Journal of Intelligent & Robotic Systems (2019) 95:871–886
https://doi.org/10.1007/s10846-018-0937-y

Watercolour Robotic Painting: a Novel Automatic System for Artistic Rendering

Lorenzo Scalera[1] · Stefano Seriani[2] · Alessandro Gasparetto[1] · Paolo Gallina[2]

Received: 10 January 2018 / Accepted: 24 September 2018 / Published online: 28 September 2018

Abstract

In this paper, we present a novel robotic system that produces watercolour paintings by means of a 6-degree-of-freedom collaborative robot. After an analysis of traditional watercolour, different non-photorealistic rendering techniques are applied in order to elaborate digital images into artworks. Several algorithms, aimed at processing both the backgrounds and the details, are implemented. Then, the resulting rendering is converted into a sequence of trajectories that the robot reproduces on paper. During the process, the artist controlling the system can change both the algorithm parameters and the hardware variables (e.g. brush type, colour dilution, etc.) in order to obtain a different artistic rendering. The challenge is indeed not to faithfully reproduce an image but to introduce a personal and original contribution to the artwork. The robotic painting system described in this paper was named "Busker Robot" and it is the first automatic system that uses the watercolour technique for artistic rendering. It was installed at the "Algorithmic Arts and Robotics" exhibition during the international event "Trieste Next" (Trieste, Italy, September 2017) and won an Honorable Mention at the 2018 International Robotic Art Competition (RobotArt).

Keywords Robotic painting · Watercolour · Non-photorealistic rendering · Artistic rendering · Collaborative robotics

1 Introduction

Artistic robotic painting is a process that raises interest and fascination. In recent years, artists and engineers have increasingly brought industrial robots from the factories to public exhibition spaces. In this environment, they have been employed as both physical and intelligent media to develop novel artistic expressions, in which the border between machine and artist is not always well defined [1]. At the same time, even if art and technology are continuing to approach and merge themselves, traditional mediums, such as painting, will always have an implicit power associated with the history of art [2].

One of the first artists that used a painting machines to create complex and random patterns was Jean Tinguely (1925-1991) [3]. In the mid-1950s he produced a series of generative works titled Métamatics: machines that produce artworks. In the '70s Harold Cohen (1928-2016) created AARON, a plotter designed to produce artistic images, considered as the most important painting machine in contemporary art [4]. In most recent years, several examples of artistically skilled robots can be found. In 2002, a robot able to perform a portrait like a human painter was developed in Robotlab and exhibited for the first time at the Center for Art and Media Karlsruhe (ZKM) in Germany [5]. Other examples are given by the work of Calinon et al., who developed in 2005 a humanoid robot drawing human portraits by means of a 4-degree-of-freedom (DOF) robotic arm holding a quill pen [6], and by the Artist Robot, a robot installed in 2006 in the Futuroscope Park in France drawing portraits of visitors like an human artist [7]. Furthermore, Yao et al. developed a Chinese painting robot able to draw bamboos leaves by means of a brush [8], whereas Aguilar et al. presented an articulated painting arm that interprets images into painted artwork [2]. During

✉ Lorenzo Scalera
scalera.lorenzo@spes.uniud.it

Stefano Seriani
sseriani@units.it

Alessandro Gasparetto
alessandro.gasparetto@uniud.it

Paolo Gallina
pgallina@units.it

[1] University of Udine, via delle Scienze 206, 33100, Udine, Italy

[2] University of Trieste, via Valerio 10, 34127, Trieste, Italy

the painting process, the artist controlling the system is able to influence the outcome by managing both the hardware (i.e. palette and brush type) as well as the algorithmic parameters. In 2009 Kudoh et al. studied and developed a painting robot with multi-fingered hands and stereo vision, that was able to obtain a 3D model of an object, compose a picture and paint it on the canvas [9]. In the same year, Lu et al. developed a robotic manipulator that uses visual feedback to determine strokes position and orientation [10].

In more recent years, Sun et al. studied a calligraphy robot which consists of a 6-DOF robot arm that accurately reproduces Chinese Calligraphic characters [11]. Furthermore, a robotic manipulator was built in 2013 by Kim et al., who proposed a system capable of detection and extraction of brightness in images and to draw it by repeated reproducing overlapped lines [12].

Nowadays, a large number of artists uses manipulators or robots to create artistic graphics and paintings. Tresset and Fol Leymarie created Paul, a robotic installation able to produce sketches of people using visual feedback to guide the drawing task [13]. Benedettelli built in 2013 Legonardo, a LEGO portrayed robot, inspired by the work of 18th Century Swiss watch and automata makers Pierre Jacquet-Droz and Henri Maillardet [14]. In 2015 Jain et al. developed a force-controlled portrait drawing robot [15]; in the next year Luo et al. presented a 7-DOF robot capable of painting colourful pictures equipped with a visual control system and a three-finger gripper to grasp the brushes [16], whereas Berio et al. proposed a friendly Baxter robot skilled at drawing graffiti strokes [17]. Other interesting examples of robotic applications to art can be found in the works of Scalera et al. [18] and Ago [19], where airbrush painting techniques by means of industrial manipulators are introduced. As far as brushes painting is concerned, one of the most impressive examples of artistic robot, capable of reproducing non-photorealistic images, is given by the machine developed at University of Konstanz, Germany, by Deussen et al. [20, 21]. More recently, another example of robotic painting can be found in [22], where a semi-autonomous, impedance-controlled collaborative robot capable of creating pen art on an arbitrary surface is presented.

At the present time, no examples of robotic systems painting using watercolour techniques can be found in Literature. In this paper, we propose different non-photorealistic techniques for watercolour robotic painting. In particular, several algorithms have been implemented in order to elaborate digital images into artistic artworks. Backgrounds and large areas are extracted from the input images and converted into a series of paths that can be reproduced by a 6-axis UR10 collaborative robot. Our challenge is not to reproduce the input figure in a faithful manner but to introduce a novel and original artistic expression. Other than in arts, applications of the robotic system in industrial manufacturing processes could be found in the painting of ceramics and inlaid woods, such as design objects and furniture.

In our work, robot motion is extremely important and the techniques that we have implemented, in combination with watercolour painting, have been chosen and developed in order to enhance the aesthetic appreciation of the public. In particular, to show a high motor activity and the gestures of the artist is one of the targets of our artistic expression. In fact, action priming, when congruent with the artist's painting style, better contributes to the appreciation of the artworks, as demonstrated by Ticini et al. [23].

Our robotic system, called Busker Robot (Fig. 1), was installed at the "Algorithmic Arts and Robotics" exposition during Trieste Next (Trieste, Italy, September 2017) [24], with a great appreciation by the public. Recently, our robot participated at the 2018 International Robotic Art Competition (RobotArt) and won an Honorable Mention with more than 3500 on-line votes [25]. The contest, at which took part 19 teams from all over the world with more than 100 robot-created artworks, aims to produce something visually beautiful with robotic systems and physical brushes, combining image processing and path planning but also the artistic sensibility of the teams.

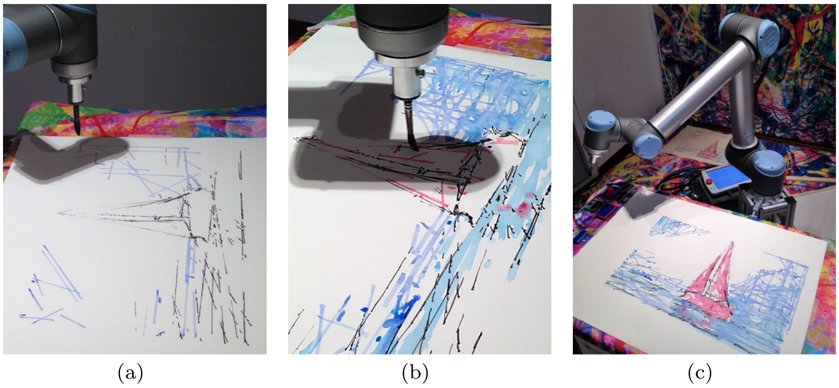

(a) (b) (c)

Fig. 1 Busker Robot painting Sailboat at the "Algorithmic Arts and Robotics" exhibition during "Trieste Next", September 2017

The paper is organized as follows: Section 2 provides a description of watercolour painting and in Section 3 the whole architecture of the automatic system is presented. In Section 4 the algorithms for artistic rendering are described, whereas in Section 5 the implementation of the non-photorealistic techniques on Busker Robot is presented. In particular, an analysis of brush strokes and an automatic brush change system are described. Finally, in Section 6 the experimental results are shown and Section 7 reports the conclusions of this work.

2 Watercolour and Gouache Painting

Watercolour, also known as *aquarelle* from the Latin, is an extremely ancient painting technique, in which finely ground pigments are suspended in a water-based solution. The origins of watercolour date perhaps to palaeolithic cave paintings, but the technique has been adopted for manuscripts illustration since Egyptian art, after the discovery of papyrus, as well in the ancient Chinese and Japanese tradition. Modern watercolour painting has its origins in the Northern Renaissance: Albrecht Dürer (1471-1528) with his magnificent landscapes is considered one of the earliest exponents of this technique. Today, the traditional watercolour refers to the English school of landscape painters, when artists such as Thomas Girtin (1775-1802) and J.M.W. Turner (1775-1859) explored both the expressive nature and technical aspects of the medium to create more painterly, expressive and even turbulent effects [26, 27].

In the present scientific Literature, several works can be found, dedicated at characterizing the most important watercolour effects and simulate them automatically. Curtis et al. described the various effects of the technique and presented a model that creates convincing watercolour renderings [28]. Lum and Ma presented a watercolour inspired method for the rendering of surfaces using several layers of semitransparent paint [29]. Burgess et al. created a similar methodology for rendering images in the style of watercolour [30]. Van et al. applied a three-layers paper model to digital watercolour painting for the simulation of pigment and water mechanisms [31]. Furthermore, Bousseau et al. presented an interactive watercolour rendering technique that recreates the specific visual effects of lavis aquarelle allowing the processing of both images and 3D models [32]. Other examples of watercolour rendering for illustration and animation are given by Luft and Deussen [33] and, in recent years, by Montesdeoca et al., who implemented characteristic effects of traditional watercolour, such as edge-darkening, paper distortion and granulation (see Section 2.1), with impressive results for 3D scenes and animations [34, 35].

In order to provide the reader with a comprehensive overview on the watercolour technique, in the rest of this section we discuss and survey some of the most important features of watercolour physical nature and its related effects.

2.1 Properties of Watercolour

Differently from the relatively opaque paints often used in oil and acrylic painting, watercolour artworks are realized with the application to paper of a semi-transparent suspension of pigment particles in water. A *pigment* is a solid material in the form of small grains ranging about 0.05 to 0.5 μm. Density and dilution of the pigments in water determine their capacity of adhere and penetrate into the medium or being suspended in water. The other ingredient of watercolour is the paper, typically made from linen or cotton rags pounded into small fibres. Different textures of paper can be found: *hot-pressed* paper has a smooth and hard surface and it is preferred for detailed work, *cold-pressed* one is used for both detailed work and smooth washes and, finally, *rough* paper, ideal for washes. The final appearance of watercolour derives from the interaction between watered paint and paper, a complex phenomena that can lead to several effects. The main properties are listed below:

Transparency: the transparent nature of watercolour allows to build tonal values through layers of colour. Several consecutive washes of watered paint can be applied on the paper, each over others that have already dried. This process, known as *glazing*, allows to achieve a very clear and even effect [26, 28–30].

Mixing: colours belonging to different layers can be mixed on paper in order to change the paint temperature as well as to produce light or shadow effects in the painting. Paint mixing can also be performed on the colour palette to achieve additional tonalities.

Lightness: light colours are created using less paint and often lighting effects are achieved avoiding painting in the highlight regions, unlike in oils, where white paint is usually added for lightness effects [29].

Diluting: the amount of water mixed with the paint determines the colour intensity as well as affects its transparency. Watercolour flows on the paper when diluted with water, generating uncontrollable effects, such as:

- *Backruns*: branching shapes with several darkened edges appear when a puddle of water spreads back into a damp region of paint (see Fig. 2a) [28].
- *Granulation*: grainy texture appears when pigments separate from the binder and settle into the "valleys" of the paper (see Fig. 2b) [28].
- *Flow patterns*: striations occurs when brush strokes are spread on a wet surface [28].
- *Paper ripples*: the dilution with water can make the paper ripple and distort, a phenomena that has to be taken into account in automatic painting.

Resolvability: watercolour can be easily redissolved when dry, both on the colour palette and on paper with a simple addition of water [26].

Brush effects:

- *Wet-in-wet* and *wet-in-dry*: the two basis brushing techniques consist in the application of a brush imbued with paint to paper that is already saturated with water (wet-in-wet) or to dry paper (wet-in-dry) [28].
- *Dry brush*: irregular gaps and edges can be visible when a nearly-dry brush applies paint only to the raised areas of a rough paper (see Fig. 2c) [28, 32].
- *Edge-darkening*: the gradual pigment accumulation towards the edges of painted areas (see Fig. 2d) [28, 31, 32, 35], due to the Marangoni flow effect on surface tension [36].

2.2 Gouache Painting

Finally, the concept of gouache painting is briefly recalled. The French term *gouache* means literally "muddy pool" and refers to a *body colour* that, differently from watercolour, is opaque [26]. Gouache, often made in *tempera* paint, can be seen as a final covering phase of watercolour painting, realized to empathize the most important details and contours. In fact, since watercolour is meant for large areas and backgrounds, thin and intensive strokes can be applied to mark dark edges (usually realised in black hue), as well as to emphasise highlights (in white or bright paint).

2.3 Automation of Watercolour Painting

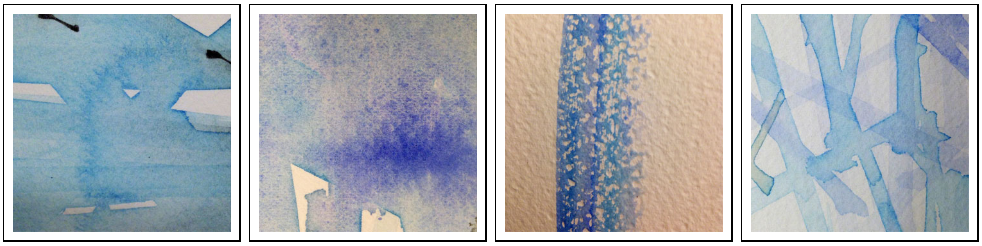
Watercolour painting presents several complexities and technological challenges that have to be solved for a proper automation of the process. Differently from other forms of paint, water effects in watercolour are extremely difficult to predict and control, as explained in Section 2.1, and, for these reasons, the choice of paper, colour dilution and brushes plays an important role in the painting task. A stroke and brush characterization, presented in Section 6, is needed for a proper understanding of the automatic painting. Moreover, other variables can influence the outcome: above all the ripple of the wet paper due to the water flow. As a consequence, the z-coordinate of the paper changes with respect to the robot reference frame (see Fig. 11). Especially in the final gouache phase, where the most important details and contours are painted, this inconvenient could badly affect the results if a feedback from the environment is not implemented. A practical solution of the problem is to perform, differently from the traditional technique, the gouache painting before the watercolour, when the paper is still dry. In this manner, the subsequent washes of transparent watercolour and consequent ripples in the paper do not affect the finishing of details.

3 Architecture of the System

In this section an overview on the architecture of the robotic painting system is presented. It is composed of both software (e.g. the algorithms for artistic rendering) and hardware components (e.g. the robot and its painting set-up).

The image processing algorithms have been implemented in Matlab$^{®}$, in the form of Graphical User Interface (GUI)

Fig. 2 Examples of watercolour effects


(a) Backruns. (b) Granulation. (c) Dry brush. (d) Edge-darkening.

on a laptop running Windows 10 64bit operating system. The original image, which consists of a digital image in *JPG* format, is imported into the software as a pixel matrix, which is processed by the artistic rendering algorithms to extract the backgrounds and details as explained in detail in the following sections. The resulting sequences of points are fed to the path planning module, which produces a sequence of trajectories to be completed by the robot, including the instructions to steer it during the paint refill, the brush change and draining. The list of commands is written in Matlab by adopting the UR Script Programming Language and stored in a *script* file that is finally deployed on the controller of the UR10 to be compiled and executed in real-time. Fig. 3 reports an overview on the system.

4 Algorithms for Artistic Rendering

In this section, the *non-photorealistic rendering* techniques (NPR), that have been implemented for the elaboration of digital images into artistic results, are described. In the recent years, several techniques have been proposed in Literature; in particular, we refer to *image-based artistic rendering* (IB-AR), which focuses on artistic stylization of a two-dimensional digital image [37]. It has to be underlined that in this paper we implement the processing algorithms only on images which have been converted in grey-scale.

The algorithms described in the following have been adopted for the processing of backgrounds and large areas, painted in watercolour, and of contours and thin details, realized by means of gouache technique. In order to correctly represent different regions of a scene (e.g. sky and landscape) we need to segment the input image, an operation that is typically impossible to fully automate, as explained in the works of Zeng et al. [38] and Lindemeier et al. [21]. In this preliminary study, a segmentation procedure, based on both manual areas selection and on RGB colours identification has been applied.

After the image segmentation, each part can be further divided into foreground and background layers, by adopting the concept of grey-scale threshold, as implemented by Seriani et al. in an automatic path-planning algorithm for photorealistic grey-scale images [39]. In particular, given a specific threshold intensity I_T, each point Q_i of the input grey-scale image, with intensity $I(Q_i)$, is converted into a binary pixel with a value $I_{BW}(Q_i)$ equal to 1 when $I(Qi) \geq I_T$ and null otherwise. Grey-scale thresholding is the basis for the algorithms that will be explained in the following.

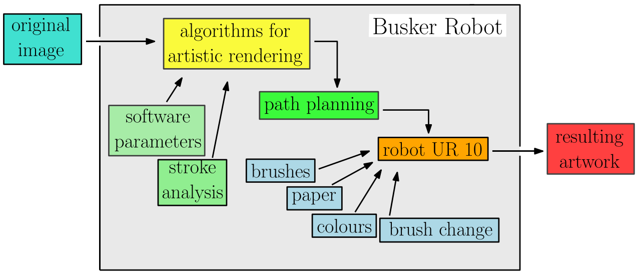


Fig. 3 Architecture of the system

4.1 Algorithms for Backgrounds and Large Areas

We analyse the algorithms for backgrounds and large areas: hatching, small angle random hatching and gradient-based random strokes. These techniques are used for the rendering of wide and uniformly coloured areas.

4.1.1 Hatching

The first technique that has been developed for a uniform filling of large areas is Hatching. It consists in closely spaced parallel lines that fully cover a layer of the image (or image part) defined by a grey level. For a more aesthetic outcome, the angular direction of the lines should be chosen in order to follow the subject shape or the light angle. Dark effects and shadows can also be obtained by combining more hatching layers with different angles and lines spacing, resulting in a cross-hatching effect. Figure 4 shows two examples of hatching filling, obtained with different thresholds, angles and distance between lines, starting from the image in Fig. 6a (Trieste Seafront).

4.1.2 Small Angle Random Hatching

A random perturbation of the starting and ending points of each single parallel line of the Hatching algorithm has been introduced in order to create a motion effect in the layer filling and make the resulting artworks look more natural. In this manner, Small Angle Random Hatching has been obtained. This feature introduce inaccuracy in the painting and allows to enhance the aesthetic appreciation of paint covering, since non perfectly parallel lines, similar to those painted by a human artist, introduce a motor activity in the painting and, therefore, are usually preferred by the observers [23]. Figure 5 reports the same random fillings as Fig. 4 with the additional introduction of the random perturbation.

4.1.3 Gradient-Based Random Strokes

The third technique that has been implemented for the covering of a large and well-determined image zone consists of Gradient-Based Random Strokes. After the definition of a grey-scale threshold, a pre-determined number of points are randomly generated inside the area. For each point

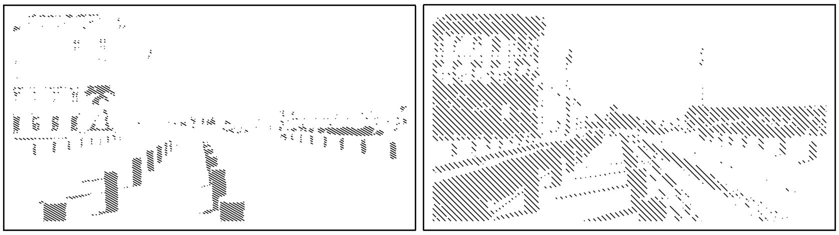


Fig. 4 Examples of hatching

(a) Threshold: 0.1; angle: 30°; lines dist.: 3 *px*.

(b) Threshold: 0.3; angle: 45°; lines dist.: 6 *px*.

$Q_i(x, y)$, the direction $\theta(x, y)$ of the gray-scale gradient $G = (G_x, G_y)$ is calculated as

$$\theta(x, y) = arctan\left(G_y, G_x\right) \tag{1}$$

where G_x and G_y are the horizontal and vertical directions of the gradient magnitude. Then, starting from each point, a stroke perpendicular to the gradient direction is generated. Each stroke crosses the interested area till the layer border is found, resulting in an original and artistic covering. Several different threshold can be chosen and the resulting lines overlapped for a more artistic outcome. Furthermore, the random generation of points ensures that each time the algorithm is run, the result is different, holding to a unique and unrepeatable configuration in its own style. Figure 6 reports the results of the application of random strokes algorithm on three sample images: Trieste Seafront, Sailboat and Amanda Seyfried (just simulated, not painted by the robot).

In both the Hatching and Random Strokes algorithms, the thickness of the lines does not indicate the real brush thickness but it is only representative of the resulting image. The real width of the strokes is, in this context, correlated with the z-axis position of brush with respect to the paper, as it will be explained in Section 5.1. The outcome could indeed completely change due to this parameter. Moreover, thanks to watercolour properties, in the case of a low distance between lines a random hatching filling could result in a uniform coverage.

4.2 Algorithms for Contours and Details

In this section, we present the algorithms for contours and details: Canny Edge Detector, Hough Transform and Skeletonization.

4.2.1 Canny Edge Detector

The first technique that we have adopted for the extraction of contours and details from an input image is J.F. Canny's Edge Detector [40]. This algorithm, presented in 1986, is considered one of the most efficient and widely used edge detection operators in computer vision systems and image processing. It consists of the following steps:

- Image smoothing with a low-pass Gaussian filter to remove the noise.

  The equation of a 2D Gaussian filter kernel is given by

  $$G(x, y) = \frac{1}{2\pi\sigma^2}e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{2}$$

  where the standard deviation σ determines the width of the kernel.
- Computation of the intensity gradients of the image.

  Since an edge in a image may point in whatsoever direction, the Canny algorithm uses four filters to detect horizontal, vertical and diagonal edges in the blurred image. The edge detection operator (such as Roberts, Prewitt, or Sobel) returns a value of the first derivative

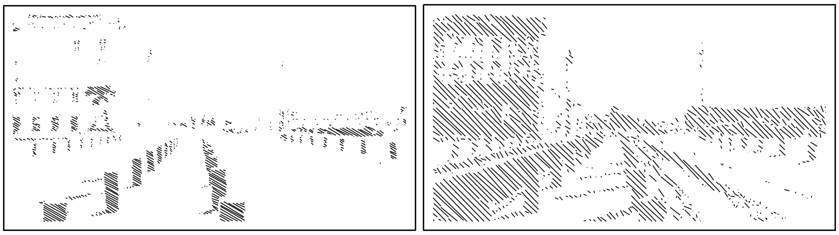
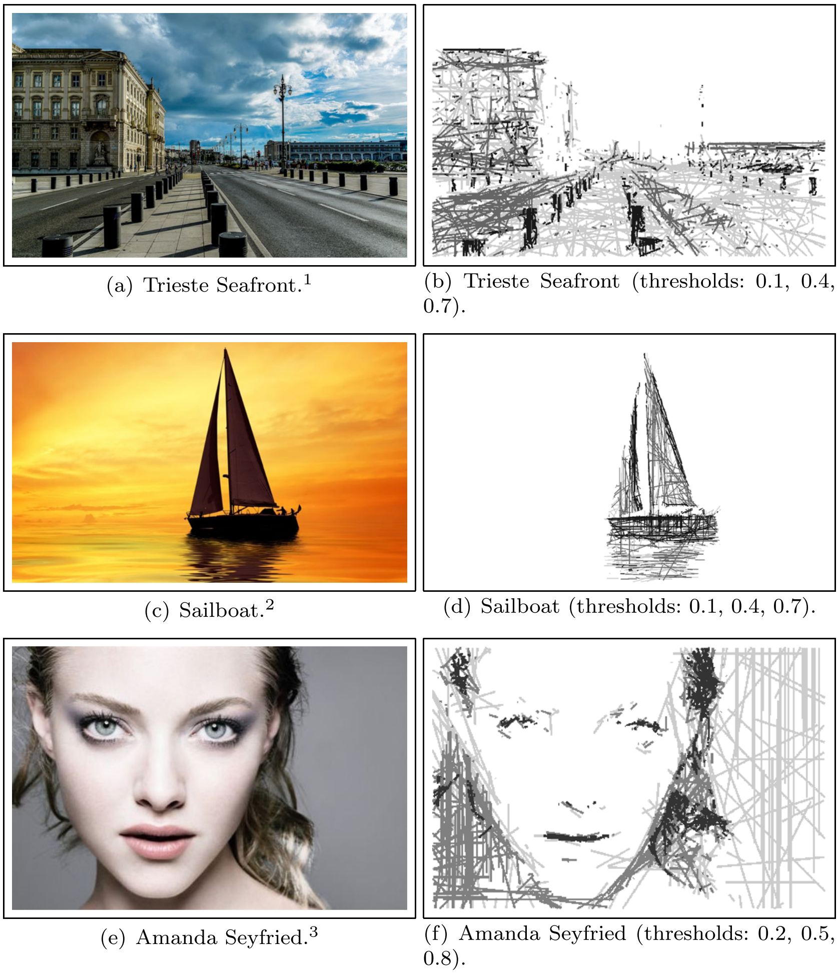


Fig. 5 Examples of small angle random hatching

(a) Threshold: 0.1; angle: 30°; lines dist.: 3 *px*.

(b) Threshold: 0.3; angle: 45°; lines dist.: 6 *px*.

Fig. 6 Examples of gradient-based random strokes



(a) Trieste Seafront.[1]

(b) Trieste Seafront (thresholds: 0.1, 0.4, 0.7).

(c) Sailboat.[2]

(d) Sailboat (thresholds: 0.1, 0.4, 0.7).

(e) Amanda Seyfried.[3]

(f) Amanda Seyfried (thresholds: 0.2, 0.5, 0.8).

in horizontal G_x and vertical direction G_y. The edge gradient magnitude can be determined as

$$G = \sqrt{G_x^2 + G_y^2} \tag{3}$$

and its direction θ from Eq. 1. The edge angle is then rounded to one of four angles representing vertical, horizontal and the two diagonal directions (0°, 45°, 90°, 135°).[1] [2] [3]

– Application of non-maximum suppression to remove thick edges and to obtain a binary representation of the edge points.

  A point is considered to be part of an edge, if its gradient magnitude G in the direction θ is grater than the magnitude in the perpendicular direction.

– Hysteresis thresholding.

  The edges extraction from the map originated in the previous step is executed by means of a double thresholding procedure. Two threshold levels are defined and compared with the gradient magnitude in every pixel Q_i. If $G(Q_i)$ is lower than the lower threshold, the point is eliminated, else if the gradient is grater than the higher threshold, the point is accepted as part of a contour. Otherwise, a point with a value of $G(Q_i)$ ranged between the two levels is accepted only if adjacent to a previously accepted point. In this manner, a binary image depicting edges and details is finally obtained.

A process of Voronoi skeletonization [41], aimed at extracting a region-based shape feature representing the general form of an object, is then applied to the black/white image in order to extract and save the paths that have to be reproduced by the robotic arm. The output of Canny Edge Detector for the Trieste Seafront image is reported in Fig. 7a.

4.2.2 Hough Transform

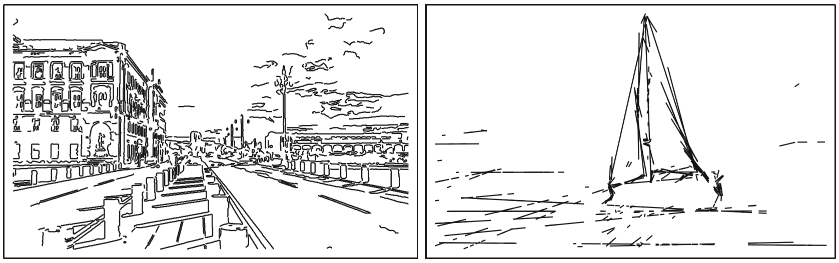
The Hough Transform [42, 43] is a feature extraction technique used in computer vision and digital image

[1] http://www.dersutmagazine.it/turismo/week-end-a-trieste-cosa-vedere-mangiare/

[2] http://panoramaplawniowice.pl/

[3] http://beautytakesthecity.blogspot.it/2011/02/ad-campaign-cle-de-peau-beaute.html/

Fig. 7 Examples of contours and details detection



(a) Canny Edge Detector (threshold 0.3). (b) Hough Transform (gradient threshold 0.3, detection sensitivity 0.001).

processing, that, in its classical version, is based on the recognition of lines in an image. The transform uses a parametric representation of a line

$$r = x\,cos(\phi) + y\,sin(\phi) \tag{4}$$

where r is the normal segment length from the origin to this line, whereas ϕ is the orientation of r with respect to the horizontal axis. Therefore, it is possible to associate with each line of the image a pair (r, ϕ). A two-dimensional array, called accumulator, is then adopted by the linear Hough Transform to detect the existence of a line described by Eq. 4. For each pixel with coordinates (x_i, y_i) and its neighbourhoods, the algorithm, by applying a relative gradient threshold, determines if there is enough evidence of a straight line at that pixel. If so, (r, ϕ) of that segment is calculated and the corresponding accumulator's bin value incremented. By extracting the bins with highest values, typically by looking for local maxima, the most likely lines can be extracted. The output of the algorithm can be controlled by means of two parameters: the gradient threshold and the detection sensitivity, which is a value inversely proportional to the number of features to be considered as lines. Finally, the coordinates of starting and ending points of each stroke are stored into a matrix and elaborated into paths for the robot. An example of Hough Transform is reported in Fig. 7b.

4.2.3 Skeletonization

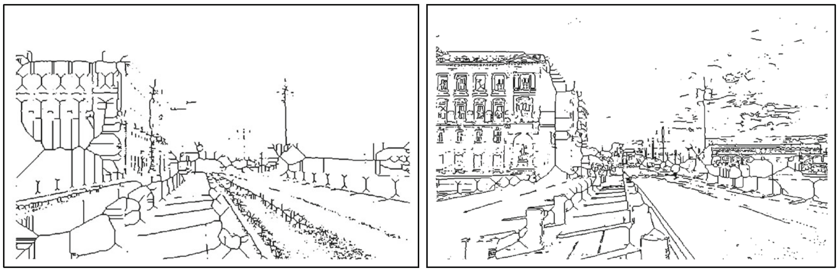
The third technique that has been implemented for the rendering of contours and details is Skeletonization. This algorithm is suitable for the rendering of dark edges. It consists in a thresholding procedure, that can be applied on both the grey-scale image or on the gradient field, in which I_T is set to a very high value. In this manner, only the darkest details of the image remain visible. Then, a skeletonization is applied [41], in order to extract the paths. This algorithm can be used preferably with light grey-scale images with dark contours; otherwise, different values of threshold have to be applied. In Figs. 8a and b two examples of Skeletonization, adopting a grey-scale and a gradient threshold respectively, are reported.

4.3 Overview on the Algorithms

In Fig. 9 a graphical overview on the implemented algorithms is shown. It has to be underlined that the algorithms here presented are not all used together for the rendering of a single image: it is up to the artist to choose the best combination of techniques for the rendering of a particular image. Therefore, there is no specific preference rule on the application of an algorithm in place of another. Combinations of different techniques can be applied on the artist's aesthetic appreciation and fulfilment. Furthermore, the artist is also able to influence the outcome of each single algorithm by setting and adjusting the software parameters before the real execution of the artwork. Table 1 summarizes the free software parameters for each image processing algorithm.

To reduce data size and to avoid dotted style, especially in the Random Strokes algorithm, a filter of the strokes has been implemented as final step of each algorithm, before the path planning module. In this manner, the artist can set the

Fig. 8 Examples of Skeletonization



(a) Grey-scale threshold equal to 0.3 (b) Gradient threshold equal to 0.3.

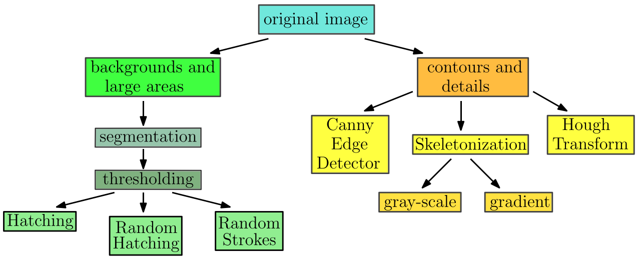


Fig. 9 Graphical overview on the implemented algorithms

minimum length of lines and the algorithm deletes all the strokes (paths) shorter than that value.

5 The Robotic System

In this section, the experimental implementation of the above analysed artistic rendering algorithms into real watercolour painting is presented. Watercolour painting is performed by a UR10 robot, by Universal Robots. It is a 6-DOF robotic arm with a working radius of 1300 mm from the base joint. It weighs 28.9 Kg, its payload is 10 Kg and it has a repeatability of ± 0.1 mm, suitable for the application. The robot tool is equipped with an aluminium flange that allows easy integration of paintbrushes. In Fig. 1 an image of the robot performing watercolour painting is shown. Since the UR10 is a collaborative robot and is provided with force limits and collision-detecting security systems, a human-robot interaction is allowed during painting. An artist can operate side by side with the robotic arm, adjusting the paint dilution, monitoring the watercolour drying on the paper and intervening whenever it is necessary.

In order to provide the robot with an appropriate environment, a calibration procedure is performed prior to starting the painting task. First of all, the position of the equipment such as paper (rough 300 g/m^2), paint cups and brush tools repository (see Section 5.3) has to be fixed relative to the robot base reference frame, which corresponds to the point $\{0, 0, 0\}^T$ of the robot workspace. Furthermore, a calibration of the drawing board surface has to be carried out, in order to compensate planarity errors due to a non perfect positioning of the robot base and drawing support. For this purpose, an arbitrary number n of points $P_i = \left\{ x_i \;\; y_i \;\; z_i \right\}^T$ on the drawing board is measured with respect to the robot reference frame. Then, the minimum mean square error surface $Z = aX + bY + c$ is computed by means of the pseudo-inverse matrix $A^+ = (A^T A)^{-1} A^T$, where the coefficient vector $K = \left\{ a \;\; b \;\; c \right\}^T$ is obtained as

$$K = A^+ b \tag{5}$$

where

$$A = \begin{bmatrix} x_i & y_i & 1 \\ \dots & \dots & \dots \\ x_n & y_n & 1 \end{bmatrix} \qquad b = \begin{Bmatrix} z_i \\ \dots \\ z_n \end{Bmatrix} \tag{6}$$

The maximum dimension of the drawing surface is 450 × 850 mm. The size of the resulting artwork can be rescaled

Table 1 Free software parameters for each image processing algorithm

Algorithm	Free parameters
Hatching	– grey threshold – distance between lines – angle of orientation – random perturbation of end-points (in the Small Angle Random Hatching) – minimum length of lines
Gradient-Based Random Strokes	– number of levels – grey threshold for each level – number of lines for each level – minimum length of lines
Canny Edge Detector	– grey threshold – Canny threshold (filter kernel width) – minimum length of edges
Hough Transform	– gradient threshold – sensitivity of line detection – minimum length of lines
Skeletonization	– grey-scale or gradient threshold – minimum length of lines

by software before the path planning module in a resolution-independent manner to adapt the dimensions of the artwork to the ones of the drawing surface.

5.1 Brushes

In this work, we have adopted Raphaël Soft Aqua brushes, ideal for watercolour, since they present a high fluid retention capacity, and Maimeri Venezia watercolours in tubes of 15 ml. As it will be explained in Section 5.3, the brushes have been equipped with plastic supports, that allow a simple integration with the robot and with the tool automatic change system. The supports have been manufactured in PLA by means of fused deposition modelling technology, using an Ultimaker 2^+ 3D printer. Four different brushes have been adopted and numbered from 1 to 4, as shown in Fig. 10. Prior to providing the analysis of watercolour strokes, the tip of the four brushes have been characterized. The inflection of the tip Δt with respect to the z-coordinate has been measured, as reported in Fig. 11. By considering $z = 0$ as the point in which the brush tip touches the working area without inflexion, five measures have been acquired for each brush, by decreasing the z-coordinate with equal steps of $\Delta z = 1\ mm$. Figure 12 reports four steps of the process, for brush number 2 (as in Fig. 10). In Fig. 13 the values of tip inflection are reported for all the brushes.

The tip inflection of brush 4 is lower with respect to the other brushes, since it is the brush that presents the shortest bristles and, therefore, a lower inflection Δt on equal values of Δz.

5.2 Strokes Analysis

Painted strokes realized with the watercolour brushes have been characterized in order to obtain the relations between the brush z-coordinate and the strokes width. In Figs. 14 and 15, two examples of strokes, respectively realized with brushes 3 and 1 at heights $\Delta z = 3$ and 5 mm, are reported together with the relative dimensionless intensity trend along the x-coordinate. The images of the strokes have been digitalized and the colour intensity has been analysed in the Matlab environment and averaged with the stroke width. As it can be seen, intensity decreases along the strokes (as 1 represents white) as the colour on the tip reduces. Furthermore, by for high values of Δz the phenomena of split brush occurs (Fig. 15). A precise and rigorous characterization of intensity along the stroke could be developed in the future with further analysis and experimentations. In this work, an optimization of the length of the stroke L that each brush can perform between a colour refill and the next one has not been performed. The colour refill algorithm consists of the command lines capable of steering the robot to the position of the watery pigment cup, dipping the tip and drain it. The maximum length L_{max} that a brush can travel before the tip gets dry has been manually tuned for each brush. The relation between the dipping depth of brushes and L_{max} will be taken into account in future developments of this work.

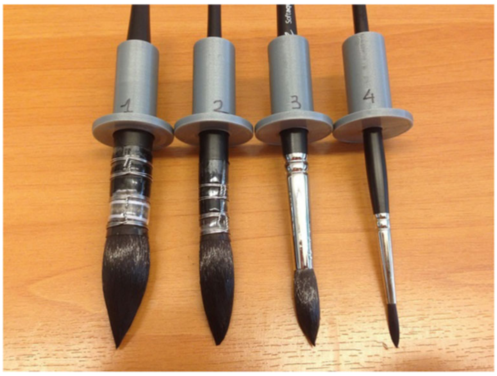


Fig. 10 Watercolour brushes with 3D printed supports

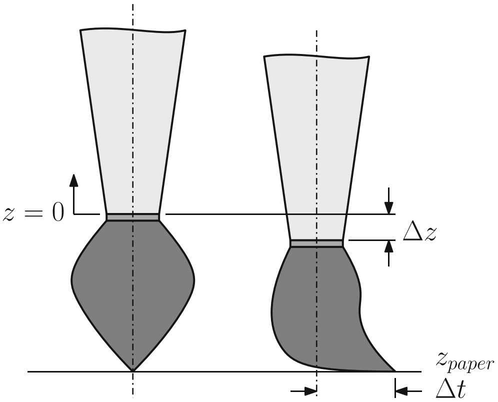


Fig. 11 Inflection of the brush tip with respect to the z-coordinate

On the contrary, a characterization of stroke thickness has been performed with interesting results for the four brushes and for five different depths from $z_1 = 1\ mm$ to $z_5 = 5\ mm$. For each brush and for each Δz, nine different strokes each one with a length equal to $l = 500\ mm$ have been realized and processed in order to determine the mean thickness distribution. Results are reported in the form of box-plots in Fig. 16. An interesting trend emerges, since stroke thickness increases by increasing the z-depth of brushes and by switching from thin (e.g. number 4) to thick brushes (e.g. number 1). This result has been then practically applied in order to pre-determine watercolour strokes thickness on the paper, a procedure that was manually performed.

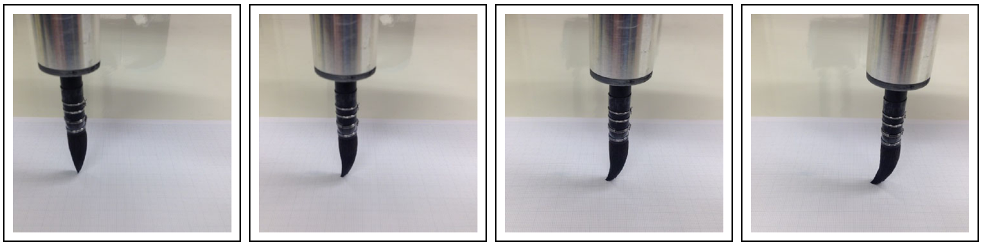

(a) $\Delta z = 0\ mm.$ (b) $\Delta z = 1\ mm.$ (c) $\Delta z = 2\ mm.$ (d) $\Delta z = 3\ mm.$

Fig. 12 Inflection of the brush tip with respect to the z-coordinate (brush 2)

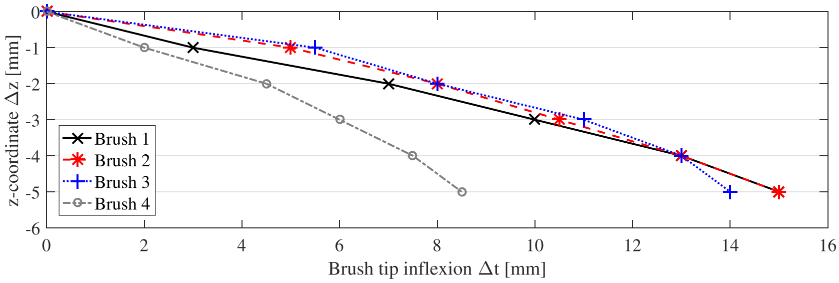


Fig. 13 Brush tip inflection with respect to the z-axis

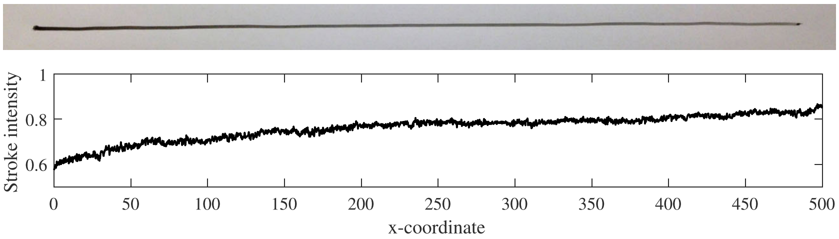


Fig. 14 Example of stroke and relative colour intensity along the x-coordinate: brush 3, $\Delta z = 3\ mm$

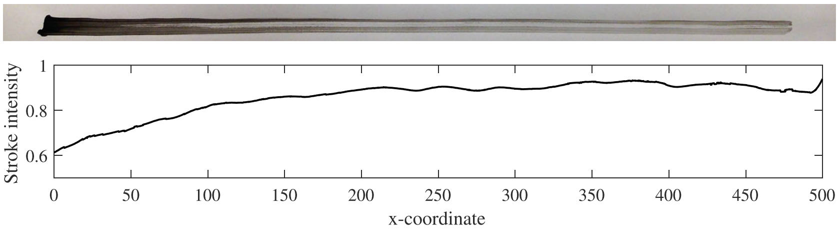


Fig. 15 Example of stroke and relative colour intensity along the x-coordinate: brush 1, $\Delta z = 5\ mm$

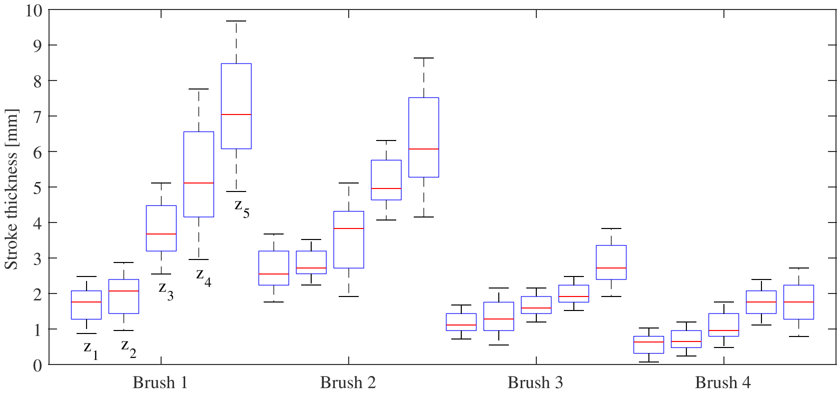
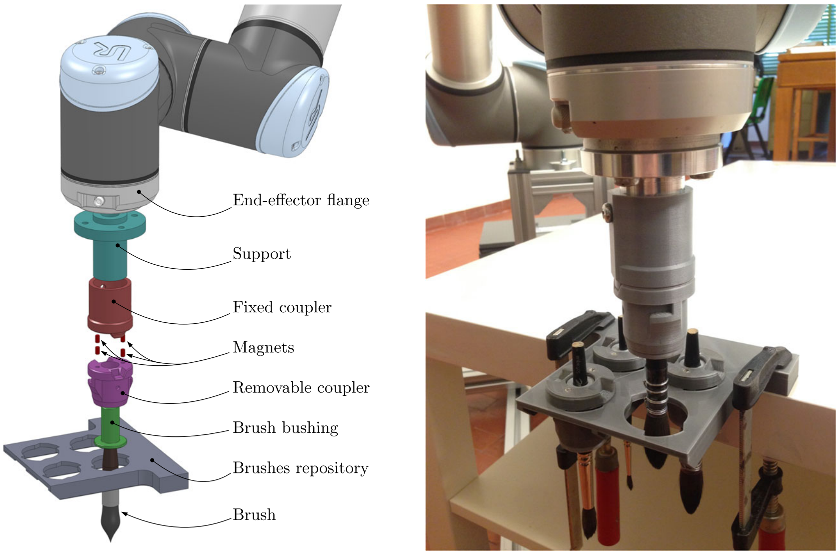


Fig. 16 Box-plots of stroke thickness distribution

Fig. 17 Automatic brush change



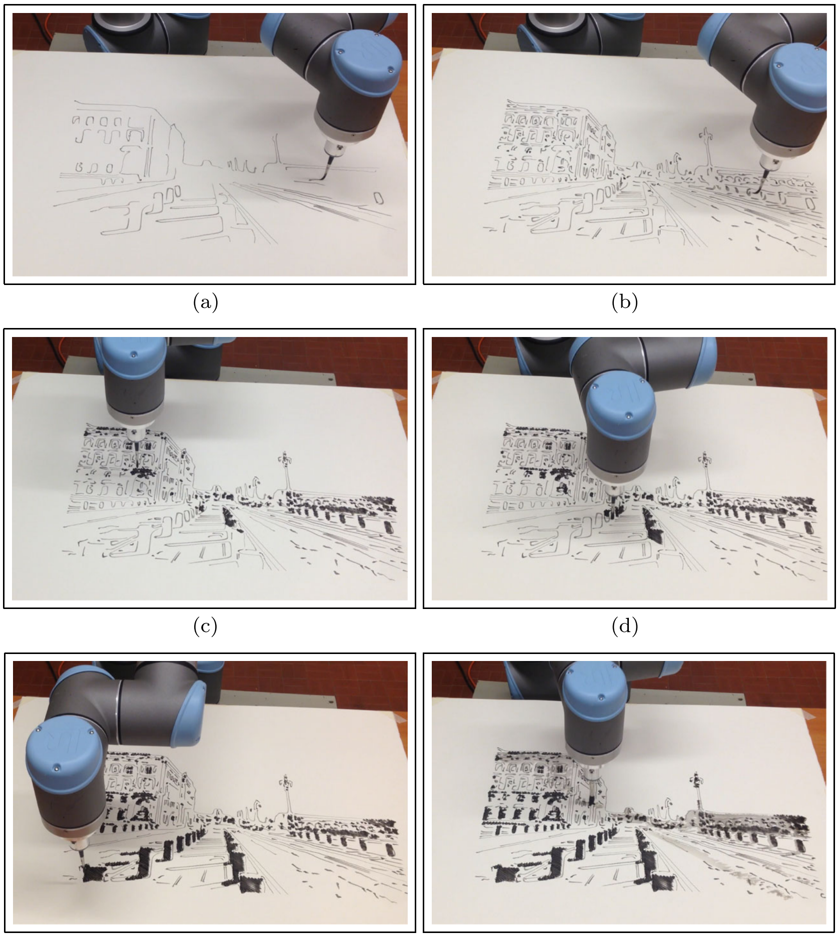
(a) CAD exploded view.

(b) Implementation.

5.3 Automatic Brush Change

Since the size of brushes has to be changed as the robot paints new layers with strokes of different thickness, an automatic brush change system has been implemented. Figure 17 reports the CAD exploded view (a) and the real implemented system (b). The automatic brush change is modular and it is composed of a fixed coupler, connected

Fig. 18 Frames sequence of Busker Robot painting Trieste Seafront



(a)

(b)

(c)

(d)

(e)

(f)

to the robot end-effector flange by means of an aluminium support, and by a removable coupler, rigidly connected to each of the four brushes. Fixed and removable couplers, 3D printed in PLA, can be joined together by means of four cylindrical magnets. The brushes repository is composed of a plastic plate provided with four properly shaped holes. A groove in the removable couplers allows an easy pick up and release of each brush tool with just a 45° clockwise or counter-clockwise rotation of the 6th robot joint. The steps of the algorithm used to change the brushes are the following:

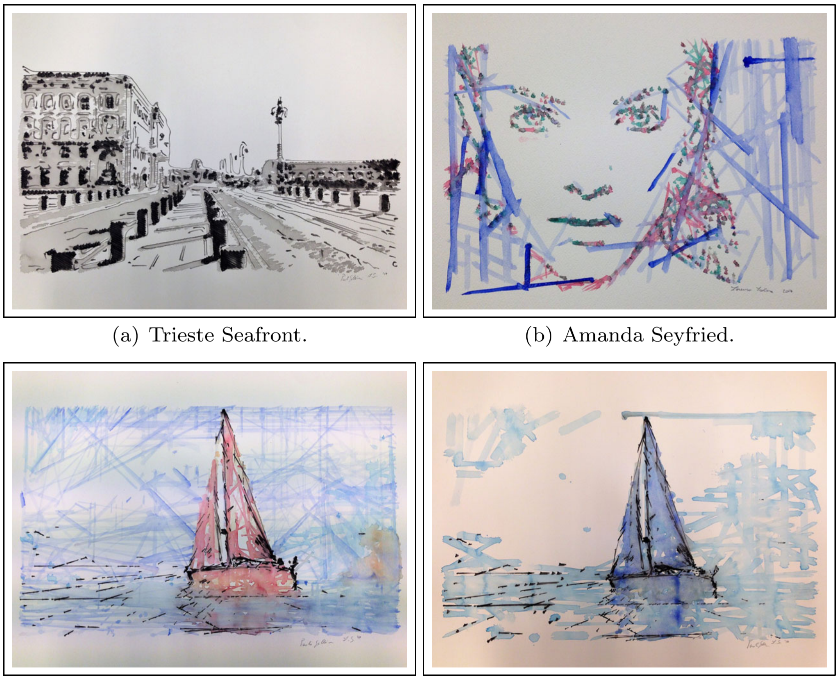
- align the vertical axis of the robot flange with the axis of the empty hole in the brush repository, maintaining the robot flange at a distance $\Delta z_B \geq L_{Bi}$, where L_{Bi} is the length measured from the tip of the current brush (i-th) to the robot Tool Center Point (TCP);
- rotate the 6-th joint of the robot to align the groove of the removable coupler with the shaped hole;
- decrease the z-coordinate of the TCP till the repository plane is reached;
- rotate the 6-th robot joint of 45° clockwise;
- increase the z-coordinate of the TCP of Δz_R, which is the length measured form the back-tip of the brushes in the repository and the repository plane: in this manner the brush (i-th) is released;
- align the vertical axis of the end-effector with the axis of the following brush ($i + 1$-th) to be picked up;
- decrease the z-coordinate of the TCP till the repository plane is reached;
- rotate the 6-th robot joint of 45° counter-clockwise: the brush ($i + 1$-th) is unhooked from the repository and mounted on the fixed coupler;
- increase the z-coordinate of the TCP of $\Delta z_{B(i+1)}$.

To the best of Authors' knowledge, only in the work of Deussen et al. [20], an automatic brush change system has been implemented until now.

The whole painting process takes approximately one to three hours depending on the complexity of the pictures, the number of strokes, the robot velocity and the number of layers to be executed.

6 Experimental Results

In this section, we demonstrate our experimental results by showing the pictures of the resulting artworks obtained by means of the presented algorithms and of the watercolour painting robot. The artworks parameters are reported in Table 2. Six frames of Busker Robot painting Trieste Seafront are reported in Fig. 18. The artwork is composed of three layers: the first reproduces the details of the image and it has been obtained by means of Canny algorithm (layer 1: threshold 0.3), the other two consist of a black and a grey covering respectively, obtained with the Random Hatching algorithm (layer 2: threshold 0.1, angle 30°, lines distance: 3 px; layer 3: threshold 0.3, angle 45°, lines distance: 6 px, random perturbation activated). The final artwork is reported in Fig. 19a. Figure 19b reports the elaboration of


(a) Trieste Seafront.

(b) Amanda Seyfried.

(c) Sailboat 1.

(d) Sailboat 2.

Fig. 19 Resulting artworks

Table 2 Artworks parameters

Artwork	Layer	Algorithm	Brush	Colour
Trieste Seafront	1	Canny	4	Black
	2	Hatching	4	Black
	3	Random Hatching	2	Grey
Amanda Seyfried	1	Gradient Skeletoniz.	4	Green
	2	Random Strokes	4	Red
	3	Random Strokes	2	Blue
Sailboat 1	1	Hough Transform	4	Black
	2 (sea and sky)	Random Strokes	3	Blue
	3 (sea and sky)	Random Strokes	2	Light Blue
	4 (boat)	Random Strokes	3	Red
	5	Random Strokes	3	Ocher
Sailboat 2	1	Hough Transform	4	Black
	2 (sea and sky)	Random Strokes	1	Light Blue
	3 (boat)	Random Strokes	3	Blue
	4 (boat)	Random Strokes	3	Ocher

Amanda Seyfried using a gradient skeletonization (gradient threshold 0.45) and a Random Strokes background (grey-scale threshold: 0.4 and 0.8, number of lines: 400 and 200, for layers 2 and 3, respectively). Furthermore, in Fig. 19c and d two different Sailboats, obtained with Hough Transform (gradient threshold 0.3, detection sensitivity 0.001) and Random Strokes rendering, are proposed. The backgrounds of the two artworks have been obtained with a different grey-scale threshold and number of random lines. The values of the parameters used for the layer 2 (we report the predominant layer only as example) of Sailboat 1 and Sailboat 2 are respectively: grey-scale threshold 0.9 and 0.8, number of lines 300 and 200. It has to be underlined that different parameters of the same technique and hardware variables can lead to completely different outcomes, starting from the same input image.

With reference to the properties described in Section 2.1, we can analyse the effects of watercolour that appear in the resulting artworks of Fig. 19. In the picture of Trieste Seafront (Fig. 19a) we can easily identify the edge-darkening effect in the light-grey watercolour and the presence of the dry-brush effect in some black parallel lines in the lower part of the image. The resolvability of black after the application of the third layer of grey can as well be found. The effects of transparency and edge-darkening can be identified in the picture of Amanda Seyfried (Fig. 19b), where the blue layer has been applied with some strokes of diluted colour. Colour mixing, backruns and granulation are effects clearly visible in the picture of Sailboat 1 (Fig. 19c), where the high number of strokes crossing each other results in a paper saturated with water. The same effects can be found in the picture of Sailboat 2 (Fig. 19d), where a granulation effect is shown on the hull of the sailboat and on its reflection in the sea, due to the ripples of the wet paper.

In the context of watercolour robotic painting, a quantitative analysis and validation of experimental results is not possible, since the appreciation of the artworks is subjective and personal. Busker Robot was installed at the “Algorithmic Arts and Robotics” exposition during Trieste Next (Trieste, Italy, September 2017) and won an Honorable Mention at the 2018 International Robotic Art Competition (RobotArt) [25]. According to the public, the press and on-line news [44], it has been successful and the artworks have been considered unique in their novel watercolour style.

7 Conclusion

In this paper, a novel robotic system for the artistic rendering of watercolour paintings, realized by means of a 6-DOF collaborative robot, has been presented. After a survey of robotic applications to artistic painting and of recent digital watercolour rendering, the traditional aquarelle painting has been briefly analysed. Several non-photorealistic rendering techniques, aimed at processing both the backgrounds and the details of a digital image, have been implemented. The resulting rendering have been converted into a series of paths that the robot can artistically reproduce on the canvas. During the process, the artist controlling the system can change algorithm parameters and hardware variables (e.g. brush type, colour dilution and shading) in order to influence the outcome. In the context of robotic watercolour painting a characterization of brushes as well as a tool automatic change have been implemented. Interesting results and robot-painted artworks, obtained by combining different algorithms and image layers painted with both watercolour and gouache techniques, have been presented and illustrated.

In the future, we plan to further study and refine our watercolour robotic painting techniques. In particular, an algorithm for the detection of facial features could be implemented as well as a feedback system that could monitor and control the painting process by means of a camera. Furthermore, a deeper analysis of stroke intensity could be performed, e.g. to better control the colour refill on the brush with respect to the total distance covered on the paper in function of z coordinate and brush type. Other interesting improvements could be related to the dipping depth of the brush inside the paint cup and the attachment point of the brush on the paper.

Although the techniques here implemented could be further developed and refined, we consider this work a complete integration of multidisciplinary fields, including image processing, robotics and arts. We are confident that, in future, the system here presented could become an important example of robotic painting, attracting the interests of visitors for its original artistic expression.

Acknowledgements This research has been in part supported by Fondo Ricerca Ateneo, FRA 2015 (Internal Fund, University of Trieste).

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

References

1. Smith, G.W., Leymarie, F.F.: The machine as artist: an introduction. In: Arts vol. 6 no.2. Multidisciplinary Digital Publishing Institute, p. 5 (2017)
2. Aguilar, C., Lipson, H.: A robotic system for interpreting images into painted artwork. In: International conference on generative art, vol. 11 (2008)
3. Tinguely art machines. https://en.wikipedia.org/wiki/Jean_Tinguely/
4. Cohen, H.: The further exploits of aaron, painter. Stanf. Humanit. Rev. **4**(2), 141–158 (1995)
5. Gommel, M., Haitz, M., Zappe, J.: Robotlab autoportrait project: Human portraits drawn by a robot. http://www.robotlab.de/
6. Calinon, S., Epiney, J., Billard, A.: A humanoid robot drawing human portraits. In: 2005 5th IEEE-RAS international conference on humanoid robots. IEEE, pp. 161–166 (2005)
7. Jean-Pierre, G., Saïd, Z.: The artist robot: A robot drawing like a human artist. In: IEEE International Conference on Industrial Technology (ICIT), 2012. IEEE, pp. 486–491 (2012)
8. Yao, F., Shao, G.: Painting brush control techniques in chinese painting robot. In: IEEE international workshop on robot and human interactive communication. ROMAN 2005. IEEE, 2005, pp. 462–467 (2005)
9. Kudoh, S., Ogawara, K., Ruchanurucks, M., Ikeuchi, K.: Painting robot with multi-fingered hands and stereo vision. Robot. Auton. Syst. **57**(3), 279–288 (2009)
10. Lu, Y., Lam, J.H., Yam, Y.: Preliminary study on vision-based pen-and-ink drawing by a robotic manipulator. In: IEEE/ASME international conference on advanced intelligent mechatronics, AIM 2009. IEEE, 2009, pp. 578–583 (2009)
11. Sun, Y., Xu, Y.: A calligraphy robot—callibot: design, analysis and applications. In: IEEE international conference on robotics and Biomimetics (ROBIO). IEEE 2013, pp. 185–190 (2013)
12. Kim, S., Song, M., Kim, P., Lee, G., Ganbold, N., Tang, J.: A development of art robot system for representation of brightness of image. Intell. Auton. Syst. **12**, 433–441 (2013)
13. Tresset, P., Leymarie, F.F.: Portrait drawing by paul the robot. Comput. Graph **37**(5), 348–363 (2013)
14. Benedettelli, D.: Legonardo (2013). http://robotics.benedettelli.com/legonardo/
15. Jain, S., Gupta, P., Kumar, V., Sharma, K.: A force-controlled portrait drawing robot. In: International Conference on Industrial Technology (ICIT), IEEE, pp. 3160–3165 (2015)
16. Luo, R.C., Hong, M.-J., Chung, P.-C.: Robot artist for colorful picture painting with visual control system. In: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2016, pp. 2998–3003 (2016)
17. Berio, D., Calinon, S., Leymarie, F.F.: Learning dynamic graffiti strokes with a compliant robot. In: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, pp. 3981–3986 (2016)
18. Scalera, L., Mazzon, E., Gallina, P., Gasparetto, A.: Airbrush robotic painting system: Experimental validation of a colour spray model. In: International conference on robotics in Alpe-Adria Danube region. Springer, pp. 549–556 (2017)
19. Ago, V.: Robotic airbrush painting. http://violaago.com/robotic-airbrush-painting/
20. 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. Eurographics association, pp. 25–33 (2012)
21. Lindemeier, T., Metzner, J., Pollak, L., Deussen, O.: Hardware-based non-photorealistic rendering using a painting robot. In: Computer graphics forum, vol. 34, no. 2. Wiley Online Library, pp. 311–323 (2015)
22. 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: IEEE International Conference on Robotics and Automation (ICRA) (2018)
23. Ticini, L.F., Rachman, L., Pelletier, J., Dubal, S.: Enhancing aesthetic appreciation by priming canvases with actions that match the artist's painting style, Frontiers in human neuroscience, vol 8 (2014)
24. Arte algoritmica e robotica: Next, Trieste, September, 21-23, 2017, http://www.triestenext.it/arte-algoritmica-robotica/ (2017)
25. 2018 International Robotic Art Competition (RobotArt). http://robotart.org/archives/2018/artworks/ (2018)
26. Seymour, P.: The artist's handbook Arcturus (2003)
27. Watercolour, History overview, http://watercolor.net/history-overview/ (2012)
28. Curtis, C.J., Anderson, S.E., Seims, J.E., Fleischer, K.W., Salesin, D.H.: Computer-generated watercolor. In: Proceedings of the 24th annual conference on computer graphics and interactive techniques. ACM Press/Addison-Wesley Publishing Co., pp. 421–430 (1997)
29. Lum, E.B., Ma, K.-L.: Non-photorealistic rendering using watercolor inspired textures and illumination. In: Proceedings. Ninth pacific conference on computer graphics and applications. IEEE, 2001, pp. 322–330 (2001)
30. Burgess, J., Wyvill, G., King, S.A.: A system for real-time watercolour rendering. In: Computer graphics international 2005, IEEE, pp. 234–240 (2005)

31. Van Laerhoven, T., Liesenborgs, J., Van Reeth, F.: Real-time watercolor painting on a distributed paper model. In: Proceedings. Computer graphics international. IEEE, 2004, pp. 640–643 (2004)
32. Bousseau, A., Kaplan, M., Thollot, J., Sillion, F.X.: Interactive watercolor rendering with temporal coherence and abstraction. In: Proceedings of the 4th international symposium on Non-photorealistic animation and rendering. ACM, pp. 141–149 (2006)
33. Luft, T., Deussen, O.: Real-time watercolor for animation. J. Comput. Sci. Technol. **21**(2), 159–165 (2006)
34. Montesdeoca, S.E., Seah, H.S., Rall, H.-M.: Art-directed watercolor rendered animation. In: Proceedings of the joint symposium on computational aesthetics and sketch based interfaces and modeling and non-photorealistic animation and rendering. Eurographics Association, pp. 51–58 (2016)
35. Montesdeoca, S., Seah, H.S., Bénard, P., Vergne, R., Thollot, J., Rall, H.-M., Benvenuti, D.: Edge-and substrate-based effects for watercolor stylization. In: Expressive 2017, the symposium on computational aesthetics, Sketch-based interfaces and modeling, and non-photorealistic animation and rendering, p. 10 (2017)
36. Sefiane, K.: Patterns from drying drops. Adv. Colloid Interf. Sci. **206**, 372–381 (2014)
37. Kyprianidis, J.E., Collomosse, J., Wang, T., Isenberg, T.: State of the art: a taxonomy of artistic stylization techniques for images and video. IEEE Trans. Vis. Comput. Graph. **19**(5), 866–885 (2013)
38. Zeng, K., Zhao, M., Xiong, C., Zhu, S.C.: From image parsing to painterly rendering. ACM Trans. Graph. **29**(1), 2–1 (2009)
39. Seriani, S., Cortellessa, A., Belfio, S., Sortino, M., Totis, G., Gallina, P.: Automatic path-planning algorithm for realistic decorative robotic painting. Autom. Constr. **56**, 67–75 (2015)
40. Canny, J.: A computational approach to edge detection. IEEE Trans Pattern. Anal Mach Intell **6**, 679–698 (1986)
41. Ogniewicz, R.L., Kübler, O.: Hierarchic voronoi skeletons. Pattern Recog. **28**(3), 343–359 (1995)
42. Duda, R.O., Hart, P.E.: Use of the hough transformation to detect lines and curves in pictures. Commun. ACM **15**(1), 11–15 (1972)
43. Ballard, D.H.: Generalizing the hough transform to detect arbitrary shapes. Pattern Recog. **13**(2), 111–122 (1981)
44. aNEWS: Italian prof. creates robotic artist, http://www.anews.com.tr/webtv/life/italian-prof-creates-robotic-artist/ (2017)

Lorenzo Scalera achieved the Bachelor's Degree in Industrial Engineering and the Master's Degree in Mechanical Engineering, both with honours, at University of Trieste (Trieste, Italy) in 2012 and 2015, respectively. He is currently a Ph.D. student in Applied Mechanics and Robotics at the Polytechnic Department of Engineering and Architecture, University of Udine (Udine, Italy). From January to July 2018 he was a visiting PhD student at the Wearable Robotic Systems (WRS) Lab, Department of Mechanical Engineering, Stevens Institute of Technology (Hoboken, NJ, USA). His research interests are in the fields of: cable-driven robots, dynamic modelling and simulation of flexible-link mechanisms, haptics, human-robot interaction and collaborative robotics for artistic applications.

Stefano Seriani was born in Trieste, Italy, in 1986. He received the B.E. Degree in Industrial Engineering in 2010, the M.Sc. Degree in Mechanical Engineering in 2012, and the Ph.D. Degree in Applied Mechanics and Robotics from the University of Trieste (Trieste, Italy) in 2016. In 2016, he was a Research Fellow with the Institute of Robotics and Mechatronics of the German Space Agency (DLR). He is currently a Research Fellow with the University of Trieste. His research interests include space robotics, applied mechanics, computer vision, and image processing.

Alessandro Gasparetto is currently Full Professor of Mechanics of Machines at the Polytechnic Department of Engineering and Architecture, University of Udine (Udine, Italy), where he is the head of the research group in Mechatronics and Robotics and has several scientific and institutional appointments. He is currently Vice- Rector for Quality Assurance at the University of Udine and President of IFToMM Italy, member organization of IFToMM (the International Federation for the Promotion of Mechanism and Machine Science). His research interests are in the fields of: modelling and control of mechatronic systems, robotics, mechanical design, industrial automation, mechanical vibrations. He is author of about 180 international publications and of three patents in the field of industrial automation.

Paolo Gallina received the Master's Degree in Mechanical Engineering from the University of Padova (Padova, Italy) in 1996 and the Ph.D. Degree in Applied Mechanics from the University of Brescia (Brescia, Italy) in 1999. He is currently an Associate Professor of applied mechanics with the Department of Civil Engineering and Architecture, University of Trieste (Trieste, Italy). He was a Visiting Professor with the Ohio University (Athens, OH) in 2000/1. In 2002, he implemented a hands-on Mechatronic Laboratory for students in Engineering. In 2003, he implemented a Robotics Laboratory, where he carries out his main research in robotics. In 2004, he was a Visiting Professor with Colorado University (Boulder, CO, USA) at the Centre for Advanced Manufacturing and Packaging of Microwave, Optical and Digital Electronics, in order to collaborate on mechatronics and micromechanics fields. He was the Head of the Council for Students in Mechanical Engineering degree from 2004 to 2008, and the Head of the Masters program: "Safety and hygiene in the working environment" from 2006 to 2008. His research interests include vibrations, human-machine interfaces, robotics, especially applied to rehabilitation.