BAXTER PLAYING POOL

Koyya Shiva Karthik Reddy
Department of Electrical and Electronics Engineering
Rochester institute of Technology

Vishnunandan Venkatesh
Department of Electrical and Electronics Engineering
Rochester institute of Technology

***Abstract* ---The project focuses on accomplishing a complex task of playing pool with the industrial robot Baxter (14 DOF) made by Rethink Robotics. The process had many complex intermediate goals such as analysing various views so as to find the required orientation to make the shot. The task of finding the orientation was attempted with help of 3-D sensor Xtion as well as the Baxter's head camera.**

Once the desired orientation was found, visual servoing was accomplished using Baxter's hand camera to find the center of the ball so that the end effector can have a perfect strike. Baxter's inverse kinematics package provided by Baxter SDK was used during the entire course of the project.

KEYWORDS: PointCloud, OpenCV, ROS, PointCloud Library, Baxter, Inverse kinematics, desired view, Current View.

The paper majorly constitutes of V sections. Section I gives an introduction on robotics in broad, the recent improvements in the field and our approach to the problem. Section II discusses the related work in the field. Section III discusses flow of the system. Section IV discusses the results and limitations. Section V throws some light on the future scope of improvements.

I. INTRODUCTION

In today's modern world robotics has evolved a lot, robots accomplish complex tasks which a human body cannot achieve and with the improvement in processors, computation of complex problems is becoming easier for highly advanced robotic machines. In past 10 years or so drastic improvement in field of robotic vision had given a new dimension to the robotic world , affordable 3-D sensors like KINECT or XTION have opened the door to many new possibilities to be explored. Now with the help of vision and other sensors robots can be taught to understand their environments and perform actions with much more data and information. With various open sourced libraries such as open CV and pointcloud library robots can be trained to detect specific objects in their surroundings easily.

Industrial robots such as Baxter are human compliance robots and are highly precise and have their own on system processor and vision sensors that make the computation much faster. Platforms like Robot Operating system (ROS) have made it possible to build complex applications as it provides easy integration of various modules independent of the platform. Many built in packages in ROS include simulation and GUI which make understanding coding and visualizing of problems easier.

In our approach a Xtion sensor was placed on Baxter's wrist so as to record a desired view pointcloud wherein the cue ball and the pocket were perfectly aligned in a position which would result in a perfect shot (i.e. cue ball being hit into the pocket).Once the desired view was recoded the hand was moved to some random location where its current view was again analysed and recoded as a pointcloud. The pointcloud recorded as the current view was attempted to be aligned (registration) with the desired view based upon the unique features extracted from both the pointclouds so as find the required orientation. Upon analysis of the results due to inaccurate result and large computation time a secondary approach was choose to tackle the orientation problem wherein the Baxter's head camera was used to find the orientation between cue ball and the pocket.

After addressing the orientation part, using Baxter's hand camera and with help of Open CV the ball's center was tracked and the baxter's hand was moved in linear fashion with the same orientation so that the end effector (cue stick) would always be straight to the center of the ball and this would help the end effector strike the ball perfectly in the center.

II. RELATED WORK IN THE FIELD

In [1] the author discusses the problem concerned with 3-D pointclouds registration with concern being the inaccuracy and computation time of the conventional methods like Iterative closes point (ICP) and coherent point drift (CPD) algorithms, he comes up with a variation of CPD; fastCPD which is based on global convergent squared iterative EM scheme which improved the computation time of the registration process. Segmentation of the view of interest is also a challenging task as well computationally expensive as discussed in [2] where the author proposes choosing normal estimates based on tensor voting. The result shown improved performance even in noisy observations and missing data. In [3] author discusses another approach for segmentation where manual segmentation of objects from scenes were used and the automated approximation of objects with high level descriptors were made to recreate the required 3-D model. The algorithms working on point cloud are computationally expensive due to the large volume of data, therefore proper pre-processing of the pointclouds is a necessary step when working with large pointclouds, in [4] the author discusses a Super Voxels for pointcloud connectivity segmentation, which drastically reduces the time complexity of later process like feature generation and registration. The type of sensors also effect the result of experiment many sensors can be used for 3-D modelling like stereo camera, depth camera like Xtion or kinect, LIADARS, in [5] the author compares the performance of two widely used sensors namely Microsoft Kinect and Asus Xtion and concludes that the depth resolution of both sensors gets worse when the distance increases. In terms of depth accuracy, the depth sensor accuracy decreases when increasing the distance between the sensor and the planar surface. The Kinect sensor's accuracy is more sensitive than Xtion to radiant illuminated objects.

Visual Servoing plays an important role in helping the end effector align itself perfectly straight to the ball so as to make the perfect shot. Visual servoing can be divided into three groups namely position based, feature based and 2-1/2-D visual servoing. [6] Describes position based visual servoing. A moving camera was considered rather than a fixed camera. . The camera is moved about a single axis so as to compensate for the loss in depth. The paper deals with the complexities involved with deriving the position of a goal point using a single camera. Positional visual servoing has its perks as long as it can determine the position of an object definitively. If the object cannot be determined effectively then it would result in errors. [7] Talks about a method which can perform visual servoing using features from the camera rather than using positions. To reduce errors in visual servoing (error in the difference between the visual feedback and the tracked object position), positional visual servoing techniques use the camera calibration matrix and the Cartesian points of the object tracked. However if the position of the object cannot be determined effectively then this method would fail. The method proposed in this paper is to obtain feature points from the camera calibration matrix. Feature points would have an advantage over position vector points as the loss of information on depth would not destabilize them. Another way to minimize errors in visual servoing was proposed in [8] which used multiple cameras instead of single cameras. The eye in hand configuration was used here for visual servoing. Apart from the camera in the end effector a stereo camera setup was made. The proposed method was that the stereo system would determine the error and perform the error corrections and then the eye in hand camera would carry on with visual servoing.

[9] Talks about the Baxter robot tracking human body motion. Two approaches are used to control the joint of the robot, one being the inverse kinematics approach and the other being the vector approach. The paper concludes that the inverse kinematics approach is much better than the vector approach. The vector approach uses only four of the joints for controlling the positions of the arm and it also leads to a few errors which had to be stabilized. However, the inverse kinematics approach ensured all joints being used and gave satisfactory movements to the

joints provided there were inverse kinematic joint solutions for various positions. [10] Builds a kinematics model for the Baxter robot with the help of its universal robotic descriptive format (URDF) file provided by its SDK. The kinematics model is developed for the purpose of simulations and it discusses the D-H parameters of various joints of Baxter.

III. SYSTEM FLOW

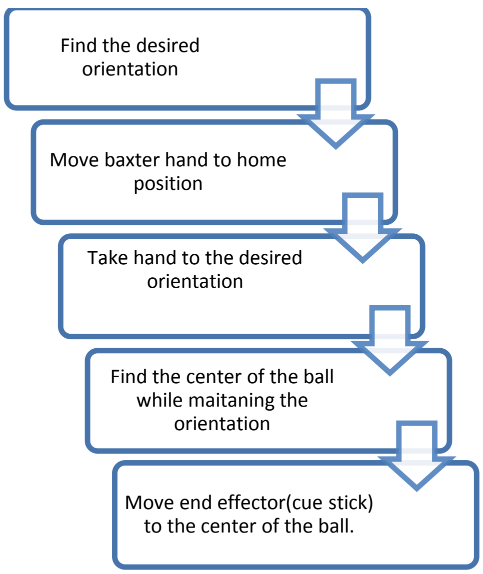

Fig. 1. Flow of the system

The entire process was performed as per the flow described in figure 1. This section is further sub divided, explaining each of the steps in further details.

A. *Orientation*

Finding the desired orientation to strike is one of the most important steps in the entire process. The process and available approaches tried to achieve are shown in the figure 2.

As mentioned in the introduction due to inaccurate results and large computation time when working with pointclouds, Baxter's head camera was used to find the orientation.

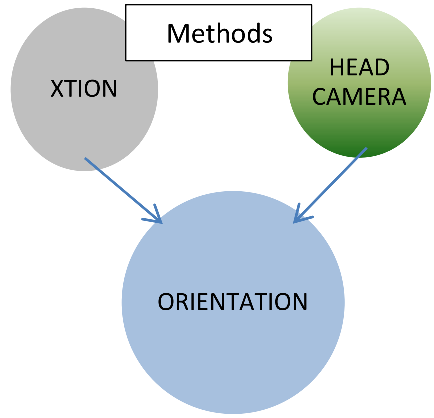

Fig. 2. Orientation

1) *Xtion and pointcloud*: Xtion depth sensor developed by Asus was used in the experiment to get the pointcloud (depth and RGB) data of the view. Pointcloud is set of data points which represent an extended surface of the object in some coordinate system which in Cartesian system is X, Y, Z where Z is the depth information(i.e. the distance of that point from the sensor) pointclouds are extensively used in 3-D modelling and other such related research. The pointcloud obtained using Xtion were was further processed using PointCloud Library [12] C++ bindings. The process overtaken on pointcloud to find the desired orientation is shown in figure 3.

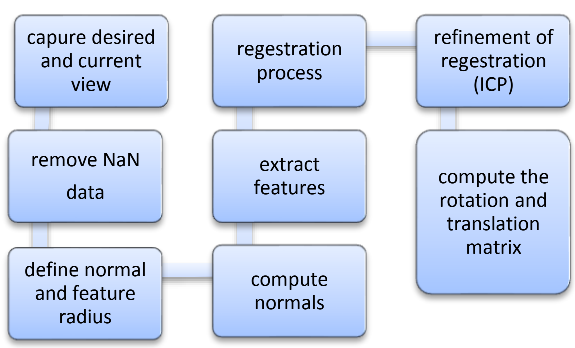

Fig. 3. Point Cloud processing

2) Head Camera: Using the head camera located above the display screen on the Baxter robot, the orientation was determined. Considering the fact that the Baxter robot was at a fixed position and the pool table arrangement was at a fixed distance from Baxter, the positions of the ball and the pocket was determined with the help of the 2-D pixel coordinates obtained from Baxter's head camera. Using the Open CV libraries in python the pool ball and the pocket were masked using various threshold methods. Contours were drawn around the ball and the pocket. The center was found using the moments function. Once the center points were found the slope between the pool ball and the pocket was calculated using mathematical operations and the angle was found in radians.

B. Baxter hand moments

Baxter robot is a 14 DOF industrial robot with 7 DOF in each arm. Each arm has seven rotational joints and eight links. The robot is said to be human complacent and is programmed to work in a safe way in any environment

Once the orientation is determined the right limb of Baxter is taken to a pre-determined home position. The orientation of this position is parallel to the Z axis of the end effector. The orientation angle obtained in Section A is then converted to the respective Baxter end effector quaternion and Baxter is commanded to align itself to the given orientation.

C. Visual servoing

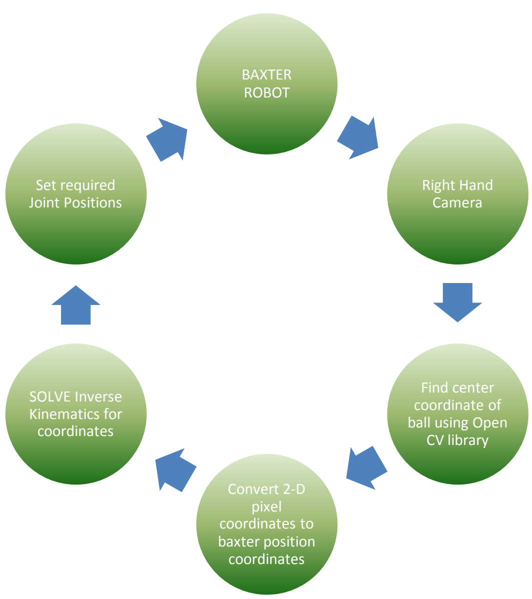

Fig. 4. Visual servoing

In this step Visual Servoing is used to ensure that the end effector (cue stick) in aligned with the center of the cue ball in the desired orientation already set using the head camera. Baxter's right hand camera is used along with Open CV libraries. The ball is masked from the image with the help of threshold functions. Dilating and Eroding is performed so as to remove small error blobs left on the masked image. Again using the moments function the center of the ball is calculated. To successfully move the hand based on the position of the center of the ball the cameras 2-D coordinates needs to be converted to Baxter's joint coordinates. Keeping the distance between Baxter's end effector and the pool table arrangement constant we were able to remove factor of depth from this process (i.e. keeping Z-axis coordinate of the end effector constant). Then the pixel/centimetre for each of the cameras 2-D axes was calculated. These pixel/centimetre values were mapped to Baxter's end effector translation values about their axes respectively. Once these values are obtained they were passed to the Inverse Kinematics solver provided by Baxter SDK [12] so as to obtain a joint solution for Baxter to move to. Visual Servoing is repeated till Baxter's end effector is perfectly aligned with the center of the ball keeping in mind that it is away from the ball at a constant depth.

D. End effector striking moment

Once the right arm of Baxter is in the specified orientation and is aligned with the center of the ball it then has to make a linear motion to strike the ball. This also means that Baxter needs to translate by its X axis by a certain amount and it needs to translate about its Y axis by the tangent of the orientation angle obtained. Since Baxter does not follow a linear interpolated form of movement when it moves from one point to another the translations made were divided into small increments to achieve linear movement. A much efficient way of doing this would be to use Jacobian matrices. But this would decrease the force with which the arm would strike the pool ball greatly.

IV. RESULTS AND LIMITATIONS

The game of pool was successfully played with help of Baxter’s vision sensors and using Baxter’s inverse kinematics package but because of fixed position of the Baxter robot the playing workspace was limited.

Following observations were made, the use of Xtion was not successful because to get a proper depth image of the view the Xtion should be at some specific distance and height but from that specific height the inverse kinematics failed more often than not. Moreover the registration process with the pointcloud was a computationally expensive operation.

The following limitations were in the system:

1. The pointcloud results were not accurate
2. Workspace was limited.
3. With the hitting action sufficient force was not be achieved and the ball was able to move only by a few centimetres.
4. Due to minute quiver movements of the Baxter's end effectors the center detection was a bit affected.

V. FUTURE WORK

Making the Baxter robot mobile would add to a greater flexibility in adding a much larger workspace for Baxter to play in. Once mobile the robot can overcome the limitations caused by the elementary inverse kinematics package by adjusting its position. One more improvement to bring into the project would be to use the pointcloud library more effectively to calculate the orientation and bring in the factor of depth also into the project with the help of the Xtion 3-D camera. A linear actuator could be placed at the end effector to substantially increase the force with which Baxter robot strikes the ball.

REFERENCES

[1] Min Lu; Jian Zhao; Yulan Guo; Jianping Ou; Li, J., "A 3D pointcloud registration algorithm based on fast coherent point drift," in Applied Imagery Pattern Recognition Workshop (AIPR), 2014 IEEE , vol., no., pp.1-6, 14-16 Oct. 2014

[2] Ming Liu; Pomerleau, F.; Colas, F.; Siegwart, R., "Normal estimation for pointcloud using GPU based sparse tensor voting," in Robotics and Biomimetics (ROBIO), 2012 IEEE International Conference on , vol., no., pp.91-96, 11-14 Dec. 2012

[3] Strand, M.; Dillmann, R., "Segmentation and approximation of objects in pointclouds using superquadrics," in Information and Automation, 2009. ICIA '09. International Conference on , vol., no., pp.887-892, 22-24 June 2009

[4] Papon, J.; Abramov, A.; Schoeler, M.; Worgotter, F., "Voxel Cloud Connectivity Segmentation - Supervoxels for Point Clouds," in Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on , vol., no., pp.2027-2034, 23-28 June 2013

[5] Haggag, H.; Hossny, M.; Filippidis, D.; Creighton, D.; Nahavandi, S.; Puri, V., "Measuring depth accuracy in RGBD cameras," in Signal Processing and Communication Systems (ICSPCS), 2013 7th International Conference on , vol., no., pp.1-7, 16-18 Dec. 2013

[6] Hespanha, J.P., "Single-camera visual servoing," in Decision and Control, 2000. Proceedings of the 39th IEEE Conference on , vol.3, no., pp.2533-2538 vol.3, 2000

[7] Navarro-Alarcon, D.; Yun-Hui Liu, "Lyapunov-stable eye-in-hand kinematic visual servoing with unstructured static feature points," in Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on , vol., no., pp.755-760, 14-18 Sept. 2014

[8] LianKui Qiu; Quanjun Song; Jianhe Lei; Yong Yu; Yunjian Ge, "Multi-Camera Based Robot Visual Servoing System," in Mechatronics and Automation, Proceedings of the 2006 IEEE International Conference on , vol., no., pp.1509-1514, 25-28 June 2006

[9] Reddivari, H.; Yang, C.; Ju, Z.; Liang, P.; Li, Z.; Xu, B., "Teleoperation control of Baxter robot using body motion tracking," in Multisensor Fusion and Information Integration for Intelligent Systems (MFI), 2014 International Conference on , vol., no., pp.1-6, 28-29 Sept. 2014

[10] Zhangfeng Ju; Chenguang Yang; Hongbin Ma, "Kinematics modeling and experimental verification of baxter robot," in Control Conference (CCC), 2014 33rd Chinese , vol., no., pp.8518-8523, 28-30 July 2014

[11] Pointclouds.org

[12] Sdk.rethinkrobotics.com