Portrait Light Painting: Holograms Through Time

Paul Effing[1,*] Eckart Cobo-Briesewitz[1,*] Jan-Hendrik Schneider and Marc Toussaint[1]

***Abstract*— With the new advancements in AI and robotics, it is becoming more realistic to envision a future where humans and machines collaborate to solve everyday tasks. Viewing a robot as an entity reacting to its environment opens possibilities for artistic expression beyond the classic observer-object relationship. In this work, we demonstrate an interactive fusion of Robotics, Image Processing, and Photography. We propose a portrait light painting installation capable of detecting observers' human faces and transforming them into light paintings via long-exposure photography. Leveraging a classical path-finding algorithm, the resulting portraits are non-deterministic, giving the installation an ever-changing characteristic. The work consists of two main contributions: the extraction and generation of human portraits from images, and the actual light painting procedure utilizing the Franka Emika robotic arm platform.**

I. Introduction

Robots have been widely used in a variety of artistic and educational applications [13], [9], [12], [2]. One interesting avenue for artistic works utilizing robots is to use them for making portraits of human faces [3]. Most work utilizing robots for making portraits limit themselves to either printing the original picture of the face after some edge filtering, perform unsatisfying style transfer or limit themselves to the 2D domain.

Light paintings are a photographic technique where a light source is moved while taking a long-exposure photograph, creating "drawings" with light [18]. As shown in Figure 4, this technique enables the creation of drawings not only in 2D but also in 3D, as the artist is not confined to a flat canvas but can use the entire space within the camera's view. However, precise drawings are challenging for humans to perform because they cannot see what is being drawn and have no way to fix errors without restarting. Robotics offers a reliable solution for executing these drawings with high precision.

One engaging type of artwork allows interaction between the viewer and the piece itself. In our proposal, we enable such interaction by having a Franka Emika robot [4] draw portraits of people it observes.

Various methods can represent pixel-based images from a camera as sequences of drawn lines. For our work, we chose an approach closely related to robotics, using the Rapidly-exploring Random Trees (RRT) [8] path-finding algorithm to generate lines to be drawn. The algorithm exhibits a strong resemblence to natural phenomena as it models organisms spreading through space such as in the branching structure of arteries in the human body or the path of lightning. Visualizing the RRTs offers an educational perspective, making abstract computational processes more tangible. Moreover the emergence of the human portrait from the combination of several RRTs shows the endless possibilities of utilizing even the simplest and best-known algorithms.

The results of our approach show intricate paths where each portrait becomes a unique piece of art. The final photographs have a great educational value by teaching the inner workings of one very important algorithm in robotics and path finding. A supplementary video can be found here.

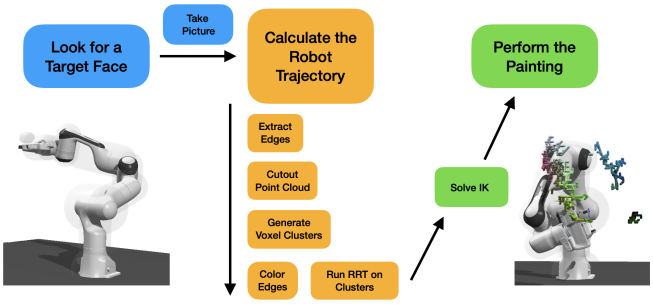


Fig. 1: **Overview of the Proposed Setup:** In the first stage, the robot searches for a human face. Once a face is detected, it processes the captured image and generates a feasible robot trajectory based on the facial features.

II. Related Work

There is a great variety of work done on robotic painting and drawing of portraits, most of which limits itself to a 2D surface. The work by Nasrat et. al. [10] combines high quality portraits with high speed drawing using a robot arm, the drawing is generated by using learning based methods that perform a style transfer from the original picture of a face to a drawn style which can be turned into robot trajectories. The work by Song et. al. [13] tries to enhance the interactive aspect of the drawing process by accompanying a robot manipulator, which is used to sketch the portraits, with a robot face which converses with the person being drawn by using LLMs.

Artists such as Patrick Tresset [16] have challenged traditional artistic roles by inverting the relationship between observer and subject. In "Human Study #1", the human is rendered passive, becoming the object of a robot's interpretation. The performative installation incorporates the machine

*Authors contributed equally.

The authors would like to thank Nandu Jagdish and Joaquim Ortiz-Haro for their valuable support in the early stages of this work.

[1] Technische Universität Berlin, 10623 Berlin, Germany p.effing, cobo-briesewitz@campus.tu-berlin.de

as both observer and creator, blurring the lines between tool and artist and turns the living subject into material for the non-living machine's imagination.

Light paintings have been combined with robotics within the fields of UAVs. The paper by Bauersfeld et. al. [1] uses long exposure images to show the resulting flight trajectories of their UAVs. While the work by Serpiva et. al. [12] proposes a system which directly allows the user to control the flight trajectory and painting path through hand gestures. The resulting light paintings can scale to large dimensions, but lack precise local control of each line due to residual forces acting on the mutirotors.

Other work has also used manipulators to perform light paintings. The work by Li et. al. [9] creates light paintings using a robot manipulator to represent sound data from the public. The work by Huang et. al. [5] performs light paintings of predefined 3D meshes by using a sweeping technique utilizing a flat panel display.

III. Background

Motion planning for robots often requires generating feasible paths while satisfying a variety of constraints. Two important approaches in this area are sampling-based planning methods, such as Rapidly-exploring Random Trees (RRT), and optimization-based methods, particularly Nonlinear Programming (NLP). While sampling-based algorithms are effective at quickly exploring high-dimensional spaces, optimization methods are well-suited for refining trajectories to satisfy complex constraints.

A. Rapidly-exploring Random Trees

RRT [8] is a classic path-finding algorithm used in robotics, with many variants developed [6], [7]. Typically, RRT explores a space by randomly sampling points and connecting them to a tree structure. The tree starts at a given location and grows until they reach a target point, yielding a path from start to goal.

B. Nonlinear Programming for Robotic Trajectory Optimization

The general form of a Nonlinear Program (NLP) can be written as:

$$\begin{aligned} \min_{x} \quad & f(x) \\ \text{subject to} \quad & g(x) \leq 0, \\ & h(x) = 0, \end{aligned}$$

where $f(x)$ denotes the objective function (i.e., the cost to be minimized), $g(x)$ represents the set of inequality constraints (such as maintaining a minimum distance from obstacles or ensuring an end-effector remains above a threshold), and $h(x)$ denotes the set of equality constraints (such as enforcing a specific position or precisely achieving a desired gripper orientation).

IV. Method

We divide the problem into separate stages, these being: capturing a human face, generating a voxel grid divided into clusters and generating and traversing separate RRTs on the voxel clusters. An overview of the different stages of the portrait generation process can be seen in Figure 2.

A. Capturing the Target Face

In the first stage of our process we capture a human face from the environment by using a wrist mounted camera on the robot. The robot randomly points the camera around the surrounding scene until it detects a human face using the method described in [17]. Using an RGB-D camera we are able to extract a point cloud from the image which gives us the necessary information for making a 3D painting.

In order to identify the relevant areas that should be painted, we apply Gaussian blur on a grayscale version of the original image, followed by an adaptive threshold [11] through which we create a bitmap of the image divided into black and white. We want the edges to be wide so that the RRTs can spread in a large area, generating a greater illusion of shading rather than very thin, rugged lines. Therefore, we start by applying Gaussian blur before applying the adaptive threshold.

Once we have isolated the pixels in the image that correspond to edges, we generate a point cloud by using the depth data from the RGB-D image, considering only the edge pixels and excluding those that do not contain any edge information.

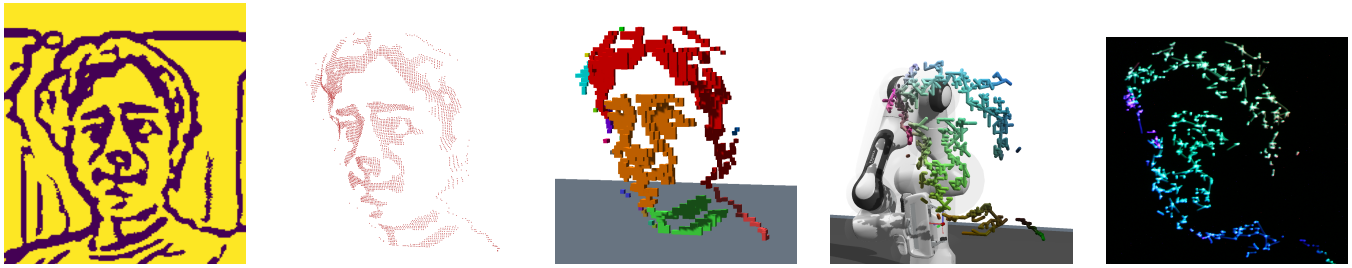
B. Generating Voxel Clusters

To generate the final trajectory, we use voxel grids to represent the free space, which is then explored by the RRT path-finding algorithm. We start by defining a voxel size v which we use to discretize the point cloud. By computing the bounding box around the point cloud we can discretize the space and divide it into voxels of size $v \times v \times v$. Each voxel stores a binary value, specifying wether it is filled or not. This boolean value is set by checking if any point in the point cloud is contained inside of the respective voxel. The Algorithm is shown in 1.

After computing the voxel grid, we assign each voxel to a specific cluster. A voxel belongs to the same cluster as all its neighboring voxels, as shown in Figure 2 (c). Any voxel which has no neighbors is ignored in the final painting.

C. Path Generation

A distinct instance of the Rapidly-exploring Random Tree (RRT) algorithm is executed for each voxel cluster. For each individual execution, a centrally located start point is initially placed within the free space defined by the boundaries of the each cluster. This ensures that the tree starts from a valid region of the space. Once initialized, the RRT algorithm proceeds with its exploration, iterating for a predetermined number of steps. During this process, the algorithm incrementally grows a tree by randomly sampling points in the free space and extending connections to nearby



(a) Adaptive Threshold (b) Point Cloud (c) Voxel Clusters (d) Robot Trajectory (e) Light Painting

Fig. 2: The figure shows the different stages of the light painting generation process. (a) We start by filtering the edges from RGB part of the RGB-D image of the recognized face. (b) Once we know which pixels correspond to edges on the image, we generate a point cloud which only contains said pixels, while excluding the ones that are part of the background. (c) From the point cloud we generate a voxel grid and divide it into clusters. (d) Once we have the voxel clusters we run an RRT per cluster to generate the robot end-effector trajectories. (e) Finally we run the path on the real robot.

Algorithm 1 Voxel Grid Construction

Require: Point cloud $\mathcal{P}$, voxel size v
Ensure: Voxel map $\mathcal{V}$

Find minimum coordinates: $(x_{\min}, y_{\min}, z_{\min})$
Initialize empty voxel map $\mathcal{V}$
for each point p_i in $\mathcal{P}$ **do**
  Compute voxel index:

$$\mathbf{idx}_i = \left(\left\lfloor \frac{p_i^x - x_{\min}}{v} \right\rfloor, \left\lfloor \frac{p_i^y - y_{\min}}{v} \right\rfloor, \left\lfloor \frac{p_i^z - z_{\min}}{v} \right\rfloor \right)$$

  if $\mathbf{idx}_i$ not in $\mathcal{V}$ **then**
    Insert $\mathbf{idx}_i$ into $\mathcal{V}$
  end if
end for
return $\mathcal{V}$

nodes. This is done without any predefined goal position in the free space, allowing the tree to continue expanding within the voxel cluster. After completing the fixed number of iterations or reaching a specified tree size, the resulting trees can be visualized by drawing lines that connect the nodes, effectively illustrating the paths that the algorithm has explored in the cluster's free space.

The light source can then trace the path defined by the RRT. To solve the inverse kinematics for the paths we use K-Order Markov Optimization (KOMO) [15], which leverages constrained optimization for robotic control. Within this framework, constraints on position, rotation, and velocity can be defined. The RRTs are transformed into NLPs by defining waypoints as position equality constraints and adding velocity constraints to ensure smooth trajectories. Velocity is critical for light painting, as it influences the brightness of the resulting lines in the final image. We also add a constraint to the end-effector rotation so that the light is always pointing towards the camera.

D. Determining Color

In order to determine the color that should be drawn, 3 different approaches were utilized: RGB cuboid mapping, original image color intensification and coloring based on distance to the RRT's root node. For each of these methods the color is queried once for the position of the starting node of a connection in the tree. The entire connection is assigned this color.

The first method defines an RGB-aligned cuboid in the end-effector space, aligning the x-, y-, and z-axes with the R-, G-, and B-axes, respectively. The cuboid's dimensions are determined by the range covered across all paths. The corresponding color-generating formulas are shown in (1)–(3).

$$r = \frac{\text{clamp}(x, x_{\min}, x_{\max}) - x_{\min}}{x_{\max} - x_{\min}} \tag{1}$$

$$g = \frac{\text{clamp}(y, y_{\min}, y_{\max}) - y_{\min}}{y_{\max} - y_{\min}} \tag{2}$$

$$b = \frac{\text{clamp}(z, z_{\min}, z_{\max}) - z_{\min}}{z_{\max} - z_{\min}} \tag{3}$$

RGB cubiod colors equations (1)–(3)*: Mapping a 3D position to normalized RGB values based on a cuboid in end-effector space.*

Secondly, the actual colors from the RGB-D image can be used to determine the color of each line in the tree. To achieve this, each voxel in the voxel grid is assigned a color by mapping its x- and y-coordinates to corresponding pixel locations in the RGB image. The retrieved colors are saturated to enhance visual contrast. For each connection in the tree, the color of the starting node is once again used to color the entire edge. For this coloring approach, we skip the adaptive thresholding step in order to color the entire face, resulting in a single cluster and therefore a single RRT.

The last method is based on the tree structure instead of the position in end-effector space. Each node is assigned a color based on its distance to the tree's root node, where the root node has a distance of zero. With an increase in distance from the root, the color gradually shifts from red

to blue. This allows for a more comprehensive view on the trees. The corresponding equations are (4)–(5).

$$r_d = 1.0 - d/d_{max} \tag{4}$$

$$b_d = 1.0 - r_d \tag{5}$$

Tree Depth Color Equations (4) & (5)*: Color is computed based on the node's tree depth d, defined as the number of hops from the root node and the maximum depth of all trees* d_{max}

V. Experiments

We use a Franka Emika robot for our experiments. The robot is equipped with a RealSense camera which is attached to the end-effector and is used to capture RGB-D images. The arm also holds a small box or "light wand" that controls an RGB light, which can be wirelessly operated via Wi-Fi. For capturing the light paintings we use a Canon EOS 500D camera with an ND filter to filter out environmental light pollution and allow for cleaner photos and longer exposure times.

A. Light Wand

For the experiments, we used a custom-built light source that enables wireless control of both color and luminance of the light wand. The light source is based around an ESP32 micro-controller [14], whose built-in Wi-Fi connectivity in combination with a lithium-ion battery as the power source allows a fully wireless operation. The light source itself consists of a 5 mm RGB LED driven by three NPN bipolar transistors and controlled via PWM signals of the micro-controller. Commands to adjust the color and intensity of the LED are sent from the control computer via transmission control protocol (TCP) over a local area network (LAN). All components are housed in a 3D-printed enclosure, allowing easy handling by the robot.

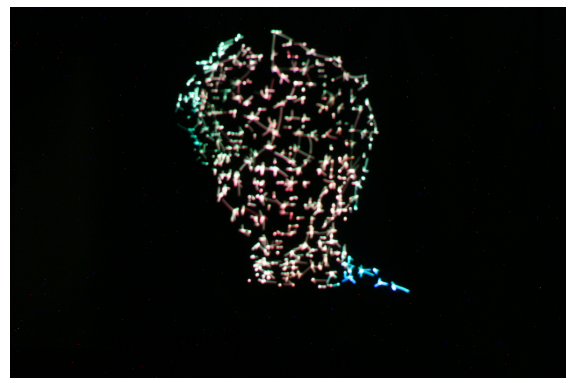
B. Results

We tested our method on the three different coloring approaches described in Section IV. The resulting images are shown in Figure 3. Figure 4 shows one light painting from three different view points. Each painting takes 8-15 minutes and has dimensions of up to $0.55 \times 0.55 \times 0.55$ meters.

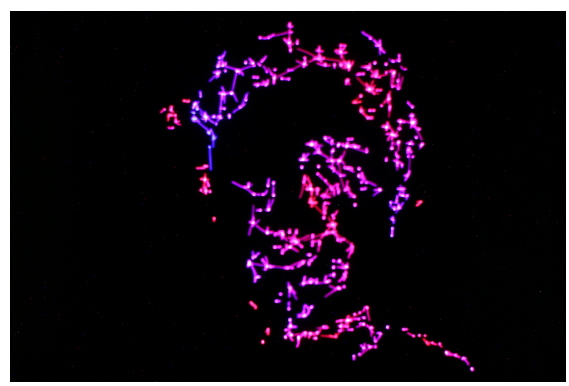
VI. Conclusion

The world is moving into a future where humans and machines collaborate to achieve goals. While Robotics has traditionally focused on closed warehouses with limited human interaction, advancements in AI technology are enabling richer human-machine interactions. Our proposal emphasizes these developments, envisioning a future where robots not only assist with ordinary tasks but also contribute to artistic beauty. We incorporated human and robot into an open ended loop of mutual observation and understanding while utilizing an established algorithm to create unique artworks emerging from simple structures.

(a) RGB Cuboid

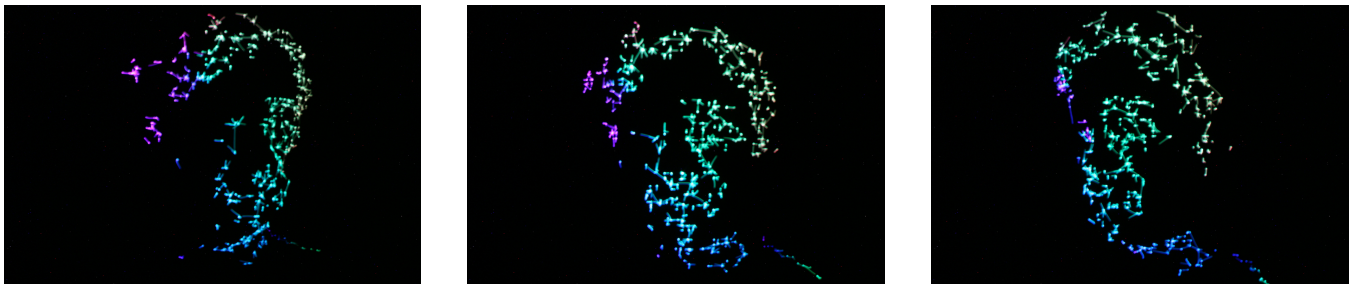


(b) Pixel Colors



(c) Root Node Gradient

Fig. 3: Three different coloring approaches for the light painting. (a) The RGB Cuboid approach maps the color of the stroke to the position of the end-effector in 3D space. (b) Another alternative is to use the original colors on the image. (c) The final coloring approach produces a gradient from the root nodes to the leaf nodes highlighting the different RRT trees in each voxel cluster.



(a) Left (b) Front (c) Right

Fig. 4: Resulting light painting from three different angles using an RGB cuboid for coloring.

Some potential future work will be to look into mobile robots for making the light paintings. This would enhance the capabilities of the current system by allowing to generate bigger paintings with more intricate patterns. In addition to this, future work might look into using multi-robot systems to speed up the painting generation process and enhance the overall artistic experience of the audience. While this work looks at more traditional path finding algorithms for generating the painting, there is great potential in using deep learning based generative models for doing the style transfer from the original image. Using learning based methods might also help in improving the depth image quality.

References

[1] Leonard Bauersfeld, Elia Kaufmann, Philipp Foehn, Sihao Sun, and Davide Scaramuzza. Neurobem: Hybrid aerodynamic quadrotor model. In *Proceedings of Robotics: Science and Systems*, Virtual, July 2021.

[2] Sylvain Bertrand, Lionel Prevost, Felicia Ionascu, Alexandre Briere, Raphael Koskas, Richard Taquet, Falitiana Andrianantoandro, and Meyssan Kanzari. Light painting with mobile robots as motivating projects for robotics and control education. In Wilfried Lepuschitz, Munir Merdan, Gottfried Koppensteiner, Richard Balogh, and David Obdržálek, editors, *Robotics in Education*, pages 27–38, Cham, 2022. Springer International Publishing.

[3] Sylvain Calinon, J. Epiney, and Aude Billard. A humanoid robot drawing human portraits. In *Proceedings of the 2005 5th IEEE-RAS International Conference on Humanoid Robots*, pages 161–166. IEEE, 2005.

[4] Sami Haddadin, Sven Parusel, Lars Johannsmeier, Saskia Golz, Simon Gabl, Florian Walch, Mohamadreza Sabaghian, Christoph Jähne, Lukas Hausperger, and Simon Haddadin. The franka emika robot: A reference platform for robotics research and education. *IEEE Robotics and Automation Magazine*, 29(2):46–64, 2022.

[5] Yaozhun Huang, Sze-Chun Tsang, Hei-Ting Tamar Wong, and Miu-Ling Lam. Computational Light Painting and Kinetic Photography . In Tunç Aydın and Daniel Sýkora, editors, *Expressive: Computational Aesthetics, Sketch-Based Interfaces and Modeling, Non-Photorealistic Animation and Rendering*. ACM, 2018.

[6] Sertac Karaman and Emilio Frazzoli. Sampling-based algorithms for optimal motion planning. In *The International Journal of Robotics Research*, volume 30, pages 846–894. SAGE Publications, 2011.

[7] James J. Kuffner and Steven M. LaValle. Rrt-connect: An efficient approach to single-query path planning. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 995–1001. IEEE, 2000.

[8] Steven M. LaValle. Rapidly-exploring random trees: A new tool for path planning. In *Technical Report TR 98-11*, 1998.

[9] Zoe Qi-Jing Li. I tell the moon my secret and the moon tells me yours. In *ACM SIGGRAPH 2024 Art Gallery*, SIGGRAPH '24, New York, NY, USA, 2024. Association for Computing Machinery.

[10] Shady Nasrat, Taewoong Kang, Jinwoo Park, Joonyoung Kim, and Seung-Joon Yi. Artistic robotic arm: Drawing portraits on physical canvas under 80 seconds. *Sensors*, 23(12), 2023.

[11] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, 1979.

[12] Valerii Serpiva, Ekaterina Karmanova, Aleksey Fedoseev, Stepan Perminov, and Dzmitry Tsetserukou. Dronepaint: Swarm light painting with dnn-based gesture recognition. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Emerging Technologies*, SIGGRAPH '21, page 1–4. ACM, August 2021.

[13] Jookyung Song, Mookyoung Kang, and Nojun Kwak. Sketcherx: Ai-driven interactive robotic drawing with diffusion model and vectorization techniques, 2024.

[14] Espressif Systems. Esp32: Wi-fi and bluetooth soc. `https://www.espressif.com/en/products/socs/esp32`. Accessed: 2024-12-13.

[15] Marc Toussaint. Newton methods for k-order markov constrained motion problems. 2014.

[16] Patrick Tresset. Human study 1. https://patricktresset.com/new/project/5-rnp/, 2011-2024. Accessed: 2025-04-28.

[17] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, volume 1, pages I–I, 2001.

[18] Wikipedia contributors. Light painting — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/wiki/Light_painting. Accessed: 2024-12-13.