MODELING OF 3D OBJECTS UTILIZING POINT CLOUDS AND R-FUNCTIONS

By

Christopher M. Wendling

A Thesis Submitted to the Graduate

Faculty of Rensselaer Polytechnic Institute

In Partial Fulfillment of the

Requirements for the Degree of

MASTER OF SCIENCE

Major Subject: COMPUTER SCIENCE

Approved by the

Examining Committee:

____

Jeffrey C. Trinkle, Thesis Adviser

____

Ana Milanova, Member

____

Mukkai S. Krishnamoorty, Member

Rensselaer Polytechnic Institute

Troy, New York

April 2012

(For Graduation May 2012)

ABSTRACT

In many common robotic tasks, interacting within a 3D environment whether physical or simulated stands as a complex task. With the advent of Microsoft's XBOX 360 Kinect sensor a new form of data is now easily accessible to model 3D environments and objects. Utilizing information form point clouds yields a main problem, being the size of the data set and the second making sense of all the data. In robotic simulations it is often necessary to determine if any part of a robot will make contact or violate parameters of an environment, which requires a complete understanding of the surround 3D environment.

In this thesis, I have laid the foundation for modeling point cloud data with Rvachev Functions, which will be referred to as R-Functions moving forward. R-Functions are simply real valued functions whose sign is determined by the signs of its arguments. These functions can be combined via intersection and unionization, which represent the minimization and maximization of the two spaces being referenced. Given a set of core functions and a definition of the intersections and unions of said functions, R-Functions can effectively model raw point cloud data. The compelling need for modeling a point cloud is to reduce the size and complexity of information provided. With the advent of newer sensors, programmers now have the ability to access a point cloud with over 300,000 points, which has a minimum file size of approximately 3.6MB. With this modeling technique, information implicitly provided in a point cloud can be captured and stored much more effectively.

In order to bound point cloud data with R-Functions, I first made a simplification to the definition of R-Functions, which is that R-Functions can only be intersected which is to say be minimized. This assumption helps reduce some computational complexity in trying to fit a set of R-Functions to the data. From here, the program adds R-Functions one at a time to the model, trying to minimize the volume of the overall solid with every additional function. The functions being added to the model are R(0) functions which are a form of isosurfaces. In order to modulate the shape of an isosurface, the defined function has a set of coefficients that have the ability to alter the final shape of the function. The value of each coefficient is modulated within a set range with a certain number of steps between each end of the spectrum. With each iteration of the program,

the allowable range for each coefficient is reduced, allowing for refined manipulation of the surface as the coefficients become more granular. Once the function ceases to improve the volume it contains, a new function is added to the system and the process is repeated. Once the total volume of a surface is reduced to a certain amount does the system quit and output results.

By programming this modeling tool for point cloud data, I learned many things. One, there is a tradeoff between program performance and accuracy of modeling, a fudge factor of sorts. Two, limiting myself to quadratic type functions reduces the sharpness of edges that can be created in the modeling tool, making it difficult to effectively bound boxy type items. Third, efficient placement of functions at the beginning saves large amounts of time later. Lastly, while this thesis only uses the intersection ability of R-Functions, great efficiencies in bounding quality can be made by using unionization also.

The significant implication of this thesis is that point clouds can be effectively modeled by a set of R-Functions given adequate time and compute capability. Through this modeling, point clouds can be interpreted and stored in a unique fashion that compresses and preserves the data implicitly available inside of a point cloud.