A Drawing Robot Base on the Gen3-lite Robot Arm

Haoran Wang, Yukun Li

Abstract— **This project is dedicated to developing an advanced six-axis robotic arm drawing system. The system uses advanced depth cameras for precise drawing on varied-tilt work surfaces. Key steps in system development start with designing a specialized conversion component for the robotic arm's end. The component's main purpose is to secure the drawing pen, also allowing easy pen type replacement by users. Next, depth cameras gather spatial dimensions of the work environment, focusing on the drawing surface's angles and positions. The system ensures drawing precision with depth camera details, simplifying images upon input. Then, the robotic arm's motion path is planned based on the simplified graphics. After calculation and programming, the robotic arm precisely moves and draws following the preset path. Each movement and drawing action is precisely calculated for correct image rendering.**

The technical innovation and application potential of this project are very wide-ranging, not only providing new tools and methods in the field of artistic creation, fostering the generation of innovative art but also offering practical auxiliary tools for education. At the same time, this technology also demonstrates its potential application value in industrial design and production. The innovation of this project lies in its combination of depth camera and high-precision robotic arm technology, accurately capturing and processing the information of work surfaces in the actual environment to achieve precise drawing of complex images. The development of this breakthrough technology signifies the successful integration of modern sensing technology and automation technology in artistic and industrial applications.

I. Introduction

Entering the new era of automation and robotics, industries across the board are seeking more flexible, intelligent and efficient solutions to improve operational precision and work efficiency. Six-axis high-precision robotic arms, with their high flexibility and precision, are widely used in various fields. Compared to rigid three-axis or four-axis devices, six-axis robotic arms can perform more complex movements and finer operations due to their freedom of movement, and they can reach any position and angle in space almost without restriction. In the following, some application examples of six-axis robotic arms in the industrial and medical fields are given:

Assembly: Widely used in electronic product manufacturing (such as the assembly of mobile phones and laptops), six-axis robotic arms can precisely position and assemble parts, such as Cartesian coordinate robots [1], shown in figure.1(a). Welding: Six-axis robotic arms are used in car manufacturing to weld body components. They can perform precise welding on complex paths, improving the quality and efficiency of welding, such as welding robots [2],shown in figure.1(b).

Surgical operations: In minimally invasive surgeries, six-axis robotic arms can help doctors perform operations with higher precision, such as the remote MIS surgical system robot [3], shown in figure.1(c), and the compact tabletop MIS robot [4], shown in figure.1(d). Rehabilitation therapy: Used in rehabilitation training equipment to assist patients in accurate physical therapy, such as rehabilitation robot A [5], shown in figure.1(e),and rehabilitation robot B [6], shown in figure.1(f)

Picking and packaging: Six-axis robotic arms are used in warehousing and logistics for automated picking and packaging processes, improving operational efficiency and accuracy, such as logistics express robots [7], shown in figure.1(g). Processing and packaging: In food processing and packaging, such as automated tray loading and food cutting, operations are guaranteed to be efficient and compliant with hygiene standards, such as food processing robots [8], shown in figure.1(h).

However, the design of robotic arms is not only for mimicking human arms to perform some repetitive work. Its anthropomorphic design characteristics suggest that it should also have certain potential in fields like art and display, introducing unprecedented possibilities for art creation and exhibitions. The following are some unique application instances:

Robotic Arm Drawing: Creatively using a six-axis robotic arm for painting, artists can combine advanced programming techniques to design complex painting paths, creating art that is hard to achieve with traditional hand drawing. And there is also some easy solution, as the 3DOF LEGO-NXT [9] shows, quickly and directly draw a rectangular, lack of precise and control. Sculpture Making: By precisely controlling the robotic arm, artists can carve on various materials to produce intricate and detailed sculptures. The robotic arm can accurately remove excess materials according to the designed 3D model, showcasing amazing details and precision.

Interactive Art Installations: A six-axis robotic arm can be used to create interactive art installations, where visitors' movements or sounds can trigger specific responses from the robotic arm, increasing the interactivity and fun of the exhibition. Education and Display: At museums or exhibitions, the robotic arm can be used to demonstrate scientific principles or art techniques, attracting and educating viewers.

The above are the advantages and current application scenarios of robotic arms. We find the accuracy and the feature of exchangeable end effectors of the robotic arm very interesting, thus we hope to combine these advantages to develop a new type of drawing system, aiming to feel the control theory and application ideas of the robotic arm during the development process. This system integrates the advanced control strategy of the depth camera and the robotic

arm to realize the function of drawing images on slanted work surfaces.

Regarding the idea of robot drawing, there are also some schemes that have been proposed and implemented, such as the simple "3DOF drawing robot using LEGO-NXT"

II. TASK1: Image Processing

A. Edge Detection

Edge extraction was implemented using the function libraries in OpenCV, which use the latest improved CANNY method [10]. The first step of the process is to apply Gaussian Blur, which smooths the image to eliminate noise points, making subsequent edge detection more accurate. Next, the image is converted to grayscale format because edge detection usually performs best on a single color channel. Then, the Sobel operator is used to calculate the gradient magnitude and direction of each pixel in the image, providing a basis for determining image edges. The fourth step is to implement non-maximum suppression, which keeps the pixels with the maximum gradient value (i.e., the most obvious edge pixels) and suppresses other non-edge responses, reducing erroneous information due to edge detection. The fifth step introduces a dual-threshold approach, distinguishing the strength of edges by setting high and low thresholds, determining which edges are real and which are potential. Finally, weak edges are gradually eliminated, identifying the real edges in the image and completing all steps of edge detection. This entire process is a key method in edge extraction for many computer vision applications, such as object recognition and image segmentation.

B. Point Set Exporting

Still in progress

III. TASK2: Plane Calibration by QR Code

A. Place a QR Code and

A function for capturing a single frame image from the camera and displaying it is implemented using the OpenCV library. First, it creates a VideoCapture object that points to the default camera device and checks whether the camera is successfully opened and the frame image is successfully read. If an error occurs, the corresponding exception will be thrown. Then, the code successfully obtains and captures a frame image and displays it in a window via the cv2.imshow() function, it also simultaneously saves the image captured by the camera as a "photo.jpg" file. Finally, the code releases the camera resources and closes all windows created through OpenCV to ensure the normal use of system resources.

B. Cropped Image Operation and Processing

QR code recognition and depth information reading are carried out using the OpenCV library, which then uses this information to locate QR codes in 3D space [11]. First, the RGB image and the corresponding 16-bit depth image are loaded using the cv2.imread function. Subsequently, a QR code detector is initialized to identify QR codes in the RGB image, and the image coordinates of the QR code vertices are obtained through the detection results. If the QR code is successfully detected, the code will traverse each vertex coordinate and read the depth value at the corresponding position in the depth image, thus obtaining the three-dimensional coordinate information of each vertex. With the 3D coordinates of the four corners of the QR code, these 3D coordinates points are defined, and the cv2.solvePnP function is utilized to solve for the pose of the QR code relative to the camera, including the rotation vector (rvec) and translation vector (tvec), which is very helpful for understanding the specific location and orientation of the QR code in the real world.

IV. TASK3: Trajectory Planning of End Effector

A. Algorithm of the Shortest Path Planning

The Dijkstra algorithm is a classic shortest path searching algorithm used to find the shortest path between two points in a weighted graph. The core idea of the algorithm is to expand the known shortest path set breadth-first, until the shortest path to the target point is found.

Initialization: Set the distance of the starting point to 0, and the distance to all other points as infinity. Mark all points as unvisited. Choose the closest point: Select a point from the unvisited points, which is the closest to the starting point (called the current point). Update neighbor distances: Examine all unvisited neighbors of the current point, update the distance from the starting point to these neighbors. If the path to a neighbor via the current point is shorter than the known path, then update the neighbor's shortest path. Mark visited: Mark the current point as visited to ensure it is not selected again. Repeat: Repeat steps 2 to 4 until all points have been visited, or the shortest path to the target point is found. The Dijkstra algorithm is applicable to both directed and undirected graphs, and the edge weights in the graph must be non-negative.

The A* (A-Star) algorithm is also a shortest-path finding algorithm, widely used in the field of graph search. Compared to the Dijkstra algorithm, the A* algorithm introduces a heuristic function to estimate the cost from any point to the target point, enabling more efficient search, their functional efficiency are compared in Andrew's article [12].

Initialization: Add the starting point to the open list. Loop search: As long as the open list is not empty, select a point estimated to have the shortest path as the current point. End condition: If the current point is the target point, then the path has been found, and the algorithm ends. Neighbor check: For each neighbor of the current point, if the neighbor is in the closed list (has been visited), ignore it. Compared to the Dijkstra algorithm, the A* algorithm significantly narrows the search range and makes path searching more efficient through the heuristic function. The choice of the heuristic function has a big impact on the efficiency of the algorithm. Common heuristic functions include Manhattan Distance, Euclidean Distance, etc. The operations are guided by a Tutorial [13].

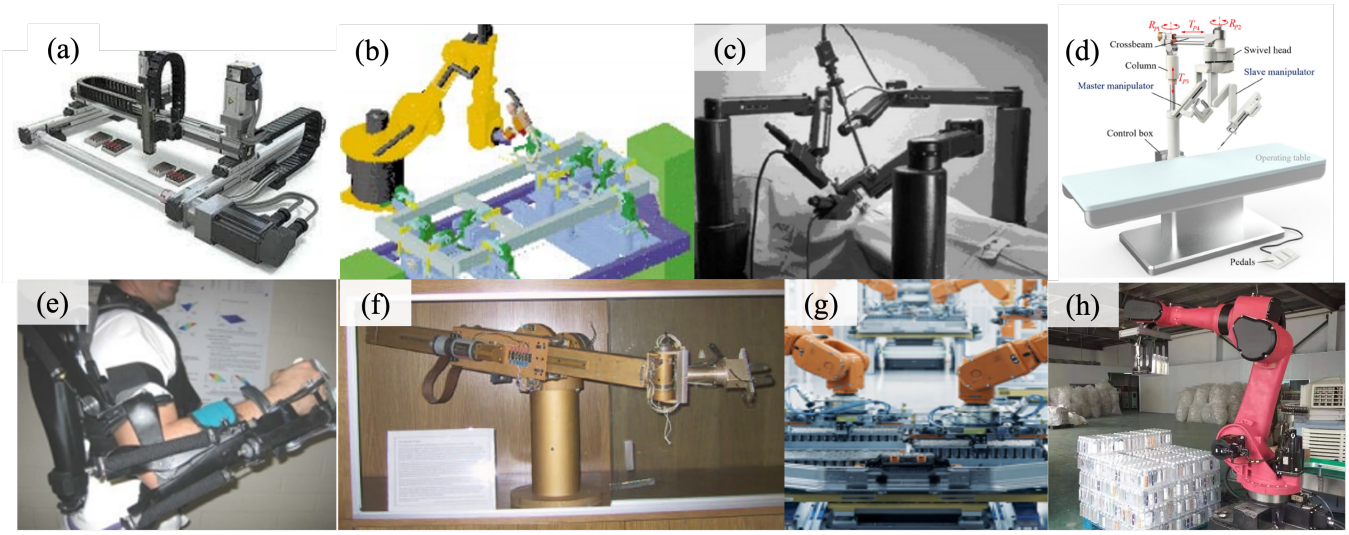

Fig. 1. A comprehensive overview of six-axis robotic arms' application across diverse sectors, highlighting their transformative roles. (a) Assembly in electronics manufacturing showcasing precision part positioning, (b) Welding in automotive construction for high-quality joins, (c-d) Advanced surgical operations emphasizing minimally invasive techniques, (e-f) Rehabilitation therapy with robots aiding in precise physical recovery exercises, (g) Enhanced efficiency in logistics through automated picking and packaging, and (h) Hygienic and efficient food processing and packaging automation. These images collectively underscore the versatility and pivotal impact of robotic technology in manufacturing, healthcare, logistics, and food industries.

B. Rotation Matrix

The code first defines a function named $otate_trajectory$, which takes a set of points and a rotation matrix as arguments, and rotates each point through matrix operations to match the orientation of an inclined plane. Next, the $adjust_z$ function is responsible for adjusting the height of each point along the Z axis to ensure that the path points maintain contact with the working plane. It receives a set of points and a Z-axis adjustment value, then modifies the height of each point according to this value. Then there is the $calculate_rotation_matrix$ function, which calculates the rotation matrix based on a given inclination angle and the axis of inclination (i.e., the normal vector). Here, it is necessary to convert the angle from degrees to radians and normalize the axis vector to a unit vector, then use the axis-angle representation (Rodrigues' rotation formula) to generate the rotation matrix. Once we have the original three-dimensional set of trajectory points, we first determine the plane's inclination angle and normal vector to get the corresponding rotation matrix for the trajectory on the new plane. Subsequently, this matrix is applied to the original set of trajectory points to get the rotated points, and finally, these points are appropriately adjusted along the Z axis as needed to ensure the correct contact of the trajectory with the working surface. After completing these mathematical operations and geometric transformations, the resulting $adjusted_points$ are a new set of trajectory points, suitable for a new operational environment that takes into account the inclination of the working surface, such as for the path planning of a robotic arm.

C. Orientation of End Effector

When calculating the plane position, the normal direction corresponding to the QR code was calculated synchronously. According to the shape of the drawing module, the robotic arm gripper should be controlled to be parallel to the plane where the QR code is located. Therefore, all you need to do is to control the orientation of the end effector to coincide with the direction of the normal vector.

V. TASK4: Motion Planning of the Robot Arm

A. Combine the End Effector

Since we cannot guarantee that the actuator will be in the exact same position in different applications, we hope to use a depth camera to identify and calculate the center position of the actuator. And according to the physical parameters of the actuator and its relative position, calculate the gripping posture and position of the robotic arm. And plan an action for gripping the actuator, moving from the initial position to the target gripping point before each drawing begins, and tighten the gripper so that the actuator is integrated onto the end effector.

B. Draw the Image

Using the previously obtained ordered coordinate set of the optimal trajectory, control the robotic arm to draw points one by one.

References

[1] A. Banerjee, "Robotics in indian industry," *Int J Eng Res Technol*, 2019.
[2] L.-O. Larsson, N. Palmquist, V. Cars, and J. K. Larsson, "High quality aluminium welding–a key factor in future car body production," *Svetsaren*, vol. 54, no. 2, pp. 17–24, 2000.
[3] P. Dario, B. Hannaford, and A. Menciassi, "Smart surgical tools and augmenting devices," *IEEE transactions on robotics and automation*, vol. 19, no. 5, pp. 782–792, 2003.
[4] H. Zhang, J. Li, K. Kong, and S. Wang, "System design of a novel minimally invasive surgical robot that combines the advantages of mis techniques and robotic technology," *IEEE Access*, vol. 8, pp. 41147–41161, 2020.

[5] A. S. Niyetkaliyev, S. Hussain, M. H. Ghayesh, and G. Alici, "Review on design and control aspects of robotic shoulder rehabilitation orthoses," *IEEE Transactions on Human-Machine Systems*, vol. 47, no. 6, pp. 1134–1145, 2017.

[6] A. Misra and A. Agah, "Design of a robotic exoskeleton arm for rehabilitation," 2002.

[7] Y. Zhang, "The application of artificial intelligence in logistics and express delivery," in *Journal of Physics: Conference Series*, vol. 1325, p. 012085, IOP Publishing, 2019.

[8] B. Kokane Sanket, S. Kalamnurikar Shalaka, and B. Khose Suyog, "Robotics in food processing industries: A review," *The Pharma Innovation Journal*, vol. 11, no. 7S, pp. 948–953, 2022.

[9] A. Hamori, J. Lengyel, and B. Resko, "3dof drawing robot using lego-nxt," in *2011 15th IEEE International Conference on Intelligent Engineering Systems*, pp. 293–295, IEEE, 2011.

[10] W. Rong, Z. Li, W. Zhang, and L. Sun, "An improved canny edge detection algorithm," in *2014 IEEE international conference on mechatronics and automation*, pp. 577–582, IEEE, 2014.

[11] Y. Gu and W. Zhang, "Qr code recognition based on image processing," in *international conference on information science and technology*, pp. 733–736, IEEE, 2011.

[12] A. V. Goldberg and C. Harrelson, "Computing the shortest path: A search meets graph theory.," in *SODA*, vol. 5, pp. 156–165, 2005.

[13] A. Candra, M. A. Budiman, and K. Hartanto, "Dijkstra's and a-star in finding the shortest path: a tutorial," in *2020 International Conference on Data Science, Artificial Intelligence, and Business Analytics (DATABIA)*, pp. 28–32, IEEE, 2020.