Check for updates

Original Article

Institution of MECHANICAL ENGINEERS

Journal of MECHANICAL ENGINEERING SCIENCE

Proc IMechE Part C:
J Mechanical Engineering Science
1–9
© IMechE 2024
Article reuse guidelines:
sagepub.com/journals-permissions
DOI: 10.1177/09544062241230171
journals.sagepub.com/home/pic
Sage

Using genetic algorithm for drawing path planning in a robotic arm pencil sketching system

Jin-Siang Shaw and Szu-Ying Lee

Abstract

The purpose of this study is to implement a robotic arm pencil sketching system through a novel drawing path planning method. Firstly, to enhance the craftsmanship and uniqueness of the artwork, the original image is transformed into an animated style using AnimeGANv2. After image processing, the image is divided into boundary and background layers for drawing path planning. In the realm of path planning, traditional methods often involve manual path design by artists, and only recently have automated path planning methods been proposed. However, these recent approaches still have some limitations. For instance, a pursuit of efficiency can lead to overly uniform strokes, making the artwork resemble a print rather than a genuine piece of art. Therefore, this study introduces a new path-planning approach based on the traveling salesman problem (TSP), utilising various algorithms for path-planning experiments. Different stroke effects generated by distinct algorithms are applied to appropriate areas. The aim is to let the generated artworks exhibit the technological appeal of algorithms while retaining the distinctive characteristics of pencil sketches. For robotic arm control, a force sensor is integrated onto the arm, and a proportional-integral-derivative (PID) controller is employed to monitor and precisely control the drawing force. Experimental results demonstrate that artworks created using genetic algorithms to generate drawing paths exhibit distinct colour-layering effects, emphasising the characteristics of pencil strokes. In comparison with previous methods, this approach effectively addresses the issue of overly uniform and rigid strokes in past techniques.

Keywords

Deep learning, force control, image processing, path planning, robot manipulator

Date received: 20 August 2023; accepted: 8 January 2024

Introduction

Since ancient times, many artists have regarded sketching as the cornerstone of their artistry. When a novice artist seeks to learn the craft of drawing and creation, the first skill to master is often sketching. By learning sketching, one can observe and capture the shapes, proportions, lines, and details of an object. This involves practising techniques such as outlining, controlling the pressure of strokes and shading. These fundamental abilities are essential for various other drawing techniques such as watercolours, crayons, and oil paintings. Modern image recognition and processing technologies have made this task relatively straightforward in terms of object observation. This study aimed to extend this capability to robotic arms, enabling them to sketch by outlining contours and accurately filling in areas, thereby creating sketches.

In this study, sketching pencils were used as the artistic media to create sketches of still lives, landscapes and portraits. Initially, images were transformed into desired artistic styles using image processing and animation-style conversion techniques. Several algorithms have been employed to plan drawing paths that capture the distinctive beauty of pencil strokes, resulting in unique aesthetics. Furthermore, a force sensor and proportional-integral-derivative (PID) controller were utilised to achieve force control in the robotic arm, ensuring consistent stroke depth throughout the drawing process. Finally, when solely using a 2B pencil for sketching, even with greater drawing pressure, areas with deeper colour tones may suffer from insufficient

Institute of Mechatronic Engineering, National Taipei University of Technology, Taipei City, Taiwan

Corresponding author:
Szu-Ying Lee, Institute of Mechatronic Engineering, National Taipei University of Technology, No. 1, Sec. 3, Zhongxiao E. Road, Da'an Dist., Taipei City 106344, Taiwan.
Email: szu0112@gmail.com

darkness, as validated by Hung's experiments. To address this issue, this experiment used an 8B pencil alongside a 2B pencil to sketch darker sections.

The application of robots to art creation dates back to the 1950s. The renowned Swiss sculptor Jean Tinguely created one of the earliest instances of robotic artwork in his Méta-matics series in 1959.[1] In 1973, Harold Cohen, a painter and engineer, created the painting robot ARRON.[2] ARRON sketched the outlines, whereas Cohen manually added colour and completed the artwork. Their collaboration spanned 40 years, making ARRON one of the longest-running and maintained artificial intelligence systems in history. With the advancement of artificial intelligence (AI) technology, robots have gained the ability to create art using deep learning techniques. They can learn and simulate various artistic styles to produce original pieces. Pindar Van Arman utilised deep learning neural networks, artificial intelligence and feedback loops to create the autonomous painting robot 'Artonomous', which reinterpreted Cézanne's Impressionist artworks.[3] This creation won the championship at the Third Robot Art Competition in 2018.

Designing and planning an AI-based drawing robot system is not a simple task. Numerous external factors must be considered when a robot engages in an actual drawing, such as paper texture, brush characteristics and coverage area. To enhance the accuracy, it is crucial to equip a robot with sensory capabilities, including vision and touch. In terms of vision, the portrait-drawing robot 'Paul' was showcased in 2011 by Tresset at London's Tenderpixel gallery.[4] Paul lacked prior knowledge of facial features but employed a camera as its 'eyes' for visual perception. It continuously provided visual feedback and analysis during the drawing process to determine where to render the shading. Iterations through this process improved its work, ultimately producing lifelike and artistically infused portraits. In 2018, Song et al. employed the KUKA Robotics collaborative robot, LBR IIWA, which featured impedance control capabilities.[5] Through impedance control, the seven-axis robotic arm executed drawing tasks on various curved surfaces.

Although the goal is drawing, there is a plethora of methods, each significantly influencing the final style of the artwork. Different creative tools and techniques significantly impact end products. Consequently, drawing path planning has become a focus of recent research. In 2004, Bosch and Herman introduced the traveling salesman problem (TSP) for drawing path planning.[6] This approach planned continuous lines to replicate artworks, and successfully recreated many famous paintings. In 2009, using a TSP version with 100,000 cities, Bosch reproduced the 'Mona Lisa'. In 2022, Wu et al. employed TSP with nearest-neighbour algorithms to plan pencil sketching paths efficiently.[7] The robotic arm successfully created recognisable sketches. Scalera et al. developed a robotic painting system that used the watercolour technique and a sponge as the painting media. A contour-filling algorithm was developed to define the sponge positions and orientations in order to colour the image contour.[8] Nasrat et al.[9] utilised CycleGAN to extract crucial facial features and implemented a path optimisation algorithm. This implement enabled the robotic arm to efficiently produce high-quality portrait drawings in a short amount of time.

In previous approaches, the emphasis was mainly on improving efficiency and precision, but artworks produced with high efficiency often risk losing their artistic characteristics. Taking pencil sketches as an example, the distinctive feature lies in the repetitive stacking and blending of pencil strokes. If a simplistic filling path is employed solely for the purpose of efficiency, the essence of pencil sketching may be compromised. This study is dedicated to developing a novel path-planning approach that integrates algorithmic properties with artistic creation. The goal is to enable the robotic arm to create pencil sketch artworks that embody both a technological and artistic aesthetic.

In summary, the main goal of this paper is as follows: (1) The implementation of a robotic pencil sketching system; (2) Achieving a broad range of colour and depth variations by combining pencils of two different hardness levels with a force sensor; (3) Path planning for contour filling using intrinsic algorithms and a comparative analysis of their effectiveness.

System description

System architecture

The robotic-arm automated drawing system developed in this study was primarily divided into five stages: style transformation, image processing, drawing path planning, robotic-arm control, and force control, as shown in Figure 1. First, to enhance the richness and vibrancy of the final artwork, the image was subjected to AnimeGANv2[10] for style transformation. This step made the lines in the image more distinct, aiding subsequent edge detection and emphasising the facial features that bring them to life. Subsequently, image-processing techniques were employed for edge detection and image segmentation, which divided the image into boundary and background layers to facilitate subsequent drawing path planning.

For drawing path planning, the list of image pixel positions was modelled as a city list in the travelling salesman problem (TSP).[11] The optimal path was determined using algorithmic calculations. Different algorithms yielded diverse drawing paths, which resulted in various stroke effects during the drawing process. In addition, a force sensor was installed on the flange surface of the robotic arm to monitor the

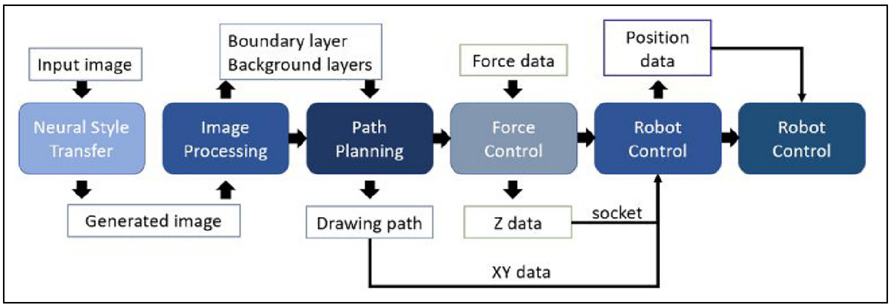

Figure 1. System architecture.

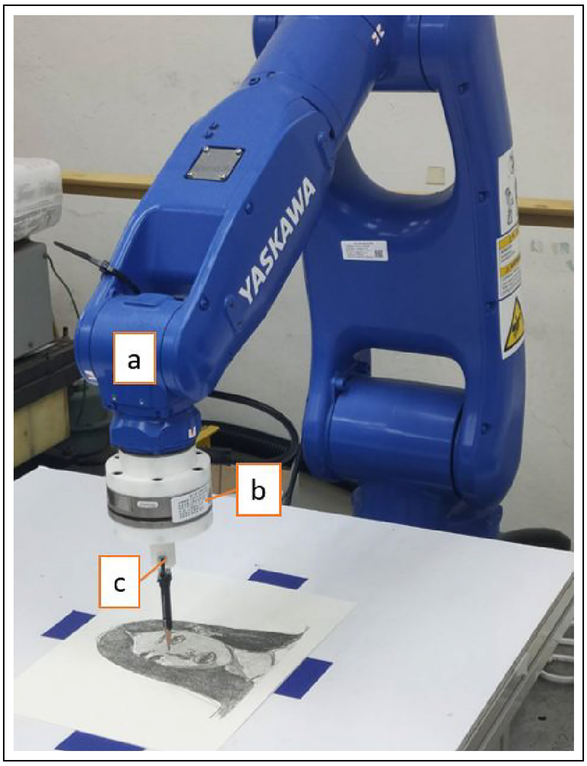

Figure 2. Robotic painting system: (a) robotic arm, (b) force sensor and (c) painting module.

real-time drawing force. During the drawing process of the robotic arm, PID control was used for fine adjustments in the Z-axis direction to regulate the drawing force and compensate for pencil wear. This achieved optimal control of the robotic arm.

Hardware architecture

The robotic arm used for this experiment was a GP7 six-axis robotic arm manufactured by a Japanese company, Yaskawa, as shown in Figure 2. This series of robotic arms is characterised by rapid movement and high precision, making them suitable for applications such as high-speed handling, part assembly and grinding processes. Thus, this model has been widely used in various industries. The robotic arm was controlled using a compact and easily installed YRC1000 microcontroller. An F/T sensor was installed on the flange surface of the robotic arm to monitor the drawing force.

Software architecture

As shown in Figure 3 of the software architecture diagram, this experiment involved five distinct phases in the overall program-development process and employed four different programming language platforms. Programming tasks for deep learning, image processing, and path planning were performed using Python 3.0. For force control, the TwinCAT3 automation software was used to facilitate real-time force data acquisition. Subsequently, C# programming software was selected, and the TwinCAT.ADS protocol was integrated within the development environment for real-time pressure data retrieval, ensuring seamless communication with the main control program. Finally, the FESIFS library was used to control the robotic arm. The control program was writtened in the VB.net development environment and the G-code was scripted on the teaching pendant for basic positional control. Finally, flag-based methodologies were integrated into a dual-sided program to monitor the real-time status of the robotic arm.

Methodology

This section introduces the research methodology in four parts: animation-style transfer, image processing, path planning and robotic-arm control.

Animation style transfer

The objective of this experiment was to enhance the resulting artwork with a richer artistic flair rather than merely replicating an image. To achieve this, we employed AnimeGANv2 to apply a comic-transformation process to the original images. This model enables the infusion of animated stylistic textures into original images, imbuing them with more vibrant lines and colours. Figure 4 illustrates the image-to-image animation style transfer results of AnimeGANv2.

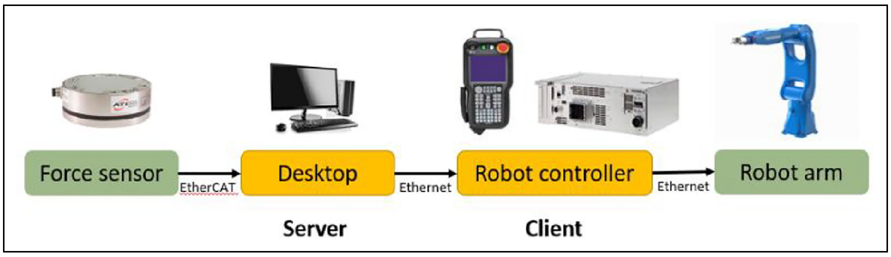

Figure 3. System communication.

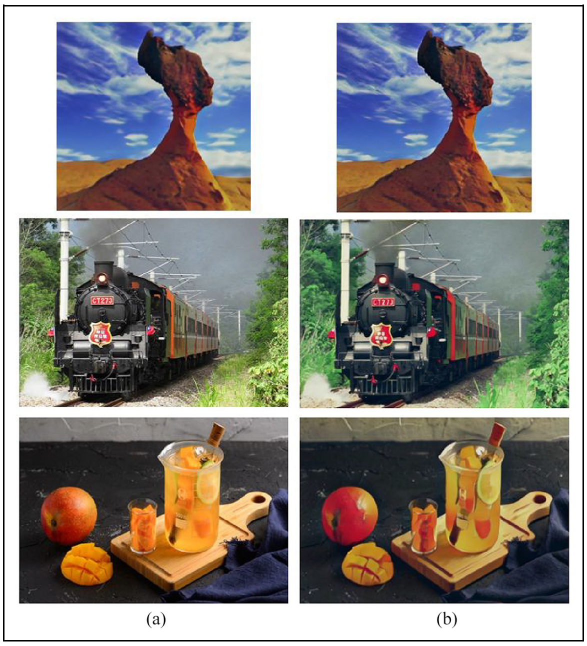

Figure 4. Image animation style transformation: (a) original image and (b) generated image.

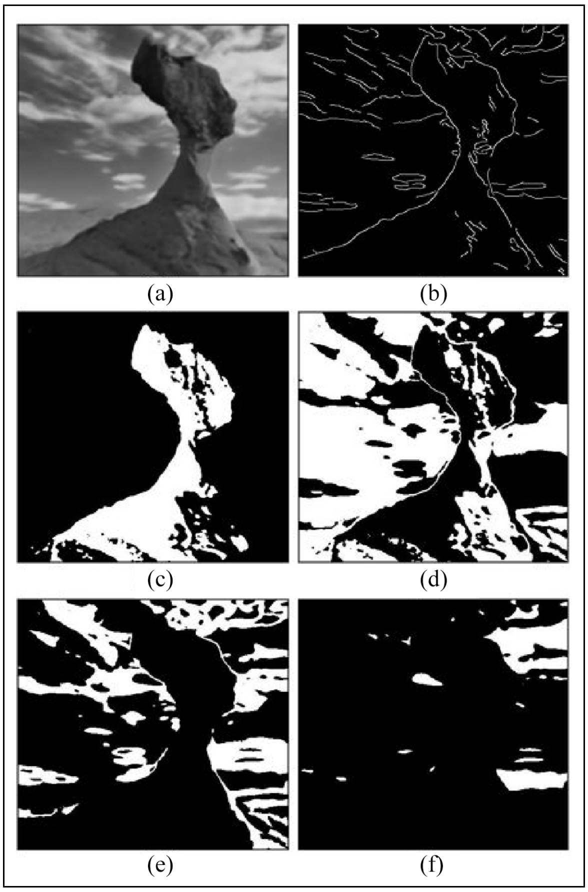

Figure 5. Image processing: (a) gray-scale conversion and filtering applied to the generated image, (b) detected boundary layer obtained through edge detection and (c)–(f) four background layers categorised using image quantisation method.

Image processing

After comic transformation, the generated images underwent several image processing procedures before proceeding to drawing path planning. The image-processing steps used in this study consisted of image preprocessing, edge detection and image segmentation.

The image preprocessing step involved grayscale conversion and image smoothing. After the coloured images were converted into grayscale, a median filter[12] was applied for image smoothing, which eliminated high-frequency noise from the images, resulting in a smoother appearance that facilitated subsequent edge detection.

The Canny[13] edge detection algorithm was used as the edge detection method. The aim was to isolate the boundary contours within the image, which served as the bases for subsequent contour rendering. Image segmentation involves dividing an image into regions with similar characteristics. In this study, image segmentation was performed based on pixel intensity. The image segmentation technique[14] divided the pixels into several layers, facilitating the subsequent application of varying colour depths to the background layer. Figure 5 shows the boundary and background layers obtained using a series of image processing procedures.

Drawing path planning

Previous drawing path planning methods have prioritised efficiency. Although improving efficiency is crucial, artistic expressions and stroke depictions take precedence in artistic creation. Therefore, this study

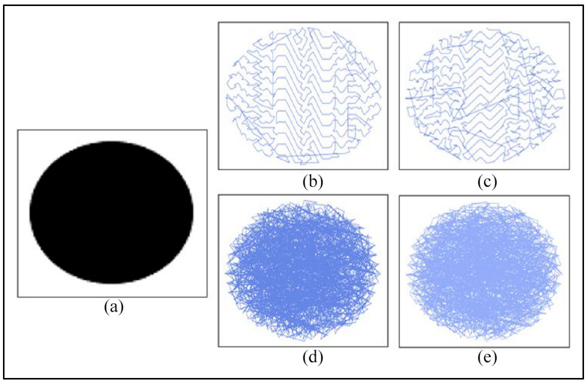

Figure 6. Comparison of path planning with different algorithms: (a) circular region intended for colouring, (b) drawing path planned by the nearest neighbour algorithm, (c) drawing path generated by the ant colony optimisation algorithm, (d) drawing path planned by the particle swarm optimisation algorithm and (e) drawing path generated by the genetic algorithm.

employed diverse drawing path planning methods to generate distinct stroke effects.

Travelling salesman problem (TSP). The travelling salesman problem (TSP) is a well-known combinatorial optimisation problem that seeks to find the shortest possible route for a travelling salesperson to visit all cities in a given list and return to the starting city, in which each city is visited only once. Owing to the similarity of the goals of a travelling salesperson and the colouring actions in drawing, there has been a growing trend in recent years to apply TSP-based approaches to drawing path planning. This study was based on TSP as the foundation for path planning, employing different algorithms, including the nearest neighbour algorithm,[15] ant colony optimisation (ACO),[16] particle swarm optimisation (PSO)[17] and genetic algorithms (GA)[18] for drawing path-planning experiments.

For the boundary layer, we prefer continuous path planning results to enable the system to quickly complete the outline drawing task. In contrast, for the background layer, which involves areas to be coloured, we aim to showcase the stacking effect of pencil colours. Therefore, we prefer more complex drawing paths for these layers. As shown in Figure 6, we conducted a drawing path planning experiment for a circular area using the four aforementioned algorithms. The results clearly demonstrate that the drawing path sequence calculated by the nearest neighbour algorithm is the most stable and efficient, making it suitable for path planning in the boundary layer. The paths generated by PSO and GA exhibited a high degree of randomness, creating a chaotic line structure that enhanced the stacking effect of the pencil colours. Because GA possesses better global exploration capabilities than PSO, allowing it to identify shorter paths within the same number of iterations, we adopted the GA as the path-planning method for the background layer. The following sections briefly introduce these two algorithms.

Nearest neighbour algorithm. The nearest neighbour algorithm is a simple and intuitive heuristic algorithm. The core idea of this algorithm is to start from an initial point and iteratively select the closest unvisited point to the current position as the next point to visit until all points are visited, forming a loop. The basic steps of the nearest neighbour algorithm are as follows.

(a) Create a list of all desired locations to visit and record the order of the traversed paths.
(b) Choose a point as the initial starting point.
(c) Start from the departure point and select the closest unvisited point from the list of paths as the next point.
(d) Add that point to the order list and mark it as visited.
(e) Update the current position to the selected point.
(f) Repeat steps 3–5 until all points are visited.

The advantages of the nearest neighbour algorithm include its simplicity and ease of implementation. It can obtain a feasible solution within a short period of time. This algorithm has often been used for path planning owing to its efficient drawing capability and suitability for outlining boundary contours.

Genetic algorithm. The genetic algorithm is inspired by the principles of natural evolution. Based on the concept of 'survival of the fittest', certain advantages or disadvantages within biological individuals lead to variations in survival capabilities, thereby affecting reproductive abilities. This process gradually preserves or eliminates certain traits. The genetic algorithm simulates the mechanism of natural selection by representing solutions as chromosomes, imitating biological phenomena, such as mating and mutation. Chromosomes are retained or discarded based on their fitness, thereby achieving optimisation. The computational process of the genetic algorithm is as follows:

(a) Create a list of cities and generate a random initial generation of individuals, where each individual represents a potential solution.
(b) Define a fitness function to evaluate the quality of each solution.
(c) Calculate the fitness value for each individual.
(d) Select a group of superior individuals as parents based on their fitness. Generate the next generation by applying crossover or mutation and update the population accordingly.

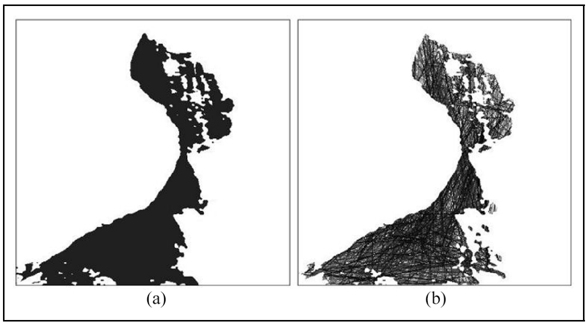

Figure 7. Drawing path planning: (a) background layer requiring colouring and (b) simulated drawing result obtained after path planning using the genetic algorithm, considering the pencil colour stacking phenomenon.

(e) Repeat steps 3-4 for multiple iterations until a termination condition is satisfied.

In this study, we simplified the pixel positions of the areas to be coloured and generated a list of cities. The randomly generated drawing paths served as chromosomes, and the fitness value of each chromosome was evaluated. In the context of the Traveling Salesman Problem, the total travelled distance is commonly used as the fitness criterion, where shorter distances correspond to higher fitness values. However, in our research, we needed to consider the even distribution of strokes, so we utilised the ratio of the total drawing area (A_d) to the total traveled distance (D_t) as the fitness function. This approach effectively prevents excessive repetition of strokes. The fitness function is presented by equation (1). Through continuous mating and mutation of chromosomes with high fitness, the experiment aimed to find the optimal solution. The experiment was conducted for 50 iterations, with a mating rate of 30% and a mutation rate of 10%.

$$Fitness\ value = A_d/D_t \tag{1}$$

The genetic algorithm exhibits strong global search capabilities. Compared to particle swarm optimisation, it is less prone to becoming trapped in local optima, making it suitable for handling problems of higher complexity. Figure 7 shows the drawing path planning results for one of the background layers.

Force control

The installation of force sensors on a robotic arm provides information regarding the interaction between the arm and its environment. This includes details regarding the contact and friction forces between objects. In sketching tasks, the robotic arm emulates the human drawing behaviour using a pencil or brush to draw on paper. This task requires high precision and subtle control. By incorporating force sensors, the robotic arm can perceive changes in pressure between the pencil and paper and subsequently adjust its motion trajectory based on this information, thus achieving accurate sketches.

The force control method used in this experiment was similar to that employed by Hung.[7] A force sensor was installed on the end effector of the robot to read the force values. Coupled with a PID controller, this setup ensured that the drawing force exerted by the robotic arm remained stable within an error margin of 0.5 N. This stability enabled the robotic arm to complete the sketching tasks while maintaining equilibrium. A discrete PID controller is represented by equation (2):

$$Z_n = Z_{n-1} + K_p F_e + K_i \frac{I + F_e dt}{2} + K_d \frac{F_e - F_{e_{n-1}}}{2dt} \tag{2}$$

To prevent excessive displacement of the robotic arm, which could lead to pencil breakage, the error values of the integral and derivative components were divided by two. This adjustment is described in equation (2), where Z_n represents the new vertical position of the robotic arm, Z_{n-1} is the current vertical position, F_e is the current error obtained by subtracting the current force value from the target force value, and $F_{e_{n-1}}$ denotes the previous error signal. Variable I represents the accumulated integral error value, which signifies the waiting time, and was set here as 0.5 s. The parameters of the PID controller were determined through multiple experiments, and the following values were obtained: $K_p = 6.3$, $K_i = 0.23$ and $K_d = 0.18$.

Figure 8 illustrates the force values recorded by the force sensor when the robotic arm performs drawing actions. It is evident that the drawing force, controlled through PID regulation, effectively stays within our specified force range. Additionally, when the detected force values exceed the expected range, the robotic arm halts the drawing motion along the XY axes until the Z-axis direction is completely corrected. This ensures the safety of the robotic arm during drawing, preventing pencil breakage or collisions.

Robot arm control

In addition to drawing path planning, aspects such as pen lifting planning and coordinate system configuration require comprehensive planning and constraints to ensure that the robotic arm operates without any unforeseen incidents. The pen lifting method and coordinate settings used in this experiment were consistent with those established by Wu et al.[7] For the liftoff path, two intermediary points were set to ensure that the robotic arm moved vertically during liftoff and touchdown, thereby preventing unnecessary marks on the paper. For the coordinate system, the

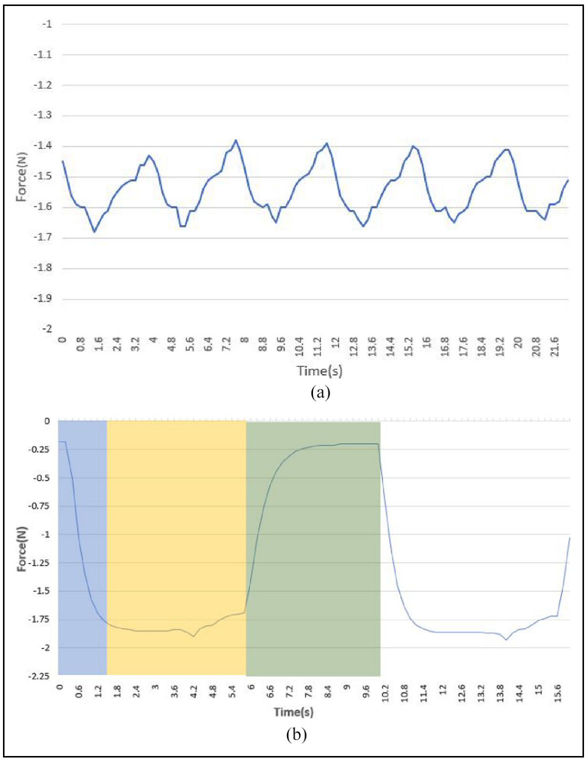

Figure 8. (a) The recorded force values when the robotic arm performs continuous drawing actions with the force of 1.5 N, and (b) the force values recorded as the robotic arm sequentially goes through the downstroke (blue segment), drawing (yellow segment) with the force of 2 N, and lifting the pencil (green segment).

tool centre point (TCP) was employed as a reference for the robotic arm's movement instructions.

Results and discussion

Path planning results

For path planning, two path-planning algorithms based on the travelling salesman problem were employed in this study. The nearest neighbour algorithm was used for boundary-layer path planning, which provided continuity and efficiency. For the background layer path planning, a genetic algorithm was employed, which is known for its high degree of randomness that effectively captures distinctive pencil strokes and enhances the artistic quality of the work. Figure 9 shows the path planning results for landscapes, still life, and portrait drawings.

Final drawing results

To facilitate the drawing process of the robotic arm, the generated images were divided into multiple layers, each with a unique desired pencil-contact force. The boundary layer was drawn using an 8B pencil and a force of 3 N. Background Layer 1 was drawn using an 8B pencil and a force of 4 N.

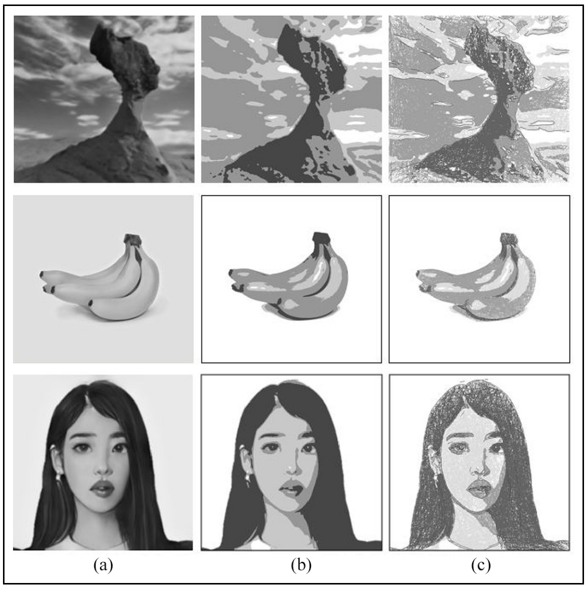

Figure 9. (a) Generated image, (b) image layering result and (c) drawing path planning result.

Background Layer 2 was drawn using a 2B pencil and a force of 5 N, whereas background Layer 3 was drawn using a 2B pencil and a force of 2.5 N. Background Layer 4, owing to its close-to-white colour, was not drawn. The use of a PID controller for force feedback control effectively maintained the drawing force within a specific range, resulting in distinct colour variations between the layers. The final drawing results are shown in Figure 10.

Conclusions

The primary features of pencil sketching include: (1) precise line detailing, (2) utilisation of grayscale, (3) the repetitive stacking and blending of pencil strokes on paper and (4) a sense of craftsmanship. In this study, a high-precision industrial robotic arm, Yaskawa GP7, is employed to outline accurate contours, while changes in pencil grayscale are achieved through the integration of a force sensor and PID control. Subsequently, a distinctive drawing path planning method is utilised to showcase the repetitive stacking effect of pencil strokes on paper. Finally, the incorporation of image cartoonisation techniques and layered drawing enhanced the overall artistic craftsmanship of the artwork. The contributions of this study are as follows.

1. Robotic arm sketching artistry: Although the use of robotic arms for artistic creation is common, most artworks involve automatic watercolour, oil painting, and pen drawings. Utilising a robotic arm with a force sensor for pencil sketching is a novel approach that offers controlled shading and expands creative possibilities.

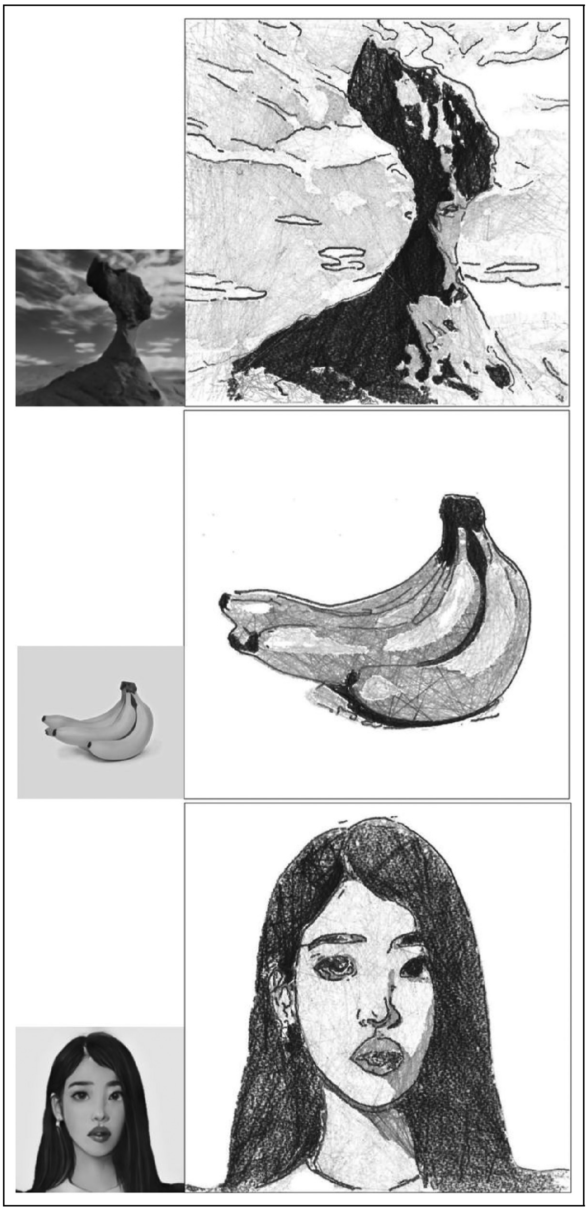

Figure 10. Final drawing results.

2. Unique stroke effects: Although robotic arm-assisted artistic creation has gained attention, many artworks involve artists manually designing and planning drawing paths for robot imitation. While some recent methods have introduced automated path planning, they often prioritise efficiency and precision, resulting in artworks that resemble prints rather than unique pieces of art. This study implemented diverse path-planning methods to ensure that the artworks created by the robot possess a combination of algorithmic precision and the technological touch of machinery while retaining distinctive pencil stroke characteristics, ultimately generating art pieces with a unique style and features. Figure 11 shows the comparison of pencil sketch results.

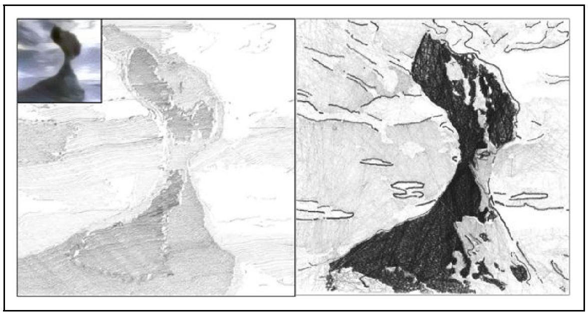

Figure 11. Comparison of pencil sketch results.

   The left image shows the drawing results of the previous method,[7] while the right side displays the results obtained using the approach in this study. It is evident that the path planning method employed in this experiment effectively enhances the stacking effect of pencil strokes.
3. Force control: In pencil sketching using a robotic arm, issues such as pencil tip wear and the exerted force must be addressed. Incorrect control can lead to inadequate contact or pencil breakage. In this study, a force sensor was employed to monitor the real-time drawing force, and a PID controller was used to maintain a stable force range within the same layer. This approach achieved pencil-tip compensation and uniform force application.

However, some aspects require further investigation. If optimised and improved step-by-step, these aspects could significantly enhance system performance. This insight is explained below.

1. PID automatic tuning: This study employed force control with PID control to maintain consistent force exertion. Combining the PID controller with methods such as fuzzy control, expert systems or neural networks enable automatic tuning, thereby eliminating manual calculations and adjustments.
2. Innovative path-planning algorithms: To enhance stroke aesthetics, this study used genetic algorithms for colour application path planning. However, the high randomness of the algorithm results in prolonged drawing times. Identifying more efficient and simultaneous artistic path planning methods would markedly improve the operational efficiency of the system.
3. Automated pencil-changing system: This study used two pencil types for various shading, requiring manual pencil changes. Designing an automated pencil-changing mechanism can significantly enhance the efficiency and provide greater flexibility in choosing artistic media.

4. Enhancement of the robotic arm drawing method: The robotic arm in this study maintained a vertical pencil orientation. Teaching the robotic arm with different drawing angles resulted in more diverse stroke effects.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This study was supported by the Ministry of Science and Technology, Taiwan (MOST 110-2221-E-027-048).

ORCID iD

Jin-Siang Shaw https://orcid.org/0000-0002-8329-9452

References

1. Cohen H. How to draw three people in a botanical garden. In: *AAAI*, 1988, pp.846–855.
2. The prophecies of AARON, https://outland.art/harold-cohen-aaron/ (2022).
3. Simpson K and Arman PV. Teaching a robot the subtleties of fine art, https://artonomo.us/products/artonomous-portrait (2021).
4. Tresset P and Leymarie FF. Portrait drawing by Paul the robot. *Comput Graphics* 2013; 37(5): 348–363.
5. Song D, Lee T and Kim YJ. Artistic pen drawing on an arbitrary surface using an impedance-controlled robot. In: *2018 IEEE international conference on robotics and automation (ICRA)*, Brisbane, QLD, Australia, 2018, pp.4085–4090.
6. Bosch R and Herman A. Continuous line drawings via the traveling salesman problem. *Oper Res Lett* 2004; 32(4): 302–303.
7. Wu PL, Hung YC and Shaw JS. Artistic robotic pencil sketching using closed-loop force control. *Proc IMechE, Part C: J Mechanical Engineering Science* 2022; 236(17): 9753–9762.
8. Scalera L, Canever G, Seriani S, et al. Robotic sponge and watercolor painting based on image-processing and contour-filling algorithms. *Actuators* 2022; 11(2): 62.
9. Nasrat S, Kang T, Park J, et al. Artistic robotic arm: drawing portraits on physical canvas under 80 seconds. *Sensors* 2023; 23(12): 5589.
10. Chen X. AnimeGANv2, https://tachibanayoshino.github.io/AnimeGANv2/ (2020).
11. Jünger M, Reinelt G and Rinaldi G. Chapter 4 – the traveling salesman problem. *Handb Oper Res Manage Sci* 1995; 7: 225–330.
12. Huang T, Yang G and Tang G. A fast two-dimensional median filtering algorithm. *IEEE Trans Acoust Speech Signal Process* 1979; 27(1): 13–18.
13. Canny J. A computational approach to edge detection. *IEEE Trans Pattern Anal Mach Intell* 1986; 8(6): 679–698.
14. Pal NR and Pal SK. A review on image segmentation techniques. *Pattern Recognit* 1993; 26(9): 1277–1294.
15. Larose DT and Larose CD. K-nearest neighbor algorithm. In: *Discovering knowledge in data: an introduction to data mining*, Hoboken, NJ: Wiley, 2014, pp.149–164.
16. Dorigo M, Birattari M and Stutzle T. Ant colony optimization. *IEEE Comput Intell Mag* 2006; 1(4): 28–39.
17. Kennedy J and Eberhart R. Particle swarm optimization. In: *Proceedings of ICNN'95 international conference on neural networks*, Perth, WA, Australia, 1995, vol. 4, pp.1942–1948.
18. Holland JH. Genetic algorithms. *Sci Am* 1992; 267(1): 66–73.