Mini Survey Paper
(Robotic Mapping)

Ryan Hamor
CPRE 583
September 2011

Introduction

The goal of this survey paper is to examine the field of robotic mapping and the use of FPGAs in various implementations. This paper provides a brief overview of robotic mapping. Some of the various problems in this field are explained along with implementations using FPGAs are explored.

Robotic Mapping Overview

The start of robotic mapping can be traced back to the early 1980s and 1990s. Two types of mapping techniques were used; metric maps and topological maps. Metric maps seek to capture the geometry of the environment in which the robot operates. Topological maps try to capture and connect different locations in the robot's world. Another aspect of robotic mapping at this time focused on world centric vs. robot centric views of mapping. A world centric map contains information about objects in a global space. A robot centric mapping view contains objects that are represented as sensor measurements observed based on the robots location.

Since the late 1990s a vast majority of the robot mapping implementations have been based off probabilistic techniques. These techniques were especially of interest since they allow for simultaneous localization and mapping (SLAM). With this approach the robot can move around its environment while mapping it out. Three popular approaches to probabilistic mapping include, Kalman Filters, expectation maximization and classification of objects.[1]

Problems Associated With Robotic Mapping

There are five main problems related to robotic mapping which are listed below:

1. The robot uses sensors to observe the environment. The list of sensors includes; odometers, GPS, cameras, sonar, infrared, etc. Each type of sensor has varying ranges and measurement errors. In conjunction with the sensors the control command is also often used to help determine location and pose. All of these measurements are statistically dependant and must be dealt with.

2. As a robot maps out the environment each measurement point that is added to the map increases the dimension of that map. This makes working with maps a very high dimensional problem.

3. As the robot takes new measurements of its environment the new data must be looked at to see if it belongs to an existing object or is some new object. This is called the correspondence problem.

4. Another issue faced in robotic mapping is the fact that the environment the robot is present in is dynamic in nature. This means that the state of a door may go from open to close and the previously measured data must be remapped.

5. A final problem in robotic mapping is that of robotic exploration. A lot of the approaches to robotic mapping require computationally intense calculations to be performed, solving these equations in real time is required for a robot to move about the environment productively. [1]

Robotic Mapping Problem Solutions

Of the five major robotic mapping problems this section will take a closer look at three of them, correspondence, map dynamics, and robotic exploration. Specific solutions to these problems shall be examined in conjunction with FPGA based implementations.

Correspondence Problem

The first problem to be discussed is the correspondence problem. One approach to the correspondence problem is through multi-planar mapping. Using a multi-planar map can help place information from new measurements into an existing map by allowing the data to be compared against surrounding data in the vertical plane.

The implementation used in [6] centers around the need for mapping of a UAV (unmanned aerial vehicle) in an urban environment. The authors describe the use of a LADAR to make 2D measurements and create a single plane. Objects detected in the 2D scan appear as line features. Two additional scans are taken at different elevations to reconstruct the 3D plane. In the implementation an FPGA is used for data collection from the LADAR. The FPGA takes data from the LADAR sensor while controlling the servo motor movement that rotates the LADAR. The LADAR data and elevation information are then sent back to a host PC for further processing. This illustrates the use of an FPGA to create control sensors and create custom messaging packets.

In [2] a 3D image is captured using segmentation based on expectation maximization (EM) and maximization of posterior marginals (MPM). For every iteration of the EM algorithm the MPM algorithm must run seven times. The authors choose to implement the MPM algorithm into hardware to help speed up the process. A ping-pong type memory access is implemented in the FPGA to keep all of the memory access used to store intermediate results readily available. This is used to reduce overhead. After the seven MPM iterations the result is pushed back to external memory and a new MPM iteration set begins. The MPM algorithm must be applied to each pixel in an image and its six surrounding neighbors. To again increase the speed of processing the image 8 parallel instances of MPM algorithm are created. While this implementation was specifically targeted at medical imaging. The algorithm could easily be ported to work in making 3D maps of a robots environment.

Dynamic Environment Problem

As stated above another key problem in robotic mapping is that the robot is not present in a static world. Object detected at a specific time in the map can be changed or missing completely. One approach to handling dynamic situations is to use occupancy grids. Occupancy grids are 2D grids that implement the Bayes filter to populate binary information on whether a cell is occupied or empty. By learning models of objects represented as occupancy grids it can be determined if a know object has changed. Thus, dynamics can be taken into account.

The approach in [5] also uses occupancy grids. Here point clouds are computed from three types of data; stereo, structure, and motion. The point clouds are created using stereoscopic imaging. The structure and motion algorithm is computed using feature tracking, camera motion estimation and triangulation. The main goal is to separate points belonging to the static scene while in motion. The 3D image comes from the point cloud estimate and is used to create the occupancy grids. Sensor data from an IMU and GPS are then used in the structure and motion algorithm to determine motion in the observed scene. In the implementation of this system an FPGA is used in the stereo reconstruction algorithm. The remaining kernels are executed in an x86 type processor.

Robotic Exploration

An interesting area of robotics and mapping is that of exploration. An example application where robotic exploration is particularly useful is that of search and rescue. One type of exploration is frontier-exploration, in this type of exploration the robot seeks to enhance the map size by moving towards open space and unexplored space, the frontier so to speak [7].

Robotic exploration requires two main items, the ability to localize and map simultaneously and the ability to do this in real time. The first approach an implementation described by [4] is based on the notion of generalized Voronoi diagrams (GVD). The GVD algorithm determines the center of free space between two detected objects and has been used extensively in path planning. The authors' main goal is to implement the algorithm efficiently in an FPGA such that no external computational units are needed. The platform for testing is a mobile robotic platform that includes 8 sonar sensors placed at 45 degree points around the robot. The authors first approach the problem in the static environment and then scale it to a dynamic environment.

The article goes in great detail about the internal structure of the hardware implementation. The main use of parallelism in the design comes from the conversion of the sensor pulse width output into distance in cm. The use of pulse width to proportionally relate to distance in centimeters is nice since it does not require an analog to digital converter. Besides computing the sensor counts to centimeters the authors algorithm is mainly a sequential one. It is interesting to note that they implement everything on the FPGA from sensor inputs to the step motor control signals.

The goal of the work presented in [3] is to take a 2D laser radar device and building on previous work where a prism was spun in front of it to achieve 3D perception. The main issue with the previous work was that the use of the prism dealt with polar coordinates, which had to be converted to Cartesian. This is computational expensive and the authors reduce the amount of time to perform the computations through a parallel DSP and FPGA solution.

The author's implementation uses the DSP to control the motor that spins the prism, read encoder signals from the motor, and interpret RS422 data from the laser radar. The FPGA is used to calculate the velocity profiles for each group of data collected by the DSP.

In [8] the authors present a method of creating a 2.5D map for a humanoid robot to navigate through. The term 2.5D comes from the fact that a 2D map of cells is created along with a member that specifies the maximum high of an object in that cell. This type of algorithm suffers from two main problems. The first problem is that the map can be noisy since no uncertainty about can be predicted and the second issue is distinguishing between obstacles to avoid surfaces that cannot be stepped on.

To help with some of the fore mentioned problems the authors use a coarse occupancy grid. The final map is composed of cells which tell if there is floor, obstacle, or unknown along with the height of any objects present in the cell.

Conclusions

Since robotic mapping and exploration involves a lot of various types of operations it is apparent the FPGA is not the sole solution all of the time. In some applications where large matrixes of data are being iterated through, such as in Kalman filter approaches, then the FPGA can greatly increase the real time effectiveness of the system. One aspect of robotic mapping and exploration that lends itself to FPGAs is the fact that so many types of sensors are involved. The ability to reconfigure an FPGA to interface with almost any type of digital device makes it extremely use full for integrating information from multiple high-speed sensors. Through this survey it is apparent the FPGAs play an important role in robotic mapping and exploration; however, at this time the main use of them is a single kernel in a large system.

References

[1] Sebastian Thrun, “Robotic Mapping: A Survey”, CMU-CS-02-111 February 2002

[2] Yan Sun, Lauren A. Christopher, “3D Image Segmentation Implementation on FPGA Using the EM/MPM Algorithm”, ICASSP 2010 pp 670 – 673

[3] Weiyan Song, Weihai Chen, Jingmeng Liu, and Lihong Xu, “An Adaptive Speed-adjustable Scheme for 3D Laser Scanning Based on FPGA and DSP, IEEE International Conference on Industrial Informatics 2008, pp 86 – 91

[4] Leena Vachhani and K. Sridharan, “Hardware-Efficient Prediction-Correction-Based Generalized-Voronoi-Diagram Construction and FPGA Implementation”, IEEE Transactions on Industrial Electronics, Vol 55, No. 4, April 2008, pp 1558 – 1569

[5] Henning Lategahn, Wojciech Derendarz, Thorsten Graf, Bernd Kitt, and Jan Effertz, “Occupancy Grid Computation form Dense Stereo and Spare Structure and Motion Points from Automotive Applications”, 2010 IEEE Intelligent Vehicles Symposium, pp 819 – 824

[6] Andrey Soloviev and Maarten Uijit De Haag, “Three-Dimensional Navigation with Scanning Ladars: Concept and Initial Verification”, IEEE Transactions on AeroSpace and Electronic Systems Vol. 46, No. 1, January 2010, pp 14 – 31

[7] Brian Yamauchi, Alan Schultz, and William Adams, “Mobile Robot Exploration and Map-Building with Continous Localization” 1998 IEEE International Conference on Robotics and Automation pp 3715-3720

[8] Jens-Steffen Gutmann, Maski Fukuchi, and Mashiro Fujita, “A Floor and Obstacle Height Map for 3D Navigation of a Humanoid Robot”, IEEE International Conference on Robotics and Automation, April 2005, pp 1066 – 1071