CS534 Fall 2013 Project Report

Visibility planning on the Baxter robot

Nandan Banerjee (*nbanerjee@wpi.edu)*

***Abstract*— The principal idea behind this project was to have a robotic manipulator reach a target while avoiding obstacles in the configuration space of the manipulator using visual servoing techniques. This project deals with one of the fundamental problems of visibility planning, i.e. the robot needs to find a spot for its hand camera which will give it a very good view of the target object and in turn will help in the localization of the target object. After the object has been localized, a regular planner can be used to plan for a trajectory to the target object. The problem of visibility planning has been tackled in this project using various AI optimization approaches.**

I. Introduction

The robot that has been used in this project is the Baxter[1] robot in the ARC[2] lab. The Baxter robot is a comparatively inexpensive robot with two 7 DOF arms and a head capable of being tilted vertically and horizontally. It also comes equipped with two hand cameras and one head mounted camera. The main idea behind this project is to find a pose for the hand camera which will ensure that the target object is visible, i.e. the important feature points on the target object are visible. Initially, it is assumed that a part of the object is visible, i.e. not all the feature points are visible but a few. Using these few feature points and given the geometric model of the object, the object is reconstructed in the world space using a PnP solver. Now that the object has been localized to some extent, the problem of visibility planning gets simplified a bit, i.e. the camera needs to see the reconstructed object. This problem is tackled using one of the optimization methods known as Simulated Annealing. A globally optimal solution is not desired as an optimal solution which gives the location of the feature points of the target object will be sufficient. Simulated Annealing does a very good job of arriving at optimal solutions. A suitable cost function/energy function was chosen which incorporated the visibility of the object (using a ray tracing metric), the manipulability of the arm and the distance of the pose in question from the object. A suitable temperature schedule based on an exponentially decreasing function is chosen and a probability function is chosen. The neighbourhood sampling is done in the task space of the robot and every sample is IK verified before proceeding further. After the algorithm outputs a good solution, the hand is moved to that location. If the object's feature points are in the field of view of the camera, then it is desired to use the PnP solver again to localize the object accurately and move to it. Other methods like random sampling and random neighbourhood sampling have also been used to compare results with the performance of Simulated Annealing.

[1]The Baxter robot was at my disposal all the time. I have also been developing software APIs for the Baxter for the past couple of months.

[2]All work was done at the ARC lab under the guidance of Prof. Dmitry Berenson.

II. Related Work

Work in the field of visibility planning has been done before at various universities. Kuffner and Vahrenkamp have tackled this problem in their paper where they distributed the task of motion planning into two parts – motion planning to get to a position near the target object which makes sure that the target object is in the field of view of the camera and from then on, use visual servoing to grasp the object. Tsai and Allen have done significant work in a model based and task driven vision system that automatically plans vision sensor parameters so that task requirements are satisfied. In one of their earlier works, they figured out the important features in an object that should be attainable like if the object is in focus, magnification and other parameters.

Cheng and Tsai have worked on this problem by defining various parameters and simulating the entire model in a graphics engine. They designed the models in their world and in place of a camera, used a light source and figured out which parts of the target object was illuminated. This gave an idea about the visible parts of an object. Work has also been done in motion planning taking into account occlusion of the object, i.e. the goal was to move the manipulator towards the object while making sure that the object had a significant part inside the field of view of the camera. Without visual sensors, work has been done by Landa, Galkowski and others from UCLA worked in the area of path planning and mapping with limited sensor data which is very similar to this project.

III. Problem Statement

The essential problem that is being tackled in this project is visibility planning of a robot manipulator. To understand it better, the system setup is described. Initially the exact location of the target/goal object is unknown. However, a small portion of the object is visible through the hand camera of the arm. From this visible portion, an approximate estimation of the location and orientation of the object is done. Then using Simulated Annealing, the task space of the hand camera is sampled and using a suitable energy function and temperature schedule, the least energy position is chosen which promises a better view of the target object. Then the hand camera is moved to the new "suitable" location. This goes on iteratively until the entire object is within the frame of the camera. Once the object is entirely visible, the location of the object can be better estimated. Then the arm is moved to the object.

Taking some ideas from the work of Cheng and Tsai, instead of reconstructing the entire world in a graphics engine, primitive ray tracing functions have been used to determine if an object is visible or not. Essentially, this method consists of shooting rays from the camera inside the field of view cone and figuring out the visible portion of an object by calculating the number of rays that hit the object.

IV. OBJECT RECONSTRUCTION

Object reconstruction is in itself a very varied field. In this project, object reconstruction is the part that deals with the reconstruction of the object features from the initial partial view of the object through the hand camera. Because the ultimate goal of this project was to test how simulated annealing performed in estimating a better viewing position, a very simple and naive approach was used to reconstruct the object.

Initially, the user is presented with a video feed from the hand camera. The user then selects the feature vectors that are visible. The geometric model of the target object is known. Using the geometric model of the object and the feature vectors as input to a generic PnP solver, the transformation matrix between the object and the camera frame is found. Using this transformation matrix, the approximate position of the object is found and is drawn on to a visualizer.

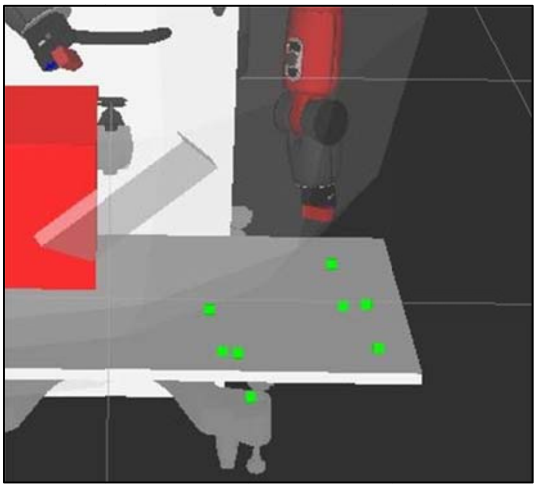

Fig 1: Green dots represent the reconstructed object from the camera. (Visualized in RViz)

V. SIMULATED ANNEALING

Simulated Annealing is a generic metaheuristic method for the global optimization problem of locating a good approximation to the global optimum of a given function in a large search space. This method is based on the metal annealing method in metallurgy which involves heating and controlled cooling of a material to increase the size of its crystals and reduce the defects. Both the defects and size are dependent on the thermodynamic energies. The same amount of cooling brings the same amount of decrease in temperature but comes with a bigger or smaller decrease in the thermodynamic free energy depending on the rate that it occurs, with a slower rate producing a bigger decrease. This notion of slow cooling is implemented in the Simulated Annealing algorithm as a slow decrease in the probability of accepting worse solutions as it explores the solution space. Accepting worse solutions is a fundamental property of metaheuristics because it allows for a more extensive search for the optimal solution.

In practice, simulated annealing is a very straightforward and simple algorithm to implement. It consists of an efficient neighbourhood sampler, an energy function, a temperature schedule and a probability function. The algorithm for Simulated Annealing is as follows –

```
Simulated Annealing
s ← s_0
e ← E(s)
s_best ← s
e_best ← e
k ← 0
while k < k_max and e > e_max
  T ← temperature(k/k_max)
  s_new ← neighbour(s)
  e_new ← E(s_new)
  if Prob(e, e_new, T) > P_thresh then
    s ← s_new and e ← e_new
  if e_new < e_best then
    s_best ← s_new and e_best ← e_new
k ← k + 1
return s_best
```

Initially, a random state is chosen. The energy for this state is calculated. The temperature is decreased according to the temperature schedule. The next state is sampled. The energy of this state is calculated. If the energy is less than the current energy, then this state is accepted. If the energy is more, then the probability function which takes the temperature and the energies of the current and the next states gives out a probability of acceptance of the new state. If it is over a certain threshold, it is accepted or else it is rejected. This goes on till the temperature reaches 0. The best energy state is stored and at the end of the algorithm, this best energy state is the one that is chosen.

A. *Neighbourhood Sampling*

The sampling space is the task space of the hand camera of the robot. Essentially the task space consists of all the positions and orientations attainable by the hand camera irrespective of collisions with itself or the environment. Random camera position samples are generated. Sampling orientations is a difficult task to do and because in this particular problem, the final orientation of the camera is known, i.e. it should point towards the target object. To specify the orientation, we need to determine the quaternions that correspond to the transformation of the Z axis in the base frame to the vector V_{cam_cent} joining the centroid of the reconstructed object and the camera position. V_{cam_cent} is essentially the Z axis of the camera. So, by rotating the Z axis of the base frame about $(V_{cam\_cent} \times Z_{base})$ by an angle given by $cos^{-1}(V_{cam\_cent}.Z_{base})$, the required orientation can be obtained. Since the angle and axis is already there, the quaternions are computed for the orientation. In this way, samples are generated for the entire task space by just generating random positions and calculating the orientations.

After a sample has been generated, the sample is tested if it has a valid IK solution and if it is in collision with any of the world objects. If it is not, then this is regarded as a valid sample.

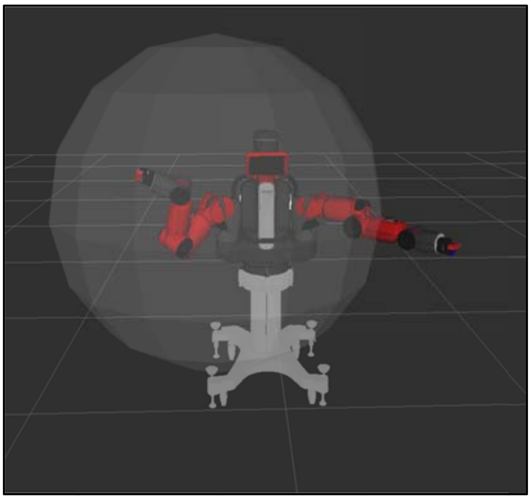

Fig 2: Grey sphere about the right arm is the task space of the right arm.

The first sample in the algorithm s_0 is a random sample. Then on, the samples are generated randomly within a 5cm neighbourhood of the previous sample.

B. Energy function

A function which encompassed all the essential attributes of a good configuration and a quantitative representation of the visible object features was devised. Two parameters (manipulability and the distance of the object from the sampled camera position) quantifying the desirability of the sampled configuration is used. The manipulability measure gives a better score for configurations that are farther away from a singular configuration. It is calculated as follows –

$$M = \min(eigs(JJ^T))$$

J is the Jacobian of the arm at that configuration and the manipulability score M is the minimum eigenvalue of the product of the Jacobian matrix.

The distance of the object from the sampled camera position is also important because of the camera estimation. If the camera is too far away from the object, then it will not be able to do a good job at estimating the position of the object.

Ray tracing as described above in section III is used to determine the visibility of the object in the image frame at the sampled location. The ray tracing approach is applied twice. In the first case, the number of rays passing through the object ($R1_{object}$) and the number of rays passing through the obstacles and then the object is determined ($R1_{obstacle\ and\ object}$). In the second case, the same thing is done but instead of all the rays, only the rays that pass through the feature points of the object are taken into account ($R2_{object}\ and\ R2_{obstacle\ and\ object}$). Then these are normalized by taking the ratio with just the rays that pass through the object in both the cases ($R1_{object}\ and\ R2_{object}$). So, the final energy function is –

$$\begin{aligned} Energy = & -[\lambda_1.\left(\frac{R1_{object} - R1_{obstacle\ and\ object}}{R1_{object}}\right) \\ & + \lambda_2.\left(\frac{R2_{object} - R2_{obstacle\ and\ object}}{6}\right) \\ & + \alpha.M + \beta.dist_{camera\_centroid}] \end{aligned}$$

C. Temperature Schedule

An exponentially decreasing temperature schedule (annealing schedule) was used in this case. One of the important criteria for a temperature schedule is that it is supposed to hit 0 at the end to let the algorithm know that its time to stop. But because the exponential never reaches absolute 0, a stop condition was set at T = 0.001.

$$\mathrm{T} = e^{-Ai}$$

The coefficient A determines the curviness of the schedule. The value of A was chosen using the following formula –

$$A = \frac{\mathbf{1}}{\boldsymbol{N}}\log\left(\frac{T_0}{T_N}\right)$$

T_0 is the initial temperature, T_N is the lowest attainable temperature and N is the number of steps that is desired to get to the minimum temperature T_N.

D. Probability Function

The probability function selected here is the one used by Kirkpatrick et al which has been widely accepted over the years.

$$\begin{aligned} P &= e^{\frac{-(E_{new} - E_{prev})}{T}}, && when\ E_{new} > E_{prev} \\ &= 1, && when\ E_{new} < E_{prev} \\ &= 0, && when\ E_{new} = E_{prev} \end{aligned}$$

E_{new} is the energy of the sampled state and E_{prev} is the energy of the current state. At the start of the process, E_{prev} is set to 0.

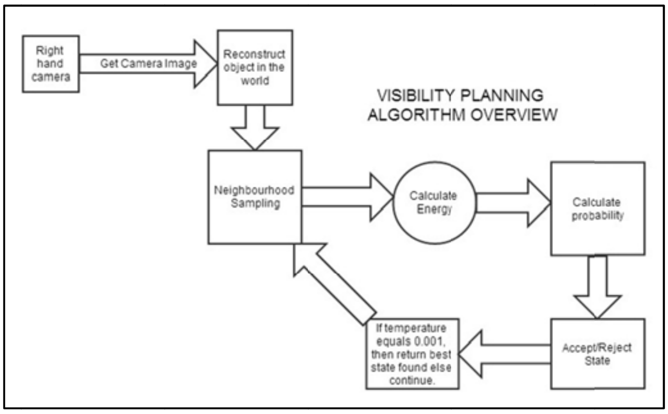

Fig 3: Flow diagram of the Visibility planning algorithm pipeline.

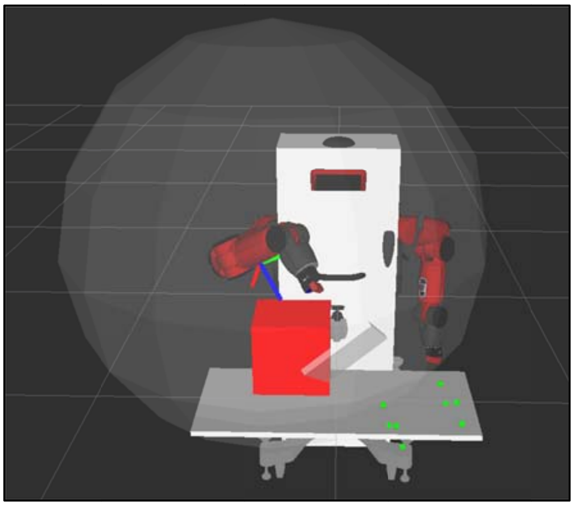

Fig 4: Screenshot depicting the sampled configuration where the algorithm is in the process of finding a better configuration. Red box is the obstacle, gray rectangular frame is an approximation of the field of view at that pose.

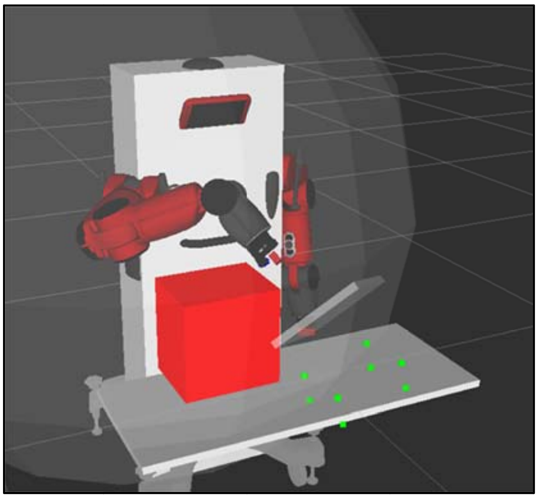

Fig 5: After the simulated annealing has been applied and the optimal goal determined.

VI. RESULTS

The results obtained in this project were quite promising. This method was compared with two naive approaches – random sampling and random neighbourhood sampling. The criterion for success was chosen as the visibility of six feature points on the object after moving to the "supposedly" better viewing position as determined by the method used. Random sampling's biggest disadvantage was sampling of valid IK configurations. The poses it sampled were mostly wrong. Random Neighbourhood sampling also didn't give anything better. Random neighbourhood sampling gave more valid IK poses because the samples were being taken in the neighbourhood of a correct pose sample. It still didn't perform well and timed out usually with really bad scores. Simulated Annealing performed successfully on 17 occasions with an average restart rate of 4.96. The time taken by simulated annealing on average was about 4 minutes. The following table shows the results in a quantitative format –

Method	Trials	Successes	Remarks
Simulated Annealing	25	17	4.96 restarts (3 to 4 minutes)
Random Sampling	9	1	Timed out. Impossible configurations sampled.
Random Neighbourhood Sampling	10	1	Timed out. Possible configurations were sampled.

Table 1: Comparison with other methods.

VII. ACKNOWLEDGEMENTS

I am very grateful to Professor Dmitry Berenson for giving me the opportunity to work and supervise on a project where I could use the valuable AI knowledge gained through his lectures and also for allowing me to work at the ARC lab with the Baxter robot. I would also like to extend my gratitude to the members of the ARC lab who helped with their suggestions during the lab meetings/discussions.

VIII. REFERENCES

[1] Michel, Scheurer, Kuffner, Vahrenkamp and Dillmann. Planning for robust execution of humanoid motions using future perceptive capability. Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, San Diego, CA, USA, Oct 29 - Nov 2, 2007.

[2] Tarabanis, Tsai and Allen. Automated sensor planning for robot vision tasks. Proceedings of the 1991 IEEE International Conference On Robotics and Automation, Sacramento, CA, USA, April, 1991.

[3] Cheng and Tsai. Visibility Optimization using Variational Approaches. Comm. Math. Sci., Vol. 3, No. 3, pp 425-451, 2005.

[4] Foissotte, Stasse, Escande and Kheddar. A Next-Best-View Algorithm for Autonomous 3D Object Modeling by a Humanoid Robot. 2008 8th IEEE-RAS International Conference on Humanoid Robots December 1 - 3, 2008, Daejeon, Korea.

[5] Landa, Galkowski, Huang, Joshi, Lee, Leung, Malla, Treanor, Voroninski, Bertozzi and Tsai. Robotic Path Planning and Visibility with Limited Sensor Data.

[6] Pito. "A sensor based solution to the next best view problem," in IAPR Int. Conf. Pattern Recognition, ICPR'96, Vienna, Austria, Aug. 1996, pp. 941–945.

[7] LaValle, Gonz´alez-Ba˜nos, Becker and Latombe. "Motion strategies for maintaining visibility of a moving target," in Proc. IEEE Int'l Conf. on Robotics and Automation (ICRA'97), 1997, pp. 731–736.

[8] Morales, Asfour, Azad, Knoop and Dillmann. "Integrated grasp planning and visual object localization for a humanoid robot with five-fingered hands," in Proc. IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems (IROS'06), Beijing, China, October 2006.

[9] Siciliano, Sciavicco, Villani and Oriolo. Robotics: Modelling, Planning and Control book. Springer.

[10] Russell and Norvig. Artificial Intelligence: A Modern Approach book, 3rd edition. December 13, 1994.

[11] Geometry Algorithms SDK, http://geomatalgorithms.com