LightPaintAR: Assist Light Painting Photography with Augmented Reality

Tianyi Wang[1], Xun Qian[1], Fengming He[2], Karthik Ramani[1,2]
[1]School of Mechanical Engineering, Purdue University
[2] School of Electrical & Computer Engineering, Purdue University
West Lafayette, IN, 47907, USA
[wang3259,qian85,he418,ramani]@purdue.edu

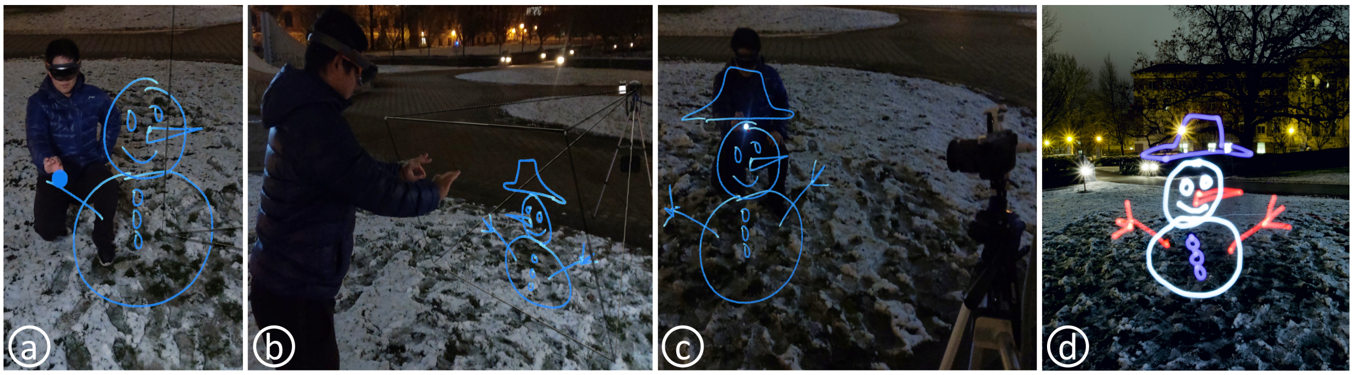

Figure 1: The workflow of creating a light painting photo using LightPaintAR. (a) A user sketches light trajectory in AR. (b) The user adjusts the pose of the trajectory by referring to the camera FOV. (c) The user does light painting by following the AR trajectory with a LED light during exposure. (d) The light painting photo created by the user.

ABSTRACT

Light painting photos are created by moving light sources in mid-air while taking a long exposure photo. However, it is challenging for novice users to leave accurate light traces without any spatial guidance. Therefore, we present LightPaintAR, a novel interface that leverages augmented reality (AR) traces as a spatial reference to enable precise movement of the light sources. LightPaintAR allows users to draft, edit, and adjust virtual light traces in AR, and move light sources along the AR traces to generate accurate light traces on photos. With LightPaintAR, users can light paint complex patterns with multiple strokes and colors. We evaluate the effectiveness and the usability of our system with a user study and showcase multiple light paintings created by the users. Further, we discuss future improvements of LightPaintAR.

CCS CONCEPTS

• **Human-centered computing → Mixed / augmented reality**; **Interactive systems and tools**.

CHI '21 Extended Abstracts, May 8–13, 2021, Yokohama, Japan

ACM ISBN 978-1-4503-8095-9/21/05.
https://doi.org/10.1145/3411763.3451672

KEYWORDS

Light Painting; Augmented Reality

ACM Reference Format:
Tianyi Wang[1], Xun Qian[1], Fengming He[2], Karthik Ramani[1,2]. 2021. LightPaintAR: Assist Light Painting Photography with Augmented Reality. In *CHI Conference on Human Factors in Computing Systems Extended Abstracts (CHI '21 Extended Abstracts), May 8–13, 2021, Yokohama, Japan.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3411763.3451672

1 INTRODUCTION

Light painting is a photographic technique for taking long exposure photos. While the camera shutter is open, photographers move a light source artistically within camera's field of view. The trace of the light is then captured by the camera sensor and becomes "light painting" [1]. Many famous artists and photographers including Man Ray, Henri Matisse, and Gjon Mili have adopted this technique in their works. The art of light painting is still popular and Jan-Leonardo[2], Hannu Huhtamo[3] and Janne Parviainen[4] are among the top contemporary light painting artists.

However, unlike drawing on a tangible canvas, there is no visual feedback in mid-air. Thus, it is difficult for end-users to move the light source along a precise trajectory, especially when they want to draw multiple stokes. Consequently, most light painting photos (Figure 2a, and b) only consist of simple patterns such as circles,

[1]en.wikipedia.org/wiki/Light_painting
[2]lightart-photography.de/en/home-en/
[3]flickr.com/photos/hhuhtamo/
[4]jannepaint.wixsite.com/jannepaint-2

Figure 2: Example light painting photos.

spirals and random curves. And, the complex patterns such as Figure 2c, can only be performed by highly professional artists. Although researchers have been automating the process of light painting kinetically [10, 13] or computationally [16], few tools were developed to assist elementary photographers and hobbyists to shoot light painting photos.

Nowadays, the development of augmented reatlity (AR) blurs the boundary between the virtual and physical world. With an AR head-mounted device (AR-HMD), end-users can directly create digital strokes in 3D space [1, 5, 11]. More importantly, users can leverage AR contents as spatial reference [3, 9] and perform predefined body movement by following the AR guidance [6, 17]. Thus, we are motivated to adopt AR mid-air visualization to guide the movement of the light source while shooting a light painting photo.

To this end, we propose LightPaintAR, an assist tool that enables end-users to first design the light patterns in AR, then follow the AR guidance to create desired light traces in their photos. Before taking a light painting photo, users can freely draw, edit or resize the light patterns in AR (Figure 1a), and place them within the camera's field of view (FOV) (Figure 1b). Upon releasing the shutter, users move a light source by following a visual hint that moves along the pre-drawn AR trail (Figure 1c). In summary, LightPaintAR is an AR application that empowers users to create 3D traces and refer them as the spatial guidance during light painting process.

2 RELATED WORKS

2.1 Assist Light Painting Photography

Conventional light painting process requires a user to manually move a light source in mid-air, which constrains the complexity of the light patterns. To alleviate this problem, multiple methods were explored. Robot arms [7, 8] and drones [10, 15] were utilized to move the light source automatically. However, these methods involves special equipment that may limit the flexibility of photography. Moreover, users may need extra knowledge to customize the light pattern. Meanwhile, Wada et al. [19] used fluorescence string to eliminate the movement of the light source during light painting. But this method was only capable of limited patterns. On the other hand, Salamon et al. [16] proposed a video processing algorithm that replaced the movement of the real light with virtual light traces. Yet, only point light source was supported while more advanced light sources such as glow stick, fiber optic and fire were difficult to be simulated.

Unlike the above methods that introduce new light painting process, LightPaintAR enhances the conventional light painting with spatial references. Original setups, including cameras and light sources, are fully compatible. Therefore, light paintings with LightPaintAR are completed in one shot without any post-processing. Moreover, users can utilize their previous knowledge of photography and reduce learning time.

2.2 AR Contents as Spatial Reference

The emerging AR technique closely binds physical environment and virtual contents through its spatial awareness and visualization [1, 2, 4, 5, 11, 12]. Some works [9, 18] allow users to create virtual contents by referring to physical environment. Contrarily, virtual contents are leveraged as spatial reference to support users' operations such as body movement and object manipulation [3, 6, 17, 20]. Inspired by these works, LightPaintAR enables users to first sketch AR traces and then use them as spatial reference during light painting. By following the AR hints, users can move a light source accurately and leave desired light traces in photos.

3 FORMATIVE INTERVIEWS

To understand photographers' needs and challenges of shooting light painting photos, we interviewed two photographers (both males, aging between 25-30). Both of them have at least two years of professional experience and have applied the light painting technique in some of their photos. During the interview, we encouraged the photographers to talk about their experience and problems that they had while shooting light painting photos, and how they overcame those problems. Through the interviews, we identified the following critical challenges.

1. Move the light source along the planned trajectory. To ensure that the light source moves as designed, a photographer first imagines the trajectory and usually practices for several times to get familiar with the body movements. Then, during the exposure time, he/she moves the light source along that trajectory. However, lack of visual feedback often leads to a significant offset between the imaginary path and the actual light trace.

2. Place the light trace at an appropriate position in the photo. The position of the light trace largely affects the composition of the photo and the aesthetic, at least it should not exceed the camera FOV. However, while standing in front of the camera, a photographer has no visual hint indicating where to draw. Further, while doing a light painting with multiple stokes, it is hard to decide the spatial relation between the next stoke and the previous ones.

3. Move the light source at a steady speed. The speed of the light source movement decides the exposure of the light trace. Moving the light too slow/fast results in overexposed/underexposed traces.

4. Make no mistakes during the painting. Curently, there is no post-editing tools for light painting photos. Any light trace that is left on the camera sensor cannot be erased, so, a tiny mistake may ruin the entire light painting.

4 DESIGN GOALS

Given the feedback from the informative interview, we distilled the following design goals (**DGs**).

DG1. Create and visualize light trajectory in mid-air. Enable users to draw and visualize their imaginary trajectory as spatial reference, so that the users can move the light source along it during the exposure time.

DG2. Edit the light traces. Allow users to correct any mistakes before taking a photo. Users should be able to resize or re-scale the trajectory, or delete a part of it.

DG3. Visualize the camera FOV. Allow users to adjust the position of the light path based on the camera FOV to determine the best photo composition.

DG4. Visual hint of moving speed. A visual hint should be provided to users during the exposure time so that the users can move the light source steadily by following it.

5 LIGHTPAINTAR SYSTEM DESIGN

In this section, we demonstrate the LightPaintAR interface. LightPaintAR is an AR application that runs on a Microsoft Hololens 2 [14]. The light painting photos are shot by a common digital camera. A QR code is placed next to the camera for camera frustum localization in AR (Figure 3a). All the user operations are supported by the hand detection and gesture recognition capability provided by Hololens 2.

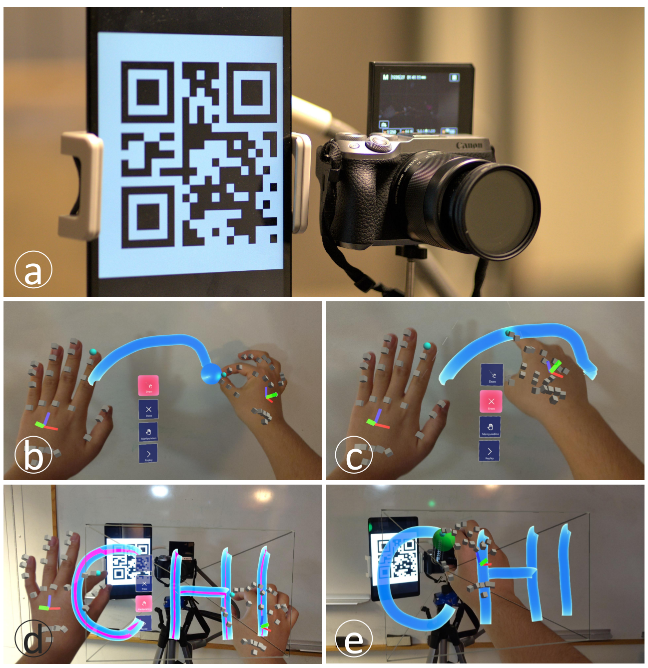

Figure 3: LightPaintAR system design. (a) The digital camera that shoots the light painting photo with a QR code mounted together for AR localization. (b) *Paint* function. (c) *Erase* function. (d) *Manipulate* function. (e) *Replay* function.

LightPaintAR implements four main functions, ***Paint***, ***Erase***, ***Manipulate***, and ***Replay*** (Figure 3b-e). A user can select the functions using a menu which is floating next to the left hand. Using the *paint* function, the user can draw and visualize light trajectories in AR (DG1). The *erase* and *manipulate* functions help the user to delete an unwanted stroke and change the position of the AR trajectory (DG 2 and DG3). After the user is satisfied with the trajectory, he/she can use the *replay* function and move a light source following a moving AR hint to complete a light painting photo (DG4).

Paint function. As shown in Figure 3b, after choosing the *paint* function, a blue sphere, which represents an AR pen tip, appears at the user's index finger. When the user performs pinch gesture and moves the hand, the blue sphere follows the hand and leaves paths in-situ.

Erase function. The user can erase an unwanted stroke by choosing the *erase* function and touch that stroke with the index finger (Figure 3c).

Manipulate function. Using *manipulate* function, the user can perform the pinch gesture with both hands to grab the light paths (Figure 3d). In this way, the user can move, rotate or resize the light paths in 3D. Note that in current implementation, we only allow users to manipulate the light traces as a whole, rather than adjusting strokes individually. An AR frustum which represents the camera FOV is generated based on a QR code attached next to the camera. Users can use the frustum as a reference when they move the light paths. The offset position of the QR code to the camera is manully calibrated. But in future upgrades, we propose to use the camera's screen to display the QR code.

Replay function. During the actual light painting process, to ensure that the light source precisely moves along with the light paths at a steady speed, we provide a sphere as a visual reference (Figure 3e). When the hand that holds the light source is spatially close to the sphere, the sphere moves smoothly along the light paths. If the hand is far away from the sphere (distance $\geq 0.15m$), it turns red and waits for the user to catch up. After finishing one stroke, the user covers the light with another hand and move to the beginning of another stroke. Further, the user can switch the color or type of the light source during the transition from one stroke to another. Note that in current implementation, LightPaintAR is not connected with the camera. A researcher manually releases the shutter as the user selects the *replay* function.

6 PRELIMINARY USER EVALUATION

6.1 Study Setup

We conducted a three-session user study to evaluate the usability of LightPaintAR. The study took place in a 3m by 3m indoor area with no internal light source. Note that the area was not totally dark to allow for Hololens 2 spatial tracking function. During the study, the users were asked to draw light traces using a Lume Cube[5] LED light. The light painting photos were taken with a Canon EOS M6ii camera with a EF-M 11-22mm lens (aperture: f11, shutter: 120s, ISO: 125). A variable ND filter was applied to control the brightness (Figure 3a).

In the first session, the users were introduced to light painting photography. After a three-minute practice, they were asked to light-paint the words "CHI 2021" using the LED light without any AR guidance. In the second session, the users went through a five-minute tutorial about LightPaintAR AR interface. Then, the users drew the same "CHI 2021" with the AR references which were created by themselves using LightPaintAR. The third session was an open-ended study. The users were encouraged to light-paint

[5] lumecube.com

any pattern they liked using LightPaintAR. The entire user study took approximately 40 minutes. Users took a two-minute break between the sessions. After all 3 sessions, the users were asked to finish a 5-point Likert-type survey regarding the main functions of LightPaintAR. Meanwhile, a standard System Usability Scale (SUS) questionnaire and a conversation-type interview were conducted to evaluate the overall user experience.

We invited 6 users (4 males and 2 females, aging from 24 to 33, AVG=26) to the user study. All six users had experience of taking photos with their phones or cameras. But all of them had limited knowledge about light painting. Additionally, four users had experience with AR/VR devices before.

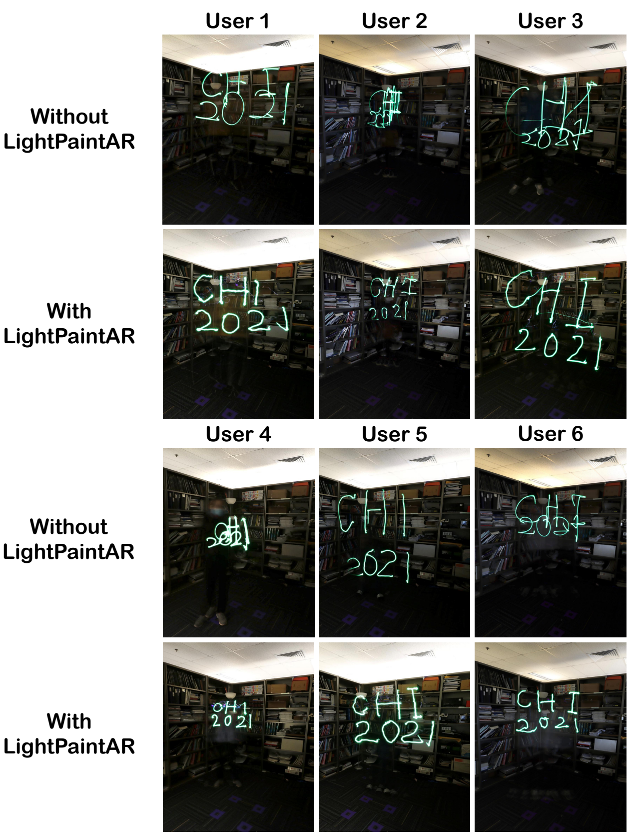

Figure 4: The users' light-painted "CHI 2021" without LightPaintAR (top row) and with LightPaintAR (bottom row).

6.2 Objective Performance

During the first two sessions, all six users were able to finish the light painting without and with LightPaintAR. While using LightPaintAR, all the users could use the four functions without any instructions. Some of the users had difficulties with the Hololens 2 bare hand interactions at the beginning, but soon they got used to it after a few-minute practising. We demonstrate the light-painting photos created in session 1 (Figure 4 top row, without LightPaintAR) and session 2 (Figure 4 bottom row, with LightPaintAR).

As illustrated in the results, most users could not draw the characters properly without any assistance. In the photos created by user 1,2,3,4, and 6, characters overlaid with each other. Meanwhile, user 2,3, and 4 could not accurately separate different strokes and most strokes were connected by a thin light trace. Besides, user 1 and 5 did not put the words in the middle of the photo. On the contrary, the characters drawn with LightPaingAR in session 2 looked more clear and recognizable. Nevertheless, these pictures showed some other problems. The characters drawn by user 3 seemed not parallel to the camera sensor. Also, part of the light traces created by user 2 was dim, which might be caused by not pointing the directional LED light source to the camera lens while drawing. Additionally, some strokes performed by user 1 and user 5 were a little jittery. This phenomenon reveals that not all users were used to the visual reference ball which moves at a fixed speed in *replay* mode. An adjustable moving speed and suggestions on corresponding light source intensity might be preferred.

In Figure 5, we demonstrate a collection of light-paintings created by the users in session 3. On average, the users took 5 to 10 minutes to design and edit their light paths in AR. The users were able to come up with various designs and apply different colors to their paintings.

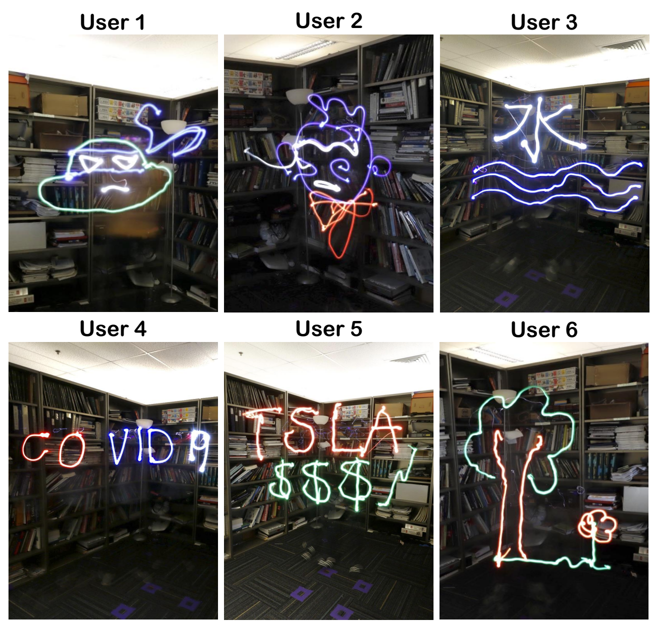

Figure 5: Light-paintings created by the users using LightPaintAR.

6.3 Subjective Ratings and Interviews

The Likert-type ratings regarding the users' subjective feelings are shown in Figure 6. In general, the users agreed that the visual references provided by LightPaintAR were necessary for light painting (Q2, AVG=5.00, SD=0.00) and they were satisfied with the photos created using LightPaintAR (Q1, AVG=4.67, SD=0.52). "*The photo shows exactly what I thought. I don't think I can make a better one without your system.* (User 2)" "*The AR trails are really necessary* *because I can precisely follow it and even resume at the correct position after I switch another LED.* (User 6)" Moreover, most users gave positive feedback towards the interactions of the *paint*, *erase*, and *manipulate* functions (Q3, AVG=3.83, SD=0.98; Q4, AVG=3.83, SD=0.75; Q5, AVG=4.33, SD=0.82). "*After getting used to it, I like the way of drawing and moving my sketches, it's very intuitive.* (User 1)" Meanwhile, the users welcomed the visual hint provided during the *replay* function (Q6, AVG=4.00, SD=1.10). "*I like that moving sphere most, because it explicitly tells me how fast should I move the light.* (User 3)" Finally, in the SUS questionnaire, the users gave an average SUS score of 86.7 (SD=8.4), which indicates a general satisfaction of LightPaintAR interface usability.

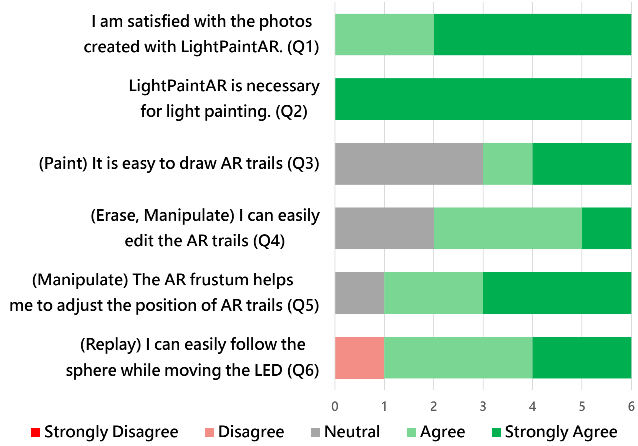

Figure 6: Likert-type survey result of the user study.

7 DISCUSSION AND FUTURE IMPROVEMENTS

Visualization of the AR trails. In current implementation, LightPaintAR only allows users to draw trails in blue color. Although single-color trails are enough as spatial reference, some users mentioned that the AR trails should be color-coded to better support the light painting process. "*I think the color of the stroke should be the same as the LED light I plan to use. So, AR can give me a more authentic preview and remind me of the correct LED to use.* (User 6)" As an improvement, we will enable users to change the color of an AR stroke or choose a color before drawing a stroke. Meanwhile, another user mentioned that the AR trails should indicate the light painting progress. "*It would be better that those trails can tell me which strokes are finished and which are not.* (User 2)" To this end, we propose to generate a simulated light trace during the *replay* function by tracking the position of the LED light source, so that users can receive a real-time feedback of the light painting progress.

Project the AR trails onto the camera focal plane. While doing light painting with LightPaintAR, users are actually projectively transferring 3D AR trails to the 2D camera image plane. Consequently, the light traces may be distorted if the 3D AR trails are not parallel to the camera focal plane. Such distortion is demonstrated by the picture created by user 3 in session 2 (Figure 4 bottom row). User 3 suggested that "*I should directly draw on the focal plane to avoid the distortion.*" However, user 6 mentioned that this distortion may by leveraged to make the light painting look more stereoscopic. In the future version of LightPaintAR, we will allow users to sketch directly on 2D focal plane or sketch in 3D with a 2D preview of the projected trails.

AR reference for directional light source. During the user study, we adopted a directional LED light bulb whose light was strong only in one direction. During the light painting, we observed that some users were not able to point the LED light bulb to the camera, which resulted in dim light traces, as shown in the photo created by user 2 in session 2 (Figure 4 bottom row). After the user study, both user 2 and 4 mentioned that it was difficult to see the LED light in their hands due to the occlusion caused by the bright AR trails. Given these comments, we propose to track the 6D pose of the LED light and visualize both the correct direction and the actual direction of the light bulb in AR, so that users can adjust the direction of the light source accordingly.

Support more types of light source. The point light source is not the only acceptable light source for light painting photography. Many other tools such as light tubes, fiber optics and plexiglass boards are also frequently used in light painting. Different tools may need moving in different manners (e.g. speed and direction). For future improvements, other light sources will be available during the *paint* function with different representations of the AR trails.

Control the camera from AR headset. In the current setup, LightPaintAR is a stand-alone AR application and is not connected with the digital camera that actually shoots the light painting photo. So an additional person is required to control the camera (a researcher did that during the user study) while a user is doing the light painting. In future upgrades, we will integrate digital camera SDKs into LightPaintAR to control the camera remotely. Additionally, LightPaintAR should compute correct camera parameters (aperture, shutter speed and ISO speed) based on the length of the light trace and environmental light intensity, and adjust the replay speed accordingly.

Generate AR trails directly from pictures. Currently, LightPaintAR only supports manually sketched AR trails. Thus, users who are not good at sketching cannot create high-quality AR reference for light painting. In the future, we will add a function that can automatically generate AR trails from images to relieve those users from hand-sketching.

8 CONCLUSION

In this paper, we proposed LightPaintAR, an AR application that supports photographers and hobbyists to create light painting photos. LightPaintAR allows users to design and edit light painting patterns in AR, and use the virtual trails as a spatial reference while moving the light source. We distilled fundamental design goals from the formative interviews with professional photographers. To fulfill the requirements, we developed an AR interface with key functionalities which create AR reference that assists light painting. Through a user evaluation, we proved that LightPaintAR improves light painting accuracy and the overall experience. Meanwhile, the user feedback also revealed interesting research directions to further enhance and expand the system capabilities. Therefore, we believe that LightPaintAR introduces a new perspective of improving light painting with the help of AR.

9 ACKNOWLEDGEMENT

This work was partially supported by the National Science Foundation under grants Future of Work at the Human Technology Frontier (FW-HTF) 1839971. We also acknowledge the Feddersen Chair Funds. We thank Zhengzhe Zhu for providing photography equipment. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding agency.

REFERENCES

[1] Judith Amores and Jaron Lanier. 2017. Holoart: Painting with holograms in mixed reality. In *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems*. 421–424.

[2] Rahul Arora, Rubaiat Habib Kazi, Tovi Grossman, George Fitzmaurice, and Karan Singh. 2018. Symbiosissketch: Combining 2d & 3d sketching for designing detailed 3d objects in situ. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–15.

[3] Yuanzhi Cao, Tianyi Wang, Xun Qian, Pawan S Rao, Manav Wadhawan, Ke Huo, and Karthik Ramani. 2019. GhostAR: A Time-space Editor for Embodied Authoring of Human-Robot Collaborative Task with Augmented Reality. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*. 521–534.

[4] Danilo Gasques, Janet G Johnson, Tommy Sharkey, and Nadir Weibel. 2019. What you sketch is what you get: Quick and easy augmented reality prototyping with pintar. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–6.

[5] Google. [n.d.]. Tilt Brush. https://www.tiltbrush.com/

[6] Ping-Hsuan Han, Kuan-Wen Chen, Chen-Hsin Hsieh, Yu-Jie Huang, and Yi-Ping Hung. 2016. Ar-arm: Augmented visualization for guiding arm movement in the first-person perspective. In *Proceedings of the 7th Augmented Human International Conference 2016*. 1–4.

[7] Yaozhun Huang, Sze-Chun Tsang, and Miu-Ling Lam. 2016. Computational swept volume light painting via robotic non-linear motion. In *ACM SIGGRAPH 2016 Posters*. 1–2.

[8] Yaozhun Huang, Sze-Chun Tsang, Hei-Ting Tamar Wong, and Miu-Ling Lam. 2018. Computational light painting and kinetic photography. In *Proceedings of the Joint Symposium on Computational Aesthetics and Sketch-Based Interfaces and Modeling and Non-Photorealistic Animation and Rendering*. 1–9.

[9] Ke Huo and Karthik Ramani. 2017. Window-shaping: 3d design ideation by creating on, borrowing from, and looking at the physical world. In *Proceedings of the Eleventh International Conference on Tangible, Embedded, and Embodied Interaction*. 37–45.

[10] Roman Ibrahimov, Nikolay Zherdev, and Dzmitry Tsetserukou. 2020. DroneLight: Drone Draws in the Air using Long Exposure Light Painting and ML. In *2020 29th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*. IEEE, 446–450.

[11] Shunichi Kasahara, Valentin Heun, Austin S Lee, and Hiroshi Ishii. 2012. Second surface: multi-user spatial collaboration system based on augmented reality. In *SIGGRAPH Asia 2012 Emerging Technologies*. 1–4.

[12] Kin Chung Kwan and Hongbo Fu. 2019. Mobi3dsketch: 3D sketching in mobile AR. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–11.

[13] Steve Mann, Ryan Janzen, Tao Ai, Seyed Nima Yasrebi, Jad Kawwa, and Mir Adnan Ali. 2014. Toposculpting: Computational lightpainting and wearable computational photography for abakographic user interfaces. In *2014 IEEE 27th Canadian Conference on Electrical and Computer Engineering (CCECE)*. IEEE, 1–10.

[14] Microsoft. [n.d.]. Microsoft Hololens. https://www.microsoft.com/en-us/hololens

[15] Kejia Ren and Paul G Kry. 2019. Single stroke aerial robot light painting. In *Proceedings of the 8th ACM/Eurographics Expressive Symposium on Computational Aesthetics and Sketch Based Interfaces and Modeling and Non-Photorealistic Animation and Rendering*. Eurographics Association, 61–67.

[16] Nestor Z Salamon, Marcel Lancelle, and Elmar Eisemann. 2017. Computational light painting using a virtual exposure. In *Computer Graphics Forum*, Vol. 36. Wiley Online Library, 1–8.

[17] Rajinder Sodhi, Hrvoje Benko, and Andrew Wilson. 2012. LightGuide: projected visualizations for hand movement guidance. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 179–188.

[18] Ryo Suzuki, Rubaiat Habib Kazi, Li-Yi Wei, Stephen DiVerdi, Wilmot Li, and Daniel Leithinger. 2020. RealitySketch: Embedding Responsive Graphics and Visualizations in AR through Dynamic Sketching. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. 166–181.

[19] Marina Wada, Akito Nakano, Yoshifumi Mizuno, and Hisakazu Hada. 2016. The Light Painting by a Fluorescence String Figure. In *Proceedings of the 13th International Conference on Advances in Computer Entertainment Technology*. 1–5.

[20] Tianyi Wang, Xun Qian, Fengming He, Xiyun Hu, Ke Huo, Yuanzhi Cao, and Karthik Ramani. 2020. CAPturAR: An Augmented Reality Tool for Authoring Human-Involved Context-Aware Applications. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. 328–341.