Dynamic Trajectory Generation for LPD Surgery

Sundar Vengadeswaran, Sagar Syal, and Bren Gelyn Padlan
Systems and Computer Engineering,
Carleton University, Ottawa, ON, Canada
sundarvengadeswaran@cmail.carleton.ca, sagarsyal@cmail.carleton.ca, brengelynpadlan@cmail.carleton.ca

***Index Terms*—Robotic Surgery, Laparoscopy, Pancreaticoduodenectomy**

I. Introduction

A. Motivation

Pancreaticoduodenectomy, commonly referred to as the Whipple procedure, is a complex and intricate surgical operation used to treat the spread of pancreatic cancer [1]. Pancreatic cancer often arises at the head of the pancreas, however, the surrounding organs can also be affected [1]. As such, the procedure is considered to be extremely dangerous since it usually involves the removal of not only the head of the pancreas, but also portions of the small intestine (duodenum), gallbladder, bile duct, and stomach [1]. As a result, due to the reality of removing organs, patients of the surgery go through extensive amounts of blood loss, and an increased recovery time [1].

Alternatively, a safer, minimally invasive approach to this surgery is the Laparoscopic Pancreaticoduodenectomy (LPD) procedure, which involves making multiple small incisions along the abdomen of the patient and removing the head of the pancreas using a laparoscope [2]. In LPD surgery, laparoscopic needles, and a camera are inserted through the incisions, with the surgeon operating on the pancreas using camera vision. This laparoscopic approach is better than traditional open surgery as it reduces blood loss and the risk of complications [2]. Nevertheless, LPD is still considered one of the most complex operations for a surgeon to complete [2].

B. Problem Description / Challenges

LPD surgery is characterized by its minimally invasive approach, however it faces inherent risks such as organ and vascular injuries as it is an extremely difficult operation [3]. Owing to the fact that the pancreas is situated deep in the body relative to any possible incision points, it is a difficult area to reach through any laparoscopic approach [3]. Despite technological advancements facilitating complex surgeries, these risks highlight the urgent need for enhanced surgical entry techniques and effective complication management. Additionally, with the first case of LPD surgery being successfully completed in 1994, advancements in LPD techniques have not progressed in comparison to other surgical technologies, due to laparoscopic restrictions [3]. This project aims to address the challenges mentioned above, advancing patient safety and surgical outcomes in the field of laparoscopy.

C. Existing Solutions

The initial cases of LPD surgery were completed using non-robotic techniques, yet, advancements were hindered for approximately a decade due to the nature of the laparoscopic procedure [3]. Thus, in 2001, the first case of Robotic Pancreaticoduodenectomy (RPD) was successfully completed [4]. The RPD procedure allows surgeons to operate on the patient using robotic arms, yet the risk of injuring organs and vascular structures is still prevalent, as the arms are still teleoperated [4].

D. Proposed Solution

This project proposes a solution to enhance robotic assistance for LPD surgeries by developing a system that can precisely detect vascular structures and create a dynamic trajectory to safely reach the area of the pancreas. To achieve this objective, an algorithm utilizing OpenCV identifies obstacles, such as organs and vascular structures, based on their bright colors, serving as a proof of concept. This algorithm reads depth measurements from an ultrasonic sensor to avoid these obstacles dynamically, generating a new trajectory that takes the closest unobstructed path. This involved calculating the directions of the sensor and camera, and relating it to the correct axis of the robot's joints so that the depth measurement can be used accurately. As a proof of concept, a regular web camera and HC-SR04 ultrasonic sensor was attached to the end effector of the Meca500 Robotic Arm to simulate a robotic arm in an LPD procedure.

II. System Design

The physical system was designed so that the web camera and ultrasonic sensor would have the most direct line of sight to any structures that may be obstructing the path to the pancreas. The web camera was mounted on top of the Meca500's end effector, and the ultrasonic sensor was mounted below. The physical offset between the camera and ultrasonic sensor was also accounted for and, can be seen in section III.B. The system was developed in three separate streams; Obstacle and Structure Recognition, Distance and Position Calculation, and Trajectory Generation. The system operates through these components in this specific order. In order to simulate the obstacles, such as other organs and vascular structures near the pancreas, semi-spheres were used and, were colour-coded to differentiate between obstacles and the pancreas itself.

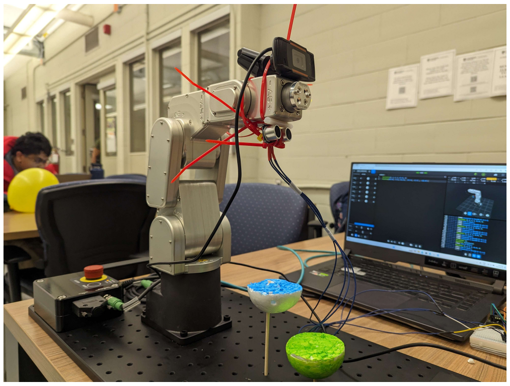

Fig. 1. The HC-SR04 and Web Camera mounted to the Meca500

III. Development Methodology

A. Structure Recognition

The algorithm utilized OpenCV libraries to detect any obstacles within the camera's frame. Furthermore, two separate functionalities were developed; obstacle detection, and localization.

Using OpenCV's VideoCapture library, the algorithm was successful in recognizing obstacles based on their specific colour and hue. Blue and green were selected for their distinction, with blue representing potential obstacles like organs and vascular structures and green symbolizing the a potential pancreas, demonstrating the algorithms capabilities in a proof of concept scenario.

The library was then used to detect the edges of the detected objects given their circular shape, as can be seen in Figure 2. The edges were represented by a circumference line, and a centre point. This centre point was then localized, with reference to the edges of the camera frame. The program completed this operation by using the pixel count between the obstacle and edge of the camera frame, after which it converted the measurement to millimetres.

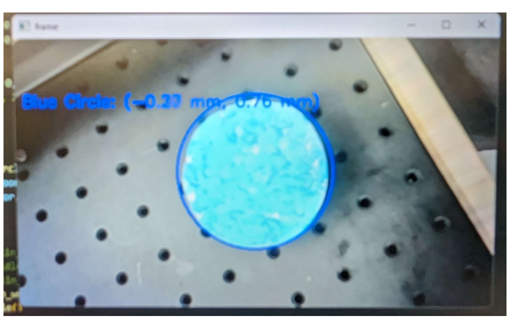

Fig. 2. Camera Detection Software recognizing blue obstacle

B. Distance and Position Calculation

The position of the object with respect to the camera frame was obtained from the localization of the object's center point, as previously mentioned. Given this position, the x and y values of the object in the camera frame were determined.

The HC-SR04 ultrasonic sensor was then utilized to implement depth detection. After the camera localizes the obstacle in its frame, the robot, with the ultrasonic sensor mounted underneath, would move linearly downwards at a distance between the camera and the ultrasonic sensor.

The distance readings acquired from the ultrasonic were processed by obtaining three consecutive values. The relative differences were calculated between these readings to obtain an accurate value of the depth. This technique filters out anomalies from the readings.

Using the depth measurement as z value and the obstacle position as x and y values, the three-dimensional position of the obstacle was found. This point in the Camera Reference Frame (CRF) is then transformed to the World Reference Frame (WRF) using a transformation matrix that conforms to the Meca500 convention via current axes.

C. Trajectory Generation

In order to develop a dynamic trajectory generation algorithm, two possibilities were considered. The first of which was to entirely generate a new trajectory from beginning to end, using an efficient planning algorithm, such as the Rapidly-exploring Random Tree model [5]. This algorithm would quickly simulate several tree paths in all directions from the start point, in this case, the end effector [5]. The trees would then multiply into several child branches, until a branch reaches the desired end position, being the pancreas [5]. Obstacles detected by the camera would be represented as frontiers that the branches would have to generate around [5]. The algorithm would then subsequently choose the most efficient path with the least amount of branches to the pancreas location [5].

However, due to the complex nature of this algorithm, the final system design attempted to employ a much simpler, intersection-based trajectory algorithm. Through this model, an initial parametric line denoted as L1, was made from the end effector to the centre of the detected pancreas, as seen in Figure 3. Given the centre point location, and circumference of an obstacle in front of the pancreas, a 2D plane with circular boundaries was then created to represent the obstacle's point in space. The circular boundaries, are fine-tuned to the obstacle's radius.

The algorithm then compared the initial parametric line with the circular plane made for the obstacle, in essence determining whether or not an intersection exists, denoted as P1. If an intersection was recognized, the algorithm attempted to generate a second parametric line around the circumference of the obstacle, denoted as L2, and subsequently generated several small lines, denoted as Lx, from the original intersection point, P1, to the points on the L2. The small line, Lx, which formed the smallest angle with the initial parametric line, L1, would then lead to the point on the edge of the obstacle (circumference) closest to the robot's end effector, denoted by P2, as seen in Figure 3.

Ultimately, creating a constraint beside P2, which the end effector is forced to travel through, would result in the closest

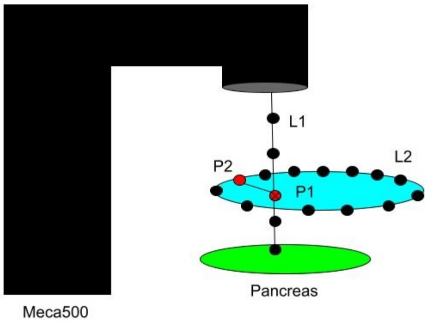

Fig. 3. Parametric Lines and Intersection Point on Obstacle

and most efficient unobstructed path to the pancreas. Nevertheless, when implemented in several tests, it was determined that recognizing a intersection based on a comparison between a parametric line and a 2d plane, was unreliable. Thus, the final iteration of the trajectory generator always created the robot's end effector constraint 2mm to the right-most point on L2 of the obstacle detected. Although not as dynamic and efficient as the previously described algorithms, the final iteration still generated the trajectory constraints based on the obstacle's position in the WRF.

D. System Implementation

An overview of the top-level system implementation can be seen in Figure 4. The distance and position transformations allow for the integration of the obstacle detection and trajectory generation components.

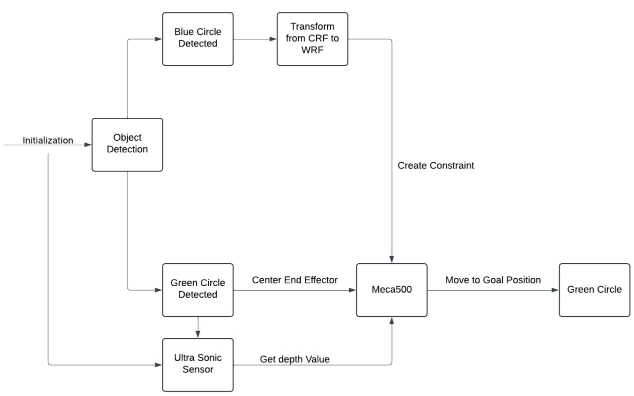

Fig. 4. System Block Diagram

IV. Experimental Validation

To test the functionality of the system design, a simple simulation of the LPD procedure was developed. The subsections below elaborate on the materials used, the experimental setup, and the findings.

A. Materials

To implement the proposed solution, the following materials and hardware components were utilized.

1) **Meca500 Robotic Arm**: This was the robot used to perform a simulation of the LPD procedure.
2) **Web Camera**: This was utilized to obtain the position of the obstacles in the camera reference frame.
3) **HC-SR04 Ultrasonic Sensor**: This component was applied to obtain the depth distance of an obstacle from the camera.
4) **Zip Ties**: These are used to attach the web camera and ultrasonic sensor to the tip of the robot.
5) **Styrofoam Obstacles**: These act as the 'dummy' vascular structures and the pancreas itself in the system. From Figure 1 above, the blue Styrofoam represents the structures around the pancreas, while the green one represents the pancreas.
6) **Wooden Stands**: These enable the depth difference between the pancreas and the vascular structures in its proximity.

B. Setup

The setup of the system, as a proof of concept, uses the Meca500 robot with the web camera mounted above the robot's arm and ultrasonic sensor beneath it as well. For this design, the measured length between the camera and ultrasonic sensor was 40mm. In front of the robot, the styrofoam balls were placed propped up by the wooden sticks. This setup can be seen in Figure 5.

Initially, the robot was reset to its zeroed position. Then, the fifth joint was moved by approximately 87 degrees to make the robot flange to point downwards. The setup in Figure 5 displays how the robot appears when the specified parameter mentioned previously was applied.

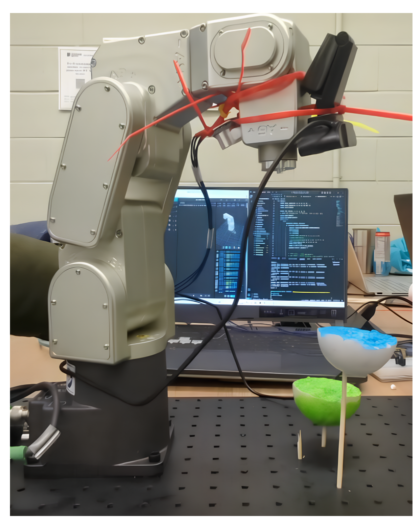

Fig. 5. Initial Position of Robot

C. Experiments

The system was tested using three test scenarios: obstacle was in fixed position, obstacle was in a fixed position then was moved randomly inside the workspace, and lastly, the obstacle and target were placed together in the workspace. These experiments aimed to test obstacle recognition, position calculation, and trajectory planning algorithms.

In the first test scenario, the functionality of the obstacle recognition was tested. It was expected that the system would successfully detect the obstacle, identify its color and shape, and sends an (x,y) point to the robot. The robot must move to this desired point.

For the second test, the system was expected to dynamically detect an obstacle that was re-positioned. The obstacle would be moved at a random position after the robot centers the object. The robot must dynamically move as the obstacle is moved.

The third test included placing two different structures (one blue, one green) in the workspace. The obstacle recognition was expected to find and identify the relationship between the structures. The camera module should return the (x,y) point of the obstacle. The depth measurement must then be read from the ultrasonic sensor and set as the z value of the obstacle. The robot must move towards the green structure without hitting the other structures.

D. Results

The first and second experiments described previously demonstrated that the obstacle recognition algorithm functions as intended. The tests showed that the algorithm accurately recognized both a stationary and moving obstacle. The robot also moved to the desired position, resulting in the object being at the center of the camera frame. Additionally, the robot also dynamically adjusted itself when the obstacle was re-positioned. However, an issue was encountered where the robot overshoots when trying to move to the position specified by the camera module, thereby, it takes around three to five attempts for the robot to eventually center the obstacle.

On the other hand, the last test confirmed the functionality of the integrated system. This integrated system included all three parts of the system: object recognition, position calculation, and trajectory generation. The system successfully detected the obstacle (blue structure) and the target (green structure) and obtained their x,y, and z values. However, the trajectory path algorithm was not able to determine that there was an obstacle between the tip of the robot and the target. An alternative solution was to use a simpler trajectory generator discussed in the methodology section. Using this, the robot was then able to smoothly maneuvered through the workspace avoiding the obstacle to reach the target structure. The system performed the expected behaviour.

V. Conclusion and Discussion

To reiterate, the results of the experiments demonstrate that the structure recognition algorithm is functional, as the Meca500 successfully identifies two distinct obstacles, represented by blue and green circles. This performance matches the expected behavior for the algorithm. However, the trajectory generation algorithm did not perform as expected during testing, leading to the adoption of a simpler method for generating trajectory points.

For future improvements, a dynamic trajectory generation would be implemented as originally planned. The Klampt Python Trajectory Algorithm could be utilized to generate the trajectory map. The structural recognition can also be improved by creating a more accurate colour scheme, which will enable the algorithm to easily determine the distinctions and relationships between each structure.

References

[1] J. R. D'Cruz, S. Misra, and S. Shamsudeen, *Pancreaticoduodenectomy*. StatPearls Publishing, July 2023.

[2] H. Li, X. Zhou, D. Ying, and S. Zheng, "Laparoscopic pancreaticoduodenectomy," *Hepatobiliary Surg. Nutr.*, vol. 3, pp. 421–422, Dec. 2014.

[3] J. Merkow, A. Paniccia, and B. H. Edil, "Laparoscopic pancreaticoduodenectomy: a descriptive and comparative review," *Chin. J. Cancer Res.*, vol. 27, pp. 368–375, Aug. 2015.

[4] P. C. Giulianotti, A. Mangano, R. E. Bustos, F. Gheza, E. Fernandes, M. A. Masrur, A. Gangemi, and F. M. Bianco, "Operative technique in robotic pancreaticoduodenectomy (RPD) at university of illinois at chicago (UIC): 17 steps standardized technique," *Surg. Endosc.*, vol. 32, pp. 4329–4336, Oct. 2018.

[5] S. Karaman, M. R. Walter, A. Perez, E. Frazzoli, and S. Teller, "Anytime motion planning using the RRT," in *2011 IEEE International Conference on Robotics and Automation*, IEEE, May 2011.