robotics

MDPI

Article

Non-Photorealistic Rendering Techniques for Artistic Robotic Painting

Lorenzo Scalera [1,2,*], Stefano Seriani [3], Alessandro Gasparetto [1] and Paolo Gallina [3]

[1] Polytechnic Department of Engineering and Architecture, University of Udine, 33100 Udine, Italy; alessandro.gasparetto@uniud.it
[2] Faculty of Science and Technology, Free University of Bozen-Bolzano, 39100 Bolzano, Italy
[3] Department of Engineering and Architecture, University of Trieste, 34127 Trieste, Italy; sseriani@units.it (S.S.); pgallina@units.it (P.G.)
* Correspondence: lorenzo.scalera@uniud.it

Received: 21 December 2018; Accepted: 7 February 2019; Published: 11 February 2019

Abstract: In this paper, we present non-photorealistic rendering techniques that are applied together with a painting robot to realize artworks with original styles. Our robotic painting system is called Busker Robot and it has been considered of interest in recent art fairs and international exhibitions. It consists of a six degree-of-freedom collaborative robot and a series of image processing and path planning algorithms. In particular, here, two different rendering techniques are presented and a description of the experimental set-up is carried out. Finally, the experimental results are discussed by analyzing the elements that can account for the aesthetic appreciation of the artworks.

Keywords: painting robot; collaborative robot; image processing; non-photorealistic rendering; artistic rendering

1. Introduction

Robotic painting is a challenging task that is motivated by an inner wish to discover novel forms of art and to experiment the technological advances to create something that can be aesthetically appreciated. Developing an automatic robotic painting system is hard, since the process comprises several different fields, including robotics, automation, image processing and art.

Busker Robot has been developed since 2016 at University of Trieste, Italy, in collaboration with University of Udine, Italy. The robotic system is composed of a six degree-of-freedom (DOF) robotic arm and a series of image processing and path planning algorithms that are capable of interpreting an input digital image into a real artwork. The aim of the work is, therefore, not to faithfully reproduce an image, as typical printers or plotters do, but to introduce an original contribution. The image processing is the result of the extensive implementation of non-photorealistic rendering (NPR) techniques with a trial and error procedure that is developed to the fulfillment of the artist. In each algorithm that we have implemented, several parameters can be controlled in order to modify the desired outcome. Moreover, these techniques are not fully deterministic, since random effects have been introduced, as explained in the paper, to obtain a different and non-repeatable result every time the algorithm is run.

The robotic machine that we have adopted for our purposes is a UR10 collaborative robot, produced by Universal Robots. Its collaborative features, such as force and speed limits as well as collision-detection systems, allow an artist to work side by side with the robotic arm. For example, the paint level, the type of brush and color can be changed or adjusted during the painting process, when needed.

Busker Robot, which name refers to street artists, has been previously presented in [1,2] and has been showcased for the first time in 2016 at the exhibition "Art, Science, Technology, Robotics" in Trieste, Italy. Then, it was shown at the SPS IPC Drives Italy 2017, at the "Algorithmic Arts and Robotics" exhibition during the international event Trieste Next 2017, at "Piccolo Teatro" in Milan, 2017, and, more recently, at the international festival "Robotics" in Trieste, 2018 (Figure 1) [3]. Furthermore, it took part to the 2018 International Robotic Art Competition (RobotArt), a context where 19 teams from all over the world competed and more than 100 artworks were created by painting robots; Busker Robot won an Honorable Mention [4].

Figure 1. Busker Robot at the international festival "Robotics" (Trieste, November 2018).

With respect to previously published works [1,2], in this paper: (a) we present *two novel non-photorealistic rendering techniques*, *cross-hatching* and Random Splines, which are adopted for the artistic rendering of light and shades on an image, and (b) we employ *non-diluted black Indian ink*, instead of the previously adopted watercolor and gouache techniques.

The change from watercolor to ink was prompted by the idea to experiment the filling of areas with patterns of thin lines, in order to obtain different effects from those generated by water and pigments in the watercolor and gouache techniques. With respect to watercolor, in which the brush precision is not a big issue since the effects generated in the watered paper are uncontrollable and random, by handling the ink, the accuracy is much more important. Moreover, the height of the brush, the interval between two consecutive color refills and the planarity of the paper are much more critical.

With the ink painting technique several layers of thin lines and curves are applied to the paper, each one overlapped to the previous one, in order to produce dark effects in the shaded areas and lightness in the areas that are not covered by strokes. Experimental results show the feasibility of the proposed approach and good performances of the system architecture.

In the next section we provide a brief background including related works, in Section 3 we describe the architecture of the painting system, in Section 4 we present the non-photorealistic rendering techniques, and in Section 5 we include a description of the hardware set-up. The results are reported in Section 6, whereas Section 7 highlights the conclusion of this work and possible future improvements.

2. Related Work

Since the middle of the twentieth century, when the first programmable machines were adopted and what is known as "industrial robotics" started [5], research in this field has been focused in developing manipulators capable of replacing human labor in hazardous environments, which increases workplace safety and enhances production levels. By contrast, in recent years, the original industrial use has evolved and several artists have brought robotic machines from the factories to public exhibition spaces to develop novel artistic forms.

One of the earliest painting machines can be identified in the draughtsman automaton by Pierre Jaquet Droz, built in the 1760s [6]. In the last century, Jean Tinguely (1925–1991) can be considered one of the first artists that created painting machines to produce artworks with mostly random patterns [7]. Harold Cohen (1928–2016) devised AARON, a plotter capable of reproducing computer-generated artworks, considered pioneer in algorithmic and generative art [8].

In more recent years, several artists, engineers and computer scientists developed robots and machines skilled at drawing and painting. One example is given by the humanoid drawing human portraits by S. Calinon et al. [9], whereas G. Jean-Pierre and Z. Said developed the Artist Robot, a 6-DOF industrial robot programmed for drawing portraits using visual feedback [10]. Other examples are given by the works of C. Aguilar et al., who developed a system composed of an articulated painting arm and a machine-learning algorithm aimed at determining a series of brush strokes to be painted on the canvas [11]. An interesting example of robotic installation capable of drawing sketches of people is given by the machine developed by P. Tresset and F. F. Leymarie [12,13]. The system, based on visual feedback to iteratively augment and improve the sketch, draws using the equivalent of an artist's stylistic signature. Furthermore, T. Lindemeier et al. presented e-David, a modified industrial robot that works with visual feedback and it is based on non-photorealistic rendering techniques [14,15]. The painting machine applies different layers of strokes on the canvas and produces beautiful artworks that mimic those produced by humans. Recently, the brush technique has been further improved by J. M. Gülzow et al. [16], who developed a method to analyze human brushstrokes and replicate them by means of the robotic painting system with impressive results.

More recently, examples of robots capable of producing artworks can be found in the work of X. Dong et al., who adopted a Scara robot to paint stylized portraits obtained with an algorithm based on a triangle coordinate system [17], of D. Berio et al., who adopted a collaborative Baxter robot to produce graffiti strokes [18], and by D. Song et al., who presented a 7-DOF robotic pen-drawing system, that is capable of creating pen art on arbitrary surfaces [19]. Furthermore, a robot artist performing cartoon style facial portrait painting using NPR techniques has been presented in [20].

Another application of robots in art, other than in pen drawing and brush painting, is that of spray painting, a task typically employed in industry [21–23]. Examples of robotic spray painting for artistic purposes can be found in the works of Viola Ago [24] and L. Scalera et al. [25], who adopted industrial robotic arms equipped with an airbrush, and in the work of A. S. Vempati et al., who developed PaintCopter, an autonomous UAV for spray painting on 3D surfaces [26].

Finally, other examples of artistically skilled robots are given by painting mobile platforms, such the ones by L. Moura, who developed Robotic Action Painter in 2006 [27], and by C.-L. Shih, who presented the trajectory planning and control of a mobile robot for drawing applications [28].

3. Busker Robot

Busker Robot is a system with amodular architecture, consisting of both hardware and software parameters (Figure 2). The hardware is composed of the painting machine, a 6-DOF robotic arm by Universal Robots, mounted in front of the painting support and equipped with a changeable brush. The software consists of the image processing and the path planning algorithms, that have been implemented in MATLAB®. The image processing algorithms, i.e., the non-photorealistic rendering techniques, are applied to a digital input image that is processed in order to extract the contours, the details and the backgrounds that have to be reproduced on paper. Several different algorithms

have been developed for Busker Robot [1,2]: it is the artists responsibility to choose which technique, or combination of techniques, has to be applied case by case.

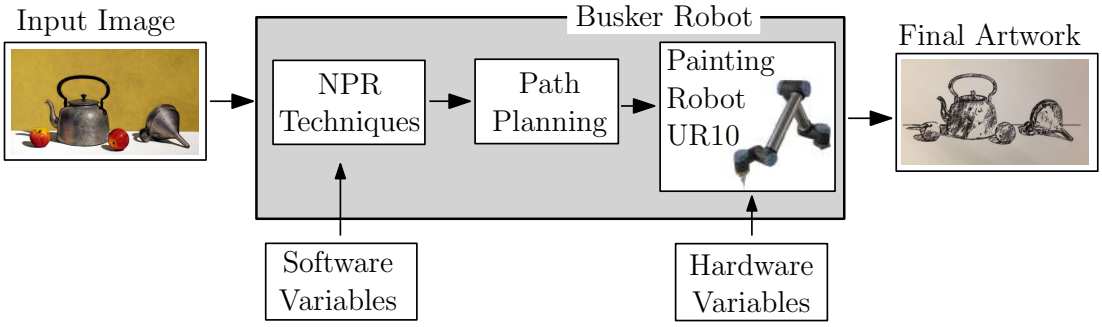


Figure 2. Overview on the system architecture.

The output of the artistic rendering is a structured list of points that identify the processed image. In particular, a MATLAB structure is adopted to handle the pixel coordinates of each single line or curve that composes the final image: each stroke is represented by an array that contains the coordinates of all the required way-points. These pixel coordinates are then scaled into the corresponding coordinates in millimeters on the painting canvas with a linear transformation. A trajectory planning is applied to these paths, in order to define the velocities profiles that the robot end-effector has to follow during the execution of the painting task. The trajectories are planned by adopting trapezoidal speed profiles for each motion: after the acceleration phase, the end-effector of the robot linearly moves with constant speed and circular blends through the way-points. After a stroke is completed, the robot tool z-coordinate is increased in order to lift the brush from the paper; it is then moved linearly at constant speed towards the first point of the next stroke but remaining at a higher z-coordinate; finally, it is lowered at paper level and the new stroke is started.

In the trajectory planning module, all the commands for the paint refill and the brush drying are as well defined. The output of this process is the list of commands for the robot, written in the specific UR Script Programming Language in a specific *script* file, which includes all the instructions needed for a specific layer of the artwork (Section 4). In particular, each command line corresponds to the motion of the robot end-effector through a specific way-point and it includes the Cartesian position coordinates in the operational space, the orientation of the tool in axis-angle representation, the velocity, the acceleration and the blend radius.

The execution of the painting task is then started and monitored from a remote computer connected to the robot controller via Ethernet. It is worth noting that the software is modular and, therefore, if any parameter regarding the painting set-up has to be changed, only the trajectory planning can be rerun, based on the same results of the artistic rendering module.

4. Non-Photorealistic Rendering Techniques

The two novel non-photorealistic rendering techniques implemented in this work are described in the following. These techniques can be adopted for the processing of the light and shades of an image in a manner that can enhance the aesthetic appreciation. Indeed, we would like to introduce a motor activity in the brush painting, so as to recall the gestures of a human artist in the robotic strokes. It has been demonstrated that the aesthetic experiences are enhanced in dynamic paintings that evoke a sense of movement and activate the brain area of visual motion during the observation [29,30].

Similarly to [1], the algorithms here presented are based on a thresholding process that is applied to a grey-scale version of the original image. For a given threshold I_T, i.e., the value of a specific grey intensity level (between 0 and 1), each pixel of the original image with intensity $I(P_i)$, is transformed into white if $I(P_i) \geq I_T$, or into black, otherwise. In this manner, the image can be decomposed into several layers, each one characterized by a different area to be painted, in a way similar to [31].

The algorithms presented in the following, *cross-hatching* and Random Splines, are applied to several layers of the same image. The paths resulting from the algorithms are then overlapped to obtain the final result. The NPR algorithms can be used for the processing of the areas with uniform grey intensity level of an image; furthermore, contours and details can also be added to the final result by adopting other techniques such as Canny Edge Detector, Hough Transform or Difference of Gaussians [1,2,32]. The Canny Edge Detector algorithm has been adopted in this work.

4.1. Cross-Hatching

The first NPR technique that we present in this paper is the *cross-hatching*. It consists of several layers of cross-hatch marks that allow to create nuanced differences in tone and light. The basic idea of hatching is to cover an area with parallel lines placed closely together: the areas where the hatching is placed will appear darker or in shadow, whereas the areas without hatching will appear illuminated. After the first set of lines, a second set of hatch marks can be placed on top, with a different orientation with respect to the first layer. Several sets of closely spaced lines can be drawn, corresponding to different thresholds. In this way, more saturated dark effects are obtained in the darkest side of the artwork. In the literature, examples of *cross-hatching* rendering can be found in [33–35].

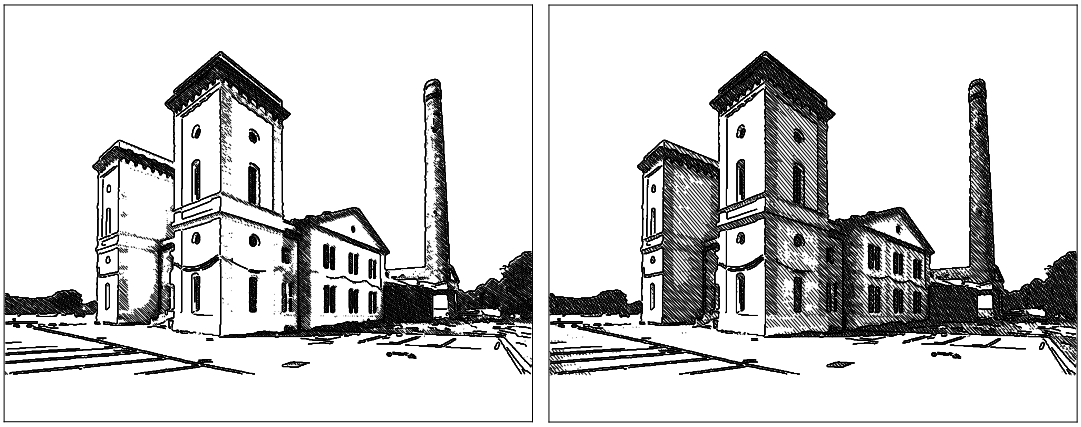
In order to give the artwork greater vibrancy and to reproduce the gesture of an human painter, random effects have been implemented in our algorithm, so as to obtain small deviations (of the order of few millimiters) in the alignment and linearity of the single lines. These are generated by introducing a random perturbation in the longitudinal and transversal direction of a predefined number of points in the line, which becomes a poly-line. The points of the poly-line are then automatically blended to feed the robot with a smooth path. The maximum error for each of the two directions can be manually set before running the algorithm. The lines are finally filtered in order to remove the shortest segments which would otherwise be painted as dots, by the robot.

Two input images have been adopted for the evaluation of the algorithms: the Hydrodynamic Power Station and the Still Life with Tea Kettle, reported in Figure 3a,b. In this work, we used the Canny edge detector algorithm to define the contours for both the two images. Simulated results of the *cross-hatching* algorithms are reported in Figures 4 and 5. In these images, the thickness of the lines is not representative of the real thickness of the brushstrokes, which is given by the z-coordinate of the brush. The threshold and orientation employed for each layer of the images are reported in Tables 1 and 2. From the figures it can be seen that, by adopting higher thresholds, the total covered area increases, the shades are better rendered and the light contrast is more marked.

Table 3 reports the list of software parameters that can be manually tuned in the *cross-hatching* technique.

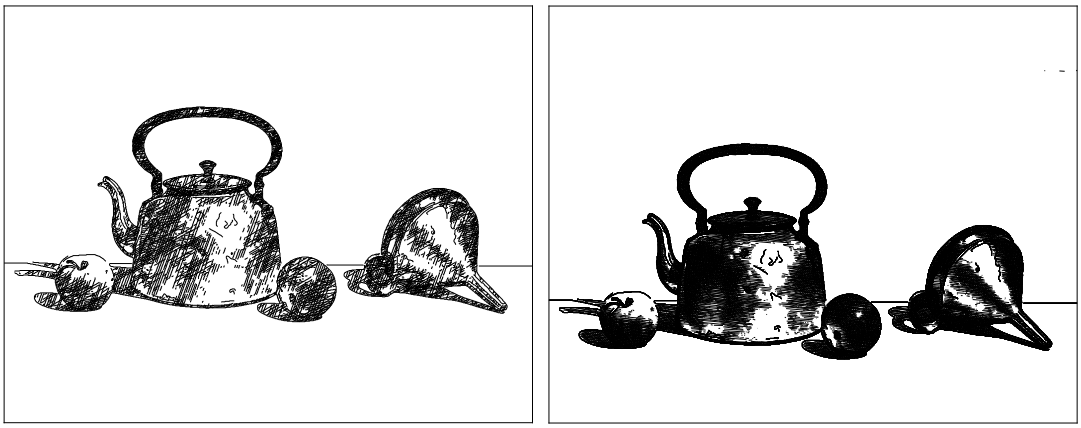
(a) (b)

Figure 3. Original images. (**a**) Hydrodynamic Power Station (Trieste, Italy). (**b**) Still Life with Tea Kettle.



(a) (b)

Figure 4. Hydrodynamic Power Station, simulations with *cross-hatching*, using the parameters reported on the left (**a**) and on the right (**b**) of Table 1.



(a) (b)

Figure 5. Still Life with Tea Kettle, simulations with *cross-hatching*, using the parameters reported on the left (**a**) and on the right (**b**) of Table 2.

Table 1. Threshold and orientation for each layer of the Hydrodynamic Power Station, simulations with *cross-hatching*.

Layer	Figure 4a		Figure 4b	
	Threshold	Angle	Threshold	Angle
1	0.20	80°	0.30	70°
2	0.30	30°	0.40	40°
3	0.35	45°	0.45	50°

Table 2. Threshold and orientation for each layer of the Still Life with Tea Kettle, simulations with *cross-hatching*.

Layer	Figure 5a		Figure 5b	
	Threshold	Angle	Threshold	Angle
1	0.10	5°	0.25	85°
2	0.15	50°	0.30	80°
3	0.30	10°	0.35	50°
4	0.40	30°	0.40	10°
5	0.45	70°	0.45	5°

Table 3. Software variables for the non-photorealistic rendering techniques.

NPR Technique	Software Variables
Cross-hatching	- grey threshold - distance between lines - angle of orientation - maximum error in longitudinal direction - maximum error in transversal direction - minimum length of lines
Random Splines	- grey threshold - number of random points - size of the box - survival rate parameter - minimum length of lines

4.2. Random Splines

A different technique for the rendering of light and shades is Random Splines. Differently from other works in the field of NPR, where splines curves have been used to approximate the contours of the subjects to be painted, e.g., in [36], in this paper we adopt spline curves to fill areas characterized by uniform grey intensity level in a random manner. In this technique, a predefined number of points is randomly generated inside the area defined by a grey-scale threshold. For each point a survival rate is computed, which is proportional to the grey intensity of that point in the original image and to a constant defined by the user. A selection of the points is then applied in order to facilitate the survival of those positioned in the darkest side of the image.

A box centered in each point is generated and three random points are selected within the box boundaries. These points are interpolated with a spline curve, which is then sampled in order to save the sequence of way-points that the robot end-effector has to follow at constant linear speed during the painting task. The splines are sampled in the Cartesian space to obtain a good compromise between accuracy and number of points: a high number of points could, indeed, increase the computational effort of the robot controller and introduce vibrations in the tool if the distance between the points is smaller than the blend radius. A graphical example of the spline generation is shown in Figure 6.

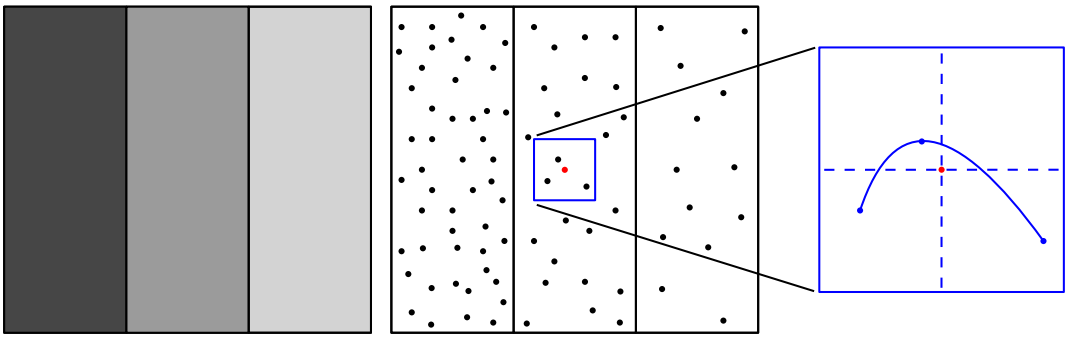
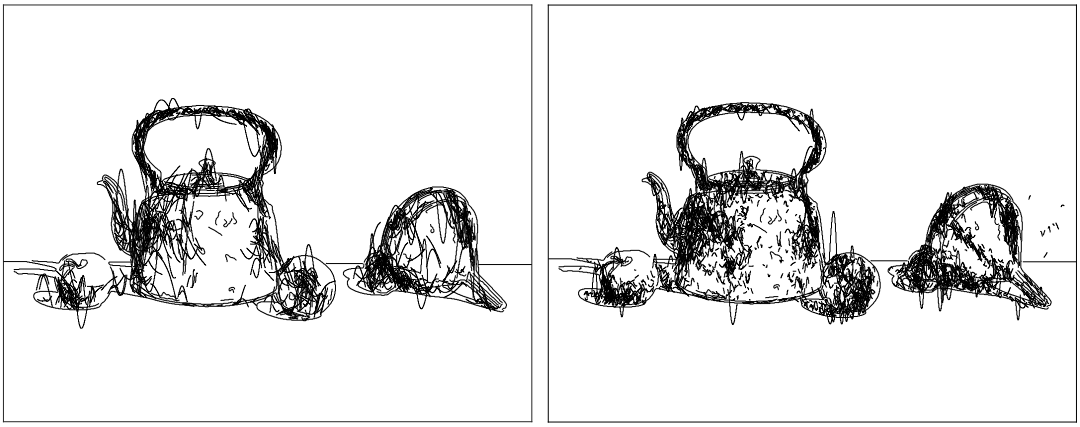


Figure 6. Example of spline generation in the Random Splines algorithm.

The size of the box allows to control the total length of the splines, whereas the survival rate constant allows to concentrate the strokes in the darkest part of the original grey-scale image. Since the points are placed inside a thresholded layer but the box is independent from the layer contours, the resulting splines can be drawn even in parts of the image that do not belong to that layer. This results in lines that exceed the natural border of an object in the image. The resulting splines can be finally filtered to remove those with the shortest path. The software parameters that can be manually defined are listed in Table 3. Two examples of Random Splines for the Still Life with Tea Kettle are shown in Figure 7, obtained with the parameters reported in Table 4.



(a) (b)

Figure 7. Still Life with Tea Kettle, simulations with Random Splines, using the parameters reported on the left (**a**) and on the right (**b**) of Table 4.

Table 4. Threshold and box size (pixel) for each layer of the Still Life with Tea Kettle, simulations with Random Splines.

Layer	Figure 7a		Figure 7b	
	Threshold	Box Size	Threshold	Box Size
1	0.15	30	0.25	30
2	0.20	20	0.30	25
3	0.20	30	0.35	20
4	0.30	60	0.40	15
5	0.40	40	0.45	10
6	0.40	50	0.50	10

5. The Painting Machine

The painting machine that we adopted for Busker Robot is a 6-DOF UR10 robot. It is an industrial collaborative robot, usually adopted for pick-and-place, material handling, assembling or other manufacturing processes in environments where a human-robot interaction is needed. It is provided with speed and torque limits as well as collision-detection systems, that allow a human to work safely side by side with the machine. However, the speed of the robot has to be kept low, since it affects the quality of the painted strokes.

The UR10 can handle 10 kg of payload and can count on 1300 mm of working radius, which allows to easily paint on a 450 mm $\times$ 850 mm surface. Furthermore, a repeatability of ± 0.1 mm on the positioning of the end-effector is ensured by the manufacturer [37]. The six axes allow to paint on non-horizontal surfaces and to perform complex motions such as dipping the brush in the paint cup or scrape it in order to remove excess paint. The robotic arm is fixed on an aluminum frame and its flange is equipped with a custom support for the application of the brushes by means of 3D printed plastic supports. An automatic brush change system has been developed for the adoption of brushes with different size, and used, in a previous work [1], to paint strokes with varied thickness.

The target surface for the painting task is identified with respect to the base coordinate system of the robot (Figure 8). Since a parallelism error between the base of the robot and the target surface could affect the application of color on the paper, a calibration of the painting board is carried out before starting a new artwork. Furthermore, if the target surface is not perfectly planar, a planarity error arises and an approximate interpolating plane has to be defined by more than 3 points. This calibration is performed by measuring a set of n points $P_i = \{x_i \; y_i \; z_i\}^T$, with $i = 1, \ldots, n$, on the drawing surface, with respect to the robot reference frame $\{0 \; 0 \; 0\}^T$. After equipping the end-effector of the robot with a

rigid and sharp tool with the same length of the brush, the tip of this is positioned so as to touch the surface and the corresponding pose of the robot is recorded. In order to calculate the approximating surface $Z = a_0X + a_1Y + a_2$, the following over-determined linear system has to be solved:

$$\begin{cases} a_0x_1 + a_1y_1 + a_2 = z_1 \\ \dots \\ a_0x_n + a_1y_n + a_2 = z_n \end{cases} \quad (1)$$

By rewriting the system in matrix form $AH = B$, with

$$A = \begin{bmatrix} x_1 & y_1 & 1 \\ \dots & \dots & 1 \\ x_n & y_n & 1 \end{bmatrix} \quad \text{and} \quad B = \begin{Bmatrix} z_1 \\ \dots \\ z_n \end{Bmatrix} \quad (2)$$

The coefficient vector $H = \{a_0\ a_1\ a_2\}^T$ can be obtained as $H = A^+B$, where $A^+ = (A^TA)^{-1}A^T$ is the pseudo-inverse matrix of A that minimizes the squared sum of errors. The z-coordinate of the robot tool is then automatically updated via software to paint on the approximating surface.

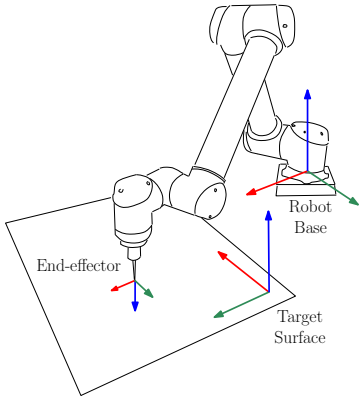


Figure 8. Robotic system reference frames.

Furthermore, the position of the paper, of the color cups and of the brush change repository is measured with respect to the robot reference frame, positioned in the central point of the robot base.

In this work, we adopted non-diluted black Indian ink provided to the robot in small plastic containers. After the dipping, the robot wipes off excessive paint from the brush on the edge of the ink cup.

A characterization of the brush strokes has been performed in order to estimate the influence of the z-coordinate of the brush (Figure 9), the traveled path and the robot speed on the stroke parameters: the black intensity and the thickness. This enables us to control the thickness of the strokes, to determine the traveled length after which the brush has to be dipped in the ink and to ensure a smooth deposition of the paint along the strokes.

To highlight the differences between watercolor and ink painting, some examples of stroke characterization are reported in Figures 10 and 11 for a watercolor and an ink stroke, respectively. It can be seen that in the ink case the black intensity is much more constant along the stroke, with respect to the watercolor one, where the dilution of pigments leads to transparency effects.

An experimental characterization has been performed and a set of linear strokes, each 250 mm long, has been painted by the robot while varying the z-coordinate from $z_1 = 0$ mm to $z_3 = -2$ mm with $\Delta z = 1$ mm, and the maximum robot speed, from $v_1 = 0.1$ m/s to $v_3 = 0.3$ m/s with $\Delta v = 0.1$ m/s. The maximum robot acceleration has been varied accordingly in order to maintain the same length of the constant velocity phase in the trapezoidal speed profile. Each stroke has been repeated three times and the results have been digitized and processed in order to obtain mean and standard deviation

values. Figures 12 and 13 report these results, where the effects of the brush z-coordinate and robot speed on black intensity and stroke thickness can be seen, respectively.

From Figure 12, it can be seen that the black intensity is almost constant along the strokes and it is not affected by the z-coordinate or by the robot speed. The black intensity shows a limited increase in the last part of the stroke. This could be due to the trapezoidal motion profile, which is characterized by a deceleration phase that can affect the paint deposition process. In this manner, a slower speed can cause an increase in the paint deposition per linear distance. On the other hand, a high painting speed can cause undesirable effects in the starting and end points of the stroke, due to a rough landing and detachment of the brush on the target surface, as well as the dry brush effect [1].

Conversely, the stroke thickness shows a decreasing trend along the stroke (Figure 13). The z-coordinate of the brush affects the stroke thickness, since the more the bristles are pressed against the paper, the larger the strokes are painted. The robot speed seems not to be relevant in the tested range of values. The values of the mean thickness along the whole strokes have been computed as well and the results are reported as box-plots, in Figure 14.

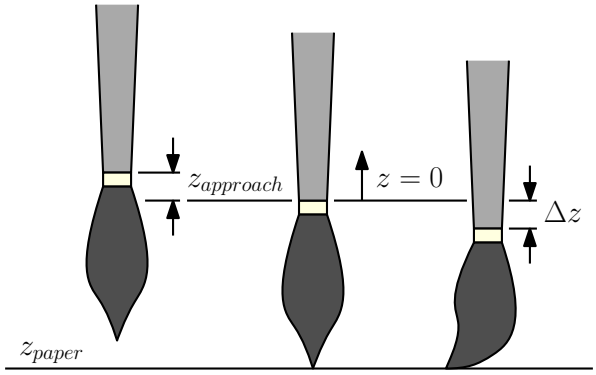
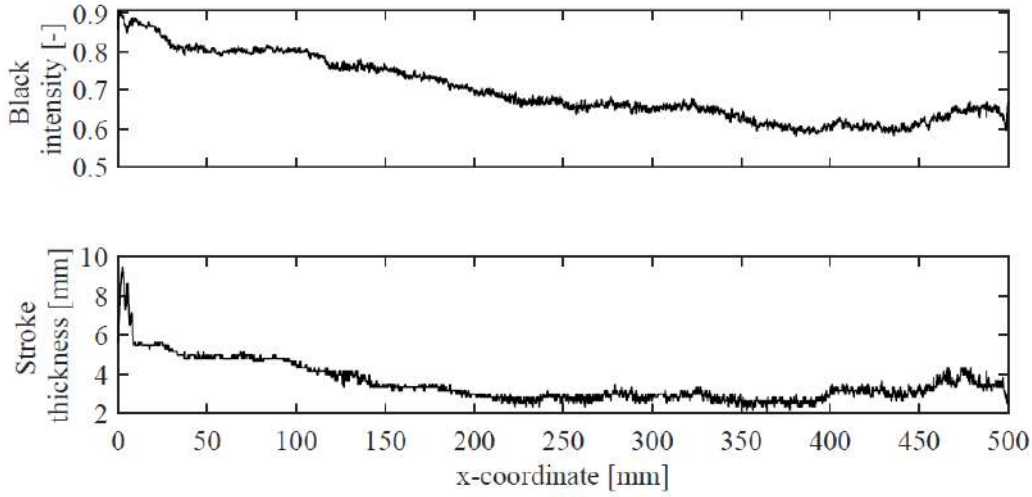


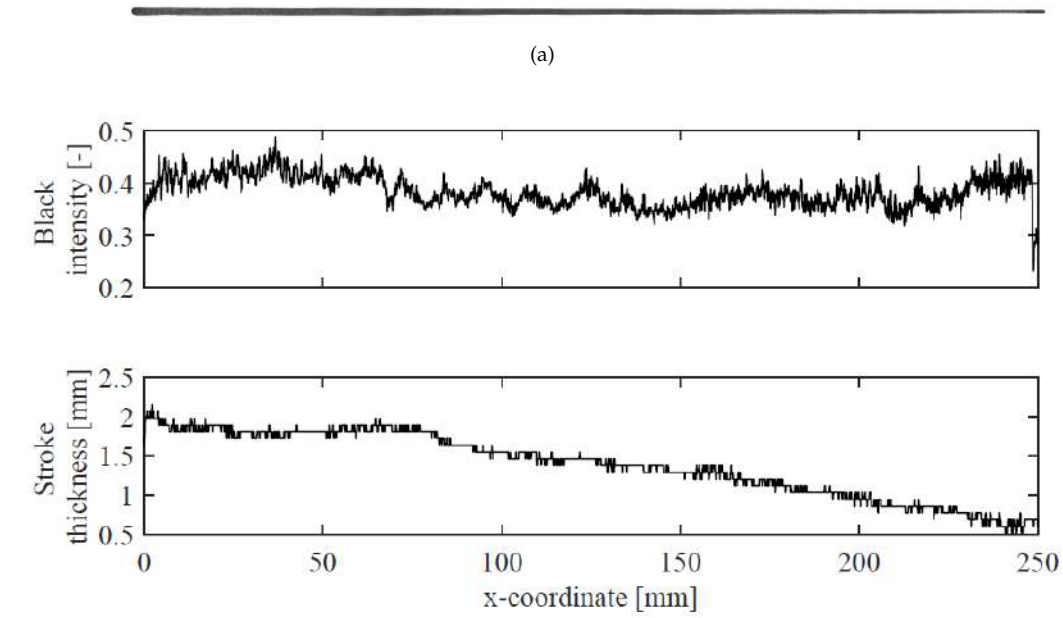
Figure 9. Positioning of the brush on the painting surface.

(a)



(b)

Figure 10. Example of *watercolor* stroke characterization: original photo (**a**), graphs of intensity and thickness (**b**).



(a) (b)

Figure 11. Example of *non-diluted ink* stroke characterization: original photo (**a**), graphs of intensity and thickness (**b**).

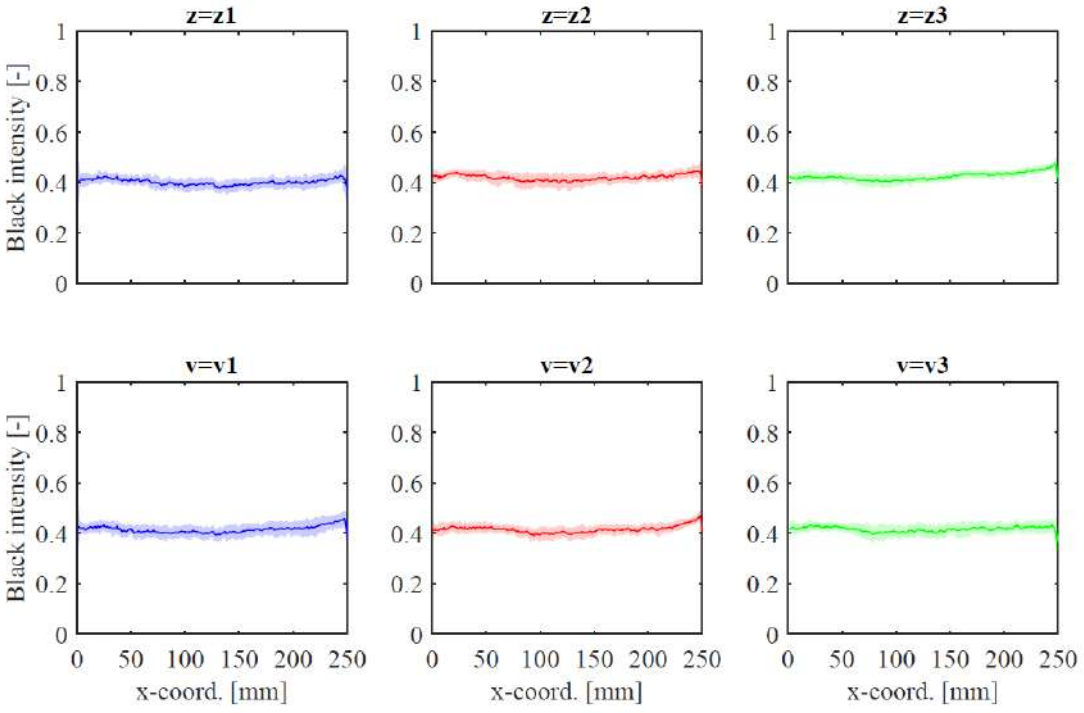


Figure 12. Effects of brush *z*-coordinate and robot speed on black intensity, mean and standard deviation values.

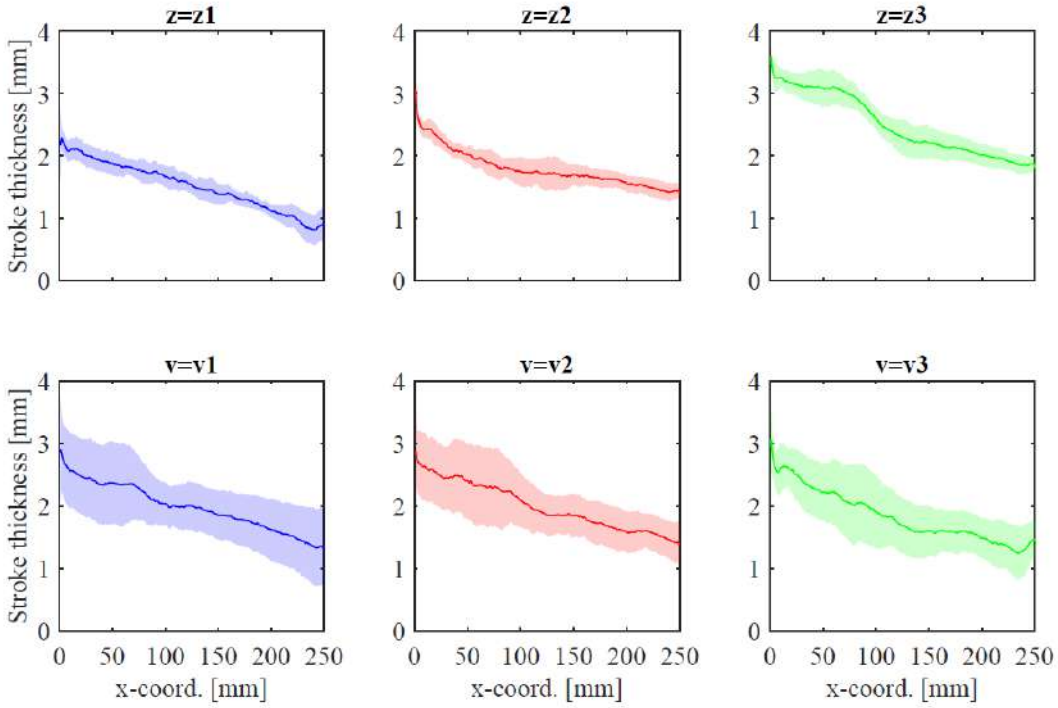


Figure 13. Effects of brush *z*-coordinate and robot speed on stroke thickness, mean and standard deviation values.

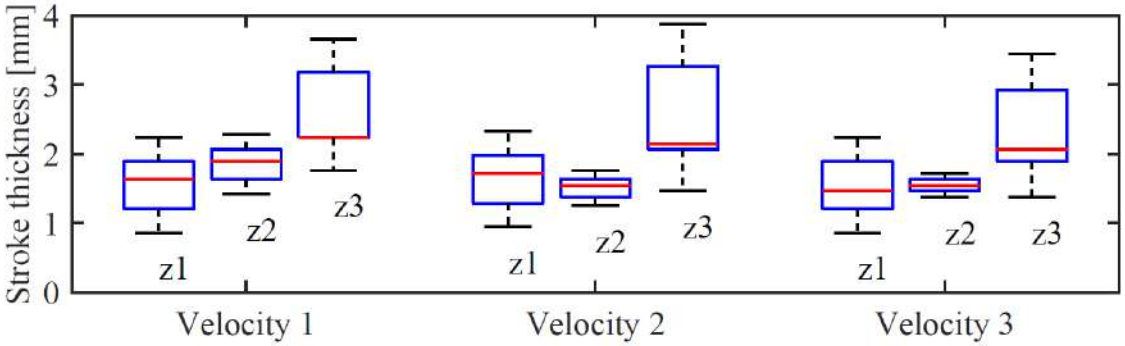


Figure 14. Box-plots of stroke thickness distribution: effects of robot speed and brush *z*-coordinate.

Busker Robot takes from one to three hours to complete one artwork, depending on the complexity of the subject, the number of layers and the speed of the robot. The total painting time can be optimized by finding the best compromise between quality of the strokes and speed of the robot, in order to minimize the brush travel time, while, at the same time, ensuring a smooth deposition of the paint along the strokes. A frame sequence of Busker Robot painting Hydrodynamic Power Station is reported in Figure 15.

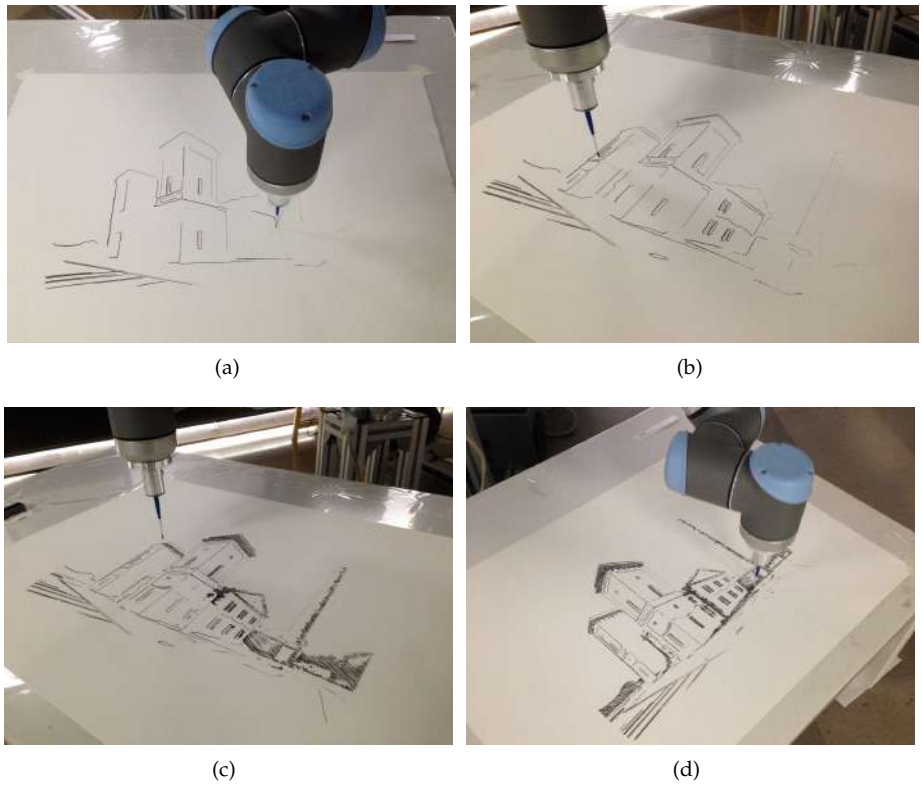


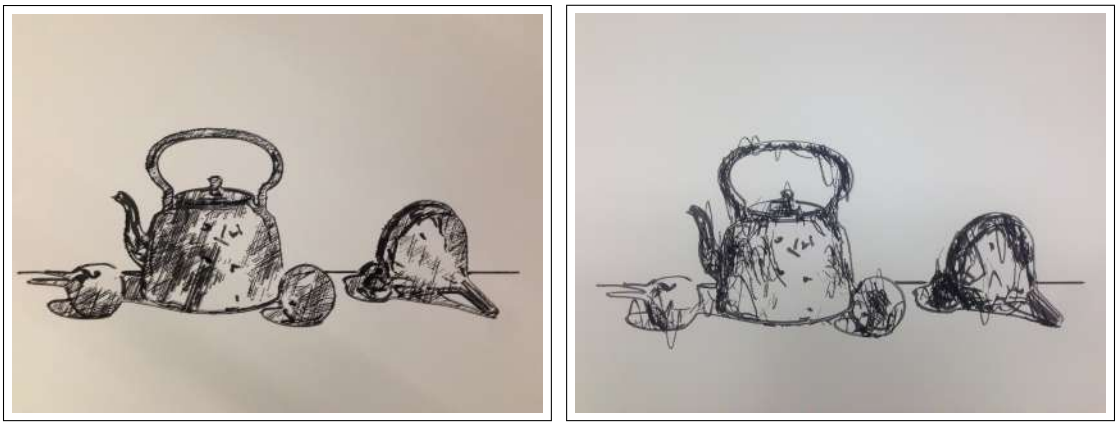
Figure 15. Busker Robot painting "Hydrodynamic Power Station". (**a**)–(**d**) show a frame sequence of the painting process in progress.

6. Results

Experimental results of the robotic painting system are reported in Figures 16 and 17, where the resulting artworks "Hydrodynamic Power Station" and "Still Life with Tea Kettle" are shown. The second subject has been realized with both the *cross-hatching* and the Random Splines techniques, to have a visual comparison between the results of the two NPR algorithms (Figure 17a,b).

The three artworks have been realized with the parameters reported in Tables 1, 2 and 4, respectively. While the *cross-hatching* reproduces the effects of the light and shades in the image with a clean and ordered style, the Random Splines approach creates a configuration of lines that are closer to a sketch drawn by a human. The lines exceed the contours and the result is not accurate and tidy as in the *cross-hatching* case.

Figure 16. Hydrodynamic Power Station, final artwork with *cross-hatching*.



(a) (b)

Figure 17. Still Life with Tea Kettle, final artworks. (**a**) *Cross-hatching*. (**b**) Random Splines.

A precise calibration of the brush strokes is very challenging and, therefore, it can be seen that not all the strokes appear with the same thickness and color intensity. The interaction between brush bristles and paper, the ink deposition, the ripples of the paper and the amount of paint in the brush are not easy to be modeled and controlled, especially without feedback, and a lot of work can be done to improve the results. Nevertheless, the artworks produced by our automatic painting system are unique and can be considered an example of integration between robotics, image processing and art.

Even though a quantitative analysis of the artworks is not possible, since the qualitative comparison with the original image and the aesthetic appreciation is subjective, Busker Robot obtained a good feedback and has been considered of interest by the press and the public at several exhibitions.

7. Conclusions

In this paper we presented non-photorealistic rendering techniques for artistic robotic painting, that have been implemented for the processing of digital images into real artworks with Busker Robot, our robotic painting system. The system is capable of reproducing an image that has been previously processed through the application of artistic rendering algorithm. In particular, we presented two novel techniques, the *cross-hatching* and the Random Splines, that are used to process the light and shade of a painting. Experimental results shown the feasibility of the proposed approach and system architecture. In particular, two sample images have been tested with the new algorithms and the robotic system, with interesting results. Future works will include a further brush stroke characterization to further improve the quality and the repeatability of the lines. A visual feedback can as well be introduced to give the robotic system a consciousness of the current status of the painting, that can be used to correct possible errors or place the next stroke.

Author Contributions: Investigation, methodology and software, P.G. and L.S.; data analysis and processing, L.S. and S.S.; writing—original draft preparation, L.S.; writing—review and editing, L.S., S.S., P.G. and A.G.; supervision, P.G. and A.G.

Funding: This research was partially funded by Fondo Ricerca Ateneo, FRA 2015 (Internal Fund, University of Trieste), https://www.units.it/intra/ricerca/fra.

Acknowledgments: The authors would like to thank M. A. Grimaldi for his help in the development of the Random Splines algorithm.

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

References

1. Scalera, L.; Seriani, S.; Gasparetto, A.; Gallina, P. Watercolour Robotic Painting: A Novel Automatic System for Artistic Rendering. *J. Intell. Rob. Syst.* **2018**, doi:10.1007/s10846-018-0937-y. [CrossRef]

2. Scalera, L.; Seriani, S.; Gasparetto, A.; Gallina, P. Busker robot: A robotic painting system for rendering images into watercolour artworks. *Mech. Mach. Sci.* **2019**, *66*, 1–8, doi:10.1007/978-3-030-00365-4_1. [CrossRef]
3. Robotics, Festival of Art and Robotics, Trieste 2018. Available online: https://robotics.gruppo78.it/?lang=en (accessed on 18 December 2018).
4. International Robotic Art Competition (RobotArt). Available online: http://robotart.org/archives/2018/artworks/ (accessed on 18 December 2018).
5. Gasparetto, A.; Scalera, L. From the Unimate to the Delta robot: the early decades of Industrial Robotics. In Proceedings of the 6th IFToMM International Symposium on History of Machines and Mechanisms, HMM 2018, Beijing, China, 26–28 September 2018.
6. Pierre Jaquet Droz. Available online: https://history-computer.com/Dreamers/Jaquet-Droz.html (accessed on 10 January 2019).
7. Jean Tinguely. Available online: https://en.wikipedia.org/wiki/Jean_Tinguely/ (accessed on 18 December 2018).
8. Cohen, H. The further exploits of AARON, painter. *Stanford Hum. Rev.* **1995**, *4*, 141–158.
9. Calinon, S.; Epiney, J.; Billard, A. A humanoid robot drawing human portraits. In Proceedings of the IEEE-RAS International Conference on Humanoid Robots, Tsukuba, Japan, 5–7 December 2005; pp. 161–166.
10. Jean-Pierre, G.; Saïd, Z. The artist robot: A robot drawing like a human artist. In Proceedings of the 2012 IEEE International Conference on Industrial Technology, Athens, Greece, 19–21 March 2012; pp. 486–491.
11. Aguilar, C.; Lipson, H. A robotic system for interpreting images into painted artwork. In Proceedings of the 11th Generative Art Conference, Milan, Italy, 16–18 December 2008; Volume 11.
12. Tresset, P.A.; Leymarie, F. Sketches by Paul the robot. In Proceedings of the 8th Annual Symp. on Computational Aesthetics in Graphics, Visualization, and Imaging, Eurographics Association, Annecy, France, 4–6 June 2012; pp. 17–24.
13. Tresset, P.; Leymarie, F.F. Portrait drawing by Paul the robot. *Comput. Graphics* **2013**, *37*, 348–363. [CrossRef]
14. Lindemeier, T.; Metzner, J.; Pollak, L.; Deussen, O. Hardware-Based Non-Photorealistic Rendering Using a Painting Robot. *Comput. Graphics Forum* **2015**, *34*, 311–323. [CrossRef]
15. Lindemeier, T.; Spicker, M.; Deussen, O. Artistic composition for painterly rendering. In Proceedings of the 21th International Symposium on Vision, Modeling and Visualization (VMV 2016), Bayreuth, Germany, 10–12 October 2016.
16. Gülzow, J.; Grayver, L.; Deussen, O. Self-Improving Robotic Brushstroke Replication. *Arts* **2018**, *7*, 84. [CrossRef]
17. Dong, X.; Li, W.; Xin, N.; Zhang, L.; Lu, Y. Stylized Portrait Generation and Intelligent Drawing of Portrait Rendering Robot. *DEStech Trans. Eng. Techno. Res.* **2018**. [CrossRef]
18. Berio, D.; Calinon, S.; Leymarie, F.F. Learning dynamic graffiti strokes with a compliant robot. In Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Korea, 9–14 October 2016; pp. 3981–3986.
19. Song, D.; Lee, T.; Kim, Y.J.; Sohn, S.; Kim, Y.J. Artistic Pen Drawing on an Arbitrary Surface using an Impedance-controlled Robot. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 21–25 May 2018.
20. Luo, R.C.; Liu, Y.J. Robot Artist Performs Cartoon Style Facial Portrait Painting. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 7683–7688.
21. Trigatti, G.; Boscariol, P.; Scalera, L.; Pillan, D.; Gasparetto, A. A new path-constrained trajectory planning strategy for spray painting robots-rev.1. *Int. J. Adv. Manuf. Technol.* **2018**, *98*, 2287–2296, doi:10.1007/s00170-018-2382-2. [CrossRef]
22. Gasparetto, A.; Vidoni, R.; Saccavini, E.; Pillan, D. Optimal path planning for painting robots. In Proceedings of the ASME 2010 10th Biennial Conference on Engineering Systems Design and Analysis, Istanbul, Turkey, 12–14 July 2010; pp. 601–608.
23. Trigatti, G.; Boscariol, P.; Scalera, L.; Pillan, D.; Gasparetto, A. A look-ahead trajectory planning algorithm for spray painting robots with non-spherical wrists. *Mech. Mach. Sci.* **2019**, *66*, 235–242, doi:10.1007/978-3-030-00365-4_28. [CrossRef]

24. Viola Ago. Robotic airbrush painting. Available online: http://violaago.com/robotic-airbrush-painting/ (accessed on 18 December 2018).
25. Scalera, L.; Mazzon, E.; Gallina, P.; Gasparetto, A. Airbrush Robotic Painting System: Experimental Validation of a Colour Spray Model. In Proceedings of the International Conference on Robotics in Alpe-Adria Danube Region, Turin, Italy, 21–23 June 2017; pp. 549–556.
26. Vempati, A.S.; Kamel, M.; Stilinovic, N.; Zhang, Q.; Reusser, D.; Sa, I.; Nieto, J.; Siegwart, R.; Beardsley, P. PaintCopter: An Autonomous UAV for Spray Painting on Three-Dimensional Surfaces. *IEEE Rob. Autom. Lett.* **2018**, *3*, 2862–2869. [CrossRef]
27. Moura, L. A new kind of art: The robotic action painter. Available online: http://generativeart.com/on/cic/papersGA2007/16.pdf (accessed on 8 February 2019).
28. Shih, C.L.; Lin, L.C. Trajectory Planning and Tracking Control of a Differential-Drive Mobile Robot in a Picture Drawing Application. *Robotics* **2017**, *6*, 17. [CrossRef]
29. Ticini, L.F.; Rachman, L.; Pelletier, J.; Dubal, S. Enhancing aesthetic appreciation by priming canvases with actions that match the artist's painting style. *Front. Hum. Neurosci.* **2014**, *8*, 391. [PubMed]
30. Chatterjee, A.; Vartanian, O. Neuroscience of aesthetics. *Ann. N.Y. Acad. Sci.* **2016**, *1369*, 172–194. [CrossRef] [PubMed]
31. Seriani, S.; Cortellessa, A.; Belfio, S.; Sortino, M.; Totis, G.; Gallina, P. Automatic path-planning algorithm for realistic decorative robotic painting. *Autom. Constr.* **2015**, *56*, 67–75. [CrossRef]
32. Canny, J. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **1986**, *6*, 679–698. [CrossRef]
33. Winkenbach, G.; Salesin, D.H. Rendering parametric surfaces in pen and ink. In Proceedings of the 23rd annual conference on Computer graphics and interactive techniques, New Orleans, LA, USA, 4–9 August 1996; pp. 469–476.
34. Praun, E.; Hoppe, H.; Webb, M.; Finkelstein, A. Real-time hatching. In Proceedings of the 28th annual conference on Computer graphics and interactive techniques, Los Angeles, CA, USA, 12–17 August 2001; p. 581.
35. Zander, J.; Isenberg, T.; Schlechtweg, S.; Strothotte, T. High quality hatching. *Comput. Graphics Forum* **2004**, *23*, 421–430. [CrossRef]
36. Lewis, J.P.; Fong, N.; XueXiang, X.; Soon, S.H.; Feng, T. More optimal strokes for NPR sketching. In Proceedings of the 3rd International Conference on Computer Graphics and Interactive Techniques in Australasia and South East Asia, Dunedin, New Zealand, 29 November–2 December 2005; pp. 47–50.
37. Universal Robot UR10 Technical Details. Available online: https://www.universal-robots.com/media/1801323/eng_199901_ur10_tech_spec_web_a4.pdf (accessed on 10 January 2019).