Large-Scale Painting of Photographs by Interactive Optimization

Romain Prévost[a,b], Alec Jacobson[a,c], Wojciech Jarosz[b,d], Olga Sorkine-Hornung[a]

[a] *ETH Zurich*
[b] *Disney Research*
[c] *Columbia University*
[d] *Dartmouth College*

Abstract

We propose a system for painting large-scale murals of arbitrary input photographs. To that end, we choose spray paint, which is easy to use and affordable, yet requires skill to create interesting murals. An untrained user simply waves a programmatically actuated spray can in front of the canvas. Our system tracks the can's position and determines the optimal amount of paint to disperse to best approximate the input image. We accurately calibrate our spray paint simulation model in a pre-process and devise optimization routines for run-time paint dispersal decisions. Our setup is light-weight: it includes two webcams and QR-coded cubes for tracking, and a small actuation device for the spray can, attached via a 3D-printed mount. The system performs at haptic rates, which allows the user – informed by a visualization of the image residual – to guide the system interactively to recover low frequency features. We validate our pipeline for a variety of grayscale and color input images and present results in simulation and physically realized murals.

Keywords: interactive spray painting, painting approximation.

1. Introduction

Spray paint is affordable and easy to use. As a result, large-scale spray paint murals are ubiquitous and take a prominent place in modern culture (see Ganz [1] for many examples). Spray painters may cover large "canvases", such as walls of buildings, with minimal scaffolding hassle, and the diffusive spray allows spatially-graded color mixing on the fly. However, manual creation of interesting spray paintings is currently restricted to skilled artists. In addition, the large scale of the painting, compared to the close-range spraying (from distances of 10–40 centimeters) makes it challenging to orient and precisely position oneself for accurate spraying, forcing the artist to keep a global vision while focusing on local changes.

Though traditional (e.g. inkjet) printing on large-format paper is possible, it requires access to expensive non-standard equipment. Further, depending on the target surface, it may be impossible to attach paper or canvas. Paint provides a practical alternative, and decidedly induces a certain aesthetic character. Naïve solutions to assisted spray painting, such as procedural dithering or half-toning, are tedious and do not take advantage of the painter's perceptual expertise: A human can easily tell which important areas of an image need further detail. Non-interactive systems inherently lack this ability, potentially wasting precious painting time on unimportant regions. Stenciling is another obvious candidate, but it necessitates quantization to solid colors and may require many topologically complex stencils. Large-scale murals would also require cumbersome large-scale stencils. Finally, stencils do not necessarily inherit any aesthetics particular to spray painting: the same paintings could be made with brushes or rollers.

Our solution is a "smart" spray can. From a high level, rather than spraying a solid color, our can *sprays a photograph* (see Figure 2). Our system tracks the position and orientation of the spray can held by the user, who may be regarded as a cheap alternative to a robotic arm. Our optimization then determines on-the-fly how much paint to spray, or, more precisely, how long to spray, and issues appropriate commands to an actuating device attached to the spray can (see Figure 1). By simultaneously simulating the spraying process, we visualize a residual image that indicates to the user the locations on the mural that could benefit from more painting (see Figure 3). We also monitor the *potential residual* as well as the *potential benefit* for the current spray can color. These properties are respectively the maximum

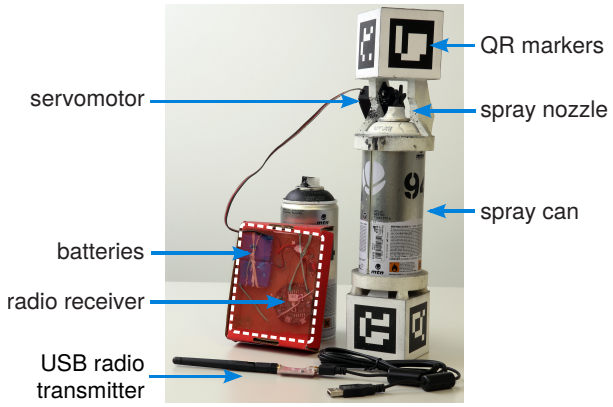


Figure 1: Paint commands are transmitted via radio directly connected to a servomotor actuating the spray nozzle. Two webcams track QR markers pasted to a 3D-printed mount.

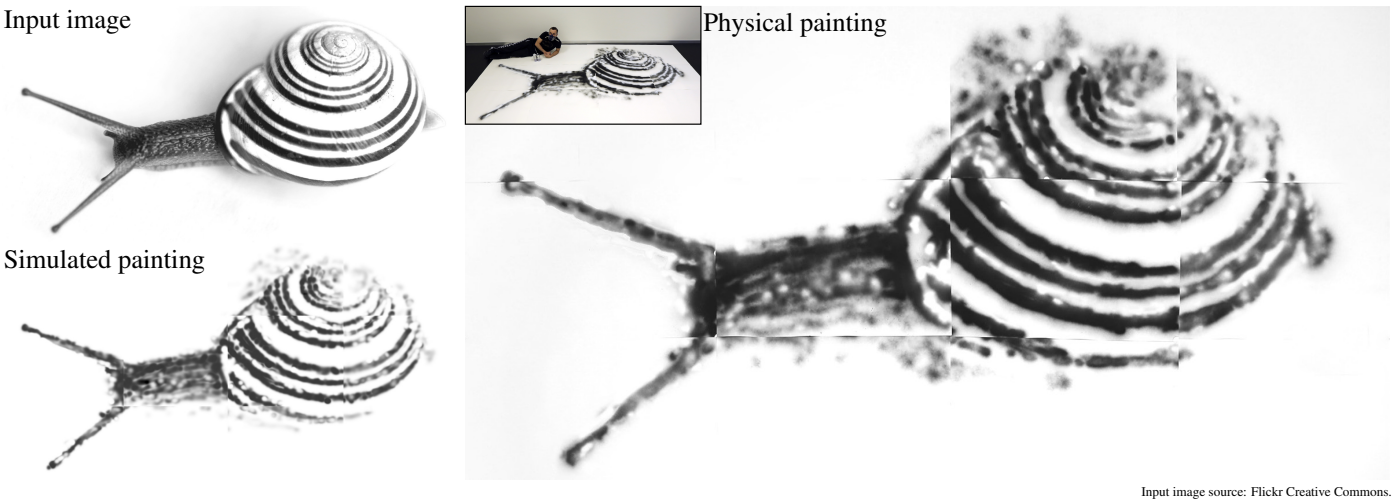


Figure 2: Our device and real-time optimization enables spray painting murals of any input photograph. We track the user's movements and simulate the spraying process to choose the optimal amount of paint to spray in order to best approximate the input image.

amount of error that can possibly be reduced by adding more paint of the current color, and the expected effect of adding more of the current color. When little progress can be made with the current color, the user is prompted to switch color, and the process is repeated until satisfaction.

We demonstrate the effectiveness of this process for a variety of input images. We present physically realized paintings, as well as simulated results. The physical murals validate that our simulation matches reality and show that our model captures the image content while preserving some of the spray paint aesthetic.

2. Historical perspective and related work

Computer-aided painting is an old and well-studied subject among both scientists and artists. Artist Desmond Paul Henry unveiled his *Henry Drawing Machine* in 1962. This machine created physical realizations of procedurally generated drawings. One year later, Ivan Sutherland's famous SKETCHPAD pioneered interactive virtual interfaces for drawing and modeling. Now, the modern frontier of research in computer-aided painting is more specialized and spans a variety of interfaces and applications.

The E-DAVID robot is an actuated arm that grips a paint brush and applies paint to a physical canvas with human-like strokes [2, 3]. The robot follows a predetermined sequence of strokes which approximate an input grayscale image, with optional dynamic updating of this sequence based on visual feedback to account for inaccuracies. Other robotic art systems have been developed, such as Yao's Chinese painting robot [4] or Paul the sketching robot [5]. None of these systems would be appropriate for large-scale spray painting. In particular, using a robotic arm is not an affordable option for the canvas size we target. Artist Uri Lehni presented a spray painting system comprised of an actuated spray can moving automatically across the 2D span of a wall [6]. This fully automatic system is limited to preprogrammed curves, creating line drawings with a single color. Our interactive system utilizes the natural frequency range of the spraying process when varying the distance of the can to the canvas. This allows a better span of the color spectrum with a small number of paint colors.

Another branch of research has focused on assisting novice users in various artistic creation tasks: painting [7], drawing [8, 9], 2D manufacturing [10], sculpting [11, 12], architecture [13], and airbrushing [14, 15]. These projects provide access to artistic techniques to unskilled users without the ambition to train them. Shilkrot and colleagues' concurrent work presents an augmented airbrush, allowing novice users to experience the art of spray painting. Although our work shares a similar approach, our system differs in many aspects. Shilkrot and colleagues designed an expensive automated airbrush, whereas our device is a very inexpensive, plug-and-play mechanism that can be mounted on any standard spray can. While being more precise, their magnetic tracking system would be challenging to adapt to the scale of large murals. We instead use a vision-based tracking

Figure 3: For our physical experiments, the user waves our modified spray can before a 1.5 m^2 canvas. The monitor to the left shows the *potential residual* for the current color, as well as the spray can location, helping the user to position himself and to find areas to improve. See the accompanying video for a full painting session.

system, which proves reliable and represents a more affordable option. On the other hand, their system is acting as a fail-safe to assist novice users and potentially train them in the long term, whereas in our system the user's role is more passive.

A vast amount of previous research has dealt with creating paintings virtually. We may separate these works into those that apply a painterly rendering style to an existing image [16, 17, 18] and those that simulate painting virtually to allow users to create new images interactively. The latter group spans a variety of painting styles: brush painting [19, 20, 21, 22], watercoloring [23, 24], fluid jet (cf. Jackson Pollock) [25], ink splattering [26], and – most relevant to our work – air brushing [27] and spray painting [28]. Konieczny and Meyer's airbrushing system refashions a standard airbrush as an input device to a simulation that faithfully replicates the airbrushing process virtually. Their simulation interpolates splats between frames, similar to our method or using infinitesimal spacing in Adobe® Photoshop's® brush tool. Our system works with a single paint color at a time, so we do not require the more complicated Kubelka-Munk paint mixing model, opting instead for a time-continuous form of the alpha-compositing `over` operator [29].

An alternative to our continuous optimization might be to lean on previous painterly rendering algorithms to build a discrete set of paths, strokes or patches. In particular, we refer the reader to Hertzmann's survey on stroke-based rendering [30]. Image abstraction or segmentation techniques (e.g. DeCarlo and Santella [31]) have been altered to design stencils fit for physical painting [32]. These methods inherently simplify the input image, eliminating details and color gradients. The mosaics of Hausner [33] or Kim and Pellacini [34] offer another beautiful form of abstraction, but require significant planning and organization compared to our computer-assisted spray painting. Finally, none of these alternatives take advantage of the diffusion process characteristic of spray paint, which we leverage, for example, to blend gray tones from only black and white paint.

In engineering, there exist sub-fields of research devoted to high-resolution simulation of the spray painting process, typically centered around electro-static sprays used in the automotive industry [35, 36]. These models are far too complicated and fine-grained for our needs. They simulate individual particles with the goal of achieving constant paint film thickness across optimized robotic painting paths [37, 38].

3. Method

The user will stand before a *canvas* (e.g. wall or sheet of paper) and wave a programmatically actuated spray can equipped with a wireless receiver. Running on a nearby computer, our real-time algorithm determines the optimal amount of paint of the current color to spray at the spray can's tracked location. Our run-time system can be broken down into four parts: 1) physically actuating the spray can, 2) spray can tracking, 3) simulating the spray process, and 4) optimizing the amount of paint to spray. Additionally our system provides the user with guiding feedback visualization in real time.

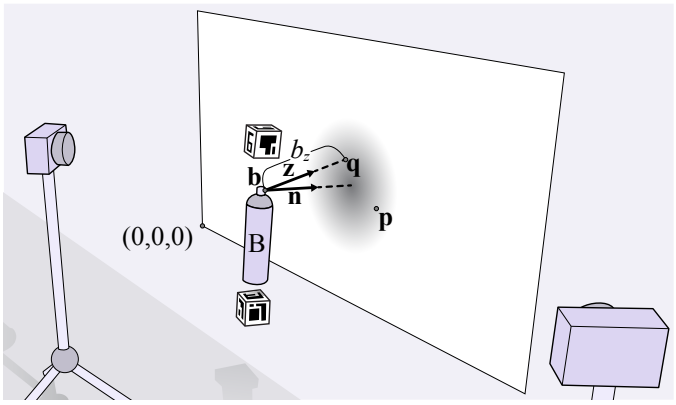


Figure 4: Two webcams on tripods track QR markers on our device to determine the spray nozzle position **b** and normal direction **n**.

3.1. Hardware and tracking

We use standard aerosol paint cans (65 mm diameter, 158 mm height, 400 ml capacity) with replaceable plastic caps: clean caps ensure consistent spray profiles. Spray caps are interchangeable, and traditional spray paint artists may use a variety of caps with different profiles: thin caps for details, wide caps for filling large areas, and caps with anisotropic profiles for stylized strokes. We found by experimenting that thin caps (specifically Montana Colors'® *94 Cap*) provide a good tradeoff in terms of resolution and diffusiveness. However, nothing restricts our system to use different caps, or paint cans. Therefore we provide additional simulated results (see Figure 12 and supplemental material) to show how other spray characteristics would spawn various artistic styles in the final painting.

When a can is in use, we mount it inside a custom-designed, 3D-printed enclosure (see Figure 1). A servomotor sits directly above the can's nozzle so that a small plastic arm attached to the rotor presses down on the nozzle when actuated. Though the paint deposition of standard spray cans is dependent on the amount of pressure applied to the nozzle, we found that this is difficult to predict and instead treat the can as either off (no pressure, arm up) or on (full pressure, arm down). Given the tracked position and orientation of the nozzle and the current status of the painting, our optimization will determine how *much* paint to spray by determining *whether or not* to spray for a small fixed time step.

We assume that the working canvas is a flat surface of roughly 1 m × 1.5 m size. In our experiments, rather than scale or move our setup to cover larger canvases, we subdivided input images and paint each part on a large sheet of A0 paper (see Figure 2). We track the position of our device using two webcams on tripods facing the canvas from the left and right side (see Figure 4). The cameras are calibrated once using OpenCV[1] by placing a checkerboard reference image on the canvas. Mounted in our enclosure, the nozzle of the spray can sits at a known position and orientation with respect to two 3D-printed cubes atop and below it. At run-time, we use the ArUcO[2] library to

[1] http://opencv.org/

[2] http://www.uco.es/investiga/grupos/ava/node/26

track QR markers pasted to each face of the cubes. We take a median across markers and cameras to compute a 3D position and orientation of the spray nozzle. The cubes are rotated out of phase by 45° to ensure that at least one marker is fully visible at any time. In practice, usually four markers are tracked simultaneously with precision on the order of 5 mm and 1°.

Without loss of generality, let $\mathbf{b} \in \mathbb{R}^3$ be the 3D position of the spray nozzle so that $|b_z|$ is the orthogonal distance to the 2D canvas (with $\mathbf{z}$ pointing towards the canvas, see Figure 4). By construction, our system is invariant to rotations in the canvas plane, thus we only need to know the outward normal direction at the nozzle $\mathbf{n} \in \mathbb{R}^3$. Since these change over time, we will use $\mathbf{x}(t)$ to denote the quantity $\mathbf{x}$ at time t when necessary.

3.2. Spray painting simulation

Our optimization will decide whether or not to spray based on the nozzle position $\mathbf{b}(t)$, direction $\mathbf{n}(t)$ and the current status of the painting. Dynamically capturing the spray painted canvas (e.g. with a third camera) is difficult and prone to error. The painter and device would occlude much of the canvas, and, more importantly, the region being sprayed. Further, quality capturing would depend heavily on the camera and lighting conditions. Instead we propose simulating the spray painting process in tandem with the physical painting. This simulated approximation of the physically realized painting may then be used immediately in our optimization scheme.

We now derive a simple model for spray painting simulation and then provide a calibration procedure to fit parameters which best decrease its approximation error.

Given the spray nozzle position $\mathbf{b}(t)$, direction $\mathbf{n}(t)$ and a duration of spraying Δt, we would like to approximate the new color at each point $\mathbf{p} = (p_x, p_y, 0)$ on the canvas. High-resolution spray painting simulators model the spray process by simulating individual particles or ray casting [39]. These methods would be too costly for our real-time simulation, so instead we take this discrete particle point of view as motivation for a continuous approximation. Intuitively, imagine that each small region on the canvas can only hold a fixed number of small paint particles. For each new paint particle arriving, a current particle (with a possibly different color) must be bumped out. This grossly simplifies the paint mixing process. However, we observe that since we spray only a single color at a time and spray paints dry quickly that little to no complicated paint mixing occurs.

Let $\mathbf{C}(\mathbf{p}, t) : \mathbb{R}^3 \times \mathbb{R}^+ \to \mathbb{R}^3$ be the RGB color on the canvas at position $\mathbf{p}$ and time t. Then our continuous *particle replacement policy* may be formulated succinctly by writing the new color at a position $\mathbf{C}(\mathbf{p}, t + \delta t)$ as a linear blend between the current color $\mathbf{C}(\mathbf{p}, t)$ at that position and the current paint color $\mathbf{B} \in \mathbb{R}^3$:

$$\mathbf{C}(\mathbf{p}, t + \delta t) = \mathbf{C}(\mathbf{p}, t) + f(\mathbf{b}(t), \mathbf{n}(t); \mathbf{p})\delta t \left(\mathbf{B} - \mathbf{C}(\mathbf{p}, t)\right), \quad (1)$$

where the blending factor $f(\mathbf{b}(t), \mathbf{n}(t); \mathbf{p})\delta t$ is the spray paint *deposition profile* f (paint density per second) integrated for a small time change δt. Determined by the nozzle position and direction, this integrated profile is the paint deposited at a position on the canvas $\mathbf{p}$ during an infinitesimal amount of time. For now, we assume f is known, and later we will discuss calibrating to a given spray can to fit the necessary parameters.

In the limit as δt goes to zero, this blending formula becomes a first-order ordinary differential equation:

$$\frac{\partial \mathbf{C}}{\partial t}(\mathbf{p}, t) + f\mathbf{C}(\mathbf{p}, t) = f\mathbf{B}, \quad (2)$$

where we drop the arguments of f for brevity. Solving this simple ODE for $\mathbf{C}$ reveals a continuous expression for the new color:

$$\mathbf{C}(\mathbf{p}, t) = \mathbf{B} + \exp\left(-\int_{t_0}^{t} f(u)du\right)(\mathbf{C}(\mathbf{p}, t_0) - \mathbf{B}), \quad (3)$$

where $f(u) = f(\mathbf{b}(u), \mathbf{n}(u); \mathbf{p})$. The linear blending factor is now an exponential corresponding to the accumulated deposition profile since time t_0.

What remains is a full description of the deposition profile f. Previous methods have used 2D parabolic functions, bivariate Gaussians or more complicated and general profiles [39, 40]. These models appear well justified for electro-static spray guns, but we found that a radially symmetric 2D Gaussian is sufficiently accurate and general for standard aerosol spray paints (see Figure 5).

Let us first assume that the spray nozzle is directed perpendicularly to the canvas, i.e. $\mathbf{n} = \mathbf{z} = (0, 0, 1)$, then we may write f as a 2D Gaussian:

$$f_\perp(\mathbf{b}, \mathbf{n}; \mathbf{p}) = \frac{\beta}{2\pi r^2} \exp\left(-\frac{\|(\mathbf{p} - \mathbf{b}) \times \mathbf{n}\|^2}{2r^2}\right) \quad (4)$$

where β is the average paint deposition amount, r is the standard deviation or intuitively the "width" of the profile's bell shape. Both these parameters depend on the distance along the nozzle direction $|(\mathbf{p} - \mathbf{b}) \cdot \mathbf{n}|$, which we dropped here for brevity's sake. In the next subsection we will present a calibration scheme to fit β and r.

We can generalize this Gaussian model to allow an arbitrarily tilted spray nozzle direction $\mathbf{n}$. We bring the profile $f_\perp$ in the directional domain by dividing by the Jacobian with respect to the angle between $(\mathbf{p}-\mathbf{b})$ and $\mathbf{n}$. We then rotate the profile by the spray nozzle tilt angle $\cos^{-1}(\mathbf{n} \cdot \mathbf{z})$, and bring everything back to the spatial domain by multiplying by the Jacobian with respect to the planar coordinates. These Jacobians, $\cos\theta / \|\mathbf{p}-\mathbf{b}\|^2$ and its reciprocal, account for the change of paint deposition depending on the direction of an infinitesimal paint particle arriving on the canvas. Our final, general spray paint deposition profile is:

$$f(\mathbf{b}, \mathbf{n}; \mathbf{p}) = f_\perp(\mathbf{b}, \mathbf{n}; \mathbf{p}) \frac{\|\mathbf{p} - \mathbf{b}\|^3}{|(\mathbf{p} - \mathbf{b}) \cdot \mathbf{n}|} \frac{|(\mathbf{p} - \mathbf{b}) \cdot \mathbf{z}|}{\|\mathbf{p} - \mathbf{b}\|^3} \quad (5)$$

$$= f_\perp(\mathbf{b}, \mathbf{n}; \mathbf{p}) \frac{|b_z|}{|(\mathbf{p} - \mathbf{b}) \cdot \mathbf{n}|}. \quad (6)$$

Figure 6 demonstrates the range of behavior of our full spray model.

3.3. Calibration

We need accurate estimates of the parameters in our spray simulation model: namely, the color $\mathbf{B}$ for each spray paint can

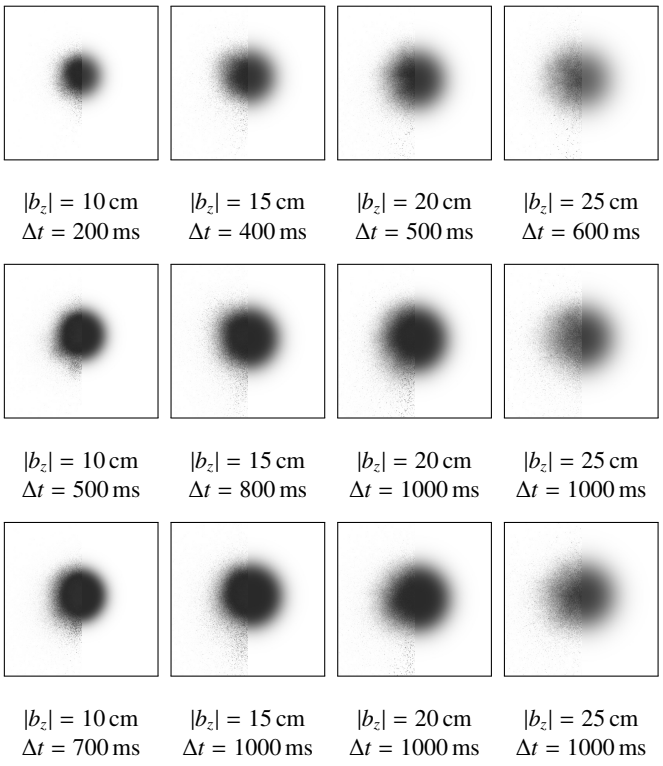


Figure 5: For each calibration sample we show the physical splat on the left half and the analytically fitted splat on the right. Zooming reveals the particle dots on the left and the smooth variations on the right. $|b_z|$ is the distance to the canvas and Δt the duration of spray actuation.

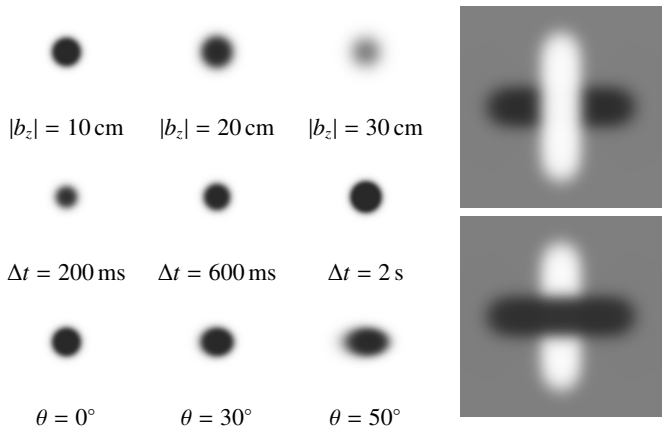


Figure 6: Left: Distance to the wall $|b_z|$ (top), spray duration Δt (middle), angle between $\mathbf{n}$ and $\mathbf{z}$ (bottom) affect our simulated spray pattern. Right: Our approach correctly models that spray painting is order-dependent: compare black-then-white to white-then-black.

and the Gaussian spray deposition profile height β and width r. These parameters could be potentially derived from manufacturer specification tables listing colors, aerosol pressures, cap diameters, etc. Instead of relying on proprietary information, we devise a generic and efficient method to measure the parameter values. The following calibration procedure is performed once at the beginning of a painting session.

Firstly, each paint color $\mathbf{B}$ is measured by scanning a sample canvas after a long spray. The parameters β and r depend on a variety of factors – most notably the choice of cap and the spray can pressure. However, we observe that they are largely consistent across paint colors. Therefore, the following procedure is simplified by assuming the spray paint color $\mathbf{B}$ is some degree of black.

Using a level to ensure orthogonal spraying direction (see Equation (4)), we position the spray can on a tripod at measured distances $|b_z| \in \{10\text{cm}, 15\text{cm}, 20\text{cm}, 25\text{cm}\}$. We actuate the spray can for given time intervals Δt, using shorter times for close distances and longer times for far. We scan each splat, storing it in a function $\mathbf{T}(\mathbf{p}) : \mathbb{R}^3 \to \mathbb{R}^3$. During calibration the canvas is known to be blank (white) at time t_0, so Equation (3) reduces to:

$$\begin{aligned}\mathbf{C}(\mathbf{p}) &= \mathbf{B} + \exp\left(-\int_{t_0}^{t} f(u)du\right)(\mathbf{1} - \mathbf{B}) \\ &= \mathbf{B} + \exp\left(-\frac{\beta\Delta t}{2\pi r^2}\ \exp\left(-\frac{\|(\mathbf{p}-\mathbf{b})\times\mathbf{n}\|^2}{2r^2}\right)\right)(\mathbf{1} - \mathbf{B}).\end{aligned} \tag{7}$$

Since the spray nozzle is fixed at a known distance $|b_z|$, direction $\mathbf{n}$ and the amount of spray time is accurately dispensed by our actuating servomotor, the only unknowns are β and r (as well as (b_x, b_y) for each image, but the values do not directly matter). We fit these values for each distance, by considering three scanned images corresponding to the same distance but with different spray durations Δt. We minimize the squared difference between our model and the scanned image summed over all scanned pixel locations $\mathbf{p}$ from all three images:

$$\underset{\beta,r}{\text{argmin}} \sum_{\Delta t} \sum_{\mathbf{p}} \|\mathbf{C}(\mathbf{p}) - \mathbf{T}(\mathbf{p})\|^2. \tag{8}$$

Starting from an approximated guess, we solve this nonlinear 2D surface fitting problem by leveraging a Levenberg-Marquardt optimizer with analytic gradients (computed from Equation (7)). Despite the nonlinearity of the function, the fitting converges in less than a minute. In Figure 5, we compare the scanned splats with their fitted analytic counterpart. For each distance, we obtain one β and one r. Repeating the procedure for four different distances, we can extract the behavior of these two functions as a function of distance. A linear representation seems appropriate for both β and r, therefore we simply fit a line (see Figure 7). These fits always show that spray deposition rate decreases with distance and the radius increases. Intuitively, these trends account for paint particles not arriving at the canvas because of air resistance, gravity, etc.

As a verification, we also repeat this fitting procedure with respect to duration rather than distance. However, we see nearly constant fits and therefore do not consider β and r dependent on duration.

3.4. Real-time optimization

With our actuation, tracking, simulation and calibration in tow, we proceed to our real-time optimization. Our goal is to minimize the difference between an input target image $\mathbf{T}(\mathbf{p})$ and the physical sprayed image $\mathbf{P}(\mathbf{p})$. We assume that our simulated sprayed image approximates its physical counterpart: $\mathbf{C}(\mathbf{p}, t) \approx \mathbf{P}(\mathbf{p}, t)\ \forall \mathbf{p}, \forall t$. This is empirically confirmed in the following section.

Given this, we proceed by breaking time into small steps $\Delta t = 100\,\text{ms}$, matching the response time over our radio-activated

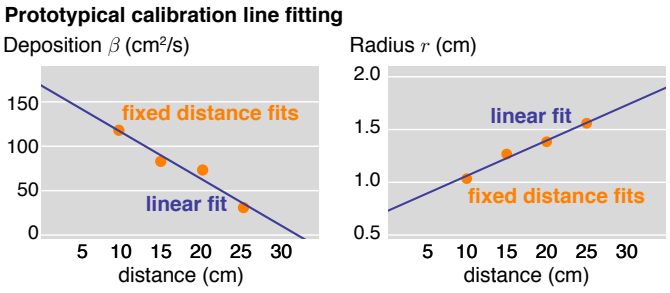


Figure 7: Our calibration fits a linear model for the rate β (on the left) and the radius r (on the right) with respect to distance.

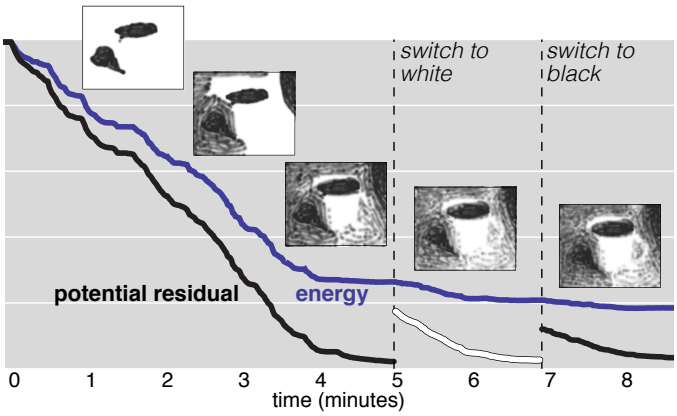


Figure 8: The potential residual depends on the current color. When near zero, the user is prompted to switch colors.

servomotor. Our tracking system runs at more than 20 Hz, so we may assume that position $\mathbf{b}$ and direction $\mathbf{n}$ are known at the beginning of the time step and also that we can accurately (via first-order extrapolation) approximate their values at the end of the time step (which has not yet taken place). For each time step, we ask whether or not to spray with the current color $\mathbf{B}$. We pose this as an energy minimization problem, where our energy can be written as a function of the simulated image at a point in time:

$$E(\mathbf{C}(t)) = \sum_{\mathbf{p}} \|\mathbf{T}(\mathbf{p}) - \mathbf{C}(\mathbf{p}, t)\|^2, \tag{9}$$

where we measure the color distance in CIE Lab for better correspondence with human perception. For each step forward in time, we would prefer to decrease this energy or at least keep it constant. We compute the predicted error at the next time step $E(\mathbf{C}(t + \Delta t))$ both for spraying with $\mathbf{B}$ and for not spraying by linearly extrapolating the trajectory. If the energy decreases more than a certain small threshold (accounting for unknown noise and error), then we choose to spray, triggering the actuator. If not, the actuator waits for the next time step and the energy and simulation remain the same. We update the simulation $\mathbf{C}(\mathbf{p}, t + \Delta t)$ according to this decision as soon as the trajectory for $t \to t + \Delta t$ is known.

We additionally have a more conservative mode, which compares $E(\mathbf{C}(t+\Delta t))$ if spraying with any one of the available colors and not spraying. We choose to spray with $\mathbf{B}$ if this choice has the minimal expected energy. Intuitively, this assumes that some time in the future the user will pass by the same region with other, possibly more beneficial, paint colors. All our results were produced starting with the conservative mode and switching to the normal mode afterwards.

Finally, we also show simulated results where at each time step we choose the best predicted improvement among all available colors. This mode presumes a device with multiple spray cans attached simultaneously (which we have not built in practice), but provides some insights on the benefits and limitations of our system. In Figure 12 and in the supplemental material, we refer to this mode as the *optimal strategy*.

3.5. Feedback and visualization

To provide guidance to the user, we display a pseudocolor visualization of the current error $E(\mathbf{C}(t))$ (Equation (9)) on a nearby monitor (see Figure 3). This visualization helps especially during the first passes over the canvas when the target content is just starting to appear. Additionally, to facilitate the user's positioning, we display a cursor and a range corresponding to the area of the canvas targeted by the spray can given its current position and orientation.

We also plot the maximum possible amount of energy that may be reduced using the current color, or *potential residual*. For each pixel, we compute whether adding paint with the current color $\mathbf{B}$ would decrease or increase the overall energy. We solve a tiny optimization problem for each pixel $\mathbf{p}_i$:

$$\begin{aligned} \underset{\alpha_i}{\text{argmin}} \quad & \|\mathbf{T}(\mathbf{p}_i) - (\mathbf{C}(\mathbf{p}_i, t) + \alpha_i (\mathbf{B} - \mathbf{C}(\mathbf{p}_i, t)))\|^2, \\ \text{subject to} \quad & \alpha_i > 0, \end{aligned} \tag{10}$$

where the non-negativity constraint ensures that paint is added and not removed. This is easily solved by first solving without the positivity constraint:

$$\alpha_i^* = \frac{(\mathbf{B} - \mathbf{C}(\mathbf{p}_i, t))^\mathsf{T} (\mathbf{T} - \mathbf{C}(\mathbf{p}_i, t))}{(\mathbf{B} - \mathbf{C}(\mathbf{p}_i, t))^\mathsf{T} (\mathbf{B} - \mathbf{C}(\mathbf{p}_i, t))}, \tag{11}$$

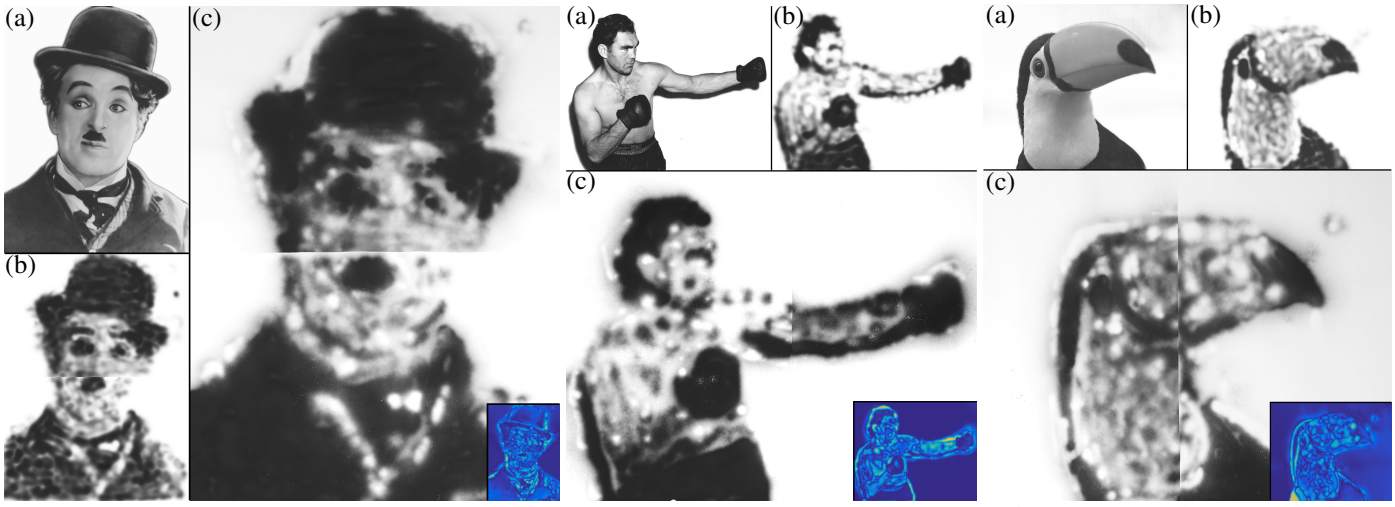
and then only keeping the result if positive: $\alpha_i = \max(\alpha_i^*, 0)$.

The total *potential residual* $R_\mathbf{B}(t)$ for a given color $\mathbf{B}$ is the sum of the individual per-pixel residuals masked by whether α_i is positive:

$$R_\mathbf{B}(t) = \sum_{\mathbf{p}_i} \begin{cases} \|\mathbf{T}(\mathbf{p}_i) - \mathbf{C}(\mathbf{p}_i, t)\|^2 & \alpha_i > 0, \\ 0 & \text{otherwise.} \end{cases} \tag{12}$$

This quantity proves very useful in practice, see Figure 8 for a visualization. If the potential residual for the current color $R_{\mathbf{B}_1}(t)$ vanishes, then no more progress can be made with the current color. Hence, this is a hint to the user that a paint color switch is in order. Once the paint can is exchanged, the potential residual for the new color $R_{\mathbf{B}_2}(t)$ is recomputed and the user may continue iterating until satisfaction.

We also provide the user a per-pixel visualization of the *potential benefit*. We map the numerator of Equation (11) to a hue color between blue (negative effect) and red (positive effect), passing by green (no benefit). Indeed, this quantity is



Original images (subfigures (a)) source: Flickr Creative Commons & Wikimedia Commons.

Figure 9: *Chaplin*, the *Boxer* and the *Toucan* have been physically realized. These demonstrate how closely our simulation (b) matches the physical realization (c) and recover a low frequency version of the target (a). Additionally, we show difference images between the target and the physical painting in the bottom right corners.

close to zero when the painting matches the target, in which case adding more paint can only make things worse, or matches the current paint color, in which case adding more paint would have no effect. When the current paint color differs from the painting, there is a potential benefit in spraying. The dot product in Equation (11) is positive/negative if adding the current paint color would improve/degrade the painting, respectively. Therefore, the user can also rely on this visualization to paint in the positive-effect areas.

4. Results

We have validated our system by spray painting a set of photographs, as shown in Figures 2, 9, 10, 11, 12 and the accompanying video. Table 1 presents some statistics of our experiments.

A typical painting session begins by registering the cameras with respect to the wall. Then the user can directly start spraying the chosen input image. An extract of a typical painting session is shown in Figure 3. A monitor showing the current potential residual helps the user determine which part of the image may be improved with the current color. When satisfied with the first layer, the user switches the spray can to a different color and continues. The energy rapidly decreases and then stabilizes after a while, and the potential residual is a good indicator when to switch cans (see also Figure 8). A session usually lasts between 10 and 15 minutes for one A0-format-sized canvas for grayscale images, and about 25 to 30 minutes for color paintings, which require more color switches. The accompanying video demonstrates a full painting session.

Painting on a physical canvas. Due to the difficulty of obtaining permission to spray paint a building, and the challenges arising from unpredictable weather conditions (e.g. wind, rain), we opted to paint on paper sheets of A0 size. This also facilitated convenient archiving and scanning of our results. To create larger paintings, we divided the input images into segments arranged in a grid, each corresponding to a single paper sheet, and spray painted each sheet separately. Figure 9 and 11 present four physical paintings realized with our system. The *Snail* (Figure 2) is a larger-scale example realized by combining twelve A0 sheets, covering a total surface of almost 12 m^2. These results empirically validate that our spraying model closely matches the physical process and truly scales to large canvases by simply translating the setup. Additionally, we present difference images, which show how our system is able to reproduce the low-frequency features of the target image (for more details we refer the reader to our supplemental material).

Painting on a virtual canvas. In addition to physical spray painting, we ran several virtual simulations using a large LCD display as our virtual canvas coupled with either our tracking system or a mouse (no tilt in this case) to define the spray can trajectory. For virtual paintings larger than the display, we split the image into a grid and paint each region in succession, as in the physical setup, while always considering a single, larger virtual canvas. As such, painting at the border of a region can affect the neighboring regions, thus avoiding artifacts due to mismatch between individual paper sheets. This shows again how our system scales by simply translating the tracking setup to the current working area. In theory, our framework even parallelizes: different painters in charge of each color could work simultaneously. In Figure 10 we evaluate our virtual setup on the target images from Shilkrot et al. [15]. We use different types of paint and different painting tools resulting in different aesthetics. While their results can appear aesthetically more pleasing, our paintings usually match our simulations more closely, because we prevent effects which we cannot replicate in simulation (e.g. paint advection or runoffs). A more quantitative comparison is difficult, because this would require using both systems in the same settings.

Painting aesthetics. In Figure 12 and in our supplemental material, we show how our system retains enough freedom to allow for very different painting aesthetics. In particular, the pointillistic style of our physical results can be attenuated by changing some of the properties of the device (e.g. trigger duration), the

Original images (top insets) source: www.123rf.com, used with permission.

Figure 10: The *Tiger* and *Frog* are two large-scale examples realized virtually with our system. We only use up to five different paint colors. While being limited in the color spectrum, our optimization in CIE Lab space is able to find a perceptually acceptable match. The two inset images show the target, as well as Shilkrot et al.'s results.

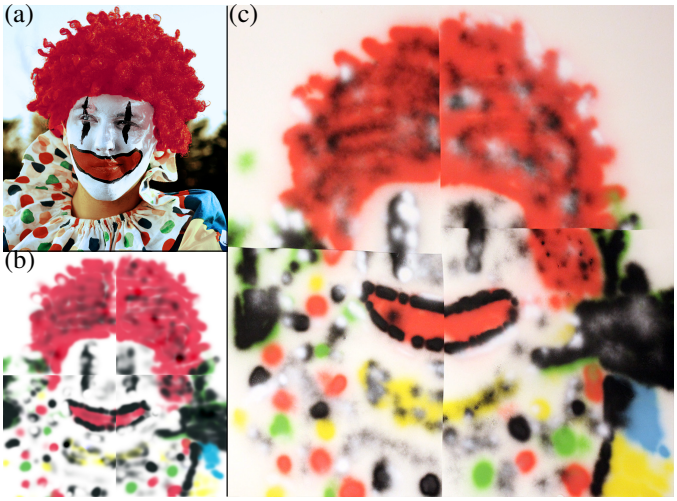
specifications of the spray can (e.g. deposition profile), the trajectory (e.g. human vs. robot), and the optimization strategy (e.g. conservative vs. normal mode). We refer the reader to our supplemental webpage for more examples.

5. Discussion and future work

Analogous to the "sculpting by numbers" approach of Rivers et al. [11], we do not aim to train the user to become a skilled,

Table 1: Statistics about our datasets and experiments.

Image	#A0 Sheets	Size (m^2)	Painting time (min)	#Color switches
Boxer	2	1.7×1.2	13	4
Toucan	2	1.7×1.2	21	6
Chaplin	2	1.2×1.7	30	12
Snail	12	4.7×2.5	80	26
Clown	4	1.5×1.7	52	21
Chameleon		5.2×3.0	129	7
Frog		3.5×3.3	72	5
Tiger		4.5×3.0	85	5



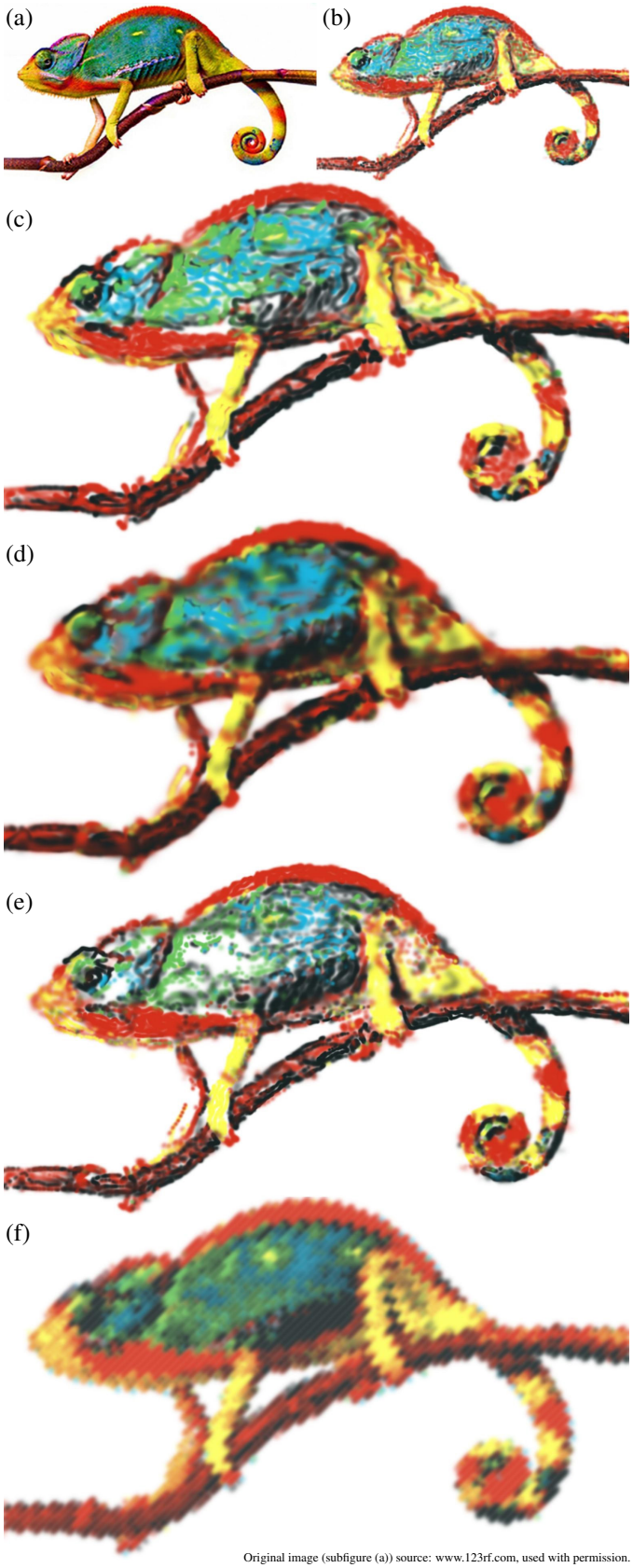
Original image (subfigure (a)) source: Flickr Creative Commons.

Figure 11: The *Clown* is a medium-size painting realized with six colors. Guided by the user, our system is able to recover important semantic features, even at this limited resolution.

unassisted spray painter, nor are we expecting to reach the quality of professional artists. Instead, our system provides the basic technology to spray paint an input image. Without it, a novice would only produce a rough abstraction of the image, especially for the scale we target. However, our current system does not offer a very creative user experience and considerably limits the artistic freedom of the user. As in Shilkrot et al.'s airbrushing system [15], adding an override trigger to our device could improve creativity at the expense of automation. Doing so empowers the user to paint manually while the system only acts as a fail-safe. Our high fidelity online simulation could become the bedrock of creative or educational systems. Our system enables researchers to explore multiple future directions (e.g. automation, user interaction, quality, training) either by completely replacing the user by a robot with full control over the trajectory, or by leveraging the user's creativity with more complex tools such as stencils, or even by training the user by developing a guidance system.

Several improvements could push the quality of our results or practicality of our system further. We would like to optimize the design and performance of our device to allow better and faster control over the spray response. Our system currently works best with high-contrast images that are not dominated by high frequency detail. We rely on our small time intervals for actuation to blend colors in a soft way, but a more direct approach would need to control the pressure precisely. This would require us to build a much more sophisticated – but not necessarily much more expensive – device or switch to a different spraying tool. Another direction could be to design a special cap with aperture control. While these modifications are interesting engineering problems and could significantly aid in reconstructing smooth gradients and high frequencies, their integration in our current optimization would be relatively straightforward. Moreover, our simulation and optimization run mostly on the GPU and perform one order of magnitude faster than the servomotor actuation rate. Therefore, using a higher quality servomotor would directly impact the quality of the paintings.

We used an affordable and efficient tracking solution to proto-



Original image (subfigure (a)) source: www.123rf.com, used with permission.

Figure 12: (a) Target, (b) Result made by the user with our physical setup, (c) Optimization/triggering time $\Delta t = 2sec$ with the same setup, (d) Different nozzle with larger deposition radius r and rate β and the normal strategy, (e) Different nozzle with small dispersion radius r and the conservative strategy, (f) Generated diagonal scanline trajectory with optimal strategy

type our painting setup, and custom-made state-of-the-art tracking is out of the scope of this project. Nevertheless, we believe than one direction for improving the quality of the paintings lies in using more sophisticated tracking technology and advanced tracking algorithms. When relying solely on a vision-based system, the lighting conditions can become an issue. It would also be beneficial to make the setup even more lightweight, eliminating the external cameras. A vision-based tracker fully embedded on the spray can, while appealing, seems extremely challenging due to restricted view, airborne paint particles, lack of suitable features to track, etc. It would be interesting to explore the use of other types of sensors (e.g. inertial or time-of-flight).

Our feedback display already serves as a good visual aid to guide the user, but an interesting avenue for future work would be to investigate other interfaces to train inexperienced users. We considered other options, such as projecting information directly onto the wall, into virtual reality glasses, or onto a phone/tablet attached to the spray can.

Finally, even though we are able to produce color paintings, we believe more investigation of the suitable color spaces and color mixing models would prove useful. Moreover, in our model, colors can mix, but we believe that some features can become incredibly challenging to reproduce if the choice of initial colors was inadequate. An interesting (though challenging) idea for future work would be to suggest colors to buy, given an input image and a paint catalog.

6. Conclusion

We presented an interactive system and an online spray painting simulation algorithm, enabling novice users to paint large-scale murals of arbitrary input photographs. Our system aids the user in tasks that are difficult for humans, especially when lacking artistic training and experience: it automatically tracks the position of the spray can relative to the mural and makes decisions regarding the amount of paint to spray, based on an online simulation of the spraying process. We devise a lightweight calibration method and a fast spraying simulation resulting in close matching between our simulation and the murals. We presented a number of physically-realized and simulated murals that demonstrate the flexibility of our system. We hope that this work will inspire further interactive, creative applications of computer graphics in physical environments.

Acknowledgments

The authors would like to thank Gilles Caprari for his help in developing the prototype version of the device, Maurizio Nitti for the concept art he created, and the Computer Science department of ETH Zurich for lending us a painting workspace. We also thank our colleagues from DRZ, IGL and CGL for insightful discussions and early user testing.

References

[1] Ganz N. Graffiti World: Street Art from Five Continents. Harry N. Abrams; 2009.

[2] Deussen O, Lindemeier T, Pirk S, Tautzenberger M. Feedback-guided stroke placement for a painting machine. In: Proc. CAe. 2012, p. 25–33.

[3] Lindemeier T, Pirk S, Deussen O. Image stylization with a painting machine using semantic hints. Computers & Graphics 2013;37(5):293–301.

[4] Yao F, Shao G. Painting brush control techniques in chinese painting robot. In: Proc. IEEE International Workshop on Robot and Human Interactive Communication. 2005, p. 462–7.

[5] Tresset PA, Leymarie FF. Sketches by Paul the robot. In: Proc. CAe. ISBN 978-1-4503-1584-5; 2012, p. 17–24.

[6] Lehni U, Hektor . In a beautiful place out in the country. In: Wenn Roboter Zeichnen. Kunstmuseum Solothurn; 2004,.

[7] Flagg M, Rehg JM. Projector-guided painting. In: Proc. UIST. ISBN 1-59593-313-1; 2006, p. 235–44.

[8] Laviole J, Hachet M. Spatial augmented reality to enhance physical artistic creation. In: Adjunct Proc. UIST. ISBN 978-1-4503-1582-1; 2012, p. 43–6.

[9] Iarussi E, Bousseau A, Tsandilas T. The drawing assistant: Automated drawing guidance and feedback from photographs. In: Proc. UIST. ISBN 978-1-4503-2268-3; 2013, p. 183–92.

[10] Rivers A, Moyer IE, Durand F. Position-correcting tools for 2d digital fabrication. ACM Trans Graph 2012;31(4).

[11] Rivers A, Adams A, Durand F. Sculpting by numbers. ACM Trans Graph 2012;31(6).

[12] Zoran A, Shilkrot R, Paradiso J. Human-computer interaction for hybrid carving. In: Proc. UIST. ISBN 978-1-4503-2268-3; 2013, p. 433–40.

[13] Yoshida H, Igarashi T, Obuchi Y, Takami Y, Sato J, Araki M, et al. Architecture-scale human-assisted additive manufacturing. ACM Trans Graph 2015;34(4).

[14] Shilkrot R, Maes P, Zoran A. Physical painting with a digital airbrush. In: SIGGRAPH Emerging Technologies. 2014,.

[15] Shilkrot R, Maes P, Paradiso JA, Zoran A. Augmented airbrush for computer aided painting (cap). ACM Trans Graph 2015;34(2).

[16] Haeberli P. Paint by numbers: Abstract image representations. In: Proc. ACM SIGGRAPH. 1990, p. 207–14.

[17] Hertzmann A. Painterly rendering with curved brush strokes of multiple sizes. In: Proc. ACM SIGGRAPH. 1998, p. 453–60.

[18] Zeng K, Zhao M, Xiong C, Zhu SC. From image parsing to painterly rendering. ACM Trans Graph 2009;29(1).

[19] Baxter B, Scheib V, Lin MC, Manocha D. DAB: Interactive haptic painting with 3D virtual brushes. In: Proc. ACM SIGGRAPH. 2001, p. 461–8.

[20] Baxter W, Wendt J, Lin MC. IMPaSTo: A realistic, interactive model for paint. In: Proc. NPAR. 2004, p. 45–148.

[21] Chu N, Baxter W, Wei LY, Govindaraju N. Detail-preserving paint modeling for 3D brushes. In: Proc. NPAR. 2010, p. 27–34.

[22] Lu J, Barnes C, DiVerdi S, Finkelstein A. Realbrush: Painting with examples of physical media. ACM Trans Graph 2013;32(4).

[23] Curtis CJ, Anderson SE, Seims JE, Fleischer KW, Salesin DH. Computer-generated watercolor. In: Proc. ACM SIGGRAPH. 1997, p. 421–30.

[24] Luft T, Deussen O. Real-time watercolor for animation. J Comput Sci Technol 2006;21(2):159–65.

[25] Lee S, Olsen SC, Gooch B. Interactive 3D fluid jet painting. In: Proc. NPAR. 2006,.

[26] Lei SIE, Chen YC, Chen HT, Chang CF. Interactive physics-based ink splattering art creation. Comput Graph Forum 2013;32(7):147–56.

[27] Konieczny J, Meyer G. Airbrush simulation for artwork and computer modeling. In: Proc. NPAR. 2009, p. 61–9.

[28] Marner MR. Digital airbrushing with spatial augmented reality. In: Proc. ICAT, Demos. 2010,.

[29] Porter T, Duff T. Compositing digital images. In: Proc. ACM SIGGRAPH. 1984, p. 253–9.

[30] Hertzmann A. Tutorial: A survey of stroke-based rendering. IEEE Comput Graph Appl 2003;23(4):70–81.

[31] DeCarlo D, Santella A. Stylization and abstraction of photographs. ACM Trans Graph 2002;21(3).

[32] Bronson J, Rheingans P, Olano M. Semi-automatic stencil creation through error minimization. In: Proc. NPAR. 2008, p. 31–7.

[33] Hausner A. Simulating decorative mosaics. In: Proc. ACM SIGGRAPH. 2001, p. 573–80.

[34] Kim J, Pellacini F. Jigsaw image mosaics. ACM Trans Graph 2002;21(3):657–64.

[35] Ye Q, Domnick J, Scheibe A, Pulli K. Numerical simulation of electrostatic spray-painting processes in the automotive industry. In: Proc. HPCSE. 2004, p. 261–75.

[36] Ye Q. Using dynamic mesh models to simulate electrostatic spray-painting. In: Proc. HPCSE. 2005, p. 173–83.

[37] Antonio JK. Optimal trajectory planning for spray coating. In: Proc. Robotics and Automation; vol. 3. 1994, p. 2570–7.

[38] Chen H, Fuhlbrigge TA, Li X. A review of CAD-based robot path planning for spray painting. Industrial Robot 2009;36(1):45–50.

[39] Konieczny J. Application, rendering and display of automotive paint. Ph.D. thesis; University of Minnesota; 2009.

[40] Conner DC, Greenfield AL, Atkar P, Rizzi A, Choset H. Paint deposition modeling for trajectory planning on automotive surfaces. IEEE Trans Automation Science and Engineering 2005;2(4):381–92.