Muralidhar Kuluru, Shiva Sirasala, Venkata Jammalamadaka, Moritz Spiller*, Thomas Sühn, Alfredo Illanes, Axel Boese, Michael Friebe

Collaborative Robot as Scrub Nurse

Abstract: Under-staffing of nurses is a significant problem in most countries. It is expected to rise in the coming years, making it challenging to perform crucial tasks like assessing a patient's condition, assisting the surgeon in medical procedures, catheterization and Blood Transfusion etc., Automation of some essential tasks would be a viable idea to overcome this shortage of nurses. One such task intended to automate is the role of a 'Scrub Nurse' by using a robotic arm to hand over the surgical instruments. In this project, we propose to use a Collaborative Robotic-arm as a Scrub nurse that can be controlled with voice commands. The robotic arm was programmed to reach the specified position of the instruments placed on the table equipped with a voice recognition module to recognize the requested surgical instrument. When the Surgeon says "Pick Instrument", the arm picks up the instrument from the table and moves it over to the prior defined handover position. The Surgeon can take over the instrument by saying the command "Drop". Safe pathways for automatic movement of arm and handover position will be predefined by the Surgeon manually. This concept was developed considering the convenience of the Surgeon and the patient's safety, tested for collision, noisy environments, positioning failures and accuracy in grasping the instruments. Limitations that need to be considered in future work are the recognition of voice commands which as well as the returning of the instruments by the surgeon in a practical and safe way.

Keywords: speech recognition, minimally invasive surgery, surgical instrument, handover

https://doi.org/10.1515/cdbme-2021-1035

1 Introduction

The medical industry witnessed a lot of technological advancements in recent years and demand for surgical robots has increased rapidly. Many organisations and research groups have specialized in the field and have successfully built robots that are not only capable of performing simple operations but complex surgeries as well. Therefore, a variety of technologies is currently still under development to increase sensing capabilities and safety of such systems [3, 4]. Tele-manipulation systems for surgical procedures such as the "Zeus" [5] have been developed to increase the accuracy and efficiency for minimally invasive cardiac surgery by haptic methods. Cyberknife [6], a robotic surgery system that delivers radiation therapy to tumours with sub-millimetre precision treating tumours in areas of the body that were once surgically complex.

This draws attention to several other tasks where robots can be utilized or employed. The shortage of nurses is a major problem in most countries and nurses handle too many tasks [1]. One such task is the role of Scrub Nurse as it is often a stressful job. A scrub nurse is a trained person who assists the surgeons and the medical team in the operating room in handling instruments. Scrub Nurses require a high level of skill and a high level of concentration. They must be alert and careful at all times to the needs of the surgeon as cutting material is handled a small mistake can lead to devastating consequences. Some surgical interventions take place for hours that may lead to fatigue and loss of focus resulting in ineffective communication, and during emergency operations at night, regular nurses have to act as Scrub Nurses [2]. There are few robots that are developed to provide assistance in surgeries as well. The Penelope CS robot of the Robotic Systems & Technologies, Inc. (USA) is a well-known commercial device. A Scrub Nurse Robot System has been developed by Tokyo Denki University and the Tallinn University of Technology [8] that mimic the behaviour of a human SN. This model had a problem with reachability and could not attain good results. The Quirubot RSN [2] of nBio lab at Miguel Hernandez University was developed as an SN tool. Gestonurse [7] developed to assist surgeons by passing the surgical instruments during simple procedures uses hand gestures for recognising the instrument. The setup also used Voice modules as an alternative but the modules were ineffective in a noisy environment. The aim of this project is to develop a system using the robotic arm that can perform the task of Scrub Nurse capable of performing the tasks such as:

- Identify the instruments on the table
- Pick the instrument from the table when Surgeon says command "Pick Instrument"
- Move to the desired handover position specified by the Surgeon and release the instrument
- Interact in a comfortable way via GUI.

***Corresponding Author :Moritz Spiller**: INKA – Innovation Laboratory for Image Guided Therapy, Medizinische Fakultät, Otto-von-Guericke Universität Magdeburg, Deutschland moritz.spiller@med.ovgu.de
Muralidhar Kuluru, Shiva Sirasala, Venkata Jammalamadaka, Thomas Sühn, Alfredo Illanes, Axel Boese, Michael Friebe: INKA – Innovation Laboratory for Image Guided Therapy, Medizinische Fakultät, Otto-von-Guericke Universität Magdeburg, Deutschland

In this paper, we discuss about the various modules that were part of the project and the implementation procedure.

2 Methods

2.1 Components

Franka Emika Panda Robot* -** The Panda Robot from Franka Emika is a collaborative robot and has been used for handling instruments. A controller with an Ethernet connection enables the communication between the PC and arm through Local Area Network(LAN). ***ROS (Robot Operating System)* -** ROS is a set of software libraries and tools that provides a framework and also help your programs to build robot applications. We are using the latest stable version ROS Melodic. ***MoveIt Motion planner - Moving a robot arm accurately is a non-trivial task, as the arm should follow a sequence of values of every joint to move from start pose to goal pose which is to be calculated. To simplify this, MoveIt API w can be used to produces a motion plan for joints. ***Open CV* -** OpenCV is an open-source library that includes several hundreds of computer vision algorithms. ArUco Markers provided by OpenCV have been used for determining the position of the instruments that are placed on the table. An ArUco marker is a synthetic square marker composed of a wide black border and an inner binary matrix determining its identifier (id). ***Intel Real Sense Library* -** Intel RealSense SDK 2.0 is a cross-platform library for depth cameras developed by Intel. The library helps is used for calibrating and launching the camera. ***Pocket Sphinx*** - Sphinx is an offline speech recognition module developed by CMU Sphinx. Sphinx module mainly focuses on practical application development. This software supports several languages like US English, UK English, French, Mandarin, German, etc.

2.2 Speech Recognition Module

We implemented an offline speech to text module over the online modules considering the scenario and environment of the operation theatre. We have used two different offline Sphinx modules, the Pocket Sphinx model and the Acoustic model.

***Pocket Sphinx with Corpus file* -** In this model, a list of all the words to be recognized is converted to a corpus file, used as the input data to build a language model using Sphinx Language model tool available online. The corpus file consists of a list of words and a dictionary describing the pronunciation of all the words in that list. Using this model, it was possible to get the desired output for different voice profiles in silent

Figure 1: Surgical instruments positioned on the table with ArUco marker in the upper left corner.

surroundings. However, the performance dropped in under noisy environmental conditions. One drawback of this model is that the speaker must be very clear in pronunciation of the words. However, this model requires no training and is user-friendly, which works with different user's voices and frequency.

***Training an offline Acoustic model* -** On the other hand, the acoustic model is trained with the voice recordings (WAV files of speaking each of those commands, name of the instruments and keywords) from different users as the first step. Later a dataset is prepared which contains the curated data as corpus file *.fields* (list of files for training) and *.transcription* files (transcription for training). These files are used to train a new language model using the Sphinx train module in Sphinx We trained this model to overcome the limitation of the noisy environment but could not get desired results due to less training data. The main reason for making the acoustic model challenging to implement is that the data-set needs to be prepared or converted into a specific format to train it and send to the model. Obtaining lot of data for the instruments requires recordings from many users, and it seems like an arduous task.

2.3 Tray Registration

The objective of this module is to obtain the position of the instruments on the table and utilize them to move the robotic arm to the desired position. ArUco markers are used to detect the position of the instruments. The black border of Aruco marker facilitates its fast detection in the image, and the binary codification allows its identification and the application of error detection and correction techniques. A calibrated camera attached near the gripper of the robotic arm is used to detect the marker. When an ArUco marker is placed in the visibility range of the camera, the camera recognises the marker and display the marker's id and position. The detection file sends the position of the marker via ROS messages for every fraction of a second, and we do not need the position to change every second as this impacts the position of the instruments. Therefore, we need only one position and this is to be done when the SCAN button is clicked in the GUI. To facilitate the above requirement, we have employed a service and response functionality in a separate file, and when the SCAN is button is clicked, the file triggers a service, and a call is sent to the detection file via the main file which requests for the position of the ArUco at that instance of time. The response is sent back to the code and the position id reflected in the main file.

Since the detection is based on edges and borders, the camera recognises every marker placed in the visibility range and obtains positions of various markers. To overcome this, the id of the marker whose position is required has been specified in advance in the launch file, and this provides the position of that specific marker. The instruments are placed on the table, and the positions are taken concerning the marker, as shown in Fig 1. For a wider range of instruments, the corpus file should be prepared for the new set of instruments and then be included in the main code. The text-based module requires only the corpus file and for the voice-based model, the data with the voice recordings are required.

Each instrument has a fixed gap between them which can be changed if required. x-axis position is taken from the center of the marker to the center of the instrument vertically, and y-axis is the distance between two instruments horizontally. We have placed instruments 7 mm apart and these positions are later stored in a CSV file. This CSV file contains the names of the instruments and the above positions parallel to them. In case the instrument is included in the file but not present in the tray, the arm will still move to the position and perform the pick action. A mechanism to check for the presence of all instruments included in the file is not implemented yet. Further, the pick action can only be performed successfully if the orientation of the instrument is less than 10 degrees off to the original position. Misplacement above 10 degrees results in a change of center of gravity causing problems for the gripper to hold the instrument

2.4 Robot Control Module

To program the translations and movements of the robot, we have used MoveIt Motion planning API along with the libfranka library and used franka_ros to establish communication with the controller. We divided the pick to handover process of the instrument from the tray into the following events:

ArUco position **-** In this event, the robot end-effector moves over to the ArUco marker position with the given ArUco marker position, and this position is stored as ArUco position. ***Instrument position*** **-** In this event, the end effector translates to right above the instrument and offset in height is decreased by moving the effector down. ***Gripper open / gripper close*** **-** The end effector fingers are controlled in this event where the fingers are programmed to go further away or move towards each other according to the required task. ***Pre-grasp position*** **-** After the end effector is in the instrument position, the end effector is programmed to move further down, eliminating the offset in height such that the instrument is situated right in between the fingers of the end effector. ***Post-grasp position*** **-** This event is programmed to prevent irregular movements of the robot from the Pre-grasp position to handover. ***Handover position*** **-** This position is given by the user manually moving the robot before starting the program. Once the program is executed, it first stores this position as handover position, and the instrument is picked by the user at this position with special commands to release the instrument.

Figure 2: System setup

2.5 GUI

A user interface (UI) is a channel between human and computer interaction where a user will interact with a computer or machine to complete tasks. We have developed two graphical user interfaces, the main GUI for controlling the robot and another to get recordings of the instruments as a part of the Voice Acquisition for Speech Recognition model. The GUI is primarily developed to register the tray position via the ArUco marker through a depth-sensing camera attached to the robotic arm. In addition, The GUI helps to Start and Stop the arm and as an alternate to voice recognition. The GUI can also be used select the desired instrument, pick and drop and to change the handover position. Controlling the robotic arm with voice commands is the priority of the project and the GUI is safety-relevant backup.

We have developed another GUI for preparing a data set with the voice recordings of the instruments as the names of these instruments are uncommon, and a data set for such is not currently available. We recorded the voices with varying

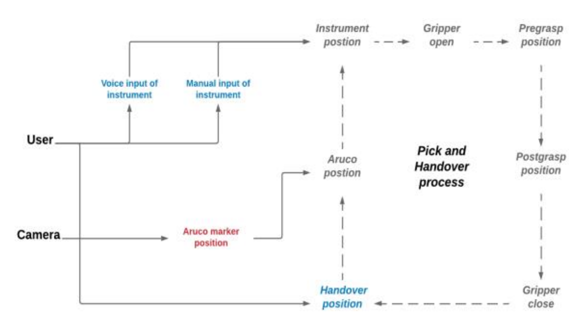

Figure 3: Process flow of the interaction of user and cobot.

distances of 30 cm, 60 cm and 1 m so as get a difference between the voices and avoid monotony. Voices were recorded with varying distances to the microphone in order to make the model more robust, in case of surgeon speaking with varying voice levels and distances to the system. The recordings are obtained in the .np and .wav files, and the .wav files are later compressed and converted to MS WAV format with a specific sample rate of 16KHz and 16 bits. This is used for training the model, and as the model has specific requirements, the above conversions were made.

3 Results

We took the instruments used in appendectomy surgery for testing our setup. We tested the setup in an environment with less noise using the Pocket Sphinx module with corpus file as our speech recognition module and it is sensitive to the phonetic sounds. Six users from different nationalities used up to 10 random instrument names. Little practice and training to pronounce the words in a way the system understands it, always provided better results. Major problems for recognition were improper dictation of the word, noise in the OR and low voice of the surgeon. The module was able to detect the given voice commands and give an accuracy around 85 %. The errors were mostly relating to the names of the instruments. As the module recognizes the names of the instruments based on the phonetic sounds, any word that was uttered in the OR that has a similar phonetic sound as of the instruments was recognized and it occurred with few instruments and when there is noise in the OR. Initially, in the pick and handover process, we only programmed the arm such that the robot from the ArUco position moves directly to the instrument position to pick the instrument and finally releases it in the handover position. After testing that process, the end effector would often collide with other instruments on the table while moving towards and away from the tray. To address this issue, we added pre-grasp and post-grasp position, as shown in Fig. 3, to ensure no collision occurs while picking and handing over the instrument. As we can see in Fig. 3 the ArUco maker position input from the camera is used to find the respective position. The user gives the inputs of the desired instrument via the customized GUI, causing the robot to move to the instrument position and performing the process gripper open > pregrasp position > postgrasp position > gripper close > handover position. After the robot control module was implemented the pick and handover process was checked >250 times for different instruments at different locations in the tray. The robot performed accurately in all cases without any errors, and using GUI; we were able to pick the correct requested instruments and release them without any issues. Due to the limited reachability of the arm, the working area is less compared to human nurse. But it would not be a problem if all the instruments are placed within the reach.

4 Conclusion

We have developed a Robotic arm that could act as a scrub nurse and after several tests and updates, the initial objectives have been achieved. The system could obtain the positions of the instruments placed on the table. Commands given by the surgeon can be detected by the voice module and the robotic arm was able to grab the instruments. The arm moves to the desired position without any collisions and handover the instrument to surgeon without any errors. And finally, the GUI can be used to start and stop the process, fetch ArUco position and pick instruments. We believe this system could bring a potential impact in various surgeries. As we have tested the setup in a calm environment, we were able to achieve great accuracy, but the OR environment will be noisy. If the phonetic sound of a word said in the OR matches with any of the names of the instruments in the corpus file, the module assumes it as the name of the instrument, and the Arm picks that instrument. We could overcome this problem if we use an acoustic model in which we train the model with the names of the instruments. If the voice recognition module is replaced, this setup has the potential to be used as a Scrub Nurse.

References

[1] L.M. Haddad et al., 2020 Dec 14. In: StatPearls [Internet]. Treasure Island (FL): StatPearls Publishing; 2021.

[2] C. Pérez-Vidal, et al., Steps in the development of a robotic scrub nurse. Robotics and Autonomous Systems, 60(6), 2012.

[3] A. Illanes, et al., Surgical audio information as base for haptic feedback in robotic-assisted procedures. Current Directions in Biomedical Engineering. 2020;6(1)

[4] C.H. Chen, et al. Texture differentiation using audio signal analysis with robotic interventional instruments. Computers in biology and medicine vol. 112 (2019): 103370.

[5] H. Reichenspurner, et al. "Use of the voice-controlled and computer-assisted surgical system ZEUS for endoscopic coronary artery bypass grafting." The Journal of thoracic and cardiovascular surgery 118.1 (1999): 11-16.

[6] W., Kilby, et al. "The CyberKnife® robotic radiosurgery system in 2010." Technology in cancer research & treatment 9.5 (2010): 433-452.

[7] M. Jacob, et al. "Collaboration with a robotic scrub nurse." Communications of the ACM 56.5 (2013): 68-75..

[8] F. Miyawaki, et al., Scrub nurse robot system-intraoperative motion analysis of a scrub nurse and timed-automata based model for surgery, IEEE Transactions on Industrial Electronics 52(5) (2005)