Computer-Aided Design and Applications

http://www.cadanda.com

An Integrated Platform for 3D Measurement with Geometric Reverse Engineering

J. A. P. Kjellander[1] and Mohamed Rahayem[2]

[1]Örebro University, johan.kjellander@oru.se
[2]Örebro University, mohamed.rahayem@oru.se

ABSTRACT

Geometric Reverse Engineering (GRE) can be described as the process of fitting surfaces to point data and connecting them to topologically well defined CAD models. We have mounted a laser profile scanner on an industrial robot with a turntable and interfaced them to an Open Source CAD platform. With this tool we have developed an integrated system that can automatically plan and control the robot movements needed to measure an object of unknown shape. Details of this work have been published earlier but we have not described the platform used to integrate the hardware with the GRE software. This paper illustrates the multi disciplinary nature of that problem and investigates what the requirements are for a suitable CAD tool.

Keywords: geometric reverse engineering, 3D measurement system, CAD tool.
DOI: 10.3722/cadaps.2009.877-887

1. INTRODUCTION

Geometric reverse engineering (GRE) is concerned with the problem of creating Computer-Aided Design (CAD) models by interpreting point data measured from the surfaces of real world objects. An introduction to the topic which is often referred to is a paper by Vàrady et al. [24]. In this paper the process is described as a sequence of four steps:

1. Data capture.
2. Preprocessing.
3. Segmentation and surface fitting.
4. CAD model creation.

Step 1 is closely related to 3D measurement technology. Manually operated measurement systems exist and yield good results but automatic systems have a potential of increasing productivity of course. For objects with similar shape it is relatively easy to design a measurement system which is automatic. A common solution is to use an optical sensor in combination with a mechanical device that moves the sensor so that it can see all parts of the object. Pito and Bajcsy [23] presented a simple system by combining a fixed range camera with a turntable. A more flexible system is described in [3], [13] where a Coordinate Measurement Machine (CMM) is used in combination with a laser profile scanner. A laser profile scanner with two Charge-Coupled Device (CCD) cameras fixed on a 3-axis transport mechanism is described in [11]. A motorized rotary table with two degrees of freedom and a laser scanner mounted on a Computer Numerical Control (CNC) machine with four degrees of freedom is described in [21]. Gallieri et al. in [5] presented a system based on a range laser scanner mounted on the arm of an industrial robot in combination with a turntable. A Next Best View (NBV) algorithm is used to select viewpoints minimizing the amount of unseen surface.

If the shape of the object is unknown, a planning algorithm is needed, that automatically adapts to the shape of the object and directs the numerically controlled mechanical device to move along paths that makes it possible for the sensor to see all parts of the object without collision with the object or the measurement hardware itself, see [3], [5], [11], [13], [21], [23]. For a comprehensive survey on automated view planning, see Scott et al. [19].

In step 2, a topology is created that connects neighboring points with each other, usually into a triangular mesh. Triangulation methods are described in [9], [16], [18] and [22]. The measurement result can then be displayed as a shaded facetted surface.

Step 3 divides the mesh into regions that can be fitted with higher order geometrical primitives like planes, quadrics or higher degree polynomial or NURBS surfaces. A lot of work has been published on this subject; e.g., [1], [2], [4], [20], [26].

In step 4 finally, surfaces are merged into a complete topological and geometrical boundary representation. Steps 2-4 are usually automatic but may involve different software products and threshold values may need to be adjusted manually to yield good results.

To completely automate the entire GRE process we believe that steps 1 through 4 need to be integrated. Such a system would need a numerically controlled mechanical device for orientation of the sensor and an automatic planning algorithm to control the movements. An iterative approach can then be applied where the system during step 1 first makes a coarse measurement of the object and then iteratively more precise measurements based on the result of previous iterations. During segmentation and fitting it may also be possible to increase the quality of the result by automatically making refined measurements with optimal sensor distance and angle of view.

An industrial robot with a turntable is a flexible device. It's fast, robust and relatively cheap. This is true also for a laser profile scanner. The combination is thus well suited for industrial applications where automatic GRE of unknown objects is needed in real time. To test the idea, we have set up a system of this kind in our laboratory, see Figures 1 and 2. We have also developed software to automatically plan and control the robot movements, acquire the measured data and perform planar segmentation and fitting. The concept was first presented by Larsson and Kjellander 2004, see [8]. The details of motion control and data capturing were published 2006, see [9]. An automatic path planning algorithm was published 2008, see [10] and details on accuracy 2008, see [14]. Recent work on fast planar segmentation was published 2009, see [27].

Fig. 1: Robot arm, turntable and laser profile scanner.

Integrating a robot with a laser profile scanner and software for the different types of data processing described above is greatly simplified if a common platform is available. We have chosen to use a tool named Varkon, see [7], available as C source code on Source Forge, see [25] and maintained by our research group. The system also includes a geometric modeling language MBS (Swedish abbreviation for Model Definition Language), based on a general purpose programming language with support for visualization, user interaction and external communication.

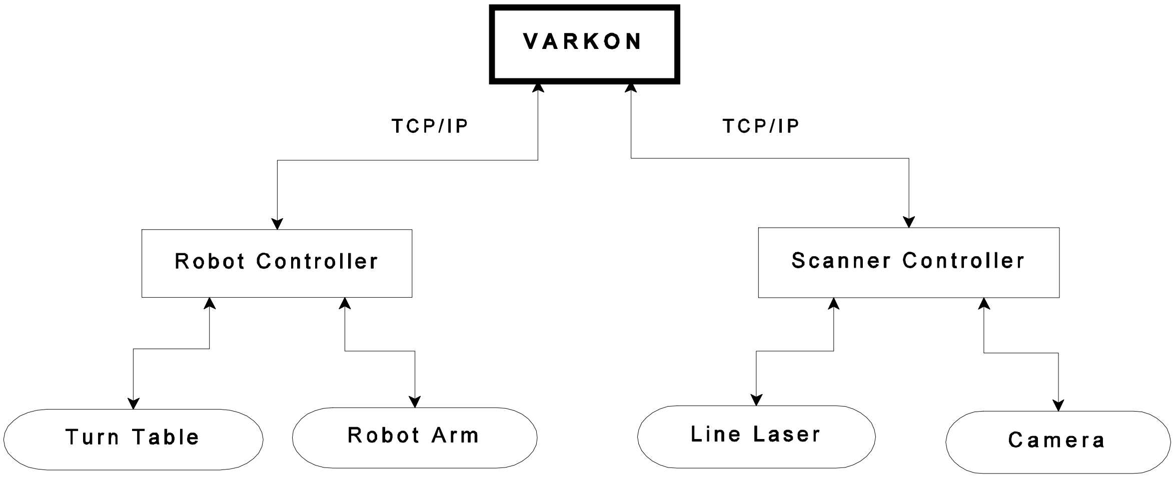

Fig. 2: Sub-systems controlled by Varkon.

This paper will not present any new major theories or techniques. Instead it will focus on the integration of 3D measurement with GRE and the requirements for a suitable tool.

2. TOOL REQUIREMENTS

The most important requirement for a tool of this kind is that it can model, represent and visualize 3D objects and perform geometrical operations on them. This is what CAD systems usually do but the application at hand also requires: (1) real time robot motion control and data acquisition, (2) real time profile scanner control and data acquisition, (3) triangulation of point data, (4) path planning and collision control, and (5) segmentation and fitting of point data.

A requirement that is valid in all application development is that the tool offers an environment on the right level of abstraction. We believe that a programming interface based on a high level geometric programming language is best suited for this application, especially if it also supports activities 1-5 above. The Varkon system includes a suitable programming language but did not include support for all the details of 1-5 above when the project started. The Open Source distribution model of the Varkon system however, encourages the development of new functionality and the architecture of the system is prepared for this. When needed, we have therefore added support that was missing to the basic Varkon system. Such job is done on C language level using Varkon's internal API's but this job has been a relatively small part of the research project. The application described in this paper is entirely developed using Varkon's high level programming language, MBS.

2.1 Geometric Modeling

The Varkon system includes a Graphical User Interface (GUI) which is used to create and execute generic models defined by MBS statements. The explicit result is stored in a persistent database and visualized using 3D graphics, see Figure 3.

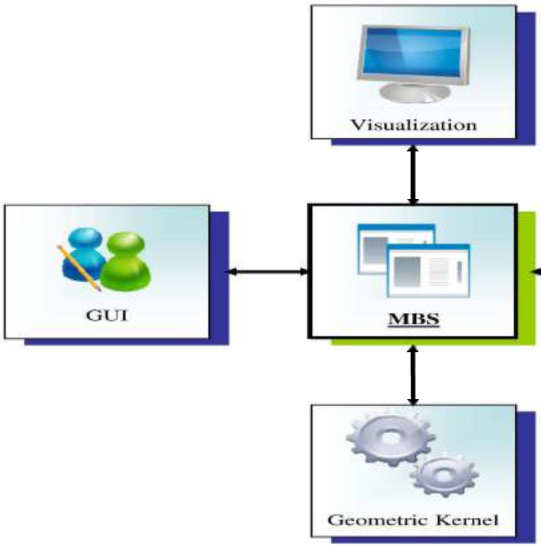

Fig. 3: The Varkon system.

MBS (Version 1.19C) includes 64 methods to create a total of 16 different geometric entity types. The most important entity types in this application have shown to be coordinate system, point, line, curve, bounded plane and triangular mesh.

MBS also includes a library of 40 different procedures and functions that can be used to perform various operations on geometric entities. The code sample in Figure 4 illustrates 3 methods to create straight lines using geometric functions.

```
MODULE Rectangle (FLOAT width:=2, height:=1);
  BEGINMODULE
    lin_ang(#1,vec(0,0),0,width);           ! Line with angle
    lin_offs(#2,#1,-height);                ! Line with offset
    lin_free(#3,startp(#1),startp(#2));     ! Line between 2 positions
    lin_free(#4,endp(#1),endp(#2));         ! Line between 2 positions
  ENDMODULE
```

Fig. 4: Code to create a rectangle from four lines.

Geometric modeling is a key issue in this application but steps 1-5 above require support for a wide range of complementary topics. MBS supports built in data types, evaluation of algebraic expressions and program flow control but also a total of 250 additional functions and procedures covering communication, visualization user interface and mathematics. We believe that a programming language of this kind is an important requirement for a successful implementation. See appendix 1 for screenshots showing the visual result of more complex MBS models.

2.2 Real Time Robot Motion Control and Data Acquisition

The robot consists of an arm “ABB IRB 140” with six rotational joints, a turntable “IRBP 250 L” and a controller “S4C”. The controller has a standard TCP/IP interface that can be used for remote control using RPC (Remote Procedure Call). To communicate with the robot we implemented support for TCP/IP and RPC in the MBS language. To control the robot movements we developed a concept of a scan path which is defined by three space curves, see Figure 5. With these curves as input, Varkon calculates a series of robot poses and turntable angles and sends them to the robot. While the robot is moving it collects actual robot poses at regular intervals together with timestamps for each actual pose. When it reaches the end of the scan path it transfers the actual poses back to Varkon, see [9].

Geometric modeling of space curves and support for RPC through TCP/IP is the main requirements for this part of the system.

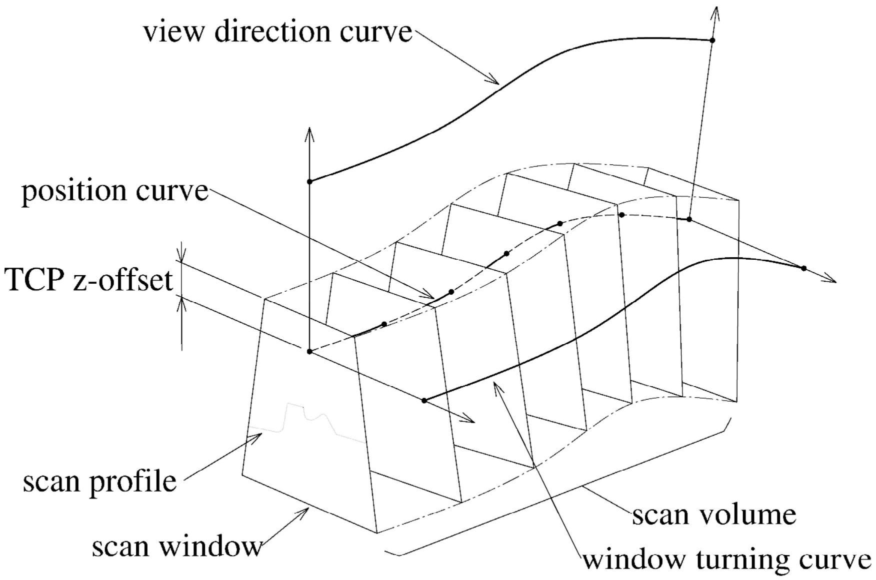

Fig. 5: Curved scan path defined by three curves.

2.3 Real Time Laser Profile Scanner Control and Data Acquisition

The laser profile scanner consists of a line laser and a camera "Sony XC-ST30CE CCD" connected to a controller with a TCP/IP interface which can be used to set scanner parameters and collect scan data. The line laser projects a straight line onto the surface of the object to be measured and the camera captures the image of the projected line. The image is then processed in the controller and the result is a set of 2D coordinates (a profile) that is sent to Varkon together with a timestamp, see Figure 6.

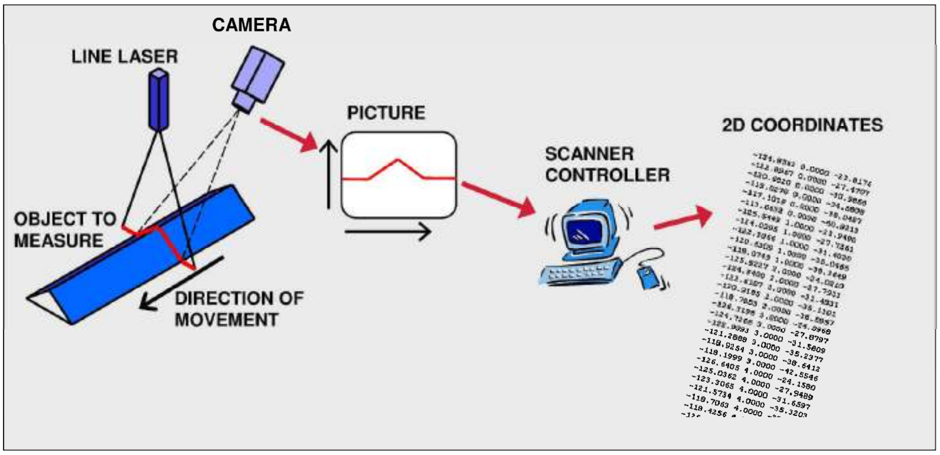

Fig. 6: From line laser to 2D profile data.

For each point in a 2D profile its corresponding 3D coordinates can be computed using data from the robot and interpolation based on the timestamps from the robot and laser profile scanner. This part of the system requires support for geometrical calculations and TCP/IP.

2.4 Triangulation and Storage of Point Data

It is inherent in the design of the laser profile scanner that the points delivered are well organized. Each profile represents points that are ordered from left to right in the camera picture and the profiles themselves are ordered sequentially along the scan path. After having filtered out points that are obviously erroneous, it is therefore relatively easy to connect remaining points to a triangular mesh. To support this we implemented a MESH entity in MBS with a vertex-edge-face relationship topology. Each scan path executed by the robot thus creates a MESH entity in the Varkon database. We believe that a natural representation of the measured data is an important requirement. Each MESH object also stores the original 2D profiles for later processing.

2.5 Path Planning and Collision Control

The system automatically creates the curves needed to define each scan path using an iterative process where the MESH entity of a previous scan path is intersected with a set of orthogonal planes. Each intersect yields a piecewise linear curve with two ends that may or may not overlap earlier scan paths. Ends with no overlap represent some part of the object that has not yet been scanned. A new scan path can then be created by connecting ends with no overlap that are reasonably close to each other. This algorithm was published 2008, see [10]. Figure 7 shows the result of the algorithm used on a relatively complex object.

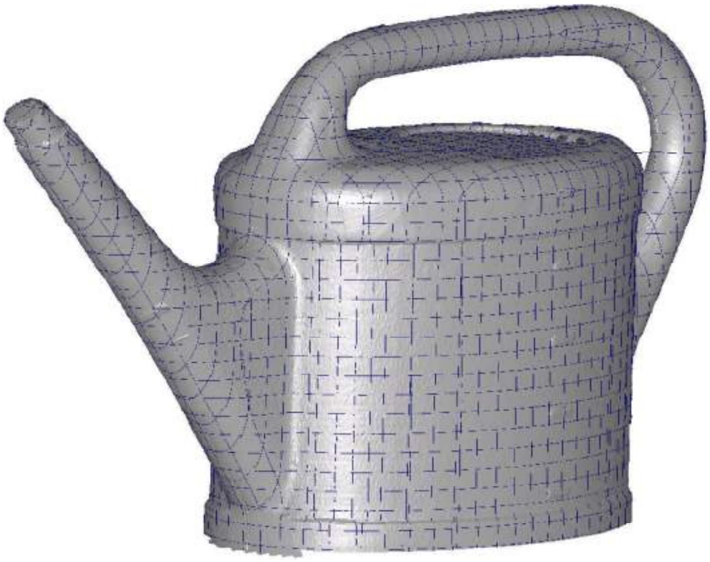

Fig. 7: Automatically scanned watering can.

A core part of the algorithm is to find curve ends that can be connected into a new scan path. This is a 3D search problem and we solved it by implementing an MBS procedure (POS_IN_CONE) that uses a current position and direction vector to test if candidate positions are inside or outside a limited cone pointing in the direction of the vector.

Each automatically generated scan path is tested for collision before it is executed in the robot. This is done by simulating the scan movement in Varkon with a parametric 3D model of the robot and a model of what is currently known about the scan object. A reasonable number of steps are taken along the scan path and for each step a robot/object intersects and robot/robot intersects is calculated. In order to make these calculations reasonably fast we model the surface of the robot with planar surfaces and the object with an orthogonal mesh of bounded planes, B_PLANE in MBS. As new scan paths are executed and more object surface is revealed a space carving algorithm is used to decrease the size of the object collision model.

Path planning and collision testing as described above requires geometric entities like MESH and B_PLANE but also fast search algorithms in 3D and fast plane/plane intersects.

2.6 Segmentation and Fitting of Point Data

Segmentation and fitting is the third step in the GRE process according to Vàrady et al. [24]. We have implemented a method for planar segmentation, see [15], based on a curvature estimation method described by Sacchi et.al. [17], where a seed triangle is selected and a region growing algorithm is used to connect neighboring triangles that are coplanar. The result is fitted with planes using Principle Components Analysis (PCA), see [12].

As a test object we have manufactured a copy of an object introduced by Hoschek et al. [6] and used by Sacchi et al. [17] and others, as a reference. See Figure 8. The object was measured using a linear scan path from left to right in the figure. The resulting triangular mesh is shown in Figure 9. Regions without triangles represent portions of the object that were not seen due to occlusion either of the camera or the laser.

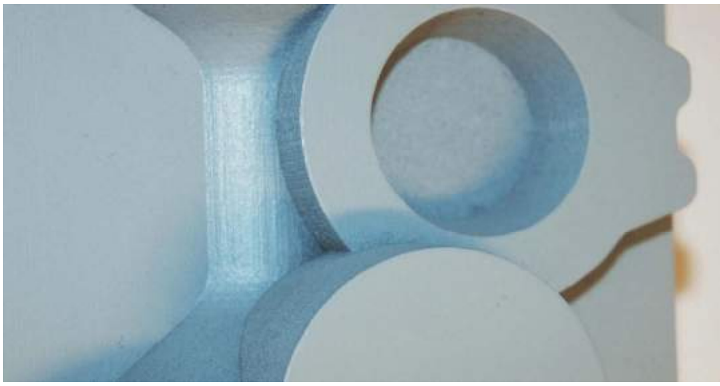

Fig. 8: The test object.

The result after segmentation is shown in Figure 10. Black triangles are seeds and grey areas the corresponding regions. The region in the lower centre is under segmented due to the fact that it was close to the camera and slightly out of focus.

Segmentation as described above requires an organized point set like the MESH entity and fast geometrical calculations in 3D to select appropriate seeds. To speed up and simplify plane fitting we also implemented a dedicated MBS procedure for the PCA method.

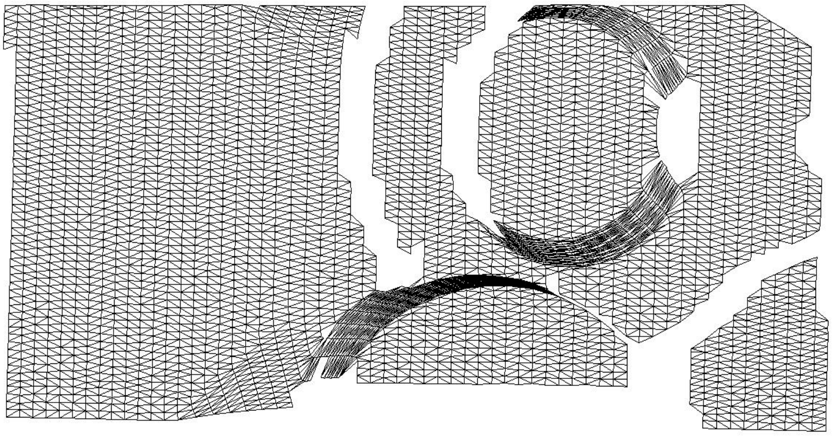

Fig. 9: Triangulated mesh.

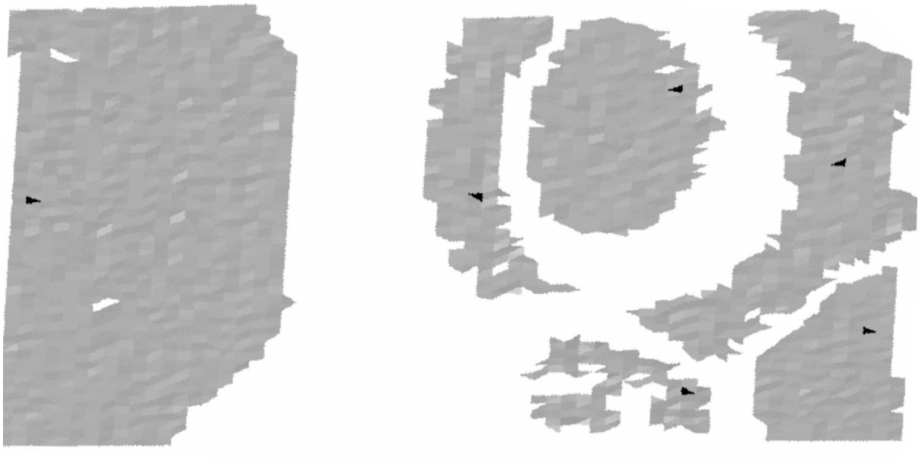

Fig.10: Regions with seeds.

2.7 Segmentation based on Laser Profiles

Section 2.6 shows how triangles are used for segmentation. With an integrated GRE system however, it is also possible to use lower level measurement data like 2D laser profiles directly from the scanner. In [27] we show that this is very effective compared to traditional segmentation based on triangulated point data. During the measurement process we store all laser profiles together with corresponding robot orientations. We then segment the profile data into straight line segments. A straight line segment is a strong indication that the corresponding surface is planar. A 3D region growing algorithm is then applied on all line segments and each region is finally used to fit a corresponding plane.

Figure 11 shows an object with a planar surface to the left which is tangent to a cylindrical surface with relatively large radius. In [27] we show that planar segmentation based on laser profiles for this object is 25 times faster than planar segmentation based on triangles. This is due to the fact that segmentation of profiles into straight line segments reduces the amount of data used in the region growing process considerably.

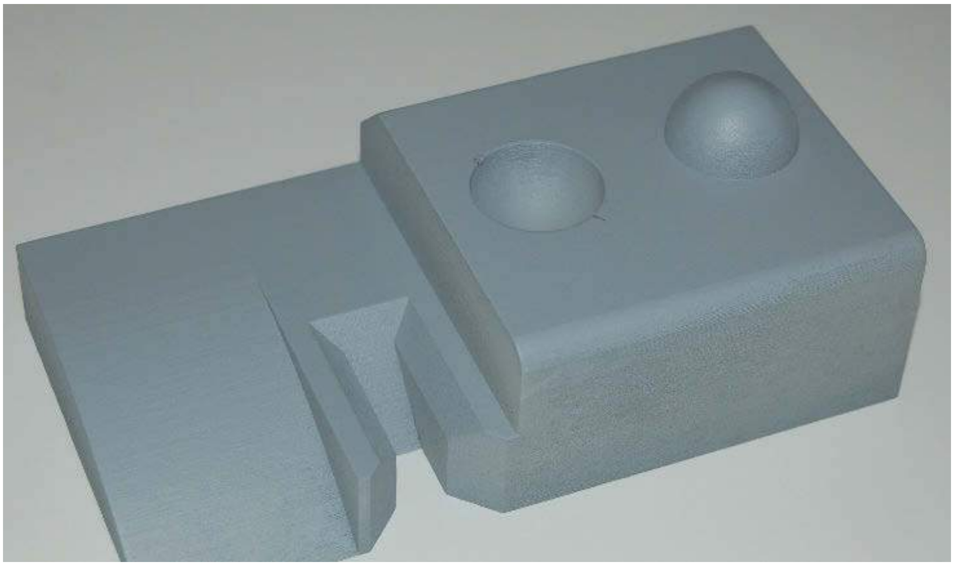

Fig. 11: Object with large radius.

3. CONCLUSIONS

This paper describes a system under development using a single software platform, where GRE software communicates directly with a laser profile scanner and an industrial robot. For each part of the system, we have identified what we believe to be important requirements for the development tool. It should support:

1. Geometric modeling in 3D, wireframe and surfaces.
2. Communication protocols, TCP/IP and RPC.
3. A MESH entity for 2D and 3D measurement data.
4. Fast geometrical search algorithms in 3D for path planning.
5. Bounded planes for collision models.
6. Fast geometrical calculations in 3D for seed selection and region growing.
7. Fitting of planes.

An integrated GRE system is not limited to use 3D point data only. It can also access lower level data directly from the measurement system. In section 2.7 we present an example of how this can be used to speed up computations.

So far, we have only implemented GRE algorithms for planar surfaces but work is under progress to add GRE of curved surfaces as well. Encouraged by the result described in section 2.7 we are now implementing elliptical segmentation of laser profiles. A good fit of an ellipse indicates that the corresponding surface may be cylindrical or spherical. This will not change the requirements listed above but it may add new requirements. It is reasonable to believe for example that support will be needed for new fitting algorithms and also for intersects between curved surfaces. Although it is technically possible to develop a system of this kind using a range of other tools it is our opinion that a single tool on the right level of abstraction is always a better choice even if it means that some of the job has to be spent on low level tool development. The Open Source distribution model is ideally suited for this.

4. ACKNOWLEDGEMENTS

We would like to thank Dr. Sören Larsson, who has contributed to earlier work in this project for helpful discussions. The project is founded by the Department of Technology, Örebro University, Sweden and the Libyan Higher Education Ministry, Libya.

5. REFERENCES

[1] Benkö, P.; Vàrady, T.: Segmentation Methods for Smooth Point Regions of Conventional Engineering Objects, Computer Aided Geometric Design, 36(6), 2004, 511-523.

[2] Besl, P. J.; Jain, R. C.: Segmentation through Variable-Order Surface Fitting, IEEE Transactions on Pattern Analysis and Machine Intelligence, 10(2), 1988, 167-192.

[3] Chan, V.; Bradley, C.; Vickers, G.: A Multi-Sensor Approach for Rapid Digitization and Data Segmentation in Reverse Engineering, 122(4), 2000, 725-733.

[4] Chen, Y. H.; Liu, C. Y.: Robust Segmentation of CMM Data Based on NURBS, International Journal of advanced manufacturing technology, 13(8), 1997, 530-534.

[5] Gallieri, M.; Fasano, A.; Impoco, G.; Cignoni, P.; Scopigno, G.; Parrini, G.: RoboScan: an Automatic System for Accurate and Unattended 3D Scanning, Proceedings of the 2nd Symposium on 3D Data Processing, Visualization, and Transmission, 2004, 805-812.

[6] Hoschek, J.; Dietz, U.; Wilke, W.: A geometric concept of reverse engineering of shape: approximation and feature lines, Proceedings of the international conference on Mathematical methods for curves and surfaces II Lillehammer, 1997, 253-262.

[7] Kjellander, J. A. P.: Interactive Generic Product Modeling Based on a Procedural Language, Technical Report, Örebro University, Department of Technology, 2001.

[8] Larsson, S.; Kjellander, J. A. P.: An Industrial Robot and a Laser Scanner as a Flexible Solution towards an Automatic System for Reverse Engineering of Unknown Objects, Proceedings of ESDA04 - 2004 7th Biennial Conference on Engineering Systems Design and Analysis, 2004.

[9] Larsson, S.; Kjellander, J. A. P.: Motion Control and Data Capturing for Laser Scanning with an Industrial Robot, Robotics and Autonomous Systems, 54(6), 2006, 419-512.

[10] Larsson, S.; Kjellander, J. A. P.: Path Planning for Laser Scanning With an Industrial Robot, Robotics and Autonomous Systems, 56(7), 2008, 615-624.
[11] Lee, K. H.; Park, H.; Son, S.: A Framework for Laser Scan Planning of Freeform Surfaces, The International Journal of Advanced Manufacturing Technology, 17(3), 2001, 171-180.
[12] Lengyel., E.: Mathematics for 3D Game programming and Computer Graphics, Chalers River Media, Hingham, Massachusetts, 2002.
[13] Milroy, M.; Bradley, C.; Vickers, G.: Automated Laser Scanning Based on Orthogonal Cross Sections, Machine Vision and Applications, 9(3), 1996, 106-118.
[14] Rahayem, M.; Kjellander, J. A. P.; Larsson, S.: Accuracy analysis of a 3D Measurement System Based on a Laser Profile Scanner Mounted on an Industrial Robot with a Turntable, Proceedings of ETFA'07 12th IEEE conference on Emerging Technologies and Factory Automation, 2007, 880-883.
[15] Rahayem, M.; Kjellander, J. A. P.; Larsson, S.: Geometric Reverse Engineering Using a Laser Profile Scanner Mounted on an Industrial Robot, Proceedings of the 6th International Conference of DAAAM Baltic on Industrial Engineering, 2008, 147-152.
[16] Roth, G.; Wibowoo, E.: An Efficient Volumetric Method for Building Closed Triangular Meshes from 3-D Image and Point Data, Proceedings of the conference on Graphics interface, 1997, 173-180.
[17] Sacchi, R.; Poliakoff, J. F.; Thomas P.D.: Curvature Estimation for Segmentation of Triangulated Surfaces, Proceedings of IEEE conference on 3-D Digital imaging and Modeling, 1999, 536-543.
[18] Schroeder, W. J.; Zarge, J. A.: Decimation of triangle meshes, Proceedings of the 19th annual conference on Computer graphics and interactive techniques, 1992, 65-70.
[19] Scott, W. R.; Roth, G.; Rivest, J.: View Planning for Automated Three-Dimensional Object Reconstruction and Inspection, ACM Computing Surveys, 35(1), 2003, 64-96.
[20] Shamir, A.: A Survey on Mesh Segmentation Techniques, Computer Graphics Forum, 27(6), 2007, 1539-1556.
[21] Son, S.; Park, H.; Lee, K. H.: Automated laser scanning system for reverse engineering and inspection, International Journal of Machine Tools and Manufacture, 42(8), 2002, 889-897.
[22] Turk, G.; Levoy, M.: Zippered polygon meshes from range images, Proceedings of the 21st annual conference on Computer graphics and interactive techniques, 1994, 311-318.
[23] Pito, R.; Bajcsy R.: A solution to the Next Best View Problem for Automated CAD Model Acquisition of free-Form Objects Using Range Cameras, Proceedings of the SPIE Symposium on Intelligent Systems and Advanced Manufacturing, 1995.
[24] Vàrady, T.; Marin, R. R.; Cox J.: Reverse Engineering of Geometric Models-An introduction, Computer-Aided Design, 29(4), 1997, 255-268.
[25] Varkon Homepage, http://varkon.sourceforge.net .
[26] Woo, H.; Kang, E.; Wang, S.; Lee, K. H.: A New Segmentation Method for Point Cloud Data, International Journal of machine Tools & Manufacture, 42(2), 2002, 167-178.
[27] Kjellander, J.A.P.; Rahayem M: Planar segmentation of data from a laser profile scanner mounted on an industrial robot, International Journal of Advanced Manufacturing Technology, Springer 2009, (DOI 10.1007/s00170-009-1951-9).

Appendix:

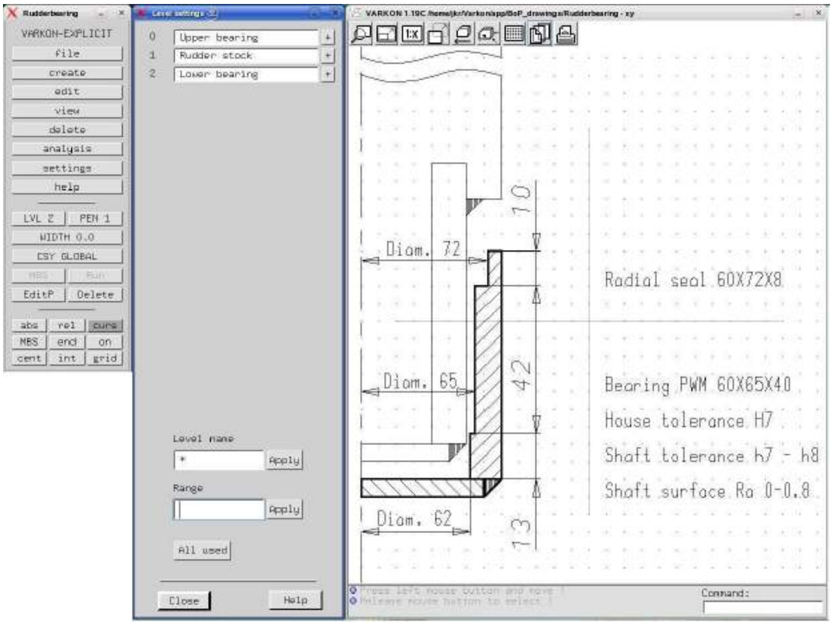

Fig. 12: The Varkon system used in a 2D drafting application.

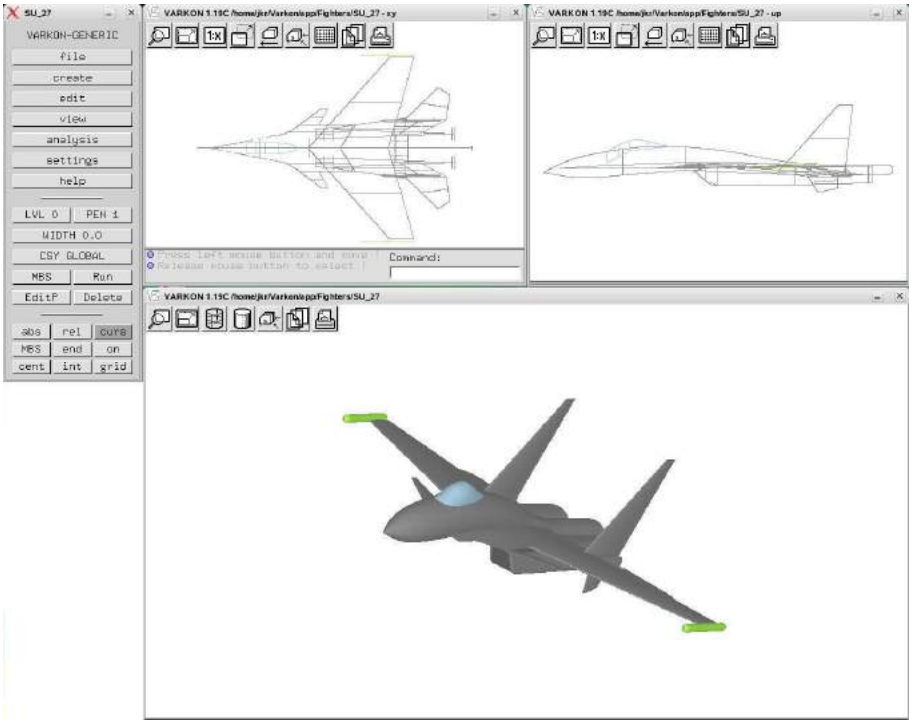

Fig. 13: The Varkon used in a surface modeling application.