EUROGRAPHICS 2017 / L. Barthe and B. Benes
(Guest Editors)

Volume 36 (2017), Number 2

Computational Light Painting Using a Virtual Exposure

Nestor Z. Salamon[1], Marcel Lancelle[2] and Elmar Eisemann[1]

[1]Delft University of Technology, The Netherlands
[2]ETH Zurich, Switzerland

Figure 1: *Light Paintings created with our approach. The production only took minutes, design and modifications are performed in real time.*

Abstract

Light painting is an artform where a light source is moved during a long-exposure shot, creating trails resembling a stroke on a canvas. It is very difficult to perform because the light source needs to be moved at the intended speed and along a precise trajectory. Additionally, images can be corrupted by the person moving the light. We propose computational light painting, which avoids such artifacts and is easy to use. Taking a video of the moving light as input, a virtual exposure allows us to draw the intended light positions in a post-process. We support animation, as well as 3D light sculpting, with high-quality results.

Categories and Subject Descriptors (according to ACM CCS): I.3.4 [Computer Graphics]: Graphics Utilities—Paint systems

1. Introduction

In recent years, light painting has gained significant popularity in photography. There are two variants, both sharing a slow shutter speed to capture light trajectories on the image plane. One option is to move the camera, while keeping light sources fixed - known as Kinetic Light Painting, e.g., Lorenzi and Francaviglia [LF07] installed a large set of light sources during the Generative Art 2007 conference. In this paper, we focus on the second option, where the camera is static and the lights are dynamic. While some large-scale attempts have been made with hundreds of people [PS15], most creations are performed by a single person, who needs to skillfully move the light source with the goal of producing a target light trail on the camera sensor. Such a process is tedious, requires high precision, and usually involves many attempts because the user cannot see the past stroke and there is no feedback while drawing. Moreover, artists usually wear black cloth and move fast to not corrupt the result by appearing in the background. We propose computational light painting, where the light painting is made in a post-process by drawing on the screen. Consequently, the precision is high and the output is easy to control, which makes precise animation possible.

The input to our approach is a video of a person moving a light source along arbitrary trajectories through the scene, which is usually captured within a minute. Our method processes the input to remove artifacts caused by the artist and removes the light source. The latter allows us to replace it by a smaller synthetic light, which can then be drawn at any position. Removing the person from the background avoids ghosting effects, which are typical for real light paintings. We then produce a data structure to query an approximation of the relit environment for any light position. It is, hereby, possible to paint light trajectories and accumulate the illumination contributions. We show various results, including simulated professional light brushes [Car11].

Specifically, our work makes the following contributions:

- A simple video acquisition methodology;
- A preprocess to remove acquisition artifacts;
- A robust light interpolation method;
- A design interface for light painting.

2. Background and Related Work

Light painting dates back to 1880: the *Pathological Walk From in Front* by Étienne-Jules Marey and Georges Demeny. The authors, both psychologists, used the technique to understand human motion. Only in the early 1900s, the first artistic light drawings were produced by Man Ray and later Picasso. The theoretical background was presented in 1940 by Leslie Walker [Wal40]. Afterwards, the artform gained momentum, as summarized by Lance Keimig [Kei15]. Today, we see a renewed interest in the topic. The availability of digital cameras makes it cheap to rely on trial-and-error attempts, which motivated many hobby artists. Nonetheless, the process remains time consuming and requires a high level of skill for convincing results.

The most recent development has been presented by the Austrian company FilmSpektakel. They propose a light painting pipeline named Holopainting [Fil16]. They first acquire a subject with a ring of 24 connected cameras, which serves as a 3D scanner to recover a moving subject. The imagery is then transferred to an LED stick, which illuminates the 3D scene automatically during the long-exposure shot. From recording to final result, the process takes days, including manually cropping each captured image to fit the subject in the LED stick.

Faster, but less accurate applications (e.g., [MB15, MA13]) simplify the process by accumulating many short exposed images to a single long-exposure shot. The principle is similar to Telleen et al. [TSY*07], who introduced the virtual exposure, which also forms the basis of our approach. Given this collection of frames, the user can delete or replace time periods. Nonetheless, artifacts due to the user's presence in the scene remain and the painting still needs to be performed actively. The latter also holds for recent approaches using virtual reality [Rob16].

Related to our approach are also relighting scenarios. In virtual environments, it has been a long established concept [NSD94], but similar principles of basis images for lighting can also be found for real-world applications. An image-based approach to light design was presented in [ADW04]. Manders and Mann [MM06] proposed a solution to add light on the fly by employing an adapted flash equipment that triggers the camera shutter to accumulate the lit results. Similarly, Boyadzhiev et al. [BPB13] capture a set of flash photographs with distinct light positions, then used in a composition method guided by the user. Product relighting is addressed analogously in [BCPB16]. The authors record a video, illuminating a product from different angles and find image snippets, which match design principles. Later, the user can choose and compose them.

Low-frequency environmental lighting can be achieved via Bayesian relighting [FBS05], which involves a light probe to estimate the environmental illumination. Knowing the illumination can be useful for scene reconstruction (e.g., [Woo89]), and material estimates (e.g., [MLP04]), as well as composition [Deb98]. An advanced light installation, supporting precise capture and relighting, is the light stage [Deb12]. An overview of image-based lighting can be found in [Deb05].

Our work is inspired by the effectiveness of low-cost image-based relighting solutions. It requires no calibration and provides access to a difficult-to-master artform. This latter aspect, we share in spirit with work around spray-can images [PJJSH16]. Nonetheless, a major difference is that our solution is compatible with the entire image-processing toolbox, such as Photoshop, or even advanced supporting solutions, such as ShadowDraw [LZC11].

In the following, Sec. 3 describes our approach, including acquisition and processing (Sec. 3.1), light interpolation (Sec. 3.2), and the user interface (Sec. 3.3). We will then present our results (Sec. 4) before concluding (Sec. 5).

3. Our Approach

An overview of our approach is shown in Fig. 2. The input is a video recorded with a moving light source. First, we process the video to remove artifacts related to the capturing process; we want to obtain *lighting images*, which encode the illumination of the scene. Neither the light source nor the person carrying the light source should be included (Sec. 3.1). Next, these frames are organized to provide a basis for new light positions, which are approximated by interpolation (Sec. 3.2). Finally, we will explain our interface and how the actual light painting is performed (Sec. 3.3).

3.1. Acquisition and Processing

The acquisition process is simple and allows for a robust derivation of the scene illumination. We use a white light source, which allows us to tint the illumination in different colors in a post-process. The user is asked to move through the scene from left to right and then back, waving the light source in front. In our setup, we relied on a standard light bulb attached to a stick or a flashlight in a thin plastic cup acting as a diffuser. To avoid automatic camera adjustments, the recording camera is in manual mode.

Our goal is to transform each frame of the input sequence into an image that only encodes the light added by the moving source and free of artifacts from the person moving it. The first step linearizes the response curve of the camera [Art16]. We convert our images by fitting a gamma correction followed by a normalization step. The so-linearized images are then processed further.

Ambient Lighting

In order to obtain only the added light, we subtract the ambient lighting A from each frame. To determine A, we rely on the first few frames of the video - before the moving light source appears in the scene. Accumulating these frames gives a good estimate of A for an exposure time equivalent to the summed frames [TSY*07]. In consequence, we need to divide by their number to obtain the background illumination per frame. Subtracting A from all frames then gives us a new video containing only the added light.

Removing the Light Source

We are very forgiving when estimating light-source sizes, shape, and positions. Hence, even if the captured source is slightly larger or some positions have not been properly sampled, we can produce a convincing result by interpolating between the images. Nonetheless, leaving the light source (or the person manipulating it) visible

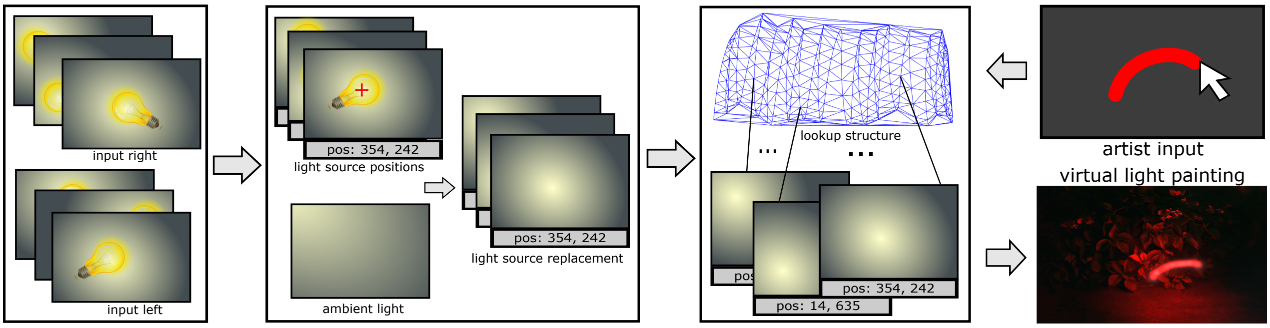

Figure 2: *Overview: The input frames are from a static scene with a moving light source. For each frame, the light position is obtained and its illumination of the scene extracted. These contributions are stored in a lookup structure, which can be queried to design light paintings in real time.*

in the image produces unrealistic ghosting. We first focus on light-source removal before addressing the person ghosting in Sec. 3.2.

Light-source tracking. In all of our experiments the directly visible light source fully saturates the sensor. This greatly simplifies the detection. We employ a high threshold (1.0 on all channels) and then grow these regions up to a threshold of 0.9 before applying a morphological closure to remove noise. Still, this process is insufficient in the presence of reflections. To robustly estimate the light position, we leverage the temporal coherency of the video with a consistently moving light source. In consequence, we perform a tracking over time to derive a mask that we can use for the light-source removal.

We choose a standard tracking solution for the light position. The tracking is initialized on each detected region in the thresholded image. Next, we match each region to the following frame and verify i) a low size variation $\Delta s < 20\%$ and ii) a small-distance movement $\Delta d < 5\% s$, where s is the light's estimated size; the numbers have been determined experimentally. As the user is asked to move from left to right during acquisition, we can trigger the tracking when the light enters the scene from the left. Further, the movement being parallel to the image plane, the light will keep a constant size. To match two succeeding frames, we rely on a sum of squared differences and determine the best translation vector.

Figure 3: *Our tracking algorithm automatically finds the center of the light sources with different sizes, occlusions and reflections.*

Light-source replacement. Having a mask that covers the light source, we now inpaint the corresponding region to remove the light source from each frame. While a standard inpainting [FFLS08] could be used, it is only a coarse estimate because the light source bleached all structural information. Instead, we inpaint the region based on the input-video frames.

To recover the image structure behind the light source, one could compute a mean over all images, but this process causes strong ghosting artifacts due to the high-contrast light source. A second option is a median, but it is generally too dark due to the short illumination of each part. We found that a trimmed mean (10% of the darkest and brightest samples removed) provides a robust trade-off between the artifacts in the mean and the darkness of median (Fig. 4).

Figure 4: *Methods to compute a neutral image for light source replacement. From left to right: Mean, median and trimmed mean.*

Copying the patch directly from the trimmed-mean image T would result in visible seams. A solution to this problem is a Poisson reconstruction with the pixels around the masked region serving as constraints and T as the guide image [PGB03]. Hereby, elements that are lit, extend their illumination naturally into the uncovered region. In practice, we use Tanaka et al.'s variant [TKO12], which leads to less color bleeding. Still, the solution exhibits smaller artifacts because the gradients in T might not have a consistent magnitude. To adapt T, we need to scale it with a correcting factor. As the patch lies directly underneath the light position, it would appear fully illuminated, if the view was not blocked by the light source. For this reason, we are interested in the maximal possible magnitude of gradients for this area. We estimate it by using the average magnitude of the top 1% gradients in this region over all images, excluding the ones covered by the source. We then scale T to have its gradients match the result. A comparison of the inpainting processes is shown in Fig. 5. We refer to the resulting images as the *lighting images*.

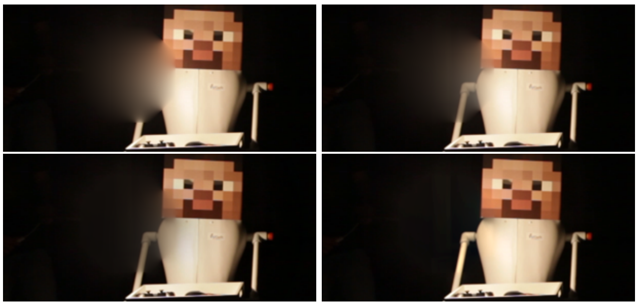

Figure 5: *Light source replacement. Top: Poisson without and with guide [PGB03]. Bottom: Tanaka et al. [TKO12] and our result.*

3.2. Light Interpolation

The preprocessing results in a set of lighting images $\{L_i\}$ corresponding to the added light of the moving source at position p_i. Here, we describe how to approximate the lighting image L_n of a new virtual point light at position p_n with an emission E. E is assumed to be given relative to the light source used in the capturing process. If the virtual light is at a captured position, there is a p_k equal to p_n, and we obtain $L_n := E * L_k$. If not, we will define $L_n := E * \sum_i w_i L_i$, for a suitable set of weights $\{w_i\}$.

One weight definition is the inverse distance metric (Shepard interpolation); $w_i = \hat{w}_i / \sum_i \hat{w}_i$, where $\hat{w}_i = \frac{1}{|p_n - p_i| + \varepsilon}$. The parameter ε has a strong impact on the outcome and can lead to a diffuse solution (Fig. 6). A more suitable and local interpolation uses a Delaunay Triangulation of $\{p_i\}$, where the weights are defined via barycentric coordinates (Fig. 7). Hereby, only the surrounding three vertices are used for interpolation. Consequently, locality is increased, while being continuous over the entire domain.

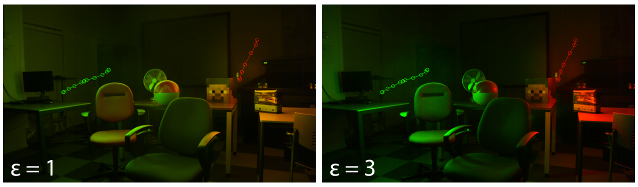

Figure 6: *Shepard interpolation of captured sources varying* ε.

Figure 7: *A Delaunay triangulation of light centers defines the weights to interpolate the captured light images.*

Person-aware weights. While the interpolation allows us to define new light positions, we still need to handle the artifacts caused by the person moving the light source. To this extent, we introduce **person-aware weights**.

As the capturing process is executed twice (from left to right and vice versa), it is easy to determine the side (left/right) of the image contaminated by the person. Please notice that also all shadows cast by this person lie on this side. Consequently, we can remove the part of the image that contains the contaminations, resulting in the sets $\{\hat{L}_i^l\}$ and $\{\hat{L}_i^r\}$. L_i^l is equal to the original input frame, except that all pixels left of p_i^l are zero. Unfortunately, we cannot assume that the light source will visit the exact same locations during both passes, i.e., for a position p_k^r there might not be a corresponding position p_i^l. Thus, we cannot directly combine the image parts to produce one artifact-free set.

To fuse the images, we rely on the weights. First, we perform the Delaunay triangulation for the right $\{p_i^r\}$ and left pass $\{p_i^r\}$ independently, and derive corresponding weights $\{w_i^r\}$ and $\{w_i^l\}$. Next, we compute two result images using only left ($L_n^l := \sum w_i^l L_i^l$) and right image parts ($L_n^r := \sum w_i^r L_i^r$). Additionally, we produce corresponding weight images $W^r = \sum w_i^r \chi_i^r$ and $W^l = \sum w_i^l \chi_i^l$, where χ_i^x is the characteristic function that is one if the pixel lies in the valid part (left/right) of the image. For example, $w_i^l \chi_i^l$ is an image where all pixels left to the light position p_i^l contain w_i^l, while the rest is zero. We define $L_n := (L_n^r + L_n^l)/(W^l + W^r)$, which is a blended image without gaps. To avoid seams, we define χ_i^l (χ_i^r) not as a step but as a smooth ramp function to blend the image contributions. Fig. 8 shows the result before and after person-aware weights.

Figure 8: *Ghosts by the light-handling person (left) are avoided by our approach (right).*

3.3. Light Painting Interface

We have seen how to query a lighting image L_n for a virtual point light. Based on this result, we can reconstruct an approximation of the corresponding illuminated scene by $L_n + A$, where A is the ambient light image. To extend the principle to general virtual sources, we exploit the linearity of light transport; we approximate the stroke via a set of point lights and sum the corresponding light images. Consequently, we can use arbitrary pixel input.

Our interface enables the user to define strokes of varying thickness, color, and intensity. Intensity is defined with respect of the exposure time of a frame. If one wants to simulate a different exposure time, all intensities and the ambient image should be scaled before addition. Strokes can either be drawn as sampled freeform curves (represented by dense polylines) or splines. We support a special drawing mode, which keeps track of the speed at which the mouse moves - slow meaning brighter, which mimics the effect during a real exposure. The properties and control points of the curve can still be modified after its creation and values are interpolated between the control points (exemplified by a color gradient in Fig. 9, right). Additionally, vector or bitmap images are supported (Fig. 10).

Figure 9: *Intensity (left) and color (right) variation along a stroke.*

Our interface also allows the user to define key-framed animation and path definitions. The light is then interpolated according to the user-provided animation. There are two options for the output: assuming a short-exposure in each frame, or an accumulation (resulting in an increasingly long exposure) along the trajectory.

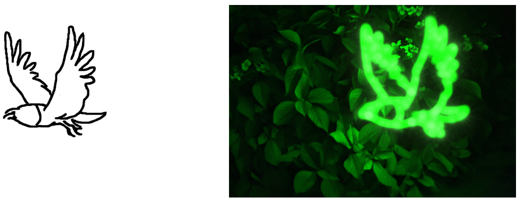

Figure 10: *Vector input (based on Whitaker and Halas [WH13]) and the resulting light painting.*

Extensions

There are several extensions to our basic approach, explained so far, which are also exposed in the interface of our prototype implementation. They can improve the quality of the results drastically.

Lens Flare and Starburst Pattern. To increase realism and an increased sense of light [RIF*09], we can add common camera artifacts [HESL11, JKL*16, LE13]. To this extent, we convolve a starburst pattern [HESL11] induced by the aperture with the stroke input.

Depth Capture. Our approach can be extended to 3D light sculpting. The capturing process acquires one depth layer at a time. These layers are combined in form of a 3D Delaunay triangulation to enable 3D interpolation. To determine the 3D position of each light source, we rely on its relative size. This depth/size parameter is then exposed in the interface. A user can choose a value for each key point of a stroke and also animate these values.

Specularities. For very specular objects, we require a detailed acquisition to avoid undersampling artifacts. Or existing algorithms for artifact-free specularity interpolation [FLBS07] could be applied. As an alternative, we propose to remove specularities by thresholding the images again, while excluding the light source, followed by our inpainting algorithm.

GPU Acceleration. We mapped the light painting accumulation efficiently on the GPU. We first precompute two images, a weight image, where each pixel contains the barycentric weights of the triangle that contains it, and an index image, which contains the indices of the triangle's vertices. For each input pixel on the stroke, we can thus retrieve the weights and index of the light images that need to be accumulated. The composition can then be achieved via alpha blending.

4. Results

Aside from the GPU painting interface, we implemented our method in Matlab. The GPU painting interface runs at 60 fps for 960x540 images on a GTX 980 but the preprocessing takes around 5/20 seconds per frame (960x540/full HD). The latter could be improved but our focus was on showing the high potential of computational light painting. We provide many examples, ranging from indoor scenes, to outdoor examples, on a large and small scale. The scenes were captured by different artists with different equipments. The videos were recorded at frame rates from 24 to 30 fps and with manual focus; ISO and aperture parameters were set according to environmental characteristics (strong/weak environment light, distance camera-light source, etc.). All videos were processed as obtained and recorded in a single attempt.

While light painting animations can take months [Wu16], our approach can provide results in the order of seconds, using key frame animation coupled with a path trajectory (Fig.11 and supplementary video).

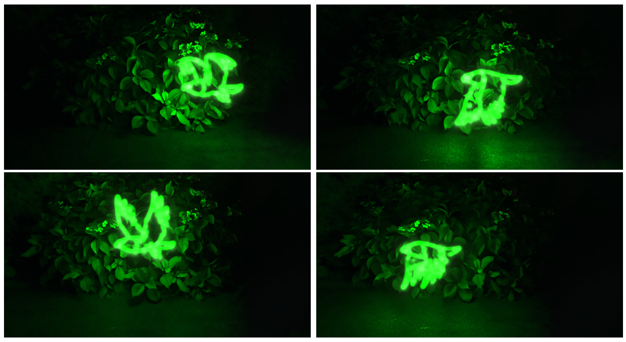

Figure 11: *Frames of an animation (see supplemental video).*

Predefined drawings or single strokes (Fig. 12), potentially augmented with flares for more realism (Fig. 13), are basically done instantly (no example took more than a minute of design).

Figure 12: *Using drawings and fonts as input points.*

Figure 13: *Different strokes with and without lens flare.*

Complex drawings with layers, several shapes, or strokes took us up to five minutes to design. A layered result is shown in Fig. 14. The same has been applied for the Eurographics logo in the theatre (Fig. 19, top left).

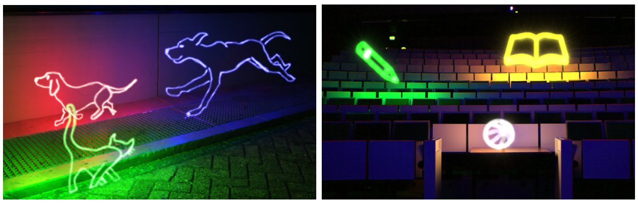

Figure 14: *3D drawing in different layers.*

For cases where the scene was not consistently sampled or recorded from only one side, our approach still produces convincing results, but cannot remove all artifacts. Fig. 15 (left) shows our result for a person standing behind the light source and, Fig. 15 (right), uses only few light samples. Moreover, if the light source is weak, we cannot scale the brightness without introducing noise. These artifacts occur also in traditional light paintings. Similarly, the behavior for non-static scenes is identical and results in motion blur. As an example, Fig. 19 (top, right) was recorded with a strong wind moving the leaves and Fig. 19 (bottom, right) shows a moving cow in the background.

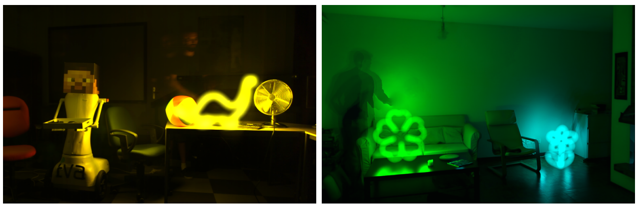

Figure 15: *Results with inputs recorded from one side, with the person stands behind the light source or in a sparsely sampled scene.*

Fig. 16 illustrates a challenging example and shows the effectiveness of our solution in handling specular surfaces. The coarse sampling could lead to artifacts, but our work removes these problems via inpainting of the specularities. Large mirrors can still lead to problems and ghosting effects cannot be completely avoided. In the future, using polarized filters could help detect such cases.

Besides light painting, our method can also be used for product light design or relighting. In Fig. 17, an artificial lamp is placed in an apartment scene and the illumination steered with our solution. The wide variety of scenes that can be handled is also illustrated in the overview (Fig. 19).

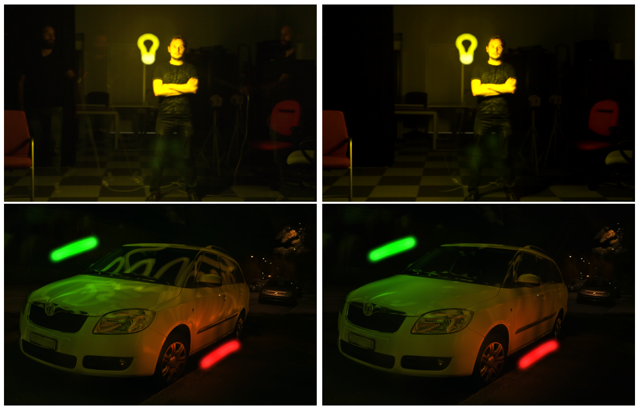

Figure 16: *Ghost removal (top) and specularity treatment (bottom).*

Figure 17: *Virtual placement of a wall lamp with different positions and color temperatures.*

To examine precision, we tested a synthetic scene. Fig. 18 (top, left) shows the reference long exposure for a light source moving horizontally in the middle of the scene. Fig. 18 (bottom) shows the results obtained using our approach interpolated via Delaunay Triangulation.

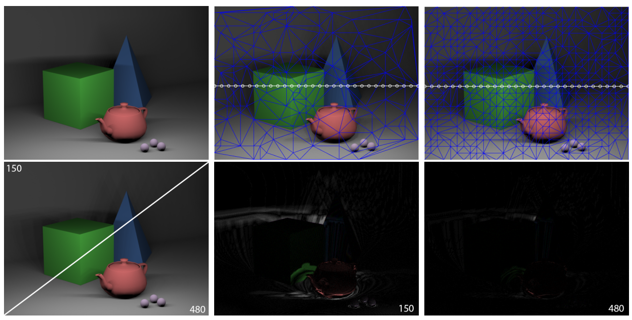

Figure 18: *Top: synthetic ground truth (left), a sparse (150 points, middle) and a dense (480 points, right) light sampling. Bottom: our results and its respective absolute differences to the ground truth (differences increased* 800% *for better visualization).*

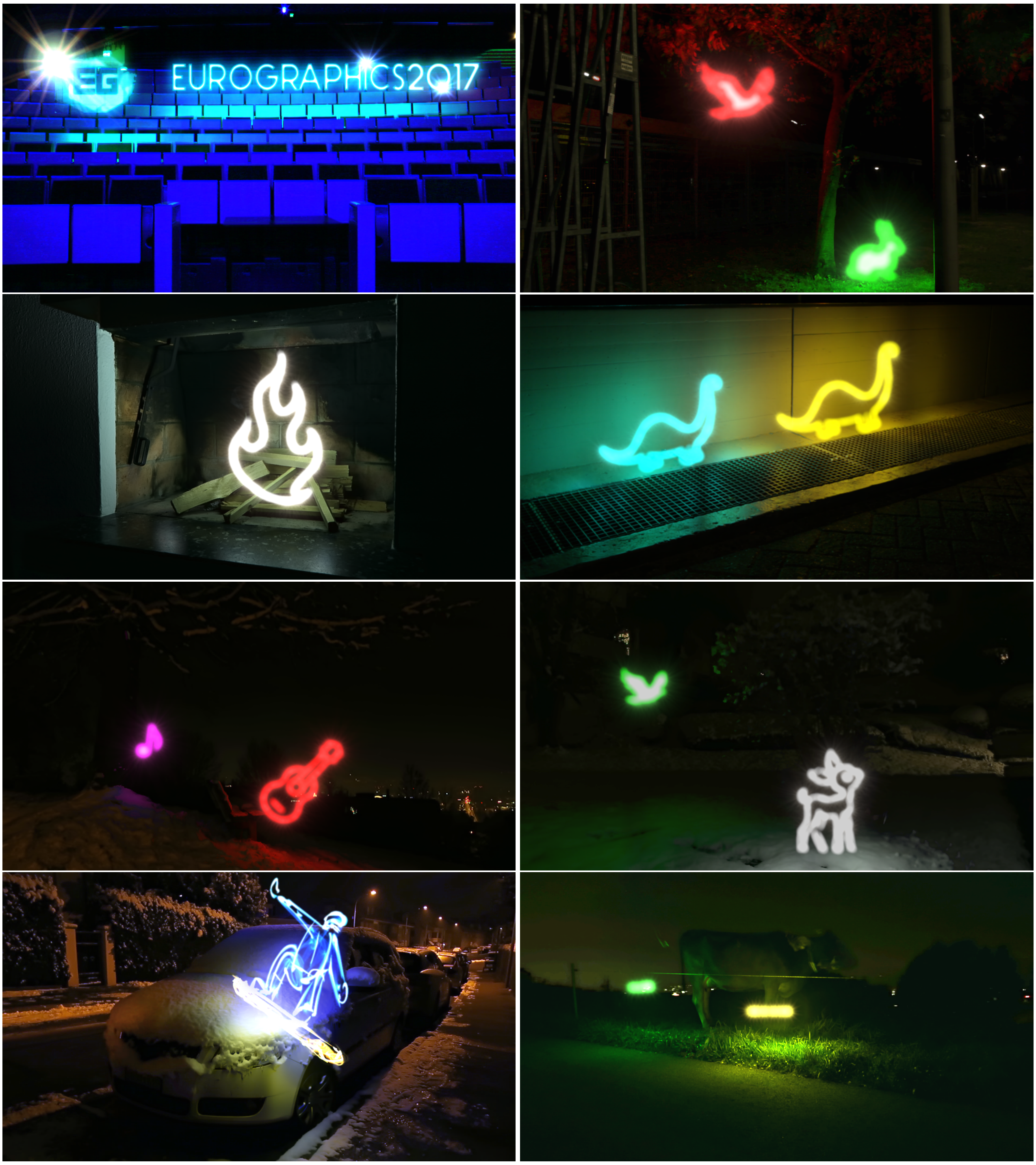

Figure 19: *Several challenging examples illustrating our solution.*

5. Conclusion and Future Work

We have presented an easy-to-use pipeline for light painting. It usually requires skill and is tedious but our solution simplifies the creation process. Our method addresses common artifacts with a novel inpainting solution, and avoids ghosts resulting from the artist being present in the scene. It requires a minimal and low-cost hardware setup, while delivering high-quality results and increased artistic freedom by moving creative choices to post production.

In the future, we would like to investigate new ways of scene capture, e.g., via a drone. Alternatively, an accelerometer can lead to good position estimates, even when the source is occluded. Further, polarization filters could help in removing reflections in mirror surfaces, for which we could then recompute the light reflection during reconstruction.

Acknowledgments

This work was partially funded by the Brazilian agency CNPq and the Intel VCI at Saarland University.

References

[ADW04] ANRYS F., DUTRÉ P., WILLEMS Y. D.: Image-based lighting design. In *Proceedings of the 4th IASTED International Conference on Visualization, Imaging, and Image Processing* (2004), p. 15. 2

[Art16] ARTMANN U.: Linearization and normalization in spatial frequency response measurement. *Electronic Imaging 2016*, 13 (2016). 2

[BCPB16] BOYADZHIEV I., CHEN J., PARIS S., BALA K.: Do-it-yourself lighting design for product videography. In *International Conference on Computational Photography (ICCP)* (2016). 2

[BPB13] BOYADZHIEV I., PARIS S., BALA K.: User-assisted image compositing for photographic lighting. *ACM Trans. Graph. 32*, 4 (2013), 36–1. 2

[Car11] CARLSEN M.: Light painting with the world's most powerful flashlights. https://www.youtube.com/watch?v=U_iOqM8b55g, 2011. Accessed: 2016-09-12. 1

[Deb98] DEBEVEC P.: Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 1998), SIGGRAPH '98, ACM, pp. 189–198. 2

[Deb05] DEBEVEC P.: Image-based lighting. In *ACM SIGGRAPH 2005 Courses* (New York, NY, USA, 2005), SIGGRAPH '05, ACM. 2

[Deb12] DEBEVEC P.: The Light Stages and Their Applications to Photoreal Digital Actors. In *SIGGRAPH Asia* (Singapore, Nov. 2012). 2

[FBS05] FUCHS M., BLANZ V., SEIDEL H.-P.: Bayesian relighting. In *Proceedings of the Sixteenth Eurographics Conference on Rendering Techniques* (2005), Eurographics Association, pp. 157–164. 2

[FFLS08] FARBMAN Z., FATTAL R., LISCHINSKI D., SZELISKI R.: Edge-preserving decompositions for multi-scale tone and detail manipulation. SIGGRAPH '08, ACM, pp. 67:1–67:10. 3

[Fil16] FILMSPEKTAKEL: Holopainting combines light painting, stop motion, and hyperlapse. http://bit.ly/2dOpq0t, 2016. Accessed: 2016-09-12. 2

[FLBS07] FUCHS M., LENSCH H., BLANZ V., SEIDEL H.-P.: Superresolution reflectance fields: Synthesizing images for intermediate light directions. In *Computer Graphics Forum* (2007), no. 3, pp. 447–456. 5

[HESL11] HULLIN M., EISEMANN E., SEIDEL H.-P., LEE S.: Physically-based real-time lens flare rendering. *ACM Transactions on Graphics 30*, 4 (2011), 1. 5

[JKL*16] JOO H., KWON S., LEE S., EISEMANN E., LEE S.: Efficient Ray Tracing Through Aspheric Lenses and Imperfect Bokeh Synthesis. *Computer Graphics Forum (Proc. EGSR'16) 35*, 4 (2016), 99–105. 5

[Kei15] KEIMIG L.: *Night Photography and Light Painting: Finding Your Way in the Dark*. CRC Press, 2015. 2

[LE13] LEE S., EISEMANN E.: Practical real-time lens-flare rendering. *Computer Graphics Forum (Proc. of EGSR) 32*, 4 (2013), 1–6. 5

[LF07] LORENZI M. G., FRANCAVIGLIA M.: Painting with light: generative artworks or setting in motion the fourth dimension. In *Proceedings of the 10th Generative Art Conference* (2007), pp. 12–14. 1

[LZC11] LEE Y. J., ZITNICK C. L., COHEN M. F.: Shadowdraw: Real-time user guidance for freehand drawing. *ACM Trans. Graph. 30*, 4 (July 2011), 27:1–27:10. 2

[MA13] M-APPS: Light painting/strokes camera. http://bit.ly/2dU9BX5, 2013. Accessed: 2016-09-05. 2

[MB15] MIEDZA J., BECKWITH J.: Lightpaint live. http://lightpaintlive.com/, 2015. Accessed: 2016-09-05. 2

[MLP04] MATUSIK W., LOPER M., PFISTER H.: Progressively-Refined Reflectance Functions from Natural Illumination. In *Eurographics Workshop on Rendering* (2004). 2

[MM06] MANDERS C., MANN S.: Handheld electronic camera flash lamp as a tangible user-interface for creating expressive visual art works. *Proceedings of the 14th annual ACM international conference on Multimedia - MULTIMEDIA '06* (2006), 509. 2

[NSD94] NIMEROFF J. S., SIMONCELLI E., DORSEY J.: Efficient re-rendering of naturally illuminated environments. In *IN FIFTH EUROGRAPHICS WORKSHOP ON RENDERING* (1994), p. 359–373. 2

[PGB03] PÉREZ P., GANGNET M., BLAKE A.: Poisson image editing. In *ACM Transactions on Graphics (TOG)* (2003), vol. 22, ACM, pp. 313–318. 3, 4

[PJJSH16] PRÉVOST R., JACOBSON A., JAROSZ W., SORKINE-HORNUNG O.: Large-scale painting of photographs by interactive optimization. *Computers and Graphics 55* (2016), 108–117. 2

[PS15] PERES M., SAVAKIS A.: Crowdpainting with Light: Participatory Imaging at the Big Shot. In *Proceedings of the IEEE International Conference on Computer Vision Workshops* (2015), pp. 32–36. 1

[RIF*09] RITSCHEL T., IHRKE M., FRISVAD J. R., COPPENS J., MYSZKOWSKI K., SEIDEL H.-P.: Temporal Glare: Real-Time Dynamic Simulation of the Scattering in the Human Eye. *Computer Graphics Forum* (2009). 5

[Rob16] ROBERTSON A.: Painter of light: my awkward adventures in virtual reality art. http://bit.ly/1KV9rM1, 2016. Accessed: 2016-09-26. 2

[TKO12] TANAKA M., KAMIO R., OKUTOMI M.: Seamless image cloning by a closed form solution of a modified poisson problem. In *SIGGRAPH Asia 2012 Posters* (2012), ACM, p. 15. 3, 4

[TSY*07] TELLEEN J., SULLIVAN A., YEE J., WANG O., GUNAWARDANE P., COLLINS I., DAVIS J.: Synthetic shutter speed imaging. *Computer Graphics Forum 26*, 3 (2007), 591–598. 2

[Wal40] WALKER L. C.: *The technique of painting with light*. Nash-Jones, 1940. 2

[WH13] WHITAKER H., HALAS J.: *Timing for animation*. Focal Press, 2013. 5

[Woo89] WOODHAM R. J.: Shape from shading. MIT Press, Cambridge, MA, USA, 1989, ch. Photometric Method for Determining Surface Orientation from Multiple Images, pp. 513–531. 2

[Wu16] WU J.: Rhythm of light. https://vimeo.com/184667838, 2016. Accessed: 2016-10-06. 5