Light Source Estimation in Mobile Augmented Reality Scenes by Using Human Face Geometry

Emre KOC*, Nonmember and Selim BALCISOY†, Member

SUMMARY Light source estimation and virtual lighting must be believable in terms of appearance and correctness in augmented reality scenes. As a result of illumination complexity in an outdoor scene, realistic lighting for augmented reality is still a challenging problem. In this paper, we propose a framework based on an estimation of environmental lighting from well-defined objects, specifically human faces. The method is tuned for outdoor use, and the algorithm is further enhanced to illuminate virtual objects exposed to direct sunlight. Our model can be integrated into existing mobile augmented reality frameworks to enhance visual perception.

key words: Mobile Augmented Reality, Outdoor Illumination, Real-time Rendering

1. Introduction

Virtual Object Lighting constitutes a critical component in Augmented Reality (AR) applications. Today it is common to see many AR applications in desktop applications and mobile devices. In order to increase the feeling of reality in an AR scene, virtual objects should be lit realistically by using the information from the surrounding environment. In this paper, we propose methods to solve two distinct issues in the AR domain. The first is extracting illumination information from the human face in an outdoor scene and the latter is the lighting of virtual objects realistically by using the extracted lighting information.

We propose a framework which uses Lambertian surface properties to extract lighting information from a predefined geometry, which can be a human face. This framework contains a face detector in order to place the face model on the input image. After placing the model, we extract light directions and process sensor data so that real-world light directions and light intensity information can be calculated correctly.

Environmental lighting is essential for the realistic perception of augmented reality scenes; therefore, extracting lighting information from predefined objects has always been an interesting problem for the graphics community. A recent survey by Swan and Gabbard [1] discusses the effects of lighting and shadows on user’s perception of reality and depth. Augmented reality applications that provide only tracking a coded marker or tracking a surface do not provide a real depth feeling to users. It is also hard to extract lighting information by using predefined objects like a mirrored ball in outdoor applications. Hence, we need a tool which will allow us to obtain information about the environmental lighting which does not depend on any special object, which users do not often carry with them, and which does not depend on the visibility of the sky, a large area of ground or nearby objects in order to get a reference lighting and shadow estimation. To help users who are investigating their surrounding environment from a screen, an AR system should be able to detect lighting information dynamically and update the virtual object’s lighting and shadow information in real-time. Our motivation is to propose a light estimation technique that can extract lighting information for a scene while the user investigates the given outdoor area.

An outline of the rest of the paper is as follows. In Sect. 2, we give an overview of previous research on both light direction estimation and virtual object lighting techniques. In Sect. 3, we define how our light estimation system works and its basis. Section 5 explains how we efficiently light objects on a mobile system with estimated light directions. Section 7 shows test results for the proposed system. Finally on Sect. 8, we summarize our results and propose future work on the framework.

2. Related Work & Background

Both light source estimation from a given geometry and lighting in augmented reality involve extensive literatures.

2.1 Light Source Estimation

Several light source estimation techniques have been proposed in the literature. We will try to review them according to the technique they employ to capture light information from the environment.

Light source detection from a predefined geometry is the main technique used in most of the previous works. Previously, there were extensive studies about shape and reflectance recovery by using the method shape from shading [2]–[5]. However, these studies did not extend to recovering illumination information from an object with known shape and reflectance but instead focus on light direction. Sato et al. [6] proposed a technique to handle this problem with the help of cast shadows of an object to a scene. This method works on scenes with cast shadows but there wasn’t any extension to the extraction of light information.
only from a non-convex object. Due to the difficulty of the problem, most studies involve several assumptions. Jensen et al. [7] proposed a solution to solve real-time image-based lighting problems for outdoor augmented reality scenes where the illumination conditions are dynamically changing. They created some constraints and assumptions like no precipitation as they heavily depend on surface reflectance in the environment. Apart from what makes this system hard to adapt in different environments, they need a 3D model of the scene with an HDR environment map recorded in the centre of the scene. Even though they do not require a detailed 3D model, a rough model for a new environment is hard to create in mobile augmented reality scenarios.

Different groups of researchers are interested in resolving the photometric problem, estimating light source direction in real-time by using specific shiny objects, most often mirrored balls [8]. Debevec [9] proposed a technique to capture scene radiance and global illumination in order to have correct lighting for virtual objects. Even though the techniques presented have realistic visual quality renders, using a mirror ball and correctly placing it in the center of a scene to capture HDR images takes a long time to setup and cannot accommodate lighting changes in real-time. A recent study by Tominaga et al. [10] estimates the illuminant spectra of an omni-directional light distribution from the images of a camera aiming at a mirrored ball. They proposed algorithms to measure spectral radiance distributions in an outdoor scene. By extracting radiance distributions, they achieve a realistic representation of both sky and sun but again the problem of using a hard-to-find mirrored ball arises in mobile outdoor augmented reality. Kanbara et al. [11] proposed a technique for real-time estimation of a light source environment in an augmented reality system. They used a marker tracking system that contains a small black mirrored ball. Even though this method allows real-time tracking of the light changes it is not feasible in a large augmented reality setup such as an excavation site. Since light source estimation depends on the small mirrored ball, environmental effects such as dust or mud might occlude the clear reflections on its surface.

Another group of researchers studied light source estimation from a scene by using fish-eye lens cameras. Yoo et al. [12] capture light information by using a 185 degree fish-eye lens and Neutral density filter (NDF) to have no changes in hue of color rendition by modifying the intensity of all wavelengths or colors of light equally. They stated that lighting maps can be extracted precisely by using fish eye lenses with NDF. Fish-eye camera models can give very detailed information about lighting information in a scene but they are difficult to setup and hard to maintain since the main prerequisite is they should always face light sources. This constraint makes them impractical for rapidly changing mobile augmented reality applications.

Estimating light information from the human eye is also a well-studied topic. Tsumura et al. [13] determines where the light source is by using the reflections in the image of the eye. They propose a method that uses the reflectivity of the human eye to capture up to three light sources. [14], presents a detailed analysis of the characteristics of the cornea image of an eye which was taken by a catadioptric (mirror + lens) imaging system. They show that geometric parameters of the corneal system are suitable for environment map extraction from a single image. Wang et al. [15] improves this technique by separating corneal reflections in an image of human irises. They estimate illumination from the surrounding scene by using human iris features such as chromaticity and illumination correspondence between two irises. Even though extracting an environment map from the eye seems feasible in indoor scenes, it becomes infeasible in outdoor scenes where direct sunlight comes into the eye.

Estimating lighting information from the human face was studied for cancelling out lighting variations in order to enhance the performance of face detection/recognition applications [16], [17]. Basri et al. [16] show a low-dimensional linear subspace which is effective for face recognition by using spherical harmonics basis. They used spherical harmonics for representing lighting information on the human face. Lee et al. [17] proposed a technique for estimating directional lighting in uncalibrated images of faces in frontal poses. They estimated the principal lighting direction by using least-squares formulation with the Labertian illumination model. Their technique uses surface normals as light direction vectors and they assume light direction does not change with intensity information in each pixel, which is the same as our assumption.

2.2 Lighting Virtual Objects

Rendering realistic light in real-time has long been a challenge in the computer graphics community. There have been several methods for calculating shading, such as Phong interpolation [18], normal-vector interpolation shading [19], and per-fragment lighting [20]. Most of these techniques can be implemented faster than real-time in current graphics hardware. Even though these models can shade virtual objects correctly for a viewer to have a depth perception, they are unable to simulate the occlusion of light and shadow casting. Williams proposed a solution to shadow calculations [21] called shadow mapping. This method allowed the graphics community to have complex shadows even for dense geometries or curved surfaces. Although calculating soft shadows runs in real-time, having more realistic scenes requires powerful display cards and programmable pipeline capabilities. Off-line rendering methods such as Monte Carlo ray tracing [22], [23], radiosity [24] are able to provide higher visual quality than standard lighting methods in computer graphics but they are costly and their complexity increases proportionally with the included lights in the scene.

Spherical Harmonics Lighting (SHL) was introduced to the graphics community by Sloan et al. [25]. They introduced a method that transforms low-frequency incident lighting to transferred radiance which includes shadows, occlusions and inter-reflections. They defined these trans-
ferred functions in low-order spherical harmonics. They even extended this work by compressed per-point transfer matrices from a high-dimensional surface signal by *clustered principal component analysis* [26]. Green explained the underlying mathematics of the proposed method and expressed several details about spherical harmonics lighting that the original paper did not cover; this paper was the main guide in our rendering system development [27]. These methods opened a new way of defining global illumination in computer graphics. Previous techniques allowed realistic renderings with several assumptions, but they mostly captured light information from a limited angle. **SHL** allows dynamic changes in light and even dynamically moving objects.

### 2.3 Motivation

There are several methods proposed both for the exportation of light source direction and the illumination of virtual objects in a real scene. These methods motivated us to develop an augmented reality framework that works on a specific scenario where a user points a mobile device, which has a double-sided camera, to a field where the user wants to inspect any kind of virtual content. Primarily, we needed a light source estimator that will work on a specific object, mainly the human face, and does not depend on expensive math calculations. Secondly, since most of the previous work on light source estimation works on a single image or a series of images which are consequently taken, they did not need a light space projection between two different cameras. Therefore, we needed sensor information on how our mobile device is placed in the real-world in order to relate extracted light information with the augmented view. Third, in outdoor scenes, the sun is the main light source, so we wanted to know which direction the sun is coming to our scene. Fourth, there is a need to know where we will put our virtual objects in the view of the back facing camera, so there should be a tracking mechanism involved. Last, we need to illuminate virtual objects in an augmented reality setup to make them seem as if they are in the same scene.

In order to suffice these requirements in a framework, we eliminated light source estimation techniques that take a very long time and need extensive geometry information about the scene. We focused on estimating light source automatically, without having any user interaction or requesting any information from the user. Since we defined our system to work on a mobile environment, we put constraints on the mobile device hardware to have Digital Compass, Gyroscope and a two-sided camera setup, which are very common in recent end-user phones and tablets. Even though we performed some tests in an indoor controlled environment, there is a need to assume that a light source is placed distantly. Our assumption is that our face and virtual objects that will be placed have the same illumination properties. Therefore, we defined our system to work only for daylight configuration in an outdoor scene and we used a tracking system that works with square markers.

In practical terms, we wanted to build a framework that can be used in scenarios where users navigate through an outdoor environment for information retrieval. For instance using this framework in an excavation site can help archaeologists to investigate that site with correct depth perception using the help of realistic lighting of virtual objects. Having depth perception is a crucial feature in virtual object rendering in augmented reality, which we explained in Sect. 5. By having correct depth perception, archaeologists can use this framework for virtual object placement on the excavation site. Usage of this system can also be extended to historical site visitors to show how a demolished historical building would be seen during that time of the year.

### 3. Estimating Light Directions

In this section we will introduce the light source estimation engine which constitutes the core of our outdoor lighting framework. First, we will describe how we place our face model to a front-facing camera view and then we will continue with explaining the light direction estimation procedure.

#### 3.1 Face Geometry, Detection and Pose Estimation

We define our face geometry to be a 2D normal map of a base human face. This face geometry is created by using the base face geometry from [28]. Unnecessary parts such as eyes and back facing geometry is removed to increase precision (Fig. 3). Final face geometry is perspective projected to 2D image by using the parameters of front facing camera. Each pixel in this image contains a 3D vector that shows the pixel normal.

Even though detecting faces is straightforward within an outdoor scene, light variations can make this process an ill-posed problem. We used OpenCV Face Detection algorithm [29]. We know that our user is directly looking at a mobile device screen and a front-facing camera is placed close to the screen. Therefore, we assume that the distance of a user’s face from the camera is between 20 cm - 80 cm. Cascaded Haar classifier also allows capturing the left and right eye’s position separately. We defined a condition to obtain face rotation from the image as follows by using Eq. (1).
bertian Cosine Law, a lambertian surface will have the same assumption about the surface is Lambertian. From the Lambertian smooth face model that contains lighting information, our texture. We apply bilateral filtering to remove any high frequency texture details (right).

Fig. 2  HSL color space contains lightness information and RGB space defines a color by using three main colors [31] (left). Applying a bilateral filter to a lightness image gives a very smooth lightness image without any high frequency texture details (right).

![Matching normal map with captured and bilateral filtered image.](image)

Fig. 3  Matching normal map with captured and bilateral filtered image.

\[
\beta_{\text{Head}} = \begin{cases} 
\arctan(2(P_{R_i} - P_{L_i}, P_{R_i} - P_{L_i})) & \text{if } P_{R_i} > P_C \text{ and } P_{L_i} < P_C, \\
\arctan(2(P_{L_i} - P_{R_i}, P_{L_i} - P_{R_i})) & \text{if } P_{R_i} < P_C \text{ and } P_{L_i} > P_C, \\
0 & \text{otherwise}
\end{cases}
\]

(1)

Where \( \beta \) represents ‘Roll’ of user head, \( P_C \) is the center position of the face region and \( P_R \) and \( P_L \) are the right and left eye positions, respectively. In this definition, as the classifier can output rotations up to 25 degrees, from the conducted experiments, we can say that in most cases the system can detect roll angles up to 15 degrees.

3.2 Estimating Azimuth and Zenith Angles

In this section, we will show how we compute azimuth and zenith angles from a given input image by using our prior knowledge of human face geometry. In an outdoor setup, we know that in a daylight configuration, we expect to see the direct sunlight’s specular component on a user’s face. One problem can be the occlusion of clouds which will have a direct effect on the specular component’s visibility. We defined a similar approach to [30] by defining each light with an unknown luminance \( L_j \) and unknown unit direction \( \omega_j, j = 1 \cdots N \) and we also assume the human face as a Lambertian surface. We start by converting the color domain of our image from RGB color space to HLS color space (See Fig. 2 left side) where HLS stands for hue, saturation and lightness. Now we have a single channel image that contains lightness information invariant from hue and saturation information.

Even though the image is invariant from the color information, it may contain high frequency variations on its texture. We apply bilateral filtering to remove any high frequency variations on the face texture [32]. Since we have a smooth face model that contains lighting information, our assumption about the surface is Lambertian. From the Lambertian Cosine Law, a lambertian surface will have the same apparent brightness from any angle that it is viewed. However, we know that our camera is placed as parallel as possible to a user’s face since face detection will not work otherwise, so we propose a technique for this specific scenario where a ray traced from our pinhole camera to a user’s face reflected by that surface normal, approximates light direction more accurately. This scenario holds especially when light is received from angles in the range \([\pi..2\pi]\), which are azimuth angles where light is coming from the back side of the user’s head. If we were to assume \( N \) to show the direction of light, azimuth angles would be in range \([0..\pi]\) which will not be enough to approximate light vectors coming the from the back side.

For the lighting of virtual objects we use light directions in spherical coordinates so we can define light direction by our reflection vector for each pixel with the following formula

\[
\begin{align*}
 r &= \sqrt{x^2 + y^2 + z^2} \\
 \theta &= \cos^{-1}\left(\frac{z}{r}\right) \\
 \varphi &= \tan^{-1}\left(\frac{y}{x}\right)
\end{align*}
\]

(2)

where \( \theta \) denotes angle of light from the z up axis, \( \varphi \) defines the angle of light in projected to horizontal x-y plane. \( x, y \) and \( z \) are the values of a vector that is the reflection of vector coming from the camera by the surface normal at the given pixel. \( r = 1 \) since we use normalized vectors.

As we have two spherical coordinates in the face plane and intensity for each pixel, we cluster these spherical coordinates with respect to their magnitude and see if we have any light source other than sunlight.

In order to simplify the clustering of vectors, we will assume that our base intensity value is the average intensity in the face texture. This assumption may fail in a perfect diffuse lighting environment during an overcast day; however, in that condition we can only use a dim ambient light for the object. Masking the input vectors by thresholding the input image leaves only the vectors that have higher intensity from the specific direction; we call these as Intensity Vectors defined in spherical coordinates.

3.3 Clustering Intensity Vectors

The notion of \textit{k-means clustering} is suitable for grouping a set of points; however, we assume that in the case where...
we have more than one light source or a high intensity reflection from a different direction than sunlight, any pixel in the image may be lit by the two light sources. We borrow the idea of Fuzzy C-Means (FCM) which was developed by [33], [34] and allows one piece of data to belong to two or more clusters.

We do not know how many light sources are reflecting through the user’s face so we assume there is only sunlight in the scene and intensity vectors should point close to that direction. Calculating a standard deviation of an azimuth angle gives how diverse the light vectors that are distributed in the scene are. Azimuth angle defines the light direction rotated in z axis; in our conducted tests with our mobile platform, by using 85 different images we concluded that having a standard deviation larger than 20 degrees shows that the user’s face is lit by another strong light source. During the initial calculation of standard deviation on azimuth angle, in the case where the value is large, we initialize FCM with cluster count two. We repeat this step for each cluster until we get an acceptable standard deviation.

Diversity in zenith angle is a feature that is not as straightforward to extract as azimuth angle. In the case where we have one sunlight and a very strong reflection from a floor, it would be very hard to distinguish the large intensity distribution of sunlight from the reflection of the floor in the same azimuth angle. In order to capture any extreme lighting conditions where there is a strict distinction between intensity vectors, we apply a separate FCM process by starting with two clusters as explained with azimuth clustering. In the first iteration, we check for mean angle differences between two clusters. If the mean value difference is larger than the defined angle limit parameter of our algorithm, which we defined as 30 degrees during our tests, it is possible to conclude that we have a separate light source coming through a different angle; it is left to the application programmer to define the limits of cluster count and angle difference limit for zenith angles.

4. Light Direction Projection

Projection of light direction from front-facing camera to back-facing camera is one of the fundamental features of our system. In the previous section, we showed that we can find azimuth and zenith angles from the face image, but these values are in the coordinate space of a mobile device’s front-facing camera. In order to map these spherical coordinates to real-world coordinates, we need to use some sensor information available as a requirement in our mobile device.

Sensor initialization plays a crucial role in determining how the mobile device is positioned in world coordinates. Digital compass streams information about device rotation with respect to the earth’s true north. Even though the sensor is accurate, there is a need to make sure that the device is positioned perpendicular to the vector coming from the center of our planet. In this stage, we can continue with the initialization of a gyroscope. The gyroscope is a device for measuring orientation that we can use to track rotations in three dimensions Yaw, Pitch and Roll. Since the only available data from the gyroscope is the amount of change in orientation, there is a need to initialize the Yaw axis by fixed rotation information, which is the compass direction. We need to inform users about the initialization process and guide them to place the device perpendicularly to the world’s axis by using the accelerometer information. As this process is done once for each execution of the software, it does not affect the usability of the system.

Knowing the device orientation in real-world space allows us to project lighting information captured from the front-facing camera to real-world coordinates with the predicted intensities.

5. Lighting Virtual Objects

Working on a mobile platform requires further optimizations for having real-time rendering rates as a result of lower CPU and GPU power. In previously augmented reality applications, most of the work has been done on how people can use this new technology in different environments [17]. In those applications researchers defined how the interaction with augmented reality systems should be defined [1].

Shadow mapping is an essential component for a user to understand where an object is placed and at what height and depth it resides. As stated in [35], shadows are effective for recognizing spatial relationships in the depth direction. The effect was especially significant in the case of monocular display where there are no stereo spatial cues. Such a result underlines the importance of shadow casting for depth perception. Since we are working on a single display mobile device, users can only understand the depth information by using shadows (See Fig. 5).

Real-time lighting in a mobile device requires an efficient method for realistic rendering. Calculating diffuse lighting without any shadow calculation in SHL is more efficient than using default OpenGL lighting since we reuse the calculated parameters from SH coefficients in each frame. Even though we see the surface structure, we cannot be sure about the depth of objects. As we can see in Fig. 5, shadow information allows us to perceive depth information correctly. Therefore, we use spherical harmonics coefficients with a addin factor for visibility testing results for each vertex. Visibility data gives us both shadow and occlusion information for objects in the scene.

6. Skylight Model and Tone Mapping

Lighting from sunlight needs a robust skylight model. We
used the Preetham [36] skylight model which was explained in detail for spherical harmonics lighting in [37]. The output of Preetham’s skylight model gives High Dynamic Range (HDR) color. We implemented Reinhard tone mapper [38] for tone mapping HDR values to visible range. Most HDR calculation is done in image domain where a virtual scene is rendered to a render target that supports 16 bit Floating point values. Then, the algorithm for tone mapping is applied to pixels. We implemented this system for per-vertex color calculation.

We approximate the key of the scene by using Eq. (3).

\[
\hat{L}_{w} = \frac{1}{N} \exp \left( \sum \log(\delta + L_{w}(i)) \right)
\]

Where \( L_{w}(i) \) is the “world” luminance for vertex \( i \), \( N \) is the total number of pixels in the image and \( \delta \) is a small value to avoid the singularity that occurs if vertices exist that do not receive any light. As mentioned in [38], if the scene has a normal-key we would like to map this to middle-grey of the displayed image by using Eq. (4).

\[
L(i) = \frac{\alpha}{\sum L_{w}(i)}
\]

Where \( L(i) \) is a scaled luminance and \( \alpha = 0.18 \) called exposure. For low-key and high-key vertex colors they allowed \( \alpha \) to be changed as proportional to 0.18. We used this average luminance value to tone-map HDR colors with Eq. (5).

\[
\hat{L}_{w}(i) = \frac{L(i)(1 + \frac{\hat{L}_{w}(i)}{\Gamma_{white,pass}})}{1 + L(i)}
\]

We did not implement the automatic dodging-and-burning feature explained in the paper, which controls how very high luminance appears blurry. From our tests in an outdoor environment with diffuse objects, the above formula gives acceptable results. Implementing automatic dodging-and-burning also requires convolution operations which Reinhard implemented by using multiplication in the FFT domain. Even though this gives better results in highlights it adds more complexity to each frame’s render routine.

7. Case Study

We ran our defined methods on several input images which were taken at various times of the day. We gathered all images by using a two sided camera built in to an iPad and read sensor information in order to process them on a PC. In order to make sure our system is cross-platform compatible, we only used open-source libraries which can be ported to any mobile platform.

7.1 Controlled Environment Tests

In order to see our system’s performance clearly, we captured several images in a controlled environment where we specified the light direction with only a single light source without any reflections from other surfaces.

These tests were performed on Macbook Pro with the built-in i-Sight camera. As the OpenCV face detector runs at 25 fps and applies bilateral filtering to the face region detected, it slows down the capture rate to 15-20 fps where capturing gets slower as the face image regions occupy a larger area. The results show that for controlled light environments where a light source is not as infinite as sunlight, highlights on the image can mislead the algorithms since there are two separate peak points in the image as a result of very close lighting. From the algorithmic perspective, we can treat these types of differences as different light sources or we can limit FCM clustering and take the average of the light directions for a single light.

We can clearly see the effect of peak points in Table 1. In Image A, light reflects both from cheek and forehead. This increases error rate in zenith \( \phi \) angle. This also affects azimuth error rate because algorithm takes normal into account. Normal in forehead points to a light from upper section. Even though this is not a major contribution to main direction, it increases error rate. These errors are minimized in Image B since light is coming from a major direction. Some minor error introduced as a result of reflections in nose and cheek. In Image C, we have the smallest azimuth error rate. Light is coming through the right side of the face with no major reflection from other face sections.

---

**Fig. 6** Controlled Environment Test Image Sequence, Image A: top-left image (0,0), Image B: image at (9,4), Image C: image at (5,3). Test results for these images can be found in Table 1.

| Image | R (\( ^\circ \)) | E (\( ^\circ \)) |
|-------|----------------|----------------|
| A     | 110°           | 160°           |
| R     | 10°            | 90°            |
| Image C | 110°           | 140°           |
| A     | 95.48°         | 144.7°         |
| R     | 14.52°         | 4.7°           |

**Table 1** Real Measured Light Directions (R), value returned by our estimation (A) and absolute error (E) for the zenith \( \phi \) and azimuth \( \theta \) angles in the scene. Test images are marked with their respective letters in Fig. 6.
7.2 Outdoor Tests

We also performed tests in outdoor environments in order to measure the performance of the algorithm with sunlight. In outdoor scenes, our system works as expected since the sunlight comes from an infinitive direction. The main problem with this approach is having an overcast day where clouds cover the intensity of the sun. Cameras available in mobile environments use auto-adjustments for exposure settings and for now, we can only obtain the camera parameters via EXIF information embedded in JPEG files. As a parameter for global light intensity, we use exposure values from embedded image information and define exposure in HDR lighting in inverse-proportion. Mapping between camera and HDR exposure levels are not exact representations. Therefore, it is expected that we have some inconsistencies in object lighting due to overcast weather conditions.

The difference between intensity images that have been captured in different time of the day, sun position and intensity can be seen in Fig. 7. Among these images we compared most problematic one which is sun light coming from the back side of our user's head. Since our approach depends on the illumination information of the user’s face, we cannot get a strong light direction vector in these scenarios. As in overcast weather condition, our system uses the general light directions like we receive full diffuse light. System also decreases the overall lighting factor which gives us a less lit scene. This case can be seen in Fig. 8. Since our approach does not use back facing camera as a light source estimation, we cannot accomodate partial shadow information in the scene. Resulting illumination is correct only if the object is in full shadow.

Virtual objects lit by our method have ambient occlusion from visibility factor used in SHL. We also use exposure property in SHL definition to increase realism by simulating light differences between full sunlight and overcast day (See Fig. 10). As we extract light direction information from human face, it allows us to modify light direction changes in real time.

### Table 2

| Image 1 | Image 2 | Image 3 |
|---------|---------|---------|
| φ | θ | φ | θ | φ | θ |
| R | 69.74° | 139.26° | 69.74° | 139.47° | 59.79° | 168.88° |
| A | 47.25° | 228.27° | 63.46° | 133.21° | 71.35° | 172.43° |
| E | 22.49° | 89.01° | 6.28° | 13.74° | 11.56° | 3.55° |

| Image 4 | Image 5 |
|---------|---------|
| φ | θ | φ | θ |
| R | 71.21° | 147.52° | 72.05° | 147.91° |
| A | 75.4° | 145.31° | 83.34° | 151.65° |
| E | 4.19° | 2.21° | 11.29° | 3.74° |

7.3 Discussion on System Performance

Detecting outdoor lighting from a single image can have many side effects depending on the weather conditions. We have tested and demonstrated our method in strong, mid and low direct sunlight. Performance in different weather conditions like snow, storm or sky covered with dense clouds was not evaluated so there is no data available for those scenarios. However, our system is capable of extracting lighting direction in a mobile augmented reality scenario. Sensor results are measured very precisely for testing purposes and any kind of drifting in sensor results are neglected during tests. Since our assumption is that the user looks directly to our mobile device screen, software cannot detect a correct direction in case there is a head rotation in another z-axis. In a totally overcast weather our software’s rate of error increases dramatically in azimuth direction (See Table 2 Image 1 θ value).

This is a result of full diffuse lighting in the sky which distributes light evenly so that extracting a direction is generally impossible. Exposure level is an indication of such scenarios where we detect a large standard deviation in light
KOC and BALCISOY: LIGHT SOURCE ESTIMATION IN MOBILE AUGMENTED REALITY SCENES BY USING HUMAN FACE GEOMETRY

Fig. 9  Performance comparison for mobile devices.

Fig. 10  Virtual Objects rendered in different lighting conditions. Color values are similar to the grass colors; our green object have similar brightness.

8. Conclusion and Future Work

In this paper, we proposed a new technique on how to extract lighting information from a real scene by using human face geometry. We also proposed a complete system to handle augmented reality illumination in outdoor scenes. Our system is built on top of most common open-source libraries like OpenCV and ARToolkit. These libraries have been ported to different mobile platforms which allows our system to be integrated easily to different hardware. Our system also allows mobile application programmers to use recent hardware in mobile devices for real-time augmented reality applications.

Our work is a novel method to capture a simplified illumination model in real-time. This model can be extended to more complex lighting calculations by using a robust 3D face tracker system.

In this paper, we aimed to propose a real-time light extraction method specifically tuned for mobile devices. This system is easy to integrate into different platforms while requiring minimal hardware.

References

[1] J.E. Swan II and J.L. Gabbard, “Survey of user-based experimentation in augmented reality,” Proc. 1st International Conference on Virtual Reality, 2005.
[2] B.K.P. Horn, “Obtaining shape from shading information,” in The Psychology of Computer Vision, chapter 4, pp.115–155, 1975.
[3] B.K.P. Horn and M.J. Brooks, “The variational approach to shape from shading,” Computer Vision, Graphics, and Image Processing, vol.33, no.2, pp.174–208, 1986.
[4] M.J. Brooks and B.K.P. Horn, “Shape and source from shading,” in Shape from Shading, ed. B.K.P. Horn and M.J. Brooks, pp.53–68, MIT Press, Cambridge, MA, 1989.
[5] A.P. Pentland, “Linear shape from shading,” Int. J. Comput. Vis., vol.4, pp.153–162, 1990.
[6] L.S. Yoichi, Y. Sato, and K. Ikeuchi, “Illumination distribution from shadows,” CVPR99, pp.306–312, 1999.
[7] T. Jensen, M.S. Andersen, and C.B. Madsen, “Real-time image based lighting for outdoor augmented reality under dynamically changing illumination conditions,” GRAPP’06, pp.364–371, 2006.
[8] M. Kanbara, M. Kanbara, and N. Yokoya, “Geometric and photometric registration for real-time augmented reality,” IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR), p.279, 2002.
[9] P. Debevec, “Rendering synthetic objects into real scenes: bridging traditional and image-based graphics with global illumination and high dynamic range photography,” Proc. 25th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH ’98, pp.189–198, 1998.
[10] S. Tominaga and N. Tanaka, “Spectral measurement of ambient lighting and its application to image rendering,” MVA, pp.554–557, 2002.
[11] M. Kanbara and N. Yokoya, “Real-time estimation of light source environment for photorealistic augmented reality,” Proc. 17th International Conference on Pattern Recognition, Cambridge, United Kingdom, pp.911–914, 2004.
[12] J.D. Yoo and K.H. Lee, “Real time light source estimation using a fish-eye lens with nd filters,” International Symposium on Ubiquitous Virtual Reality, vol.0, pp.41–42, 2008.

Directions and the exposure level is slower than a direct sunlight scene. In the case where we have a large standard deviation and intensities in the image are not as high as direct sunlight configuration, we decrease the used SH coefficient bands to two where the first band defines ambient color and the second band defines diffuse color. Having more bands allows us to capture more detail for both shadow and lighting; however, in full overcast scenes, the detail needed for lighting coefficients is less.

Rendering engine uses only vertex coloring for illumination, we tested our default 3D model scene which contains a plane, torus and sphere in five different mobile devices. Rendering performance comparison can be seen in Fig. 9. Light extraction stage works in a separate thread so it does not effect rendering performance in multicore devices.
[13] N. Tsunura, M.N. Dang, and Y. Miyake, “Estimating the directions to light source casting curved shadows on curved surfaces using images of eye for reconstructing 3d human face,” Color Imaging Conference ’03, pp.77–81, 2003.

[14] K. Nishino and S.K. Nayar, “Eyes for relighting,” ACM SIGGRAPH 2004 Papers, SIGGRAPH ’04, pp.704–711, 2004.

[15] H. Wang, S. Lin, X. Liu, and S.B. Kang, “Separating reflections in human IRIS images for illumination estimation,” Proc. Tenth IEEE International Conference on Computer Vision - volume 2, ICCV ’05, pp.1691–1698, Washington, DC, USA, 2005.

[16] R. Basri and D.W. Jacobs, “Lambertian reflectance and linear subspaces,” IEEE Trans. Pattern Anal. Mach. Intell., vol.25, pp.218–233, Feb. 2003.

[17] K.-C. Lee and B. Moghaddam, “A practical face relighting method for directional lighting normalization,” International Workshop on Analysis and Modeling of Faces and Gestures, p.6, 2005.

[18] A. Watt and M. Watt, Advanced Animation and Rendering Techniques, Addison-Wesley Professional, 1992.

[19] J.D. Foley, A. van Dam, S.K. Feiner, and J.F. Hughes, Computer graphics: principles and practice, 2nd ed., Addison-Wesley Longman Publishing, Boston, MA, USA, 1990.

[20] J.M.V. Verth and L.M. Bishop, Essential Mathematics for Games and Interactive Applications, 2nd ed., Elsevier, 2008.

[21] L. Williams, “Casting curved shadows on curved surfaces,” SIGGRAPH Comput. Graph., vol.12, pp.270–274, Aug. 1978.

[22] J.T. Kajiya, “The rendering equation,” SIGGRAPH Comput. Graph., vol.20, pp.143–150, Aug. 1986.

[23] R.L. Cook, T. Porter, and L. Carpenter, “Distributed ray tracing,” SIGGRAPH Comput. Graph., vol.18, pp.137–145, Jan. 1984.

[24] N. Durlach and A.S. Mavor, eds., Virtual Reality: Scientific and Technological Challenges, ch.9, pp.304–361, 1995.

[25] P.-P. Sloan, J. Kautz, and J. Snyder, “Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments,” ACM Trans. Graphics, vol.21, pp.527–536, July 2002.

[26] P.-P.J. Sloan, J.D. Hall, J.C. Hart, and J. Snyder, “Clustered principal components for precomputed radiance transfer,” ACM Trans. Graphics, vol.22, pp.382–391, 2003.

[27] R. Green, “Spherical harmonic lighting: The gritty details,” Archives of the Game Developers Conference, March 2003.

[28] S. Inversions, “Facegen,” June 2012. http://www.facegen.com/

[29] AbhinayE, “Face detection using opencv,” June 2012. http://opencv.willowgarage.com/wiki/FaceDetection

[30] J. Lopez-Moreno, S. Hadap, E. Reinhard, and D. Gutierrez, “Light source detection in photographs,” CEIG 2009, pp.161–168, 2009.

[31] Wikipedia, “HSL and HSV,” June 2011. http://en.wikipedia.org/wiki/HSL_and_HSV

[32] E.A. Khan, E. Reinhard, R.W. Fleming, and H.H. Bülthoff, “Image-based material editing,” ACM SIGGRAPH 2006 Papers, SIGGRAPH ’06, pp.654–663, 2006.

[33] J.C. Dunn, “A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters,” J. Cybernetics, vol.3, no.3, pp.32–57, 1973.

[34] J.C. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms, Kluwer Academic Publishers, Norwell, MA, USA, 1981.

[35] X. Zou, J. Kittler, M. Hamouz, and J.R. Tena, “Robust albedo estimation from face image under unknown illumination,” Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, vol.6944 of Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, March 2008.

[36] A.J. Preetham, P. Shirley, and B. Smits, “A practical analytic model for daylight,” Proc. 26th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH ’99, pp.91–100, 1999.

[37] R. Habel, B. Mustata, and M. Wimmer, “Efficient spherical harmonics lighting with the preetham skylight model,” in Eurographics 2008 - Short Papers (K. Mania and E. Reinhard, eds.), pp.119–122, Eurographics Association, April 2008.

[38] E. Reinhard, M. Stark, P. Shirley, and J. Ferwerda, “Photographic tone reproduction for digital images,” Proc. 29th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH ’02, pp.267–276, 2002.