Augmented Reality platform for enhancing integration of virtual objects

Mohamed El-Zayat

Faculty of Electrical Engineering, Sarajevo
Bosnia and Herzegovina

Abstract

With the wide spread of high end processors integrated in mobile devices, ranging from 1GHz processors, to dual core processors and hybrid processors (GPU and CPU on one chip), augmented reality became more popular solution for visualization and navigation. This work proposes an augmented reality platform for organizing and enhancing integration of computer generated objects by introducing lights, shaders and shadows, in pursuing for better experience for the end user, emphasizing on outdoor environments.

Keywords— augmented reality, platform, enhancing integration, real time sun tracking

1 Introduction

Augmented reality (AR) is a relatively new and promising concept. The ability of superimposing digital elements on a physical world with means of interaction with the surrounding world is quite intriguing idea, since AR introduction in 1968 by Ivan Sutherland [1]. However, the technology by that time and for almost next 3 decades was quite limited to lab research, since the mobility nature of AR, and lack of capable mobile processor at that time.

Since the recent rapid development of GPUs, CPUs and recently hybrid processors, AR became increasingly popular. There are two main trends in AR research: registration, where researchers try to solve misalignment and world tracking problems, since the human visual system is capable of detecting even the smallest visual errors; and integration, where researchers are directed towards the enhancement of computer generated object integration with the surrounding environment.

This work proposes an AR mobile platform for enhancing integration of virtual objects in outdoor environments. Section 3 illustrates the proposed AR mobile platform. Section 4 covers registration and Section 5 covers integration, proposing a real time sun tracking system for capturing the current lighting condition of the environment. Section 6 presents results compared to other AR platforms. Finally, in section 7 we conclude the paper and give some directions for future work.

2 Related work

In recent years we have seen significant advances in two promising fields of user interface research: virtual environments, in which 3D displays and interaction devices immerse the user in a synthesized world, and mobile computing.

Previous research in mobile AR has addressed a variety of application areas including 3D mobile AR systems for exploring urban environments [3], enhancing registration through making a hybrid registration for outdoor AR [4], improving teaching with mobile AR for learning and training [5], location based AR for indoor environments [6], enhanced computer generated objects rendering using environment illumination [9] . In pursuing better registration of AR objects, researchers are trying to combine computer vision with sensors for achieving more accurate results [7]. Additionally a combined solution for illumination techniques for AR objects is discussed in [10]. One commercial platform that caught many mobile device users' attention, is Layar AR browser (Figure 2.1).

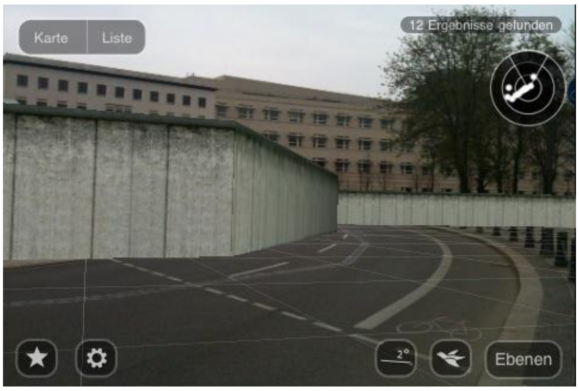


Figure 2.1 Layar Browser, the original appearance of the Berlin wall (image courtesy of layar.com)

This paper proposes the AR mobile platform for arranging AR objects with an emphasis on enhancing the integration of computer generated objects in the outdoor environment, by introducing lights and shaders to the augmented objects.

3 AR platform

Making AR systems that work outdoors is a natural step in the development of AR toward the ultimate goal of AR displays that can operate in any environment. A user walking outdoors could see spatially located information directly displayed over live camera stream, helping to navigate and identify features of interest [4].

While designing the platform, the mobility and optimization factors were taken into consideration as described in the upcoming sections. Figure 3.1 shows the overall architecture of the platform.

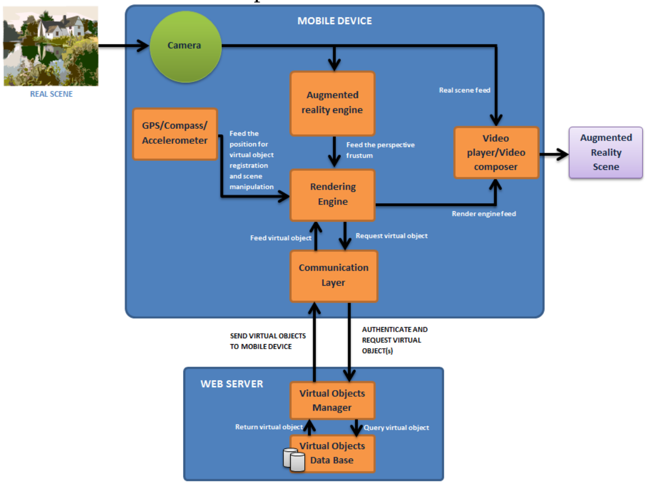


Figure 3.1 AR Mobile Platform architecture

In order to achieve higher performance and decrease the disk usage by 3D models and their associated textures, all 3D models are stored on a web server, and will be downloaded on the mobile device once the user is near the location where the virtual object should be displayed.

3.1 Server side

A database containing 3D models along with the associated textures, and additional lighting data for improving the integration of the virtual object, resides on the server side. Lighting details are covered in Section V.

The Virtual Objects Manager is responsible for handling authentication and requests to the database. This component would be crucial in case this platform is implemented for a mobile provider.

Using the server side for managing virtual objects will relieve the users from constantly updating and downloading the complete application once 3D models get updated, thus increasing the performance and saving unnecessary storage load.

3.2 Client side

This platform requires a mobile device that is GPS capable along with at least accelerometer and compass. Since most mobile device vendors are integrating these sensors as a standard in their devices, we believe that in few years these devices will be common among users.

Mobile devices contain two major components: registration and integration components. Registration component is responsible for AR registration, which could be done using complete sensor based registration (i.e. combining GPS, accelerometer and compass data as discussed in Section IV), or hybrid one as discussed in [4]. A hybrid registration could combine some elements of computer vision and sensor data to improve the integration of computer generated objects. Therefore, the registration system could handle unstructured and unprepared environments [9], and in this case, the AR engine will be activated in the augmentation pipeline.

Once the user gets to a desired location detected by GPS, where a virtual object resides, registration component will generate the frustum that will be handed to the rendering engine.

Rendering engine will send a request to the communication layer to load 3D object and its associated data from the server side, thus rendering the virtual object and “clearing” the background with the camera feed, hence superimposing the virtual object over the physical world.

4 AR registration

In order to enhance the integration of augmented objects, improving registration is required. GPS data is required to determine the position of the virtual object in the physical world and the position of the user according to the position of virtual object, hence calculating the position of the frustum according to the physical world using equation 4.1:

$$(x_2 - x_1)^2 + (y_2 - y_1)^2 < r^2 \quad (4.1)$$

x_1, y_1 represents user position, while x_2, y_2 represent the virtual object’s position. r is the range value. If the user is in the range of the detected object, he/she will be notified and the frustum will be generated.

In order to detect the rotation of the user, a compass will be used for azimuth rotation direction, and the accelerometer will control frustum altitude as shown in figure 4.1:

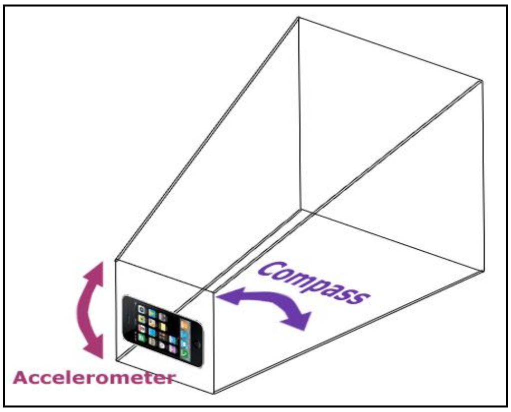


Figure 4.1 frustum controlled by sensors

The registration is sensor based, in pursuing for saving CPU/GPU cycles to enhance the rendering of augmented objects, hence improving the integration.

5 AR integration

There are several factors that have to be taken into consideration for outdoor rendering of augmented objects. One of these factors is lighting, which is a crucial component in rendering any object in a scene.

In order to improve the integration and create a realistic scene, AR platform should track sun position in real time, thus approximating lighting conditions of physical objects along with their shadows. Therefore a directional light is used to simulate sunrays.

Earth is relatively spherical celestial object that rotates around itself eastwards every approximately 24 hours and around the sun approximately every 365 days. The axis on which earth rotates is the Polar axis. The great circles that intersect with the Polar axis are called meridians. The great circle equidistant from the North and South Pole is the equator [12].

Since Earth rotation axis is tilted, by declination of $23.4°$, it results in changing the relative position of the sun as the Earth moves in orbit. This change reflects on the angle of the sun rays according to the equatorial plane. This angle is called declination.

Figure 5.1 shows the Sun position towards the Earth along with the above described angles.

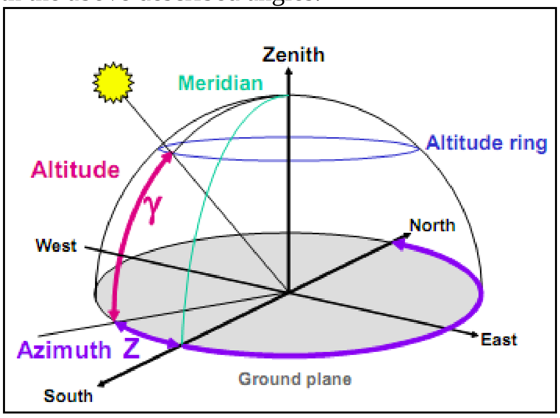


Figure 5.1 Sun position towards the Earth [12]

γ is the altitude of the sun above the ground (horizon) plane, z is the azimuth, which is the compass direction of the sun on the ground plane.

Declination is calculated using equation 5.1:

$$Declination = 23.4 \times \sin\left(\frac{360 \times (284+N)}{365}\right) \; degrees \quad (5.1)$$

where N is the number of the day for which the declination is being calculated, January 1[st] being day number 1.

The Azimuth may be expressed in two ways; either as the angle clockwise from North or as the angle East of or West of South. Although the former is most often used, we used the latter convention.

Azimuth and altitude of the sun can be calculated using the following equations [12]:

$$\sin\gamma = (\cos D \times \cos L \times \cos H) + (\sin D \times \sin L)$$

$$\cos z = \left(\frac{(\cos D \times \cos H \times \sin L) - (\sin D \times \cos L)}{\cos\gamma}\right)$$

Table 5.1 shows the legend for the above equations:

Symbol	Variable	Definition
D	Declination	The angle of the sun rays to the equatorial plane, positive in the summer.
L	Latitude	Angle from the equator to the position on Earth's surface
H	Hour angle	The angle the Earth needs to rotate to bring the meridian to noon. Each hour of time is equivalent to 15 deg.
N	Day number	The day number, January 1[st] is 1.

Table 5.1 Azimuth Altitude equation legend

The colour of the Sunrays plays an important role in displaying the time of the day, and also determines the colour temperature of objects. This problem can be approached by implementing atmospheric scattering algorithms to change sun and atmosphere's colour depending on the time of the day. However in order to decrease the CPU/GPU load, and increase the frame rate, sun colour could be determined through basic hardcoded RGB values [11] as shown in table 5.2:

Source	RGB(0-255)	RGB(0-1)
Sun at sunrise or sunset	182, 126, 91	0.71, 0.49, 0.36
Direct sun at noon	192, 191, 173	0.75, 0.75, 0.68
Sun through clouds/haze	189, 190, 192	0.74, 0.75, 0.75

Table 5.2 Sun colour at different times of the day

Using those values, the RGB numbers between the two stages of the sun during the day could be extrapolated based on the starting time of the simulation.

Another factor that has to be taken into consideration is the night time, where no sun or any light source is available except the presence of the moon, at a certain times of the month. In real life, the moonlight illumination is almost unnoticeable in urban or artificially illuminated areas, thus the moon light factor will be neglected in this case (Figure 5.2).

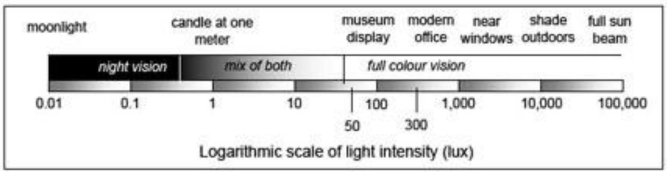


Figure 5.2 Logarithmic scale of light intensity (image courtesy of Canadian conversation institute)

In order to compensate the loss of sun light, Rendering Engine component queries for artificial light that is associated with the desired model from database on the server side. The artificial light data contain position, light type, and the diffuse components.

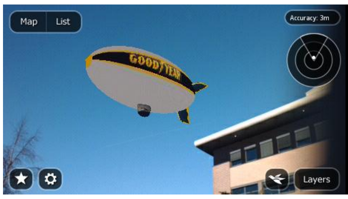
At the early dawn or late sunset, where sun's illumination is not strong enough to illuminate the augmented object, the directional light is turned off, and the queried artificial lights are activated.

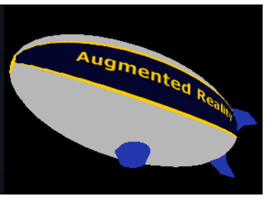
One possible way to speed up lighting calculations is by performing the light map, which is especially useful when used in conjunction with multitexturing. Additionally, texture baking could be used as an alternative technique for increasing the frame rate [13].

6 Results

By the time of writing this paper, tests have been running under OpenGLES2.0 emulators, and since computers video cards can handle OpenGLES2.0 with almost no frame rate loss, as a result, frame rate data will be neglected for the time being.

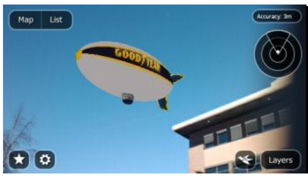
Current models are using per-vertex illumination, and the shaders component of this platform is still under development. As a consequence, directional light and spotlight with insignificant diffuse values for smoothing shadows have been added. However, the loss of per-pixel illumination does not highly affect the outcome; an image displaying the color temperature of the models at different times of the day is shown in Figure 6.1 compared with Layar render for the augmented 3D object:





(a) Layar Augmented 3D object at noon (image courtesy of Layar.com)

(b) Default OpenGL Light (i.e. OpenGL lights not enabled)



(c) Direct sun at noon

(d) Test render direct sun at noon

(e) Sunset

(f) Sun at Dawn

(g) Cloudy/Hazy weather

Figure 6.1 Simple zeppelin model, under several lighting conditions compared against Layar AR platform renderer.

Initially, comparing Figure 6.1 (a) with (b), that Layar AR Platform did not include any lights for illuminating its 3D objects. Nevertheless we could see a dramatic change even on per-vertex lighting when comparing (c) and (d) renders. It is worth noting that comparing Figure 6.1 (c) and (d) carefully, the latter has lighting temperature close to the camera feed and almost has the same sunray direction.

Color temperature of objects during noon could have a bit of yellow tint to it, as shown in Figure 6.1 (c) on the building bellow the zeppelin and the test zeppelin render(d). Figure 6.1 (e) shows the zeppelin model under sunset lighting condition, while for the cloudy or hazy weather as showed in (g) light color tends to be white. We believe that, completing the shader component will gain better results.

After several tests, one could disagree with [11] for using same values for sunrise and sunset, since during sunrise (i.e. dawn), objects' color temperature tends to be cold, thus has blue tint to it as shown in Figure 6.1 (f), hence one could suggest to add one more RGB values for sunrise. We suggest that these values would be (0.50, 0.49, 0.60) for R, G and B, assuming that RGB values goes from 0 to 1.

7 Conclusions and future work

In this paper a solution for AR organization and integration problems, in a context of generic AR platform is presented. This solution deals with two major topics: integration where virtual objects are illuminated according to the time of the day, by tracking sun position in real time, thus estimating the correct color temperature and shadows; and organization where all objects are organized in a database along with its illumination data on remote server. 3D objects are downloaded upon a query from the client side (i.e. mobile device). Hence superimposing the downloaded object after light calculations are finished.

The presented platform is still under development, though several features could be implemented, such as calculating length of shadows, compensating for rainy, snowy or cloudy weather, where most of the objects tend to have no shadows since light distortion is very high and the object is illuminated almost from all sides.

Implementing several lighting techniques, such as light mapping or texture baking could assist in increasing the frame rate.

Current calculations for predicting Sun position still lack a very important parameter, which describes the accurate position of the user on Earth. Therefore Sun position would be calculated with a higher precision.

Another important factor that could affect the experience of the user is the quality of the camera in their mobile device. Since modern mobile devices are equipped with autofocus and/or have automatic exposure correction, one possibly would notice the difference of color temperatures and light intensity between the augmented object and the physical world.

This platform could not fully enhance the integration if augmented objects exist in urban areas. The enhancements which will be negatively affecting the experience, are when the augmented object's position has special geological properties, or it is surrounded by other higher objects that cast shadows on it.

References

[1] S. Cawood and M. Fiala Augmented Reality: A Practical guide. The Pragmatic Programmers, 2007.

[2] www.layar.com.

[3] S. Feiner, B. MacIntyre, T. Höllerer, A. Webster. A Touring Machine: Prototyping 3D Mobile Augmented Reality Systems for Exploring the Urban Environment. In Personal Technologies pp 208-217, 1997.

[4] K. Satoh, M. Anabuki, H. Yamamoto, and H. TamuraA Hybrid Registration Method for Outdoor Augmented Reality. ISAR'01, 2001.

[5] R. Wichert. A Mobile Augmented Reality Environment for Collaborative Learning and Training. Tivoli Systems Inc, 2002.

[6] G. Reitmayr and D. Schmalstieg. Location based Applications for Mobile Augmented Reality AUIC03, 2003.

[7] M. Kanbara and N. Yokoya. Real-time Estimation of Light Source Environment for Photorealistic Augmented Reality ICPR04, 2004.

[8] R. Azuma. The Challenge of Making Augmented Reality Work Outdoors, In Mixed Reality: Merging Real and Virtual, pp 379-390, Springer-Verlag. 1999.

[9] L. Chai, W. Hoff, T. Vincent. 3-D Motion and Structure Estimation Using Inertial Sensors and Computer Vision for Augmented Reality, Teleoperators and Virtual Environments, 2005.

[10] S. Pessoa1, E. Apolinário1, G. Moura, J. Paulo S. Lima, M. Bueno, V. Teichrieb, J. Kelner. Illumination Techniques for Photorealistic Rendering in Augmented Reality, SVR 2008.

[11] J. Birn. Digital Lighting & Rendering- Second Edition, New Riders, 2006.

[12] Building Environment 1 lecture notes, University of Bath.

[13] P.Ridout. iPhone 3D programming, O'REILLY, 2010.