-HMD 12
4.3.2 Deep learning-based lighting estimation 13
4.4 Object Tracking and Mapping Method 20
4.4.1 Markerless object tracking and model mapping 20
4.5 AR System Design for Proof of Concept(POC) 21
4.6 Implementation 22
4.6.1 Execution of the Deep Learning Network Model on HoloLens 2 23

5 Evaluation and Results **24**
5.1 Experiment Setup 24
5.2 Rendering 24
5.3 Light Direction and Intensity Estimation 25

6 Discussion **27**
6.1 Contribution 27
6.2 Dataset 27
6.3 Results 28
6.4 Limitations and Future Work 28

7 Conclusion **31**

1 Introduction

Augmented Reality(AR) has seen good advancements in recent times, with its definition extended to the type of content that is augmented. There are different types of AR, such as marker-based, markerless, location-based, or projection-based, that use different methods to track and display virtual content in the physical world. Different AR levels, such as augmentation, annotation, or manipulation, determine how much the virtual content affects or changes the physical world. By choosing the right type and level of AR, we can create AR experiences that are compatible, coherent, and consistent with the physical world and the user's goals[4]. Contextual experiences such as Mobile AR, Sonic AR, Spatial AR, Head-mounted display(HMD) Wearable AR and similar as seen in Figure 1 have become interesting research areas connected to immersive augmented experiences.

While most researchers have focused on improving technology in terms of hardware and software, much less work has been done in exploring the relationships between virtual and physical content in AR experiences. Here, Virtual refers to the content displayed using projections with a procam(projector-camera) system or HMD. In contrast, physical refers to the rigid or non-rigid objects/surfaces in the visible area.

Projection Mapping(PM) on the other hand is a form of Spatial AR which allows viewers to experience visual content without using an HMD. Some of the latest research trends on computational PM were identified by Iwai [12] suggesting recent cutting-edge technological advancements in PM. Advanced methods such as Dynamic PM and High-speed PM can also be used to optimise virtual content using computational projections similar to the work also presented by Iwai et al .[11] and [13] and hybrid systems such as the Hybrid Optical see-through head-mounted display with Spatial AR(HySAR) which is a combination of a projector(projection mapping) and an HMD calibrated together to render content in parallel[9].

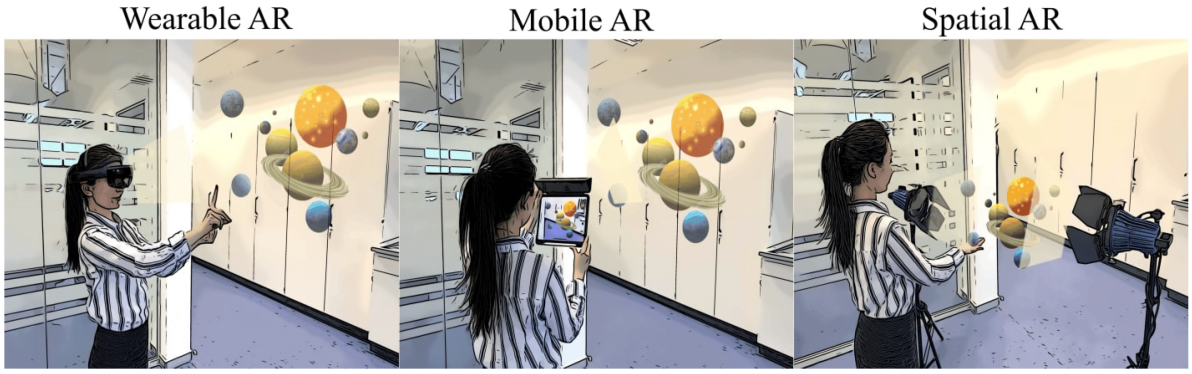

Figure 1: Illustration of AR experiences by Makhataeva et al.[17]

The need for optimizations has led to the investigation of more relationships between the virtual and physical content to optimize strategies to tackle the associated challenges such as alignment, integration, interaction, latency, spatial awareness, and object recognition all of which will be evaluated to see the impact of the new relationships identified. This brings us to our research question - how to optimize interaction and realism in AR-HMD for virtual content?

Hence, through this thesis study, we would like to identify relationships, methods and improvements that might answer this question while providing a novel contribution to the combined field of Augmented Reality(AR) and Projection Mapping(PM) strictly in the case of Optical see-through AR HMD(OST-HMD). Table 1 compares some important characteristics of three major AR types - SAR, OST-HMD and VST-HMD. There are various reasons why OST-HMD is a better choice for this study as compared to video see-through head-mounted display(VST-HMD) -

1. Real-world interactivity is severely limited in the case of VST-HMD due to distortions and lags from the camera feed.

2. OST-HMDs maintain a real-time view of the physical world through transparent optics whereas VST-HMD shows a digitized view of the same.

3. The Field of view(FOV) of the display is limited in VST-HMD by the camera's FOV compared to the flexible FOV(currently narrow) and peripheral vision of OST-HMDs. An interesting point to note here is that flexible FOV means, the FOV is only limited by the optical see-through display size. However, images can also be projected on larger displays.

Characteristic	SAR	OST-HMD	VST-HMD
Field of view	Wide	Narrow	Limited by Camera
Spatial resolution	Low	High	High
Viewpoint independent rendering	Not applicable	Suitable	Suitable
Maximum intensity	Low	High	High
Dynamic range	High	Low	High
Color space	Narrow	Narrow	Wide
Interaction	Not applicable	Yes	Yes

Table 1: Comparison of SAR, OST-HMD, and VST-HMD

Due to the aforementioned reasons, our research study aligns better with OST-HMD as the immersiveness and interaction on OST-HMD are well suited for optimizing realism for virtual content on AR HMD. The rest of the report is structured as follows: The related research section outlines what relatable research has been done in the field of Projection Mapping(PM). The fundamentals section contains definitions of the concepts used in the following sections. The methodology section outlines the algorithms that were used and the relationships explored and analysed. The experiments and results section shows the tests conducted and results obtained on the new findings. The discussion contains a review of the methods and analysis of the results, as well as possible future research in this area. Finally, the conclusion includes a summary of this thesis research and some concluding remarks.

2 Related Research

Within our research, we explored and studied several research papers that found factors impacting how virtual content is integrated into AR. Iwai et al.[9] classified virtual content into two components - view dependent(VD) and view independent(VI) when combining Projection mapping(Spatial AR) with Optical see-through HMD(OST-HMD) AR in their work for a Hybrid Spatial AR(HySAR) demonstrating a hybrid rendering engine equipped with parallel rendering paths for the improved material rendering of an object. These components are associated with the dichroic reflection model[19] where rendered images such as diffuse reflections are view-independent and specular reflections are view-dependent.

Some hardware enhancements to existing HMDs for enhancing virtual content have been done such as by Talukder et al.[22], by using a photo and electrical responsive liquid crystal smart dimmer for augmented reality displays to enhance the ambient contrast ratio for virtual content when ambient brightness increases. This requires placing the smart dimmer in front of the AR display which helps to control the incident background light. A balanced ambient contrast ratio improves relationships with physical objects such as realism where the virtual content appears to belong in the physical environment, depth perception where an accurate sense of depth enables interaction with virtual content intuitively and realistically, and visual consistency where consistent virtual content is more acceptable within the real world context.

In our research, since we aim to enhance AR content in HMDs by using PM(Spatial AR) methods, it's also essential to know the current methods used in PM. Since PM content tends to be highly real-time for its context, several research groups have studied and adapted radiometric compensation techniques. Bimber et al.[8] propose an algorithm that performs content adaptation and radiometric compensation in real-time, and reduces visual artifacts while preserving a maximum brightness and contrast implemented on a GPU. Radiometric compensation is a widely used method in Projection Mapping(Spatial AR) for colour blending and reproduction, also used by HySAR[9].

Colour blending is also essential to virtual content integration in AR for OST-HMDs. Langlotz et al.[15] propose the application of radiometric compensation in optical see-through HMDs to compensate for colour blending in real-time and with pixel accuracy by adding an additional beam splitter that helps capture an image of the environment through a camera as seen from the user's eye. Although their work is monoscopic, an important relationship - user perception of the physical objects has been utilized to improve the virtual content overlaid on the HMD display and can be extended for stereoscopic displays.

Other sophisticated projection techniques such as aligning the projection to the user's view and position can add more spatial perception of virtual content in AR HMDs. Byun et al. [3] used a pan-tilted procam (projection-camera) system where the actuated projector orients itself to match the user's view of the projected surface helping them to understand better the spatial relationship of virtual contents in an augmented scene.

Several research studies have also combined Spatial AR(PM) techniques with AR OST-HMDs like Hololens in industrial applications such as for Collaborative Robot Programming(Interactive Spatial AR with Head-Mounted Display, ISAR-HMD) by Bambusek et al.[1]. This approach

however relies on a projector which projects a user interface on a touch-enabled table along with Kinect sensors to provide interaction with virtual objects. Some works like Mobile Spatial Augmented Reality on Tangible objects (moSART) by Cortes et al.[5] utilizes the immersive, wide FOV(Field of view) benefits of Spatial AR by getting rid of the AR display and using a mobile head-mounted procam (projection-camera) system, however, this has a limitation of variety of virtual content like 3d objects that can be displayed as well as occlusion problems. By using PM techniques such as Spatial mapping and 3D reconstruction in AR HMDs for manipulating virtual content, we could eliminate the need for a projector. This would require extending the AR HMDs with sensors such as cameras, depth sensors, or LiDAR (light detection and ranging) to gather information about the objects' shape, position, and relationship in the real world.

Einbadi et al.[6] surveyed several deep neural models for light estimation in indoor and outdoor settings. We studied multiple indoor methods from the survey for both indoor scenes as well indoor scenes with spatially varying illumination, Gardner et al.[7] demonstrates a method that trains a DNN to predict the direction, distance, size, and colour of a pre-defined number of light sources from a single LDR input image. Another interesting approach by Kan et al.[14] outlines a deep learning-based CNN that can predict a dominant lighting direction using an RGB-D image as input. The network architecture provided can be trained with synthetic or real datasets.

Our study shares conceptual parallels with HySAR[9] and ISAR-HMD[1] in its fusion of Optical See-Through Head-Mounted Displays (OST-HMD) and Projection Mapping, alongside exploring the role of radiometric compensation in Augmented Reality (AR). Diverging from these works, our research unveils novel interactions between virtual elements and physical environments. We demonstrate how integrating deep neural networks for light estimation and object tracking enables the presentation of virtual content in AR OST-HMDs, eliminating the need for external projectors.

3 Fundamentals

3.1 Optical See-through Head Mounted Displays (OST-HMD)

Optical See-through HMD uses Liquid crystal on silicon (LCOS) [16] material to display images. The liquid crystal is sandwiched between a layer of glass and a silicon wafer. The silicon wafer's top metal layer has two key functions: First, it is a mirror that reflects the light, and second, the mirror's voltage drives the liquid crystal, twisting it to create an image. When the polarized light reflects from the mirror, the light can project through the optical system so the user can see the image.

3.2 Projection Mapping and methods used for image processing

1. **Projection Mapping**
   Projection mapping (PM) is a technique in which the system first scans a room with at least two cameras and then produces an internal map of the space. Based on this internal map, the system projects virtual content onto the physical surfaces/objects in the room to create desired effects.

2. **Radiometric Compensation**
   Radiometric compensation is a technique used to correct for variations in lighting conditions and ensure consistent visual appearance in augmented reality (AR) applications. In procam(projector-camera) systems such as for projection mapping, radiometric compensation is used to adjust the projected image to account for various factors affecting the quality and clarity of the displayed content. This may involve compensating for ambient lighting conditions, colour variations, and other environmental factors that can impact the visibility and legibility of the projected image. By applying radiometric compensation techniques, projection systems optimize the displayed content to ensure that it is accurately represented and easily viewable by the audience. In general terms, radiometric compensation involves adjusting pixel values in an image based on factors such as ambient light, sensor characteristics, and display properties. The goal is consistent brightness and colour perception across different lighting conditions.

   One common approach is to use a gain and offset correction. The corrected pixel value (P_c) can be expressed as:

   $$P_c = (P_i * gain) + offset$$

   where:

   - P_c is the corrected pixel value.
   - P_i is the original (uncorrected) pixel value.
   - $gain$ is the gain factor for radiometric compensation.
   - $offset$ is an offset value for additional correction.

   The values for gain and offset are typically determined through calibration procedures that involve measuring the system's response under various lighting conditions.

3. **Bidirectional reflectance distribution function(BRDF) Models**
   Bidirectional reflectance distribution function(BRDF) is a fundamental concept in computer graphics and vision. BRDF defines the way light is scattered at an idealized point on an opaque surface when illuminated from a particular incoming direction and viewed from a particular outgoing direction. It is usually represented as a mathematical function that takes incoming and outgoing light directions as input and outputs the ratio of outgoing light radiance to incoming light irradiance. BRDF is used as a key component in image rendering algorithms and is crucial for displaying realistic and accurate virtual content. Various BRDF models are developed to simulate the interaction of light and the choice of a specific BRDF model depends on the characteristics of the material being simulated and the desired level of realism in the rendered images. For example - *Phong reflectance model* also referred to as "Phong Shading" is a BRDF model used for specular reflections which provide plastic-like specularity.

3.3 Deep learning Convolutional Neural Networks

A Deep Learning Convolutional Neural Network (CNN) is a type of artificial neural network specialized for processing data that has a grid-like topology, such as images. It employs layers of convolutional operations, pooling, and often fully connected layers to automatically and adaptively learn spatial hierarchies of features from input data. CNNs are widely used in image and video recognition, image classification, medical image analysis, and other tasks involving visual data.

3.4 Data Augmentation

Data augmentation refers to the process of generating a diverse set of training data by manipulating the lighting conditions, camera positions, and orientations in a simulated 3D environment. Specifically, we vary the positions and intensities of point light and sunlight and alter the camera's position and orientation around a 3D object (a cube). This variety in lighting and viewpoint creates a rich dataset that helps the neural network learn to generalize better, making it more robust to varying real-world conditions. This technique is particularly valuable in deep learning for enhancing the model's ability to understand and interpret different lighting scenarios and camera perspectives.

4 Methodology

4.1 Relationships review

In this study, various relationships were reviewed between virtual and physical objects in AR through related literature on Projection mapping and AR HMDs. These relationships concisely define how an AR experience is built and executed with numerous essential components like cameras, projectors(in the case of Spatial AR like PM), sensors and computational algorithms collaboratively working to deliver a coherent output. Based on the PM approach for Spatial AR, some of the relationships that exist between virtual and physical objects are -

1. **Alignment and Integration**
   This involves aligning virtual content precisely with physical objects. The relationship here is seamless integration, where virtual elements appear as part of the physical environment. The quality of this alignment is critical to the effectiveness of the AR experience. To achieve correct alignment and integration, calibration of the virtual content concerning its projection environment is required.

2. **Spatial awareness**
   Spatial awareness dictates a precise understanding of the physical environment. This includes factors such as the dimensions of surfaces, angles, distances, and the overall 3D geometry of the space. High spatial awareness enhances the coherence, realism, and overall effectiveness of the AR experience.

3. **Contextual relevance**
   Contextual relevance defines the context for selecting virtual content that interacts harmoniously with physical objects. This includes considerations such as the thematic connection between the virtual and physical objects and the interaction of virtual content with physical objects/surfaces. This relationship outlines the significance of classifying virtual content that can visually pair with the physical surface.

AR OST-HMDs(Wearable AR) are conceptually the same in terms of projecting virtual content over physical objects/surfaces, they entail the following additional relationships -

1. **Object recognition and Tracking**
   AR HMDs employ object recognition and tracking to identify and track specific objects in the physical environment, this way they superimpose virtual content onto the identified physical objects. This relationship relies on computer vision algorithms using cameras and depth sensors to accurately measure object coordinates, geometry and distance with machine learning and pattern recognition to enable accurate virtual content overlay on such objects.

2. **Occlusion**
   Occlusion is a crucial aspect of creating realistic and immersive AR experiences, as it helps to blend virtual and physical objects seamlessly. When a virtual object is occluded by a physical object, it means that the virtual content is visually hidden or obscured by the real-world object. This creates a more convincing illusion that virtual objects exist in the same space as physical ones, enhancing the overall sense of depth and realism in the AR environment.

4.2 Proposed relationships

To further enhance the capabilities of AR HMDs image rendering, we propose the following novel relationships -

1. *Dynamic Lighting interaction*
   This aspect involves dynamically adjusting the virtual content's lighting properties to align with the physical environment's lighting conditions, including ambient light intensity, color temperature, and direction. Implementing an algorithm to modify virtual objects' shading, shadows, and highlights ensures they correspond with the real-world lighting, enhancing the realism of the augmented scene.

2. *Object reflectance/Material Matching*
   This relationship focuses on analyzing and matching the reflectance properties of physical objects in the environment with those of virtual objects. By measuring the specular and diffuse reflection components of physical surfaces, we can dynamically adjust the virtual object's reflectance properties. This approach aims to achieve a seamless blend of virtual and physical elements, contributing to a cohesive augmented reality experience.

3. *Color Distribution/Perceptual Realism*
   The goal here is to adapt the color grading of virtual objects to mirror the color temperature and white balance of the physical environment. By continuously analyzing the real-world scene's color distribution, we can dynamically adjust the virtual object's color palette and tones. This ensures consistency and harmony with the surrounding environment, thus enhancing perceptual realism in the AR experience.

4. *User Engagement and Interaction*
   Finally, We propose adding an interactive layer that enables users to engage with virtual objects within the physical world. This interaction can vary in form, such as interactive lighting effects that create visual cues and signals. The aim is to guide user interactions, fostering active participation and engagement within the augmented environment.

4.3 Virtual Image Enhancement Method

Initially, we aimed to adapt methods used in PM for OST-HMD systems. Several methods were studied and conceptualized that we can utilize to apply the proposed relationships. One such method uses the generalized radiometric compensation as shown in section 2 and captures reference images to calculate a gain and offset. These parameters can then be iteratively tuned by using an optimization algorithm like linear regression or histogram matching to minimize the difference between observed pixel values and reference values to obtain the desired image. However, we find that the iterative nature of this approach and pre-calibration time would be unsuitable for real-time dynamic rendering on AR HMDs.

Exploring further, we hypothesize adapting radiometric compensation utilized in Projection Mapping(PM) for OST-HMD displays. The following section describes, how we attempted its adaptation.

4.3.1 Adapting radiometric compensation for OST-HMD

As seen in Bimber et al.[2] for their generalized approach for radiometric compensation, the forward light transport equation can be adapted for AR OST-HMD systems as follows-

$$c_\lambda = T_\lambda^{-1} p_\lambda + e_\lambda \tag{1}$$

Here:

- c_λ is the compensated image on the AR headset display of *mxn* resolution.
- T_λ^{-1} is the pseudo inverse of the light transport matrix, which incorporates the characteristics of the AR HMD technology and how it transforms the input image.
- p_λ is the original image stored on the AR HMD.
- e_λ is the environmental light component to adjust for the display conditions.
- λ represents a single color channel

To calculate a compensated image(for each colour channel in RGB), we need to acquire the light transport matrix. This matrix allows the image of any scene to be expressed as a matrix-vector product. By construction, T takes into account all transport paths from light sources to camera pixels in procam systems. Therefore, if we know T, we can render the scene from the camera's viewpoint under any illumination condition with shadows, caustics and interreflections all included [O'Toole et al][20].

An issue we see is that the light transport matrix is captured for a static scene with a known amount of light sources. Common real-life scenarios for AR OST-HMDs however could vary due to the difference in light sources and the compensation for the projected image may not be accurate. An adaptive approach to this method would be to capture the light transport matrix in real-time, however depending on the resolution of the images, the light transport could take a sparsely large form, which can cause long computation delays for real-time rendering. Additionally, to compute the light transport matrix, we need the projection light pattern *p* matrix in equation 1. For a static scene, this can be pre-generated for calibration. On OST-HMDs, this is extremely complex for a dynamic lighting scene. Moreover, a direct rendering approach is generally employed in OST-HMD devices compared to the camera-captured image rendering in PM. In direct rendering, the virtual content is directly displayed in the display's field of view. In contrast, camera-captured image rendering where typically radiometric compensation is used involves images captured by a camera to compensate for projector characteristics and environmental factors.

As seen in Langlotz et al.[15], we could also utilize a real-time radiometric compensation that was applied specifically on a custom-built OST-HMD. The paper describes an approach to calculating perceived radiance R, the image scene by the user's eye-

$$R = t_B E + I F r_B \tag{2}$$

where the term $t_B E$ is the environment light transmitted through the beam splitter B which is part of the OHMD. The amount of transmitted light depends on the used beam splitter and its light-transmissive factor t_B (e.g., 0.5 for half-transparent mirrors). The term *I* describes

the radiance of the displayed image. The form factor F of the device describes the effects of varying image intensities across the entire display surface; for example, the projected brightness of a pixel falls off at the edge of the display due to vignetting. The reflected light depends on the reflective factor r_B of the beam splitter used as part of the OST-HMD. However, as these components play a significant role in evaluating the light transmission and interaction factors, the actual hardware components and their assembly are proprietary for most commercial AR HMDs like the one we intended to use for experiments, HoloLens. Since the specific details of the components are unknown, we could not adapt this approach within the scope of this research study.

4.3.2 Deep learning-based lighting estimation

Our study of PM methods revealed that accurate lighting estimation is crucial for realistic virtual object rendering. However, adapting traditional PM methods for real-time rendering in Augmented Reality Optical See-Through Head-Mounted Displays (AR OST-HMDs) proved complex and unsuitable for dynamic environments. Consequently, we explored an alternative approach employing a deep-learning neural network. AR OST-HMDs can leverage such a trained model to estimate lighting using inputs from their cameras and depth sensors. Given the requirement for image processing, convolutional neural networks(CNN) emerged as a fitting choice. After reviewing various methods and current research on model inference in AR systems, we identified the network model from Kan et al.[14] as particularly compatible with our AR OST-HMD, which provides real-time RGB-D image pairs for input. In contrast to PM methods, the neural network approach offers dynamic lighting adaptation in AR HMDs, making it more apt for real-time rendering applications.

Light direction and intensity estimation

Kan et al.[14] described their method to estimate a dominating light direction in terms of Euler angles $[\phi, \theta]$ using a neural network. We adapted this network with additional output layers to estimate a new light parameter- intensity(i) thereby extending the original model's capabilities. As described in the previous work, we closely follow the method to estimate relative Euler angles which are relative to the camera coordinate space and transformed to world space after estimation. For this, the method adds the direction of the camera also in terms of Euler angles in world space to the estimated light directions $[\phi, \theta]$. These computations are described by below equations:

$$\phi_l = \phi_c + \phi \tag{3}$$

$$\theta_l = \theta_c + \theta \tag{4}$$

Finally, the Euler representation of the dominant light direction is transformed into the vector representation *(x, y, z)* to be used for rendering.

Our new light intensity estimation by its nature is independent of such complexity. However, the intensity value needs to be transformed between two rendering systems Blender-our dataset source and Unity-our AR rendering system for which no conversion mapping exists as they use different interpretations of light intensity. Blender uses watts to depict power for light sources whereas intensity on Unity is used for brightness. Initially, we set the same values as the estimated values which need to be adapted experimentally. The additional load on

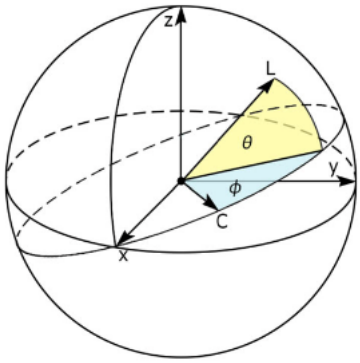

Figure 2: By Kan et al.[14] Relative Euler angles ϕ and θ of dominant light direction which is regressed by the neural network from an input RGB-D image. The angles ϕ and θ are relative to the camera pose C. L denotes a dominant light direction

the network to regress intensity values as well, requires us to train the network with diverse datasets where not only light and camera positions are randomized but also a variety of lights and intensities are incorporated into the training dataset. In the next sections, we describe how we adapted the Kan et al.[14] network in our research and our modifications to the network extending its capabilities.

1. *Network Architecture* The neural network for light source estimation in Kan et al.[14] uses residual blocks of convolutional layers to avoid the problem of vanishing or exploding gradients [He et al.][10]. These blocks use a shortcut connection from the beginning to the end of a block to let the network learn only a residual value from an original input. The shortcut connection and the result of a block are merged by an addition operation. The structure of the network for light source estimation is depicted in Figure 3. The network architecture begins with an input layer designed for images sized 160 $\times$ 120 with four channels, corresponding to RGB-D image data. This is followed by a convolutional layer featuring 64 kernels, each of size 7 $\times$ 7, which also employs strides to reduce the image size by half. Subsequently, a max pooling layer further halves the image resolution. The core of the network consists of 48 convolutional layers, organized into 16 residual blocks. These blocks are structured with an increasing number of kernels, as indicated by the dotted connections in Figure 3. Following the convolutional layers is an average pooling layer, leading into four fully connected layers that progressively decrease in neuron count. All layers utilize the ReLU activation function [Nair et al.][18], except for the final two dense layers. The network culminates in a layer that directly regresses the Relative Euler angles (ϕ and θ) for light direction.

   We conceptualized two network models by adapting and scaling the residual neural network from Kan et al.[14]. Our first network as shown in Figure 4 consists of an additional average pooling with three fully connected(fc) layers that regress the intensity value from the average pool output. These fully connected layers utilize the ReLu activation function except for the last dense layer. We hypothesise that the fully connected layers that are designed to capture high-level features from the pooled feature maps produced by the convolutional layers and an extra average pooling layer before the fully connected layers might help in reducing the dimensions of the feature maps and thereby computational load on the additional fc layers. By adding extra fully connected layers specifically

for intensity prediction, the network can learn a separate mapping from these high-level features to intensity values. This allows for better prediction accuracy for light intensity, which is a *scalar* value.

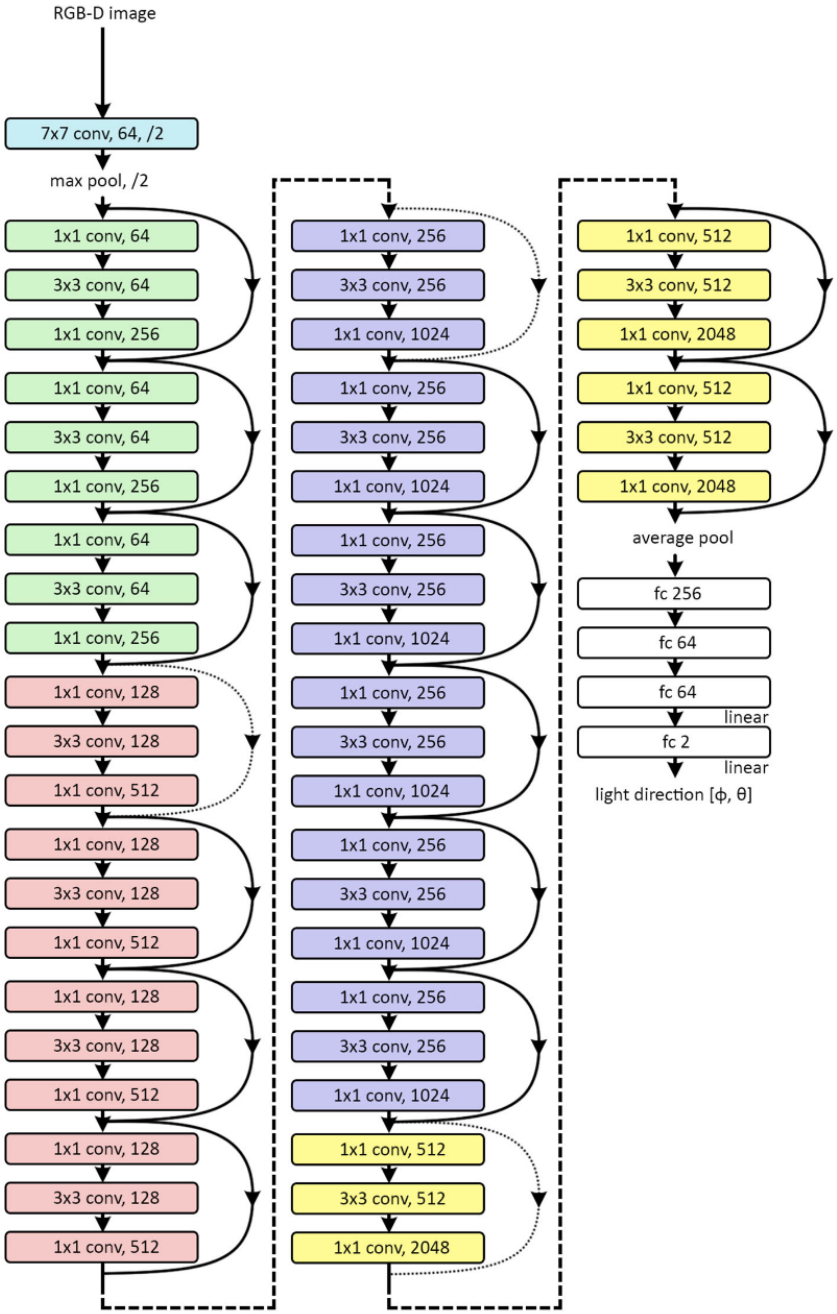

Figure 3: Deep learning residual network by Kan et al.[14]. Shortcuts for residual blocks [12] are indicated by curved arrows. Shortcut connections from the beginning to the end of blocks ensure that inner convolutional layers will compute a residual value. Dotted shortcuts mark the increase in dimensionality. Blocks with different dimensions are highlighted with different colors. All activation functions are ReLu except the last two layers which contain linear activations. Each layer indicates the size of a kernel for convolution as well as the number of kernels. Fully connected layers (fc) indicate the number of neurons

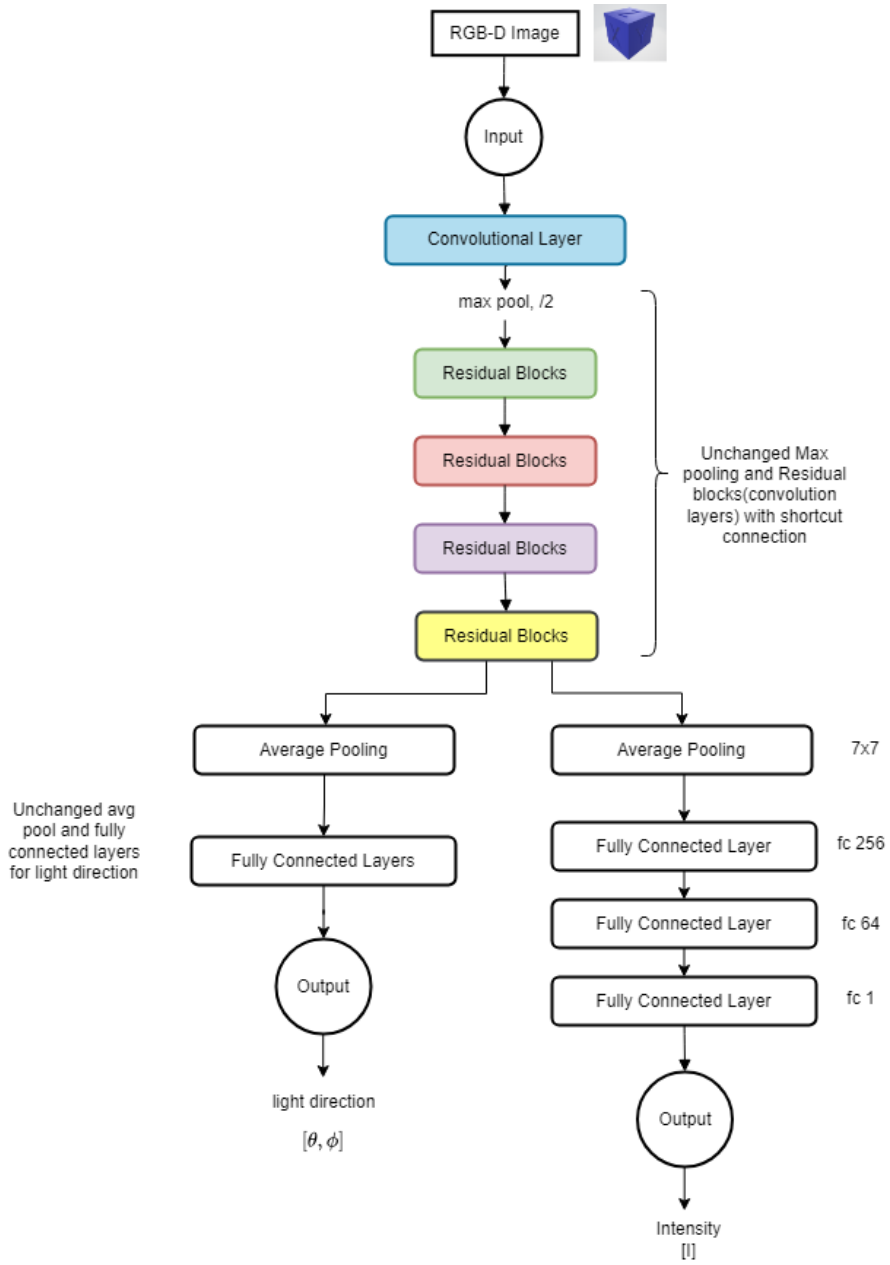

Figure 4: First network modified with only fully connected layers for intensity prediction

Our second network is built upon the first network and consists of two additional convolutional layers(64 in and out channels, 3x3 kernel filters) after the first input convolutional layer along with the average pool output layer that connects with the fully connected layers for intensity prediction. We hypothesise that additional convolutional layers can be used to extract more complex or abstract features from the input RGB-D data, which can be useful for understanding intricate patterns or details in the data for intensity correlation.

2. *Dataset* Training a deep neural network requires a large dataset with good diversity. We experimented with multiple data sources both real and synthetic. As the network is designed to take RGB-D images as input, we generated a synthetic dataset containing 10000 images(RGB and Depth pair) using Blender in a scene where we put a single shape- a cube on a plane which aligns with our testing scene described in section 4.6.

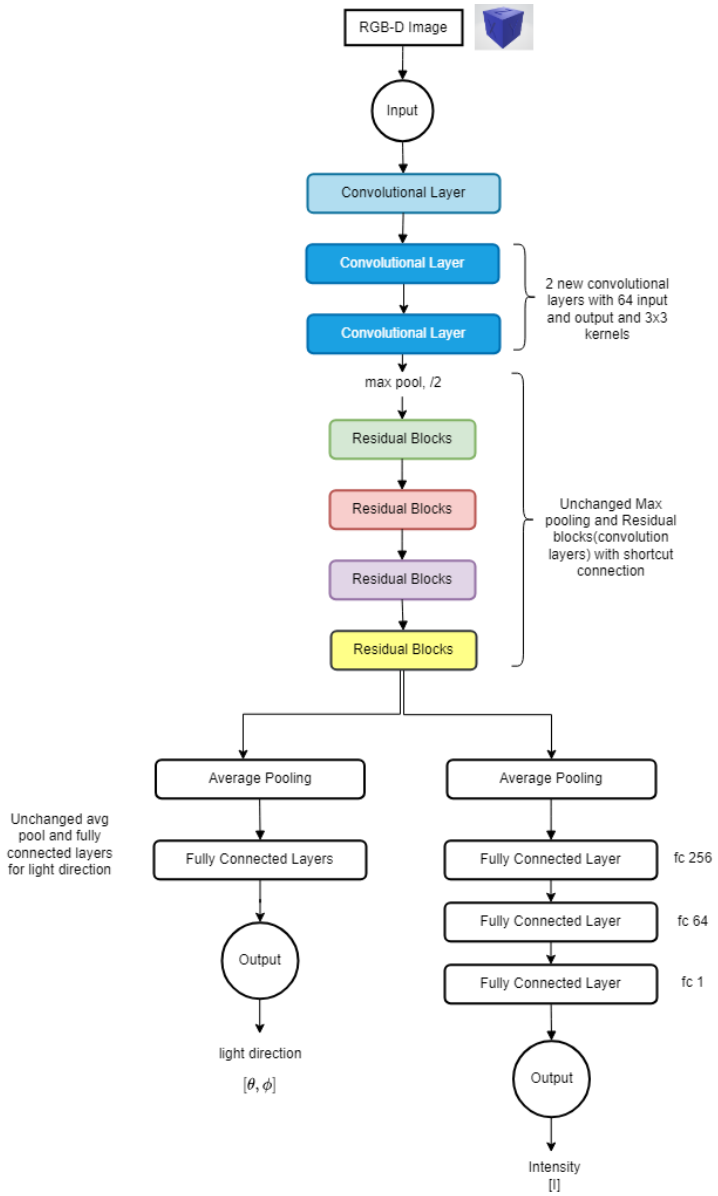

Figure 5: Second network modified with additional convolutional layers(64, 3x3) after the input convolution, average pooling and fully connected layers for intensity prediction

As we intend to experiment with physical cubes and virtual cubes, it is hypothesized that the network can learn patterns from this scene setup using high-level features like diffuse and specular reflections, shadows and image angles. Following data augmentation was necessary for our dataset for the network to learn efficiently as it is considerably smaller than the dataset used by Kan et al[14], hence in our script, we varied light types(point and sun), positions, color and rotations(for sun), and camera position and distance in our script for Blender. This provides us with different angles, shadows and light reflections on surfaces adding good diversity to the dataset. Our observations highlighted challenges in Blender, primarily due to platform limitations such as memory management. These constraints necessitated extensive development time to script a solution capable of generating large datasets efficiently. As an alternative to Blender, Unity was also explored to generate a similar dataset as shown in Figure 7. Configuring a

scene to simultaneously capture depth and RGB images presented challenges. This setup, involving two cameras with distinct rendering settings, proved inefficient in producing synchronized RGBD images, despite extensive testing with various settings.

Figure 6: Blender scene for synthetic image dataset generation

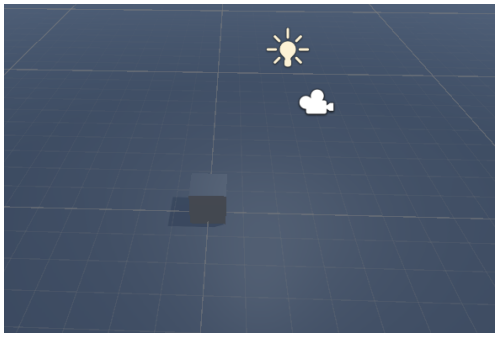

Figure 7: Unity Scene for synthetic image dataset generation

To train and test our network with real-world data, we attempted to capture RGBD images using a Kinect One. We emulated our Blender scene setup for data acquisition, utilizing the PyKinect2 Python library, as depicted in Figure 8. The captured data was validated by normalizing the depth information and integrating it with the RGB images. However, this approach yielded poor-quality images. Moreover, creating a high-quality, augmented dataset comparable to Blender's was time-consuming. Consequently, we opted against using this method and did not include a real dataset. Given the improved performance of the network in Kan et al.[14] with synthetic data, we are optimistic that our modifications and diverse dataset will enhance the network's accuracy in estimating light direction and intensity for our Proof of Concept (POC) scene.

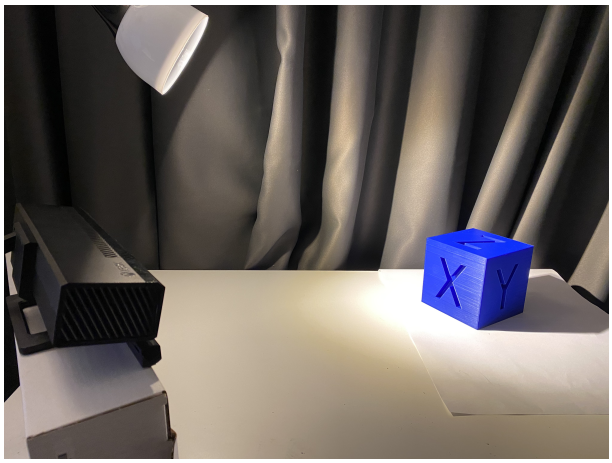

Figure 8: Kinect one setup for real image dataset generation

3. *Training* We use the same training technique as in Kan et al.[14] with some modifications and a few considerations based on experimentation. Going incrementally, we first trained the original network as a baseline to test our dataset, hyperparameters from the previous work and platform. Based on the findings from our training experiments and platform limitations, we adjusted the initial learning rate to 0.0001 while also using a learning

scheduler to optimize the learning rate every 10 epochs and a weight decay of 0.0001 that penalizes losses which helps in regularizing the model. Using weight decay encourages the network to maintain smaller weight values, leading to a simpler model which is less likely to fit the random noise in the training data and more likely to generalize well to new, unseen data. This is especially useful because we have a limited amount of training data, and it helps the model focus on learning the most important patterns. The epoch target was set to 50 but is dependent on the training if we see signs of overfitting or underfitting for the last 5 epochs. Empirically, we found that a 0.00005 learning rate worked best for our training dataset, model and platform and hence the model was trained to our epoch target. Both RGB and depth data were normalized with zero mean normalization that pre-processes the input dataset. A stochastic gradient descent optimizer was used to train the network which is generally employed for such image processing tasks. The order of input data for training was randomized. As deep neural networks require longer training times with high-quality datasets, we incorporated an early stopping mechanism to check for signs of overfitting. To achieve this, after every epoch, we calculate average validation losses and if the validation losses increase consistently for 5 epochs, we stop the training and save the model. The network was trained using CUDA on Nvidia GTX 1060. We evaluated the results of the network in terms of both mean squared error and the real-time AR application on HoloLens 2 to assess applicability to the light source and intensity estimation in real conditions.

Finally, we chose the first network based on our training and testing experiments. We observed that the second network did not perform well during training and showed signs of early overfitting with high training and validation losses. With this experiment, we learned that deeper neural networks require even larger datasets and longer training times to perform with better accuracy. However, with adequate data and training optimization, we hypothesise that the second network could perform better than our first network given its design which can be explored in future research. Figure 9 shows training and validation losses from the training of the first network. Epochs 1-10 were trained with a learning rate of 0.0001 and Epochs 11-50 were trained with 0.00005 which led to better training and validation losses over the full course of training.

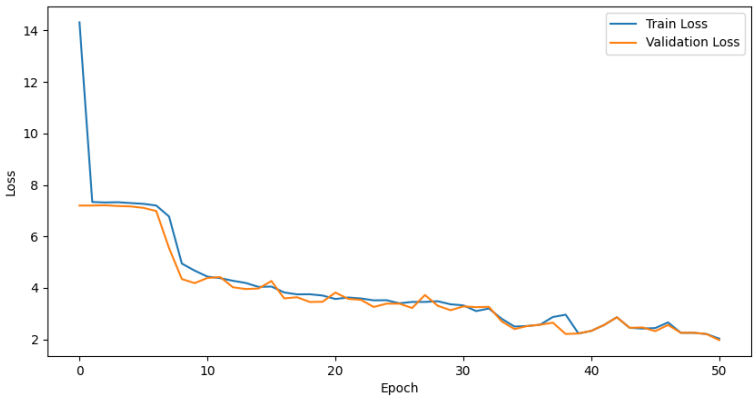

Figure 9: Training vs Validation Losses for the first network

4. *Temporal Coherence* We follow the same temporal coherence methodology as in Kan et al.[14]. Testing their effectiveness allows us to transfer proven methodologies to other AR systems such as OST-HMDs. We initially set the number of light estimations N to 4 for the outlier filtering which can be updated experimentally. The smoothing is performed once we have 4 light estimates and use the average of the inliers with a smoothing and neighbourhood threshold of 0.1 for both the first and second derivatives. These are described in the equations below-

$$\frac{\partial l(i)}{\partial t} \approx l(i) - l(i-1) \tag{5}$$

$$\frac{\partial^2 l(i)}{\partial t^2} \approx \frac{\partial l(i)}{\partial t} - \frac{\partial l(i-1)}{\partial t} \tag{6}$$

5. *Rendering* As shown in Figure 11, we used HoloLens 2 and its onboard main camera and depth sensor to capture RGB and depth images respectively while its display shows the virtual objects with no scene light adjustments. Synchronized RGBD images are then used to infer from the trained model, which after obtaining multiple light estimates only from synchronized images, we apply smoothing and add relative camera pose to transform the estimated light direction in world space. The single dominant directional light in the scene is then updated with the resulting rotation and intensity. This process however does not occur in real time due to the limitations of RGBD synchronization and outlier filtering. If the user does not change position and angles too fast, we can see a moderate adaptation of the light from these estimated values. The virtual object consistently appears in the scene while the light estimation is being processed every frame in the background.

4.4 Object Tracking and Mapping Method

Markers such as QR codes have already been used in Kan et al.[14] and similar previous works such as HoloLensARToolkit[Qian et al.][21] for AR OST-HMDs to track virtual objects as well as the RGBD camera. As we aim to obtain a realistic rendered scene as in PM, we explored a markerless object-tracking technique. Also, this helps us design a scene as realistically as possible without markers and within the scope and purpose of this research.

4.4.1 Markerless object tracking and model mapping

We experimented with different markerless object tracking such as OpenCV for Unity and Vuforia framework and decided to use Vuforia for our object tracking and model mapping as it is publicly available and integrates well with Unity while performing better than Open CV object tracking especially well suited for AR usecase. The model in this context should not be confused with the neural network component of our POC application. Here, the model is a 3D shape that will be used as a virtual object showing interaction with different light directions and intensities. Below we describe how we approach object tracking and mapping -

1. *Model* We generated a 3D cube model with dice faces using a combination of Blender and Inkscape tools which are publically available and give us models in Unity-compatible format. This model as shown in Figure 10 was later imported into Unity and integrated with scripts that use the Vuforia framework.

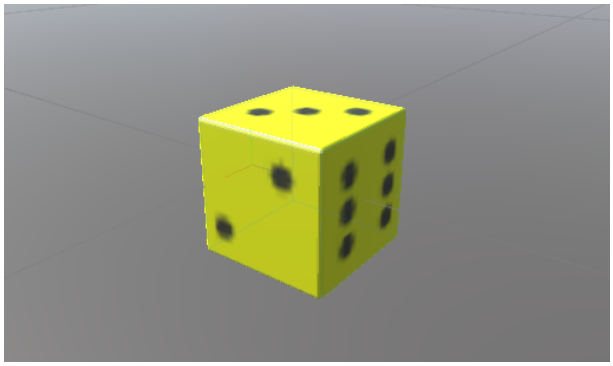

Figure 10: Model of the virtual cube in the form of a Dice

2. *Training* Using Vuforia as our markerless tracker, we explored two options for 3D object tracking training- standard model targets and advanced model targets. Standard model targets use guide views which are essentially a mesh representation of the 3D shape and do not require to train object detection. However, they suffer from long first detection, and low stability and do not work for continuous model tracking. In contrast, advanced model targets are trained with a 360-degree view of the shape. Both approaches require a 3D model CAD file. We used the same 3D model that was used for 3D printing a physical cube to train an advanced model target. We experimented with both object tracking and found advanced model targets to perform better in terms of detection, alignment and tracking.

3. *Integration* Once trained, the generated model database can be imported into Unity and configured with the means of a script that handles model target behaviour. The generated model acts as a parent object for the 3D virtual object that we want to render. During rendering the script tracks the detected physical object every frame and the 3D virtual object moves with the physical object. We observed fluctuation in tracking and detection and a reduction in frame rate with real-time tracking since the tracking is also performed with the same main camera that is being used to capture images for light estimation. We also observed that the training dataset can be better if a physical object model with dissimilar faces is used for training, a similarity in two or more faces, causes inaccurate detections, which can simply be improved by unique faces on each side of the physical cube.

4.5 AR System Design for Proof of Concept(POC)

As shown in Figure 11, our system running on HoloLens is centred around the collective usage of Light estimation and Object tracking and mapping modules. Both estimation and tracking module processes access the RGB and Depth sensor on the HoloLens in every frame of the rendered scene. As soon as these values are computed, we render the updated parameters of light while continuously tracking the physical cube which is superimposed with a virtual cube.

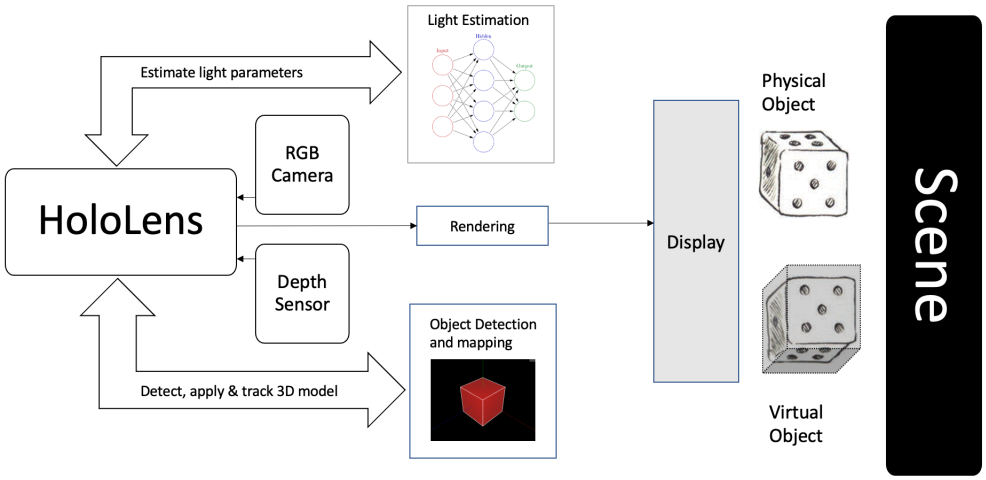

Figure 11: AR System Design for Proof of Concept(POC)

4.6 Implementation

The network was implemented using PyTorch in Python and subsequent training and testing was also performed in Python itself. We implemented our methods for RGB-D image capturing on HoloLens 2 using its research mode capabilities[23]. This required us to build a plugin in C++ that interfaces with the research mode APIs provided by Microsoft which access the on-board sensors of the HoloLens 2. This plugin was compiled into a DLL and then integrated with Unity. The light estimation and object mapping were written in CSharp using Unity(v2022) and Microsoft's Mixed Reality toolkit(v2.8.3) which is required to deploy applications on the HoloLens. Despite integrating error handling into our plugin extensively, the experimental nature of HoloLens 2 sensors led to unforeseen failures that could not be fully resolved.

RGB-D Frame Acquisiton
We capture RGBA and Depth frames for each Update call of Unity's mono behaviour scripting executing per frame. Additionally, we had to incorporate a timestamp technique to get corresponding RGB and depth images from the HoloLens sensors. The accuracy was empirically set to 100(milliseconds) of time difference as there is a considerable difference in the frame rates of the main RGB camera(30 fps) and the depth sensor(5 fps). We also observed a delay in depth frame acquisition from the C++ side to Unity's CSharp side of the implementation and hence our synchronization method prevents light estimation until we obtain a synchronized RGBD image input with an acceptable threshold. Once a synchronized frame is obtained, we scale down and adjust the depth frame(by default 320x288) for potential field of view(FOV) differences, and calculate corresponding pixel coordinates from the RGB image. These depth pixel values are then normalized and copied into a single RGBA texture by replacing the alpha channel values giving us a combined RGBD texture.

Scene
To test and demonstrate the results of light estimation, our POC(Proof of Concept) scene as shown in Figure11 consists of two cubes shown as dice - a physical and a virtual dice. The physical cube is used for reference to see how closely the virtual cube matches the physical cube in terms of light interaction and intensity. To accomplish this, we created a scene in Unity

with a single cube that will be rendered in an AR scenario superimposed over another physical cube(10cm) which was custom 3D printed. The superimposed physical cube can manipulate the virtual cube's position and orientation.

Shaders
Initially, we had explored writing a custom shader for rendering such as the Phong BRDF model for specular reflections but decided to use Unity's built-in shaders as the focus of our research was more on light estimation and mapping virtual objects and not specifically rendering.

4.6.1 Execution of the Deep Learning Network Model on HoloLens 2

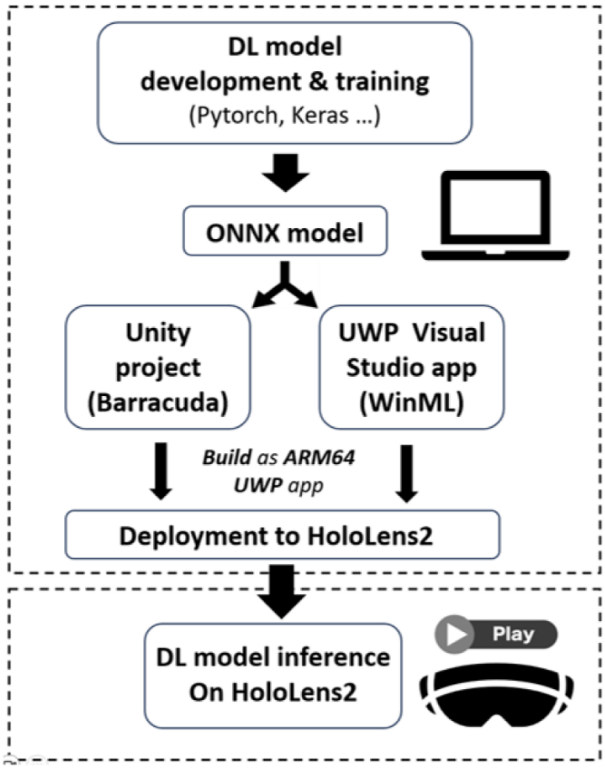

Figure 12: Overview of the integration of DL network model on HoloLens 2

Figure 12 shows the general integration pipeline for the deep learning neural network model on HoloLens 2 [Zaccardi et al.][24]. The pipeline works as follows - once the model is trained using Pytorch, the model is saved in a *pytorch* format which then needs to be converted to an ONNX model compatible with Unity and HoloLens. This model is integrated with Unity and Unity's Library Barracuda for deep learning model loading and inference, which is then built as a Universal Windows Platform(UWP) app with ARM64 architecture that can be deployed to HoloLens 2. On-device loading and inference are performed using Barracuda APIs by creating a tensor from the synchronized RGBD image acquired in real-time and using the tensor as an input to the model to get the expected output.

5 Evaluation and Results

5.1 Experiment Setup

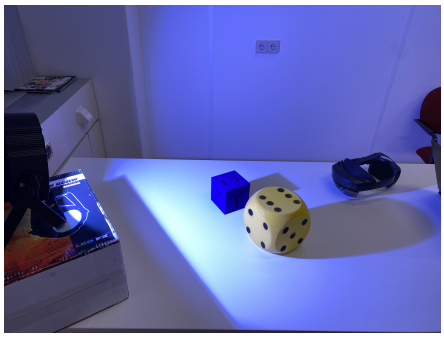

Figure 13: Setup 1- Light and object with Hololens

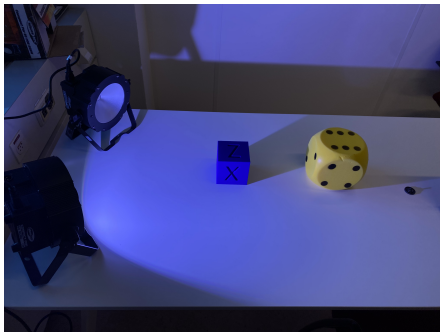

Figure 14: Setup 2- Varying light angles and intensity as well as object position

Shown in Figure 13 and 14 are our two different setups for running experiments. We set up two different light positions and object positions to test our light estimation and object tracking to evaluate how our light estimation and object tracking perform with different positions and angles.

5.2 Rendering

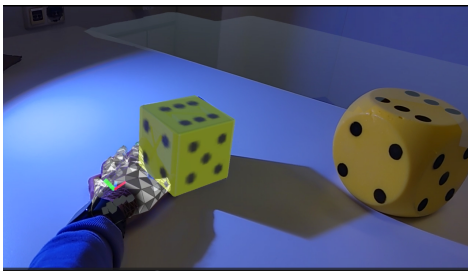

Figure 15: User interaction

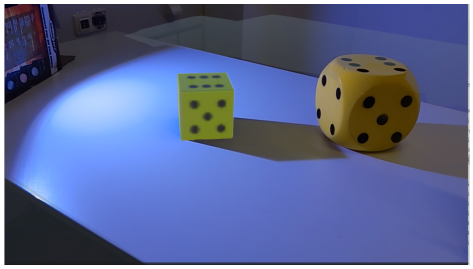

Figure 16: Light Interaction

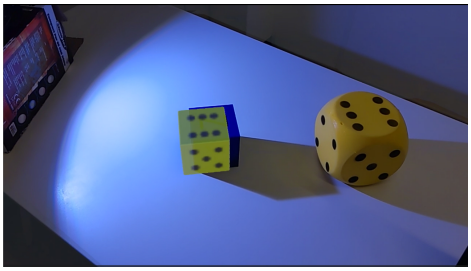

Figure 17: Realignment on orientation change

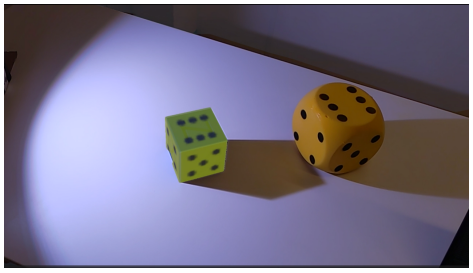

Figure 18: Interaction from different angle

In our initial evaluation, we focused on rendering the Proof of Concept (POC) Scene in an environment with a single dominant light source and ambient lighting. This was tested on the HoloLens 2, taking into account its computational, rendering, and sensor bandwidth

limitations. Our assessment involved two distinct modules: first for estimating light direction and intensity, and the second for object tracking, which aligns a virtual cube model with a physical blue cube. Figures 15 to 18 illustrate various aspects of user interaction, light interaction, and the system's response to different orientations and angles from our evaluation. Overall, the system performance was satisfactory under these conditions, with both modules functioning in tandem. Notably, Figure 17 highlights a minor misalignment issue, attributable to the added burden of simultaneous image capture and tracking on a single camera stream. We observed coherence in lighting updates with 4 frames.

5.3 Light Direction and Intensity Estimation

Light Direction Estimation We evaluate the accuracy of light direction estimation on a synthetic test dataset consisting of 100 images(RGB-D pairs). We measure this in terms of mean squared error for both the estimated values, and we also measure the angular error for the estimated light directions to the ground truth light directions.

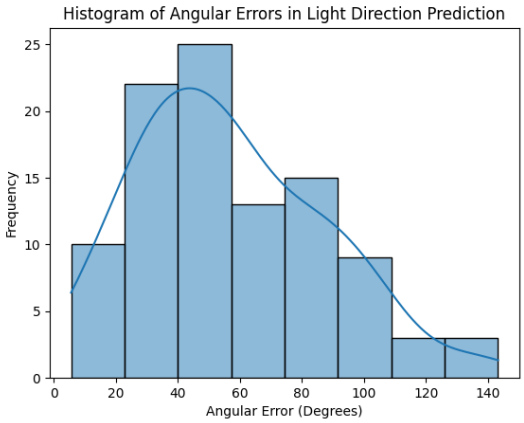

Figure 19: Histogram of angular error in light prediction to ground truth

Figure 19 illustrates the distribution of angular errors between the predicted and actual light directions in our test dataset. We observed a mean angular error of 56.97 degrees and a mean squared error of 0.34 on the synthetic test dataset(not seen by the model during training), with a range of errors that suggest variability in the model's predictive accuracy. The histogram exhibits a rightward skew, indicating that while the majority of predictions are close to the ground truth, as evidenced by the peak at lower error degrees, there are notable instances of large errors. Despite these outliers, the general trend indicates that the model often predicts light direction within a small angular deviation from the true values.

The breadth of the error distribution underscores that, although the model displays a reasonable level of accuracy overall, it does not consistently perform well across all test cases. This variability highlights opportunities for model refinement. Consistent with the properties of deep learning methodologies, our analysis suggests that the model's predictive capability could be enhanced by training on a more extensive and diverse dataset, potentially exceeding 25,000 images. Such an expansion of training data is likely to furnish the deep network with a richer set of features for learning, thereby improving accuracy and reducing the frequency of significant prediction errors.

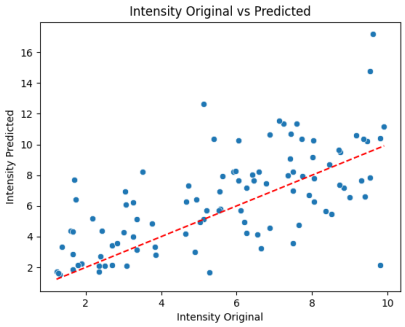

Figure 20: Scatter plot of Intensity prediction

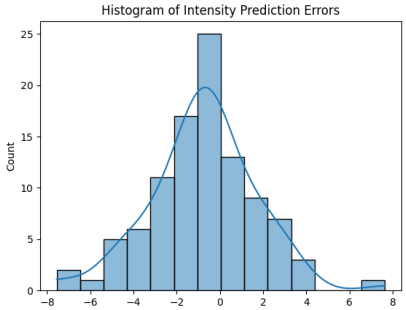

Figure 21: Error in Intensity Prediction

Intensity Prediction Similarly, in assessing the precision of our model's intensity estimations on the test dataset, a mean squared error (MSE) of 6.5 was recorded. As depicted in Figure 20, the scatter plot delineates the correlation between the actual and the predicted intensity values. Data points aligned with the red dashed line represent precise predictions, with proximity to this line reflecting higher accuracy. The dispersion of data points around this ideal line suggests a variability in prediction precision. Notably, while a significant number of predictions approximate the actual values, as indicated by their closeness to the line, there is a discernible dispersal, particularly at higher intensity levels. This dispersal signifies a diminution in prediction accuracy within this range. Conversely, the histogram of intensity prediction errors presented in Figure 21 exhibits a distribution that predominantly clusters around zero. This concentration near the origin implies that the model frequently forecasts intensity values with minimal deviation from the actual data. The distribution presents as nearly symmetrical but with a minor skew to the right, hinting at a modest subset of predictions where the errors exceed the average. The bell-shaped curve of the error distribution is reminiscent of a normal distribution, which typically denotes that the prediction errors are random rather than indicative of any inherent systematic bias within the model.

Together, these observations suggest that our model achieves a moderate level of accuracy in intensity prediction, with a tendency towards reliable estimations. However, the increased prediction error at higher intensities points to a potential area for model refinement.

6 Discussion

6.1 Contribution

We studied several pieces of literature on PM to identify methods that we could potentially adapt in our research for optimizing realism and interaction in AR. Despite not finding a method suitable for dynamic adaptation in this context, we recognized the strong potential of PM methods in AR OST-HMDs. Through the study of PM methods, we identified essential components and relationships of an augmented scene and how those components can be manipulated to create a realistic, interactive and immersive experience. As we could not apply PM radiometric compensation methods in the time frame for this research, we moved toward deep learning neural networks and identified a method that could be combined with our research and adapted with modifications required for our research goal. Our main contribution to realism and interaction in AR systems is a novel method which innovatively combines light direction and intensity estimation with markerless object tracking, enabling real-time rendering which is independent of an AR platform as long as we provide the necessary real-time input data(RGB-D image in our case). Hence our method can be applied to any AR OST-HMD system that can run a deep neural network inference in real-time realistic rendering while efficiently computing the related image post-processing. To the best of our knowledge and study of similar previous literature, a deep neural network that estimates light direction, as well as intensity, has not been combined with markerless object-tracking approaches for creating a realistic rendering system that gives us a similar experience as Projection Mapping while also adding interactivity.

6.2 Dataset

A crucial aspect of employing a deep neural network is its training with an appropriate dataset. In our study, we faced limitations due to the dataset's size and its focus on a single shape, intended to align with the Proof of Concept (POC) scene. This limitation led to less accurate model predictions, despite achieving better validation losses during training. We trained the model using a synthetic RGB-D dataset, which yielded satisfactory performance in both synthetic and real-world test scenarios. However, for enhanced accuracy, it's imperative to train the network with a more extensive and diverse dataset, encompassing both synthetic and real-world data.

Creating a high-quality synthetic dataset presented significant challenges, particularly when using tools like Blender and Unity, which impacted both the quality and variety of the data. In generating this dataset, our scripts were based on several assumptions about real-world geometry(cube on a plane), possibly leading to less variety in the dataset. This highlights the importance of pre-filtering the dataset through a robust quality pipeline, a process that demands an extended timeframe for data acquisition. In parallel, our efforts to capture a real dataset faced hurdles in pre-processing, a critical step to ensure the data's suitability for training. The time-intensive process of obtaining a diverse, high-quality real dataset, coupled with the absence of suitable open-source alternatives, limited our ability to incorporate such data within the constraints of our study's timeline. These experiences underscore the complexities involved in dataset preparation for deep learning applications, especially in the context of balancing synthetic and real-world data for enhanced model training and accuracy.

Despite these challenges, the model demonstrated moderate effectiveness, trained solely on a synthetic dataset. Our evaluation result trends indicate that training the network with a larger, high-quality dataset, potentially incorporating both synthetic and real data, could significantly refine the model and improve prediction accuracy.

6.3 Results

The outcomes of our Proof of Concept (POC) strongly align with the theoretical relationships outlined in section 4.2. Our deep learning neural network model excels in estimating light direction and intensity, a critical component for facilitating dynamic lighting interaction as per our proposed framework. This proficiency is integral to achieving perceptual realism in Augmented Reality (AR). It allows for the precise adaptation of virtual lighting to reflect real-world conditions, thereby ensuring that the virtual object's shading, shadows, and highlights are consistent with the physical environment's lighting. This alignment significantly improves how the object's highlights and colors are perceived through the AR Optical See-Through Head-Mounted Display (OST-HMD), echoing our intended relationship of color distribution and perceptual realism. Moreover, the ability of our approach to adapt virtual objects' material properties enhances object reflectance. This is evident in the seamless integration of virtual objects with their physical counterparts, ensuring that their specular and diffuse reflection components are in harmony.

Additionally, our object-tracking module plays a pivotal role in fostering user engagement and interaction. It creates a seamless bridge between the physical and virtual worlds, allowing for intuitive and meaningful interaction with virtual objects. This not only resonates with our proposed relationship of user engagement and interaction but also enriches the overall experience by introducing interactive elements and visual cues that encourage user participation in the augmented environment. Overall, the distinct modules of our POC collectively contribute to an immersive AR experience that closely adheres to our proposed relationships. This not only validates our theoretical framework but also demonstrates its practical applicability in enhancing physical-virtual object interaction, both in terms of visual congruence and user interactivity.

6.4 Limitations and Future Work

In developing a real-time light estimation and rendering system for this research, we faced numerous challenges and made strategic decisions. These choices, while not ideal in every aspect, were optimized to demonstrate the effectiveness of our approach. A significant limitation was our inability to implement a radiometric compensation method for AR Optical See-Through Head-Mounted Displays (OST-HMDs), a technique that would have brought us closer to integrating Projection Mapping methods in AR. This limitation was primarily due to the constraints of our chosen AR platform and the inherently static nature of conventional PM approaches. This experience highlights the distinct limitations and potential applications of Spatial AR(PM) versus Wearable AR(OST-HMD), underscoring the unique challenges in adapting technologies across different AR applications.

We decided to use one of the two proposed neural network architectures due to its suitability for the time frame of this research. As outlined in the Training section of the methodology,

the second network when trained with appropriate data could potentially perform much better than the first network adding more accurate light estimation for real-time rendering. Since we could not evaluate this network, we find that experimenting with the second network could be a good case for future research.

We performed transformation from camera space to world space in our prediction tests with synthetic test dataset, but we did not perform this transformation in our POC implementation for testing on HoloLens as our camera is aligned with the user's field of view and also since obtaining camera pose from HoloLens presented challenges that could not be resolved. However, by adding the relative camera Euler angles to the estimated light direction, a better accuracy in the light direction can be obtained.

Finally, our approach involved using RGB-D inputs for both training and evaluating our neural network, where depth values were integrated into the alpha channel of RGBA images. This methodology, while effective, presents opportunities for enhancement. One notable improvement would be the addition of a separate channel to the image input to incorporate surface normals. Integrating surface normals would provide the model with more detailed information about the geometry of the observed scene, potentially leading to more accurate light estimation and rendering. This additional data could help the network better understand the nuances of object surfaces and their interaction with light, thereby improving the overall quality and realism of the AR experience. Furthermore, integrating surface normals could aid in refining the model's ability to interpret complex scenes, particularly those with intricate geometrical features, enhancing the model's applicability and robustness in diverse real-world applications.

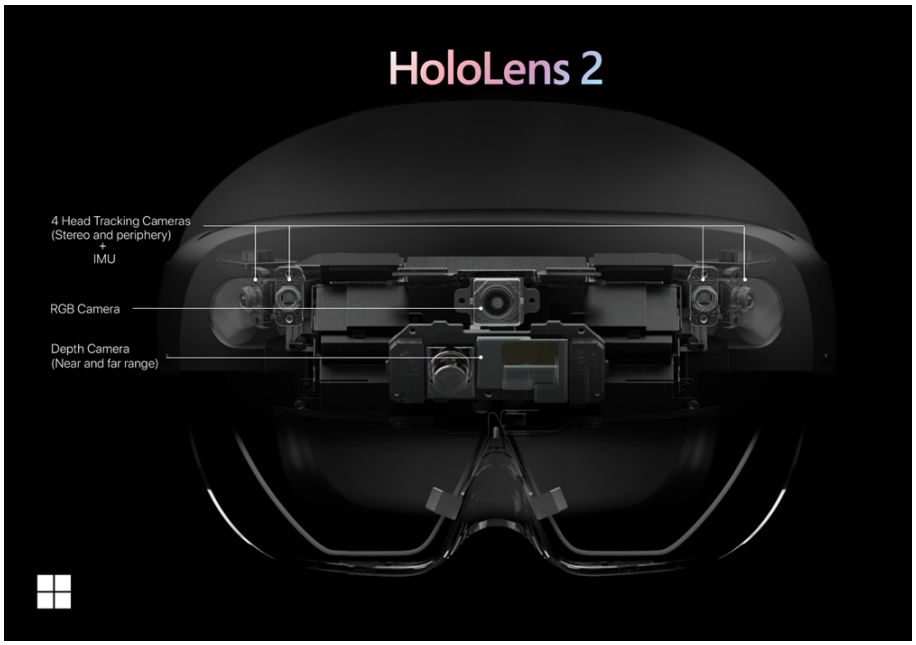

Figure 22: Hololens 2 front view by Microsoft

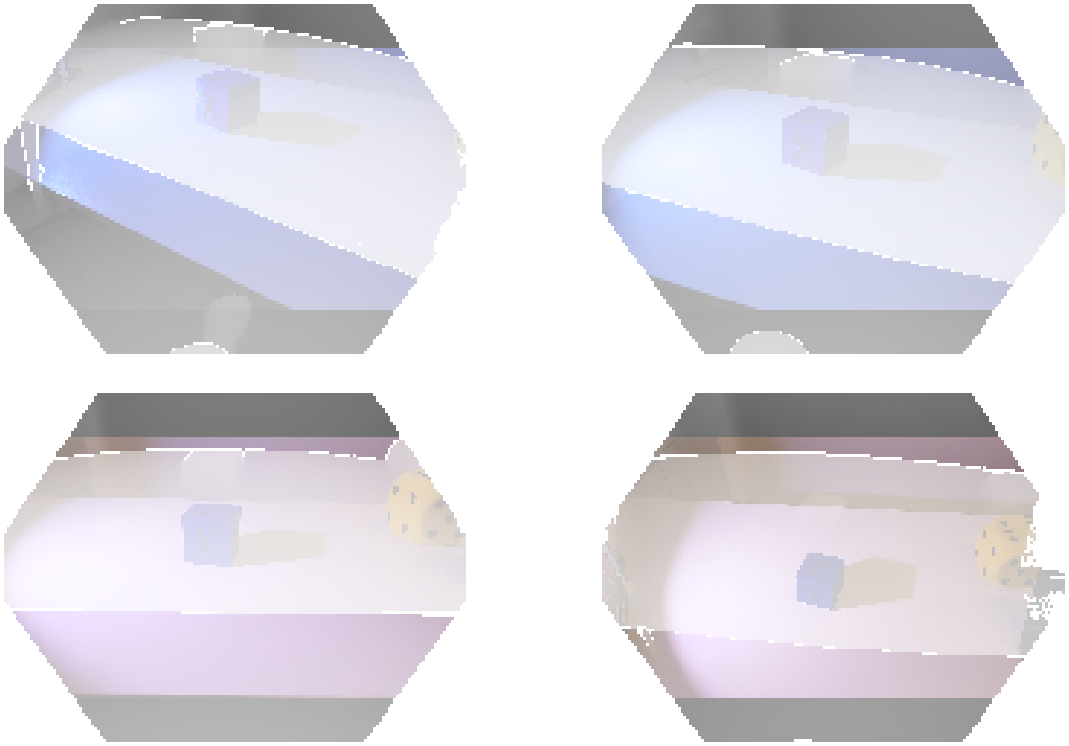

Figure 23: RGBD frames from Hololens 2

HoloLens Research Mode and Real-time RGB-D frame acquisition An AR platform is a crucial component of developing an approach for realistic rendering. A substantial effort was spent in this research to develop a plugin that can provide real-time depth image frames from the depth sensor on HoloLens using its research mode capabilities[23]. Currently, the depth sensor provides a hexagonal frame where its field of view does not fully align with the field of view of the Hololens RGB camera and the camera and sensor are located at an offset on the setup of HoloLens 2 shown in Figure 22. This limitation was unknown when we chose this platform as a testing AR system for our approach.

Acquiring an undistorted depth frame which is accurately synchronized with the frame capture from the RGB camera is currently not possible with HoloLens 2 due to the difference in their frame rates, field of view and offset between the sensors. However, complex approaches like using sensor poses to find sparse correspondences through back projection and reprojection might give better results. Figure 23 shows different frames when visualized as a PNG image normalized to RGB colour range(0-255). We implemented a timestamp technique to identify synchronized frames within a threshold, we did not find a frame less than 100 ms of difference between the frame acquisition from both sensors, this led to possibly less inaccurate representation of the real-world scene, and consequently less accurate predictions from the model for light estimation and intensity. Moreover, as the camera is shared between the RGBD frame acquisition and object tracking, the frame rate of the cameras due to shared usage is dropped which does not give us sufficient synchronized frames for real-time lighting changes in the rendered scene. Hence a moderately accurate result was obtained. If these limitations are addressed, our method could also potentially be further enhanced by utilizing inertia measurement units(IMU) like the accelerometer and gyroscope data to track user pose and apply this information while rendering the virtual objects.

7 Conclusion

The work undertaken in this project, which involved studying Projection mapping methods, adapting the Deep Light model for intensity prediction in augmented reality (AR) applications and its integration with an object tracking framework, presented a series of challenges and learning opportunities. Creating a specialized dataset and iteratively refining the model showed us the adaptability and complexity inherent in deep learning. A critical aspect of our success was the precise adjustment of the network architecture and the training of a 3D model for object tracking, both essential for improved real-time rendering performance. Our results demonstrate notable advancements in light direction estimation, intensity prediction, and object tracking. However, they also reveal the intricate balance required in the training of both Deep Convolutional Neural Networks and object-tracking models, highlighting the need for ongoing optimization and refinement. This project lays important groundwork for future research in AR, illustrating the potential and challenges of applying deep learning in specific, real-world scenarios. It is a significant step forward in the integration of complex computational models within the dynamic realm of AR.

Acknowledgements We would like to express our deepest gratitude to Edwin van der Heide, our first supervisor, for their invaluable guidance in navigating the complexities of our research question, their patience, and their expert advice throughout this research. We are also grateful to Daisuke Iwai from Osaka University for joining us as a second supervisor midway through our thesis. His guidance, constructive advice, and critical feedback have been instrumental in shaping our research. We sincerely thank the faculty and staff of LIACS, Leiden University, for their unwavering support and assistance. Our heartfelt thanks go to my colleague, Adel Qaddoumi, who inspired us to delve into the field of Extended Reality (VR/AR). His insights and assistance have significantly enhanced the quality of this work. We are equally thankful to the Leiden Learning and Innovation Center for their generous support in providing a HoloLens 2 device, which was crucial for our research. Finally, we wish to thank all those who have directly or indirectly contributed to the completion of this thesis.

References

[1] Daniel Bambuŝek et al. "Combining Interactive Spatial Augmented Reality with Head-Mounted Display for End-User Collaborative Robot Programming". In: *2019 28th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*. 2019, pp. 1–8. DOI: 10.1109/RO-MAN46459.2019.8956315.

[2] O. Bimber and G. Wetzstein. "Radiometric Compensation through Inverse Light Transport". In: *Computer Graphics and Applications, Pacific Conference on*. Los Alamitos, CA, USA: IEEE Computer Society, Sept. 2007, pp. 391–399. DOI: 10.1109/PG.2007.47. URL: https://doi.ieeecomputersociety.org/10.1109/PG.2007.47.

[3] Junghyun Byun and Tack-Don Han. "PPAP: Perspective Projection Augment Platform with Pan-Tilt Actuation for Improved Spatial Perception". en. In: *Sensors (Basel)* 19.12 (June 2019).

[4] LinkedIn community. *How do you design AR for spatial and contextual awareness and understanding?* URL: https://www.linkedin.com/advice/0/how-do-you-design-ar-spatial-contextual-awareness.

[5] Guillaume Cortes et al. "MoSART: Mobile Spatial Augmented Reality for 3D Interaction With Tangible Objects". In: *Frontiers in Robotics and AI* 5 (2018). ISSN: 2296-9144. DOI: 10.3389/frobt.2018.00093. URL: https://www.frontiersin.org/articles/10.3389/frobt.2018.00093.

[6] Farshad Einabadi, Jean-Yves Guillemaut, and Adrian Hilton. "Deep Neural Models for Illumination Estimation and Relighting: A Survey". In: *Computer Graphics Forum* 40.6 (2021), pp. 315–331. DOI: https://doi.org/10.1111/cgf.14283. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.14283. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.14283.

[7] Marc-André Gardner et al. "Learning to Predict Indoor Illumination from a Single Image". In: *CoRR* abs/1704.00090 (2017). arXiv: 1704.00090. URL: http://arxiv.org/abs/1704.00090.

[8] Anselm Grundhöfer and Oliver Bimber. "Real-Time Adaptive Radiometric Compensation". In: *ACM SIGGRAPH 2006 Research Posters*. SIGGRAPH '06. Boston, Massachusetts: Association for Computing Machinery, 2006, 56–es. ISBN: 1595933646. DOI: 10.1145/1179622.1179686. URL: https://doi.org/10.1145/1179622.1179686.

[9] Takumi Hamasaki et al. "HySAR: Hybrid Material Rendering by an Optical See-Through Head-Mounted Display with Spatial Augmented Reality Projection". In: *IEEE Transactions on Visualization and Computer Graphics* 24.4 (Apr. 2018), pp. 1457–1466. ISSN: 1077-2626. DOI: 10.1109/TVCG.2018.2793659. URL: https://doi.org/10.1109/TVCG.2018.2793659.

[10] Kaiming He et al. *Deep Residual Learning for Image Recognition*. 2015. arXiv: 1512.03385 [cs.CV].

[11] Daisuke Iwai. *Computational Projection Display for AR/VR*. 2017. URL: https://daisukeiwai.org/share/paper/Iwai_IMID17.pdf.

[12] Daisuke Iwai. "Latest Research Trends on Computational Projection Mapping". In: (2018).

[13] Daisuke Iwai, Yuta Itoh, and Parinya Punpongsanon. "Computational Augmented Reality Displays". In: *Proceedings of the 2018 ACM International Conference on Interactive Surfaces and Spaces*. ISS '18. Tokyo, Japan: Association for Computing Machinery, 2018, pp. 477–479. ISBN: 9781450356947. DOI: 10.1145/3279778.3279808. URL: https://doi.org/10.1145/3279778.3279808.

[14] Peter Kán and Hannes Kaufmann. "DeepLight: light source estimation for augmented reality using deep learning". In: *The Visual Computer* 35.6 (June 2019), pp. 873–883. ISSN: 1432-2315. DOI: 10.1007/s00371-019-01666-x. URL: https://doi.org/10.1007/s00371-019-01666-x.

[15] Tobias Langlotz, Matthew Cook, and Holger Regenbrecht. "Real-Time Radiometric Compensation for Optical See-Through Head Mounted Displays". In: *IEEE Transactions on Visualization and Computer Graphics* 22 (July 2016), pp. 1–1. DOI: 10.1109/TVCG.2016.2593781.

[16] Po King Li. "LCOS and AR/VR". In: *Information Display* 34.2 (2018), pp. 12–15. DOI: https://doi.org/10.1002/j.2637-496X.2018.tb01068.x. eprint: https://sid.onlinelibrary.wiley.com/doi/pdf/10.1002/j.2637-496X.2018.tb01068.x. URL: https://sid.onlinelibrary.wiley.com/doi/abs/10.1002/j.2637-496X.2018.tb01068.x.

[17] Zhanat Makhataeva and Huseyin Atakan Varol. "Augmented Reality for Robotics: A Review". In: *Robotics* 9.2 (2020). ISSN: 2218-6581. DOI: 10.3390/robotics9020021. URL: https://www.mdpi.com/2218-6581/9/2/21.

[18] Vinod Nair and Geoffrey E. Hinton. "Rectified Linear Units Improve Restricted Boltzmann Machines". In: *Proceedings of the 27th International Conference on International Conference on Machine Learning*. ICML'10. Haifa, Israel: Omnipress, 2010, pp. 807–814. ISBN: 9781605589077.

[19] Shree K. Nayar, Xi-Sheng Fang, and Terrance Boult. "Separation of Reflection Components Using Color and Polarization". In: *International Journal of Computer Vision* 21.3 (Feb. 1997), pp. 163–186. ISSN: 1573-1405. DOI: 10.1023/A:1007937815113. URL: https://doi.org/10.1023/A:1007937815113.

[20] Matthew O'Toole and Kiriakos N. Kutulakos. "Optical Computing for Fast Light Transport Analysis". In: *ACM Trans. Graph.* 29.6 (Dec. 2010). ISSN: 0730-0301. DOI: 10.1145/1882261.1866165. URL: https://doi.org/10.1145/1882261.1866165.

[21] Long Qian, Anton Deguet, and Peter Kazanzides. "ARssist: augmented reality on a head-mounted display for the first assistant in robotic surgery". In: *Healthcare technology letters* 5.5 (2018), pp. 194–200.

[22] Javed Rouf Talukder, Hung-Yuan Lin, and Shin-Tson Wu. "Photo- and electrical-responsive liquid crystal smart dimmer for augmented reality displays". In: *Opt. Express* 27.13 (June 2019), pp. 18169–18179. DOI: 10.1364/OE.27.018169. URL: https://opg.optica.org/oe/abstract.cfm?URI=oe-27-13-18169.

[23] Dorin Ungureanu et al. "HoloLens 2 Research Mode as a Tool for Computer Vision Research". In: *arXiv:2008.11239* (2020).

[24] Silvia Zaccardi et al. “On-Device Execution of Deep Learning Models on HoloLens2 for Real-Time Augmented Reality Medical Applications”. In: *Sensors* 23.21 (2023). ISSN: 1424-8220. DOI: 10.3390/s23218698. URL: https://www.mdpi.com/1424-8220/23/21/8698.