LICS - Light Paint Coding System

Mobile Device Based Information Encoding and Decoding Using Light Paintings

Kai Röhr* Sönke Momsen* Tobias Weiss*
Daniel Burmeister† Bashar Altakrouri† Andreas Schrader†
Institute of Telematics, University of Lübeck, Germany (all three columns)

ABSTRACT

Light painting is a photographic technique, in which images are created using long exposures from a scene containing one or multiple light sources that move relative to the camera. We demonstrate an approach to encode and transmit information in such a painting via sequences of colored light blobs using smartphone devices.

1. INTRODUCTION

In many areas of information transmission, visual encoding is used to transport data in a more compact, transformed, and fault-tolerant way. A specific method to carry such visual encoded data with temporal dependency is optical telegraphy. Examples for such kind of data transmission include the Morse code or smoke signals. Beyond this, few use the phenomenon of persistence of vision [1], which allows for conversion of some spatio-temporal information into just spatial information. Current examples include the ”LilyPad Wristband POV” [2] or the iOS-Application ”SwingWords” [3], which use this principle to transmit text messages. In the realm of photography, persistence of vision can be achieved using long-exposure photographs and various moving light sources. As an art form, this is called “light painting”. With the ever growing pervasiveness of smartphones, nearly everybody now has the tools available to create such light paintings. Not only are those devices capable of capturing light, but the screen can also be made into a paint brush by using it as a light source. If visual information is provided in such a time independent way, color-coding may increase the data density. This can be seen in the ”High Capacity Color Barcode” [4] or photoTAN [5], which is already used by a number of banks. Inspired by the works of [6] to create artistic codes, we created a proof-of-concept to show that this can also be done in the realm of photography. Conceivable applications would be advertising or multimedia postcards.

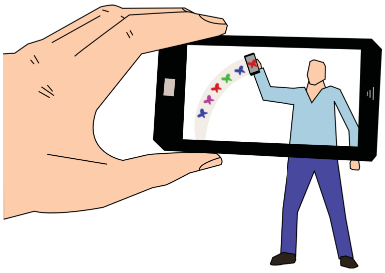


Figure 1: Schematic illustration of the painting process and setup using two smartphone devices.

2. PAINTING PROCESS AND SETUP

This work was performed as a student exercise within the course “Mobile Multimedia Systems” at the University of Lübeck. Our setup involves the usage of two Android devices. Whereas one of them is used as a light source, the other one is used for capturing and decoding. No additional communication or coupling between the devices needs to be done. The user holding the light source device will need to perform some setup though. This involves entering the data to be encoded, choosing the flash symbols, and setting the duration of the flashes as well as the time between them (Fig. 3 (a)). The user is able to adjust brightness, size, and shape of the light flashes. In addition, an approximate time it will take to do the light painting is calculated. After setup, the light painting is created by moving perpendicular to the image plane. The beginning of this motion is detected via the built-in accelerometer and will cause the device to flash in the colors, determined by the encoded sequence. Another smartphone is able to capture and decode the sequence. The painting process is illustrated in figure 1.

3. THE CODE

Our light painting code is based on sequences of colored blobs of light. Based on the limitations of frame size, the encoded sequences should be as short as possible with a maxmium of informational content. Satisfying these requirements is done by limiting our approach to encoding only URLs with a two bit coded color scheme, as they can refer to any content. In order to determine where to start parsing the sequence, we used a dedicated color as the starting symbol. In practice, blobs are not always distinctly separated and can bleed into each other. Avoiding two consecutive blobs of the same color to be mistaken for a single blob is done by inducting a repetition symbol. For example, the bi-

*e-mail: {roehr,momsen,weisst}@informatik.uni-luebeck.de
†e-mail: {burmeister,altakrouri,schrader}@itm.uni-luebeck.de

MOBIQUITOUS 2014, December 02-05, London, Great Britain
Copyright © 2014 ICST 978-1-63190-039-6
DOI 10.4108/icst.mobiquitous.2014.258053

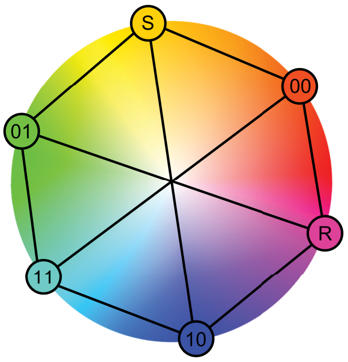
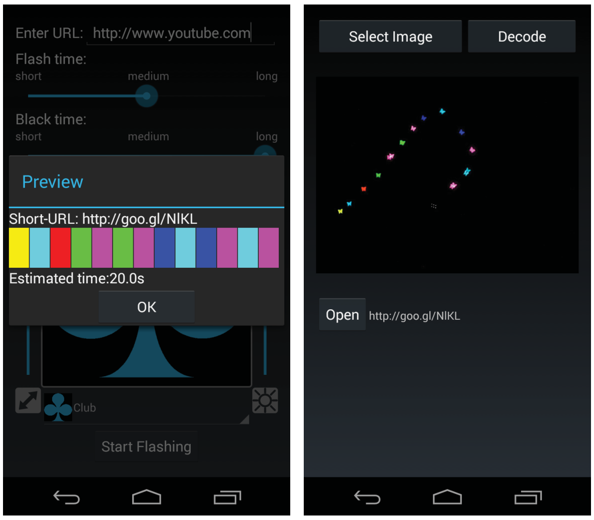


Figure 2: The distribution of colors on the HSV color wheel, where “R” represents the repetition sign and “S” the starting symbol.


Enter URL: http://www.youtube.com

Flash time:

Black time:

Preview

Short-URL: http://goo.gl/NIKL

Estimated time:20.0s

Open http://goo.gl/NIKL

(a) Create activity

(b) Decode activity

Figure 3: Screenshot of the app showing the create and decode activities.

nary sequence 0000 won’t be encoded as “red red”, but rather as “red magenta”, where magenta is the color of the repetition symbol. To provide some robustness towards changes in the color, especially changes in brightness and saturation, we use maximally distant colors from the perimeter of the HSV color wheel (see Figure 2).

Using this method, we were able to encode any URL into a color sequence. However, the 2 bit code still produces very long sequences. To further reduce the code’s length, we utilize a URL shortener like the one provided by Google [7]. The first part of a shortened URL is always identical, so we only need to encode the last few characters. Each character is 6 bits long since only numbers, uppercase, and lowercase letters are used. That way, any character can be represented by three blobs.

4. DETECTION AND DECODING

Decoding a captured lightpainting (see figure 3b) is done by matching each blob of a sequence to the code color scheme using color quantization based on nearest neighbor algorithm. Thereafter, blobs are filtered by the area of their convex hull, the distance between blobs and the similarity of the blobs. By using the repetition symbol, we were able to obtain robustness against side effects like partials occlusion of a blob. Finally, the algorithm determines the starting color and uses a spatial nearest neighbor approach to find the sequence. This way of pathfinding makes the code location, scale, and shape invariant, but it doesn’t offer the possibility for self-intersecting paths. We figured out that even after performing color quantization there are often multiple colors within a single blob. Therefore we determine the color of the blob by using the most prominent one. Due to the fact that Android does not yet provide a possibility to set arbitrary exposure time, we used a workaround provided by the OpenCV [8] framework to get low level video data. Consecutive frames of this video stream are thereby accumulated to create a final decodable image.

5. CONCLUSIONS

In this work, we presented a lightpainting based approach to transmit robust color encoded information as well as capturing and decoding them by taking long exposure images. We observed, that background objects impact the information transmission process as they may be mistaken as parts of the encoded message. Hence our approach achieved the best results in total darkness. Thereupon, we examined a 3 bit encoding approach using more colors, but the low quality of color reproduction in either the display or the camera often leads to confusions in the color quantization process.

6. REFERENCES

[1] Max Wertheimer. Experimentelle Studien über das Sehen von Bewegung. *Zeitschrift für Psychologie*, 61:161 – 265, 1912.

[2] Kylie Peppler and Diane Glosson. Learning about circuitry with e-textiles in after-school settings. *Textile messages: dispatches from the world of E-textiles and education. Peter Lang Publishing, New York, NY.*

[3] Swingwords. `http://www.vidiag.de/swingwords.html`. Accessed: 22.08.2014.

[4] Devi Parikh and Gavin Jancke. Localization and segmentation of a 2d high capacity color barcode. In *Applications of Computer Vision, 2008. WACV 2008. IEEE Workshop on*, pages 1–6. IEEE, 2008.

[5] Max Günther and Bernd Borchert. Online banking with nfc-enabled bank card and nfc-enabled smartphone. In *7th IFIP International Workshop on Information Theory and Practice*. Springer, Heraklion, Greece, May 28–30 2013.

[6] Rupert Meese, Shakir Ali, Emily-Clare Thorne, Steve D. Benford, Anthony Quinn, Richard Mortier, Boriana N. Koleva, Tony Pridmore, and Sharon L. Baurley. From codes to patterns: Designing interactive decoration for tableware. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI ’13, pages 931–940, New York, NY, USA, 2013. ACM.

[7] Google url shortener. `http://goo.gl/`. Accessed: 22.08.2014.

[8] Gary Bradski and Adrian Kaebler. *Computer vision with the OpenCV library*. O’Reilly Media, Inc., 2008.