

Article

Physically Motivated Model of a Painting Brush for Robotic Painting and Calligraphy

Artur Karimov ^{1,*}, Maksim Strelnikov ¹, Sergei Mazin ¹, Dmitriy Goryunov ¹, Sergey Leonov ²
and Denis Butusov ^{3,*}

¹ Youth Research Institute, Saint Petersburg Electrotechnical University “LETI”, Saint Petersburg 197022, Russia; mdstrelnikov@etu.ru (M.S.); svmazin@etu.ru (S.M.); dsgoriunov@etu.ru (D.G.)

² Public Relations Department, Saint Petersburg Electrotechnical University “LETI”, Saint Petersburg 197022, Russia; svleonov@etu.ru

³ Computer-Aided Design Department, Saint Petersburg Electrotechnical University “LETI”, Saint Petersburg 197022, Russia

* Correspondence: aikarimov@etu.ru (A.K.); dnbutusov@etu.ru (D.B.)

Abstract: Robot artistic painting and robot calligraphy do require brush models for brushstroke simulation and painting robot control. One of the main features of the brush is its compliance, which describes the relationship between the brush footprint shape and the pressure applied to the brush. In addition, during motion, the brush footprint position lags from the brush handle position in a complicated manner. To date, the question of creating a physically correct model of these effects and choosing the best method for the model parameter calibration has not been presented in the literature. In the current paper, we derive equations of the brush contact patch motion, give their closed-form solutions, and investigate three methods for the brush model calibration: capturing brush footprints on a matte glass with a camera, painting calibration brushstrokes, and capturing a brush shape side projection with a camera. As we show, calibration brushstrokes give us primary information on brush contact patch displacement during painting, and capturing the brush side projection allows the accurate estimation of the gap from the brush tip to the center of the contact patch. Capturing brush footprints is useful for creating a brushstroke executable model. As an example, a model for a round artistic brush was created and verified in three tests, including measuring the coordinates of an angular brushstroke center line, simulating an angular brushstroke, and writing a signature using a robotic setup.

Keywords: robotic painting; computer creativity; painting brush; robot calligraphy; brush simulation; physical brush model



Citation: Karimov, A.; Strelnikov, M.; Mazin, S.; Goryunov, D.; Leonov, S.; Butusov, D. Physically Motivated Model of a Painting Brush for Robotic Painting and Calligraphy. *Robotics* **2024**, *13*, 94. <https://doi.org/10.3390/robotics13060094>

Academic Editor: Dan Zhang

Received: 10 May 2024

Revised: 8 June 2024

Accepted: 18 June 2024

Published: 20 June 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Computer creativity is a growing topic of research, which involves several interdisciplinary investigations, including studies in computer science, philosophy, music, and fine arts [1]. The success of artificial intelligence (AI) in visual arts became exceptionally prominent due to commercially available examples of AI-powered programs capable of creating realistic images [2]. In fields of visual art such as Chinese calligraphy, AI successfully passes an artistic version of the Turing test [3]. In addition to solving the challenges of generating artistic-looking images, some efforts are aimed at adapting the performance of traditional physical artistic techniques to robotic systems, which is supported by the rapid development of the related fields in robotics, such as sensor design for collision detection [4], positioning using data fusion [5], target location, and active maneuvering in an unknown environment [6]. These advances resulted in new industrial standards such as collaborative robots and new software libraries for solving different problems faced by robots in the real world. General advances in machine vision, distance and contact

sensors, etc. provide a solid basis for specific branches of robotics such as artistically skilled robots. Various tasks are faced by developers of the latter systems from modifying paints and tools for better usability by robots to creating new mathematical models and control algorithms. During the last decade, a significant amount of studies have been dedicated to solving these problems, including creating machines for acrylic painting by brush and palette knife [7–12], watercolor brush painting [13], spray painting [14,15], etc.

The majority of artistic robots use a round brush as a painting tool. The artistic brush consists of bristles clamped to a handle with a metal ferrule. Bristles comprise a flexible tuft which allows collecting a certain amount of paint and softly applying it to a canvas. The round brush is the most versatile variant of a brush and is the simplest to use in artistic robotic painting because of its symmetry in any direction. The width of the brushstroke painted by a round brush depends on the pressure applied to a brush, which in turn depends on the distance of the robot brush-holding mechanism to the canvas. The compliance of brush bristles causes some problems in the robot path planning, since the brush contact patch lags from the robot tool center position (TCP) during its motion. This produces a disparity between the TCP trajectory and the actual shape of the brushstroke. The precise robot control system should include a finely tuned brush model reproducing these features of the brush with a high level of fidelity.

Some works on artistic robots propose using special short-bristle brush designs with no need for elaborate modeling and calibration [7,16], but their expression capabilities are much lower than those obtained by a common brush. The other works propose different approaches to taking brush compliance into account. Early attempts to create artistic brush models refer to the 1990s, among which an example of a physically-based model can be found in a work [17] considering compliance of the brush bristles and its influence on the brushstroke shape. Later, a model of a conic Chinese brush with a straight center line was described in a work [18]. A more complicated 3D model of a brush exploiting the energy minimization principle in its shape adjustment was described in another work [19].

The previously mentioned works were not aimed at modeling a particular real brush. The situation changed when interest in robotic painting and calligraphy led to the need for more accurate brush simulation [20].

One of the early examples of brush calibration for a calligraphy robot was given in [21]. The more complicated approach to brush modeling was proposed in [22], which utilizes a regressing model of a brush lag depending on the brushstroke width and the trajectory curvature estimated in a set of experiments. Another empirically correct model was proposed in [23], where difference equations of the brush dynamics were introduced, the coefficients of which were found on a calibration phase. Nevertheless, only a few words were said about the method of brush calibration and its theoretical justification. A notable question for creating a data-driven brush footprint model was considered in a work [24], and a feedback-based approach based on an internal model control for a calligraphy robot was described in [25]. One of the most detailed models of a brush was described in [26]. The brush footprint trajectory in this work was modeled by a simple difference equation, but no closed-form solutions for the brush footprint coordinates were given.

Even though the majority of artistically skilled robots use a brush as a painting tool, and many works are dedicated to this topic, the question of creating a physically correct brush model and developing a simple and robust method for adjusting its parameters is still open. Forces acting at a brush contact patch were still not considered, nor was there any theoretical description of a brushstroke curvature induced only by bristles compliance. Moreover, there is still a question regarding how to calibrate a brush, since each research proposes different experiments requiring various equipment and different measurements.

So, the current work fills this gap. We propose a new physically motivated model of a brush contact patch movement relative to a brush handle trajectory. Then, we prove several theorems on a brush contact patch trajectory. After that, we propose an experimental setup and a detailed description of a calibration method allowing us to obtain an empirically correct model for use in robotic painting and calligraphy systems. The obtained model

is verified in three tests, including measuring the coordinates of an angular brushstroke center line, simulating an angular brushstroke, and writing a signature using a robot.

The significance of the correct brush model in robotic visual art is undoubted, since it allows solving the robot’s path-planning problem and implementing brushstrokes properly as they were designed by simulation, thus avoiding the number of failures in art processes which does not assume correction such as calligraphy or watercolor, and decreasing the number of iterations in a feedback-guided painting process with acrylic or oil paints.

The paper is organized as follows. In Section 2, an introduction to the physics of a painting brush is given. Equations of motion of a brush contact patch and a brush tip are given in the form of differential equations and their closed-form solutions. All results are presented in theorems, and their corollaries are supplied with proofs. In Section 3, the results are presented. Three calibration methods were investigated, resulting in two models with different reliability. Section 4 summarizes the recommended brush model calibration method. In Section 5, brief conclusions are given.

2. Materials and Methods

Let us describe the painting brush by introducing the following formalism. Suppose that a brush is mounted on a robot and always moves orthogonal to a canvas plane. Suppose also that the brush stiffness is high relative to the friction force between the brush and the canvas, which is often true for synthetic and bristle brushes used in acrylic and oil painting. Therefore, each bristle can be modeled as an elastic beam with clearly defined boundary conditions. The following lemma ensures that under these assumptions, the solution of the bristle equation solution is unique.

Lemma 1. *Let the bristle be modeled as the static Euler–Bernoulli beam*

$$\frac{\partial^2}{\partial z^2} \left(EI(z) \frac{\partial^2 w(z)}{\partial z^2} \right) = q(z), \tag{1}$$

where $w(z)$ is the bristle deflection in the direction of the brush longitudinal direction x , $EI(z)$ is the bristle flexural rigidity, and $q(z)$ is a distributed load. The bristle Equation (1) has the unique solution $w(z)$.

Proof. Let us determine four boundary conditions for Equation (1). Considering the normalized variable $z \in [0, 1]$ with no loss of generality, the boundary conditions at the bristle root (the beam clamped end) are $w(0) = 0$, and $\partial w(0)/\partial z = 0$. The bristle tip is a sliding beam end with the corresponding initial conditions $\partial w(1)/\partial z = 0$, $\partial^3 w(1)/\partial z^3 = 0$. Figure 1 illustrates our proposals. The existence and uniqueness of the generalized solution of the Euler–Bernoulli beam once four boundary conditions are known was proven in a work [27], and special cases of unique solutions are given in several papers, e.g., [28,29]. □

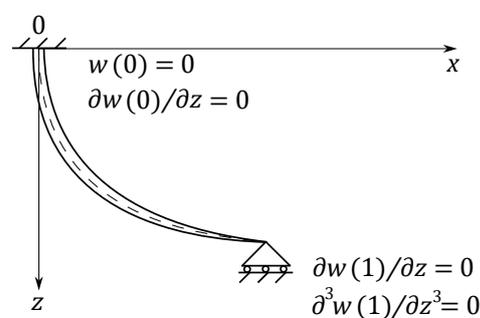


Figure 1. Boundary conditions for the Euler–Bernoulli model for the brush bristle.

Let us consider an example. The paper [30] proposes exponential decay in the elastic modulus of the brush, leading to a complicated linear–exponential solution for

$EI(z)$. Simplifying equations for the flexural rigidity, we adopt $EI(z) = \exp(1 - z)$ and $q(z) = \exp(1 - z)$. The Equation (1) would read as shown below:

$$\frac{\partial^2}{\partial z^2} \left(e^{1-z} \frac{\partial^2 w(z)}{\partial z^2} \right) = e^{1-z}, \tag{2}$$

which has the following solution:

$$w(z) = c_4 e^z z^3 + c_3 e^z z^2 + c_2 e^z z + c_1 e^z + 1. \tag{3}$$

where $c_i, i \in [1 \dots 4]$ are constants determined by boundary conditions.

Let the robot move within relatively slow speeds, such as 25 mm/s, so the bristle inertia does not affect the brush contact patch shape, and bristles are always bent properly, as shown in Figure 2a. First, we determined the shape of the brush contact patch as the shape of the envelope of all fragments of bristles contacting with a canvas. This envelope is defined exactly by the longitudinal and transverse positions of bristles. From Lemma 1, we know that the longitudinal position of a single bristle is uniquely defined by z . By the energy minimization principle allowing only one optimal relative position of the bristles in a bunch [19], the longitudinal and transverse positions of the bristles would also be uniquely defined. During brush turns, different bristles acquire and lose contact with the canvas, but if the round brush is not worn, the envelope will preserve its shape for the given value of z due to the rotational symmetry of the brush. A short video demonstrating the effect of the brush contact patch shape immutability during rotation can be found at the link [31]. For a particular value of z , the dynamic of a contact patch is defined by a lag r between a center of forces applied to a contact patch and a projection of a brush handle on a canvas plane. To locate the contact patch position relative to the center of forces, a gap b from the brush tip to the center of the forces should be also known; see Figure 2a.

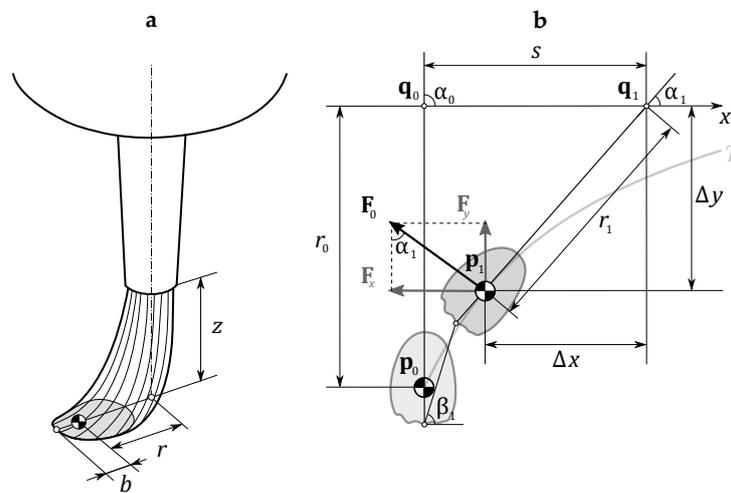


Figure 2. (a) Normal geometry of a brush. (b) Dynamics of a brush contact patch movement.

2.1. Equation of Brush Movement with Normal Geometry

Denote α an angle of the brush foot median line relative to the direction of the robot movement and the lag r between a center of forces applied to a brush contact patch and a brush handle axis. In Figure 2b, schematic drawings are given explaining the geometry of a brush foot movement. Let us consider a general case: when z changes during movement, r also changes and should be considered as a function of s .

Theorem 1. *The equations of motion for the brush center of forces are shown below:*

$$\frac{d\alpha}{ds} = -\frac{\sin \alpha}{r(s)} \quad (4)$$

$$\Delta x = -r(s) \cos \alpha \quad (5)$$

$$\Delta y = -r(s) \sin \alpha, \quad (6)$$

where α is an angle between vector pointing to the center of forces $\mathbf{p} = (\Delta x, \Delta y)^\top$ and the direction of the brush root movement $\mathbf{v} = \mathbf{q}_1 - \mathbf{q}_0$, $s = \|\mathbf{v}\|$ is the brush root displacement, and r is the lag of the brush contact patch depending on s . Equation (4) has the following closed-form solution:

$$\alpha(s) = 2 \arctan \left(\tan \left(\frac{\alpha_0}{2} \right) \exp \left(- \int_0^s \frac{1}{r(\xi)} d\xi \right) \right). \quad (7)$$

Proof of Theorem 1. Let us consider a local coordinate system, in which a projection of a brush handle onto the canvas plane moves along the x -axis. A brush contact patch is pulled by a leverage of length r from the handle axis to the center of a friction force \mathbf{F}_0 . The center of the force follows a trajectory T . The friction force is split into projections on coordinate axes \mathbf{F}_x and \mathbf{F}_y . When the robot TCP moves along the x -axis and travels a distance s , an angle between the brush footprint centerline and the x -axis changes from α_0 to α_1 . The forces make the following work:

$$-\|\mathbf{F}_x\|(\Delta x_1 - \Delta x_0) + \|\mathbf{F}_y\|(\Delta y_1 - \Delta y_0) = A, \quad (8)$$

where Δx and Δy are displacements between the center of forces \mathbf{p}_1 and the robot TCP projection onto the canvas \mathbf{q}_1 , which are calculated from trigonometrical considerations as

$$\Delta x = -r(s) \cos \alpha$$

and

$$\Delta y = -r(s) \sin \alpha.$$

The work of the force F_x is negative, since the direction of movement is opposite to its direction. Suppose that the sum is zero considering that no energy is obtained or lost, i.e., $A = 0$. From Figure 2b, one may see that

$$\begin{aligned} F_x &= F_0 \sin \alpha_1, \\ F_y &= F_0 \cos \alpha_1, \\ \Delta x_1 - \Delta x_0 &= s + r_0 \cos \alpha_0 - r_1 \cos \alpha_1, \\ \Delta y_1 - \Delta y_0 &= r_0 \sin \alpha_0 - r_1 \sin \alpha_1, \end{aligned} \quad (9)$$

where s in (9) is the robot TCP displacement along the x -axis. Therefore, the force in Equation (8) can be rewritten as follows:

$$-F_0 \sin \alpha_1 (s + r_0 \cos \alpha_0 - r_1 \cos \alpha_1) + F_0 \cos \alpha_1 (r_0 \sin \alpha_0 - r_1 \sin \alpha_1) = 0.$$

After division by F_0 , it can be simplified to

$$-\sin \alpha_1 s + r_0 \sin(\alpha_1 - \alpha_0) = 0.$$

Considering an infinitesimally small translation ds , writing this equation in terms of differentials, and taking into account that $\sin d\alpha = d\alpha$, the equation reads as shown below:

$$-\sin \alpha ds - r(s) d\alpha = 0,$$

where, eventually, we obtain a differential equation describing the behavior of the angle α as follows:

$$\frac{d\alpha}{ds} = -\frac{\sin \alpha}{r(s)}. \tag{10}$$

Consider the solution of (10). Let us move the terms of the equation between its sides so that there are terms with α on the left and terms with s on the right and integrate the following:

$$\int_{\alpha_0}^{\alpha_1} \frac{1}{\sin \alpha} d\alpha = - \int_0^{s_1} \frac{1}{r(s)} ds.$$

The left side of the equation has the exact solution

$$\log\left(\tan\left(\frac{\alpha}{2}\right)\right)\Big|_{\alpha_0}^{\alpha_1} = - \int_0^{s_1} \frac{1}{r(s)} ds,$$

while the right side of the equation depends on the function $r(s)$. Let us introduce a new variable ξ in the right side of the equation and change α_1 to simply α to exclude subscripts:

$$\log\left(\tan\left(\frac{\alpha}{2}\right)\right) - \log\left(\tan\left(\frac{\alpha_0}{2}\right)\right) = - \int_0^s \frac{1}{r(\xi)} d\xi,$$

which immediately leads to (7). \square

The equations of motion for the brush represent a system of differential–algebraic equations, and once an initial angle α_0 and initial and final positions of the robot \mathbf{q}_0 and \mathbf{q}_1 are known, it is easy to calculate the robot position in a global coordinate system. Let the translation vector $\mathbf{v} = \mathbf{q}_1 - \mathbf{q}_0$ have coordinates $(x_v, y_v)^\top$. Then, the angle of rotation for translation between the local and global coordinate systems is determined by the pair

$$\begin{aligned} \cos \gamma &= \frac{x_v}{\|\mathbf{v}\|}, \\ \sin \gamma &= -\frac{y_v}{\|\mathbf{v}\|}, \end{aligned}$$

from where the coordinates of a new position for the center of forces \mathbf{p}_1 are found as

$$\begin{aligned} \Delta \mathbf{p} &= \begin{pmatrix} \cos \gamma & \sin \gamma \\ -\sin \gamma & \cos \gamma \end{pmatrix} \begin{pmatrix} \Delta x \\ \text{sign}(\mathbf{w}) \Delta y \end{pmatrix}, \\ \mathbf{p}_1 &= \mathbf{p}_0 + \Delta \mathbf{p}, \end{aligned}$$

where \mathbf{w} is a result of a cross-product $\mathbf{w} = (\mathbf{q}_0 - \mathbf{p}_0) \times \mathbf{v}$. This term is needed to verify the actual brush direction of the turn. If the brush moves from bottom to top as shown in Figure 2b, then $\text{sign}(\mathbf{w}) = 1$, and if the brush moves from top to bottom in a mirrored manner with respect to Figure 2b, $\text{sign}(\mathbf{w}) = -1$.

The next theorem is dedicated to the equation for the angle β , shown in Figure 2b, in a special case $r = \text{const}$.

Theorem 2. *The angle β between the direction of the robot TCP movement and a line of the brush tip displacement depends on α, r, b and the value of displacement s_1 as follows:*

$$\beta_1 = \text{arccot} \frac{s_1 - \frac{r+b}{r} \int_0^{s_1} \sin^2 \alpha ds}{\frac{r+b}{r} \int_0^{s_1} \sin \alpha \cos \alpha ds}. \tag{11}$$

Proof of Theorem 2. The brush tip located at a distance b from the center of force along the brush foot centerline changes its coordinates as follows:

$$\begin{cases} \Delta x_b = s + (r + b)(\cos \alpha_0 - \cos \alpha_1), \\ \Delta y_b = (r + b)(\sin \alpha_0 - \sin \alpha_1). \end{cases} \quad (12)$$

For simpler notation, denote $d = r + b$. In terms of differentials, (12) reads as

$$\begin{aligned} dx_b &= ds + d(\cos \alpha - \cos(\alpha + d\alpha)), \\ dy_b &= d(\sin \alpha - \sin(\alpha + d\alpha)). \end{aligned}$$

Applying again the Taylor expansion for the trigonometric functions, the latter equation simplifies to

$$dx_b = ds + d \sin \alpha d\alpha, \quad (13)$$

$$dy_b = -d \cos \alpha d\alpha. \quad (14)$$

The actual coordinates of the brush tip may be found via integration:

$$\begin{aligned} x_b &= x_{b0} + \int_0^{s_1} \left(1 + d \sin \alpha \frac{d\alpha}{ds} \right) ds, \\ y_b &= y_{b0} - \int_0^{s_1} d \cos \alpha \frac{d\alpha}{ds} ds, \end{aligned}$$

where x_{b0}, y_{b0} are the coordinates of the initial brush tip position. Substituting the value of $d\alpha/ds$ from (10), we finally obtain

$$\begin{aligned} x_b &= s_1 - \frac{d}{r} \int_0^{s_1} \sin^2 \alpha ds, \\ y_b &= \frac{d}{r} \int_0^{s_1} \sin \alpha \cos \alpha ds. \end{aligned}$$

The angle β between the x -axis and a line connecting an initial brush tip position with the actual brush tip position is determined by the following equation:

$$\cot \beta_1 = \frac{x_b - x_{b0}}{y_b - y_{b0}}.$$

Substituting (13) and (14) into the latter equation, we obtain the following:

$$\cot \beta_1 = \frac{s_1 - \frac{r+b}{r} \int_0^{s_1} \sin^2 \alpha ds}{\frac{r+b}{r} \int_0^{s_1} \sin \alpha \cos \alpha ds},$$

which immediately leads to (11). \square

Figure 3 illustrates the solution of the brush dynamic Equation (7) for α , (5) and (6) for the brush center of forces, and

$$\Delta x_b = -(r(s) + b(s)) \cos \alpha, \quad (15)$$

$$\Delta y_b = -(r(s) + b(s)) \sin \alpha \quad (16)$$

for the brush tip. The parameter values are $r = 3, b = 1$ and the initial values $x_0 = 0, y_0 = 0$, and $\alpha_0 = \pi/2$. The red line in Figure 3a is the trajectory of the brushstroke center line, the blue line is the trajectory of a brush tip, and a thick black line shows the trajectory of the robot TCP.

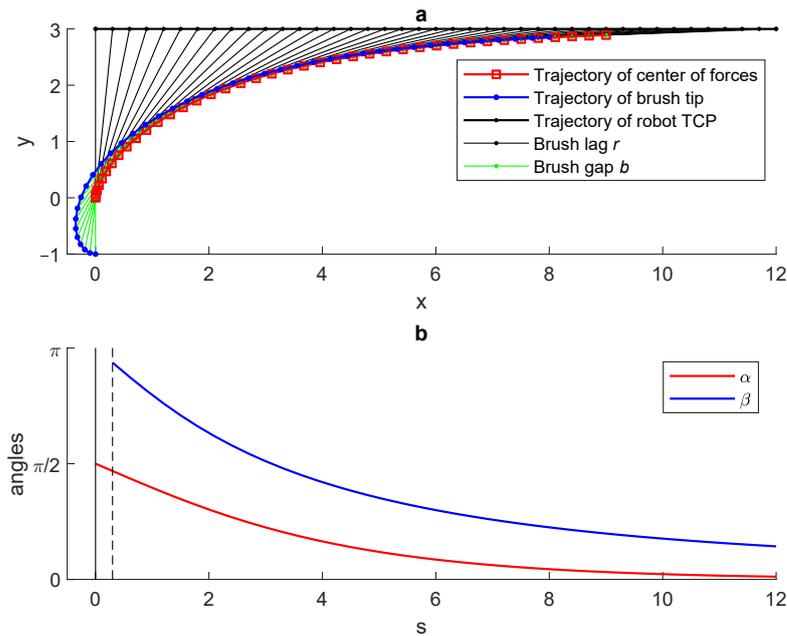


Figure 3. Solution of the brush curve dynamics. (a) Coordinates of the brush center of forces, brush tip, and the TPC of the robot. (b) Angles α and β .

In Figure 3b, the line for β starts from the value denoted by a vertical dashed line located at $s_0 = \Delta s$, where $\Delta s = 0.3$ is the discretization step.

2.2. Brush Movement with Loose Tension

If the brush changes its direction of movement by the angle greater than $\pi/2$, or when it touches the canvas with a too steep angle, a situation depicted in Figure 4a occurs: the actual distance between the center of forces of the brush contact patch to the brush axis r' is less than the radius r inherent to the brush under the normal conditions for a particular value of z . Practically, this means that the bristle bunch loses tension and bends due to compressive forces. Nevertheless, due to the uniqueness of the Euler–Bernoulli equation solution and energy minimization principle, the brush would still possess a uniquely defined geometry, and the brush tip would still be pointing in the direction opposite to the brush root exactly as shown in Figure 4a.

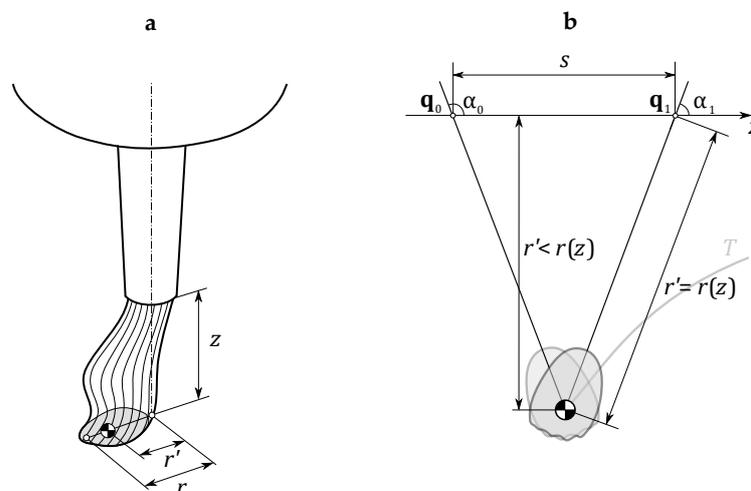


Figure 4. (a) Geometry of a brush with an improper type of bristle bend. (b) Dynamics of a brush contact patch movement in case $\alpha_0 > \pi/2$.

This allows for simple treatment of this case. After calculating the distance between the new robot TCP and the brush contact patch center of reassurance r' , one should compare it with $r(z)$ inherent to the brush under normal conditions, as shown in Figure 4b. If $r' \geq r(z)$, then the model (4) is valid, and new α , x and y , as well as x_b and y_b , are calculated as described earlier. Otherwise, x and y remain the same, and a new α is chosen so that it is the angle of a straight line connecting the TCP and the center of forces of the brush contact patch. The brush contact patch is then rotated according to this angle, and new x_b and y_b values are calculated.

2.3. Executable Model of a Painting Brush

A realistic brush model calibrated as described before can be used in a variety of scenarios. One of the simplest is simulating the brushstroke behavior for a given brush root trajectory. Let the trajectory consist of straight segments. The current segment connects two points \mathbf{p}_0 and \mathbf{p}_1 , where $\mathbf{p} = (x, y, z)^\top$. The previous segment is denoted as \mathbf{d}_0 , and the current segment is denoted as $\mathbf{d}_1 = \mathbf{p}_1 - \mathbf{p}_0$. The angle between these two segments can be determined from a formula for the scalar product:

$$\alpha_0 = \arccos \frac{\mathbf{d}_0 \mathbf{d}_1}{\|\mathbf{d}_0\| \|\mathbf{d}_1\|}.$$

For the first segment, $\alpha_0 = 0$. The actual angle of rotation of a brush footprint with respect to the canvas coordinate system θ is calculated by

$$\theta = \alpha + \text{atan2}(y_1 - y_0, x_1 - x_0),$$

where atan2 is a function that returns an angle of a planar vector in a full range $[-\pi, \pi]$. From a brush footprint library, two images of footprints I_1 and I_2 closest to the z values in \mathbf{p}_0 and \mathbf{p}_1 are chosen.

The segment \mathbf{d}_1 is parameterized by a variable $t \in [0, 1]$. In each point $\mathbf{p}_d(t)$ of the segment, $z(t)$ is calculated as a linear interpolation, from which $r(z)$ and $b(z)$ are estimated. A distance r' between the current point \mathbf{p} and $\mathbf{p}_d(t)$ is compared to $r(z)$, and if $r' < r(z) - \varepsilon$, where ε is a small regularization value, typically, $\varepsilon = 0.01$, then the new angle θ is calculated as the angle of a straight line between \mathbf{p} and $\mathbf{p}_d(t)$. The values $x_b(s(t))$ and $y_b(s(t))$ are found by simply rotating by the angle θ around the point on a plane with coordinates x_0 and y_0 . Otherwise, the angle $\alpha(s(t))$ is calculated by the formula (7), and $x_b(s(t))$ and $y_b(s(t))$ are found in a local coordinate system by the formulas (15) and (16). Then, the coordinates of a brush are translated into a global coordinate system.

A special procedure $I_c = \text{MergeAndRotate}(I_1, I_2, c, \theta)$ was used for merging images of brushstrokes in a proportion c and rotating it by the angle θ , where z is an actual z -position of the robot TCP. For the normal brush geometry, $c = (z - z_0)/(z_1 - z_0)$, and I_1 and I_2 are brush footprints corresponding to TCP heights z_0 and z_1 , but if $r' < r(z) - \varepsilon$, then $c = \sqrt{(z - z_0)^2 + r'^2} / \sqrt{(z_1 - z_0)^2 + r^2(z)}$, and I_1 is the brush footprint obtained when the brush touches the canvas vertically for the particular value of z . A synthetic image of the current shape of a brush footprint I_c is depicted on a canvas. As a result, a whole segment of a brushstroke appears. Figure 5 illustrates the flowchart of the described process.

In a feedback-guided painting process with a robot, intermediate steps of path planning and trajectory optimization can be added [10,16].

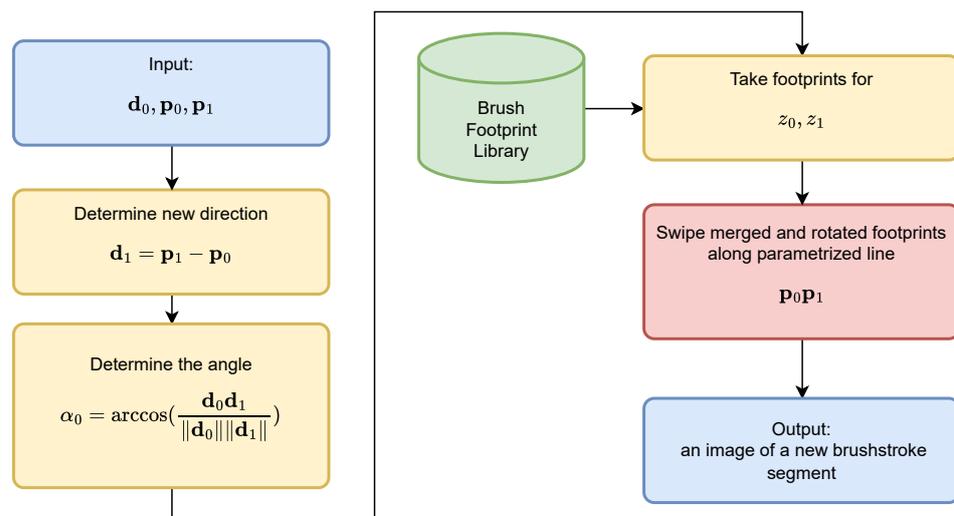


Figure 5. Schematic representation of an algorithm for the brushstroke simulation.

2.4. Finding Brush Parameters from Calibration Brushstrokes

One of the most versatile methods for brush parameter estimation is painting a set of calibration brushstrokes of different shapes. Some examples of methods for designing calibration brushstroke sets can be found in the literature [10,22,32].

For different values of the brush root height above the canvas z , we propose painting brushstrokes of two types. The first type is a straight brushstroke, and the second type is an angular brushstroke employing a situation when the robot TCP changes its direction of movement by 90 degrees. Examples of such strokes are depicted in Figure 6.

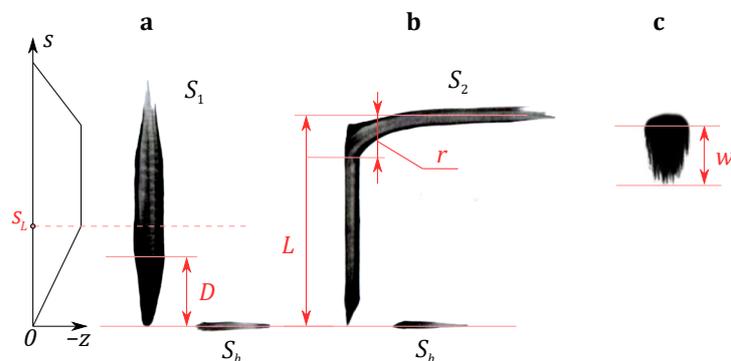


Figure 6. Examples of calibration strokes. (a) Straight stroke for estimating b . (b) Angular stroke for estimating r . (c) Brush footprint for estimating w .

Let us start from Figure 6b, which shows an angular stroke S_2 . By design, we know a distance L , at which the robot TCP turns by 90 degrees, and therefore, we can visually detect a point where the stroke starts turning and measure a distance between this point and a line of subsequent movement, which equals r . Having measured r at various TCP z values, we can find an approximate function $r = f_r(z)$. An auxiliary horizontal stroke S_h shown in the bottom of the panel is needed to accurately measure L in a real experimental environment, since the point when the brushstroke starts in an unreliable reference: its position is highly affected by a canvas relief, robot calibration, amount of paint remaining at the brush tip, etc.

Figure 6a shows a straight brushstroke S_1 needed to find the parameter b . To the left of the brushstroke, a trajectory of the robot TCP is shown. The point s_L is the point where the robot stops lowering the brush and goes further at one height. Due to the lag between the robot TCP and the brush contact patch, the distance D between the brushstroke starting point and the point where it stops expansion is less than s_L . The exact position of this

point depends on a distance w between the brush footprint tip and the widest point of the footprint; see Figure 6c. One can measure the distance D and estimate the parameter b using Equation (17):

$$b = s_L - D + w. \tag{17}$$

Of course, the parameter w is also a function of z , and this function should be preliminarily estimated. As for the brush footprints, a way to capture them is by taking a photo on the reverse side of a matte glass, as we will describe further.

2.5. Direct Way to Measure $b + r$

A distance d from the brush root to the brush tip can be observed directly. This distance equals the sum of the required brush parameters $d = b + r$ and is useful for estimating b more accurately than using the other ways described before.

A brush slowly lowering while sweeping the canvas can be captured by a video camera, as shown in Figure 7. Providing a scale and correcting possible lens aberration to achieve high linearity of the image, one can obtain a dataset on directly observed values of $b + r$ for different z .

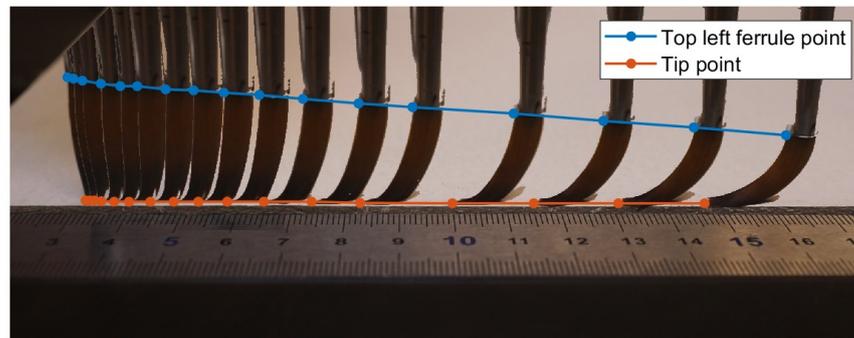


Figure 7. Dataset for estimating d from video footage with adaptive time discretization steps.

In video frames, one should measure the positions of a certain ferrule point $\{x_{fi}\}, \{z_{fi}\}$ and positions of the brush tip $\{x_{ti}\}, \{z_{ti}\}$. Then, from these data, one determines the constants $d_0 = x_{f0} - x_{t0}$, $z_0 = x_{t0} - x_{f0}$, and calibrates the distances d_i and vertical positions z_i as follows:

$$\begin{aligned} d_i &= x_{fi} - x_{ti} - d_0, \\ z_i &= x_{ti} - x_{fi} - z_0. \end{aligned}$$

Once the function $r = f_r(z)$ is known, brush radii are easily found, and a dataset for b is obtained via a simple relation

$$b_i = d_i - f_r(z_i). \tag{18}$$

2.6. Finding Brush Parameters from Angles

Previous subsections propose reliable yet still indirect ways to measure r and b separately. Alas, these parameters cannot be measured directly because the position of the center of forces is unknown, and only moving the brush along a trajectory can reveal the brush dynamics. Nevertheless, angles α and β can be measured, and a series of these measurements may be used for another type of indirect estimation of r and b .

Theorem 3. Suppose we have a set of measurements $\{\alpha_i\}, \{\beta_i\}$ and $\{s_i\}$, $i \in [0 \dots n]$. To find r , introduce a function

$$F(\alpha_0, \alpha_i) = \left(\log \left(\frac{1}{\sin \alpha_0} + \cot \alpha_0 \right) - \log \left(\frac{1}{\sin \alpha_i} + \cot \alpha_i \right) \right). \tag{19}$$

Having computed this function in all pairs (α_0, α_i) , we can find r via least squares:

$$r = -\left((F(\alpha_0, \alpha_1) \dots F(\alpha_0, \alpha_n))^{\top} \right)^+ (s_1 \dots s_n)^{\top}, \tag{20}$$

where $^+$ stands for the pseudo-inverse of a matrix and $^{\top}$ stands for the vector transpose.

Proof. Rewrite (10) as follows:

$$ds = -r \frac{d\alpha}{\sin \alpha}.$$

Integrating this equation with a constant value of r , we obtain the following:

$$s_i = -r \int_{\alpha_0}^{\alpha_i} \frac{d\alpha}{\sin \alpha}.$$

This equation has the following analytic solution:

$$s_i = -r \left(\log \left(\frac{1}{\sin \alpha_0} + \cot \alpha_0 \right) - \log \left(\frac{1}{\sin \alpha_i} + \cot \alpha_i \right) \right).$$

Substituting the function $F(\alpha_0, \alpha_i)$ from (19), we obtain a simple relation

$$s_i = -rF(\alpha_0, \alpha_i),$$

so r is easily found via least squares using Equation (20). \square

Having estimated r , we can now find b .

Theorem 4. Suppose we have a set of measurements $\{\alpha_i\}$, $\{\beta_i\}$ and $\{s_i\}$, $i \in [0 \dots n]$, and r is also already estimated. Denote

$$\begin{cases} I_{1i} &= \int_0^{s_i} \sin^2 \alpha ds, \\ I_{2i} &= \int_0^{s_i} \sin \alpha \cos \alpha ds, \\ \mathcal{I}_i &= I_{1i} + I_{2i} \cot \beta_i. \end{cases} \tag{21}$$

So, we can find an auxiliary variable μ via the LSM:

$$\mu = \left((\mathcal{I}_1 \dots \mathcal{I}_n)^{\top} \right)^+ (s_1 \dots s_n)^{\top}, \tag{22}$$

and eventually estimate b as follows

$$b = (\mu - 1)r. \tag{23}$$

Proof. Once a series of angles $\{\beta_i\}$ and thus a series $\{\cot \beta_i\}$ are available, rewrite Equation (11) in the following manner, taking into account the notation (21):

$$\cot \beta_i = \frac{s_i - \mu I_{1i}}{\mu I_{2i}},$$

where $\mu = (r + b)/r$. Therefore,

$$(I_{1i} + I_{2i} \cot \beta_i)\mu = s_i.$$

Denoting $\mathcal{I}_i = I_{1i} + I_{2i} \cot \beta_i$, the latter is used for estimating μ via the LSM using Formula (22) and then finding b using formula (23). \square

Thus, both unknown parameters r and b can be found from experimental data containing angles $\{\alpha_i\}$, $\{\beta_i\}$ and displacements $\{s_i\}$.

3. Results

In this section, a description of the experimental results and their interpretation is given.

3.1. Evaluating Brush Footprints and Angles

A synthetic artistic brush “Malevich 6” was used for the experimental investigation. The handle of the brush was cut, and the brush was glued to a plastic holder specially designed for a robotic painting.

To create a brush model, an experimental setup was designed, as shown in Figure 8a. It features a Jaka Zu 3 collaborative robot (Jaka Robotics Co., Ltd., Shanghai, China) with a brush attached via a removable 3D-printed adapter. During the experiment, the wet brush touched the semi-matte acrylic screen, making the contact patch of the brush clearly visible on the other side of the screen. The brush contact patch was photographed from the other side using a camera with a macro lens. Our setup used a Canon M200 camera (Canon Inc., Tokyo, Japan) with a Canon 28 mm 3.5 f lens (Canon Inc., Tokyo, Japan). The robot was controlled from a tablet PC with Zaka control software installed. A Wi-Fi connection to a desktop computer via Canon EOS Utility made it possible to remotely capture the brush contact patch without physical contact with the camera. This ensured its immobility during the experiment.

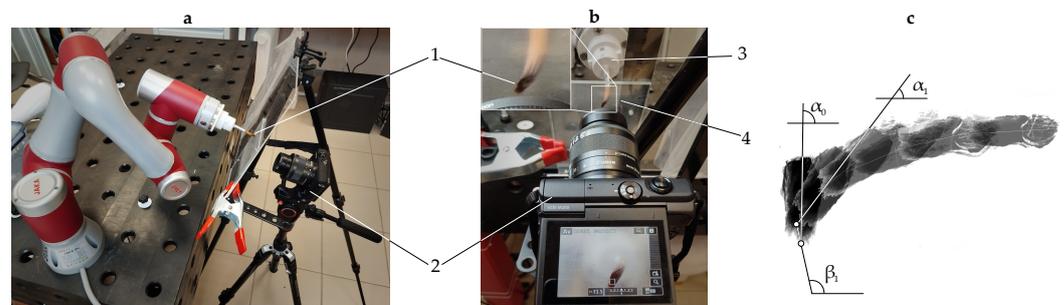


Figure 8. Experimental setup and example of the result: (a,b) Setup, numbers denote the following: 1—brush contact patch, 2—Canon M200 photo camera, equipped with 28 mm macro lens, 3—a brush mount attached to a Jaka Zu 3 collaborative robot, 4—semi-matte acrylic screen. (c) A set of overlapped brush contact patch snapshots was obtained in the experiment. Angles α_0 , α_1 , and β_1 are shown for illustration.

3.2. Estimating Brush Parameters from Canvas and Video

From samples of brush footprints given in the left panel of Figure 9, values of w were measured, and their smooth fit for the dependency on z was estimated as a function containing a third-order polynomial:

$$w(z) = \max(0, p_1z^3 + p_2z^2 + p_3z + p_4), \tag{24}$$

where $p_1 = 0.024$, $p_2 = -0.42$, $p_3 = 2.75$, and $p_4 = -2.35$. The goodness of fit of this polynomial is given by values $R^2 = 0.98$, $RMSE = 0.51$, where R^2 is the coefficient of determination, and RMSE stands for root mean square error.

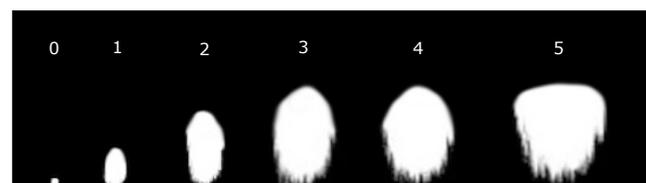


Figure 9. Experimentally obtained brush footprints for normal brush geometry and the brush that was tipped into paint by 7 mm; numbers refer to half z .

An estimation of the values for brush radius r obtained from two examples of a calibration canvas are shown in the central panel of Figure 10. The obtained estimation for the fourth-order polynomial is

$$f_r(z) = p_1z^4 + p_2z^3 + p_3z^2 + p_4z + p_5, \tag{25}$$

$$p_1 = 2.1 \cdot 10^{-4} \pm 0.0051,$$

$$p_2 = -5.0 \cdot 10^{-2} \pm 0.103,$$

$$p_3 = 0.31 \pm 0.68,$$

$$p_4 = 0.86 \pm 1.57,$$

$$p_5 = 3.1 \pm 0.9,$$

all error bounds are given with confidence $p = 95\%$. The goodness of fit is characterized by $R^2 = 0.98$ and $RMSE = 0.58$. In comparison with the estimation obtained from angles, the obtained estimation is three times more accurate in terms of RMSE and has the coefficient of determination sufficiently closer to 1, which means a lower error in data and better model accuracy.

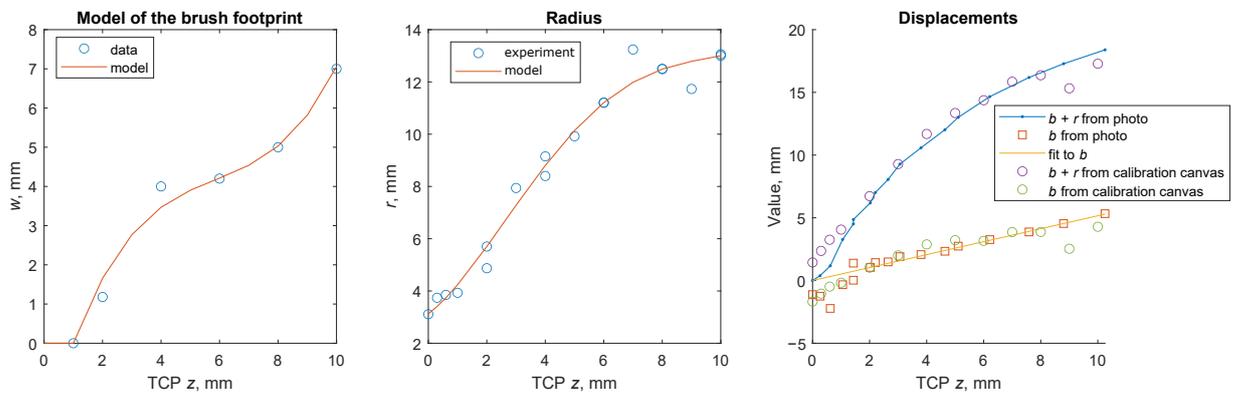


Figure 10. Estimated values of w , r , and b , obtained calibration canvas and photo (video), and their smooth fits as functions of z .

For b , a linear law was obtained from video data; see the red squares in the right panel of Figure 10:

$$f_b(z) = k_bz, \tag{26}$$

the estimated coefficient $k_b = 0.52 \pm 0.09$, $p = 95\%$. The goodness of fit estimations are $R^2 = 0.83$, $RMSE = 0.85$, which is also better than in the case of using angles. Estimation of a linear law from the calibration canvas are presented as green circles in Figure 10: $k_b = 0.46 \pm 0.11$, $p = 95\%$, goodness of fit $R^2 = 0.78$, $RMSE = 0.95$, so the most accurate estimation is obtained from the photo.

3.3. Estimating Brush Parameters from Angles

For each integer value of z in the interval $[1 \dots 10]$, a series of six snapshots was obtained, resulting in six values of α_i and five values of β_i and s_i in each series. Using Formulas (20) and (23), sets of 10 estimations for r and b were obtained. After that, a MATLAB function `fit` with the least absolute residuals (LARs) robust method was used to fit

parameters with smooth approximations. For r , the fourth-order polynomial model was obtained:

$$f_r(z) = p_1z^4 + p_2z^3 + p_3z^2 + p_4z + p_5,$$

$$p_1 = -4.56 \cdot 10^{-4} \pm 0.0396,$$

$$p_2 = -4.56 \cdot 10^{-3} \pm 0.8754,$$

$$p_3 = 0.047 \pm 6.457,$$

$$p_4 = 1.39 \pm 16.13,$$

$$p_5 = 3.4 \pm 10.0.$$

All parameter confidence intervals are given with $p = 95\%$. The goodness of fit of estimations are $R^2 = 0.82, RMSE = 1.98$.

For b , a linear law was adopted:

$$f_b(z) = k_bz, \tag{27}$$

where the fit function gave the following value: $k_b = 0.21 \pm 0.14, p = 95\%$. The goodness of fit estimations is $R^2 = 0.39, RMSE = 1.21$.

Figure 11 shows the brush parameter values in dependence of z and their fits.

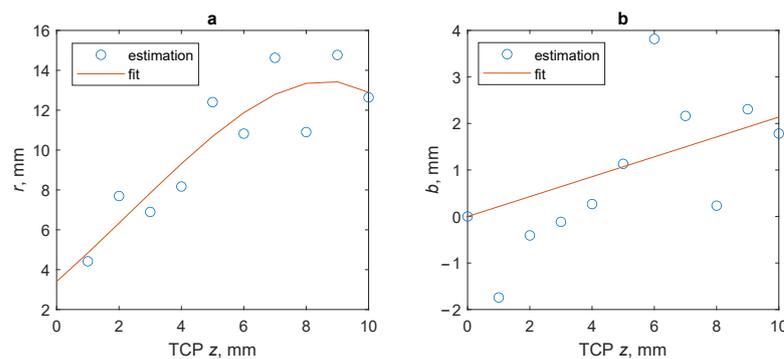


Figure 11. Estimated values of (a) r and (b) b , obtained from angles α and β in the experiments, and their smooth fits as functions of z .

Approximated dependencies $f_r(z)$ and $f_b(z)$ can be further used in the executable model of the brush allowing the accurate prediction of the brushstroke shape. One can see that the obtained values have rather large dispersion, which is also confirmed by relatively large values of RMSE.

3.4. Verifying the Brush Model

As we saw in a previous section, the parameter estimation from the canvas and video was more accurate, so we used them in further tests. The first test was used to validate the equations of motion (7), (5), and (6). On the calibration canvas, several angular brushstrokes were painted, one of which is exemplified in Figure 12. The panel (a) presents a brushstroke obtained with TCP $z = 10$ mm with two overlapped center lines: the solid line shown in yellow is determined manually from the scanned image, and the dashed red line shows the trajectory $(x(\alpha), y(\alpha))$ given by (5) and (6) for angles $\alpha(s)$ calculated by the Formula (7) for values $s \in [0, 80]$, and the estimated value of the brush radius is $r = 13.0$ mm.

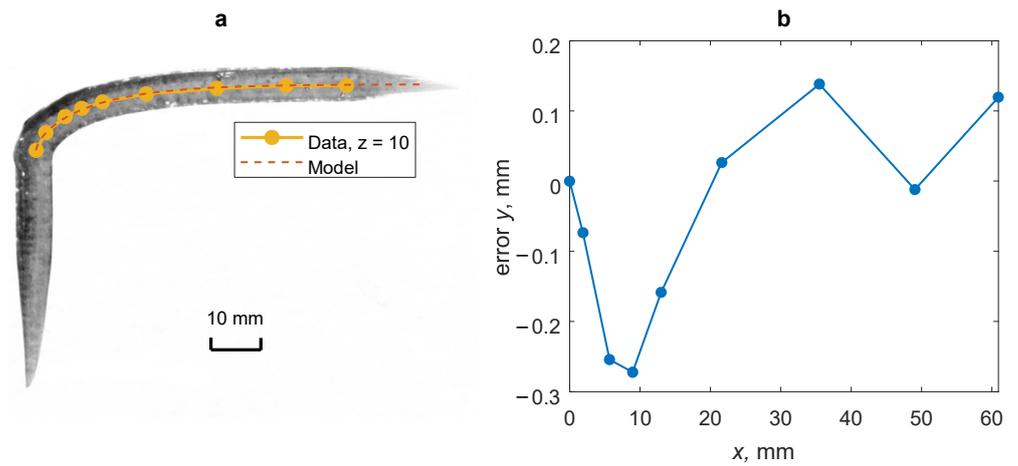


Figure 12. (a) Example of an angular brushstroke obtained in experiment, (b) error in determining the y value from this example.

In Figure 12b, an error estimation for y is given in experimental points shown by yellow markers in Figure 12a. One can see that the error absolute value is relatively small and its relative value does not exceed 2%, which is the level of measurement error, and the overall error estimation for the closed-form solution for the equation of motion (6) is $RMSE = 0.05$.

To visually estimate the quality of the obtained executable model, we present two examples of thin angular brushstrokes for $z = 3$ mm in Figure 13: in panel (a), an experimental brushstroke is shown, and in panel (b), the simulated brushstroke is presented. The brushstroke model ignores paint distribution, so both brushstrokes are depicted in a solid black color. One can see that while some details differ, the overall brushstroke modeling quality is high.



Figure 13. Comparison of two angular strokes: (a) experimental and (b) simulated.

The ability of the brush to perform smooth transitions between lines can be used in robotic painting and calligraphy: replacing often used splines with natural curves given by the brush equations of motion allows for sufficiently reducing the number of trajectory points and thus costs for the brush trajectory optimization. To check the capability of the executable model to reproduce a highly complicated brushstroke shape, an inscription “Vincent”, stylized as the signature of Vincent van Gogh, was simulated using our executable model and painted with the robot. Figure 14a shows the robot TCP trajectory designed by the use of the method from [22], where thin black lines depict trajectory parts with TCP $z > 0$, and colored thick lines depict different straight segments of the inscription with the robot TCP $z \leq 0$. Figure 14b presents the simulated inscription, and Figure 14c gives the result obtained by the robot. One can see that a complicated shape of a piecewise linear trajectory in Figure 14 leads to a smooth and recognizable result in Figure 14b that is verified by the experiment in Figure 14c.

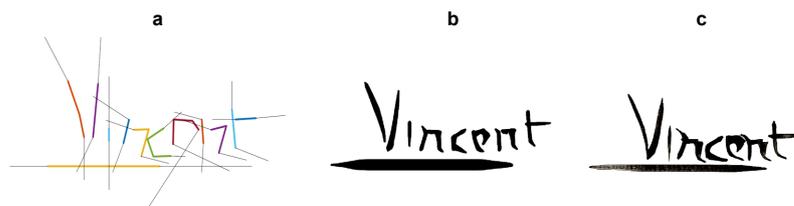


Figure 14. Experiment with writing “Vincent” inscription: (a) Robot TCP trajectory, black thin lines denote TCP path above the canvas, thick lines of random color denote the TCP path when the brush touches the canvas, (b) simulation and (c) a result written by the robot.

The disparity between the results in Figure 14b,c is mostly related to an error calibrating the TCP height above the canvas, leading to slightly longer real brushstrokes in comparison with the simulation.

One more experiment was carried out to estimate how more complex shapes determined with Bezier curves could be rendered with the proposed model. In Figure 15, the test brushstroke set is presented. Figure 15a demonstrates TCP trajectories and simulated brushstroke skeletons, and Figure 15b presents how these strokes were painted by the robot: red strokes are real scanned strokes, and blue strokes are simulated.

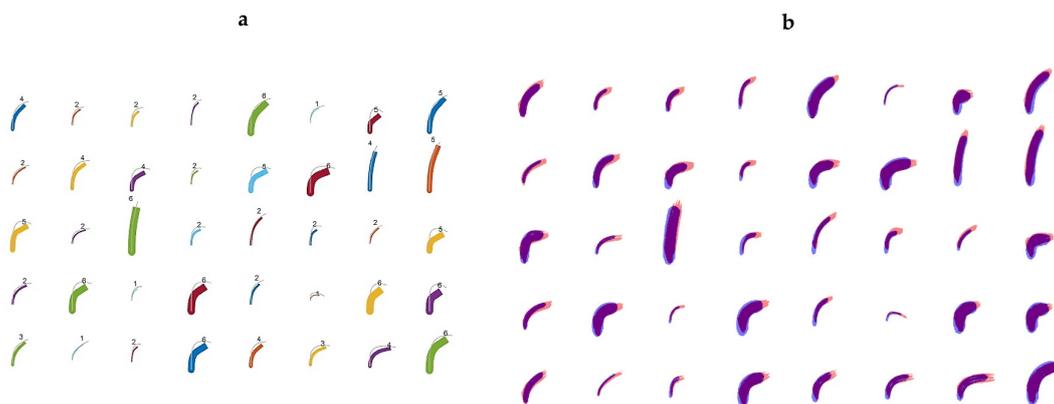


Figure 15. Brushstrokes determined with Bezier curves. (a) Gray lines—robot TCP trajectories, lines of random colors—stroke skeletons drawn with lines of different widths denoted by numbers. (b) Blue strokes are simulated, and red strokes are real strokes from the experiment.

All strokes were painted from bottom to top. One can see a good correspondence between the stroke width and the stroke’s curved shape. Nevertheless, a disparity between stroke ends is clearly observed, which corresponds to a slight mismatch between model z and the real z position of the robot TCP and slightly deeper tipping of the brush into paint than that performed during calibration. Both these errors are due to the manual calibration of the robot coordinate frame and filling the cuvette from where the robot was taking paint. This shows the importance of fully automating the process of brush calibration and setup maintenance in the future, which is the only way to ensure that the parameters are set up with satisfactory precision.

3.5. Experiments with Liner Brush

The most interesting case for verifying the model is the liner brush, since this type of brush is extremely thin and long-bristled compared to the other types of brushes, and compliance plays the most important role here. An estimation of width and radius performed for the synthetic liner brush “Gamma” number 1 from the calibration canvas is given in Figure 16. This brush has a footprint width equal to 1–2 mm, and the length of the bristles is 20 mm. The specific feature of the liner brush is the extremely small brush footprint which allows painting thin lines and letterings. On the other hand, its disadvantage is a

relatively large radius in comparison to its width, which leads to the need to use trajectories with small curvatures. Using the method of automatic brush path generation from [22], we painted two variants of the lettering “Robot”, as shown in Figure 17. The number of control points in the trajectories was relatively small, from 6 to 14 for each stroke. The curvature of the strokes was at many points excessive for the given brush radius, which led to disruptions on the letters’ edges. Nevertheless, the brush model predicted this non-ideal behavior perfectly, and the mismatch between the predicted and actual trajectories given in Figure 17 is relatively small. The video showing the process of writing the lettering with the liner brush can be found in the reference [33].

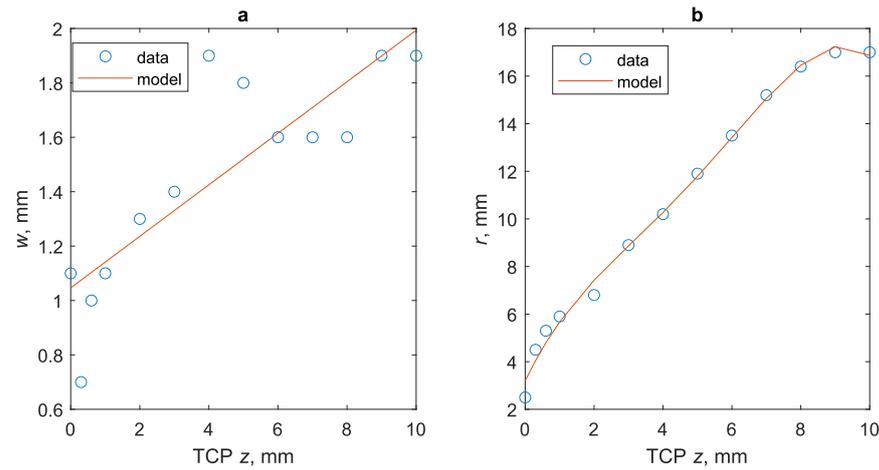


Figure 16. Estimated values of w and r for the 1 mm liner brush: (a) dependence of the brushstroke width of the TCP z , (b) dependence of r on the TCP z .

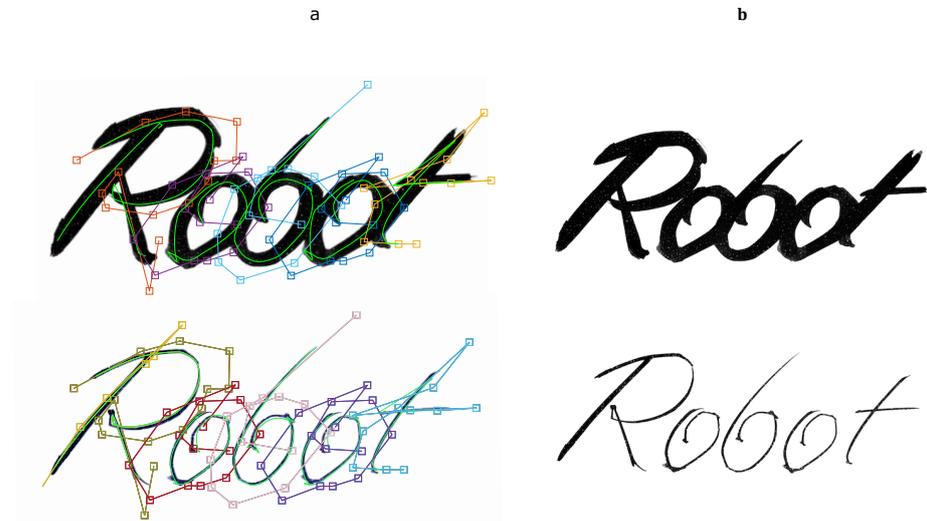


Figure 17. Two variants of the lettering “Robot” written by the robot using a 6 mm round brush (upper row) and the 1 mm liner brush (lower row). In the panel (a), the trajectory of the robot TCP is overlaid on images of letterings, and the brushstroke skeletons shown in green were obtained from the model (4). In the panel (b), images of letterings only are given.

One more experiment was carried out to clarify the repeatability of the brush trajectory. It is known that the brush is subject to the hysteresis effect, which is expressed by preserving the bristle deflection after the brush makes a stroke. This effect is strongly related to the brush wetness but is not easily controlled, so it may spontaneously occur in practice. In the experiment, the robot wrote five letters of the English alphabet “abcdo” in series, as shown in Figure 18a.

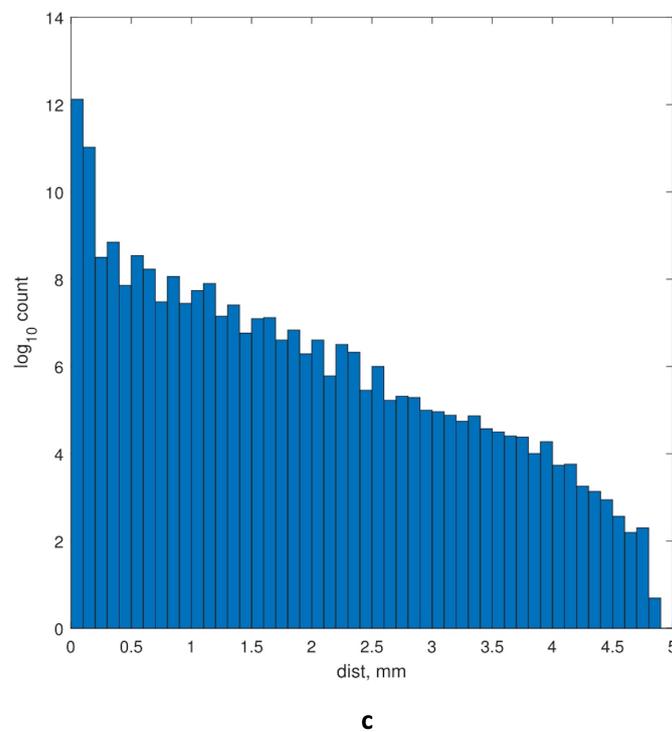
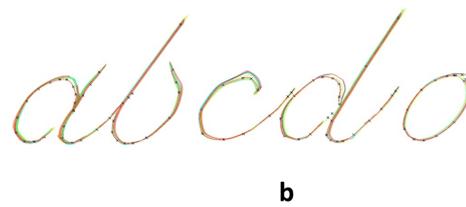
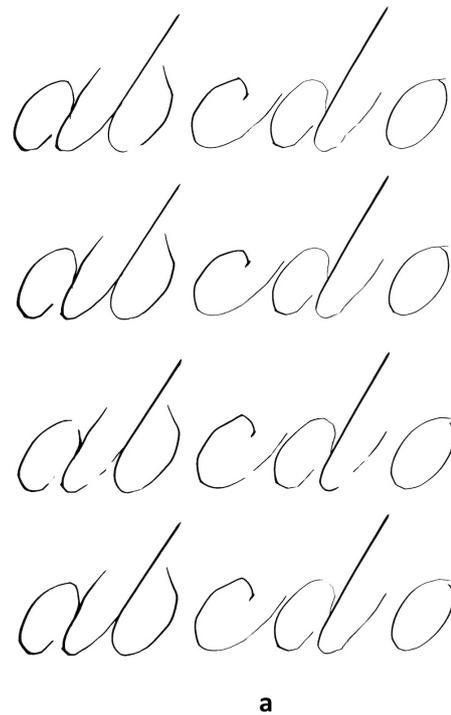


Figure 18. (a) An inscription “abcdo” written with a liner brush four times, (b) the reference trajectory, overlapping colored images of different sample inscriptions, (c) the histogram of distances between points in the painted and reference trajectory. The maximal obtained displacement is 5 mm.

Figure 18b shows all four inscriptions overlapped and a simulated trajectory with reference points marked with crosses. It is clearly seen that several elements of letters were rather variable, such as the upper parts of the letters “c” and “a”. To verify the repeatability numerically, we plotted the histogram of distances between the points of the painted and reference trajectory, as shown in Figure 18c. The maximal obtained displacement is 5 mm, but the typical values are near zero. One can see a nearly linear decay of counts in a logarithmic scale, which relates to nearly exponential decay on a linear scale.

4. Discussion: Recommended Calibration Algorithm

First, brush footprints should be obtained for different values of z using the setup shown in Figure 8a. The matte glass can be purchased in a glass shop or manufactured manually, for example, by sanding an acrylic glass. The brush tip should be wet, and the length of the wet region of the bristles should correspond to the amount of paint that is intended to be used in a real painting process. For example, in our experiment, the brush was tipped into the paint by 7 mm, so the footprint’s maximal height was 7 mm.

Second, a set of angular calibration brushstrokes as shown in Figure 6b should be painted with different values of z to obtain a reliable estimation of the brush parameter r and its dependency on z , which is described as a polynomial like (25).

Third, the parameter b should be estimated. The best result can be obtained using video footage as shown in Figure 7. The value of b is obtained using the Formula (18). A simpler but less accurate result is achieved from straight calibration strokes as shown in Figure 6a. Using the polynomial approximation, a function (24) should be estimated describing how the distance between the brush tip and the thickest part of a brush footprint depends on z . Then, the Formula (17) is used to obtain values of b . For further use, a smooth dependency of values of b on z should be obtained in the form of a linear law (23).

Using angles α and β for calibration is more error-prone, but in some cases, it is also useful for estimating parameters r and b .

The described brush calibration process contains several manual operations in the experimental result processing, which can be automated. For example, determining the parameter r from the angular brushstroke can be performed with an algorithm consisting of the following stages. First, the stroke skeleton can be found by a simple deterministic algorithm such as that described in the work [34]; then, the point of maximum skeleton deflection can be found using the Hough transform. From the y coordinate of this point, r is found as follows: $r = y_0 - y$, where y_0 is the maximal robot TCP y coordinate while drawing the angular brushstroke.

5. Conclusions

In this paper, a physically motivated model of a round painting brush was proposed. Equations of motion of the brush contact patch are presented in the form of ordinary differential equations and their closed-form solutions, which are useful in various applications including robot trajectory optimization and brushstroke simulation. Approximation of the model parameters’ dependencies from the robot TCP distance from the canvas z is given, and the method for these parameters calibration was discussed. Both equations and the executable models were carefully verified in several tests.

Future work will include developing a path-planning algorithm based on an iterative optimization of the robot trajectory and designing software utilizing the proposed model. We plan to test this algorithm in a feedback-guided painting process with acrylic paints. The proposed model will be extended to brushes of other shapes, such as flat brushes. The studies with fully automatic brush calibration are also yet to be performed. In addition to that, an interactive program for designing robot calligraphy and lettering will be developed, allowing calligraphers to implement their ideas using the painting robot instead of writing the inscriptions manually. The brush model in this program would be used for real-time path planning and brushstroke simulation ensuring the selected brush is suitable for the designed inscription.

Author Contributions: Conceptualization, A.K. and D.B.; data curation, S.M.; formal analysis, M.S. and D.B.; funding acquisition, A.K.; investigation, D.G. and S.M.; methodology, A.K.; project administration, A.K.; resources, A.K., M.S. and S.M.; software, M.S., D.G. and S.M.; supervision, A.K., S.L. and D.B.; validation, D.G., S.L. and D.B.; writing—original draft, A.K.; writing—review and editing, S.L. and D.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Russian Science Foundation, project № 22-79-00171.

Data Availability Statement: The original contributions presented in the study are included in the article; further inquiries can be directed to the corresponding authors.

Acknowledgments: Authors are grateful to A. Skuratov and E. Shteinberg for their help in the development of the robot control software.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Carnovalini, F.; Rodà, A. Computational Creativity and Music Generation Systems: An Introduction to the State of the Art. *Frontiers in Artificial Intelligence*. *Front. Artif. Intell.* **2020**, *3*, 14. [[CrossRef](#)] [[PubMed](#)]
2. Epstein, Z.; Hertzmann, A.; Investigators of Human Creativity; Akten, M.; Farid, H.; Fjeld, J.; Frank, M.R.; Groh, M.; Herman, L.; Leach, N.; et al. Art and the science of generative AI. *Science* **2023**, *380*, 1110–1111. [[CrossRef](#)] [[PubMed](#)]
3. Wang, X.; Gong, Z. Style Generation in Robot Calligraphy with Deep Generative Adversarial Networks. *arXiv* **2023**, arXiv:2312.09673.
4. Miyata, C.; Chisholm, K.; Baba, J.; Ahmadi, M. A limb compliant sensing strategy for robot collision reaction. *IEEE/ASME Trans. Mechatron.* **2015**, *21*, 674–682. [[CrossRef](#)]
5. Deng, C.; Wang, S.; Wang, J.; Xu, Y.; Chen, Z. LiDAR Depth Cluster Active Detection and Localization for a UAV with Partial Information Loss in GNSS. In *Unmanned Systems*; World Scientific: Singapore, 2024.
6. Xu, Y.; Chen, Z.; Wang, S.; Wang, J. An Active Landing Recovery Method for Quadrotor UAV: Localization, Tracking and Buffering Landing. *IFAC-PapersOnLine* **2023**, *56*, 3366–3372. [[CrossRef](#)]
7. Karimov, A.I.; Kopets, E.E.; Rybin, V.G.; Leonov, S.V.; Voroshilova, A.I.; Butusov, D.N. Advanced tone rendition technique for a painting robot. *Robot. Auton. Syst.* **2019**, *115*, 17–27. [[CrossRef](#)]
8. Karimov, A.; Kopets, E.; Leonov, S.; Scalera, L.; Butusov, D. A robot for artistic painting in authentic colors. *J. Intell. Robot. Syst.* **2023**, *107*, 34. [[CrossRef](#)]
9. Igno-Rosario, O.; Hernandez-Aguilar, C.; Cruz-Orea, A.; Dominguez-Pacheco, A. Interactive system for painting artworks by regions using a robot. *Robot. Auton. Syst.* **2019**, *121*, 103263. [[CrossRef](#)]
10. Schaldenbrand, P.; McCann, J.; Oh, J. Frida: A collaborative robot painter with a differentiable, real2sim2real planning environment. In Proceedings of the 2023 IEEE International Conference on Robotics and Automation (ICRA), London, UK, 29 May–2 June 2023; pp. 11712–11718.
11. Beltramello, A.; Scalera, L.; Seriani, S.; Gallina, P. Artistic robotic painting using the palette knife technique. *Robotics* **2020**, *9*, 15. [[CrossRef](#)]
12. Scalera, L.; Gasparetto, A.; Seriani, S.; Gallina, P. History of Drawing Robots. In *International Symposium on History of Machines and Mechanisms*; Springer: Berlin/Heidelberg, Germany, 2024; pp. 3–17.
13. Scalera, L.; Seriani, S.; Gasparetto, A.; Gallina, P. Watercolour robotic painting: A novel automatic system for artistic rendering. *J. Intell. Robot. Syst.* **2019**, *95*, 871–886. [[CrossRef](#)]
14. Chen, G.; Baek, S.; Florez, J.D.; Qian, W.; Leigh, S.w.; Hutchinson, S.; Dellaert, F. GTGraffiti: Spray painting graffiti art from human painting motions with a cable driven parallel robot. In Proceedings of the 2022 International Conference on Robotics and Automation (ICRA), Philadelphia, PA, USA, 23–27 May 2022; pp. 4065–4072.
15. Trigatti, G.; Boscarriol, P.; Scalera, L.; Pillan, D.; Gasparetto, A. A look-ahead trajectory planning algorithm for spray painting robots with non-spherical wrists. In *Mechanism Design for Robotics*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 235–242.
16. Deussen, O.; Lindemeier, T.; Pirk, S.; Tautzenberger, M. Feedback-guided stroke placement for a painting machine. In Proceedings of the Eighth Annual Symposium on Computational Aesthetics in Graphics, Visualization, and Imaging, Annecy, France, 4–6 June 2012; pp. 25–33.
17. Lee, J. Physically-based modeling of brush painting. *Comput. Netw. ISDN Syst.* **1997**, *29*, 1571–1576. [[CrossRef](#)]
18. Bai, B.; Wong, K.W.; Zhang, Y. An efficient physically-based model for chinese brush. In Proceedings of the Frontiers in Algorithmics: First Annual International Workshop, FAW 2007, Proceedings 1, Lanzhou, China, 1–3 August 2007; Springer: Berlin/Heidelberg, Germany, 2007; pp. 261–270.
19. Chu, N.H.; Tai, C.L. An efficient brush model for physically-based 3D painting. In Proceedings of the 10th Pacific Conference on Computer Graphics and Applications, Beijing, China, 9–11 October 2002; pp. 413–421.
20. Sun, Y.; Xu, Y. A calligraphy robot—Callibot: Design, analysis and applications. In Proceedings of the 2013 IEEE International Conference on Robotics and Biomimetics (ROBIO), Shenzhen, China, 12–14 December 2013; pp. 185–190.

21. Lam, J.H.; Yam, Y. Stroke trajectory generation experiment for a robotic Chinese calligrapher using a geometric brush footprint model. In Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, St. Louis, MO, USA, 10–15 October 2009; pp. 2315–2320.
22. Zingrebe, D.S.; Gülzow, J.M.; Deussen, O. Robotic Writing of Arbitrary Unicode Characters Using Paintbrushes. *Robotics* **2023**, *12*, 72. [[CrossRef](#)]
23. Wang, S.; Chen, J.; Deng, X.; Hutchinson, S.; Dellaert, F. Robot calligraphy using pseudospectral optimal control in conjunction with a novel dynamic brush model. In Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 24 October 2020–24 January 2021; pp. 6696–6703.
24. Yan, G.; Guo, D.; Min, H. Robot calligraphy based on footprint model and brush trajectory extraction. In Proceedings of the International Conference on Cognitive Systems and Signal Processing, Fuzhou, China, 17–18 December 2022; Springer: Berlin/Heidelberg, Germany, 2022; pp. 437–451.
25. Wu, R.; Chao, F.; Zhou, C.; Chang, X.; Yang, L.; Shang, C.; Zhang, Z.; Shen, Q. Internal model control structure inspired robotic calligraphy system. *IEEE Trans. Ind. Inform.* **2023**, *20*, 2600–2610. [[CrossRef](#)]
26. Lin, H.I.; Chen, X.; Lin, T.T. Calligraphy brush trajectory control of by a robotic arm. *Appl. Sci.* **2020**, *10*, 8694. [[CrossRef](#)]
27. Hörmann, G.; Oparnica, L. Generalized solutions for the Euler–Bernoulli model with distributional forces. *J. Math. Anal. Appl.* **2009**, *357*, 142–153. [[CrossRef](#)]
28. Han, S.M.; Benaroya, H.; Wei, T. Dynamics of transversely vibrating beams using four engineering theories. *J. Sound Vib.* **1999**, *225*, 935–988. [[CrossRef](#)]
29. Wang, Y.; Zhu, X.; Dai, H. Exact solutions for the static bending of Euler-Bernoulli beams using Eringen’s two-phase local/nonlocal model. *AIP Adv.* **2016**, *6*, 085114. [[CrossRef](#)]
30. Huang, L.; Hou, Z. A novel virtual 3D brush model based on variable stiffness and haptic feedback. *Math. Probl. Eng.* **2020**, *2020*, 6942947. [[CrossRef](#)]
31. Robot Painting Corner Brushstroke, 2024. Available online: <https://youtube.com/shorts/8ch2ofuWwYY> (accessed on 20 April 2024).
32. Bidgoli, A.; De Guevara, M.L.; Hsiung, C.; Oh, J.; Kang, E. Artistic style in robotic painting; a machine learning approach to learning brushstroke from human artists. In Proceedings of the 2020 29th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN), Naples, Italy, 31 August 2020–4 September 2020; pp. 412–418.
33. Robot Writing “Robot”, 2024. Available online: <https://youtu.be/GhWa2W70a4I> (accessed on 4 May 2024).
34. Lamberti, F.; Sanna, A.; Paravati, G. Computer-assisted analysis of painting brushstrokes: Digital image processing for unsupervised extraction of visible features from van Gogh’s works. *EURASIP J. Image Video Process.* **2014**, *2014*, 53. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.