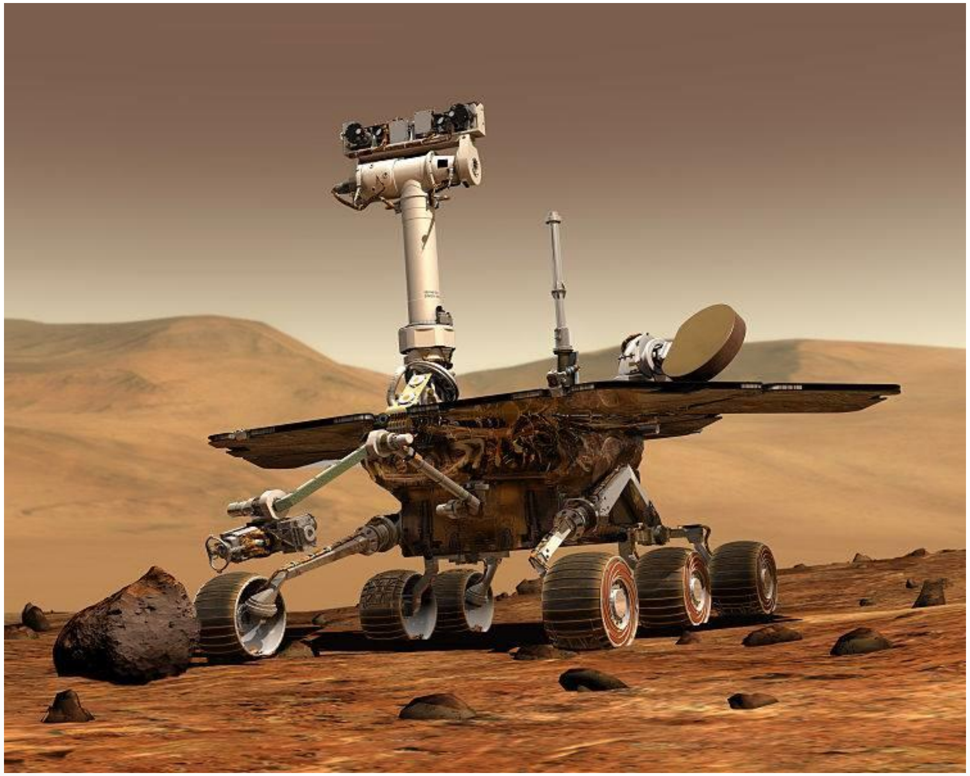
Autonomous Navigation of Humanoid Using Kinect

Harshad Sawhney 11297
Samyak Daga 11633

Motivation

- Humanoid Space Missions



http://www.buquad.com

Motivation

- Disaster Recovery Operations

http://www.designnews.com

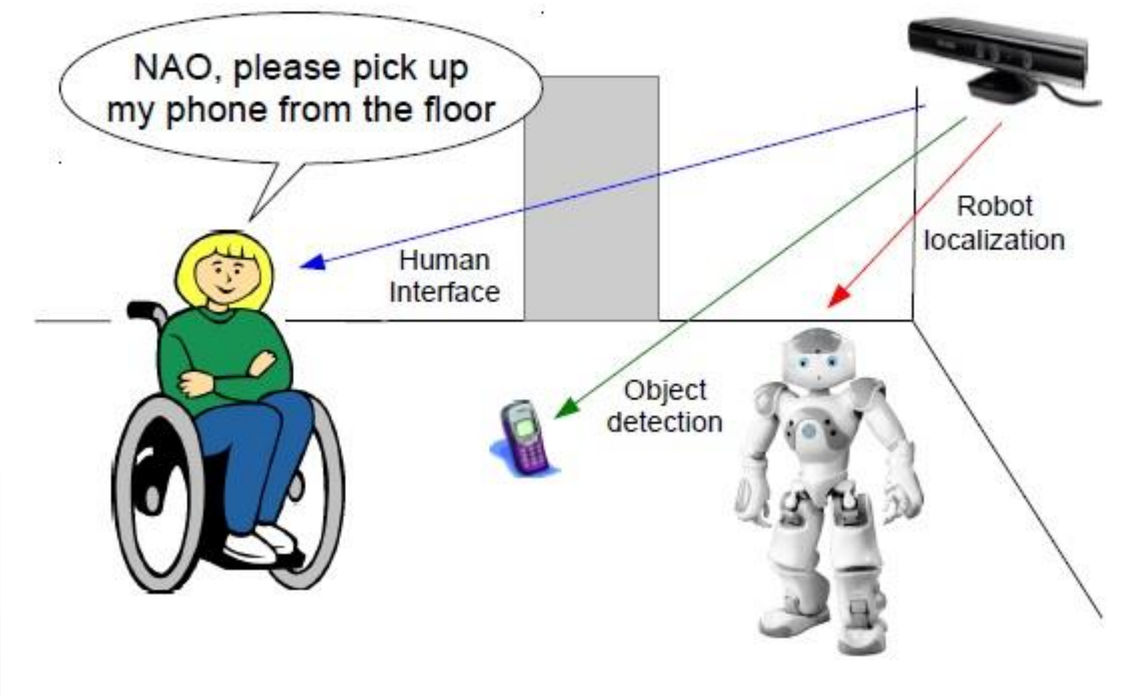
Motivation

- Helping in daily chores

http://www.ccd-design.blogspot.in/2011/05/robots-helping-us-in-old-age.html

And Finally…

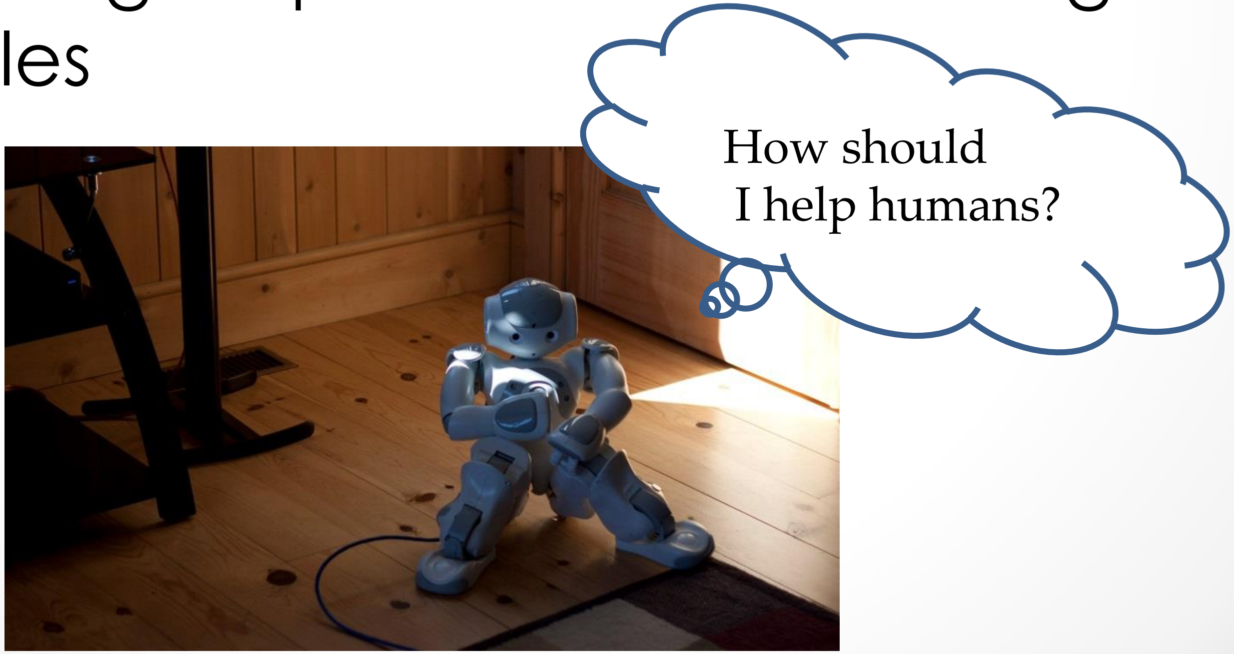
- Helping disabled at home



"Localization and Navigation of an Assistive Humanoid Robot in a Smart Environment", Cervera et al., 2012

Objective

- To achieve localization and navigation of humanoid Nao
- The robot is to be moved from the source to a desired goal position while avoiding obstacles



http://geekwithtoys.ca/category/nao/

Non-Trivial?

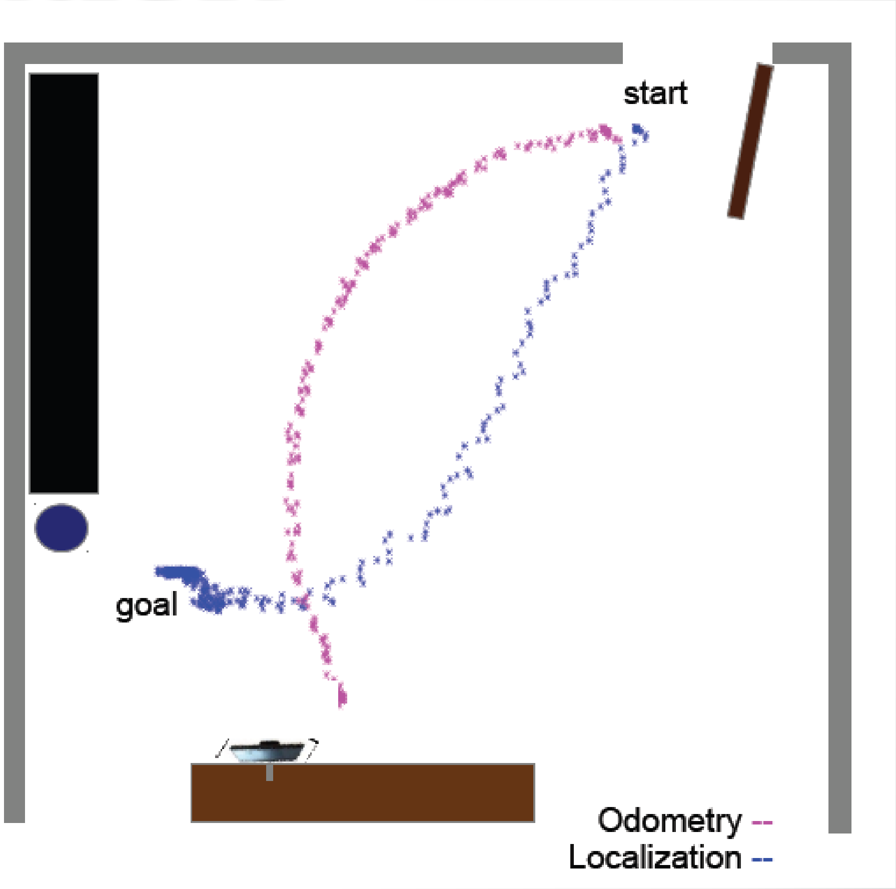
- Inaccurate footstep odometry
- Noisy onboard sensor observations
- Limitations in mechanical movements
- Constraints on degrees of freedom

Overview

- Localization and Pose estimation
- Obstacle Detection
- Path and footstep planning
- Closed-loop feedback

Localization and Pose Estimation

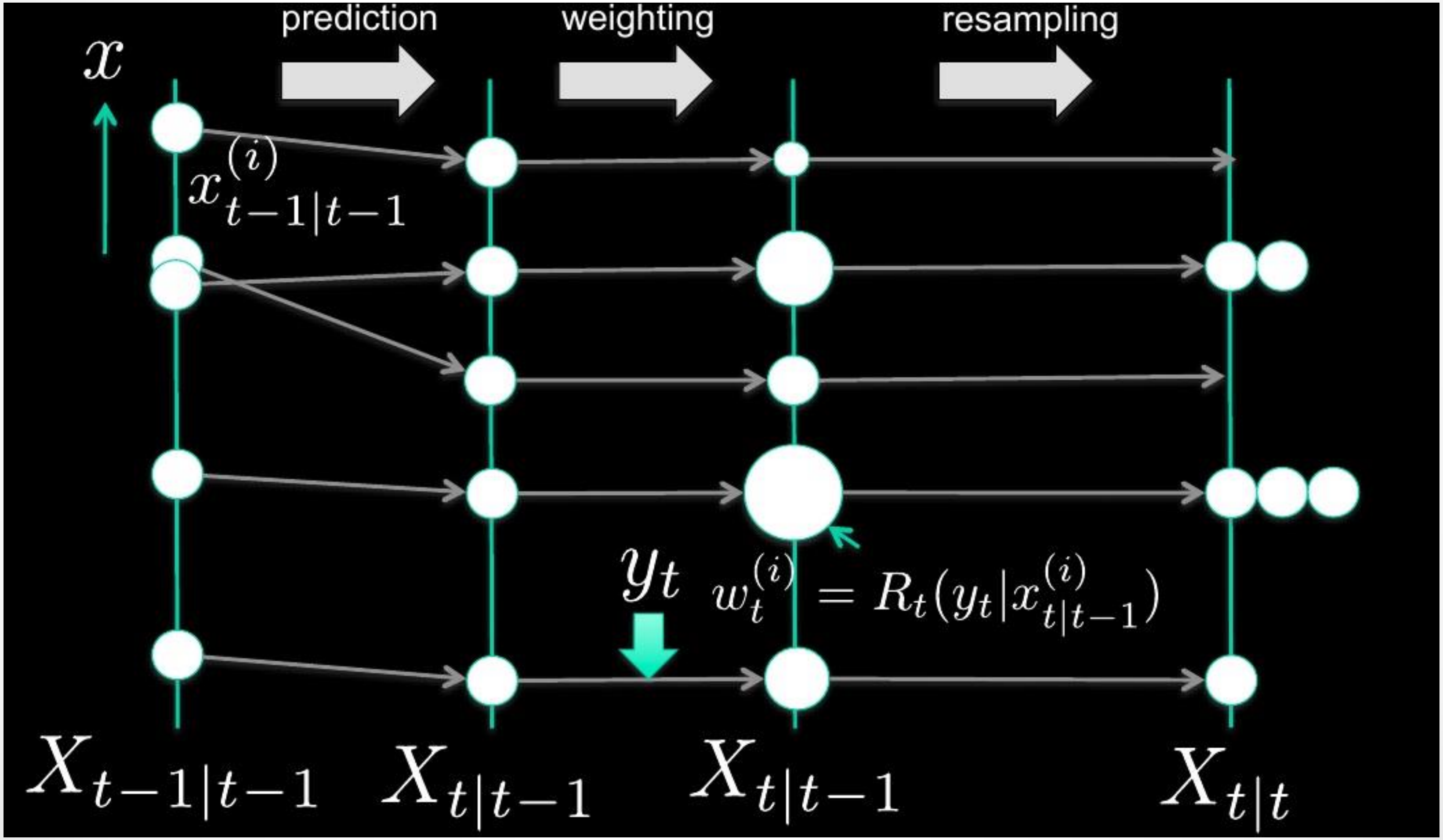
- Kinect as external sensor
- The point cloud determines the pose of the humanoid
- A rigid model of the torso and the head parts is used for 3D tracking



PCL Cloud Tracking

- ‘Tracking’ module in PCL provides algorithmic base for the estimation of 3D object
- Monte Carlo localization technique
- Calculation of the likelihood is done by combining weighted metrics like Cartesian data, colours and surface normals

Monte Carlo Localization



Borrowed from R.Ueda

Monte Carlo Localization

$$w^{(i)} = \sum_j L_{distance}(p_j, q_j) L_{color}(p_j, q_j)$$

$$L_{distance}(p_j, q_j) = \frac{1}{1 + \alpha |p_j - q_j|^2}$$

$$L_{color}(p_j, q_j) = \frac{1}{1 + \beta |p_{jcolor} - q_{jcolor}|^2}$$

p_j : a point of the hypothesis pointcloud
q_j : the nearest point of the input pointcloud to p_j

Borrowed from R.Ueda

Monte Carlo Localization

- The algorithm uses a particle filter to represent the distribution of likely states, with each particle representing a possible state
- The assumption is that the current state depends only on the previous state
- Woks if the environment is static
- At the start, the particles are uniformly distributed over the configuration space

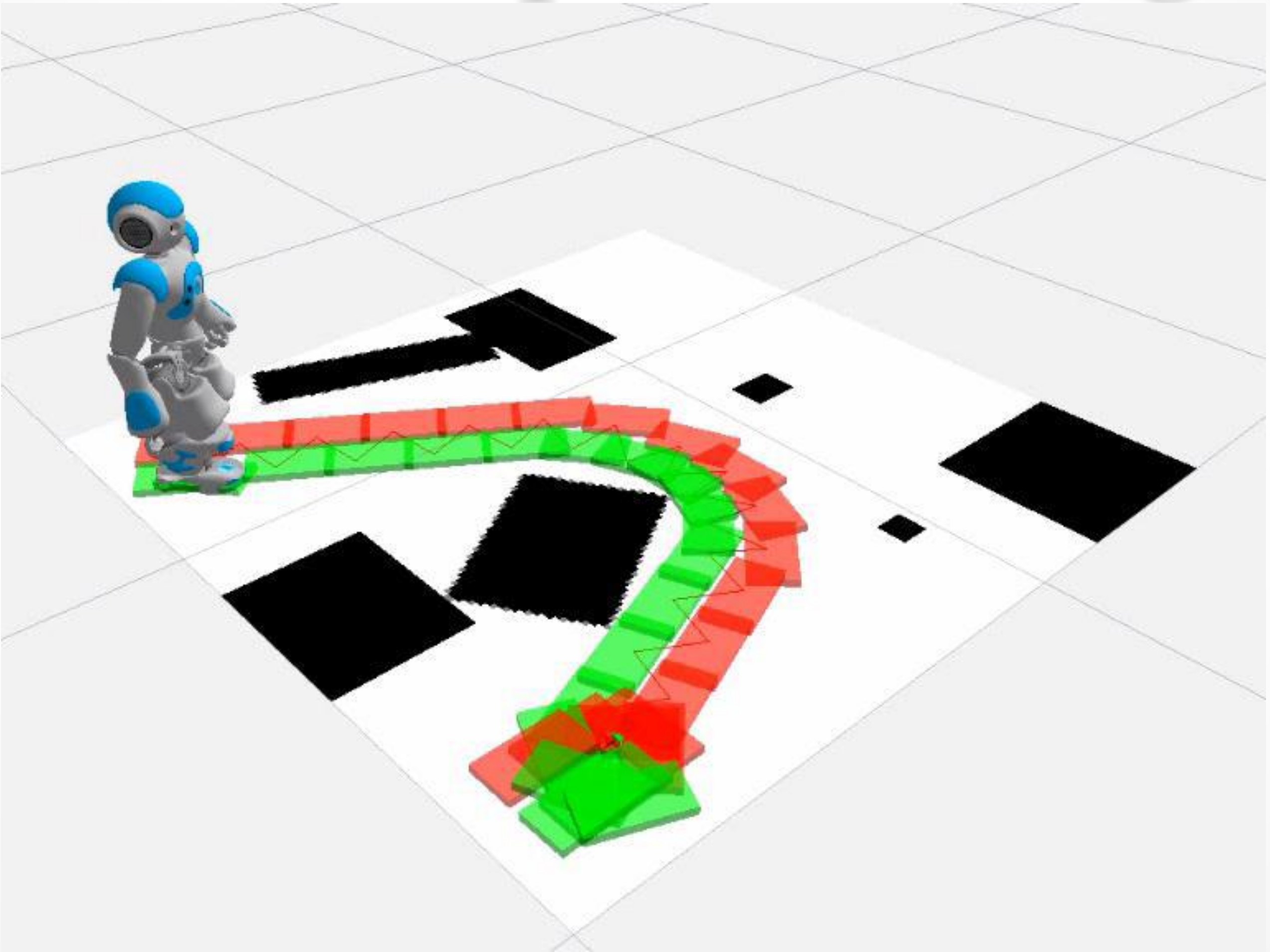
Monte Carlo Localization

- At every time *t*, the algorithm takes as input the previous state $X_{t-1}=\{x_{t-1}^{[1]}, x_{t-1}^{[2]}, \ldots, x_{t-1}^{[M]}\}$
- Motion Update : Particles are shifted according to the given actuation command u_t
- Sensor Update : Data from sensors z_t which is used to reweight the particles
- Resampling : New set of M particles are drawn using the new weighted probability distribution

Obstacle Detection

- Static obstacles
- Divide the environment into a grid
- Occupancy Grid method
- A cell is marked occupied, free or unknown
- Differentiate between shallow obstacles and obstacles with large clearance

Footstep Planning



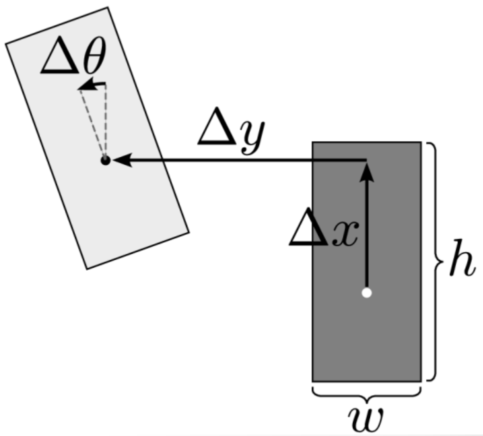
As part of the Robot Operating System (ROS) at
http://www.ros.org/wiki/footstep_planner

Footstep Planning

- Distance map- Euclidean distance to closest obstacle
- State s=(x, y, θ)
- Footstep action a= (Δx, Δy, Δ θ)
- Fixed set of Footstep actions :

  $F=\{a_1, \ldots, a_n\}$
- Successor state s' = t(s, a)
- Transition costs:

  $$C(s, s') = |(\Delta x, \Delta y)| + k + d(s')$$



Borrowed from http://www.ais.uni-bonn.de/humanoidsoccer/ws12/slides/HSR12_Slides_Hornung.pdf

Footstep Planning

- From a given stance state, all possible states which do not collide with obstacles are expanded
- Search algorithms can be used for the path planning from the start to the goal among the above expanded set
- A*, ARA*, R* etc. have been implemented

Initial Attempts

Why Kinect not mounted on Nao?

- Heavy payload of Kinect
- Center of Gravity shifts upwards leading to unstable motion
- Motion leads to error in Point Cloud detection

References

- Cervera, Enric, Amine Abou Moughlbay, and Philippe Martinet. "Localization and Navigation of an Assistive Humanoid Robot in a Smart Environment." IROS Workshop '12
- Ueda, R. "Tracking 3d objects with point cloud library."(2012)
- Thrun, Sebastian, et al. "Robust Monte Carlo localization for mobile robots."*Artificial intelligence* 128.1 (2001): 99-141.
- Hornung, Armin, et al. "Anytime search-based footstep planning with suboptimality bounds." *Humanoid Robots (Humanoids), 2012 12th IEEE-RAS International Conference on*. IEEE, 2012.
- Garimort, Johannes, and Armin Hornung. "Humanoid navigation with dynamic footstep plans." *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011.

Thank You!

Any Questions?