ornl

ORNL/TM-11198
(CESAR-89/24)

OAK RIDGE
NATIONAL
LABORATORY

MARTIN MARIETTA

# Non-Linear Surface Fitting of Laser Range Images Using a Hypercube Concurrent Computer

M. Finley

OAK RIDGE NATIONAL LABORATORY
CENTRAL RESEARCH LIBRARY
CIRCULATION SECTION
4500N ROOM 175
LIBRARY LOAN COPY
DO NOT TRANSFER TO ANOTHER PERSON
If you wish someone else to see this report, send in name with report and the library will arrange a loan.

OPERATED BY
MARTIN MARIETTA ENERGY SYSTEMS, INC.
FOR THE UNITED STATES
DEPARTMENT OF ENERGY

This report has been reproduced directly from the best [illegible]

Available to DOE and DOE contractors from the [illegible] and [illegible] Rd., [illegible] Oak Ridge, [illegible]

Available to the public from the National Technical Information Service, U.S. Department of Commerce, 5285 Port Royal Rd., Springfield, VA [illegible]
NTIS price codes—Printed Copy: A03 Microfiche A01

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.

ORNL/TM-11198
CESAR-89/24

Engineering Physics and Mathematics Division

# Non-Linear Surface Fitting of Laser Range Images Using a Hypercube Concurrent Computer

**Michael Finley**

Transylvania University

Prepared in partial fulfillment of the requirements of the Oak Ridge Science and Engineering Research Semester under the direction of Dr. Reinhold C. Mann, Research Advisor in the Engineering Physics and Mathematics Division

**June 1989**

Prepared by the
OAK RIDGE NATIONAL LABORATORY
Oak Ridge, Tennessee 37831
operated by
MARTIN MARIETTA ENERGY SYSTEMS, INC.
for the
U.S. DEPARTMENT OF ENERGY
under contract DE-AC05-84OR21400

MARTIN MARIETTA ENERGY SYSTEMS LIBRARIES

3 4456 0316205 7

## TABLE OF CONTENTS

Abstract . . . . . v

Introduction . . . . . 1

Image Acquisition . . . . . 1

Range Image Segmentation . . . . . 1

Surface Fitting . . . . . 4

Implementation on a Hypercube Concurrent Computer . . . . . 7

Conclusions . . . . . 11

# Non-Linear Surface Fitting of Laser Range Images Using a Hypercube Concurrent Computer

Michael Finley

## Abstract

Segmentation and non-linear surface fitting with laser range images are attempted on a hypercube concurrent processor. Low-level software acquires range images onto the hypercube. A substantially modified existing range image edge detector works in the hypercube image analysis environment to accomplish segmentation. The Levenberg-Marquardt algorithm implements the non-linear least squares fitting of the surfaces.

Problems were found with each part of the approach. The edge detection algorithm is unnecessarily complex and yields an all too cluttered edge map. The fitting routine is limited by hardware in the number of surfaces that it can process. Finally, the fitting approach requires input estimates of the parameters which do not appear to be generalizable for any image, thus keeping the process from being totally automated. Other research in surface fitting on serial computers has yielded results which may be helpful additions to solve the problems of automation.

## Introduction

This report describes an experimental implementation of a computer vision system using a laser range camera. It consists of three independent parts (segmentation, surface fitting and recognition), of which only one is complete (segmentation) and one is near completion (fitting); the third has become a separate project of its own. The range camera is a state-of-the-art Odetics, Inc., Laser Range Scanner and all programs in this application were written in the C language for a 64 node hypercube parallel processor.

This work was performed under the Oak Ridge Science and Engineering esearch Semester (ORSERS) at the Center for Engineering Systems Advanced Research (CESAR) of the Oak Ridge National Lab. The study was directed by Dr. Reinhold Mann.

A range image is a picture of a scene in which the brightness of a point represents the distance from the camera to that point. A device which produces such images has obvious utility in machine vision where other techniques, such as stereo triangulation with conventional cameras, require a large amount of computation while yielding ambiguous or erroneous range information. Given a range image of the world around the scanner (located, for example, on a mobile robot), a three dimensional model of the environment can be created such that objects can be recognized. The procedure from digitizing a range picture of a box, for example, to labeling that part of the picture as "box" is, however, a difficult problem. The steps taken towards a solution to this problem in this particular study are image acquisition, segmentation, and surface fitting (Fig. 1).

## Image Acquisition

The laser range camera is controlled through a DR-11W card in a VME system. The raw output of the camera is a 32768 byte buffer (software by Dr. Ralph Einstein) which must be interpreted for use. This buffer is transferred to the NCUBE parallel processor (transfer software by Dr. Judd Jones) where it can be interpreted. A program was written for this study (and permanently added to the vision laboratory software) which separates the 32K image into two 16K images: one containing information about the reflectivity of the scene and one containing the range data. This yields a range image as a 2D array ($128 \times 128$) of 1 byte (unsigned character) values. These values are interpreted as integers between 0 and 255 which correspond to distance from the camera to the points in the scene, at intervals of (approximately) 1.4 inches. The camera can also be operated in a more accurate range mode (reflectivity accuracy is lost) with range resolution up to 0.7 inches.

## Range Image Segmentation

Once the 16K range image is acquired, a program is invoked to separate the areas of the picture that correspond to different objects in the scene. This is called a segmentation program and has led to several problems in working with range images. Dr. Griff Bilbro implemented a segmentation algorithm which uses complicated morphological techniques. His code operated on synthetic data (real range images were not available at the time) in 64K images. That code was modified during

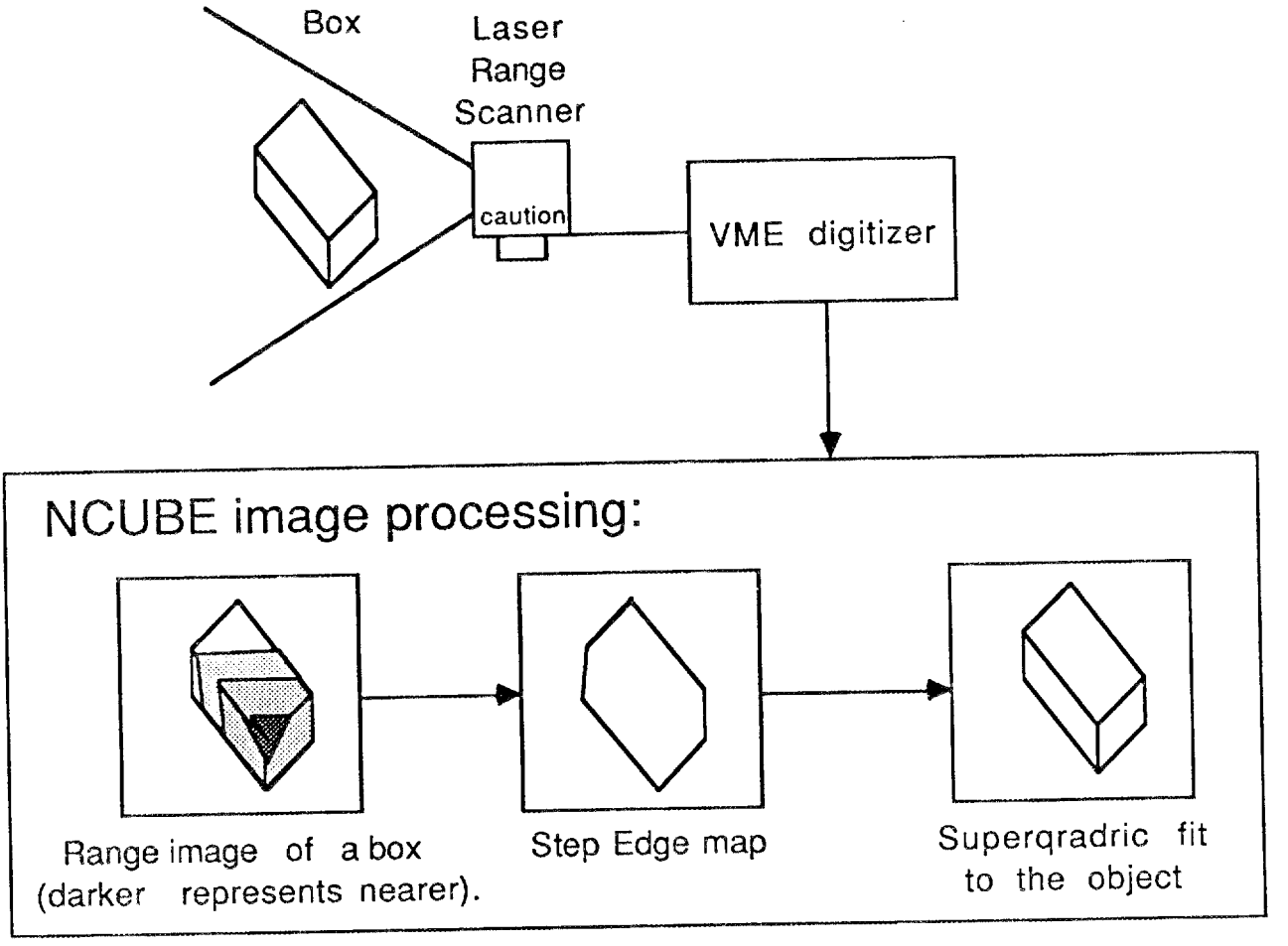


Figure 1. The Laser Range scanner produces an image. The computers digitize the image, create an edge map and fit surfaces to separate objects.

the current project to segment the 16K range images and operate in the standard vision laboratory environment. Dr. Bilbro's approach was found to be unnecessarily complex for the smaller images and needed modification to produce useful results.

Essentially, Dr. Bilbro's code finds edges of both step (discontinuities in depth) and roof (abrupt changes in surface curvature) types. The roof edges complicated the scene by finding, for example, that the faces of a cube were different objects. For the purpose of fitting surfaces (the next part of the study), the roof edge detection was eliminated (technically, the gain for roof edges has been set to zero; otherwise the code is intact). In addition, several complications with real range images cause problems in the edge detector. The problem may be explained by thinking of a person looking head-on at a wall. In this simple scenario, points on the wall directly in front of the observer are nearer than those found above or below (or left and right) the line of sight. The result is that the edge detector shows concentric circles around areas where the range finder's precision yields an equal distance. This distance changes abruptly (the least significant bit is increased by one) at the threshold of the camera's precision. A wall 3.5 feet from the range camera, for example, is seen as four concentric rings of equally distant points (Fig. 2). The solution to this problem is part of a calibration scheme (currently, the work of Dr. Frank Sweeney) that will be necessary for any application of the range images.

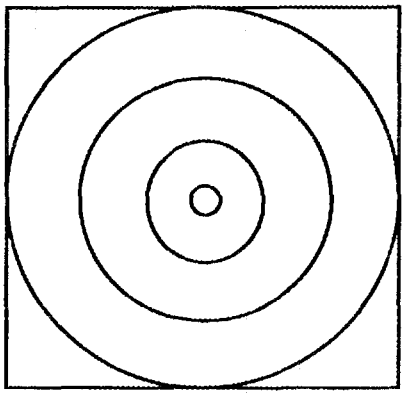

Figure 2. Edges in a range image of a flat surface.

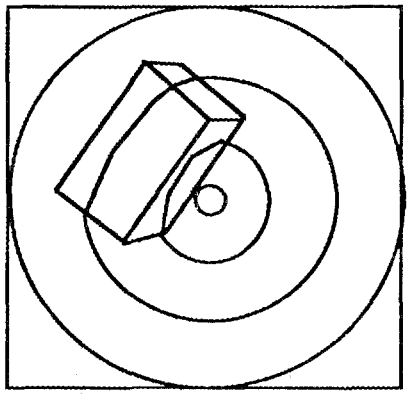

Figure 3. Edges in a range image of a box on a flat surface.

A similar problem that arises due to the range camera's resolution is that objects of a significant size in the range image are also plagued by the ring edges (Fig. 3), while smaller objects have undetectably small variations in their surfaces, and are shown to be closer than the background, but flat even when they have structure. A cylinder of diameter 5.5 inches on a wall at 3.5 feet from the camera, for example, appears to be a flat surface (Fig. 4). Simple range images were taken as data for the study, but they are disturbed by the precision problems to the extent that synthetic data (simulated surfaces) have become the only means of evaluating the fitting routine's

performance. When range images of larger objects can be taken and transformed under calibration, these problems should be resolved. A final point about the edge detector must be made: some real-world images (of part of the laboratory room) were taken and edge detection was done on them. The result is most commonly a picture cluttered with edges to the point that objects become indistinguishable. Real objects can have very complex shapes, and a different approach will be needed to produce a useful edge map from complicated images.

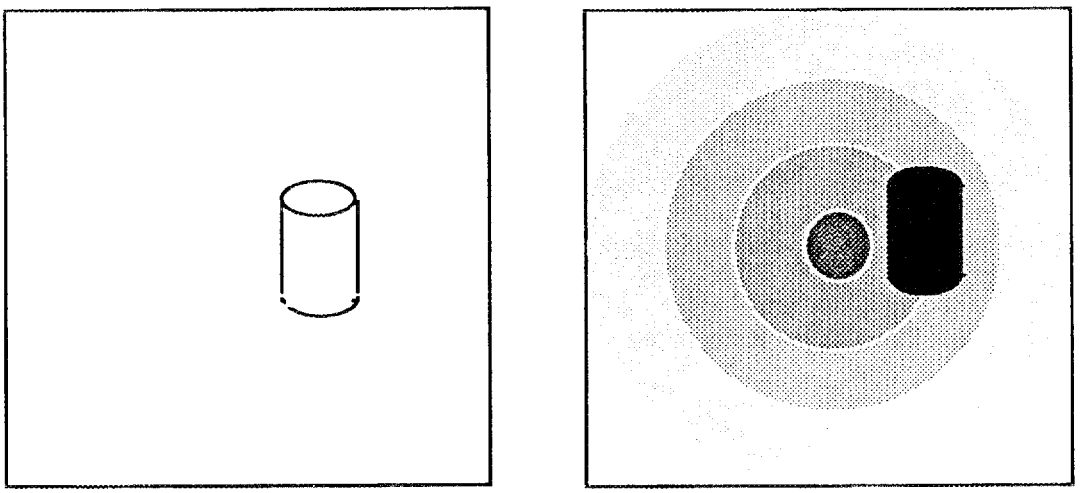

Figure 4. Left: A simple image as seen by the eye. Right: The range image (darker represents nearer; points with equal shading are determined to be equidistant from the camera).

## Surface Fitting

The third and final part of the study involves fitting surfaces to the patches outlined by the edge detector. The idea is common in image analysis: to reduce a set of N points in space to a set of M<<N surfaces which model the points. Essentially, it is a problem in data reduction; the goal is to reduce 16K of information to some smaller amount of information which significantly represents the original data in a manageable way. The method chosen is one that has been applied to range images[1,2,3] on serial computers. The advances of our implementation are the use of machine-segmented rather than hand-segmented data and the implementation of simultaneous fitting to many surfaces on a hypercube concurrent processor.

The surfaces chosen for fitting are called superquadrics. This family of curves was invented by danish designer Peit Hein as an extension of basic quadrics.[4] The functional representation of the surface depends on 5 functional parameters (scale on the x, y, and z axis and two exponents which govern the squareness in the x-y plane and along the z axis), three position parameters (center of the superquadric in 3-space), and three orientation parameters (the Euler angles). Altogether, these 11 parameters describe a wide variety of objects from any ellipsoid to any right-angled parallelepiped[3] (Fig. 5). The points on the surface can be represented in parametric

form in spherical polar coordinates. Alternatively, an implicit density function for the superquadric is given as follows[2]:

$$S(x, y, z) = \left( \left( \left( \frac{Xs}{a_1} \right)^{\frac{2}{\varepsilon_2}} + \left( \frac{Ys}{a_2} \right)^{\frac{2}{\varepsilon_2}} \right)^{\frac{\varepsilon_2}{\varepsilon_1}} + \left( \frac{Zs}{a_3} \right)^{\frac{2}{\varepsilon_1}} \right)^{\varepsilon_1}$$

Where $Xs = Xs(x_0, x, y, z, \alpha, \beta, \gamma)$
$Ys = Ys(y_0, x, y, z, \alpha, \beta, \gamma)$
$Zs = Zs(z_0, x, y, z, \alpha, \beta, \gamma)$

And $x_0$ = center of the superquadric on the X axis
$y_0$ = center of the superquadric on the Y axis
$z_0$ = center of the superquadric on the Z axis
$\varepsilon_1$ = squareness along the Z axis
$\varepsilon_2$ = squareness in the X-Y plane
$a_1$ = scale on the X axis
$a_2$ = scale on the Y axis
$a_3$ = scale on the Z axis
$\alpha, \beta, \gamma$ = Euler angles of rotation

Using this function, a point (x,y,z) lies on the surface of the superquadric if S(x, y, z) equals unity. This function can be evaluated and the parameters adjusted to yield a fit. The problem arises, however, that this definition says nothing about the points on the surface that are not represented by data. As an example, this function will fit points that lie in a plane to an ellipsoid by making the radius of the ellipsoid approach infinity. The points will lie on or very close to the surface, but the superquadric yielded is infinitely large, which is a false model of some finite surface. Instead, a second form of the surface is defined[2]:

$$R = a_1 a_2 a_3 (S - 1)$$

where a point (x, y, z) is on the surface of the superquadric if R equals zero. Using this function, the scale parameters along each axis ($a_1$, $a_2$, and $a_3$) will be minimized along with the adjustments made to the other parameters. The new function R also

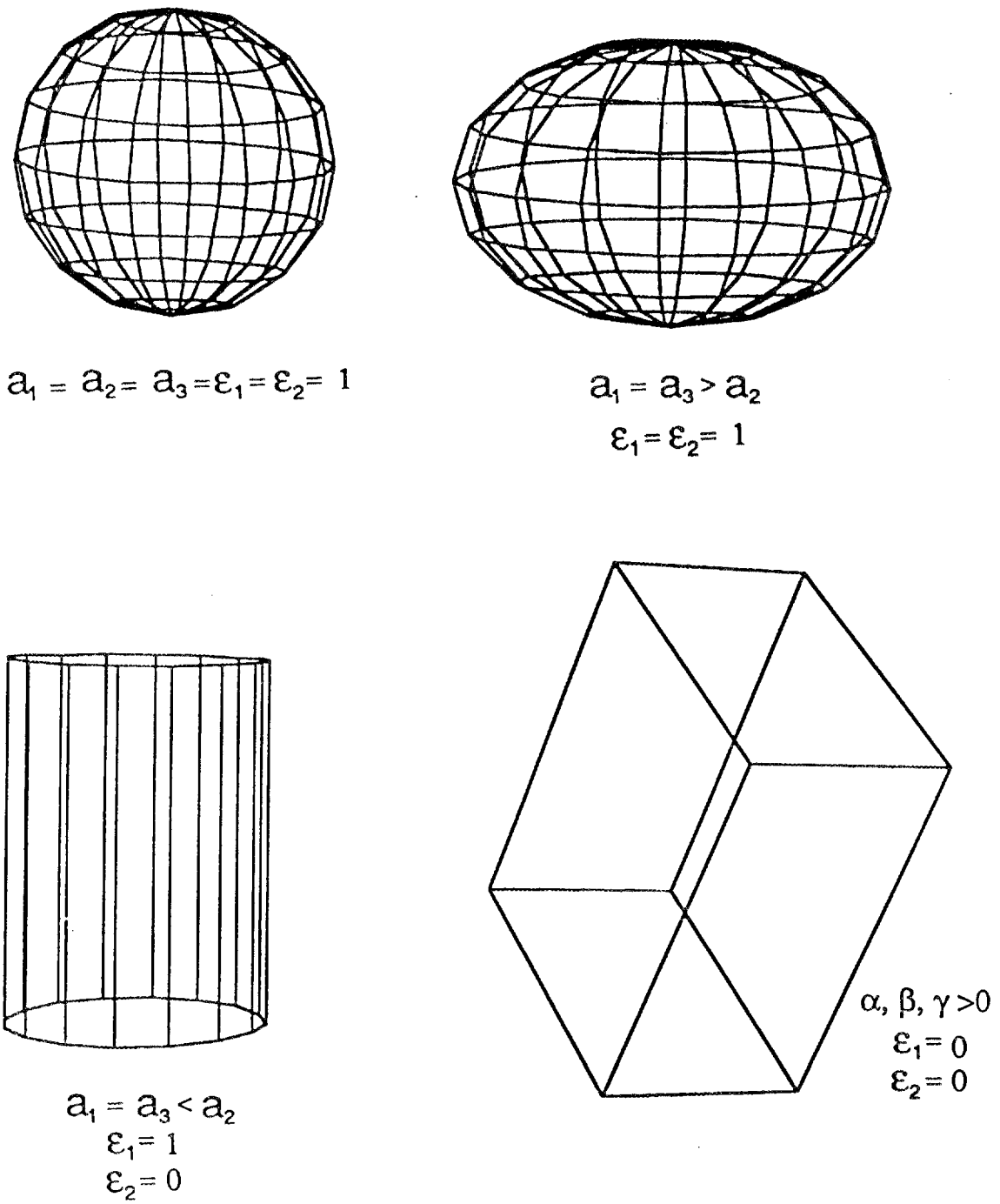


Figure 5. Sample Superquadrics

has complications, however. It is clear that when any one of the scale parameters is zero, all points will be determined to lie on the surface (in fact any point will lie on the surface) which is clearly not the case. In general, this is avoidable by ensuring that the scaling parameters are large compared to unity, therefore R is the superquadric form of choice[2].

The function is clearly nonlinear and thus requires an iterative procedure to arrive at the correct parameters. The chosen method, named after its inventors Levenberg and Marquardt, is a standard for nonlinear least-squares fitting[5] (Fig. 6). It requires three inputs: an initial guess for the eleven parameters, the gradient vector of the function with respect to the parameters at any point, and a means of measuring the "goodness of fit." The input parameters are estimated from the data, the gradient vector was found using Mathematica[tm] (which accomplishes, among other things, symbolic partial differentiation), and the measure of the fit is defined as follows[2]:

$$\chi^2 = \sum_{i=1}^{N} R_i^2$$

where $\chi^2$ represents the "goodness of fit" and N is the number of points (x,y,z) which belong, according to the segmentation of the image, to a surface.

The fitting begins by finding the error and a modified Hessian matrix for the input parameters. The solution to the matrix yields a vector of recommended changes to the parameters. The parameters are thus adjusted, and the new error and modified Hessian are computed. If the new error is smaller, the adjusted parameters are accepted and a new adjustment is calculated; otherwise, the adjusted parameters are dropped and smaller adjustments are calculated. This procedure continues until some criterion (insignificant error decrease or high number of iterations) is met, and the final best fit parameters are output[5].

## Implementation on a Hypercube Concurrent Computer

In this study, the implementation of the algorithm was in parallel on the hypercube processor. To understand how this was accomplished, a brief explanation of the hypercube architecture and parallel image processing is necessary. The hypercube consists of a set of $2^n$ (integer n>=0) processors (also called "nodes") that are linked as if they were the vertices of an n-dimensional cube. A 0-cube, for example is single node ($2^0$=1) with no links (no other nodes to link with), while a 3-cube is composed of eight nodes ($2^3$=8), each of which is linked to three other nodes like the vertices of a standard cube (Fig. 7). Image data are distributed among the nodes of the hypercube in horizontal strips of the input picture (Fig. 8). The result is a loosely synchronous SIMD (single instruction, multiple data) algorithm. Each node can operate on its part of the data, and communicate results to other nodes as needed. The goal of a good parallel program is to balance the workload of each node because the total time for execution of a program is the largest time for any single node. By balancing the number of data points on each node initially, this can be achieved approximately.

$$\Theta = [\varepsilon_1\ \varepsilon_2\ a_1\ a_2 a_3\ x_0\ y_0\ z_0\ \alpha\ \beta\ \gamma]^T$$

Levenberg-Marquardt modified Hessian Matrix

$$\alpha_{i,j} = \begin{cases} i = j\ ? & \frac{1}{2}\frac{\delta^2\chi^2}{\delta\Theta_i^2}(1+\Lambda_i) \\ i \neq j\ ? & \frac{1}{2}\frac{\delta^2\chi^2}{\delta\Theta_i\delta\Theta_j} \end{cases}$$

$$\beta_i = -\frac{1}{2}\frac{\delta\chi^2}{\delta\Theta_i}$$

Solve: $\sum_{k=1}^{M}\alpha_{j,k}\Delta\Theta_k = \beta_k$

For the set $\Delta\Theta_k$

And find a new set of parameters:

$$\Theta_k = \Theta_k + \Delta\Theta_k$$

Figure 6. A single iteration of the Levenberg Marquardt Algorithm

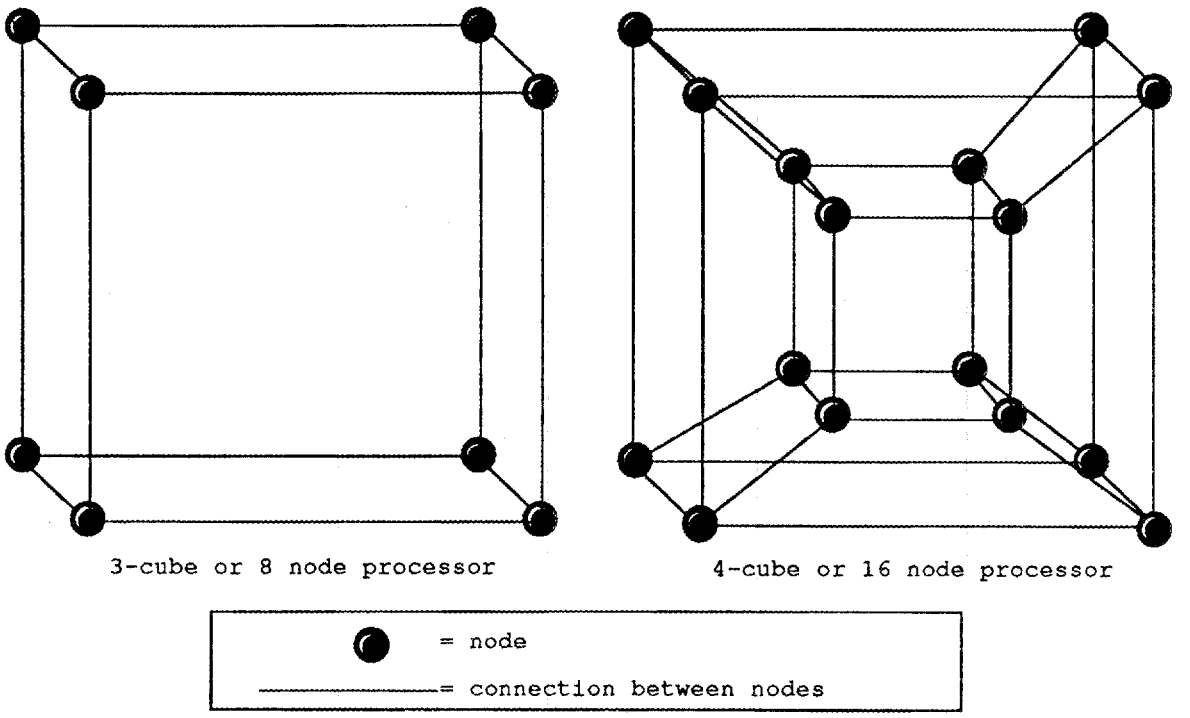


Figure 7. The Hypercube architecture (Figure by Dr. J. P. Jones).

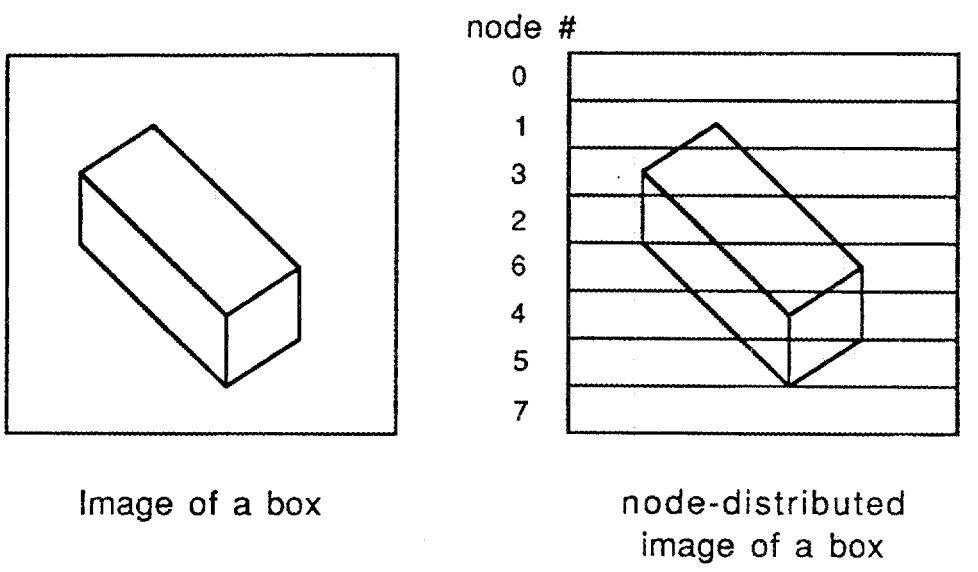


Figure 8

Given that equal strips of the range image are distributed among the nodes of a hypercube, each node will have to process its part of each surface. Clearly, more

than one node will have parts of any sizeable surface (imagine a vertical pole in the image-every node will have a piece of this image which contains points that are on the pole!). Thus each node must use the same parameters and parameter adjustments to find its local error of fit for any surface. Then, each node must tell every other node its results, so all nodes together compute a total error, and adjustments in the iterative procedure.

Unfortunately, the problems of parallel computing begin to appear when the adjustments are computed. Levenberg-Marquardt solves a modified Hessian matrix to find the adjustments. This matrix is $11 \times 11$ in the case of the eleven superquadric parameters, and it involves summation over the points on a surface, which means that it too must be communicated between all nodes. In floating point values (4 bytes), the matrix for one superquadric alone is nearly 500 bytes. In a complex scene with many objects, the amount of data that is transferred between nodes becomes very large. The manufacturer's recommendation is that about 10KB (or 20 superquadrics) should be communicated at a time. If each node processed only its own superquadric, many nodes would be idle (the difference between the number of surfaces and the number of nodes would be large-those nodes have nothing to do). If, however, only a single superquadric's matrix is communicated at a time, the communication time becomes unwieldy and the speed of the machine is lost.

The present implementation of the fitting routine solves most problems of parallelization, excluding the case of a large number of superquadrics. Since presently only simple or synthetic data are used, this has not yet been a problem. In addition, the present implementation is readily adjustable for the study of a new class of problems that this work has revealed. Namely, when one surface converges (the best fit parameters are found), many nodes will no longer have some part of their data: that part which was on the surface that is finished fitting. This leads to a load imbalance among the nodes during execution, and algorithms for this type of dynamic load balancing have not yet been studied.

With some understanding of the parallel problems, other problems intrinsic to the fitting routine are now introduced. Fitting a surface which depends only on eleven parameters to a set of $M >> 11$ points is termed an overdetermined optimization problem[1]. Essentially, the surface is overconstrained when trying to adapt to all M data points. In the algorithm, this is seen as a very large number for the "goodness of fit" or error value. In particular, when the squareness parameters become very small, their reciprocals (which are used in the definition of the surface) become very large exponents. Overflow errors are common in attempts to fit square figures because this requires the exponents to approach infinity. The limits set on just how high the exponents can go depends on the number M because the summation over all M points of the various partial derivatives (in the Hessian matrix) is usually the quantity that overflows. In addition, the problem of not allowing the scale parameters to approach zero (recall the incorrect solution that this yields) means that they must remain above unity, and their product (recall the definition of R) can be a large number when summed over all points. A possible solution to this problem can be found in the literature and consists of disregarding data points whose individual error is beyond a threshold[2]. This possibility has not been investigated.

Additionally, the use of the volume-constrained function R rules out some classes of objects. A very thin object, for example, must have one of its scaling parameters

set low, but this leads it to be less than unity, and the iterative procedure is quickly dragged to a halt by making it zero. Some researchers have used the addition of Poisson distributed noise to the data during the iterative procedure to eliminate local minima of the error function in the eleven dimensional parameter space, but exactly when to add that noise and to what types of images is not specified[1]. These same researchers show an error function which decays exponentially with further iterations, a result which has not been found under this project[1]. One last problem which appears to have been solved elsewhere is the initialization of the parameters, the input "guesses" to the Levenberg-Marquardt algorithm. Some researchers claim that "canonical" guesses (simply constants that are equivalent for every surface) suffice for a quality fit[2], and again, this conflicts with the results of the present study, which found that rather tedious procedures are needed to approximate the true parameters before the iterations begin.

## Conclusions

### Summary

The parallel code for the representation of three dimensional data with superquadric surfaces was written in C on a hypercube concurrent processor. The implementation of this study differed from others in its parallel nature and in the use of real range data. Both of these innovations have lead to severe complications in addition to other problems that are inherent to the method, and overall the program still remains in an experimental stage. The next step in the work consists of communication with other researchers in the area for resolution of some disagreements, and no doubt improvements in the programming. This communication has already been initialized with the group of Dr. R. Bajcsy from Pennsylvania State University.

### Recommended Future Work

The laser range scanner must be correctly calibrated for the segmentation programs to work. In addition, the segmentation programs need modification to deal with a larger class of images (from simple to complicated scenes), and should be data-independent. Finally, the surface fitting routine has many problems, e.g., choosing input parameters, adding noise, allowed range for different parameters, limited choice of shapes, etc. which remain to be resolved.

## Acknowledgements

I would like to thank the entire group of scientists working in the CESAR laboratory for what has been a most rewarding experience. In particular, though, I am grateful to Dr. Judd Jones for programming help, Dr. Chuck Glover for general methodology advice, and Dr. Reinhold Mann for supervising this project.

References

1. F. Solina and R. Bajcsy, "Range Interpretation of Mailpieces with Superquadrics," *Proceedings of the USPS Advanced Technology Conference*, **472-481**, Washington, D.C. (1986).

2. R. Bajcsy and F. Solina, "Three Dimensional Object Representation Revisited," *Proceedings of the First International Conference on Computer Vision*, **231-240**, London, England (1987).

3. T. Boult and A. Gross, "Recovery of Superquadrics from 3-D Information," *SPIE Vol. 448 Intelligent Robots and Computer Vision: Sixth in a Series* **358-365** (1987).

4. M. Gardiner, "The Superellipse: A Curve That Lies Between the Ellipse and the Rectangle," *Scientific American*, (September 1965).

5. W. Press, B. Flannery, S. Teukolsky and W. Vetterling, "Numerical Recipies", Cambridge University Press, Cambridge, England (1986).

## INTERNAL DISTRIBUTION

1. M. Beckerman
2. J. R. Einstein
3. C. W. Glover
4–8. J. P. Jones
9–13. R. C. Mann
14. F. C. Maienschein
15. H. S. Payne
16. F. J. Sweeney
17–21. C. R. Weisbin
22. EPMD Reports Office
23. Central Research Library
24. ORNL Technical Library
25–26. Laboratory Records
27. ORNL Patent Office
28. Laboratory Records–RC
29. B. R. Appleton

## EXTERNAL DISTRIBUTION

30–34. Mike Finley, Transylvania University, Lexington, KY 40508

35. Office of Assistant Manager, Energy Research and Development, Oak Ridge Operations Office, US/DOE, P.O. Box 2001, Oak Ridge, TN 37831

36–45. Office of Scientific and Technical Information, US/DOE, Oak Ridge, TN 37831