drozBot: Using Ergodic Control to Draw Portraits

Tobias Löw, Jérémy Maceiras and Sylvain Calinon

***Abstract*—We present *drozBot: le robot portraitiste*, a robotic system that draws artistic portraits of people. The input images for the portrait are taken interactively by the robot itself. We formulate the problem of drawing portraits as a problem of coverage which is then solved by an ergodic control algorithm to compute the strokes. The ergodic computation of the strokes for the portrait gives an artistic look to them. The specific ergodic control algorithm that we chose is inspired by the heat equation. We employed a 7-axis Franka Emika robot for the physical drawings and used an optimal control strategy to generate joint angle commands. We explain the influence of the different hyperparameters and show the importance of the image processing steps. The attractiveness of the results was evaluated by conducting a survey where we asked the participants to rank the portraits produced by different algorithms.**

***Index Terms*—Art and Entertainment Robotics, Motion Control**

I. INTRODUCTION

In this paper we present a robotic system that draws portraits of people. The idea of a portrait drawing robot has a long tradition: it goes back to the 18th century when the Swiss watchmaker Pierre Jaquet-Droz invented his automaton *le dessinateur* [1]. This machine was reprogrammable by mechanically shifting gears, meaning it was possible to have it draw different pictures. We named our robotic system *drozBot* as an homage to Pierre Jaquet-Droz. We want to follow in this tradition of using robots for art, since 250 years later the topic of letting robots draw portraits of human faces is still part of ongoing research.

A prominent example of a portrait drawing robot is Paul, the robot [1]. It is a custom designed robot that is capable of drawing the portraits of humans after taking a photograph of them. The technique behind Paul is to identify the salient lines in the image in different directions and then drawing and overlaying them to recreate the image of the person. The shading process relies on a visual feedback of the drawing. A different approach was taken in [2], where a humanoid robot drew artistic portraits of humans in a fashion that was as human-like as possible. In order to achieve that the image is binarized at different levels of gray and for each layer a trajectory is planned to cover the dark areas. The final drawing is then formed by overlaying all trajectories. Both approaches use image processing techniques on the raw image before calculating the trajectory for the drawing motion of the robot. In [3], Chen *et al.* previously used a Franka Emika robot for remote portrait drawing through teleoperation. Kuka robots have also been used to draw faces [4] and images on arbitrary surfaces [5].

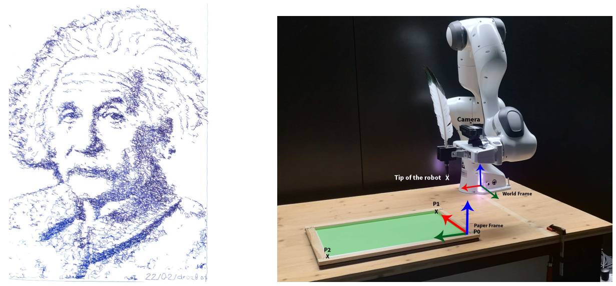

Fig. 1: Experimental setup with a 7DOF Franka Emika robot and an embedded camera and a portrait that was drawn by drozBot.

Next to drawing, robots also have become popular tools for painting. In [6] a robotic system was presented that uses watercolour to create artistic renderings. In [7] the authors taught machines to paint like human painters using deep reinforcement learning. Robotic systems for painting tasks are also of interest in industrial settings [8]. The advantage of using robots there lies in preventing the exposure of workers to harmful substances that are used for example for spray painting.

Creating art with machines is popular across various research domains. Some effort is being put into finding ways to create artistic renderings of human faces such as caricatures. Recently Generative Adversarial Networks (GAN) have become more popular to create such artistic visual renderings. Examples for the generation of caricatures are cariGAN [9] and WarpGAN [10]. These works focus on the generation of artistic renderings though and thus do not provide physical drawings.

In [11] the authors postulate that the merging of robots and arts has reached a state where we can speak of the machine as artist. In this sense it is now required to deepen our understanding of how robots can create works that have an aesthetic appeal. Art is also theorized to make robots more relatable to humans and help increase the acceptance of robots in everyday human environments [12]. Robots also become increasingly prominent in other forms of art like dancing [13].

At the Robotics, Science and Systems Conference 2021 there was a dedicated exhibition on robotics in arts. [2] The ex-

Manuscript received: February, 24, 2022; Revised May, 22, 2022; Accepted June, 16, 2022.

This paper was recommended for publication by Editor Youngjin Choi upon evaluation of the Associate Editor and Reviewers' comments. This work was supported in part by the Swiss National Science Foundation (NATAI project, Agora grant 191634), and in part by the European Commission's Horizon 2020 Programme through the COLLABORATE project (https://collaborate-project.eu/, grant 820767) and the MEMMO project (https://www.memmo-project.eu/, grant 780684).

The authors are with the Idiap Research Institute, Martigny, Switzerland and with EPFL, Lausanne, Switzerland. (tobias.loew@idiap.ch)

Digital Object Identifier (DOI): see top of this page.

[1] https://en.wikipedia.org/wiki/Jaquet-Droz_automata

[2] https://sites.gatech.edu/robotic-art-workshop/

hibition included dancing robots and painting robots. Exhibits that are related to this work include the AniPainter [14], the three stage drawing transfer [15] and the graffitized alphabets [16].

In this paper we propose to use an approach from control theory to draw realistic portraits, namely ergodic control. Ergodic control has been previously used for portrait drawing robots [17], [18]. However in this paper the authors used an algorithm based on Spectral Multi-scale Coverage (SMC), which was previously presented in [19]. SMC with regards to artistic drawings has the disadvantage that it often switches between the modes of the distribution, which causes many unwanted lines to be drawn. These lines not only look very noisy, but also keep the drawn image from being a good artistic rendering of the person. Instead we chose to utilize an ergodic control technique that is based on the heat equation. The algorithm called Heat Equation Driven Area Coverage (HEDAC) was first presented in [20] and has been successfully applied to autonomous spraying in an agricultural setting [21] and search rescue missions [22] with the specific case of the MH370 [23]. We will present how we adapted this algorithm in order for a robot manipulator to be able to draw realistic portraits.

In this paper we argue that the ergodic control technique that is used to compute the strokes to draw portrait is very similar to the artistic technique of *doodling* or *scribbling*. The concept of ergodicity in drawings also fits well with Paul Klee describing drawing as *an active line on a walk* [24]. Current research in that area suggests that more natural looking strokes lead to aesthetically more pleasing drawings [25]. The natural look is an effect of the second-order system that was employed in the computation of the trajectories. Furthermore the strokes are sorted such that a fluid motion between the strokes can be created.

From the gallery of drawings that is presented in the experiments section, it can be seen that from a close distance the lines seem to look random and it is hard to understand the image. Only viewing the image from further it becomes very clear whom the portrait is depicting. This effect introduces a certain abstractness to the drawing and makes it interesting for the spectators to view the portraits from different distances and discover its effects on how the portrait is perceived. The portraits that drozBot is drawing are different from portraits drawn by human artists in the sense that drozBot's portraits are drawn in an optimal way. This notion of optimality comes from the ergodic control that is taking actions that are minimizing a certain metric.

The contribution that was made in this paper is a robotic system that is capable of drawing artistic portraits. This system is showcasing a new and unconventional usage of known algorithms. All steps in the process, i.e. the interactive capturing of the image, the ergodic computation of the strokes, the drawing with the 7DOF robotic arm, are aimed to look as natural as possible. drozBot was exposed to the public during the Idiap 30th anniversary celebration where visitors had the opportunity to spectate it, have their portraits drawn, observe the drawing process and examine the gallery of already drawn portraits. Furthermore drozBot is going to be a part of an exhibition about artificial intelligence at the *musée de la main* in Lausanne for the duration of one year.

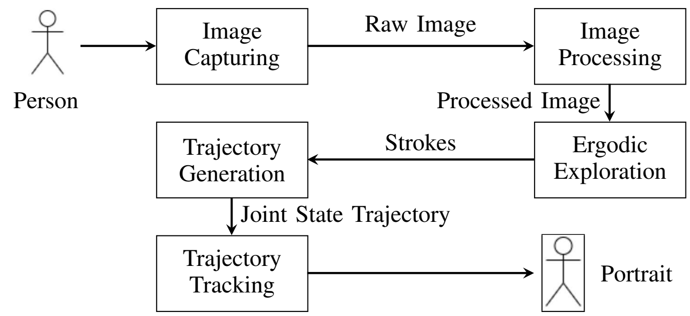

Fig. 2: Processing pipeline.

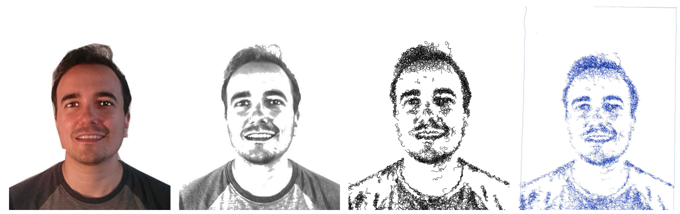

Fig. 3: Intermediary outputs of the processing pipeline. From left to right: Original image taken by the robot. Processed image that is used as input to the coverage algorithm (inverted). Strokes that were computed using the coverage algorithm. Final portrait drawn by the robot.

II. Creation of the Portrait

In this section we describe the process of creating portraits from photographs that are taken by the robot.

The process starts with the robot capturing the photograph of the person. Afterwards the image is processed to highlight important features in a grayscale version of the image. The strokes are then computed using a heat equation coverage algorithm that uses the idea of ergodic control. In the last step the joint angle trajectory is computed and executed by the robot.

As the drawing does not require a visual feedback, the proposed implementation follows the sequential pipeline as shown in Figure 2. Figure 3 displays the intermediary results that are achieved after each processing step.

A. Image Capturing

The principal input of the pipeline is an image that is taken in an adaptive manner by a robot. The person is required to sit in front of the robot and the robot will move to adjust for their height automatically. It uses the camera attached to its end-effector to track the different facial features and ensure that the face is correctly aligned in the image frame with a given margin. The margin ensures that the robot does not try to compensate all small changes in the face position that can happen because of small motions of the person or noise in the face detection. The detection of the facial features is done using the face detector of the dlib library [26].

To facilitate the image processing step, the picture is taken in a controlled environment, i.e. the user is seated in front of a white wall. Also, we ensure that the image will nicely fit on the paper by taking an image with the same width over height ratio as ISO 216 A series paper format.

B. Image Processing

In order to have an algorithm that is unbiased and is capable of drawing all skin and hair colors, an adaptive histogram equalization is performed (AHE) [27]. The purpose of performing an AHE instead of standard histogram equalization is to remove the bias introduced by the white background of the image. AHE normalizes multiple histograms computed from distinct parts of the image. After this step, since we are drawing with a single color, the image is converted to grayscale.

To increase the sparsity of the drawings a brightness and contrast adjustment is performed on the image. This processing step highlights the different features of the face and ensures that the exploration algorithm will not lose time by exploring feature-less areas of the face. Having access to the facial features from the previous processing step it is also possible to highlight those features specifically by increasing their brightness in order to increase their importance for the coverage algorithm, which results in more strokes and thus more details.

We reserve a small white rectangle in the bottom right of the drawing for the robot to sign its work.

C. Ergodic Exploration

We view the problem of drawing a portrait as a coverage problem. To this end we utilize an ergodic control technique based on the heat equation that was presented in [20]. In this section we will explain how we adapted this approach for drawing portraits and the advantages of ergodicity when doing so. The full process is shown in Algorithm 1.

The input image is viewed as the target density $p(\boldsymbol{x})$ on the bounded domain Ω. The problem of area coverage is then defined as minimizing the ℓ^2 norm of the error between the target density $p(\boldsymbol{x})$ and the achieved coverage density $c(\boldsymbol{x}, t)$ at any given time t

$$E(t) = \|e(\boldsymbol{x}, t)\|_2, \tag{1}$$

$$\text{with} \quad e(\boldsymbol{x}, t) = p(\boldsymbol{x}) - c(\boldsymbol{x}, t). \tag{2}$$

The algorithm uses multiple virtual agents that are moving over a heat source, which is depending on the target distribution $p(\boldsymbol{x})$, in order to reduce its temperature to zero. Note that in this case the term agent does not refer to the robot, but to a 2D pointmass that is moving on the canvas. Hence the trajectory of an agent represents one stroke that is then drawn by the robot.

In a first step we form the 2-dimensional heat equation

$$\frac{\partial}{\partial t} u(\boldsymbol{x}, t) = \alpha \frac{\partial^2}{\partial \boldsymbol{x}^2} u(\boldsymbol{x}, t) + \beta s(\boldsymbol{x}, t) - \gamma a(\boldsymbol{x}, t), \tag{3}$$

with the initial condition for the temperature field $u(\boldsymbol{x}, t)$ being

$$u(\boldsymbol{x}, 0) = p(\boldsymbol{x}) \tag{4}$$

and the boundary condition

$$\frac{\partial}{\partial \boldsymbol{n}} u(\boldsymbol{x}, t) = 0, \quad \text{on } \partial\Omega, \tag{5}$$

where $\boldsymbol{n}$ is a vector normal to $\partial\Omega$ and $\boldsymbol{x}$ is the position on Ω. $\alpha > 0$ is the thermal diffusivity, which is a measure of how fast the heat is spreading through the material. $\beta > 0$ and $\gamma > 0$ are tunable parameters that regulate the strength of the heat source $s(\boldsymbol{x}, t)$ and sink $a(\boldsymbol{x}, t)$, respectively.

The solution of the heat equation is the temperature field $u(\boldsymbol{x}, t)$. The temperature gradient is then used to the control motion of the agents with acceleration commands. We employ double integrator systems as the agents motion model

$$\frac{d^2}{dt^2} \boldsymbol{x}_i(t) = \nabla u(\boldsymbol{x}_i(t), t), \tag{6}$$

where $\boldsymbol{x}_i(t)$ is the position of the i-th agent at time t. The agents have a maximum acceleration and a maximum velocity, the state and the command are therefore thresholded.

The achieved coverage density along the trajectories of the agents is defined as the integral of a radial basis function along all agents trajectories.

$$\tilde{c}(\boldsymbol{x}, t) = \frac{1}{Nt} \sum_{i=1}^{N} \int_0^t \phi(\boldsymbol{x} - \boldsymbol{x}_i(\tau)) d\tau, \tag{7}$$

with ϕ being a Gaussian radial basis functions (RBF) defined as

$$\phi(\boldsymbol{x}) = e^{-(\epsilon \boldsymbol{x})^2}. \tag{8}$$

The shape parameter ϵ of the RBF can be seen as the inverse of the radius of the pen that is used for drawing. The coverage density is normalized over the domain

$$c(\boldsymbol{x}, t) = \frac{\tilde{c}(\boldsymbol{x}, t)}{\int_\Omega \tilde{c}(\boldsymbol{x}, t) d\boldsymbol{x}}. \tag{9}$$

The heat source $s(\boldsymbol{x}, t)$ at the current timestep t is computed by squaring the error between the target density $p(\boldsymbol{x})$ and the coverage density $c(\boldsymbol{x}, t)$ as defined in Equation (2). The source term is normalized across the domain for consistency. It can therefore be found as

$$s(\boldsymbol{x}, t) = \frac{\tilde{s}(\boldsymbol{x}, t)}{\frac{1}{|\Omega|} \int_\Omega \tilde{s}(\boldsymbol{x}, t)}, \tag{10}$$

$$\text{with} \quad \tilde{s}(\boldsymbol{x}, t) = \max(e(\boldsymbol{x}, t), 0)^2. \tag{11}$$

From Equation (11) it can be seen that areas with a higher value for coverage density than for the target density result in a value of zero. This helps with avoiding to over-explore sections that have already been covered sufficiently.

The heat sink $a(\boldsymbol{x}, t)$ provides a way to introduce local cooling around the agents. Local cooling has a repelling effect and thus causes multiple agents to avoid each other. It is defined as

$$a(\boldsymbol{x}, t) = \frac{\tilde{a}(\boldsymbol{x}, t)}{\frac{1}{|\Omega|} \int_\Omega \tilde{a}(\boldsymbol{x}, t)}, \tag{12}$$

$$\text{with} \quad \tilde{a}(\boldsymbol{x}, t) = \sum_{i=1}^{N} \phi\big(\boldsymbol{x} - \boldsymbol{x}_i(t)\big). \tag{13}$$

We solve the heat equation for T timesteps, then we re-initialize the agents at a new position and reset the achieved coverage density. The current achieved coverage is however deducted from the target distribution $p(\boldsymbol{x})$. This lets the agents cover the space with many short strokes. Due to the ergodic nature of the algorithm the strokes get an artistic feeling and look very human, which makes for an aesthetically pleasing drawing. These lines look like doodles, which is an artistic technique that is often employed by human artists.

We employ Gibbs sampling [28] to initialize the agents, which is a special case of the Metropolis-Hastings algorithm, a well known Markov-Chain Monte-Carlo (MCMC) technique. The burn-in period can be very short, since the sampler will converge to the relevant distribution quickly. The MCMC technique itself is ergodic in nature too [29].

The authors of HEDAC also showed in [20] that the algorithm minimizes the ergodic metric. The use of ergodic control for the coverage planning of the strokes gives the drawings a notion of optimality. Optimal in the sense that at each iteration the action minimizing the difference between the target distribution and the coverage density is chosen.

Algorithm 1: Stroke computation

Input: target distribution $p(\boldsymbol{x}, k)$, number of agents N, number of timesteps T, number of iterations K
Output: NK agent trajectories

normalize target distribution $m(\boldsymbol{x}, k)$
while $k < K$ *and* $\|p(\boldsymbol{x}, k)\| > 0$ **do**
  initialize N agents using Gibbs Sampling
  for T *timesteps* **do**
    compute coverage density $c(\boldsymbol{x}, t)$ with Eq. (9)
    compute heat sink $a(\boldsymbol{x}, t)$ with Eq. (13)
    compute heat source $s(\boldsymbol{x}, t)$ with Eq. (10)
    solve heat equation in Eq. (3)
    for *all* N *agents* **do**
      update position using Eq. (6)
  $p(\boldsymbol{x}, k+1) \leftarrow p(\boldsymbol{x}, k) - c(\boldsymbol{x}, T)$

D. Trajectory Generation

The coverage algorithm returns a set of trajectories that are expressed in the two dimensional image frame. We use a 2D min-max normalization on the set of trajectories to fit the image to the paper. Each trajectory corresponds to the displacement of one agent for a given time which in turn corresponds to one stroke that will be drawn by the robot. In order to combine them into a single Cartesian trajectory, we connect the end of each stroke with the beginning of its closest neighbor in a greedy fashion. This arc motion is computed by a quadratic curve with an offset on the z axis that depends on the distance between the strokes to delimit the strokes on the paper. If needed, we then upsample the trajectory to ensure that for a given dt between two points a maximum speed is not surpassed, which prevents inaccuracy in the Cartesian motion. The selected maximum speed thus presents a trade-off between drawing time and accuracy, which was determined empirically.

The full trajectory is now transformed from the paper frame to the robot base frame. The trajectory now represents the position of the tip of the pen in time. We assume that the pen is able to draw as long as it is in contact with the paper, therefore there is some variability allowed in the orientation.

From the Cartesian trajectory we compute the corresponding joint state trajectory by using the iterative Linear Quadratic Regulator (iLQR) [30] as a motion planner. It iteratively minimizes the cost function

$$J = \sum_{t=0}^{T-1} \|\boldsymbol{d}(\boldsymbol{\mu}_t, \boldsymbol{f}(\boldsymbol{q}_t))\|^2_{\boldsymbol{Q}_t} + \|\boldsymbol{u}_t\|^2_{\boldsymbol{R}}, \tag{14}$$

$$\text{subject to} \quad \boldsymbol{q}_{t+1} = \boldsymbol{q}_t + \boldsymbol{u}_t \mathrm{d}t. \tag{15}$$

with respect to the control command $\boldsymbol{u}_t$ by considering the problem as a normal LQR problem by estimating the system with a first-order Taylor series and the cost with a second-order Taylor series. T is the horizon of the trajectory, $\boldsymbol{\mu}_t$ is the desired position and orientation at timestep t, $\boldsymbol{q}_t$ is the actual joint position at timestep t, $\boldsymbol{f}(\boldsymbol{q}_t)$ is the forward kinematics function of the robot, $\boldsymbol{d}(\boldsymbol{\mu}_t, \boldsymbol{f}(\boldsymbol{q}_t))$ is the error function The precision matrix $\boldsymbol{Q}_t$ for timestep t, is a positive definite matrix that is used to set the required precision and correlation for each target pose. We use it to allow for some flexibility in the pen orientation, thus we set it with a relatively low precision for the orientation around the z-axis. This implicitly tells the system that the orientation of the end-effector is not important as long as the tip of the pen is on the paper plane which is parallel to the X-Y-plane. $\boldsymbol{R}$ is the constant regulation matrix that enforces a smooth trajectory by penalizing large control commands (i.e. large accelerations).

E. Compliant Trajectory Tracking

To track the joint states trajectory on the robot, we sequentially compute the torque commands to send to the robot by using an impedance controller running at $1000\,\mathrm{Hz}$

$$\boldsymbol{\tau} = \boldsymbol{K_p}(\boldsymbol{q}_{\text{desired}} - \boldsymbol{q}_{\text{actual}}) + \boldsymbol{K_v}(\dot{\boldsymbol{q}}_{\text{desired}} - \dot{\boldsymbol{q}}_{\text{actual}}), \tag{16}$$

in which the desired joint velocities are computed with finite differences. The gains of this impedance controller let us control the pressure of the pen on the paper that the robot uses while drawing. Since for an ink pen the pressure directly determines the saturation of the colour on paper, the gains have a direct impact on the resulting portrait. Thus they can be seen as an artistic tuning parameter.

III. Experiments

drozBot was presented to a broad public during a 2-day open house event to celebrate Idiap's 30th anniversary, where it created artworks of many people. In this section we present those portraits and explain in further detail the practical implementations of the robotic system. This event was used as a test bed for the robot to be showcased during a 1-year exhibition at a museum.

A. Implementation Details

In our experiments, we use a Franka Emika robot, a redundant manipulator with great manipulation capabilities, which mimics a human arm with 7 degrees of freedom, counting the shoulder, elbow and wrist joints. This leads to a more natural looking drawing process and allows the robot to draw in a similar manner as a human. The extra degree of freedom allows us to constrain the drawing orientation in order to ensure that the tip of ink pen is kept in constant contact with the paper.

For the camera, we use the OpenCV AI Kit: OAK-D [3] that provides a 4k resolution and the ability to run small neural networks directly on it.

In practice we used 50 agents with 13 timesteps and 50 iterations for the ergodic control to cover the area. This resulted in 2500 strokes required for a drawing. The number of strokes was determined empirically and presents a trade-off between the time to draw the portrait and its quality, which is determined by the density of the strokes that is necessary in order to make the portrait recognizable. The computation time of those strokes takes 5 to 6 seconds. Afterwards the time it takes for one drawing to be completed is approximately 40 minutes. Thus the computation time of the strokes is negligible compared to the physical drawing time. During this time the spectators are given the chance to view the drawing process and to see how the features of person being drawn are slowly developed, see the video accompanying the paper.

B. Image Processing

In this section we are presenting the importance of the image processing steps that were explained in Section II-B. We computed the renderings of two persons with dark and light skin, respectively. The results are shown with and without the adaptive histogram normalization and the brightness and contrast adjustment in Figure 4.

C. Artistic Exploration

The coverage algorithm presented in Section II-C features several parameters that have effects not only on the quality of the resulting portrait but also on its artistic style.

One parameter that obviously has a large influence on the outcome of the portrait is the number of strokes. In our experiments we kept the number of agents constant at $N = 50$ and modify the number of iterations K. In Figure 5 the impact of changing the number of strokes can be seen. Figure 5 with $K = 50$ corresponds to the number of strokes that we used for drawing the portraits.

Another important parameter is the number of timesteps that each agent gets to travel before being re-initialized. From our experiments we discovered that while increasing the number of timesteps led to certain areas being more densely covered, the portrait itself lost important details. This comparison can be seen in Figure 5, here the other parameters are adjusted accordingly such that both images have the same number of points in the stroke trajectories in order to ensure a fair comparison.

[3]https://store.opencv.ai/products/oak-d

Fig. 4: Renderings showing the results with (center) and without (right) the image processing steps. The images were taken from Unsplash [31].

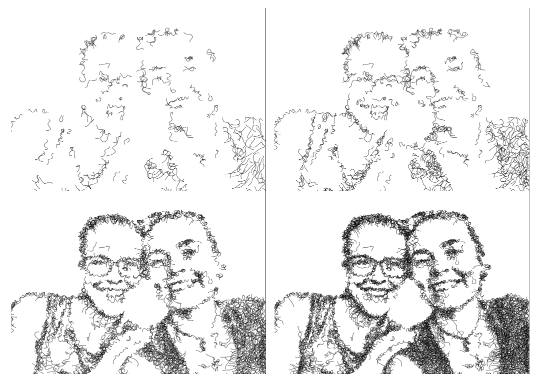

Fig. 5: Computed strokes after 5, 10, 30 and 50 iterations, corresponding to 250, 500, 1500 and 2500 strokes, respectively. The number of agents is kept constant across all images at $N = 50$.

Fig. 6: Comparison of the number of timestep for a single stroke (10 and 50 timesteps).

The last hyperparameter that we want to explicitly highlight is the shape parameter ϵ of the RBF in Equation (8). As mentioned ϵ can be seen as the inverted radius of the pen that is used for drawing. Figure 6 compares the renderings corresponding to two different pen thicknesses. Note that $\epsilon = 0.33$ means that the pen is three times thicker than with

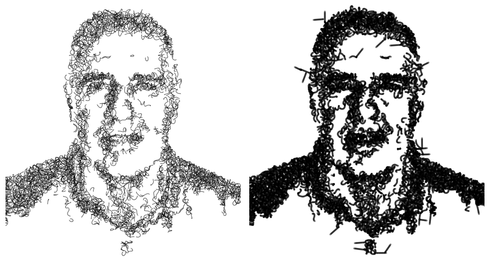

Fig. 7: Comparing the influence of the thickness of the pen, where a thicker pen causes faster convergence but results in less details ($\epsilon = 1.0$ and $\epsilon = 0.33$ that actually converged at $k = 42$ iterations).

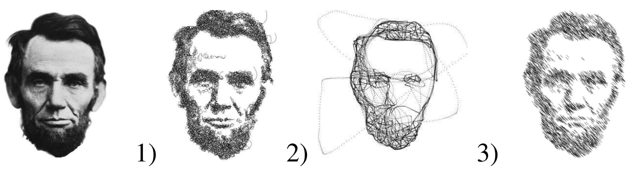

Fig. 8: Portrait generated with 1) our method compared to 2) the method presented in [17] 3) a rendering using fixed directional strokes.

$\epsilon = 1.0$.

D. Comparison to other ergodic drawing methods

In this section we compare our proposed method for drawing portraits with existing methods based on ergodic control that was presented in [17]. Furthermore we are showing a rendering of a portrait that does not use ergodic control to generate, but uses strokes with a fixed length and direction. The location of those strokes is determined by Gibbs sampling based on the processed image. The results can be seen in Figure 8. It can clearly be seen that our method outperforms the existing ergodic control techniques in terms of drawing realistic portraits.

E. Survey to evaluate the attractiveness of the generated drawings

In order to evaluate how pleasant the drawn portraits are, we conducted a survey with 35 people. In this survey we asked three different questions.

1) Ranking the pleasantness of the portraits generated with our method compared to other drawing methods.
2) Comparing several portraits that were drawn with our method by using different parameters and ranking them based on aesthetic pleasantness.
3) Comparing the portrait generated with our method to the original photo and rating how recognizable the person is.

Each question was repeated several times using different photos. The photos were chosen, such that a variety of ethnicities and genders were shown. The original images were taken from Unsplash [31] and the Flickr-Faces-HQ dataset (FFHQ) [32]. We modified the originals by cropping them to only show the faces, removing the background and replacing it with a uniform white colour and then resizing the images, such that all images have the same dimensions of 480x600 pixels.

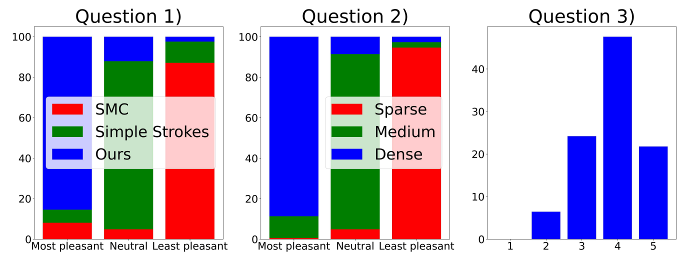

Fig. 9: Results of the survey. The scale in Q3 rates the drawing from unrecognizable (1) to very recognizable (5).

The results of the survey are shown in Figure 9. The results of Q1 show that the participants generally preferred our drawing method over the baselines ([17] and fixed strokes renderings). The comparison of the hyperparameters in Q2 revealed that people preferred drawings with more strokes, which can be explained by the denser portraits. However, in order to limit the drawing time, the number of strokes that was selected for the drawings with the robot has been reduced. The number of strokes could be increased by using a robot that remains accurate at higher velocities. The answers to Q3 show that the drawings remain recognizable, which was one of the goals of this system.

F. Experimental Results

In this section we present the drawings that drozBot did during the public event. All drawings were drawn on ISO 216 A3 paper in portrait mode.

All drawings have in common that when viewed from close the lines look very random. Only when viewing them from a distance do the lines blend together to form the image of the person that is being portrayed. This can be seen as an artistic feature of the created drawings.

IV. Conclusion

We presented drozBot, le robot portraitiste, a robotic system that draws artistic portraits of people which is achieved by means of an ergodic control technique that mimics the doodling of human artists. We identified several hyperparameters that affect the aesthetics of the portrait drawings. These hyperparameters could be exploited by a human artist to create artworks interactively.

Given the nature of this work, it is hard to determine quantitative metrics to evaluate the artistic aspect, since art can be very subjective and could be a topic of research on its own. We therefore compared to previous techniques using ergodic control on a qualitative level and showed that our algorithm produces more realistic looking portraits. In order to evaluate the attractiveness of the generated drawings, we conducted a survey that asked participants to rank our method against other methods. The survey also concluded that the portraits produced by drozBot generally have a high recognition rate.

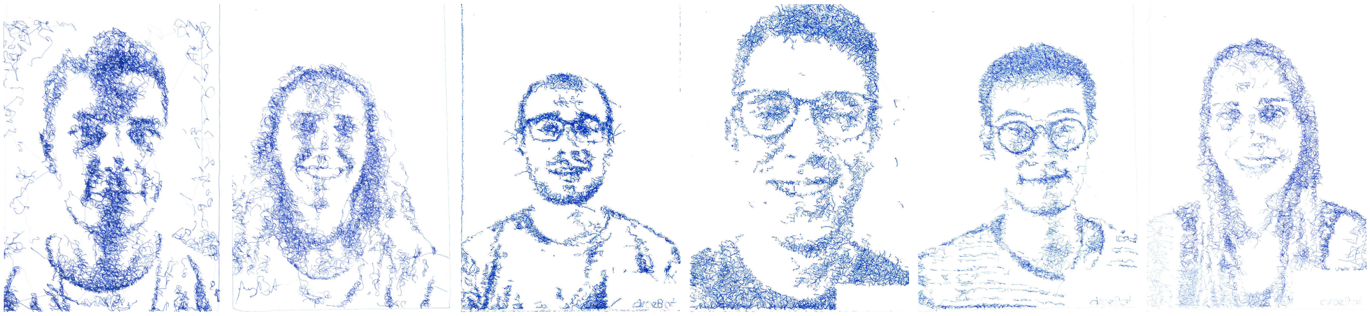

Fig. 10: Scans of selected portraits drawn by *drozBot* during the public event.

Possible extensions of this work can be done by exploiting the shape parameter of the RBF. This parameter could be used to incorporate pens of different width into the algorithm. The change in width could in that case either come from applying a varying force on the pen or changing its orientation in the case of a non-uniform pen. Latter would possibly also require to change the RBF to a more suitable representation. This improvement could be achieved by employing a closed-loop controller for the force in the vertical direction.

References

[1] P. Tresset and F. Fol Leymarie, "Portrait drawing by Paul the robot," *Computers & Graphics*, vol. 37, no. 5, pp. 348–363, August 2013.

[2] S. Calinon, J. Epiney, and A. Billard, "A humanoid robot drawing human portraits," in *5th IEEE-RAS International Conference on Humanoid Robots, 2005.* IEEE, 2005, pp. 161–166.

[3] L. Chen, A. Swikir, and S. Haddadin, "Drawing elon musk: A robot avatar for remote manipulation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021.

[4] M. Pichkalev, R. Lavrenov, R. Safin, and K.-H. Hsia, "Face drawing by kuka 6 axis robot manipulator," *2019 12th International Conference on Developments in eSystems Engineering (DeSE)*, pp. 709–714, 2019.

[5] T. L. D. Song and Y. J. Kim, "Artistic pen drawing on an arbitrary surface using an impedance-controlled robot," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018.

[6] L. Scalera, S. Seriani, A. Gasparetto, and P. Gallina, "Watercolour Robotic Painting: A Novel Automatic System for Artistic Rendering," *Journal of Intelligent & Robotic Systems*, vol. 95, no. 3-4, pp. 871–886, September 2019.

[7] Z. Huang, W. Heng, and S. Zhou, "Learning to paint with model-based deep reinforcement learning," *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8708–8717, 2019.

[8] H. Chen, T. Fuhlbrigge, and X. Li, "Automated industrial robot path planning for spray painting process: A review," in *2008 IEEE International Conference on Automation Science and Engineering*, August 2008, pp. 522–527.

[9] K. Cao, J. Liao, and L. Yuan, "CariGANs: Unpaired photo-to-caricature translation," *ACM Transactions on Graphics*, vol. 37, no. 6, pp. 1–14, January 2019.

[10] Y. Shi, D. Deb, and A. K. Jain, "Warpgan: Automatic caricature generation," in *CVPR*, 2019.

[11] G. Smith and F. F. Leymarie, "The Machine as Artist: An Introduction," *Arts*, vol. 6, no. 4, p. 5, April 2017.

[12] M. Cooney, "Robot Art, in the Eye of the Beholder?: Personalized Metaphors Facilitate Communication of Emotions and Creativity," *Frontiers in Robotics and AI*, vol. 8, p. 219, 2021.

[13] A. LaViers and M. Egerstedt, Eds., *Controls and Art.* Springer International Publishing, 2014.

[14] AniPainter – RSS 2021 Workshop on Robotics x Arts. [Online]. Available: https://sites.gatech.edu/robotic-art-workshop/2021/07/02/anipainter/

[15] Three Stage Drawing Transfer – RSS 2021 Workshop on Robotics x Arts. [Online]. Available: https://sites.gatech.edu/robotic-art-workshop/2021/07/13/three-stage-drawing-transfer/

[16] Graffitized alphabets – RSS 2021 Workshop on Robotics x Arts. [Online]. Available: https://sites.gatech.edu/robotic-art-workshop/2021/07/07/graffitized-alphabets/

[17] A. Prabhakar, A. Mavrommati, J. Schultz, and T. Murphey, *Autonomous Visual Rendering using Physical Motion*, ser. Springer Proceedings in Advanced Robotics. Springer Science and Business Media B.V., 2020, pp. 80–95.

[18] T. Sahai, G. Mathew, and A. Surana, "A chaotic dynamical system that paints and samples," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 10 760–10 765, 2017.

[19] G. Mathew and I. Mezic, "Spectral Multiscale Coverage: A uniform coverage algorithm for mobile sensor networks," in *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) Held Jointly with 2009 28th Chinese Control Conference*, December 2009, pp. 7872–7877.

[20] S. Ivić, B. Crnković, and I. Mezić, "Ergodicity-Based Cooperative Multiagent Area Coverage via a Potential Field," *IEEE Transactions on Cybernetics*, vol. 47, no. 8, pp. 1983–1993, August 2017.

[21] S. Ivić, A. Andrejčuk, and S. Družeta, "Autonomous control for multi-agent non-uniform spraying," *Applied Soft Computing*, vol. 80, pp. 742–760, July 2019.

[22] S. Ivić, "Motion Control for Autonomous Heterogeneous Multiagent Area Search in Uncertain Conditions," *IEEE Transactions on Cybernetics*, pp. 1–13, 2020.

[23] S. Ivić, B. Crnković, H. Arbabi, S. Loire, P. Clary, and I. Mezić, "Search strategy in a complex and dynamic environment: The MH370 case," *Scientific Reports*, vol. 10, no. 1, p. 19640, December 2020.

[24] P. Klee and S. Moholy-Nagy, *Pedagogical Sketchbook*. London: Faber & Faber, 1953.

[25] R. Chamberlain, D. Berio, V. Mayer, K. Chana, F. F. Leymarie, and G. Orgs, "A dot that went for a walk: People prefer lines drawn with human-like kinematics," *British Journal of Psychology*, p. bjop.12527, August 2021.

[26] D. E. King, "Dlib-ml: A machine learning toolkit," *Journal of Machine Learning Research*, vol. 10, pp. 1755–1758, 2009.

[27] S. M. Pizer, E. P. Amburn, J. D. Austin, R. Cromartie, A. Geselowitz, T. Greer, B. ter Haar Romeny, J. B. Zimmerman, and K. Zuiderveld, "Adaptive histogram equalization and its variations," *Computer Vision, Graphics, and Image Processing*, vol. 39, no. 3, pp. 355–368, September 1987.

[28] S. Geman and D. Geman, "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-6, no. 6, pp. 721–741, November 1984.

[29] C. Andrieu and E. Moulines, "On the ergodicity properties of some adaptive MCMC algorithms," *The Annals of Applied Probability*, vol. 16, no. 3, August 2006.

[30] W. Li and E. Todorov, "Iterative Linear Quadratic Regulator Design for Nonlinear Biological Movement Systems," in *Proceedings of the First International Conference on Informatics in Control, Automation and Robotics*. SciTePress - Science and and Technology Publications, 2004, pp. 222–229.

[31] Unsplash. [Online]. Available: https://www.unsplash.com

[32] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019, pp. 4396–4405.