RAPID WORLD MODELLING FOR ROBOTICS

CHARLES Q. LITTILE
Sandia National Laboratories

CHRISTOPHER W. WILSON
Spectra Research Institute

ABSTRACT

The ability to use an interactive world model, whether it is for robotics simulation or most other virtual graphical environments, relies on the users ability to create an accurate world model. Typically this is a tedious process, requiring many hours to create 3-D CAD models of the surfaces within a workspace. The goal of this ongoing project is to develop usable methods to rapidly build world models of real world workspaces. This brings structure to an unstructured environment and allows graphical based robotics control to be accomplished in a reasonable time frame when traditional CAD modelling is not enough.

To accomplish this, 3D range sensors are deployed to capture surface data within the workspace. This data is then transformed into surface maps, or models. A 3D world model of the workspace is built quickly and accurately, without ever having to put people in the environment.

KEYWORDS: Geometric Modelling, Surface Fitting, Triangulation, Decimation, Primitive Fitting

INTRODUCTION

World modelling is defined as the process of creating a numerical model of a real world environment, or workspace. This can be graphically displayed to provide the user with a 3D surface model of the workspace for simulations, analysis and task planning. In particular, world modelling enables the use of robotics systems to operate in the workspace by providing knowledge of the environment to the robot. This is a critical component of automated robotics. The robot must have a knowledge of its environment to interact with it; to grasp an object or avoid collisions while moving around.

A major disadvantage in robotics control through world models has been that geometric models *do not exist* for many objects. Generating these models using CAD design tools greatly increases the time required to complete a task with a robot. With the technologies discussed in this paper, arbitrary shapes, whether simple or complex, can be rapidly modeled to allow a robot to interact with previously undefined parts.

The objective of this project is to demonstrate a suite of hardware and software capable of collecting and processing 3D range data, and then transforming that data into surface maps. These maps, or models, will be added together to form a 3D world model of the target workspace.

Some application areas would include:

- Workspace updates, due to undocumented design changes, additions, deletions, etc.
- Automated Excavation
- Facility mapping [1]
- Underground storage tank mapping [2]
- Undersea operations
- Wherever an automated robot operates in an unstructured or changing environment.

CURRENT APPROACH

World models are traditionally generated using CAD tools. Modelers work from drawings where available. Frequently, the design does not match construction, and modifications and additions are always needed. Therefore the modeler must enter the workspace and carefully record distances and locations, then build up the model, piece by piece. CAD modeling of a robotic workspace is a manual, time consuming operation. A single object in a workspace, such as a barrel, can take hours to days. This traditional method breaks down when objects are difficult or impossible to CAD model, such as dirt or loose surfaces, or crushed or dented objects.

Even after good models are created, they must be accurately registered into the robot's world model to be useful. Any movement of objects will require updating, forcing the modeler to re-measure distances and locations.

Using the current approach, *creating the world model takes too long.*

NEW APPROACH

To overcome the problems of past approaches, and generate usable models quickly and accurately, we are developing tools to aid in model registration and in the creation of new or updated models. The primary hardware additions are video cameras and 3D range sensors. Selected range and distance measurements are used in updating and registering existing CAD models. Where traditional CAD modelling is not appropriate, dense range data is collected over an area and transformed directly into surface maps, or models. As additional areas are scanned and modeled, they are added together to form a 3D world model of the target workspace. An added advantage to this approach is that it can be done remotely, exposing sensors rather than people to the environment.

The following narrative describes the components used in this new approach.

Registration

A very real problem in collecting data from 3D range sensors is registering the data to the real world it came from. To be useful in a world model, the sensors need to be calibrated to give accurate range information relative to the sensor. Just as important is knowing where the sensor was at the time the data was collected. Given these two things, the proper transforms can be generated to translate the data to the coordinate system of the world model.

3D Data Collection

A 3D range sensor is used to capture surface data over an area. There are several 3D range sensors available [3]. Each provides a list of range values of detected surface points scattered over the scan area. With sufficient information about the range sensor, this data can be transformed into an x-y-z coordinate system. Given the position and orientation of the sensor, this data can then be registered to the real world target workspace.

Filtering

Filtering is often needed to reduce the quantity of data and enhance the quality. 3D range sensors are capable of generating enormous amounts of data. Current models may output range images of 256 by 256 readings or more (over 64,000 points per scan). This is often more data than is needed, especially in areas that have little detail, such as flat floors and walls. Resolution limits can add a false texture to truly flat surfaces. 3D range data also has noise and perhaps artifacts. These effects can often be reduced by filtering.

Editing and Segmenting

Editing and segmenting of 3D scattered data gives the user more control of subsequent processing. Usually, there is data within the scan that is not of interest. Frequently, false data points appear, perhaps due to sensor limitations with highly reflective surfaces. It is often advantageous to remove this extraneous data. Segmenting of the data allows subsections to be handled independently. This can be useful in dividing the workspace into logical objects. Both user assisted and automated techniques are needed.

Triangulation

The next step is to connect these scattered data points. The data is processed to form a triangulated surface, or a series of triangles. Often the vertices are the original scattered data points. This process may be trivial where data is collected in a uniform grid. Other techniques are available to handle truly scattered data, such as Delaunay triangulation [4].

Decimation

Due to the usually large number of scattered input data points, the resulting triangle list can be quite large; often too large to be useful in the world model. The next step is data reduction. One method is to reduce the triangle count by throwing out triangles, using a technique called decimation. As with any data reduction, decimation lowers resolution. The desire is to keep high resolution data where surfaces are changing rapidly, while giving up data where surfaces are not changing.

Primitive Geometry Fitting

Another avenue of reducing data, and perhaps aid in robotics task planning, is through primitive geometry fitting. Segments of the scattered data can be used to fit a simple geometry (spheres, cylinders, boxes...). The simple geometry object becomes the model, replacing the scattered data segment. Fitted geometries have much lower data requirements than triangulated surfaces. This technique has great promise in creating models of many surfaces, such as man-made objects (pipes, tanks, walls, floors, etc.).

RESULTS AND DISCUSSION

The results, along with some commentary of the work at our facility in the development of the components mentioned above, follows.

Registration

Data registration from our 3D range sensors is accomplished by calibrating all video and laser systems after they are installed. In the robot labs, the robot itself is used for initial calibration points. This allows us to simultaneously calibrate the sensors and back out the sensor positions for proper registration of their data output.

A problem we often face is having valid models of objects in the workcell, but incorrect locations because they have been moved around. One way we resolve this is by using stereo imaging techniques to locate pre-selected feducials, such as corners. By comparing these points to their counterpoints in the models, valid transforms can be generated to register the model with its real world position.

3D Data Collection

We are using a 3D sensor based on structured lighting. This system consists of a light source and a camera detector, with geometric processing to determine range. A high intensity tightly focused light (a laser) is steered over the target area. The reflected signal is detected through a camera. A-priori knowledge of the camera and laser (position, orientation, optics) allow triangulation techniques to map out surface range information. The beam pattern is a line. Targets larger than the beam width or field of view of the camera can be covered in multiple passes. Unlike laser range imaging sensors that provide a dense range 2D grid or image, the output of our structured lighting sensor is a string of irregularly spaced points, with little implicit connectivity.

Figure 1 is an image of a six foot bowl that sits in one of our robotics labs, from one of several video cameras mounted around the lab. The contents include a 30 gallon ribbed drum (with a large dent) imbedded in a vermiculite ‘soil’. A flat wood block was also included. Scattered data was collected from one of several overhead 3D sensors of the scene of Figure 1. The scanning gathered 17,067 x-y-z data points.

The time it takes to gather data is based on the desired density and the number of sweeps needed to cover the target area. This scan made 5 sweeps and took about 5 minutes.

In addition, we are also developing stereo imaging techniques to gather sparse range data. This will allow us to find surface points in the scan area along high contrast areas, such as edges and corners. This is complementary to the data gathered by typical 3D range sensors, which do best on broad surface areas and poorly at edges. The data will be fused to enhance coverage.

Filtering

From the beginning we often have too many points to work with, so we have developed a spatial filter to grossly reduce data counts. The filter is designed for the irregularly spaced data of our structured lighting system. This also allows it to be applied any 3D sensor data, and to multiple scans of data at once.

The initial data (17,067 points) was passed through the spatial filter to reduce data by eliminating neighbors within one inch. Additional data that fell on the floor surrounding the bowl was also removed by manual editing. This reduced the point count to 2,290 points. At this point, the data reduction is 86%. Figure 2 shows a view of the 3D range sensor data, after filtering.

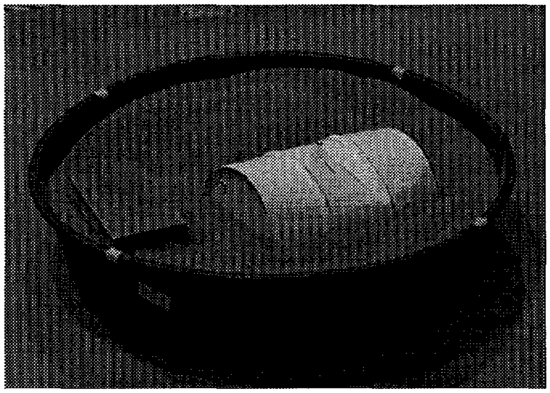

Figure 1. Image of target workspace

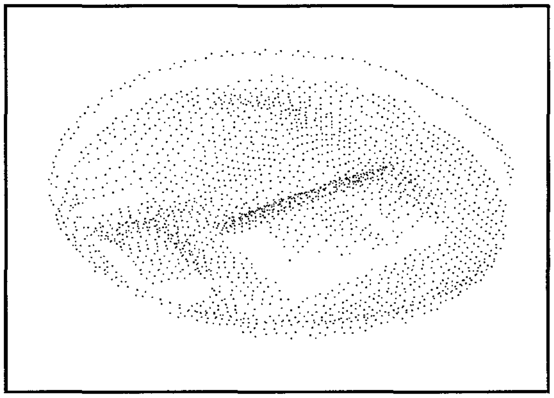

Figure 2. Scattered data from 3D range sensor

Triangulation

The next process is to connect the dots to form a triangulated surface. Our current process is based on a Delaunay triangulation [4]. This grids in x and y, but tracks the z value so irregular 3D sheets are generated.

Figure 3 is the result of triangulation of the data shown in Figure 2. The polygon count is 4,528. Notice in particular the ribs and the dent in the barrel are easily seen. This technique is computationally intensive, but at these data counts takes only a few seconds on our equipment (an SGI Indigo II). Although we can triangulate, the present algorithm has limitations when fitting data from multiple views. We are working on a full 3D triangulation algorithm.

It can be seen from Figure 2 that data was not collected over all surfaces. In particular, the area from the rim of the bowl to the fill, and at the base of the barrel, are blank. These areas fell in the shadow of the 3D sensor. A decision must be made, whether to create surfaces that span the holes or leave them blank. We usually choose to span them. Although this gives the false impression of data where none was collected, it does keep the robot out of those areas that the sensor could not see.

Another potential concern is the blanketing effect: raw surfaces tend to come out looking like a snow covered vista. It is difficult to tell when one part ends and another begins. This can be improved by editing and segmenting, and perhaps color coding of various pieces. Another way to increase user perception is by mapping video images onto the surface through texture mapping. These are planned for future work.

Decimation

We have included a decimation method [5] as part of our approach. Parameters have been added to maintain areas of high frequency, such as edges, while allowing low frequency surfaces, such as the flat areas between edges, to be reduced.

Figure 4 is a decimated version of the original surface model shown in Figure 3. Some fine detail is lost, but the salient features remain. The data count is 2,190 polygons, which is a 52% polygon reduction. This process takes a few seconds on our present system. Total data reduction is better seen comparing the original scanned data (17,067 vertices) to the final (1,121 vertices), or a 93% reduction.

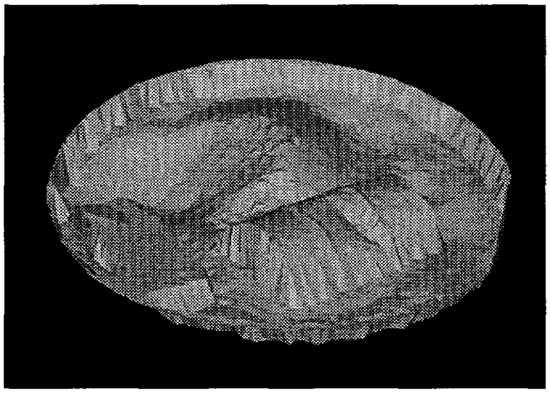

Figure 3. Result of triangulation

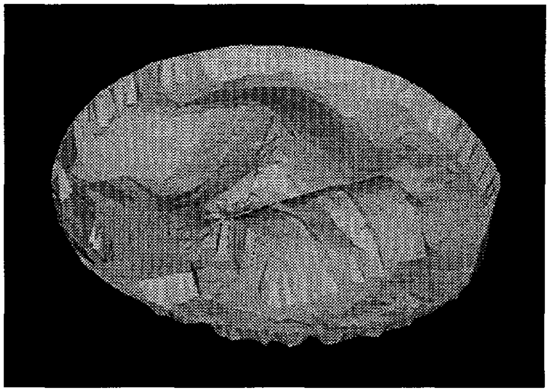

Figure 4. Result of decimation

Comments on Data Reduction

One of our requirements of the world model is that it be accessible in real time, i.e. the robot must be able to interactively engage the model as it is moving. Even with high speed display hardware, there is a limit to the number of polygons that can be re-displayed per second, which directly translates to the total polygon count for the world model. Performance is also affected in collision checking, where the moving robot model is checked against the rest of the workspace models, on a polygon by polygon basis. This places a penalty for models with high polygon counts. To simply avoid collisions, the workspace could be modeled with very crude bounding surfaces. On the other hand, operator perception benefits from high resolution. Therefore, parameters for data reduction often depend on the application needed.

Updating the World Model

The final step is to place the newly created surface into the world model. At this point, the registration issues have been resolved, and the world model becomes one step closer to representing the real workspace. The process continues until a sufficient level of modelling is reached to allow the system to interact with the environment.

The next two figures show an example of this type of interaction. A robot tool was tasked to move to the dented portion of the barrel and perform a swiping motion. Figure 5 is a view from the world model. The robot arm, the floor, and the bowl models where created by CAD modelling. The surface map from above was then added. The location and orientation of the target spot on the barrel was taken directly from the surface model. Figure 6 is a camera view of the robot tool performing the swiping motion operation. The robot can now interact with the model, both to move to an object, and to detect collisions before they happen.

To add some perspective, the bowl took about 4 hours to model, using traditional CAD modelling, with all measurements taken by hand (there where no drawings). The surface map took about 6 minutes from start to finish.

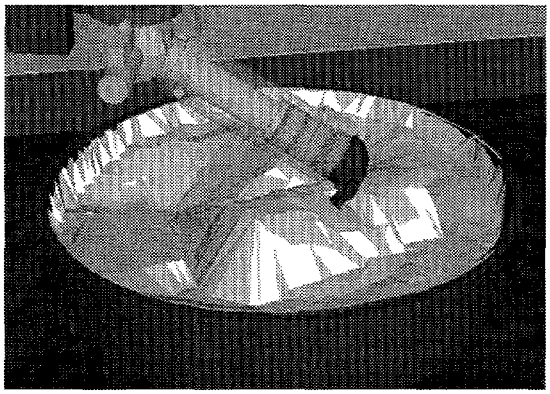

Figure 5. World model with robot

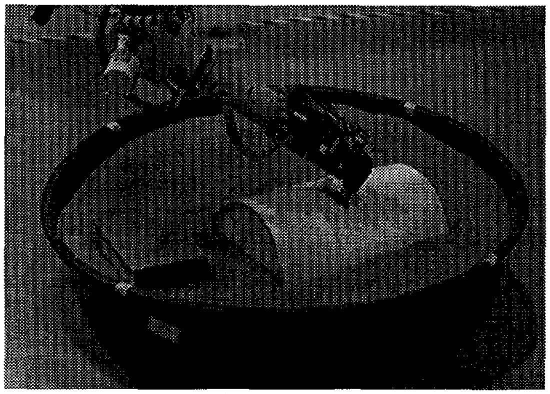

Figure 6. Real world image with robot

Primitive Geometry Fitting

We have experimented with primitive geometry fitting, using quadratic surfaces. An object, such as the barrel in Figure 3, can be replaced with a primitive geometry model of a cylinder. In some analysis tasks, such as planning the removal of the barrel, the cylinder model may be better than the more detailed barrel surface. It provides more information of the overall shape of the assumed object, even though fine detail, such as the dent and ribs, are lost.

CONCLUSION

We can now create and register models of arbitrary surfaces quickly, bringing structure to an unstructured environment. We have demonstrated 3D data collection, construction of surface models from scattered data, and have incorporated them into a working robotics system. We have developed techniques for data reduction through filtering, decimation, and primitive geometry fitting to further improve the process of rapid world modelling.

ACKNOWLEDGMENTS

This work was performed at Sandia National Laboratories, with collaboration from Oak Ridge National Laboratories. This research is sponsored by the Office of Technology Development, US Department of Energy under contract number DE-AC04-94AL85000.

REFERENCES

1. Barry, R. E., Little, C. Q., Burks, B. L. “Requirements and Design Concept for a Facility Mapping System.” *Proceedings of the ANS 6th Topical Meeting on Robotics and Remote Systems*, Monterey, CA: p. 775-783 (February 5-10, 1995)
2. Callow, D. S., Palmquist, R. D., Weatherby, J. R., Fugelso, D., "High Level Control and Data Acquisition System for Robotic Characterization of Hazardous Waste in Underground Storage Tanks," *Proceedings of the WM '96 Conference,* Tuscon, Arizona, (February 26-19, 1996).
3. Chase, B. L., Green, W. L., Abidi, M. A., “Range Image Acquisition, Segmentation, and Interpretation: A Survey.” Department of Electrical and Computer Engineering, The University of Tennessee, Knoxville, TN 37996-2100 (1995).
4. Fortune, S. J. “A Sweepline Algorithm for Voronoi Diagrams.” *Algorithmica* 2, 153-174. (1987)
5. Schroeder, W. J., Zarge, J. A., and Lorensen W. E. ”Decimation of Triangle Meshes. ”Computer *Graphics (Siggraph '92 Proceedings).* 26(2): p. 65-70, (1992).

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.