Automatic path-planning algorithm for realistic decorative robotic painting

S. Seriani [a,*], A. Cortellessa [a,1], S. Belfio [b,2], M. Sortino [b,3], G. Totis [b,4], P. Gallina [a,5]

[a] Department of Architecture and Engineering, University of Trieste, via A. Valerio 10, 34127 Trieste, Italy
[b] Department of Electrical, Management and Mechanical Engineering, University of Udine, Italy

ARTICLE INFO

Accepted 26 April 2015

Keywords:
Spray painting
Robot
Ordinary least squares
Path-planning
Voronoi

ABSTRACT

In this paper an innovative algorithm to reproduce non-uniform, photorealistic, gray-scale images on large surfaces, using an ordinary industrial spray-painting robot is proposed.
The algorithm splits the process into a set of iterative steps with decreasing spray-gun stroke diameters. Thus, it can efficiently build up the image starting with large strokes to paint the larger details of the image. Then, with increasingly smaller strokes, it can paint the rest of the smaller details.
The target image is segmented and a tool-path is computed. A set of critical points in the image is then chosen to avoid oversaturation and used to implement an algorithm to calculate spray-gun operational speed at each path point. Eventually, such conditions lead to a linear system which is solved using an ordinary least squares method.
Depending on the image to reproduce, this methodology promises to be far more efficient than painting processes where the image is built entirely at the smallest detail level. For this reason, it would be particularly suitable for large building façade decoration, for example.

1. Introduction

Nowadays, robotic painting is a very important process in industry; it spans from automotive to manufacturing, even to art. In the vast majority of applications the spray painting technique is used rather than others, i.e. brush painting. Indeed this technique couples very well with automated systems, and can be very efficient. Normally, the painting process aims at obtaining a uniformly painted surface of arbitrary shape. It is worth noting that, some aspects of robotic painting were left behind, in particular those concerning non-uniform painting. An ordinary example of this kind of painting is a photo, or a drawing. At present, no automated methodology or algorithm aimed to replicate an accurate copy of a digitized image (be it a photograph or something less realistic) on a large surface by means of an industrial spray-painting robot has been proposed. The purpose of this work is thus to present an innovative methodology and algorithm to tackle this challenge.

Several branches in the industry could be positively affected by this methodology, especially where decorative and/or functional painting is required. Automotive and furniture industries, particularly in the high-end segments, can present the need for this type of operations, and up to now this was generally met with the aid of highly skilled artisans or with complex masking systems (which, in turn, needs other professionals as well). It goes without saying that these procedures often impose great cost and time. Another promising branch is the one involved in construction; some companies (Cité-Création, in Lyon, France, http://cite-creation.com) use classic techniques like fresco and trompe-l'oeil to decorate buildings and homes, this mainly in an attempt to increase quality of life in crowded and dull city environments. Along the same line, in a less artistic way perhaps, industry buildings often need signs, text or warnings on walls or machinery, and while on small surfaces decals can be used, with larger surfaces this becomes impractical. The history of painted furniture is as old as society in itself, dating back to at least ancient Egypt; in modern times the interest in this kind of art has somewhat subsided due to high costs, but examples are still present. If, though, the process were to be completely automated and robotized, cost would lower substantially.

At present, the mechanics of the spray painting are well understood, for example Balkan and Arikan [1], Ellwood et al. [2] and Conner et al. [3] presented models for the flow rate flux of a paint gun and for the paint deposition in spray painting. Chen et al. [4] experimentally analyzed the paint coating characteristics for uniform velocity with overlapping paths, whereas Fogliati et al. [5] provided a numerical CFD analysis of the paint deposition process. Elliot et al. [6] provided a model for the

[*] Corresponding author. Tel.: +39 040 5582540.
E-mail addresses: stefano.seriani@phd.units.it (S. Seriani), alecortellessa@gmail.com (A. Cortellessa), sandro.belfio@uniud.it (S. Belfio), sortino@uniud.it (M. Sortino), giovanni.totis@uniud.it (G. Totis), pgallina@units.it (P. Gallina).
[1] Tel.: +39 040 5582540.
[2] Tel.: +39 0432 558297.
[3] Tel.: +39 0432 558241.
[4] Tel.: +39 0432 558258.
[5] Tel.: +39 040 5583829.

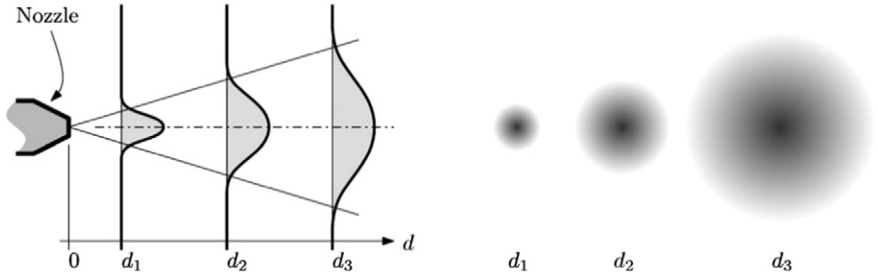

Fig. 1. Spray-paint cone morphology. The cone originates from the nozzle, and at various distances (for example d_1, d_2 and d_3) an indication of the cross-sectional paint distribution is visible in gray, along with the resulting stroke on the canvas (on the right). For clarity purpose, the distributions are normalized over the maximum value of each curve.

fluid-dynamics effects of spray-painting with a rotary nozzle. Atkar et al. [7] published a work on robotic uniform spray-painting in automotive. Robotic spray painting is also widely acknowledged; in normal industrial practice the manual teaching methods (see Baldwin [8]) are widely used. Chen et al. [9] propose a review of the current state of the art in the automatic path planning for industrial robotic spray-painting. Artistic, non-uniform painting by means of autonomous or automatic systems has been a subject of research in the last two decades, mainly since the work by Haeberli [10] on the virtual representation of images using pictorial, abstract styles. A crucial reference for our work paper is the concept highlighted by Hertzmann [11], in which the painting is carried out (virtually) with progressive decreasing-size strokes, following the footsteps of Haeberli. A series of different artistic painting robots are presented in literature, a few examples of which are in [12–14]. All these are not adequate since they tend to produce artistic-looking, non-photorealistic paintings.

The best example of system used to reproduce photo-realistic images on a surface is the printer, in its various forms. The problem of this system is that it operates essentially dot-by-dot, regardless of the detail sizes in the image. This translates in very long painting times if the surface is large. Moreover, the resolution is ultimately established by the hardware in the print head, which is a big limitation in flexibility.

For these reasons, in this paper an innovative methodology to reproduce photo-realistic images on surfaces by means of a fully automatic robotic spray-painting process is proposed. The method is centered on the concept that with a spray-painting robot we can achieve a wide range of stroke sizes; this can be exploited by using the larger strokes to paint the large details, and the smaller strokes for the smaller details. Efficiently calibrating this process leads to virtually any outcome, from low-detail and hi-speed to slow-speed but high-detail executions. The core of the methodology is an adaptive, smart algorithm that uses techniques like Voronoi diagrams, convolution and ordinary least squares

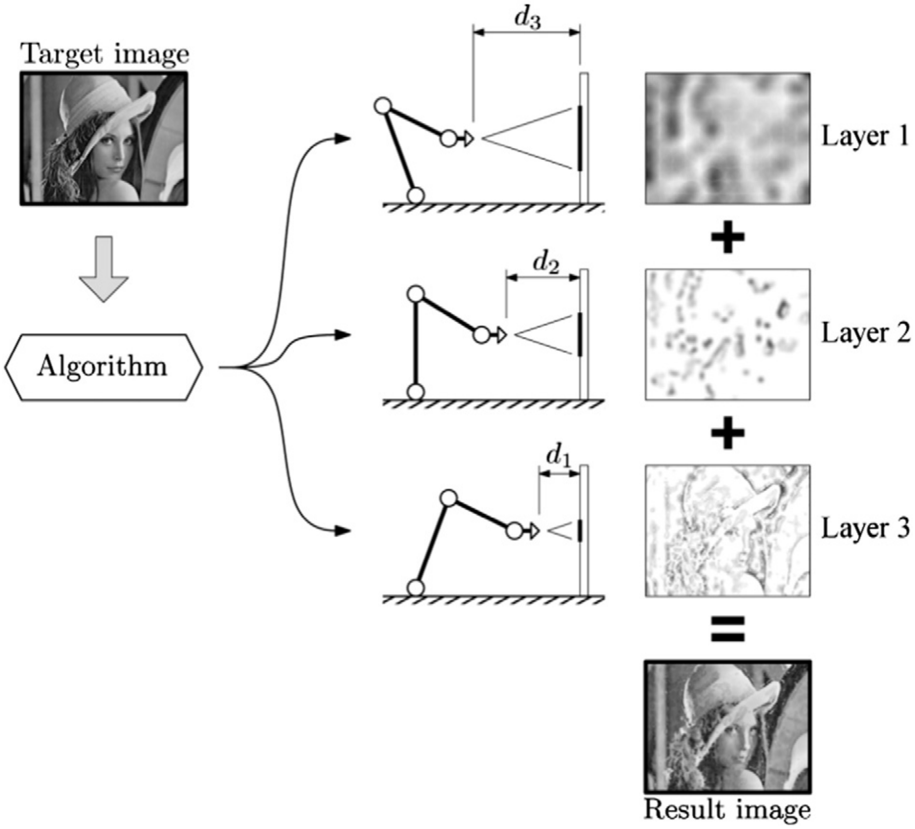

Fig. 2. The proposed methodology. The image (upper-left) is fed into the algorithm, which, according to the spray morphology and mechanics, splits the paint execution in a series of consecutive tasks, each of which takes advantage of a different stroke size, and operates mostly on details of matching size. The progressive execution of these steps generates the resulting image. Note that $d_3 > d_2 > d_1$.

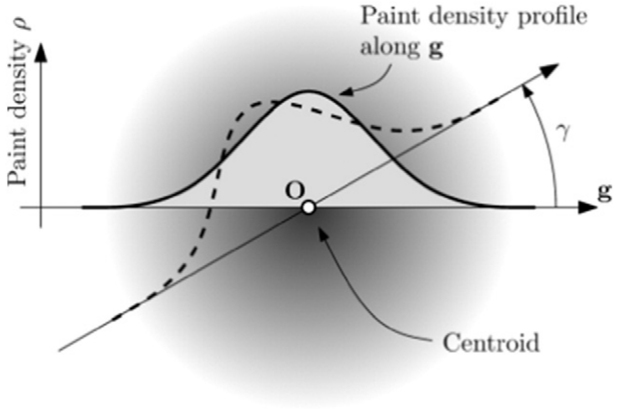

Fig. 3. Analysis of the paint density profile $\boldsymbol{\rho}$ of the paint stroke. The centroid $\boldsymbol{O}$, the axis $\boldsymbol{g}$ (coincident with $\boldsymbol{O}$) and the angle $\boldsymbol{\gamma}$ are visible. The paint density profile is also appreciable for two different orientations of $\boldsymbol{g}(\boldsymbol{\gamma})$.

(OLS) methods to define a path and feed-rate to control a general robot. In this article we use only gray-scale images, but extension to full color is straightforward.

In Section 2 the methodology in its most general form is outlined. In Section 3 the results of a preliminary experimental campaign on the morphology of a spray paint cone in an airbrush are presented. In Section 4 the actual algorithm which implements the methodology is illustrated and in Section 5 the discussion of the results with a test image is given. Finally, in the Conclusion section, the overall efficiency and accuracy of the proposed methodology are discussed.

2. Proposed methodology

The key concept in the proposed approach is that, in the spray-painting process, the size of the stroke is determined by the distance of the nozzle of the gun from the surface. This dimension can be adjusted by varying the distance of the end-effector (EE) of the robot from the target surface. The morphology of the spray-paint cone originating from the nozzle is illustrated in Fig. 1. The Gaussian-like distributions shown at distances d_1, d_2 and d_3 are the cross-sectional paint-flow profiles of the jet. It can be noticed that these represent equally well the paint collected in a unit of time on a surface if said surface was placed at that specific distance. In other words, being density the ratio of paint-covering on the surface, the distributions represent the density profile per time unit, as this is deposited on the surface.

The main challenge of the proposed approach is to split the painting task into a series of cumulative sub-tasks with increasing detail. It is worth noting that as the level of detail increases (thus requiring lower sized strokes), the time required to perform the sub-task increases. For this reason it is crucial to perform as much work as possible in the first steps, where the larger strokes are employed. The overall process is illustrated in Fig. 2.

It is easy to understand that, in order to obtain a good result, the stroke size at each step must be chosen wisely. Furthermore, the number of steps and the size of the stroke at the last step are very important for the overall quality of the finished piece. Many methods can be made available to precisely calibrate these parameters (empirical, semi-empirical, numerical optimization), but are not of our immediate concern for the purpose of this paper.

3. Experimental investigation of the spray morphology

Preliminary experimental tests were performed in order to analyze the physical characteristics of the painting process. The first one was aimed to identify the paint density profile shape at different distances of the nozzle from the surface, the second one was aimed to determine how the deposition of paint on the canvas varies with time. The paint density profile is crucial to precisely define how the paint will be distributed on the actual surface: it is the main parameter for the algorithm, as shown in the following sections. The deposition rate is comparatively less important than the density profile. Therefore, for the sake of simplicity a linear approximation was adopted.

The main objective of these tests was to provide a framework, and to define a procedure to evaluate the main physical properties involved in the system. The experimental investigation conveys a general idea of the process of measurement.

3.1. Paint density profile

This test was done by performing a quick burst with the airbrush at fixed distances from the canvas. This approach is similar to many found in literature [1]. The paint used in the experiment was black acrylic at a 2:5 paint to water ratio. Note that the airbrush nozzle was placed orthogonal to the canvas. The resulting strokes were then digitized with a *Canon PIXMA MP280* scanner; the image was then analyzed to provide a paint density profile.

Taking Fig. 3 as a reference, the axis $\boldsymbol{g}$, centered in $\boldsymbol{O}$ ($\boldsymbol{O}$ is the barycenter of the paint stroke), is progressively rotated by $\gamma \in [0, \pi]$ and the paint density along $\boldsymbol{g}(\gamma)$ is stored as $\rho(r, \gamma)$, where r is the radius. We can thus calculate the mean density profile by averaging the values of $\rho(\gamma)$ for each radius r value. A typical result is given in Fig. 4a. The process can be repeated at different distances of the nozzle from the canvas, as visible from Fig. 4b. Additionally, we find that the

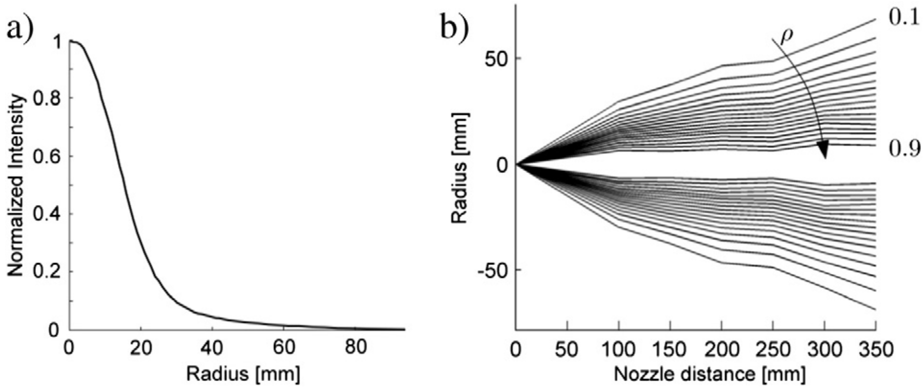

Fig. 4. Example of experimental results for the paint density profile at various distances. In a) a density profile is visible: the nozzle–canvas distance was 100 mm. In b) one can appreciate the qualitative paint density ρ distribution along a longitudinal section of the spray-paint jet; the arrow shows the direction of increment of ρ, whereas the lines represent iso- ρ levels in the spray section; the step is $\Delta\rho = 0.5$. All the graphs are symmetric.

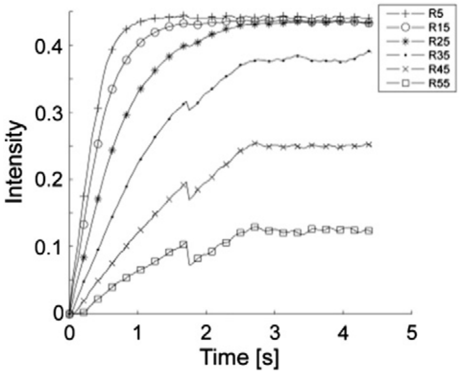

Fig. 5. Paint deposition ratio over time. The different lines represent different points along the general radius, from 5 to 55 mm (starting from the center), as shown in the legend. It can be noticed that an intensity of approximately 0.45 is equivalent to saturation due to the video-camera setup. The nozzle–canvas distance was 100 mm. The presence of a step in the R25 through R55 lines is due to the appearance of spots on the painted surface, which are in turn caused by non-adhesion of the paint to the substrate.

profile can be well approximated with a Gaussian curve. We generally find sufficient to truncate the curve at 2 or 3 σ.

3.2. Paint deposition rate law

If a small fixed region inside the boundaries of the spray-gun stroke is considered, it is possible to analyze the density of paint deposited over time; therefore, the law for the paint deposition ratio can be extracted. This was performed for a series of different points along the generic radius, and for several different distances. This experiment was performed using a video-camera (*Nikon D90*) to record the spraying process on a common paper substrate. In Fig. 5 a typical result of this experiment is shown. It is possible to observe that the first part of all the experimental curves is very close to linearity.

This property is important because it determines how two different layers of non-saturated paint interact with each other. If the deposition ratio is linear (or can be approximated as linear) the resulting density of two overlaying strokes of paint will be equal to the sum of their densities in each point.

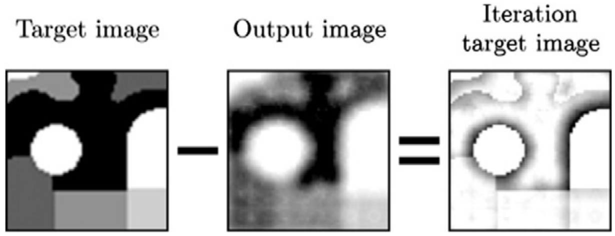

Fig. 7. Generation of the iteration target image. During a general step of the algorithm the *iteration target image* (ITI) is found by subtracting the *output image* (resulting from the preceding step) from the *target image*. Black areas are to be painted.

4. Algorithm

The methodology proposed in this paper mainly relies on a three-step algorithm. As mentioned in Section 2, the idea is to perform the painting procedure on multiple levels (layers) on the canvas, each determined by a different stroke size (given by the distance), and thus by a different detail size.

In the following sub-sections some basic definitions are provided, then the structure of the algorithm with its main operational blocks is illustrated, and finally the *timing algorithm* is described, which is the block that uses OLS to calculate the speed profile of the EE.

4.1. Definitions

In order to provide a consistent framework for the illustration of the algorithm, a set of definitions is given in the following.

Let us indicate Q_i as a general point of the surface and define the intensity $I(Q_i)$ as the value corresponding to the paint-density ρ when this is acquired by the acquisition system (camera, scanner, etc.). It is, essentially, the integral of the paint-flow per time unit, which was shown in Fig. 1, and is mapped in the 0–1 interval.

Given a specific *target image* (TI) to be painted on a canvas, for each point Q_i of the TI, a target gray-intensity $I_{gray,target}(Q_i)$ for that point can be determined. Along the same line, if the canvas is not blank (since the process is intrinsically multi-step and iterative, this happens after the first layer is deposited), a starting intensity $I_{gray,start}(Q_i)$ can be determined.

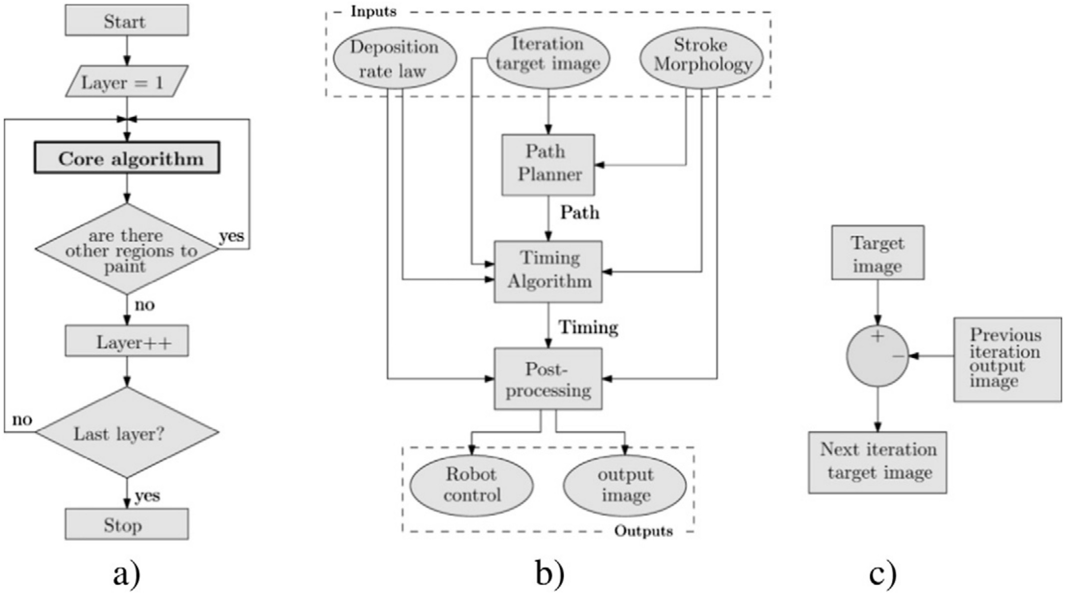

Fig. 6. The algorithm used to perform automatic path-planning for a photorealistic painting robot. In a) the general methodology is visible, the *Core algorithm* block is visible in b), with highlighted inputs and outputs; and lastly, in c) the generation of the *iteration target image* (see b)) is visible.

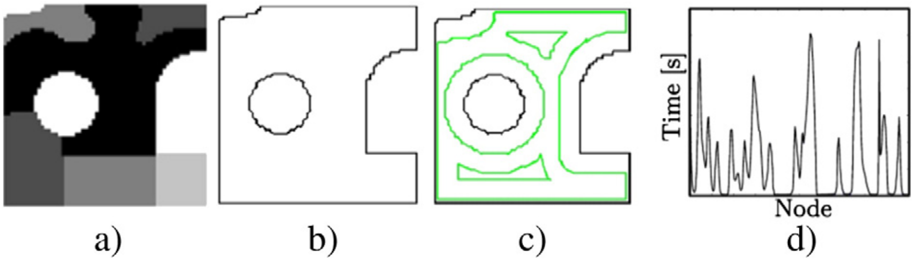

Fig. 8. Generation of the tool-path and speed profile for the robotic system. In a) a simple target image is visible, in b) the contours of the paintable area are detected; and the computed offsets which form the actual tool-path are presented in c). Finally in d) one can see the speed profile along a partition of the tool-path.

It is important to remark the concept of *over-painting*: this term means essentially oversaturation, i.e. when the coating of paint is already at its maximum density, and the spray-gun continues to apply paint; this usually translates in droplets forming on the canvas. This leads to sub-optimal outputs. Another kind of over-painting can be defined: *gray-overpainting*. This is meant in the sense that if we pick a point Q_i of the TI which has a prescribed gray-intensity level of $I_{gray,target}(Q_i) < saturation$, every intensity $I(Q_i)$ that causes $I(Q_i) > I_{gray,target}(Q_i)$ causes gray over-painting: this obviously causes non-realistic output images as well.

Thus, for each point Q_i of the canvas and TI, a distance $\Delta I_{gray,i}$ from the target to the starting gray-level. This translates in the following,

$$\Delta I_{gray,i} = I_{gray,target}(Q_i) - I_{gray,start}(Q_i). \tag{1}$$

Furthermore, for every Q_i we can define a saturation velocity $\overset{'}{I}(Q_i, Q_t)$, which is derived from the experimentally determined paint deposition rate law, illustrated in Section 3.2. This is as follows,

$$I'(Q_i, Q_t) = I'_{i,t} = I'(Q_i - Q_t) = \frac{\mathrm{d}I_{gray,i,t}}{\mathrm{d}t} \tag{2}$$

where Q_t is the barycenter of the stroke (tool-center).

It is important to recall that this relation can be sufficiently approximated as linear, hence the I' can be considered constant in the time domain, and function of Q_i alone.

Finally, S can be defined as a Cartesian map of the stroke of the spray-gun which was previously defined in polar coordinates $\rho(r, \gamma)$ in Section 3.1. The stroke map S is defined as a collection of $I'(Q_i)$ saturation velocities. Operatively, S is a matrix of coordinates (ζ_1, ζ_2) which arbitrarily discretizes the $\rho(r, \gamma)$ profile. In the algorithm a 2σ truncated Gaussian approximation is used.

4.2. Main structure of the algorithm

Since the painting process is split into different layers, at each stage the algorithm has to take into account the areas which are already saturated or have reached the prescribed intensity, in order to produce respectively neither over-painting nor gray-overpainting. In order to comply with this, critical points are determined at each step, and are used to limit the paint coating in order to prevent over-painting.

In Fig. 6 an overview of the proposed methodology is presented. The general layout is presented in Fig. 6a. The algorithm starts with the first layer at the maximum distance, it acquires the target image into the *core algorithm* and from the output image it investigates if there are any other regions still to be painted at this detail level; eventually it sub-iterates the process. If no regions are left, it can proceed to the next layer (at a closer distance) and iterate until the last layer is reached. The *core algorithm* is visible in Fig. 6b; it requires three main inputs: the deposition rate law, the stroke morphology (the size and shape of the stroke) and the *iteration target image* (*ITI*). The first two inputs are contained in the stroke map *S*. This image is obviously coincident to the target image in the first sub-iteration of the first layer, but at the following steps it is found following the diagram in Fig. 6c. An illustration of the described concept is given in Fig. 7.

The three main operational blocks are illustrated in Fig. 6b. The *Path-planner* block is based on an algorithm which uses offsets computation via Voronoi diagrams [15–17]. The boundaries for the calculation are chosen by a basic thresholding technique applied to the *Iteration target image*. The output of this process is a hierarchical collection of offsets, which is ultimately the *tool-path*; in the following, the tool-path will be referred as the vector *L*. The most important block is the *Timing algorithm*. This calculates the time the spray-gun must "rest" over each point of the tool-path, which easily translates into a *speed profile* along the path. The detailed procedures will be thoroughly explored in the

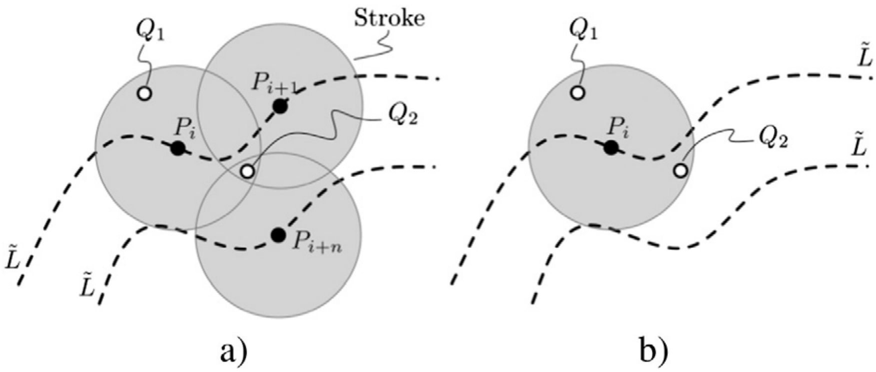

Fig. 9. Selection of critical points inside the timing algorithm. In a) the global potential critical points $K_{\eta,global}$ selection is shown, whereas in b) the local $K_{\eta,local}$ selection process is shown. These K_η critical points are the result of the selection between points Q_1 and Q_2.

Fig. 10. The target image, the "Lena" standard test image [19].

following section. The complete process in the elaboration of a generic layer is summarized in Fig. 8.

The *post-processing* generates the projected output image at the current layer, following the timing profile along the tool-path. This provides the next iteration image, as can be seen from Fig. 6b and c. This last step can be executed either by virtual representation or simulation (convolution of the stroke S with the tool-path), or by acquiring real-world results during the process (CCD camera, etc.).

4.3. Timing algorithm

As mentioned before, the timing algorithm is based on the solution of a linear system (LS) using the OLS method. This produces a speed profile for the EE to follow during the implementation of the tool-path provided by the path-planner block.

4.3.1. Linear system (LS) structure

Let the LS be in the following form:

$$At \leq B \tag{3}$$

where A is the matrix of coefficients, and B is the known terms vector. The t term is the vector of the variables; in order to better define it, let us consider a generic vectorial tool-path L generated by the *path-planner* block and juxtaposed over the image that we are processing (the ITI). By using the Bresenham line algorithm [18], we can rasterize this path; the resulting succession of points P_i corresponds to pixels belonging to the image and constitutes the discrete path $\tilde{L}$. Each of these pixels is effectively related to one variable in the LS, which is the time the end-effector is required to remain in that specific pixel boundaries. It follows that the number of variables in the LS can be (initially) assumed to be equal to the number of pixels belonging to the path $\tilde{L}$.

The matrix of coefficients A can be built according to the following rules: each column relates to a specific point $P_i \in \tilde{L}$, whereas each row is relative to each point Q_j of the ITI. Since the number of Q_j is exceptionally large, a significant number of equations would arise from this formulation. Therefore, an approach based on a restricted collection of critical points K_η is exploited. The definition and selection of these critical points will be discussed later in §4.3.2 and §4.3.3. The values are the previously described saturation velocities $I'(K_\eta, P_i) = I_{\eta,i}'$ of the critical points.

Finally, the known terms matrix B contains the target I_{gray} for the same critical Q_η points that were taken into account during the assembly of A, that is to say $I_{gray}(Q_\eta)$.

4.3.2. Critical points: Definition and determination

A critical point K_η is defined as a point which, during a specific layer painting process, is substantially more prone to over-painting or gray-overpainting than the majority of the other points of the ITI. Two kinds of K_η exist in our problem, and are related to their selection process: *global* critical points $K_{\eta,global}$ and *local* critical points $K_{\eta,local}$.

4.3.2.1. Global critical points. Let us start by illustrating the selection process of the *global* critical points by referring to Fig. 9a. As an example, let us consider the 2σ-truncated stroke centered in $P_i \in \tilde{L}$; let Q_1 and Q_2 be two generic points included in the boundaries of the stroke, which are potential critical points. The inclusion principle requires that $Q_j - P_i \leq R_{2\sigma}$, where $R_{2\sigma}$ is the radius of the 2σ-truncated stroke.

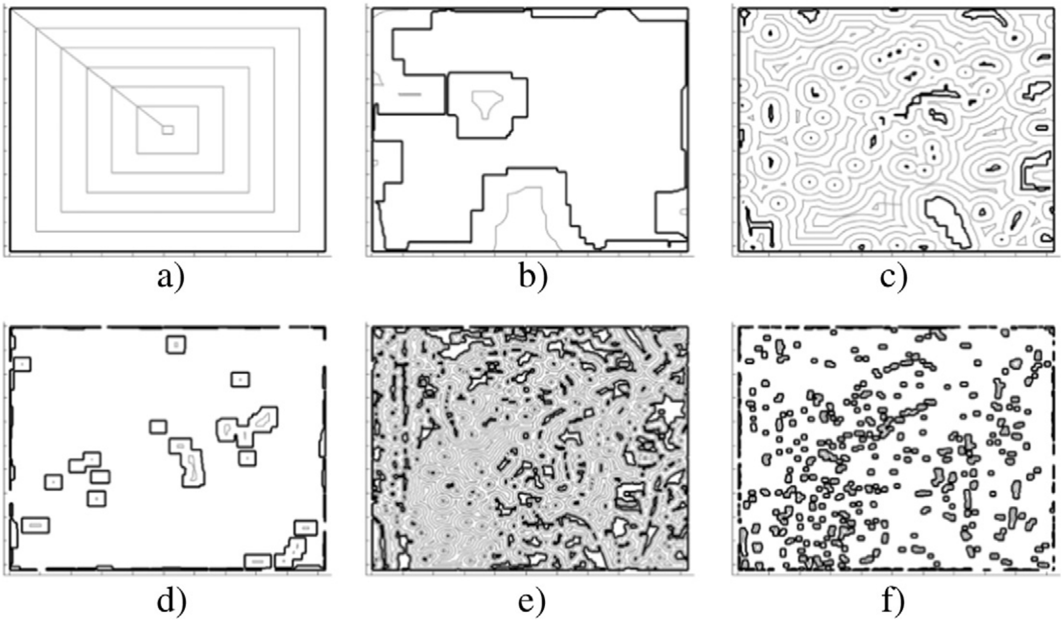

Fig. 11. Tool-paths generated via Voronoi diagram computation by the *path planner* algorithm; the area to be painted is shown with a solid black contour, whereas the tool-paths L are in gray. Since the general algorithm performs some sub-iterations, some of the steps presented in this figure belong to the same layer. In fact, *a*) and *b*) are relative to *Layer 1*, c) and d) to *Layer 2*, and *e*) and *f*) to *Layer 3*.

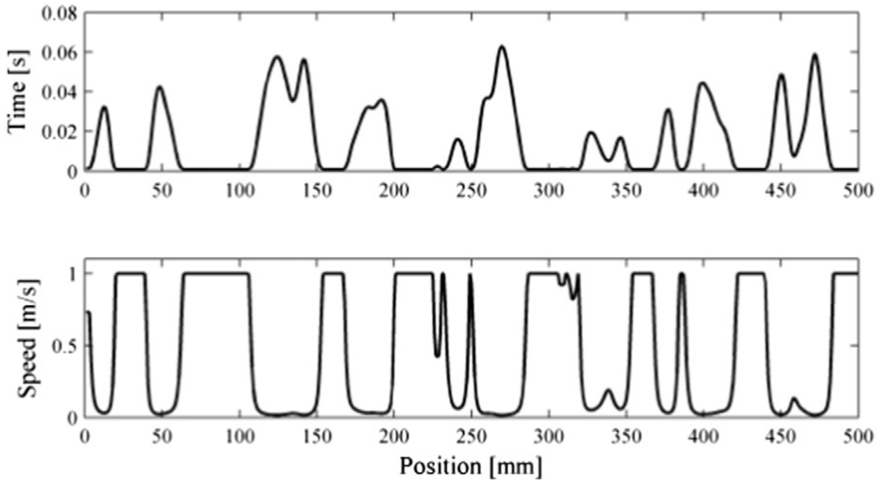

Fig. 12. Timing and speed profile of the tool-path generated by the *timing algorithm*. This is a partial result from the tool-path in Fig. 11a limited to the first 500 mm.

These are painted by different strokes. Specifically, whereas Q_1 is painted solely by the stroke originating by the EE placed in P_i, Q_2 is instead painted additionally by strokes centered in P_{i+1} and P_{i+n} as well (note: here, n is an arbitrary number ≥ 2). The algorithm has to decide which one is (more) critical. In order to comply with this, it has to calculate the saturation velocity $I'(Q_1) = I_1{}'$ and $\dot{I}(Q_2) = \dot{I_2}$ and the distance $\Delta I_{gray,1}$ and $\Delta I_{gray,2}$ at point Q_1 and Q_2, following Eq. (1) and Eq. (2).

At this point it can finally calculate the time necessary to reach $I_{gray,target}(Q_j)$ at points Q_1 and Q_2, as follows,

$$\tau_j = \tau(Q_j) = \Delta I_{gray,j} / \sum\nolimits_i (I'(Q_j, P_i)). \tag{4}$$

The selected critical point will then be $Q_{sel} = Q_j$, where Q_j is the point with the lowest time, i.e. $\tau_j = \min(\tau_1, \tau_2)$. Note that the calculation of τ_j is carried out for *every* point Q_j included in the boundaries of the 2σ-truncated stroke centered in P_i, thereby extending the reasoning explicated for points Q_1 and Q_2. It follows logically that Q_{sel} will belong to the stroke area of effect.

Operatively, this process is carried out by calculating the convolution C of the path $\tilde{L}$ with respect to the stroke of the spray-gun, as follows,

$$C(\xi_1, \xi_2) = \sum\nolimits_{\zeta_1=-\infty}^{+\infty} \sum\nolimits_{\zeta_2=-\infty}^{+\infty} S(\zeta_1, \zeta_2) L(\xi_1 - \zeta_1, \xi_2 - \zeta_2) \tag{5}$$

where, ξ_1 and ξ_2 indicate respectively the column and row of the generic ITI, and S is the already defined stroke characteristic, with ζ_1 and ζ_2 coordinates. The convolution C essentially maps the intensity per time values, that is to say, the saturation velocity $I'(Q_j, P_i)$, where $Q_{i,j}$ is the collection of all the points or pixels contained in the ITI. Note that C is the same for each point of the path L. In practical terms, C resembles a stroke that follows the tool-path $\tilde{L}$.

Finally, by solving Eq. (4) the algorithm is able to determine the timing map (τ_j) and thus select the relevant $Q_{sel,j}$, which is finally labeled as a global critical point $K_{\eta,global}$.

4.3.2.2. Local critical points. The selection process for *local* critical points is similar. The difference lies in the different way the intensity velocity map $I'(Q_j, P_i)$ is defined. In this case, in Eq. (5), the term $\tilde{L}$ is replaced by the term $\tilde{L}'$, which is the function of the point P_i we are considering. To clarify, let us consider point P_i; the matrix originating from term $\tilde{L}'$, in this case, will be equal to a matrix of zeroes with the only non-zero value corresponding to point P_i. The convolution will result in a single stroke centered in that point. The points that are being determined are thus labeled $K_{\eta,local}$.

4.3.2.3. Note. Please note that in this phase we assume the tool-path $\tilde{L}$ to have a prescribed constant speed of arbitrary value.

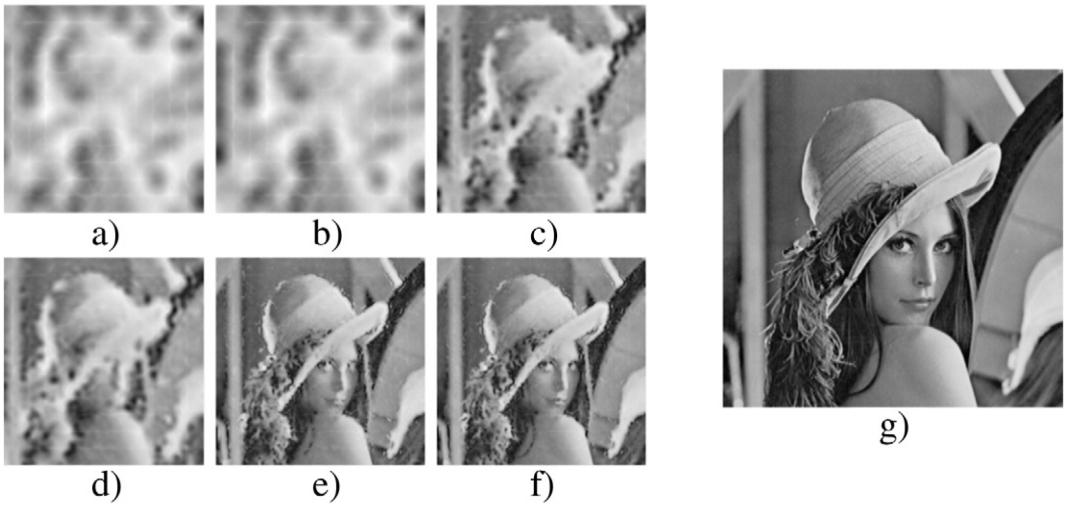

Fig. 13. Progress of the output image at completion of the various layers. *a*) and *b*) are relative to the 1st and 2nd sub-iterations of *Layer 1*, *c*) and *d*) belong to *Layer 2* and *e*) and *f*) to *Layer 3*. Finally g) is the target image, which is here shown for comparison.

4.3.3. Critical points: Selection

In this phase, inside a general layer, a pair of $K_{\eta,local,i}$ and $K_{\eta,global,i}$ critical points are determined for each point P_i of the tool-path $\tilde{L}$. The methodology prescribes that for each pair, only one is selected and the other is discarded. In order to comply with this, a LS containing both objects is assembled and preliminarily solved using OLS. For each point, the algorithm compares the value resulting from the convolution of S along $\tilde{L}$ (that is, the simulation of the painting process with the speed-profile resulting from the OLS LS's solution), and selects the point which shows the larger deviation from the target value defined in the ITI. This process is contextually carried out for all points $P_i \in \tilde{L}$; the result is a collection of K_η points, which is equal in number to points P_i.

Please note that this computationally intensive 2-step process is necessary and propaedeutical for the iterative phase, which will be illustrated in the following section.

4.3.4. Iterative refinement

This last phase consists in the adaptive iterator used to refine the results of the process described in the previous sections. The main objective is to drastically limit over-painting and gray-overpainting inconsistencies. The procedure is laid down in the following:

i. Definition and selection of critical points K_η,
ii. Assembly of the LS and solution with OLS,
iii. Analysis of the results and identification of over-paint areas (OPAs),
iv. Determination of the points P_i which associated critical point K_η is contained in an OPA,
v. 5% Reduction of the LS's known term B associated to all points P_i determined in the previous step, and
vi. Addition of one line in the LS relative to the most over-painted pixel in each OPA.
vii. Iteration of steps ii.–vi., until one of the following termination conditions is reached:
   - No more OPAs are detected.
   - The maximum number of iterations is reached (this is an arbitrary parameter).
   - The global over-painting effect (GOP) increases, which leads to divergence.

The GOP is defined as the cumulative sum of all over-painted pixels in the resulting image, when this is compared to the ITI.

5. A practical application, results and discussion

In this section we propose a practical case based on the famous 1973 “Lena” standard test image [19]. In Fig. 10a the actual 800 × 800 pixels image at a resolution of 1 pixel/mm is presented. For the test we used 3 progressively decreasing distances, which translated into the following stroke diameters and relative layers: layer 1, 81 pixels; layer 2, 27 pixels; and layer 3, 9 pixels. The shape of the stroke profile was approximated as a 2σ-truncated Gaussian curve. The “intensity” of the paint flow was assumed as 0.3 I/s, where $I \in [0, 1]$ is the intensity of the image. For the intensity, a global tolerance of $I_{tol} = 1/255$ was used in the algorithm. A speed limit of 1 m/s was imposed to the *timing algorithm*.

In Fig. 11 we show the results of the *path planner* block for each iteration and sub-iteration. It can be seen that the smaller the size of the stroke gets, the smaller the distance between the offsets gets, and the higher the number. This translates into quite long computation and execution times.

Furthermore, one can see that the plots in Fig. 11*a*, *c* and *e* are, in general, more packed. This is due to the fact that these are sub-iteration steps, and do not need to carry out much work in order to complete the corresponding layer.

The OLS method (*timing algorithm*) highlighted in Fig. 6 computes the time that the nozzle should rest over each point of the generated tool-path L. This easily translates into a speed profile, as shown in Fig. 12. Note that the speed remains always under 1 m/s, as prescribed.

Finally, in Fig. 13 is illustrated the progressive cumulative result after each of the tool-paths shown in the corresponding letter in Fig. 11 was fed into the *post-processing* block, along with the stroke parameters, the tool-path and the speed profile.

From the same figure one can immediately appreciate that the difference between the pairs *a*–*b*, *c*–*d* and *e*–*f* is not very significant; this is due to the sparseness of the sub-iterations tool-paths (which *b*, *d* and *f* are indeed). Nevertheless, the final output image (Fig. 13f) is very close to the target image (Fig. 13g), despite of the limitation of the number of iterations.

The time required for the robot to complete the painting was calculated using the speed-profile and resulted in 2692 s. Moreover, an estimate can be computed of the time that a Cartesian printer-like methodology would require completing the same painting with a similar resolution. Assuming that this Cartesian printer performs as our robot does when at the nearest distance from the canvas (smallest stroke size), the resulting execution time is 15,251 s, more than 5 times the one needed by the proposed approach.

6. Conclusion

The proposed methodology provides an efficient solution to perform photographically realistic paintings of digital images on large surfaces by means of a robotic system like an industrial spray-painting robot. The developed method promises to outperform normal printer-based methods where the result does not require a very high degree of detail, both in terms of costs and of execution time.

A general framework for the characterization of the spray-gun's spray morphology was described, which consisted in the measurement of the paint density profile shape and the paint deposition ratio. Both of these are critical to correctly operate with the presented algorithm.

This innovative approach promises several applications in the field of industrial and artistic spray-painting, allowing the complete automation of tasks previously limited to the world of craftsmanship or assisted robotics.

Important steps should be made in the future to further refine the methodology, especially in the field of optimization (tool-path, timing, etc.) and in the actual control of the robotic system (i.e. maximum acceleration control).

References

[1] T. Balkan, M.A.S. Arikan, Modeling of paint flow rate flux for circular paint sprays by using experimental paint thickness distribution, Mech. Res. Commun. 26 (1999) 609–617.

[2] K.R.J. Ellwood, J.L. Tardiff, S.M. Alaie, A simplified analysis method for correlating rotary atomizer performance on droplet size and coating appearance, J. Coat. Technol. Res. 11 (3) (2014) 303–309.

[3] D.C. Conner, A. Greenfield, P.N. Atkar, A.A. Rizzi, H. Choset, Paint deposition modeling for trajectory planning on automotive surfaces, IEEE Trans. Autom. Sci. Eng. 2 (2005) 381–391.

[4] Y. Chen, K. Chen, H. Yan, L. Wang, L. Zheng, Simulation analysis of coating at uniform velocity of robotic spray gun, Appl. Mech. Mater. 246–247 (2013) 1175–1180.

[5] M. Fogliati, D. Fontana, M. Garbero, M. Vanni, G. Baldi, R. Dondè, CFD simulation of paint deposition in an air spray process, J. Coat. Technol. Res. 3 (2) (2006) 117–125.

[6] P.T. Elliott, M.J. Steffenhagen, J.E. Glass, Spray applications: part III. Assessment of viscosities at high shear rates and dynamic uniaxial extensional viscosities on fan nozzle air sprayability, J. Coat. Technol. Res. 4 (4) (2007) 341–349.

[7] P.N. Atkar, A. Greenfield, D.C. Conner, H. Choset, A.A. Rizzi, Uniform coverage of automotive surface patches, Int. J. Robot. Res. 24 (2005) 883–898.

[8] S. Baldwin, Robotic paint automation: the pros and cons of using robots in your paint finishing system, Met. Finish. 108 (2010) 126–129.

[9] H. Chen, F. Thomas, L. Xiongzi, Automated industrial robot path planning for spray painting process: a review, 4th IEEE Conference on Automation Science and Engineering 2008, pp. 522–527.

[10] P. Haeberli, Paint by numbers: abstract image representation, Proceedings of the 17th Annual Conference on Computer Graphics and Interactive Techniques, vol. 24 1990, pp. 207–214.

[11] A. Hertzmann, Painterly rendering with curved brush strokes of multiple sizes, Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques 1998, pp. 453–460.

[12] J. Hays, I. Essa, Image and video based painterly animation, Proceedings of the 3rd International Symposium on Non-Photorealistic Animation and Rendering 2004, pp. 113–120.

[13] J. Collomosse, P. Hall, Painterly rendering using image salience, 20th Eurographics UK Conference, Eurographics Association 2002, pp. 112–118.

[14] H. Kang, S. Lee, C. Chui, Coherent line drawing, ACM Symposium on Non-Photorealistic Animation and Rendering 2007, pp. 43–50.

[15] M. Held, VRONI: an engineering approach to the reliable and efficient computation of Voronoi diagrams of points and line segments, Comput. Geom. Theory Appl. (2001) 95–123.

[16] M. Held, On the computational geometry of pocket machining, Lect. Notes Comput. Sci 500 (1991).

[17] G. Barequet, T.M. Dickerson, T.M. Goodrich, Voronoi diagrams for convex polygon-offset distance functions, Discret. Comput. Geom. 25 (2) (2001) 271–291.

[18] J.E. Bresenham, Algorithm for computer control of a digital plotter, IBM Syst. J. 4 (1) (1965) 25–30, http://dx.doi.org/10.1147/sj.41.0025.

[19] David C. Munson Jr., A note on Lena, IEEE Trans. Image Process. 5 (1) (1996).