Localization using an entropy reduction/information filter approach

Problem Description:-

Object/target localization is a classic problem in robotics. Generally it involves additional sub-problems which need to be solved to achieve the goal i.e. distinction from possibly other objects that might be present in the environment and/or simultaneously localizing the position of the robot correctly.

For the current project, a robot arm needs to be moved to achieve the goal of correctly localizing the position of the object/target such that the robot arm can reach it effectively and efficiently. The robot arm needs to decide a sequence of positions which it must move to in order to gain maximum information with respect to the environment and object/target which will allow it to localize the object/target.

Background:-

Localization using filters is pretty common in robotics. The type of filter used to solve the localization problem is decided by the constraints/requirements of the problem and/or by the limitations of the filter i.e. a histogram filter would be more suited to certain types of problems in which the state space is pretty small. However, the kalman filter can only be used for a certain class of localization problems. Specifically those in which the movement function is linear and the uncertainties are normally distributed. Particle filters are used for their simplicity.

Kalman filters have the limitation that they can only represent unimodal distributions whereas histogram and particle filters can represent multimodal distributions which are found in most real world scenarios.

Entropy is commonly used to represent uncertainty in a system/environment or state space. The lower the entropy in the environment, the more certain the belief space representation.

The current project borrows ideas from both histogram and particle filters as well uses the concept of entropy to achieve its goal of localizing the object/target.

Approach:-

The environment is simulated using a robot arm which needs to move to a sequence of positions decided by the particle filter/histogram filter logic. Entropy is used as the metric/criteria which decide which position the robot arm moves to next from its current position. The robot arm moves to a location which results in the maximum reduction in entropy of the belief space. Reduced entropy is synonymous to increased certainty of the belief space i.e. as the robot arm moves and takes observations; it becomes more and more certain about the location of the object/target.

The environment also has a set of particles which represent states. These particles are sampled randomly from the region of space in which the robot arm believes the object/target is located in.

These particles represent the belief space. Initially all states are equiprobable but this belief probability distribution changes over time as the robot arm moves.

To get the next location the robot arm must move to, a set of particles/states are picked from the initial set of particles in accordance to their probability weight distributions. A set of possible robot arm positions are picked and for each of the candidate robot arm positions, the position which has the least entropy is picked as the next location the robot arm must move to.

The process update step consists of moving the robot arm to the next calculated arm position. The measurement update step updates the probabilities based on the observations for each of the particles and the actual observation. Observations in each case are a 2-tuple {del_x/del_z, del_y/del_z} where del_x, del_y and del_z represent the transformed distances with respect to the object/target.

Resampling is not done in the above proposed method rather just the belief state probability weights are updated in each measurement update step. This is to allow the scenario of not losing particles i.e. states and to take care of the cases in the object is not represent in the camera range of observation and multiple object/target cases.

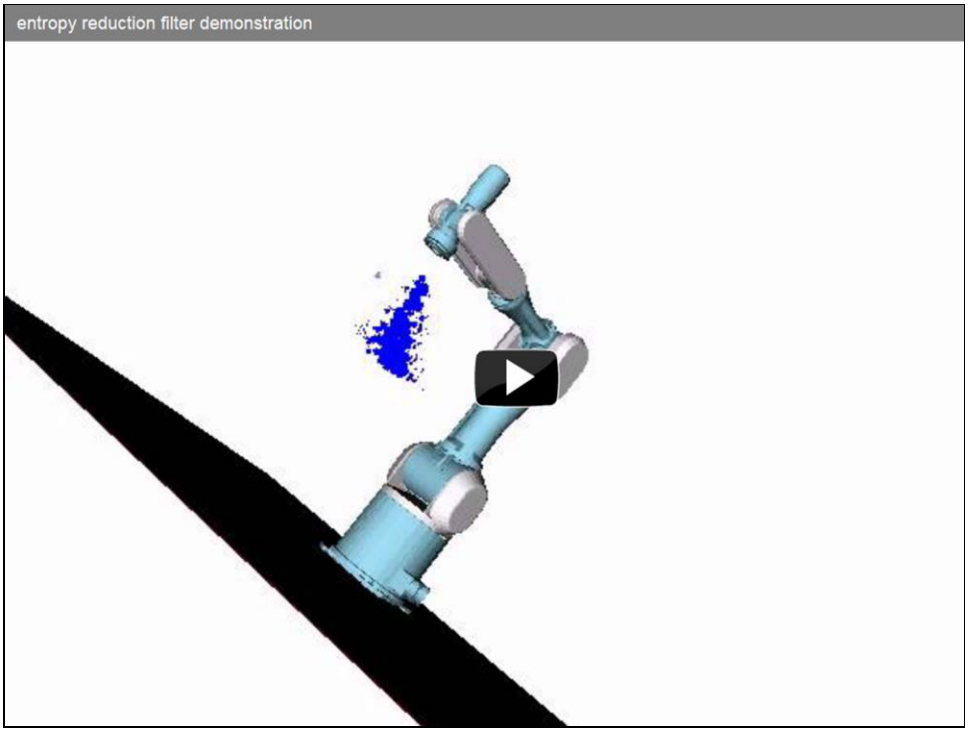
Pseudo-code for the proposed algorithm:-

```
1)    if index >= 0 and index <= actions_max: //repeat
2)        if index == 0: // initially
3)            move_to_start_position() // move to start position
4)            initialize_particles() // initialize particles representing states
5)            get_next_camera_position() // get next position robot arm
6)        else:         // subsequently
7)            process_update() // move robot arm to next position selected
8)            measurement_update() //update belief probabilities of the particles
9)            get_next_camera_position()// get next position robot arm
10)       index += 1
```

Assumptions:-

In the simulation, it is assumed that robot arm position is always known i.e. it is not a case of a SLAM problem.

Simulations:-



Note:-

- To play the above simulation, double click on the above embedded object. A maximized window will appear as the slideshow plays a Youtube! video. To play the video, go to the bottom left corner of the maximized window and click on the play button.
- The green dot in the video represents the actual object.
- The blue points represent the predictions i.e. where the robot arm thinks the object is.
- As the robot arm moves and makes observations, it becomes more certain about the object location. The blue dots become more concentrated in the areas the robot thinks the object is located in.
- The code base for the above simulation is available here [1]
- The above video can also be accessed here [2]

Conclusion:-

The proposed methodology is able to accurately localize the object as evident from the simulations. Thus the proposed method is able to achieve its goal. The resampling step is absent which allows it to be robust and possibly take care of multi object/target cases and/or the scenario in which the object is not present in the visibility range of the camera embedded on the robot arm. An interesting comparison could have been one in which the state space would be equally divided among the particles/states i.e. the states were equidistant from each other.

The above methodology can be extended to SLAM and/or the cases mentioned above. This was not done as part of the current project and can be considered future work.

<u>References</u>:-

[1] http://dl.dropbox.com/u/77702451/code%20base.zip

[2] http://www.youtube.com/watch?v=Pqn-tMojdjk