Autonomous Robot Retrieval System

Samantha Ahern, Dr Jenny Carter, Paul Wilson

Centre for Computational Intelligence, Faculty of Technology,

De Montfort University, Leicester

Abstract

Mobile robots are increasingly being deployed in environments hazardous to humans. However, many of these robots require remote control operation or are tethered, requiring the human operator to remain within a potentially hazardous radius of the area of operation. To resolve this issue an Autonomous Robot Retrieval System (ARRS) utilising Open RatSLAM based on the Lego NXT 2.0 robotics platform is proposed but could not be implemented due to memory limitations of the hardware. An occupancy grid implementation was explored but conflicts between competing tasks prevented its successful operation. In addition, the Queensland University of Technology MatLab RatSLAM implementation, the precursor to Open RatSLAM, was investigated. Updates were made to the implementation to enable the reading of video les using updated functions, and its performance with regard to accuracy of mapping and running-time was assessed. The accuracy of mapping was found to vary greatly dependent on the nature of the video footage used and running-time of the program was found to be 10 times that of the video le duration. Both of these factors suggest that the MatLab RatSLAM implementation is not suitable for real-time SLAM operations.

Literature Review findings and ARRS design

Control Architecture

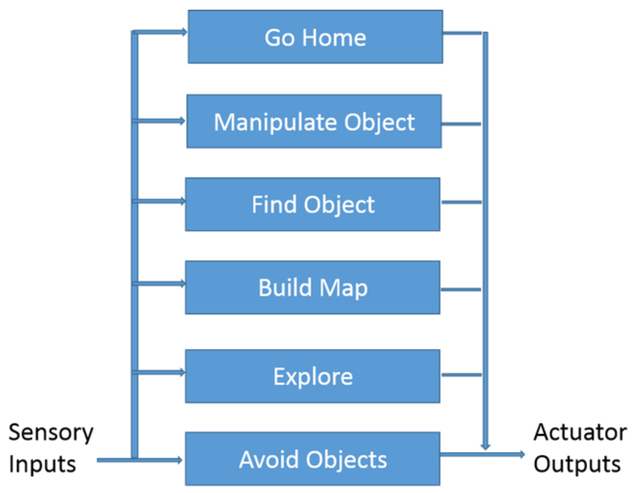
- Behaviour based and subsumptive

Navigation, Planning and Object avoidance

- SLAM commonly implemented on larger and more powerful robots
- Open RatSLAM implemented for Lego NXT in Robot C

Object manipulation

- Restricted to simple grasping motion



Proposed behaviour architecture for the ARRS

ARRS Development Issues

Stability of revised NXT construction

- Positioning of claw
- Ground clearance
- Positioning of sensors

Robot C memory limitations

- 15kb for program variables
- Open RatSLAM implementation exceeded this limit

NXC has 32kb of working memory available

- Decision made to translate Robot C implementation to NXC
- Not all functions had a like-for-like mapping
- Data structures were complex and required re-factoring
- Issues started to emerge with the NXT controller brick

NXC working memory exceeded

- Translation of Robot C Open RatSLAM to NXC not possible on Lego NXT 2.0 - abandoned

MatLab RatSLAM - Development

Queensland University of Technology code base

- Developed in 2008

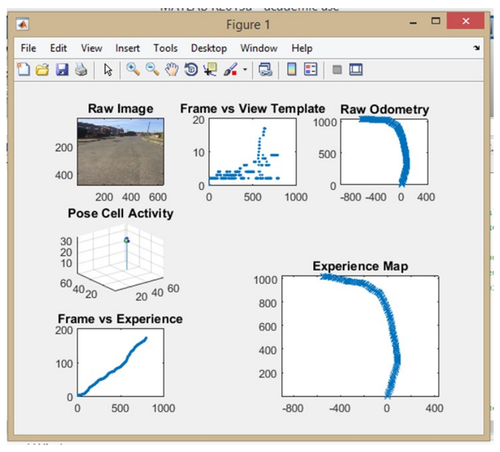
Issues:

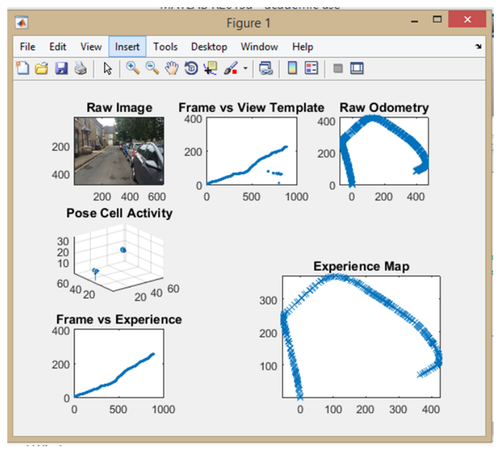
- Use of deprecated function aviread in rs_main.m
- Test file stlucia_testloop.avi created using deprecated codec cvid

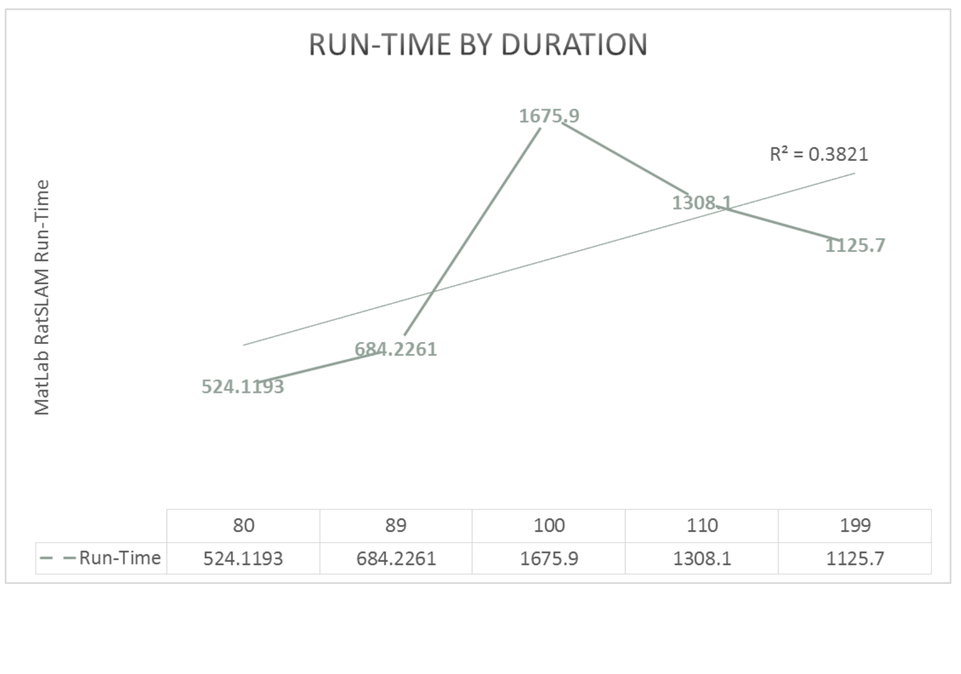
Amendments:

- aviread replaced with VideoReader object function readFrame
- stlucia_testloop.avi converted to stlucia_testloop.mp4
- Amended rs_main.m added to Queensland University of Technology GitHub repository

MatLab RatSLAM - Results







	80	89	100	110	199
Run-Time	524.1193	684.2261	1675.9	1308.1	1125.7

Future Work

NXC Development

- Build map of environment as an occupancy grid
- Compare route traversed with MatLab RatSLAM generated map

Translate Robot C Open RatSLAM implementation to Python

- Open source so accessible to all
- Can be used on low-cost computing platforms