Energy and Quality Trade-offs for Augmented Reality Systems

Emily Port
Winthrop University
Rock Hill, South Carolina, USA

Jinhan Hu
Arizona State University
Tempe, Arizona, USA

Robert LiKamWa
Arizona State University
Tempe, Arizona, USA

Abstract

Augmented reality is a growing field with lots of potential applications. To access the full potential of augmented reality, the objects and the scene need to look real which can not happen without the proper lighting. GLEAM is an illumination estimation framework that can light virtual objects so that they look like they belong in the real world. The framework utilizes a reflection probe to generate radiance samples and then create a cubemap from that. The resolution of the cubemap impacts the latency and power consumption of the framework along with the quality of the lighting. High-resolution cubemaps result in high-quality estimations, but the update times are longer and consume more power. Low-resolution cubemaps result in low-quality estimations that sacrifice accuracy and realism for speed and low energy consumption. By finding a balance between energy consumption and realism we aim to make illumination estimation more accessible for mobile devices.

To find this balance, two trade-offs have been proposed: adaptive resolution change and sub-sampling. By adaptively changing the resolution, the aim is to fix quality issues at the smallest resolution possible. If the cubemap resolution is not correct for the situation, the cubemap will not render all six sides. This creates artificial darkness that removes the realism of the object. The second trade-off is sub-sampling which can reduce latency by 38% without significant drops in quality. This method reduces the energy consumption and latency of high-resolution cubemaps with almost identical lighting results. With some more refinement, the hope is that these findings will pave the way for real-time illumination estimation to be a practical reality for AR on mobile devices.

CCS Concepts: • **Augmented Reality** → **Illumination Estimation**; • **Trade-offs**;

Keywords: Illumination Estimation, Resolution, Trade-offs, Augmented Reality

Figure 1. The GLEAM framework is being used to render a dragon statue mesh with environment aware lighting. This example demonstrates a high quality illumination estimation with full coverage of the cubemap. The GLEAM configuration that was used for this study displays the probe sample that is used for calculations along with the cubemap that is rendered using the probe sample.

1 Introduction

Augmented Reality (AR) has many applications that we have only just scratched the surface on even 20 years after its conception. It can be a medical tool to assist doctors and surgeons, a tool for architects to see a prototype of their work, or simply a way for friends to connect by playing a mobile game. The applications of AR range all over from life-saving technology to a cool gaming experience. This technology has many benefits, but there are a few drawbacks that prevent it from being accessible.

AR has many challenges it still has to overcome such as limited field of view and limited hardware capabilities just to name a few. The lighting in AR is equally important as the lighting is a major factor in the realism of a virtual object. Humans are able to perceive the slight differences that let us know if something is real are fake surprisingly well. Oftentimes people will not be able to explain how they know it is a virtual object instead of a fake one. However, a main culprit is often lighting. People are quick to notice that the shadows in a scene do not match the angle of the light source or if an object is not lit with the same color lighting that is in the rest of the room. Without proper lighting of virtual objects, it becomes difficult for people to use AR to its fullest extent. To combat this, illumination estimation methods have been developed including GLEAM[2]. GLEAM uses a reflection probe to create lighting and reflections on virtual objects that match the lighting of the environment. This technique vastly improves the realism of virtual objects in AR.

However, this ability comes at a cost. AR by itself consumes a lot of resources by continuously running the camera while also analyzing the image in real-time to create an overlay that matches the environment. By also adding in the calculations that are required to replicate the lighting of the environment as well, the power consumption is increased. The average person interacts with AR technology on a mobile device that has a small battery that can not run an application like this for a long time [3]. This is why many mobile AR applications sacrifice quality such as realistic lighting in order to lessen battery consumption. That loss of quality often breaks the magic of the AR application rendering its use unnecessary for the user [2].

To extend battery life related works tried changing rendering configurations based on the scene being rendered, using local cubemaps, and using spherical harmonics [4][1][5]. Changing rendering configurations helped reduce the power consumption of the rendering pipeline by predicting what would be the minimum quality setting the scene could render at without losing significant quality [4]. This technique was applied to full scene rendering, and it allowed the application to not waste excess energy rendering at a higher quality that provided minimal improvements [4]. Local cubemaps were also found to be an efficient way to render lighting for a scene [1]. While this is one of the most effective ways to map lighting and reflections, little has been done to improve the energy efficiency of this method. Lastly, spherical harmonics have been used to enhance the ambient lighting of an AR scene to improve the realism with fewer resources [5].

These ideas can be applied to rendering lighting estimation for AR. By combining these techniques, AR applications have the potential to maintain a similar level of quality while decreasing the energy drain.

By characterizing the existing works, the goal is to adaptively adjust cubemap resolution in order to balance energy consumption and performance of AR. This will allow realistic AR applications to be more accessible to users using small handheld or wearable devices with small batteries.

2 Related Work

Illumination estimation for AR has been studied intensively with many different solutions. The GLEAM framework from METEOR Studios is one example of an illumination estimation solution that renders realistic lighting of virtual objects in real-time [2]. GLEAM works by using radiance samples from a real-life reflection probe to generate a cubemap of the scene [2]. Their study found that raising cubemap resolutions slowed down the update intervals but increased the quality of the illumination estimation. The cubemap resolution is also connected to the probe sample size that is taken. The higher the resolution, the more pixels that are captured from the camera image to generate the cubemap. This means that the probe that radiance samples are collected from takes up less of the sample image as the resolution size goes up. Therefore, to get the most accurate results, the resolution needs to be the one that most fills in the probe sample image.

Xihe from Worcester Polytechnic Institute is another illumination estimation framework for mobile augmented reality systems, but it utilizes spherical harmonics along with machine learning and LiDAR technology in order to match the lighting of the environment [6]. This framework estimates the lighting of an environment well, but it requires communication with a remote server which can be draining on a mobile device [6]. It also requires hardware that is not yet available on most mobile devices at the time of writing [6]. Despite these drawbacks, Xihe utilizes techniques that can be leveraged to accurately replicate lighting in a lighter framework.

Adaptive frameworks for the sake of energy conservation have also been studied. Zhang et al.'s work resulted in an adaptive rendering framework based on the best energy-saving techniques that did not make significant sacrifices in quality [4]. The application starts with an energy budget and eliminates any of the rendering configurations that go beyond the set budget [4]. From there the framework selects the configuration with the lowest quality error and uses it to render the scene [4]. However, this framework was not created for mobile devices that have fewer resources to devote to these calculations. The rendering for AR is also smaller and typically less complex than a full scene rendering on a computer, so while using computational power for these calculations will be outweighed by the saved energy on a computer, a mobile device will not reap the same benefits. However, ideas from this framework can be leveraged and combined with other works previously mentioned to create a lightweight framework that will balance energy and quality needs for a given scene.

3 Methods of Measurement

The study focused on ways to improve the efficiency of the GLEAM system by first examining different cubemap resolutions. For the study, the resolutions that were used were 16 pixels, 32 pixels, 64 pixels, 128 pixels, and 256 pixels. Resolutions higher and lower than these were found to not be compatible with the GLEAM system and were therefore deemed irrelevant for the mission of the study. For the resolutions that were used, tests were run to examine power consumption, latency of critical functions, and perceptual quality.

3.1 Power Consumption

The voltage and the current of the phone can be accessed using the Android Debug Bridge (ADB). The voltage is returned in microvolts and the current is returned in microamps. Using a bash script run inside of the ADB shell, the voltage

Figure 2. From left to right are the results of using GLEAM with resolutions of 16 pixels, 32 pixels, 64 pixels, 128 pixels, and 256 pixels. The cubemap that is used for each resolution is shown in the top right hand corner of each image. In the left hand corner is the image of the probe that is being used to generate the cubemap.

and current were read every 3 seconds for a minute while GLEAM was running.

Using the formula to convert the voltage and current readings into power, we were left with an answer in picowatts.

$$P = V * I \tag{1}$$

To make the data easier to analyze the power measurements were converted to microwatts. The power measurements were then averaged together to get an overall reading for each cubemap resolution.

Along with current and voltage readings, CPU usage readings were taken during tests. Tests lasted long enough for at least twenty cubemaps to be generated which was around one minute long. The CPU readings were taken using Android studios and were recorded in percentages. For each test, the mean CPU reading was found.

3.2 Latency

The main functions that make GLEAM work are composeCubemap() and generateRadianceSamples(). The radiance samples are taken from the image of the probe in the latter function and then they are applied to a cubemap to be used by the Unity engine in the former function. These functions perform the main operations of GLEAM and therefore are indicators of the latency and performance of GLEAM.

To test the latency of these functions, print statements indicating the start and end times in milliseconds were added at the beginning and end of each function. The time was also had the frame start time subtracted from it so that each measurement took the time of a new frame into consideration. By subtracting the start time of each function from the end time the total time for the function to complete was found. For each function, twenty samples were recorded and averaged to represent the overall latency of the function for each resolution.

3.3 Perceptual Quality

For examining the quality there are two important factors to consider: the accuracy of the lighting and the completeness of the cubemap. When a face of the cubemap does not have a texture, it is rendered on the object as black. This causes the object to look very dark in images where the cubemap is not fully rendered. It appears as if the object is heavily shadowed which automatically detracts from the realism and accuracy of the lighting. The completeness of the cubemap is therefore the most critical factor as it is the first requirement for accurate lighting.

In figure 2 the different resolutions and the cubemaps that are generated are visible. The lower resolutions are much darker as the cubemap could not fully render at this resolution from this distance. For consistent testing, the mobile device was put on a tripod so that each test would be from the same distance and in the same lighting conditions.

4 Baseline Results

Each metric that was tested tended to have a positive correlation with the cubemap resolution. The higher the cubemap resolution the higher the latency and power consumption. This relationship is best demonstrated in 3 which shows the latency for the composeCubemap function for different cubemap resolutions. The latency increases at a linear rate as the cubemap resolution is increased. However, after 256 pixels for the resolution, the latency greatly increased to the point where a cubemap was not rendered after 10 minutes. due to this finding, all of the testing was done up to 256 pixels.

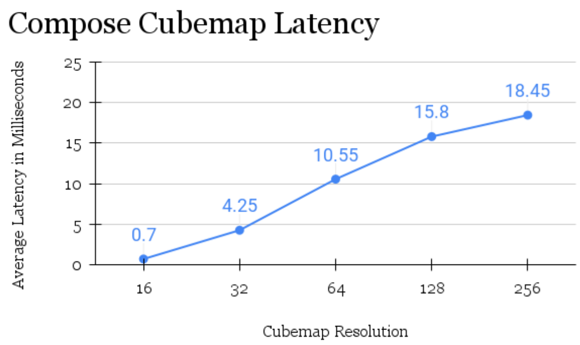


Figure 3. The cubemap resolution has a positive correlation with the latency of the function that creates the cubemap.

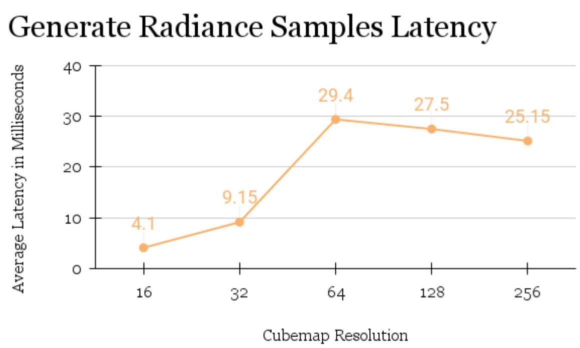


Figure 4. The cubemap resolution does not have a solid or obvious relationship with the latency of the function that generates radiance samples. However, there is a significant increase in the amount of time it takes to complete the task when the resolution is higher than 32 pixels.

The outlier for the baseline testing was the latency of the function that generates the radiance samples. For this function, the latency peaked at 64 pixels and leveled out for resolutions higher than this. After further investigation, the leveling out of the latency was determined to be caused by a radiance samples limit implemented in the code. The standard GLEAM configuration had the radiance samples limited to 8000. Therefore, all 8000 radiance samples were able to be collected starting at 64 pixels which left no room for growth in latency at higher resolutions. Future works could explore the impacts of changing this limit more in-depth and examine how cubemap coverage also affects energy consumption.

For the power usage of different cubemap resolutions as shown in 5, there was an outlying spike in power usage for the 16 pixel resolution compared to the other resolutions that showed a constant positive correlation. This spike can be attributed to the speed at which the 16 pixel resolution is able to update. When the power consumption and the

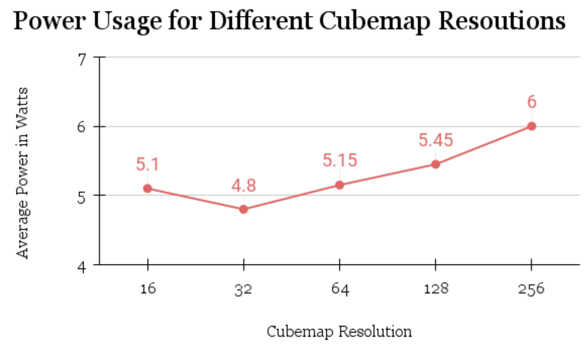


Figure 5. The cubemap resolution has and almost perfect positive correlation with the power usage. The only outlier is the smallest resolution which can possibly be attributed to the constant updating that is possible at this size.

latency of the 16 pixel resolution are multiplied to find the energy consumption, the amount of energy being saved by the 16 pixel resolution becomes more clear. However, this drop in energy use is due to the lack of information that the 16 pixel resolution is able to provide and utilize. While the energy use is promising, the results are practically useless.

During testing, it was found that the camera distance from the image target greatly impacted the performance of the resolution size. While a resolution of 16 pixels was essentially useless, the other resolutions were able to generate all six sides of the cubemap depending on their distance. The lower resolutions such as 32 pixels and 64 pixels required the camera to be far away from the image target in order for the entire reflection probe to be included in the probe image sample. For higher resolutions such as 256 pixels, the best results occurred when the camera was closer to the reflection probe so that the extra pixels were used to give more information about the probe instead of being wasted.

5 Approach

Resolution refers to the number of pixels in an image. Typically, resolution is represented in PPI which refers to points per inch for print and pixels per inch for screens. Since AR uses the camera input for its calculations, the resolution of the image stays the same. Therefore, when discussing changing the resolution of GLEAM, it means that the number of pixels collected from the camera image is increased. The amount of information the camera receives does not change based on the GLEAM resolution instead more pixels are taken from the existing pixels in the image are taken for the probe sample image. This means that when the GLEAM resolution is higher, the pixels that represent the probe compromise less of the overall probe sample image. Therefore, more calculations are performed at higher resolutions on pixels that do not contain information on the probe and are hence useless. Using a higher resolution with the GLEAM system is not useful unless the entire probe is not sampled at the current resolution.

Including the entire probe in the probe sample image is the most important factor in composing a complete and accurate cubemap. With the traditional way of changing the GLEAM resolution, a change to a lower resolution will crop out pixels with valuable information that is necessary to complete all six faces of the cubemap. While having too large of a resolution results in wasted energy consumption, too small of a resolution is detrimental to the quality.

5.1 Sub-sampling

One approach to solve the issue of reducing energy consumption without sacrificing vital image information is to use sub-sampling. Sub-sampling involves using a smaller sample of the full set of pixels to run calculations on. This method allows the system to reduce the total number of calculations it has to perform for each update of the cubemap.

Having pixels cropped out of the probe sample image completely omits all of the information that was available from that part of the probe. Sub-sampling eliminates the need for cropping while still reducing the number of pixels that are processed. Even though there are fewer pixels being used to represent each part of the probe, it is better than having no pixels for a section of the probe.

The sub-sampling that was conducted on the GLEAM system involved only evaluating every other sampling from the client. This sub-sampling did not discriminate between samplings from different parts of the probe.

5.2 Adaptive Resolution Change

Using the information that was gathered in testing, we were able to modify the GLEAM framework to adapt to the needs of the environment. Since the resolution size and the probe sample size are directly connected, the larger the resolution is, the smaller the probe appears in the sample image. When the probe does not take up enough of the image the same issues are caused that occur when the probe takes up too much of the image. In both cases, the cubemap is not able to fully form which causes artificial darkness to be rendered on the virtual object. This ruins the quality of the lighting and makes the efforts of the GLEAM framework null thereby rendering any of the extra power consumption for the task a waste.

To combat this waste of resources, the modified framework will adjust the resolution and therefore the probe sample size according to the needs of the scenario. The main factor that determines the need for a larger or smaller resolution is the camera's distance from the probe. A smaller resolution is needed when the camera is far away from the probe while a larger resolution is needed when the camera is close to the probe.

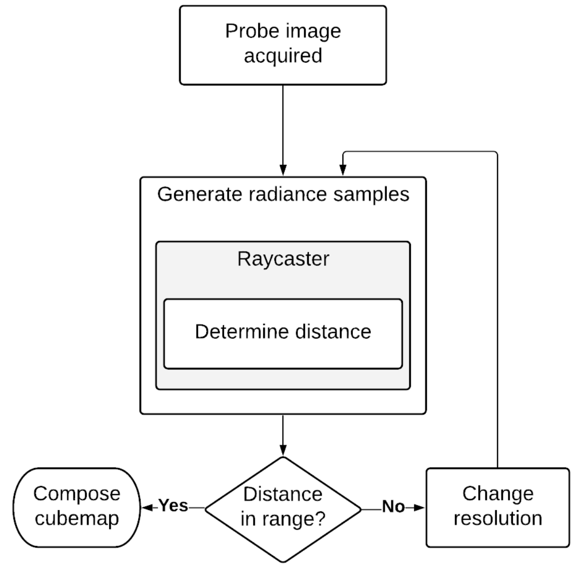


Figure 6. A raycaster is used to get samples from the probe image. By comparing the number of times the ray hits the probe and misses, the system can estimate how much of the probe is visible. If the ratio of hits to misses is not in the correct range, the resolution will change.

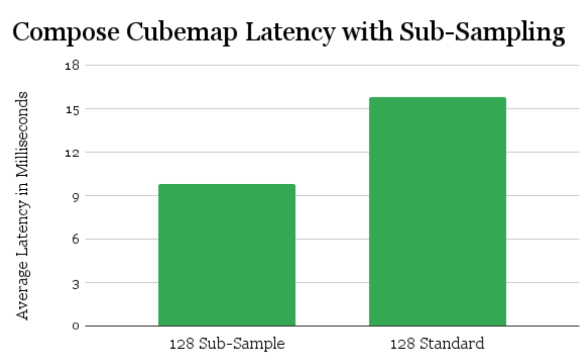
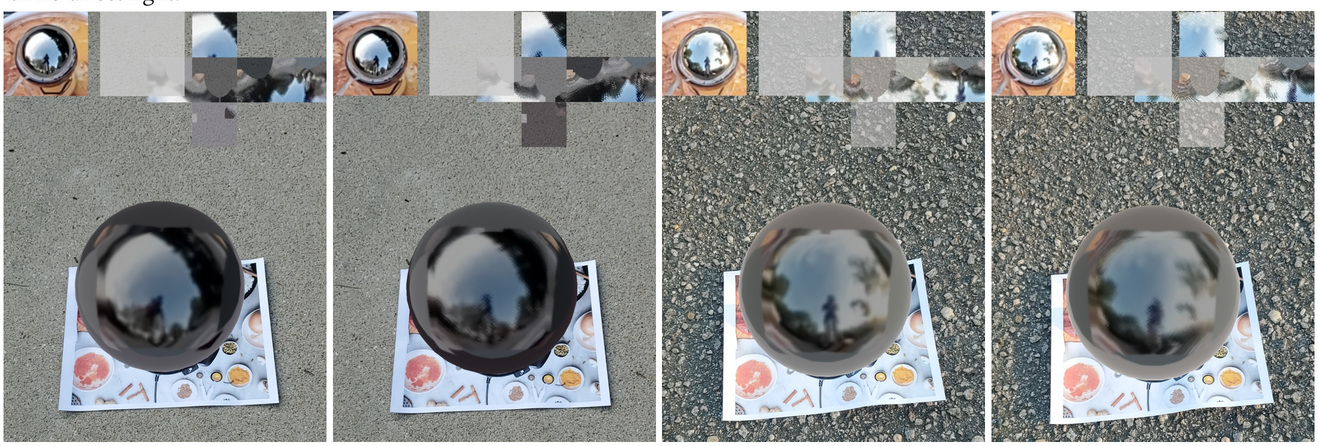


Figure 7. The sub-sampled illumination for a resolution of 128 pixels averages a 38% decrease in latency for the function that composes the cubemap compared to the standard configuration.

In order to judge the need for a change in resolution, the modified framework takes advantage of the number of rays that hit and miss the probe when raycasting. The system takes the numbers of hits and misses and checks to see if the ratio between the two is in a certain range. If the ratio is not within the range, then the cubemap resolution is changed.

(a) The first two images are in a indoor setting under a direct artificial light. The two images on the right are also in an indoor setting but with no direct light.



(b) These four images are in an outdoor environment. The two images on the left are in partial shade with indirect lighting. The two images on the right are in full, direct sun.

Figure 8. The first image for each lighting situation is the standard illumination estimation with the GLEAM system at a resolution of 128 pixels. The second image for each lighting situation is the sub-sampled illumination estimation with the GLEAM system at a resolution of 128 pixels. The standard estimations use 8000 radiance samples to compose the cubemap while the sub-sampled estimations only use 4000 radiance samples.

6 Results

Sub-sampling helped significantly improve latency for the function that composes the cubemap when half of the full set of radiance samples are used. The average latency of the sub-sampled configuration was 38% lower than the standard configuration that utilized the full set of the radiance samples.

Sub-sampling by a factor of 3 did not reduce the latency any more and it resulted in visual quality errors. A rough texture was formed by the pixels that were skipped. This gave the cubemap and the virtual object a checkerboard effect. From this finding, it was determined that sub-sampling can be done at most by a factor of 2 for a resolution of 128 pixels in order to see energy savings without significant quality loss.

The quality of the cubemaps and the reflections were not significantly impacted by sub-sampling half of the radiance samples. This can be seen in greater detail in 8. Sub-sampling did not appear to impact the ability to render all six faces of the cubemap as both configurations were able to fully render a cubemap when put in the same conditions. The reflections that were produced are also very similar with no obvious drops in quality.

For the adaptive resolution change, the proposed system was successful in changing the resolution. However, using the built-in distance variable would probably be more consistent and reliable than evaluating the probe sample. This idea does show promise in making the system use only what it needs. After more refinement, this idea can be used to resolve the quality issues that occur from not having enough samples to compose a complete cubemap. By combining this idea with sub-sampling, energy and resources can be saved even when higher resolutions are necessary for the situation.

7 Future Work

This work has many potential areas of growth that can further improve the quality and energy consumption of the GLEAM system.

7.1 Sub-Sampling Algorithm

Due to the spherical nature of the reflective probe, the edges of the probe contain more information than the center of the probe. Therefore, using a discriminating sub-sampling algorithm will enable the system to include the valuable samples from the edge of the probe while omitting more of the less valuable samples in the center of the probe. This idea may allow for higher factors of sub-sampling without the quality loss that is currently seen when the sub-sampling factor is higher than 2.

7.2 Spherical Harmonics

Another area to examine would be the addition of spherical harmonics to the GLEAM system in order to account for ambient lighting. Spherical harmonics would be used to add in a directional light to be added without utilizing many additional resources [5]. Quality loss may potentially be compensated for my adding in spherical harmonics which would enable GLEAM to use lower-quality configurations and reduce overall energy consumption.

8 Conclusion

Proper illumination is crucial for AR to be used to its full potential. More solutions are being formulated to deal with this issue of creating realistic lighting for AR. However, these solutions come with an additional problem which is the increased energy drain. For mobile devices with small batteries, this increased energy usage keeps proper illumination from being practical. However, by implementing sub-sampling and adaptive resolution change, a balance is able to be found between realism and energy consumption.

References

[1] Roberto Lopez Mendez and Sylwester Bala. 2016. Optimized Mobile Rendering Techniques Based on Local Cubemaps. In *ACM SIGGRAPH 2016 Posters* (Anaheim, California) *(SIGGRAPH '16)*. Association for Computing Machinery, New York, NY, USA, Article 35, 2 pages. https://doi.org/10.1145/2945078.2945113

[2] Siddhant Prakash, Alireza Bahremand, Linda D. Nguyen, and Robert LiKamWa. 2019. GLEAM: An Illumination Estimation Framework for Real-time Photorealistic Augmented Reality on Mobile Devices. In *Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services* (Seoul, Republic of Korea) *(MobiSys '19)*. ACM, New York, NY, USA, 142–154. https://doi.org/10.1145/3307334.3326098

[3] Sooeun Song, Junsung Kim, and Jong-Moon Chung. 2019. Energy Consumption Minimization Control for Augmented Reality Applications based on Multi-core Smart Devices. In *2019 IEEE International Conference on Consumer Electronics (ICCE)*. 1–4. https://doi.org/10.1109/ICCE.2019.8661917

[4] Yunjin Zhang, Marta Ortín-Obón, Victor Arellano, Rui Wang, Diego Gutierrez, and Hujun Bao. 2018. On-the-Fly Power-Aware Rendering. *Comput. Graph. Forum* 37, 4 (2018), 155–166. https://doi.org/10.1111/cgf.13483

[5] Yiqin Zhao and Tian Guo. 2020. PointAR: Efficient Lighting Estimation for Mobile Augmented Reality. In *Computer Vision – ECCV 2020*, Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (Eds.). Springer International Publishing, Cham, 678–693.

[6] Yiqin Zhao and Tian Guo. 2021. Xihe: A 3D Vision-Based Lighting Estimation Framework for Mobile Augmented Reality. In *Proceedings of the 19th Annual International Conference on Mobile Systems, Applications, and Services* (Virtual Event, Wisconsin) *(MobiSys '21)*. Association for Computing Machinery, New York, NY, USA, 28–40. https://doi.org/10.1145/3458864.3467886