



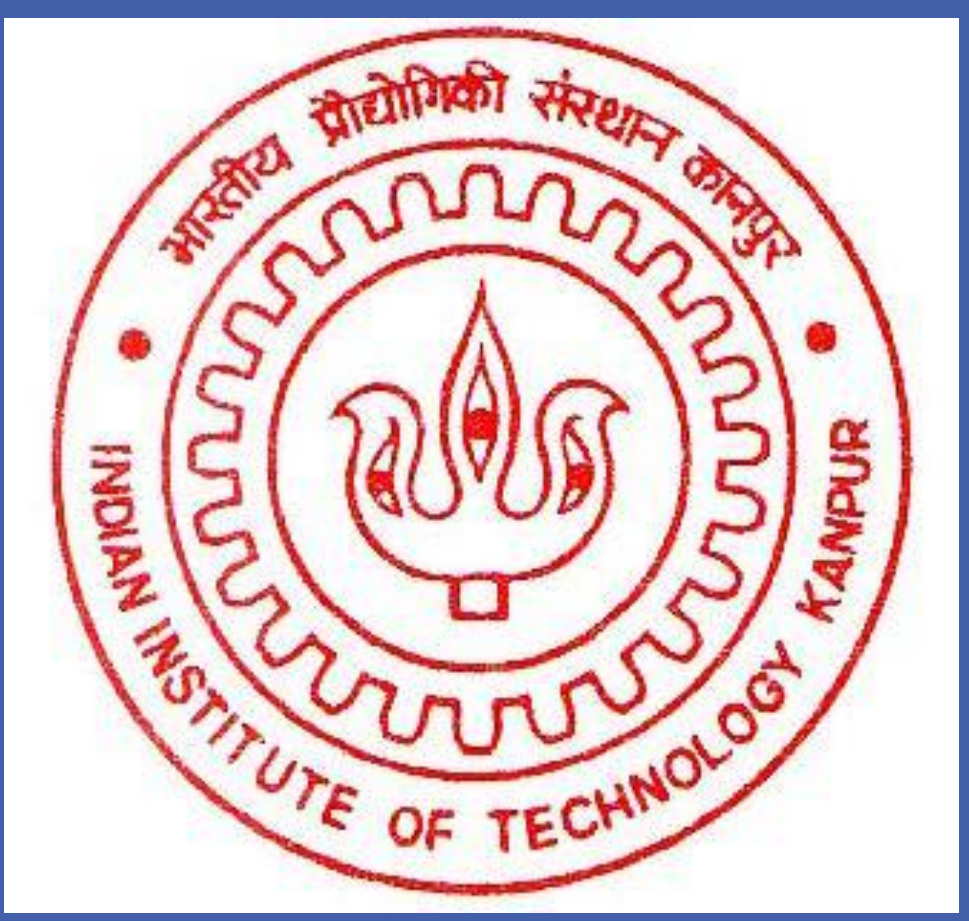
Autonomous Navigation of Nao using Kinect

Harshad Sawhney, Samyak Daga

{harshads,samyakd} @iitk.ac.in

Mentor: Prof. Amitabha Mukerjee

amit@cse.iitk.ac.in



ABSTRACT

The navigation of humanoid robots is a non-trivial task with applications ranging from assisting humans in their daily chores to manning space missions to disaster recovery operations. We present a depth based sensing approach for autonomous navigation of humanoid Nao in possibly cluttered environments. The objective is to reach a user desired goal position from a source position using the data of depth based sensor for visual servoing.

The localization of the robot is done via Monte Carlo technique which uses the RGBD data of Kinect sensor. The cluster of robot is extracted from the environment followed by KLD (Kullback-Leibler distance) sampling to accurately track the robot in real time. The motion control of Nao relies on the feedback data obtained from the depth sensor to correct its path which could possibly deviate due to noises in odometry and sensing information.

INTRODUCTION

- Autonomous navigations of humanoid robots has been a subject of widespread and lively research over the last years. The problem is highly challenging and non-trivial, with heavy potential impact in real world applications and usage.
- Inaccurate foot odometry, noisy overboard sensor information caused due to the swaying motion of the humanoid, limitation in mechanical movements and joints, and the constraints on the degrees of freedom are some of the problems associated with the navigation of the humanoid. These issues lead to the deviation of the humanoid from its expected path, resulting in not reaching the desired goal.
- In our project, we use the Kinect data sensor to track and detect the humanoid. Xbox Kinect™ is a low-cost sensor equipped with an infrared laser which provides a point cloud of the environment and gives the RGBD data of each such point. The point cloud is used to track the humanoid in real time.
- The humanoid that we employ for testing our approach is Aldebaran Nao. It is a biped robot with about 25 degrees of freedom, and a human look alike.

RELATED WORK

- Autonomous navigation has been a problem which has seen a lot of different methods. Laser sensors, 2D cameras, depth cameras have all been employed with significant variations
- Corridor navigation through the use of 2D cameras has been used by Oriollo et al. with the concept of vanishing point
- In [1] an external Kinect is used to track the 6DOF pose of Nao, which is then used to navigate to a goal position – serving as an assistant in home-like environments
- Kinect has also been mounted on Nao itself by Maier et al. which is then used to detect obstacles and plan the path by the construction of a heightmap
- KLD sampling technique proposed in [2] was used in [3], along with Monte Carlo localization, for tracking objects in a point cloud
- Monte Carlo localization was proposed in [4]

METHODOLOGY

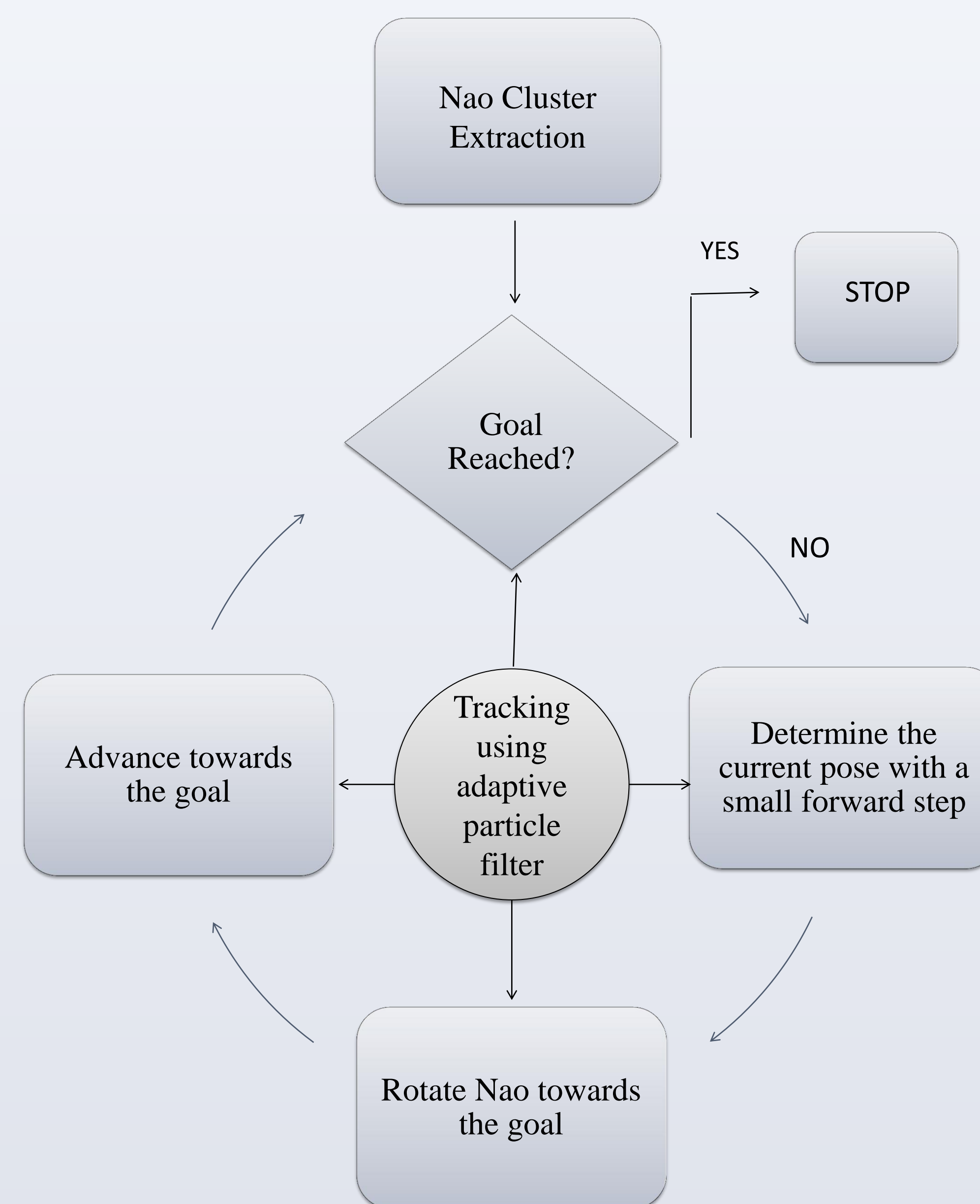
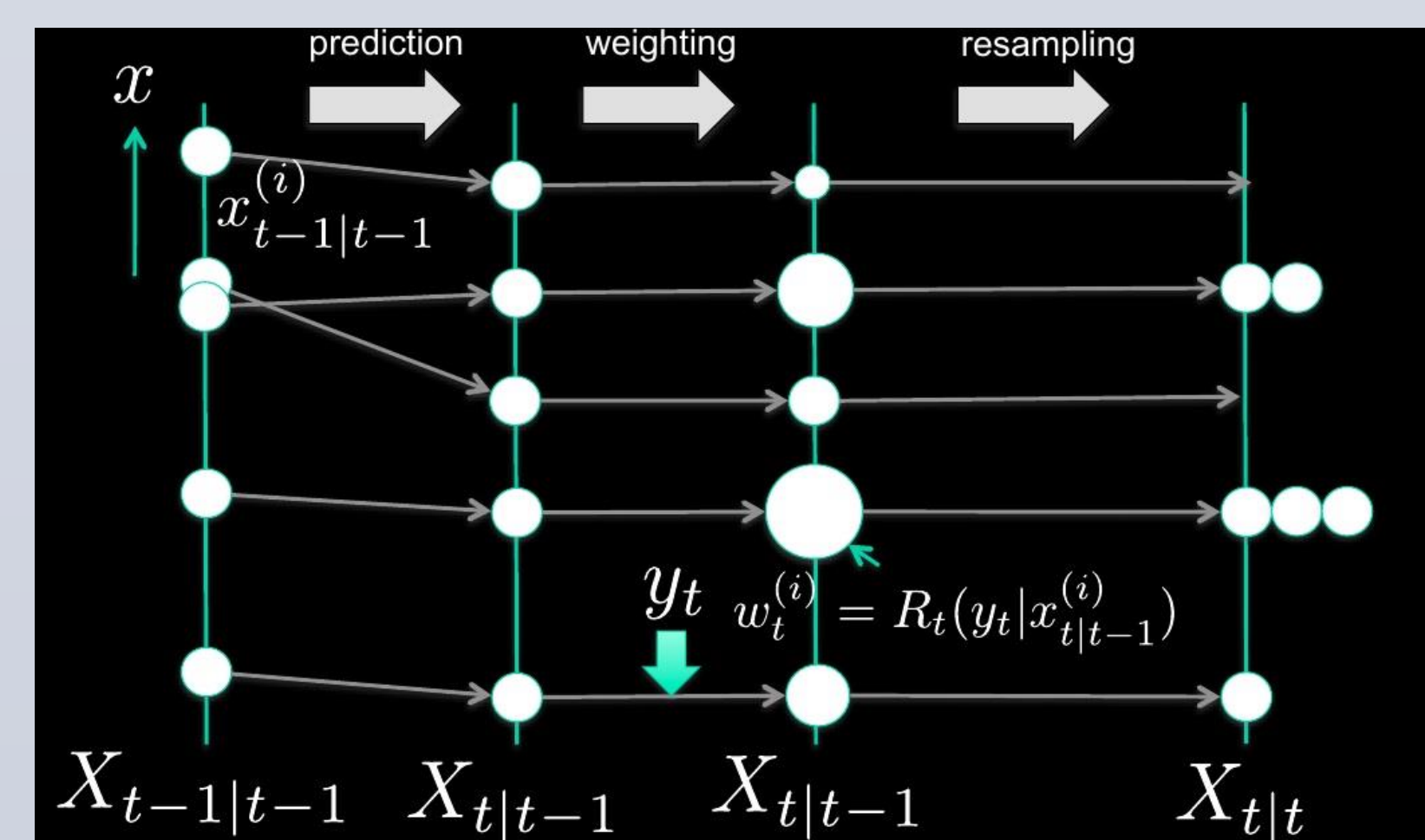


Figure 1. Flowchart depicting the approach for navigation

- A cluster representing Nao is obtained from the point cloud obtained from Kinect using Euclidean segmentation
- Euclidean segmentation - The motivation behind it is to reduce the processing time for point cloud P by dividing the unorganized data using a clustering method based on Euclidean distance metric. The algorithm is as follows:

- create a Kd-tree representation for the input point cloud dataset P;
- set up an empty list of clusters C, and a queue of the points that need to be checked Q;
- then for every point $p_i \in P$, perform the following steps:
 - add p_i to the current queue Q;
 - for every point $p_i \in Q$ do:
 - search for the set P_i^k of point neighbors of p_i in a sphere with radius $r < d_{th}$;
 - for every neighbor $p_j \in P_i^k$ check if the point has already been processed, and if not add it to Q;
 - when the list of all points in Q has been processed, add Q to the list of clusters C, and reset Q to an empty list
- the algorithm terminates when all points $p_i \in P$ have been processed and are now part of the list of point clusters C



METHODOLOGY

Particle Filter through KLD (Kullback Leibler Distance) sampling –

- This technique adapts the number of samples in particle filter or Monte Carlo methods based on the KL distance (ϵ) between maximum likelihood estimate (p) and true probability distribution (q) $K(p, q) = \sum_x p(x) \log \frac{p(x)}{q(x)}$
- The above distance is not a metric as it is not symmetric and does not obey triangle property.
- The number of samples is inversely proportional to error bound ϵ .
- It chooses small samples if density is on small part of state space and large samples if there is high uncertainty in the sample space.
- It selects large sample sets for global localization and small sample sets for position tracking.
- Thus, it effectively decreases the computation time of particle filter which is exponential in state dimensions.

- Nao Motion Control – The method of visual servoing provides a closed loop feedback for correcting the errors in odometry which makes the robot move to the goal position accurately.

(x_1, y_1, z_1) : Nao initial tracked coordinates

(x_2, y_2, z_2) : Nao tracked coordinates after it moves ϵ distance forward

(x_g, y_g) : Goal Coordinates

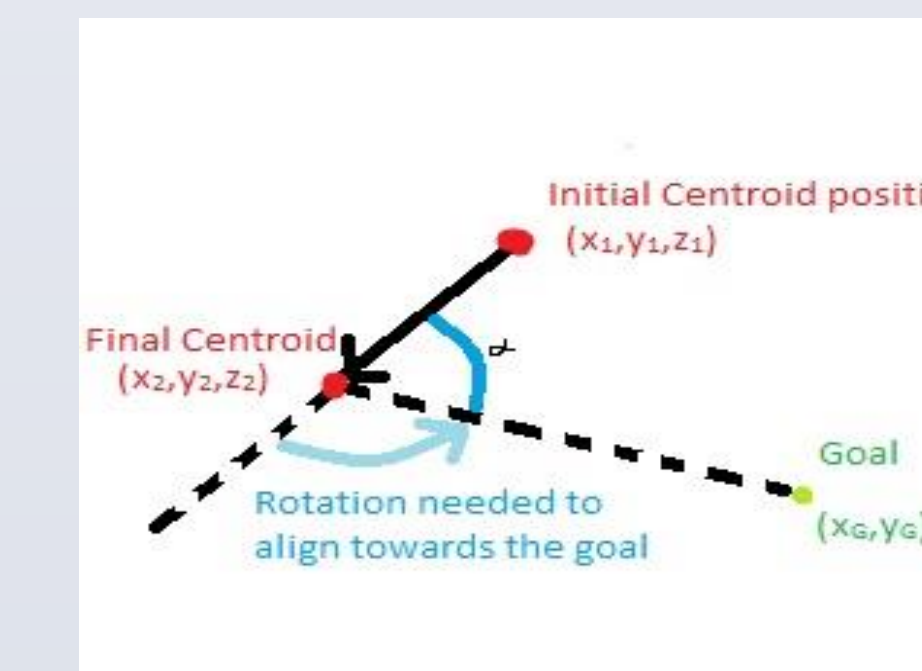
$$\alpha_1 = \frac{z_1 - z_2}{x_1 - x_2} \quad \alpha_2 = \frac{y_g - z_2}{x_g - x_2}$$

$$\alpha = \alpha_2 - \alpha_1$$

If $\alpha > 0$ then $\alpha = \alpha - \pi$

else $\alpha = \alpha + \pi$ end;

Rotate Nao by α and then move fixed distance forward towards the goal
Keep on executing the above steps till the robot reached target position



RESULTS

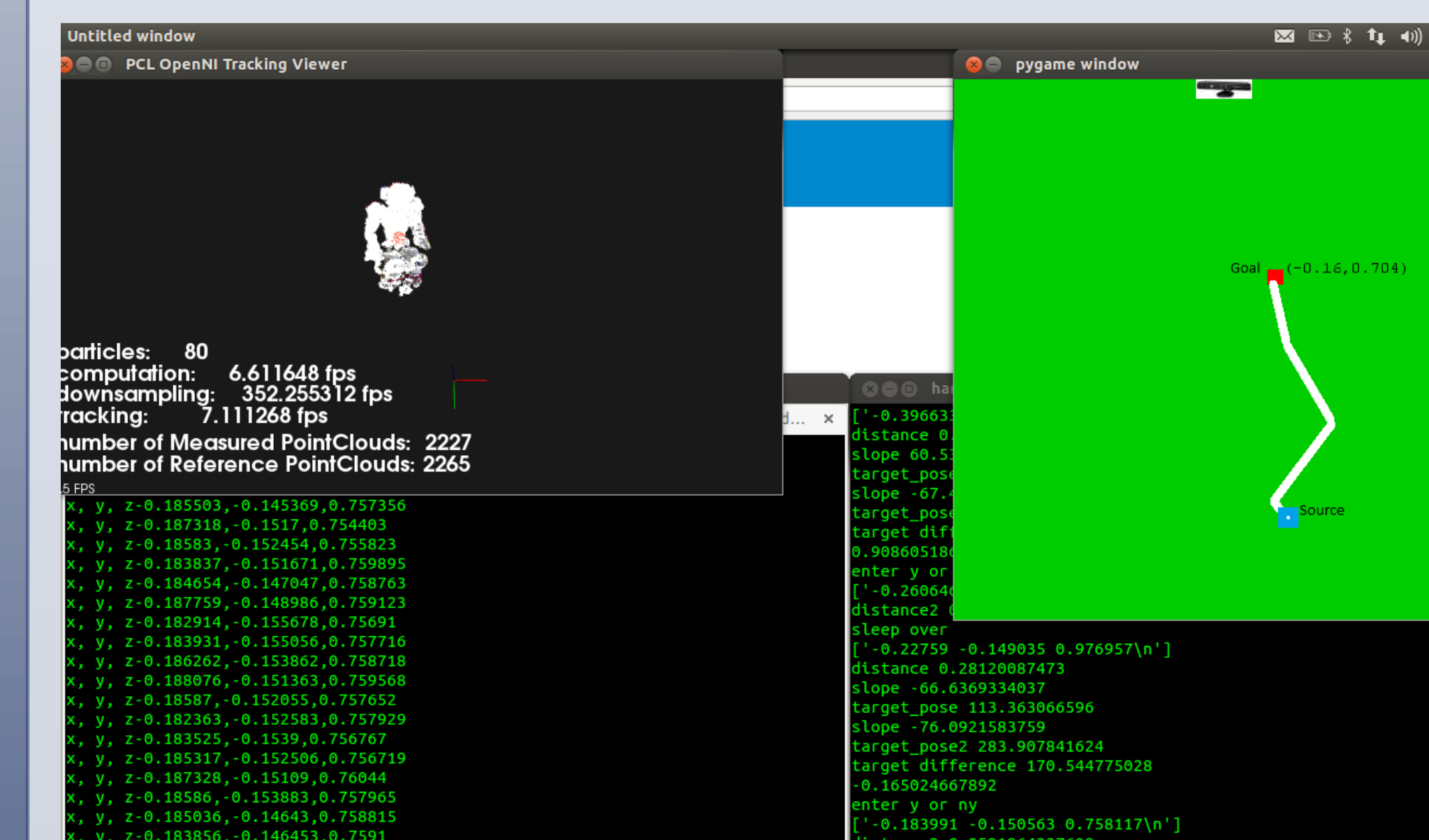


Figure 3a. This side shows the real time tracking being done using the Kinect. The tracked Nao cluster (shown in blue) and the corresponding centroid of the cluster at different points of time are shown.

Figure 3b. This side depicts the screen for giving the goal position (shown in red). The actual path taken by Nao is shown in white from the source (blue).

RESULTS

- The method was tested with 0.08 meters displacement for the determination of the pose, and 0.3 metres towards the goal after rotation



Figure 2a. Cluster of Nao extracted using Euclidean segmentation



Figure 2b. The actual image of Nao

REFERENCES

- [1] Cervera, Enric, Amine Abou Moughlba, and Philippe Martinet. "Localization and Navigation of an Assistive Humanoid Robot in a Smart Environment." IROS Workshop '12
- [2] D. Fox. "Adapting the sample size in particle filters through KLD-sampling". IJRR, 22(12):985–1004, 2003.
- [3] Thrun, Sebastian, et al. "Robust Monte Carlo localization for mobile robots." Artificial intelligence 128.1 (2001): 99-141.
- [4] Ueda, R. "Tracking 3d objects with point cloud library." (2012)
- [5] <http://pointcloud.org>

IMPROVEMENTS AND FUTURE WORK

- The pose of the robot can be directly determined by the present state of the cluster extracted through aligning two point clouds
- Obstacle detection can be employed and obstacles can be detected using the extraction of clusters from the Kinect sensor
- Obstacle avoidance algorithms can be also then be employed for finding optimal paths

ACKNOWLEDGEMENTS

Like everybody else, we owe much more than we can repay. We would like to show our sincere gratitude towards Prof. Amitabha Mukerjee, Department of Computer Science and Engineering, IIT-Kanpur for his valuable guidance throughout the project and to keep us motivated. We would also like to thank the tutors, Mr. MS Ram and Ms. Sunakshi Gupta, of the course for their support whenever we required. Finally, we thank our classmates and friends for generating that competitive spirit within us.