# Blinking LEDs, a potential solution to improve Augmented Reality interactability

**Giovanni Lin**
**Supervisor: Miguel Chavez Tapia**
**Professor: Marco Zúñiga Zamalloa**
**EEMCS, Delft University of Technology, The Netherlands**

**A Dissertation Submitted to EEMCS faculty Delft University of Technology,**
**In Partial Fulfilment of the Requirements**
**For the Bachelor of Computer Science and Engineering**

**June 18, 2022**

## Abstract

Augmented Reality has the potential to expand our interaction with our surrounding environment. A potential solution to improve this interactability is Visible Light Communication through blinking Light Emitting Diodes (LEDs). Data is encoded and then transmitted by blinking the LED, a smartphone camera is then pointed at the LED to decode the message from the blinking light. The light must blink at a high enough frequency, otherwise the blinking LED causes flicker which has negative health risks. In this research, an experiment is conducted to check the viability of this technology, by transmitting a message to multiple phones with different recording frame rates and checking whether the technology functions adequately for each frame rate. The viability is checked through multiple factors, with the most important one being flicker, due to the negative health risks. If there is no flicker, the throughput of the LED at that frame rate and blinking frequency is checked at multiple distances. The results show that only the highest recording frame rate used in the experiment is viable and that the ones below all cause flicker. This makes it hard to consider the technology viable, due to it only functioning on a limited amount of smartphones.

## 1 Introduction

Augmented Reality (AR) has the potential to expand our interaction with our surrounding environment. An example of this is Google Maps AR [6], which overlays the navigation route on to the real physical world. This improvement of interactability can enhance daily life, by making information access a lot easier. However, current day devices that support AR technologies treat appliances that can be encountered in daily life as non-interactive objects. This can be changed by using Visual Light Communication (VLC) methods, like tagging these devices with tags such as ARTags [2] or Quick Response (QR) codes [3] (Figure 1). However, these are obtrusive, do not naturally occur on these devices and thus have to be added onto the appliance. Studies [1], [9], [15] have shown that VLC can be done in a less obtrusive way, making appliances interactable through Light Emitting Diode (LED) lights on these appliances. LEDs are pervasive on daily life appliances which makes them a great candidate for the improvement of AR interactability. Data is encoded and then transmitted by rapidly blinking an LED light, a phone camera can then be pointed at this rapidly blinking light and information will appear on the phone screen in an AR environment.

There is, however, a problem with this system. The light must blink fast enough that the phone camera can pick it up, while the human eye should not be able to notice. If the light blinks too slowly, the human eye will see the flashing and this will result in flicker, which has negative health implications [10], [13]. Studies on this topic like LightAnchors [1] and InfoLED [15] have only tested cameras with a high video recording frame rate (120 and 240 Frames Per Second (FPS)), which allows for the light to blink unnoticeably fast while transmitting data at a sufficient speed. GLITTER [9] has also studied the subject, however GLITTER uses a low recording frame rate camera of 60 FPS, which does not allow for the light to blink unnoticeably fast, thus having potential health risks. With this in mind the choice is made to not compare the results of this project with those from GLITTER [9]. Section 3 explains further why the focus of this project lies with LightAnchors [1] and InfoLED [15].

Figure 1: Example of QR codes and AR, showing a full 3D model of the camera [3].

The goal of this paper is to find out whether blinking LEDs are a viable solution to AR interactability improvement. By testing if data can be transmitted from LED to camera on a wide ride range of smartphones with high recording frame rate cameras to low recording frame rate cameras, at a high enough data rate, at a practical viewing distance, without causing flicker. With binary transmission the light blinking rate is limited to the cameras frame rate. This may cause problems with cameras that can only film up to 60 FPS, because according to [10] the threshold for humans to notice flicker lies between 60 and 90 Hz and according to [13] as well, any flashing between 3 Hz and 70 Hz (flicker) can have negative health implications. Therefore, any blinking between or up to these frequencies could be potentially dangerous.

Everything is presented in the following structure. Section 2 lays out the methodology. Section 3 presents the contribution of this paper. Section 4 explains how the experiment is set up and done. Section 5 lays out the results from the experiment. Section 6 outlines the ethical aspects involved in this project. Section 7 discusses and reflects upon the project. Finally, Section 8 concludes the paper and presents potential future work.

## 2 Methodology

This research tests whether the work done by [1] and [15] is applicable to a broader spectrum of smartphones. However, before the experiment can be conducted, some methods and concepts that are used in this work have to be explained. These are: On-Off Keying (OOK), Manchester Code and why flicker should be avoided.

**On-Off Keying:** This is the modulation method used to encode the data and was also used by LightAnchors [1] and InfoLED [15]. OOK is a simple form of modulation that is done by changing the amplitude of the signal. This is then combined with binary transmission to encode a message and transmit it from the blinking LED. Simply said, if the amplitude is low then this represents a binary ‘0’ and if the amplitude is high this represents a binary ‘1’. A visual representation of this can be found in Figure 2. For the blinking LED, this then translates to light off representing ‘0’ and light on representing ‘1’. This would be binary OOK without Manchester Code. The inclusion of that is explained in the next part.

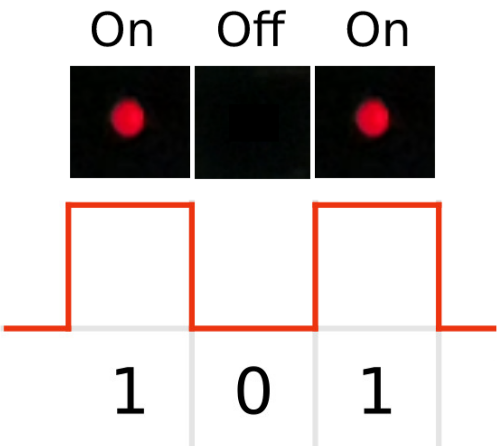


Figure 2: How the receiver detects the LED being ‘On’ and ‘Off’.

**Manchester Code:** For Manchester Code, instead of representing ‘0’ and ‘1’ as a low edge and a high edge in the amplitude, the transition from low to high and high to low is used. There are two ways of interpreting these changes as ‘0’ and ‘1’, but for this paper low to high is used for ‘0’ and high to low is used for ‘1’. This then translates to LED off to on being ‘0’ and LED on to off being ‘1’. A visual representation of this can be found in Figure 3. Manchester Code was also used in the work of [15] to encode the data.

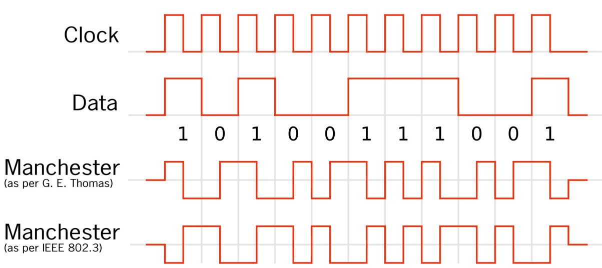


Figure 3: Two methods of Manchester Code, this paper uses first method [8].

There are advantages and disadvantages to using Manchester Code with blinking LEDs. The advantage is that OOK with Manchester Code will always have a minimum blinking frequency, while OOK without it could have an extended upper or lower edge, this is a repeated sequence of ’1111...’ or ’0000...’. An extended edge can lower the blinking frequency and increase the risk of flicker, due the the light being on and off for longer, thus lowering the blinking frequency. The disadvantage of this, is that this halves the data rate of the blinked LED, due to having to use two flashes of the light to encode one bit of data. However, for this research the choice was made to use OOK with Manchester Code, due to flicker prevention being an important factor, which is explained in the next part.

**Flicker:** Visual flicker is when a light source blinks at a certain frequency where it is noticeable by humans, but the brain can not process what is happening. This occurs in a range from 5 Hz to 60 or 70 Hz, and for some individuals even up to 90 Hz [10], [13]. This can have negative health effects such as headaches, nausea, eye strain, reduced visual performance and more [10]. Flicker can be especially endangering towards individuals with photosensitive epilepsy. According to [10], seizures can be triggered by blinking up to 70 Hz, with 15 to 20 Hz being the most sensitive area. Due to these health risks the choice was made to use Manchester Code to have a minimum blinking frequency. If the minimum blinking frequency is outside of this range, flicker should be avoidable.

## 3 Contribution

What must also be laid out before the experiment can be conducted is the contribution of this paper. As mentioned before, LightAnchors [1] and InfoLED [15] have conducted their research on phones with 120 and 240 FPS cameras. In this paper, these are considered as phones with high frame rate recording capabilities. The goal of this research is to test whether the technology of binary transmission through blinking LEDs is also applicable to a broader spectrum of phones. Phones that do not have cameras that are able to film at 120 FPS, but lower, thus being a phone with low frame rate recording capabilities. The work from [9] uses a 60 FPS camera with 30 Hz blinking, which is deep into the noticeable flicker area according to [10] and [13]. Combined with that their paper also does not mention flicker, while LightAnchors [1] and InfoLED [15] state they mostly avoid flicker. The goal of this paper is to match the blinking speed to the frame rate of the camera and gather data from this. This way, combined with Manchester Code, flicker should be the most preventable. The goal is then to compare the results of this paper with those who mention to mostly prevent flicker, which GLITTER [9] is not part of.

The method of VLC is only seen as viable or applicable to this broader spectrum of phones if the following conditions are met. First of all, the transmission data rate should be high enough. OOK with Manchester Code should give a data rate of half the blinking and camera frame rate, an example of this would be 240 Hz blinking that should give 120 bits per second. For this experiment, it should not take more than three seconds to transmit a message of five or less ASCII characters. Secondly, the LED should be able to be scanned at a practical distance. The further the camera is away from the LED the higher the chance that a video frame is interfered, which can result in dropped bits and error bursts. This paper tests the ability to scan the LED up to 2 meters. For dropped bits, there must not be more than 1 bit error every 10 bits, otherwise it would be hard to use bit error detection methods. Nor should it take too many message cycles to scan the LED. If there is visual flicker, then the scanning of the LED on that frequency is aborted and the blinking frequency is deemed unviable. The expectation is that 60 Hz blinking causes flicker, while 120 and 240 Hz blinking do not cause flicker.

## 4 Experiment Setup

The experiment is conducted using two different phones, a low end and a high end phone. An iPhone 12 is used as the high end phone and a Xiaomi Redmi Note 10 is used as the low end phone. The iPhone can record at four different frame rates, namely 30, 60, 120 and 240 FPS. For this experiment, only 240 FPS is used. The Xiaomi can record from 30 up to 120 FPS, but for this experiment only 60 and 120 FPS are used. Video at 240 FPS is used as a baseline, since [1] and [15] also used 240 FPS and showed working results. 120 FPS on the Xiaomi is used as high recording frame rate, but on a different phone. 60 FPS on the Xiaomi is used as low recording frame rate.

To modulate and blink an LED, an Arduino is used. This is an Arduino Due with a shield on it that contains an LED. Through software that can be uploaded to the Arduino, functions are made that parse bits to Manchester code and then a string of bits can be blinked by calling these functions. The blinking frequency is adjustable by changing a global value that controls the millisecond delay between light flashes. The blinking frequencies are matching to the frame rate of the camera that is filming, so 60 Hz for 60 FPS, 120 Hz for 120 FPS and so on. The message that is transmitted is "hello", this is converted to binary using the ASCII encoding scheme and then this message is blinked through the Arduino's software. Due to the clock of the LED and the clock of camera not being synchronous, there is a chance that the frame capture timing of the camera does not properly align with the flashing of the LED. Instead, the timing then aligns with the transition of the LED, where it captures the light going from 'Off' to 'On' or vice versa. This means the camera does not capture a clear 'On' or 'Off' state of the LED, which must be avoided since this gives decoding problems. This is fixed by putting a fixed delay between each message cycle, which offsets the timing of the LED to be back in sync with the frame capture timing. For this experiment this delay is the length of 1.3 symbols, so at 60 Hz this would be 1.3/60th of a second. A visual representation for clarification can be found in Figure 4.

Scanning of the LED is done using video recordings and then post image processing. This is done in separate steps due to the scale of the project, making the process less complicated due to time being limited. Before starting the recording, the blinking frequency is checked for flicker by myself and fellow students that were present during the experiment. After the flicker check, the recording phone is put on a tripod and then the blinking LED is filmed at 0.5, 1 and 2 meters of distance. Figure 5 contains an example of how the experiment is conducted. Due to the scale of the project, video recording will also be done in a dark space to minimize interference. See Section 7 for a more in depth explanation on the

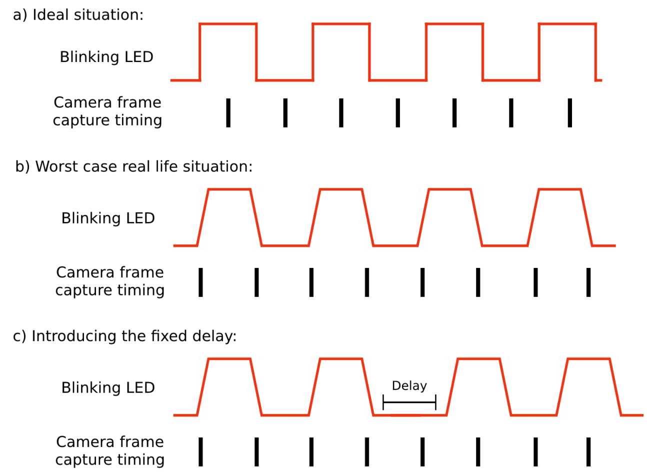


Figure 4: The ideal situation can be found in 4a, where the camera perfectly aligns with an 'On' or 'Off' state of the blinking LED. The black bar represents when the camera captures a frame. In 4b it can be seen that in real life the LED does not instantly transition, namely it takes a small amount of time. In the worst case scenario the camera frame capture timing aligns with this transition period, which means the frame captured is not a true 'On' or 'Off' state. In 4c the delay is introduced to fix this bad timing. After the delay, the timing of the frame captures is more ideal, actually capturing a true 'On' or 'Off' state.

limitations of this project.

After all videos are recorded, these are uploaded to a computer where they are processed using OpenCV. OpenCV is an image processing library which can be used with Python where video information can be processed frame by frame. This is ideal for the processing of the imagery from videos, since for each frame an area of interest has to be identified and then information from this area has to be demodulated. Using OpenCV, a static area of interest is selected and then the colour of this area is checked. The LED blinks a red light, so from this area of interest the average RGB value is taken and then for each frame the R-value is saved. Frames and their data are then paired for the Manchester Code, with each pair representing a bit. A bit string is then made out of these frame pairs. If the first frame has a higher R-value than the second frame, the light goes from "On" to "Off", which demodulates to a binary '1' which is then added to the string. If the second frame is brighter than the first, it goes from "Off" to "On", which demodulates to a binary '0' in the string. If the pair is neither of these, then this bit is considered a dropped bit and demodulates to an 'x' in the string. This string is then checked to see whether the message was successfully transferred, this is done by matching sections of the bit string form the LED with the bit string of the message. without exceeding the interference limit that was defined in Section 3.

## 5 Results

This section first discusses blinking at 60 and 120 HZ with the performance of the Xiaomi phone and it's viability. Afterwards, blinking at 240 Hz and the iPhone's performance is

Figure 5: An example of filming with the Xiaomi at half a meter.

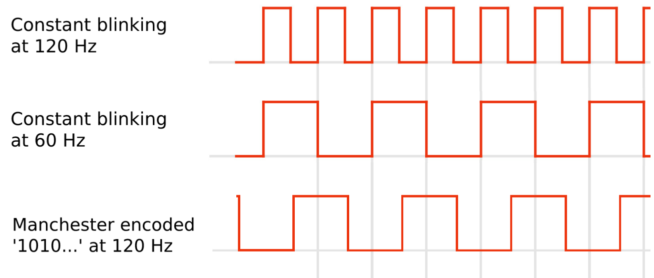


Figure 6: First graph is a depiction of constant blinking at 120 Hz. Second graph is a depiction of constant blinking at 60 Hz. The third and last graph shows the binary string "1010..." at 120 Hz.

analysed. Lastly, a comparison to previous research is made and the over all viability of LEDs and AR interactability is considered.

As expected, blinking at 60 Hz causes visual flicker. When blinking the LED at 60 Hz four bystanders confirm seeing visible rapid non stop blinking at this frequency after looking at the LED for 2 to 3 seconds, therefore 60 Hz is deemed unviable. When blinking the LED at 120 Hz minor flicker is also noticeable, short bursts of visible rapid blinking is observed. Multiple bystanders confirm seeing visible blinking at short intervals, after looking at the LED for 4 to 5 seconds. This is due to the minimum blinking frequency of 120 Hz Manchester encoded blinking being 60 Hz. How this works is that when a binary string is blinked in a Manchester encoded way, the light will go on and off at different intervals. When the string is for example '0000...' this translates to Manchester encoded light blinking "Off, On, Off, On, etc.". Then setting the blinking frequency to 120 Hz results in a consistent 120 Hz blinking. However, when the string is for example '1010...', this translates to "On, Off, Off, On, etc." which are extended upper and lower edges. At 120 Hz combined with the extended edges results in a blinking speed which boils down to 60 Hz, which is in the range of visible flicker. In Figure 6 a visual representation can be found. This means that any transition from '1' to '0' or vice versa, or any extended sequence of '10...' causes flicker at 120 Hz. Which means that blinking LEDs at 120 Hz are unviable, because extended exposure can result in the negative health effects caused by visual flicker. Which means the Xiaomi is completely unviable, due to it not being able to capture camera footage at a high enough frame rate.

Blinking at 240 Hz did not have any flickering problems, this is also confirmed by bystanders. The iPhone was able to receive packets from the LED at 0.5 meters and at 1 meter, but not at 2 meters. The LED is so small at 2 meters that any minor tremor or shaking of the camera would bring the static area of interest out of focus, which would then interrupt the bit stream. While the camera was held still by a tripod, it would still shake due to environmental factors, such as people walking around or vehicles passing by. This could be fixed by having a dynamic tracking algorithm to track the LED on the screen, such as InfoLED [15], however this is outside of the scope of this project. At 0.5 meters, the iPhone is able to pick up 4 to 5 packets out of 17 with a Bit Error Rate (BER) of 5% to 7%, which means it takes around 3 to 5 message cycles to receive the message and it has about one dropped every 15 to 20 bits. Sometimes an extra bit is detected in between the message or a section of the message's bit string got flipped, this is then caused by jittering of the blinking LED. There are also occasional short error bursts with a lot of dropped bits, this is then caused by heavy shaking or bigger tremors, as mentioned before caused by environmental factors. Due to the short distance, the LED takes up more space on the screen, which in turn then makes it harder for the LED to go out of focus of the area of interest. This make it almost unaffected by small shaking or minor tremors. At 1 meter, the iPhone was also able to pick up the 2 to 3 packets out of 18 with a BER of 7% to 10%, which means it takes around 6 to 9 message cycles to receive the message and it has about one dropped bit every 10 to 15 bits. Short error bursts were more occasional due to the LED taking up a smaller space on the screen, thus being more sensitive to shaking and tremors. This can again be fixed by having a dynamic tracking algorithm to track the LED on screen, which would greatly reduce the amount of errors.

Combining these results, it can be seen that blinking LEDs are not applicable on a broader spectrum of phones. Blinking at high frequencies is required to use this method of VLC flicker free and not all phones have this capability. This can be seen from the Xiaomi phone, only being able to film up to 120 FPS, which can only scan LEDs that blink with visual flicker. Like previous research [1], [15] a phone with 240 FPS camera, in this case the iPhone 12, can do this flicker free. However the results of this paper do not align with the work of LightAnchors [1] at 120 Hz blinking and of InfoLED [15] at 30 Hz blinking, due to the strict criteria of this report for when the technology is considered viable. LightAnchors [1] only mentions flicker once, stating that blinking at this rate is generally imperceptible, but that it depends on the message sent. InfoLED [15] states that they maintain a blinking frequency from 30 to 60 Hz and that their test with users in most cases will not notice flicker. This paper says that any amount of flickering is deemed unviable, due to the health risks from potential extended exposure. Therefore, the results of these papers, stating that they mostly avoid flicker, do not align with the results of this paper.

What must also be mentioned is that [1] and [15] using more dynamic detection methods and demodulation algorithms, thus they are able to reach further distances with less errors. However, their work is reproducible, even in smaller scales like this project. But, as mentioned before, both papers use blinking frequencies which have been identified as flicker frequencies by this paper, as well as the work from LightAnchors [1] not using Manchester Code which could cause even more flicker. Nevertheless, the technology only works without health risks at 240 Hz blinking or higher, thus requiring 240 FPS cameras or better and due to not all smartphones having these capabilities, it is hard to consider this method of VLC as a viable solution to improve AR interactability in the current day.

## 6 Responsible Research

This section is divided into two parts. The first part discusses the ethical implications of this report. The second part lays out the reproducibility of this research.

This paper raises one main ethical issues, caused by the blinking LEDs. This ethical issue is also one of the main topics in this paper, namely visual flickering and it's negative health implications. As mentioned in [10] and [13] flicker can be especially dangerous towards people with photosensitive epilepsy. Blinking frequencies from 3 Hz up to 70 Hz can trigger seizures, with highest likelihood in the range of 15 Hz to 20 Hz. Next to this, [10] also mentions that flicker can have negative health effects on people without photosensitive epilepsy. This includes eye strain, malaise, nausea, reduced visual performance, panic attacks and anxiety. For this reason, during the experiment, recording is not done when the blinking frequency has visible flicker. It is unethical to expose myself and potential bystanders to this flickering for an extended period, due to the negative health risks.

With the details provided in Section 4 others should be able to recreate the setup of the experiment and then obtain comparable results. However, there are factors that influence the reproducibility of the results, these are:

- Environmental factors which made the camera shake.
- Asking bystanders to look at the LED for flicker.
- Jittering of the LED.

First of all, people walking by or vehicles passing by that cause tremors which make the camera shake affect the results in such a way that could worsen them, especially at longer recording distances. Capturing footage in a more calm environment could improve results and capturing footage in a busier environment could worsen results. Next, the bystanders that look at the LED for flicker could have different visual capabilities, with one person being more sensitive to flicker than the other. Lastly, the jittering of the LED, which happens randomly. More jittering during the experiment could worsen the results and less jittering could potentially improve them.

## 7 Discussion

This section is divided in to multiple topics and they are be presented in the following order. First of all, the scale of the project is discussed, why certain choices are made due to its scale. Next, an elaboration on why the preference of this paper goes to single point binary transmission through LEDs instead of using higher blinking rates with the rolling shutter effect. Lastly, the definition of high recording frame rate versus low recording frame rate is discussed.

### Scale of the project

The time for this project is only 8 to 9 weeks, therefore some choices have been made to deliver a complete product instead of being too ambitious and then failing to deliver. These choices are mostly surrounding the experiment, to optimize the setting, so it gave less problems when gathering results. This includes:

- Capturing footage in a dark room, to easily discern the LED from the background, which in turn makes the demodulation process a lot easier.
- Using a tripod instead of holding the phone in your hand. The tripod keeps the camera in a static position which makes it easier to take a static area of interest and demodulate that area. When holding the phone in your hand the camera is not static, which makes getting a static area of interest a lot harder. This would require an algorithm that tracks the LED on screen like LightAnchors [1] and InfoLED [15], however this is outside of the scale of this project, due to the time it takes to develop such an algorithm.
- Do post processing of the video instead of live processing on the phone during the capturing of footage. Having to create an app or a program which grabs the live video feed from the camera, processes the image, demodulates it and then puts out information on screen would make the process a lot more complicated. To record footage of the blinking LED, transfer the video footage to a computer and then process this footage using image processing libraries, makes the process a lot less complicated. This would also require using an Android phone that supports 240 FPS recording, which was not within the resources of the project.

There was one thing that could have made the process even less complicated which was using smaller data packets, instead of sending a whole string of 40 bits. The paper of InfoLED [15] mentions that smaller packets are easier to transmit and which is applied in the work from GLITTER [9] and LightAnchors [1]. GLITTER [9] uses packets of 18 bits, with 12 bits of data and LightAnchors [1] uses packets of 16 bits, with 10 bits of data.

### Rolling Shutter and other methods

Another way of transmitting data from LED to camera is using the Rolling Shutter effect of the camera. Most phones have Complementary Metal Oxide Semiconductor (CMOS) cameras. CMOS cameras capture images by scanning rows of pixels in a sequential order from top to bottom, thus not capturing each row simultaneously. This can result in distortions on the image when trying to capture rapidly moving objects. Combine this with a rapidly blinking LED, that blinks at a higher rate than the camera's frame rate and this results in an image with horizontal bars (Figure 7).

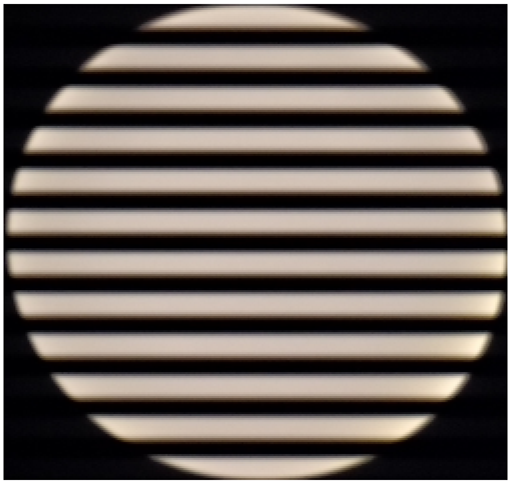

Figure 7: Image capture of an LED blinking at 1 KHz [12].

When an LED blinks a modulated message, this pattern of stripes will change on the captured frame and this pattern can than be demodulated into a message. The advantage of using this technique is being able to always blink the light at a speed which is outside of the human range for noticeable flicker, being able to support phones with lower recording frame rates, such as 30 and 60 FPS, and achieving high data rates due to each frame being able to capture many bars, thus capturing many bits. However, this technique also has its disadvantages, such as it always requiring a big LED, which is not pervasive, minor interruption during video capture due to shaking or tremors could interfere with a whole frame of data, resulting in the image not being as clean as in Figure 7, which can make it a lot harder to demodulate, and this technique is very limited in distance the LED can be scanned. Both DynaLight [11] and Luxapose [4] go up to 1.2 meters with data rate dropping off the further the distance. In the work of [12], this can also be seen, where the experiment goes up to 3 meters with the data rate dropping off the further away the LED is captured. It can also be seen with Visual Light Landmarks [7] which got results up to 8 meters, but the data rate dropping off the further away from the light.

This is less of a problem with single point binary transmission where the data or symbol rate is consistent, but going further away from the LED means higher BER. InfoLED [15] is able to go up to 7 meters, not mentioning BER in their results but dropped packets. They are able to receive about 80% of packets successfully up to 5 meters and 50% up to 7 meters. LightAnchors [1] has been tested up to 12 meters, with a BER that averages at 5.2% which is about 1 dropped bit for every 20 bits. GLITTER [9] is able to go up to 35 meters, with almost no bit errors up to 30 meters and a BER of about 9% to 10% at 35 meters which is 1 dropped bit every 10 to 11 bits. However, Rolling Shutter should not be disregarded, due to its applicability in other contexts. Take for example Luxapose [4], which uses the Rolling Shutter effect with pictures to determine the indoor localization of a person through a smartphone. The smartphone takes pictures of the LED which makes it barely affected by interruption due to shaking or tremors, since it is not capturing video and indoor localization does not require very long distances.

Other works like Epsilon [5] and CurveLight [14] use a different method of VLC, namely LED to light sensor, with the light sensor attached to a phone. Both works being indoor localization methods transmitting data from the LED to the light sensor to determine the location of a person through a smartphone. While they are able to use blinking frequencies outside of the noticeable flicker range for humans, it defeats the purpose of having readily available off-the-shelf hardware. The goal of this research is to use hardware that can be found in our daily lives, like almost everyone having a smartphone and small LEDs being present on almost every appliance that can be encountered in our daily lives.

### High versus low

For this research the choice was made to consider 120 and 240 FPS as high recording frame rates and 60 FPS as low recording frame rate, due to 240 FPS being the highest available frame rate for capturing footage. 120 FPS is also considered high due to LightAnchors [1] showing working results with 120 FPS, and 120 Hz should theoretically lie outside of the noticeable flicker threshold. This, however, is not the case, as explained in Section 5. Nowadays there are phones with recording frame rates up to 1920 FPS, which could be used for even higher data rates, with unnoticeable blinking. However, these phones are not common and not within the resources or budget of this project. 60 FPS is considered low due it being hypothesized for causing flicker, which is indeed the case, as seen in Section 5 and nowadays 60 FPS is becoming more and more common as a baseline for video footage. 30 FPS with 30 Hz blinking could have also been checked for this project, however this is far below the noticeable flicker threshold. 60 Hz is more interesting to check, due to it lying closer to the threshold limit for visible flicker.

## 8 Conclusion

This paper presents the research that has been done on the viability of blinking LEDs as potential solution to improve AR interactability. An experiment was conducted to check whether a broad spectrum of phone cameras is able to pick up a message from a blinking LED, without having visual flicker. Visual flicker is preferably avoided due to it having negative health implications [10], [13], thus any blinking that causes flicker is deemed unviable. As it turns out, any blinking frequency below 240 Hz causes flicker, thus any blinking rate below that is unviable following the definitions of this paper. Due to only one frame rate not causing flicker, it is hard to consider the technology as a whole viable until 240 FPS cameras have become more common. In the future, when cameras with frame rates of 240 FPS and above are the baseline, this experiment can be repeated to see whether the technology has become viable.

## References


[1] Karan Ahuja, Sujeath Pareddy, Robert Xiao, Mayank Goel, and Chris Harrison. LightAnchors: Appropriating point lights for spatially-anchored augmented reality interfaces. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*, pages 189–196, 2019.

[2] Mark Fiala. ARTag, a fiducial marker system using digital techniques. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 590–596. IEEE, 2005.

[3] Tai-Wei Kan, Chin-Hung Teng, and Wen-Shou Chou. Applying QR code in augmented reality applications. In *Proceedings of the 8th International Conference on Virtual Reality Continuum and its Applications in Industry*, pages 253–257, 2009.

[4] Ye-Sheng Kuo, Pat Pannuto, Ko-Jen Hsiao, and Prabal Dutta. Luxapose: Indoor positioning with mobile phones and visible light. In *Proceedings of the 20th annual international conference on Mobile computing and networking*, pages 447–458, 2014.

[5] Liqun Li, Pan Hu, Chunyi Peng, Guobin Shen, and Feng Zhao. Epsilon: A visible light based positioning system. In *11th USENIX Symposium on Networked Systems Design and Implementation (NSDI 14)*, pages 331–343, 2014.

[6] Andrew Liptak. Google is letting some users test its AR navigation feature for google maps. *The Verge*, Feb 2019.

[7] Niranjini Rajagopal, Patrick Lazik, and Anthony Rowe. Visual light landmarks for mobile devices. In *IPSN-14 proceedings of the 13th international symposium on information processing in sensor networks*, pages 249–260. IEEE, 2014.

[8] Stefan Schmidt. File:Manchester encoding both conventions.svg. Wikimedia Commons. https://en.wikipedia.org/wiki/File:Manchester_encoding_both_conventions.svg, 2006. Accessed: 2022-06-04.

[9] Rahul Anand Sharma, Adwait Dongare, John Miller, Nicholas Wilkerson, Daniel Cohen, Vyas Sekar, Prabal Dutta, and Anthony Rowe. All that glitters: Low-power spoof-resilient optical markers for augmented reality. In *2020 19th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pages 289–300. IEEE, 2020.

[10] UNIOS. Driving the Flicker-Free Effect, 2019.

[11] Michail Vasilakis. Dynalight: A dynamic visible light communication link for smartphones. 2015.

[12] Shashwat Verma. Analysing the performance and stability of led-to-camera links. 2017.

[13] Arnold Wilkins, Jennifer Veitch, and Brad Lehman. Led lighting flicker and potential health concerns: Ieee standard par1789 update. In *2010 IEEE Energy Conversion Congress and Exposition*, pages 171–178. IEEE, 2010.

[14] Shangyao Yan, Zhimeng Yin, and Guang Tan. Curvelight: An accurate and practical indoor positioning system. In *Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems*, pages 152–164, 2021.

[15] Jackie Yang and James A. Landay. InfoLED: Augmenting LED indicator lights for device positioning and communication. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*, pages 175–187, 2019.