SEMARAK ILMU
PUBLISHING
202103268166(003316878-P)

Journal of Advanced Research in Applied Sciences and Engineering Technology

Journal homepage:
https://semarakilmu.com.my/journals/index.php/applied_sciences_eng_tech/index
ISSN: 2462-1943

Impact of Luminous on Augmented Reality Response Time

Arthi D[1,*], S Sivakumari[2]

[1] Department of Computer Science and Engineering School of Engineering, Avinashilingam Institute for Home Science and Higher Education for Women, Coimbatore, India

ARTICLE INFO

Article history:
Received 1 September 2023
Received in revised form 19 April 2024
Accepted 9 May 2024
Available online 25 July 2024

Keywords:
Augmented reality; target; overlay; deep learning; tracking; lighting estimation

ABSTRACT

The goal of augmented reality is to combine elements of the actual environment with digitally created ones. In order to get realistic results, it is necessary to solve complex computer vision challenges. Some examples of these tasks include monitoring genuine 3D objects and assessing the lighting conditions of a scene. In this brief work, we explain how deep learning may be used to handle these two difficult problems in a way that is both accurate and reliable. As a potential solution to this issue, we have come up with the idea of feeding the network not only the currently active frame but also an estimate of the object's posture based on the preceding timestep in the sequence. Because of this, the network is able to repair any faults that occurred throughout the closed loop tracking process. The creation of a synthetic frame of the tracked item allows for the acquisition of the feedback, which may be thought of as an estimate of the current object posture. As a result, our approach requires a 3D rendering of something along with instruction of the tracking device using this model. We think we're the first to use deep machine learning for 6-degrees-of-freedom (DOF) dynamic tracking of items, but we can't be sure. In both cases, the latest developments are achieved by training deep convolution neural network models on massive data sets.

1. Introduction

The Augmented Reality is the technology that augments the digital information over the real world while scanning the targets. The main component of augmented reality application for a good experience depends on effects of light conditions. The light conditions have a great effect on retrieval of augmented information. Two use cases have been used for the study on the light conditions. AR for product package in marketing. It explains the usage of the product, displays the catalogue and other information that is used for ease marketing and attracting customer. Next, use case is an AR for learning that explains the sub assembly of an Offset printing machine.

[*] *Corresponding author.*
E-mail address: arthi678901@gmail.com

https://doi.org/10.37934/araset.49.1.95107

2. Literature Review

Augmented Reality is a technology that superimposes information such as 2D/3D content, image, video, animations, etc. over the real world. Light intensity plays a vital role in marker detection and displaying virtual objects over the real world. The performance of the augmented reality applications in mobile or Optical See-Through Head mounted displays, mostly rely on the light luminous of the environment. Visual coherence paves a path in achieving augmented reality experience by superimposing the visual contents on the particular location, shape of the virtual object and the interaction between the real and virtual object. This Section gives the background study on significant of luminous for AR under the following three categories (i) the performance analysis of peripherals with mobiles and OST-HMDs, (ii) interactive augmentation and environment shading-based virtual objects and (iii) advancements in AR & VR displays.

2.1 Performance Analysis of Peripherals with Mobile and OST-HMD

A mobile based augmented reality peripheral (ARP) was developed to visualize the academic rooms such as laboratories, classrooms, departments, study rooms, corridor, seminar rooms, meeting rooms and administrative service rooms in 2D and 3D form with room plan as marker. The reliability performance of the mobile based augmented reality peripheral application was determined by performing various tests by setting various light intensity, distance between the marker object and the AR peripheral and occlusion. As the result of the reliability test, the ARP app was successful above 300 Lux, where_it failed below 95 lux. The distance was optimal at 60cm and occlusion was good at 40% occlusion of the marker area.

On the other hand, marker detection on wrinkled, wet and dry markers failed [1]. The effect of light outside on the perceived sharpness of white and black pictures on Optical See-Through Heads Mounting Displays. The study was conducted in a range of ambient light intensities, from zero to 20,000 lux. When exposed to illumination levels in excess of 10,000 lux, both inside and out, they experience a significant reduction in contrast. Virtual material presented in hues other than white is going to achieved decreased contrast ratios than when the display was displaying complete white or all black. Therefore, OST-HMDs need further development as well as refining for use in outdoor settings with more intense illumination [2]

2.2 Interactive Augmentation and Environment Shading Based Virtual Object

Utilizing ray-tracing to produce a variety of lifelike effects at interactivity or real-time frame rates, this excellent renderer as well as mixing technology for augmented reality is a game-changer. Effects like refraction, reflection, as well as caustics may be simulated using interactive ray-tracing technology in AR on a single PC. The visual consistency of virtual reality is improved by these renderer techniques. Render times for various effects at various sample rates were used as performance markers. It was determined how long it takes the computer to do various rendering tasks including photon emission, Kd-tree construction, and tracing rays with estimated density. Researchers claim that the technology can remain interactive even with sceneries including 855K rectangles, like the transparent smiling Buddha. As a result, the system is capable of producing interactive augmentation [3].

Furthermore, an augmented reality system that is capable of estimating the light luminous of the outdoor scene and applying light changes to the augmented object according to the outdoor luminous. Investigators have employed a shadow quantity approach, which incorporates the amount of light Volume to evaluate the environment's shading in the scene as well as an enhanced challenge is illuminated consequently, image-based lighting to simulate the various light sources that reflect the scene, and Phong obscuring to simulate the sun's shading, allowing shadows to interact with its surroundings. The shade is derived from an environment map that represents the most recent light shifts in the scene, based on approximated local brightness characteristics. This allows the technology to instantly refresh the illumination of a simulated reality-based item depending on a picture [4].

2.3 Advancement in AR and VR Display

Both virtual reality (VR) as well as additive reality (AR) is influencing the way we take in and respond to simulated data in many forms. Because of their ability to recreate the connections between CGI as well as the actual environment, they have attracted a lot of attention and work. In recent years, a number of cutting-edge designs have emerged, each with its own set of advantages. For example, tunable lenses may minimize VAC, polarizing materials can address space constraints, as well as freeform optics can expand the field of vision. Displays that combine VR and AR have significant scientific implications and several potential practical applications [5].

2.4 Research Problems

It is difficult to infer the lighting of a scene from a single image. picture pixel values depend on several factors, including the scene's shape and substances, the lighting, the camera used to capture the picture, as well as any post-processing that was performed on the photograph. It's a poorly phrased inverse issue to separate one of those factors from others. Because numerous of the components that the influence the picture's lighting is not immediately noticed in the picture (Figure 1), doing so from just a single, restricted field-of-view photograph is very challenging. First, by presuming that the scene's morphology (and/or reflectance qualities) is supplied (measured by depth sensors, reconstructed via other techniques, or marked by a user), as well as second, by enforcing powerful low-dimensional modeling upon the illumination, this issue is often dealt with.

2.5 Proposed Work

Our goal network's design and structure are described in the next paragraph. Here, we detail the steps used to create the instructional pictures' labels as well as render them such that they more closely reflect actuality. We then provide the loss measure utilized during superimposition-based training. A brief explanation of a demonstration wraps up the present part. The brightness of the surrounding light is a major factor in the evolution of virtual reality software. The time required to get overlay information from a virtual reality app varies depending on the circumstances of the lighting. The design process as well as technique for an app that uses augmented reality is broken out here.

3. Methodology

Figure 1 depicts the different stages involved in developing and testing the performance of AR applications. In brief, its deep neural network algorithm takes in two images as inputs: one showing the item displayed in its expected location (based on the previous time stamps in the footage series), and the other showing the object as it is now being viewed. The neural network provides a direct result from the six degrees of freedom (DOF) reflecting the posture transition among the two inputs (three for translation and three for rotations in Euler angles). We use a collection of computer-generated photographs of the item, based on its 3D hypothesis, to instruct the network of computers. The first stage is to determine why augmented reality apps are needed, or the prerequisites for constructing them. To create AR apps, a target picture and knowledge, the necessary software's and hardware must be identified. It's up to you to make the marking (target picture) as well as the layered understanding. Create the necessary augmented reality apps using marker and overlay data. Make sure the augmented reality software works. Put into use on intelligent mobile the augmented reality programs you have created as well as tested. Response times testing to ensure performance of the application.

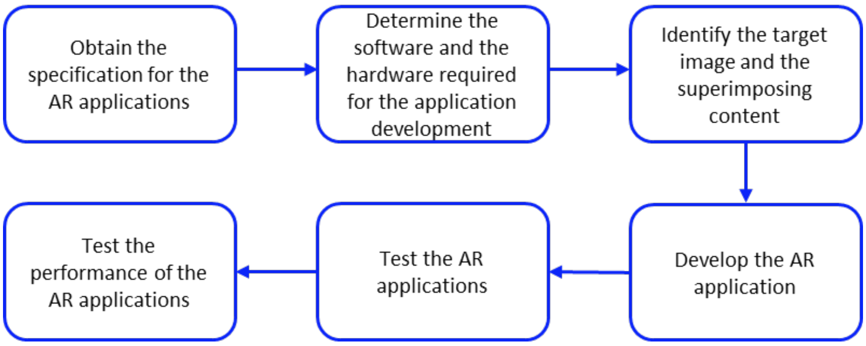

Obtain the specification for the AR applications
Determine the software and the hardware required for the application development
Identify the target image and the superimposing content
Develop the AR application
Test the AR applications
Test the performance of the AR applications

Fig. 1. Process diagram for performance testing of AR applications

3.1 Workflow Diagram

The workflow of the application performance is shown in Figure 2. The development of the application starts with

i. The creation of the target image.
ii. The target image created is both print and digital version.
iii. The target image is taken with different resolution high, medium and low respectively.
iv. The next step is creation of overlay insight.
v. The overlay insight created is image and video for the created targets.
vi. The AR applications are developed using Unity Game Engine and Vuforia SDK.
vii. Test the application for its proper functionality.
viii. Deploy the tested application software in the smart mobile.
ix. Obtain the performance test result by measuring the response time of the superimposing insight with different scanning light intensity between 46 lux and 423 lux and distance between the marker (15cm) and the scanning device.
x. Plot graph according to the response time observed and obtain the result.

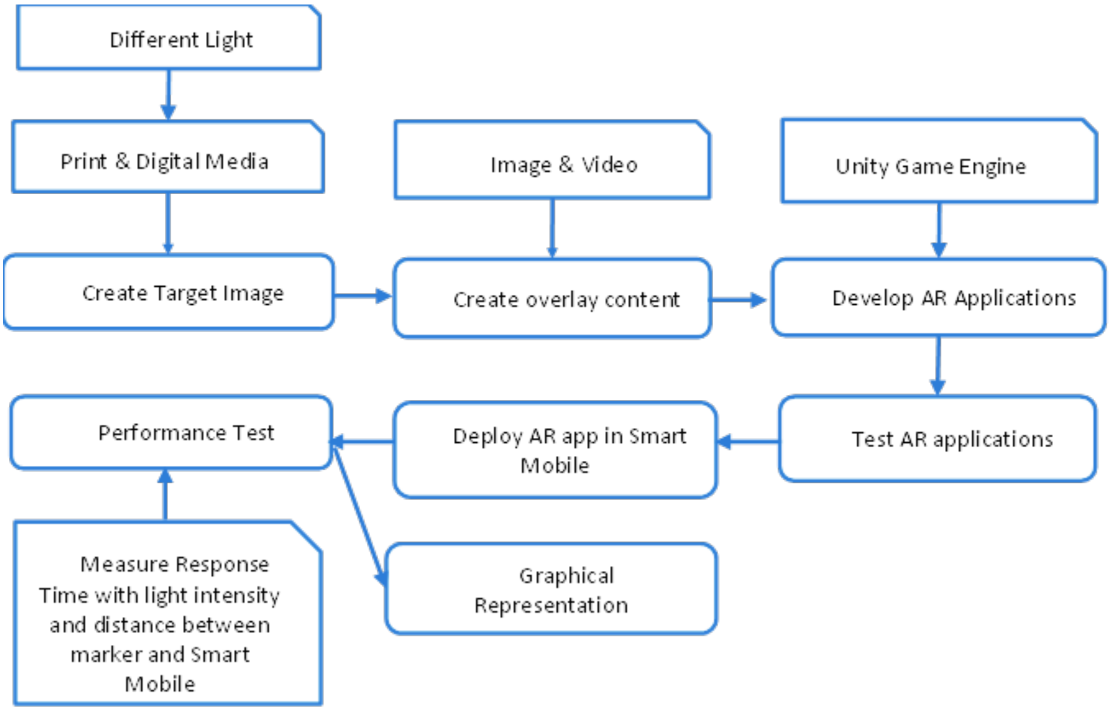


Fig. 2. Workflow of performance testing of AR applications

3.2 CNN with 6DOF

Specifically, we utilize a network of convolutional neural networks that, given a photograph as inputs, generates a low-dimensional representation of the input through a sequence of convolutions downriver, and subsequently divides into two upfront extensions to perform, respectively: (1) light intensity estimates and (2) RGB panoramic predictions. We use a big dataset consisting of panorama to train the algorithm. To feed a neural network, we first extract a regular picture from a panorama, presuming a viewpoint camera algorithm with sampled at random variables. Compared to current state-of-the-art techniques, our real-time historical 6-DOF object tracking approach is more resilient to occlusions (Figure 3). By recasting 6-DOF monitoring as a challenge in deep learning, we have made an important contribution.

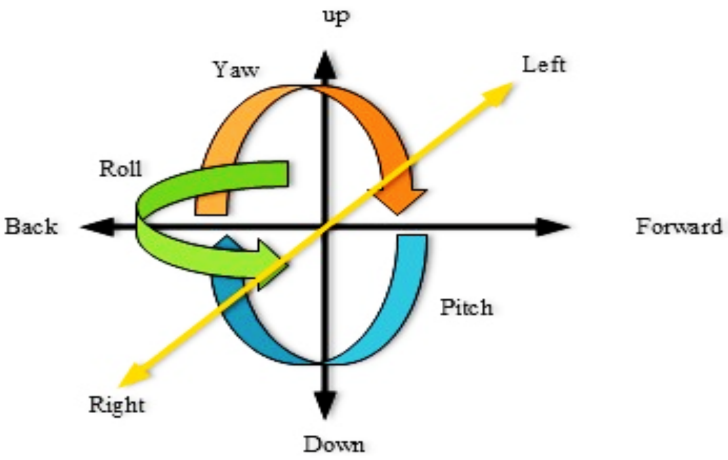


Fig. 3. DOF tracking in AR

There are three main advantages provided by this work. First, deep learning architectures are capable of being trained on massive volumes of information, making them resistant to common capture flaws including color and light variations, blurry images, and obscured subjects. Also, with a compact enough network, their GPU implementations are so efficient that they can be executed in real time on mobile GPUs. Finally, and most crucially, object-specific characteristics may be autonomously learnt from data, thus no hand-designed properties needed have been calculated. This differs from the majority of prior work, where computed predetermined characteristics by hand. It's not easy to implement a sophisticated convolutional neural network (CNN) for detection. Actually, the temporal trackers employ two images (pictures) consecutive in time. It also presupposes being aware of the thing's posture at the previous picture, which distinguishes it from detection-based tracking. One need only take the current and prior frames into account while training a deep network for that purpose.

Besides that, this approach achieves minimal errors in prediction on a "traditional" neural network test population (consisting of frame pairs as input and stiff posture changes as tar-get), it is incapable of keeping up with multi-frame sequence. Since a system has never been taught to self-correct, even seemingly little mistakes eventually add up. quickly, and after a few moments, it becomes impossible to keep track of it. As an alternative, you may provide the system with your best guess at the stance shift from the preceding frames. High errors in tracking result because this data is insufficient to teach a network accurate large-level representation. To address this issue, we suggest feeding the network not just the present frame but also a projection of an object's attitude retrieved from the preceding timestep in the series. Because of this, a network can fix closed-loop tracking mistakes. To receive a response or approximation of the present moment position of an object, an artificial picture of the monitored item is rendered. As a result, our method necessitates an a priori 3D rendering for every item, as well as the sensor is taught on a single kind of object. For 6-degrees-of-freedom (DOF) temporal object tracking, we believe that we're the initial to apply deep learning.

Our approach would not work without the CNN. Despite caring about the construction, variables, or hyperparameters to it's possible to be taught on generated picture information about an item. Inception-ResNet-v2 [6] serves as the foundation for our application. It takes features from both the Inception net and Resnet, such as consecutive as well as concurrent convolutional layers of information, and merges them. This facilitates gradient propagation across the network, which in turn makes deep networks amenable to training. With its improved accuracy over the Inception-v4 the network, InceptionResNet-v2 became available for classification assignments involving 1000 different types of data.

To make it work for this regression issue, we overlay a fully interconnected MLP (multilayer perceptron) blocks with only its characteristic extraction (see Figure 2). The weights learned during object recognition training on the dataset provided by ImageNet are used as the starting point for the model. This significantly shortens the time it takes to get a handle on our issue, since certain filters might need to be retrained and some may be modified from the ground up to fit it.

The dropout layer comes after the attribute extraction. Every fourth neuron in the aforementioned layer is turned off during training, for a total dropout rate of 0.25. Five completely linked layers come after a dropout level. For 6-dof posture estimation, they cut the total amount of neurons needed from a thousand to only six. The layers have been adjusted in batches. As activation operate, we utilize Parametric Rectified Linear Units (PReLUs). HeNormal applied the recommended weights. Trainer confluence is improved over ReLU2 as well as is achieved more quickly.

Table 1 summarizes the finished network's characteristics. There are fewer variables in this network compared to AlexNet (60 million) as well as VGG16 (138 million). It takes up 654MB of storage space on disc and is formatted as HDF5 (Hierarchical Information Format).

Seventy percent of the information is used for education, twenty percent for validation, and ten percent for testing. We conduct an assessment on fake information using a portion from the 10% test data. Completely fresh sets of information are employed for assessments on real-world photographs.

A generator coordinates the use of the training data. With a batch size of 32 photos each epoch, the algorithm loads at random pictures as well as the corresponding descriptions. There is no set protocol for the sequence of training images. However, we may infer that it will follow a normal distribution during training owing to the randomized nature of the data. In addition, all batch photos undergo standardization along with information augmentation through the program.

Table 1
Network properties

Property	Amount
Dimension of input	[299,299,3]
Dimension of output	[6]
films (total)	781
films (convolutional)	244
films (fully connected)	5
strainers	76032
factors (total)	56986462
factors (trainable)	56921698
factors (non-trainable)	64764

3.3 Camera Pose Estimation

The c-camera locations must be uniformly spaced out from the central O. The zero-mean stance results in a 50/50 chance across all dimensions. This issue may be addressed with a predictable, concave polyhedrons (such as an icosahedron). So, using the same approach as in [7], we decompose a polyhedron until the number of vertices is equal to the number of spots for the cameras we need. Our need for uniformly spaced camera vantage points is met by projecting those coordinates to the sphere's rim. Each point's angle to the central point O is arbitrarily chosen between 0 as well as S, covering S's whole volume $[r_{\min}, r_{\max}]$.

Roll, pitch, as well as yaw orientations might be used to describe the camera's perspective $(\phi_O, \theta_O, \psi_O)$ in terms of the coordinate system of the O-space we're working in. Complete reproduction is what we mean here. As the angular velocity of rotation around a point $0° \leq \alpha \leq 360°$ minor shifts in direction may have major effects on estimating, and classroom instruction can be hampered by the abrupt leap from 0 to 360°. Because of this, we use a relational diagram $(\phi_C, \theta_C, \psi_C)$ in order to become oriented.

We start by pointing the lens to the exact middle of the globe, O, which is given by the coordinates c. In addition, we set the sensor's upward vector such that it faces in an angle defined by the up vector or the thing being observed multiplied by the camera's observing orientation. This defines an initial camera orientation R_C relative to a local reference frame C. With the help of our relative roll, diameter, as well as yaw angles as well as the local reference frame C, it's possible $(\phi_C, \theta_C, \psi_C)$ to settle on a fixed camera angle. Consequently, the whole rotation matrix is $R = R_C \cdot R_{(\phi_C, \theta_C, \psi_C)}$. We limit the roll angle ϕ_C to a range of $\pm 45°$, as user-focused augmented reality apps often make the assumption of vertical camera views. We restrict the pitch _C as well as yaw angle of the camera in order to think about reducing our field of view (FOV) ψ_C to fall inside of pm half the field of view. This will lead to a 60° field of view θ_C, ψ_C being inside the margins $[-30°, +30°]$. This restricts the estimate to cases when more than 50% of the item is viewable in the picture. This relative position $(\phi_C, \theta_C, \psi_C)$ may be used in the future in conjunction with Rc to determine a position's absolute rotation.

The disadvantage of this kind of depiction is its reliance on the viewpoint of the camera. though the sensor's location is estimated incorrectly, even though its relative position is right, the exact alignment will be erroneous. The upside, nevertheless, is the fact the system we have requires only to make estimates inside a more limited range of values. (e.g. $\theta_C, \psi_C = [-30°, +30°]$ and $\phi_C = [-45°, +45°]$). If the item is still inside the field of view after the motion, the values are continuously updated.

To generate training tags and cameras postures for producing synthetic training pictures, we employ the data space described here. Therefore, we combine the c coordinate of the sensor with the orientation of the device to get a posture $p = (x, y, z, \phi_C, \theta_C, \psi_C)$ obtaining a complete 6-dof from an ancestor. These types of postures can only be achieved from viewpoints when something is partly in view. Each posture P that is generated is recorded as an identifier in a file.

3.4 Image Rendering

In order to teach the system, we produce RGB pictures artificially. The first column of the fourth figure depicts a good illustration. We had to strike a balance among picture file size as well as data density. More data is available for estimating purposes with larger photos. They also include elements like dust, dirt, scratches, manufacturing residues, and other factors that are hard to reproduce in synthetic information. This may restrict a network's extension.

We maintained a standard picture size of 299299 pixels throughout the design process. With that level of detail and field of view, $\varphi = \varphi_h = \varphi_v = 60°$, A lateral shift of 0.386" " cm per picture at far away of 100" " cm is theoretically conceivable:

$$\frac{2 \cdot d(O, c) \cdot \tan(\varphi/2)}{w} = \frac{2 \cdot 100 \text{ cm} \cdot \tan(30°)}{299px} = 0.386 \frac{\text{cm}}{px}$$

Unity's 3D 6 is used to create the visuals. We load the whole labeled data file (P) and cycle through each possible camera posture (p). For more information on how we collect real-world photographs with our actual camera. That is carried out so that the final network can more accurately estimate depth in real-world pictures. Consistent data improvements are taken into account for the presentation. The light is uniformly dispersed. We account for specular reflection. Every picture has a translucent backdrop that will be covered throughout learning as a stopgap measure to improve the quality of the information. Each picture is stretched using the LanczosFilter as well as produced at 16x target frequency (four times each image dimension [width height]). Randomized Grid Super Sampling Anti Alienation [8] is a technique used for anti-aliasing that does not alter the original picture. At last, a picture is produced and saved on disk for every caption.

The icosaheder decomposition is carried out with 400 division per edges to deal with the rising information need for the constant data upgrades. As a result, n1.6106 pictures make up the last set of images. Since PNG is a loss-free compressing design, we only need 40-80 GB of total RAM for our applications.

Before instruction, the thing in the picture has lighting as well as reflections baked onto it permanently. As such, they are seen as a consistent improvement in statistics (Figure 4, columns A).

- Light's intensity as well as angle of incidence affect how things seem, with self-shadowing as well as reflected light playing a role.

To achieve this broad effect, the lighting conditions of each representation of the item are independently randomized. In our industrial setting, we may safely presume that illumination is constantly coming in from above. For this reason, we use a vector that is random with an even distribution for the illumination source, with the average position having a top-down, center look at the item.

- Reflections - Unity3D's re-flection sensors' values of intensity are dynamically shuffled to generate a variety of reflections. Our digitized substance database defines the materials and their fixed qualities (metallic and smoothness). In addition, an actual image is randomly chosen from the COCO collection [9] as well as oriented to form a cubic environmental mapping that is then pro-projected onto reflection probe. This is achieved utilizing environmental mapping [10], which generates a reflection independent of its backdrop (a new, arbitrary backdrop is produced during retraining).

4. Implementation and Experiment

In this paragraph, we will discuss the system architecture, including the platform, applications, use cases, experiments, and conclusions.

4.1 Platform

The system architecture of the developed AR application is depicted in Figure 4. The applications were developed using unity 3D Engine and Vuforia SDK. Initially the target images were uploaded in the Vuforia Target Management in Vuforia portal.

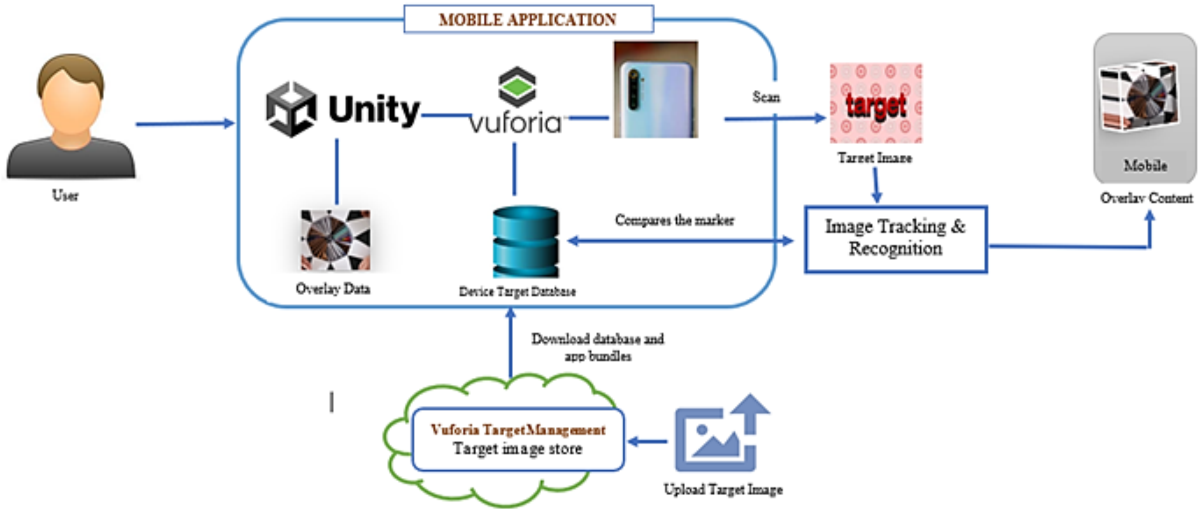


Fig. 4. Architecture diagram of AR applications

The user interaction display was designed using Unity Engine and the target database was imported. The overlay content stored in unity asset is assigned to the target images. Then the AR applications were converted into deployable apk using Android SDK.

Table 1
Software and hardware specification

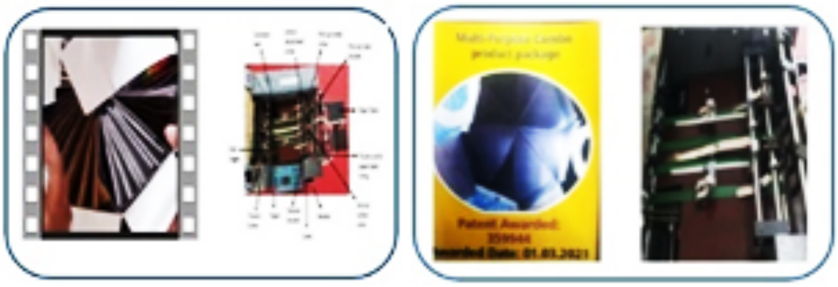
S. No	Software/hadware	Specification
1	Unity 3D engine	Unity 2018.4.36f1 (64-bit)
2	Vuforia SDK	Vuforia SDK Version 8.3.8
3	Visual studio	Version 2017
4	Android SDK	Version 4.1 'Jelly Bean'
5	Light luminous	LUX
6	Processor	Intel(R) Core(TM) i5-6300U CPU @ 2.40GHz 2.50 GHz
7	Installed RAM	16.0 GB
8	Smart device	Android device with camera

4.2 Use Cases

There are two use cases used for developing the augmented reality applications. The use cases for the AR Apps:

i. Combo box
ii. Offset Colt Machine

There are two sets of target images used for developing the applications (offset printing machine and combo box) with two different overlay content (image and video) respectively. The overlay video of the combo box explains about its usage whereas the overlay image of the offset machine depicts different parts of the machine. Size of the overlay video is 40.7 MB and image is 123KB. The Target images of the use case are of different resolution, high, medium and low based on the image quality rating in Vuforia Development Portal. The light luminous of the environment was calculated using an application LUX, procured from play store. There are two types of target images used in testing the performance of the applications are digital form and printed form. The target images for the AR applications developed are shown in Figure 5(a) and Figure 5(b) depicts the overlay content for the target images respectively.



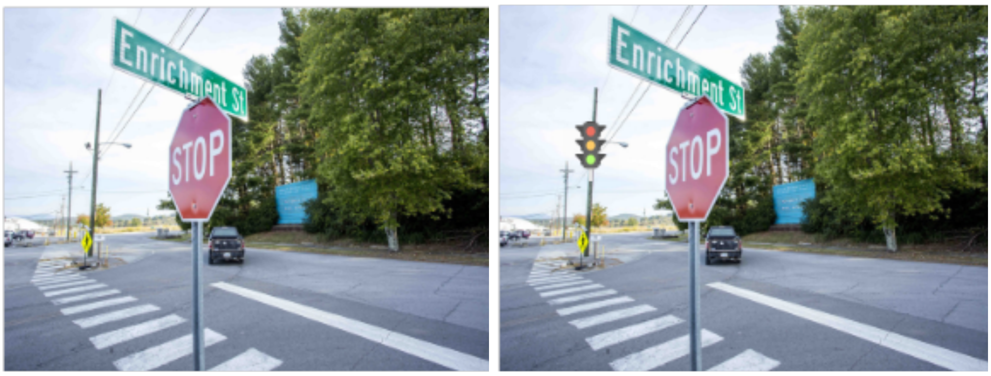
(a) (b)

Fig. 5. (a) Target image (combo box and offset machine) at 104 LUX
(b) Overlay insight of the target (video and image)

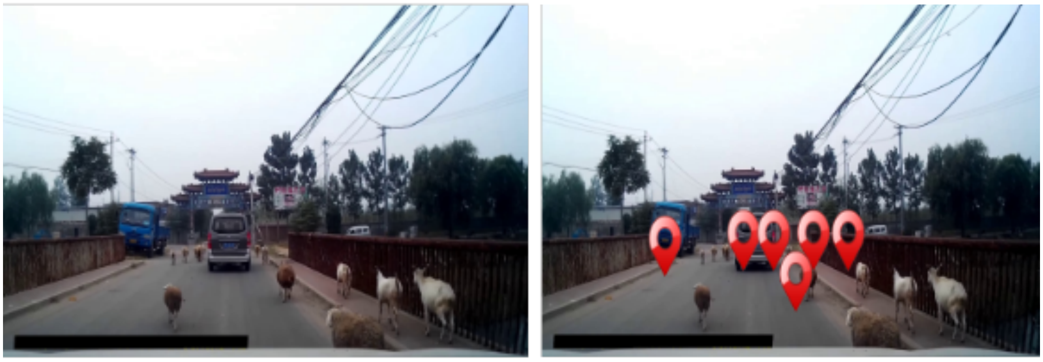
Figure 6 shows the aforementioned instances demonstrate that our technology is able to recover sources of light which aren't apparent in the shot by learning how to map from picture appearances to scenario lighting from enormous volumes of actual image data. Our lighting predictions make it possible to accurately relight digital objects as well as merge them into pictures.

When the user (with a smart device) scans the target image (combo box and offset machine) using mobile phone the Vuforia camera of the application tracks the marker in the image, as shown in Figure 6(a). Compares with the target stored in the device target database, recognizes the marker and retrieves the overlay insight (video and image) respectively, as shown in Figure 6(b).

To test the efficacy of our method, we conducted user research in which subjects were prompted to select the more genuine of two photos featuring the same re-illuminated virtual item under either the ground-truth or estimated lighting scenarios. A total of 41.8% of respondents found our predicted illumination to be exactly the same as, or more genuine than, the ground truth result. This is a substantial increase in performance over the prior experiment, which achieved a maximum of 27.7%. More information about this project, along with variations suitable for use in environments outside.



(a) (b)



(a) (b)

Fig. 6. (a) Target image (combo box and offset machine) at 104 LUX (b) Overlay insight of the target (video and image)

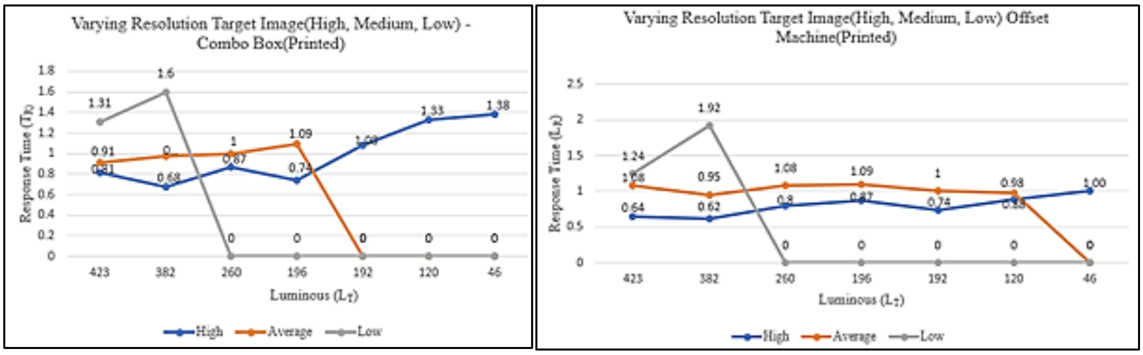
4.3 Experiment

The performance of the applications is measured in terms of the response time of the overlay content. The different overlay content displayed are video and image. The reliability is tested with different scanning light intensity, distance between the marker and the source of light and the type of target image (ie. Digital or Printed material) with different resolutions.

There are different tracks of light intensity used, (i) incandescent light (ii) fluorescent light and dark (low light source). When an obstruction is placed among the light supply as well as the object to be illuminated, the luminous flux decreases to 46 Lux from its maximum intensity of 423 LUX. A LUX light sensor program was used to evaluate the luminance level. The lens of the camera is 15cm away from the object being photographed. The two different forms of target images used were digital and printed material.

4.3.1 Digital target image

Response time of the application with digital target is depicted in Figure 7. Performance of the applications are tested using digital image as target in different scanning light intensities and the distance between the marker and the camera.



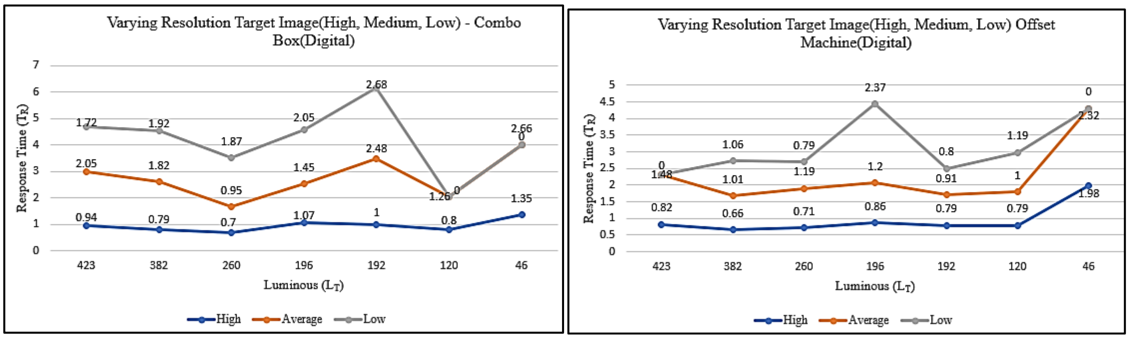
(a) (b)

Fig. 7. Response time of the AR applications under varying resolution of printed target images, different scanning light intensity and distance (a) Combo box (b) Offset machine

When the target image is printed, with different light intensity, the response time of the overlay insight is measured. It is observed that, when the target image resolution is high, the response time is optimal with the varying light luminous. When the target image resolution is average, then the response time increases with change in light luminous. As the light luminous is decreasing, the response time of the application is higher. It is observer that when the light intensity is high, the response time of the overlay insight is optimal and when the light intensity varies. When the resolution of the target image is low, the response time of the overlay insight increases and no overlay is found at certain low intensity. The response time also depend on the color variant printed as the target. If there is little color deviation in printed target, the response time varies accordingly. The light luminous (L_T) is proportional to the response time (T_R) of the overlay insight of the application.

5. Results

With digital image as target, at 260 Lux and 46 Lux, the response time of the application was higher and no overlay is augmented at very low intensity. By increasing the brightness of the mobile there was variations in response time, depicted in Figure 8.



(a) (b)

Fig. 8. Response time of the AR applications under varying resolution of digital target images, after increasing mobile brightness

The response time has certainly increased while the brightness of the mobile has increased. Following are the few observations during the experiments.

i. The performance of the application varies with the digital target images and printed target images. the performance of the app with digital marker is lesser than the printed markers due to the reflection of light on the target image.

ii. Moreover, the colors in the markers play a vital role in image recognition. The recognition of gray scale markers is faster than the RGB markers.

6. Conclusions and Future Work

In this research, we introduced deep learning-based approaches to two difficult visual analysis issues: 6-degrees-of-freedom object tracking and brightness estimate. In both situations, state-of-the-art outcomes are obtained by training deep convolution artificial neural networks on massive datasets. As a result, we expect these kinds of technologies to pave the way for more lifelike apps using augmented reality capable of respond to varying lighting and dynamic environments. The target markers can be of three forms namely, real objects, printed markers and digital markers. The experiment was performed with two of these markers such as printed and digital markers. The above experiment concludes that the response time depends on light intensity. The response time of the application can also be obtained while increasing the brightness of the smart device (mobile phone) rather by increasing the light source. Finally, it is observed that the printed markers efficient than the digital markers due to the reflections. The experimentation with the real object as marker depends on source of light, intensity, distance, reflection of object, object texture, surface, etc.

Future work involves developing the remaining app with real object as target and obtain the performance of the application.

References

[1] Budiman, E., M. B. Firdaus, and U. Hairah. "Augmented reality peripheral performance: light intensity, distance, occlusion and marker testing." In *Journal of Physics: Conference Series*, vol. 1898, no. 1, p. 012013. IOP Publishing, 2021. https://doi.org/10.1088/1742-6596/1898/1/012013

[2] Erickson, Austin, Kangsoo Kim, Gerd Bruder, and Gregory F. Welch. "Exploring the limitations of environment lighting on optical see-through head-mounted displays." In *Proceedings of the 2020 ACM Symposium on Spatial User Interaction*, pp. 1-8. 2020. https://doi.org/10.1145/3385959.3418445

[3] Kán, Peter, and Hannes Kaufmann. "High-quality reflections, refractions, and caustics in augmented reality and their contribution to visual coherence." In *2012 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 99-108. IEEE, 2012. https://doi.org/10.1109/ISMAR.2012.6402546

[4] Madsen, Claus B., Tommy Jensen, and Mikkel S. Andersen. "Real-time image-based lighting for outdoor augmented reality under dynamically changing illumination conditions." In *Proceedings: International Conference on Graphics Theory and Applications, Setúbal, Portugal*, pp. 364-371. 2006. https://doi.org/10.5220/0001353703640371

[5] Yin, Kun, Ziqian He, Jianghao Xiong, Junyu Zou, Kun Li, and Shin-Tson Wu. "Virtual reality and augmented reality displays: advances and future perspectives." *Journal of Physics: Photonics* 3, no. 2 (2021): 022010. https://doi.org/10.1088/2515-7647/abf02e

[6] Kiryakova, G., N. Angelova, and L. Yordanova. "The potential of augmented reality to change the business." *Trakia Journal of Sciences* 15, no. 1 (2017): 394-401. https://doi.org/10.15547/tjs.2017.s.01.066

[7] Kyguolienė, Asta, and Reda Braziulytė. "Application of Augmented Reality in Product Packaging: Challenges and Development Opportunities." *Management of Organizations: Systematic Research* 88 (2022). https://doi.org/10.2478/mosr-2022-0014

[8] Bermeo-Giraldo, Maria Camila, Alejandro Valencia-Arias, Javier D. Ramos de Rosas, Martha Benjumea-Arias, and Juan Amilcar Villanueva Calderón. "Factors influencing the use of digital marketing by small and medium-sized enterprises during COVID-19." In *Informatics*, vol. 9, no. 4, p. 86. MDPI, 2022. https://doi.org/10.3390/informatics9040086