CAD Model Building from Multiple Range Images

Gerhard Roth and Pierre Boulanger

Institute for Information Technology
Building M 50, Montreal Road
National Research Council of Canada
Ottawa, Canada K1A 0R6

Abstract

Reverse engineering is a process by which a geometric model is created from sensor data. In this paper we show how to create a geometric model that can be read into a Computer Aided Design system using as input multiple range images of an object. Our reverse engineering method combines a number of new algorithms in a novel way to create the parametric patches that make up the model. The resulting system greatly accelerates the model building process, and makes it possible to guarantee that the model has a specified accuracy relative to the original range data.

Keywords: CAD-Based Vision, Systems and Applications, Reverse Engineering, Model Building, Virtual Reality

1 Introduction

A growing area of research is in the building of geometric models from range data. Such data is obtained by laser rangefinders which give a detailed point by point description of the surface of an object [1, 2]. However, such a surface description consists of thousands of individual points, and is not very useful. What is needed is to create a more compact geometric description from these points, one that can be read into a Computer Aided Design (CAD) system.

In this paper we describe a method of creating CAD models from range data. The input to the model building process is a set of registered range images that cover the entire surface of the object. We have two ways to process such data. The first method is to keep the image ordering and process each image independently. The second method is to discard the image ordering, and group all the data together as a set of 3D data points. The advantage of this format, which we call cloud data, is its generality. While most range sensors provide data in image format, some do not [3]. Point clouds are the standard format for objects that are scanned using a Co-ordinate Measuring Machine (CMM) with a point probe. In industrial practice, range images are often converted to point cloud format to be compatible with CMM data.

Whether we process cloud or image data our approach is to iteratively create parametric patches and add them to the current geometric model. When there is no 3D data remaining, then the model is complete. Working from the cloud data is effective for standard surfaces like planes, spheres and quadrics. However, for free-from surfaces, working from the image data is more effective. When using images we require that each surface patch be entirely visible in a single range image. If this is not the case, we create an artificial image, which we call a pseudo-image, which satisfies this condition. These pseudo-images are created by a projection operation.

The final result of the model building process is a set of trimmed parametric patches [4] which cover the entire surface of the object. The model can be written out as a set of trimmed B-Splines in the IGES file format [5], which can be read by a wide variety of CAD systems. The process is demonstrated from start to finish by a number of examples. A block diagram of our approach is shown in Figure 1.

2 Related Research

Building geometric models from sensor data has been identified as one of the most important problems in computer vision, whose solution would have significant industrial impact [6]. While there are a number of successful methods for building triangular meshes from multi-image range data [7, 8, 9, 10, 11, 12], this is not the case for building CAD models from such data.

There are commercial systems available that claim to be able to build CAD models from range data [13]. However, the level of automation in these systems is very low. It is necessary to trace the

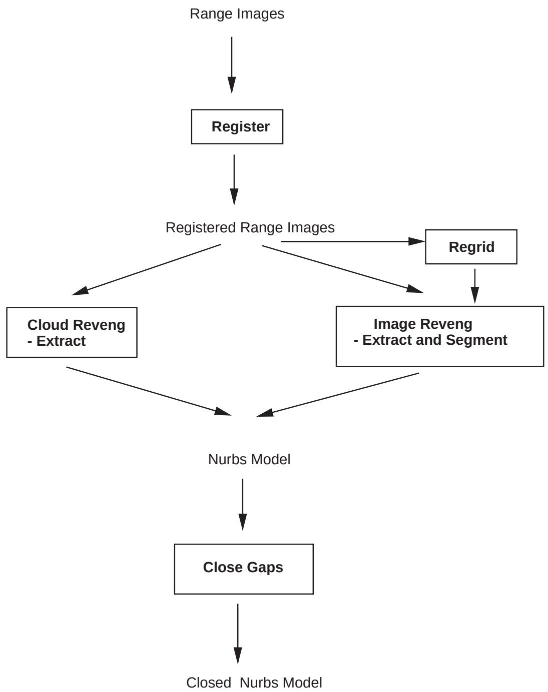


Figure 1: Overview of our reverse engineering approach.

boundaries of each parametric patch by hand, which is very time consuming and error prone. It can easily take days to create a geometric model. The goal of our research is to decrease the time required to build such models from range data by at least an order of magnitude.

There have been a number of research efforts which have attempted to achieve this goal. However, so far none have been entirely successful. Some work discusses only regridding and fitting [14]. While these are important problems, they do not actually address the problem of how to build CAD models of complex parts which contain more than one parametric surface. Other research has concentrated on representations such as superquadrics for model building [15, 16]. While useful in robotics, a superquadric representation does not describe objects accurately enough for a CAD system. Other work combines both registration and model building as one process [17], but handles only planar patches. Our method is able to quickly make models of complex objects with curved surfaces, along with those objects that contain simpler predefined surfaces, such as planes, spheres and quadrics. However, we assume that the registration process which transforms all the range images in a common coordinate frame has been completed before model building begins. This is reasonable given that there already exist a number of successful multi-image registration algorithms [18, 19].

One attempt at automating the model building process creates quadric surface models from Co-Ordinate Measuring Machine (CMM) data [20]. However, in this work the assumption is made that the topological structure of the model can be obtained from a pre-existing CAD description, which is very restrictive. There has been recent work in creating CAD models of complex objects from range data [21, 19]. However these models consist of only a single parametric patch. While this may be practical for certain symmetrical objects, for most industrial objects it will be necessary to create models that have many patches.

Our work is unique in a number of ways. We are able to use both cloud and image data as input to the model building process. Since there are usually between five and twenty-five images, we specifically address the issue of how to deal with the redundancy in the data. The models we create contain not just one, but many parametric surface patches. We also guarantee that the final model fits all the 3D data points within a given tolerance. Finally, our model building approach, while not completely automated, can create models in a time frame that is an order of magnitude faster than any commercial system of which we are aware of.

3 The Model Building Process

The goal of the model building process is to take a set of registered range images of the entire surface of the object, and from these images produce a corresponding set of parametric surface patches. Each range image is a dense sampling of the 3D geometry of the surface from a particular viewpoint.

3.1 Dealing with Data Redundancy

Any model building system that uses multiple range images as input must deal with the problem of data redundancy. There are parts of any object that will be visible in more than one range image. If these images are processed independently, the result would be a geometric model with a number of overlapping parametric patches. The question is how to avoid such an overlap. There are a number of possible approaches. When the goal is to create a triangular mesh model there are algorithms that can deal with this redundancy [7, 8, 9, 10, 11, 12]. However, in our application the goal is to create a geometric model that consists of a number of parametric patches.

The first way we deal with this redundancy is to ignore the image structure of the range data. This means taking all the range data points and grouping them together as a single cloud of disconnected data points. This cloud of points must then be subdivided into patches. This method of operation is currently used commercially [13]. In any local area of the cloud the data points may come from many different range images. Therefore only a single parametric patch will be fitted to a part of the cloud, so the resulting geometric model will not have any redundant model patches.

However, there are disadvantages to using cloud data. When all the data is grouped together as cloud of points the original point ordering in the individual range images is discarded. This makes it more difficult to automate the model building process because we no longer operate on images, but only deal with points in space. The reason is that to create a model consisting of parametric patches it is necessary to parameterize each data point. While it is clear how to do this for image data, it is not obvious how to accomplish this for cloud data.

We are able to process cloud data when creating standard surfaces such as planes, spheres and cylinders. Our approach is to find the implicit equations of these surfaces directly from the cloud data, and then to convert these implicit equations to their equivalent parametric form. This is feasible because standard analytic surfaces such as spheres and cylinders have a direct equivalent in parametric form [22, 23]. However, this approach is not practical for objects that are free-from, such as the human face, that can not be easily described by such standard surfaces.

For free-form objects we operate on individual range images. We create a parametric patch in a single range image using methods which will be described later in the paper. Once this is done, we go to all the other range images and remove the points in these images that also belong to this patch. This means that no two patches will covers the same surface area, which effectively deals with the problem of data redundancy.

The obvious question is: what happens when the entire patch boundary is not visible in a single image? This is a problem for cylindrical and spherical objects, but can also occur in other cases. Clearly we can not always guarantee that all the natural boundaries of a patch will be visible in a single range image. Our solution to this problem is to create a new image in which the entire patch is visible. We will call such an image a pseudo-image, since it was not actually taken by the rangefinder, but was created artificially. We create such a pseudo-image by interpolation using the range data from all the other images. We can create as many pseudo-images as are necessary. In practice, we find that at most half a dozen pseudo-images are required.

The process of creating a pseudo-images will be described later in the paper. For now, we will simply assume that when using images, that there is a single image (actual or pseudo) that contains the desired patch. If we are operating on cloud data, then this is not an issue. In the next section we will discuss how we actually build a geometric model from cloud data or image data. The key parameter in our approach is the maximum absolute deviation.

3.2 The Maximum Absolute Deviation

Since there will always be some approximation error, the final geometric model has a certain accuracy relative to the range data. In practice, the user usually has some a-priori upper bound on the allowable error. We call this upper bound the Maximum Absolute Deviation (MAD). For example, if the MAD is set to .1 mm, then the goal is to create a model in which every range data point is at most a distance of at .1 mm from a model patch.

On the average each data point will have a smaller error of fit than this upper bound. However, no data point should have an error of fit greater than this upper bound. In general, the lower the MAD value, the larger the number of patches that are required. The lowest possible MAD value would be at the noise level of the sensor data. The MAD value is the most important parameter in the model building process.

3.3 Finding Analytic Surfaces

The type of analytic surfaces we consider are the standard surfaces; planes, spheres and quadrics (typically cones or cylinders). We use a semi-automatic approach to find such surfaces. The user manually selects the type of patch along with a superset of range points that contain the desired patch. This set of points is interactively selected by the user. It should be noted that this selection process can be done on both images, and cloud data. Then we automatically find the patch equation and the trim curves using an extraction algorithm [24, 25, 26]. Extraction is a generalization of fitting. In a fitting algorithm we are given a particular type of surface and a set of points that belong to it. The goal of a fitting algorithm is to find the surface which most closely approximates these points. An extraction algorithm is similar to a fitting algorithm in that one of the inputs is the type of curve or surface. However, the assumption that all the points actually belong to the curve or surface is relaxed. In extraction, the assumption is that only a majority of the data points belong to the specified curve or surface. The surface returned by the extraction algorithm is the one that has the largest number of data points that are within the MAD distance of that surface.

This approach allows the user considerable flexibility, since for standard surfaces it is easy to manually choose a set of points which constitute a superset of the desired surface. The algorithm returns the subset of points which actually belong to the surface, along with the implicit equation of that surface.

Now the problem which we must now address is how to convert this implicit equation into parametric form. It is clear that our final model must consist of a set of parametric surfaces, and not implicit surfaces. This is essential because parametric surfaces, especially Nurbs [27], are commonly used in CAD systems. Our approach, which we believe is novel, is to first find the analytic surfaces, and then convert them to their exact parametric representation. It is known that all planes, spheres and quadrics can be converted to Nurbs form with no error in approximation [22, 23].

Once this conversion to parametric form is complete then we have produced a parameterization of this surface which is in some sense optimal. Using this parameterization we compute the parameter value of each data point on the surface. Then we use a simple boundary following algorithm to find the trim curve of the surface patch in parameter space. The result is a trimmed parametric surface that fits the original plane, sphere or quadric.

3.4 Finding Free-Form Surfaces

For free-form objects, the situation is different, since such standard surfaces do not describe the object effectively. In this case we work directly with parametric surface patches. There are two pieces of information associated with a parametric surface patch. The first is the degree of the patch and its type (Bézier, B-Spline, Nurbs, etc). The second is the trim curves which define the outside boundary of the patch, along with any holes in the patch. In the current commercial reverse engineering software, both of these parameters are set manually. The problem with the completely manual approach is that finding the trim curves for each patch is a very time consuming and error prone process.

In the previous section we described a semi-automatic approach to finding analytic surfaces. It is semi-automatic because the surface type is chosen manually, and then an extraction algorithm is used to automatically find the trim curves. Using this approach we can find standard analytic surfaces in both cloud and image data. For free-form surfaces, we can not work directly on cloud data, but instead we process individual range images. We use a segmentation algorithm to automatically find all the patches and trim curves in a single range image.

This segmentation algorithm groups surface patches optimally into a hierarchical structure [28]. The algorithm is divided into four parts. First, an initial partition of the range image into regions, following a first-order Bézier model, is performed using a robust fitting algorithm constrained by the position of depth and orientation discontinuities. Second, an optimal region growing algorithm based on a new Bayesian decision criteria is computed. Third, generalization to a higher-order surface model is performed based on a statistical decision method. The final result is a segmentation of a given range image into patches at the required MAD value. From this image segmentation we manually choose the individual parametric patches that are to be part of the final CAD model. This allows us to correct any possible errors of the segmentation algorithm.

In this segmentation algorithm, as in the extraction approach, the most important parameter is the MAD deviation. The MAD value represents the the largest acceptable error between the model and the underlying range data. For example, if the MAD is set to .1 mm, then every range data point must be at a distance of at most .1 mm from a model patch. Thus the MAD value represents the desired accuracy of fit for the final CAD model. The segmentation algorithm has this MAD value as its operating parameter and it produces a set of patches that are guaranteed to fit the image data within this tolerance [28].

3.5 Iterating the Patch Creation Process

Once a patch has been accepted the 3D points in all the other range images that are within the MAD distance of this patch are marked. The implication is that these marked points have been processed, and need no longer be considered. This is how our model building process deals with the problem of redundancy for multiple-image range data. For the cloud data, redundancy is dealt with directly because the cloud points belonging to a patch may have come from many different range images.

To build the entire model we find patches in the cloud data or the range images iteratively until all the range data has been processed, that is each data point is associated with a parametric patch. At this point the CAD model is complete, since every range data point has been accounted for. The CAD model also fits the range data with an error that is less than or equal to the MAD value.

3.6 Model Building Example

To make the process of model building clear we will use a number of examples. The first example is taken from the Intelligent Manufacturing Systems (IMS) test case on rapid prototyping [29]. In this project a number of tools for rapid phototyping in manufacturing were investigated, including reverse engineering. To facilitate this investigation a mechanical test part was created which had a reasonably complex geometry. We obtained different range images of this test part. A total of ten images were used to create the CAD model; two of them were created by projection from the original data in a process that will now be described.

It is clear that there will be situations where the trim curves of a patch are not completely visible in a single range image. This will occur for surfaces such as spheres and cylinders, that are not completely covered in a single range image because of the way the images are acquired. When this happens, we create a number of auxiliary images, called pseudo-images. Such images are created from the original range images by a projection operation. The most obvious type of projection is a planar projection, but it is also possible to perform a cylindrical or spherical projection. The projection operation is done in a separate program which takes the range data from all the different images, and the projection parameters, i.e. the viewing direction and the projection type, and creates a pseudo-image. Such an image looks similar to the one taken by the actual sensor, but is called a pseudo-image because it has not been taken by the rangefinder, but was created by projection. This pseudo-image creation process was used to deal with the small cylinder on the corner of the IMS part.

We will now show how to find a planar patch on the top of the IMS part using the semi-automatic extraction algorithm on both image and cloud data. In Part (a) of Figure 2 we hand-selected a shaded part of the image that contains this patch, along with some other range points. In Part (c) we show a similar selection, but as a subset of the data cloud points. We specify to the extraction algorithm that we are searching for a planar patch with an MAD value of 0.1 mm. With this input the extraction results are shown in Part (b) and (d) of the same figure.

The trim points of this patch is found automatically by a simple boundary following algorithm. Then the planar equation is converted to its equivalent Nurbs form, and the trim points are converted to a parametric trim curve. The result is a parametric surface patch that can be output in IGES format. The Nurbs patch contains a defining equation along with the trim curves, and is a much more compact data representation than the original points.

In Figure 3 we show the result of running the segmentation algorithm on a single range image of the IMS part in order to find all the patches in that image. The required MAD value was 1.5 mm, and the result is a set of first and second order Bézier patches. Using this segmentation map a user could choose which of the individual patches should be part of the final CAD model.

Extraction and segmentation are complementary tools. The extraction algorithm works for both cloud and image data, and tends to perform better than segmentation for standard analytic surfaces. However, it is iterative, and semi-automatic. The user must specify the patch type, and select a superset of the patch points as input. The result is the equation of the parametric patch, along with the trim curves. Segmentation, on the other hand, works only on range images and not cloud data, but is completely automatic. It produces better results than extraction for free-form surfaces, but sometimes over or under segments standard analytic surfaces.

The model building process we have described was used to build a CAD model of the IMS test part from ten registered range images. Currently our model contains about 100 patches, and is shown in Figure 4. This figure is a 3D rendering of a true three-dimensional CAD model, and each of the parametric patches is shaded for ease of viewing. Even though this particular model is not complete, it does cover about eighty percent of the object's surface. We have not completed the model because it would require more range images be taken with our sensor.

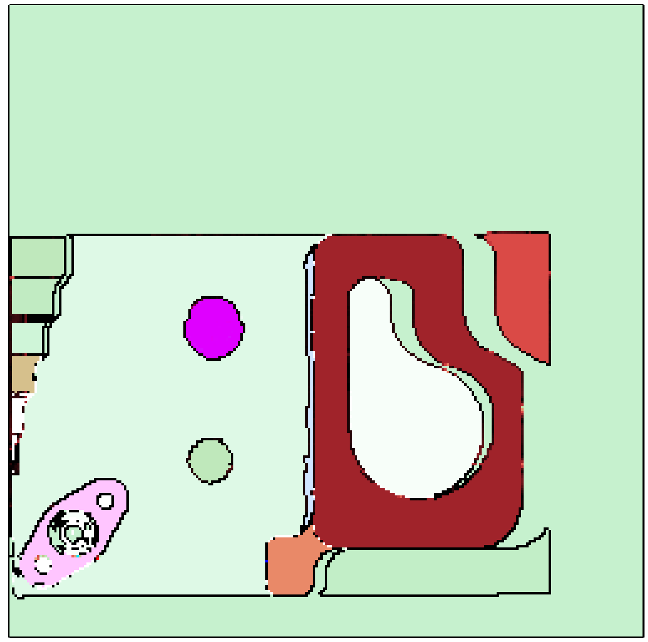


Figure 3: The label map of the range image produced by the segmentation algorithm. Each shaded region is a different parametric patch.

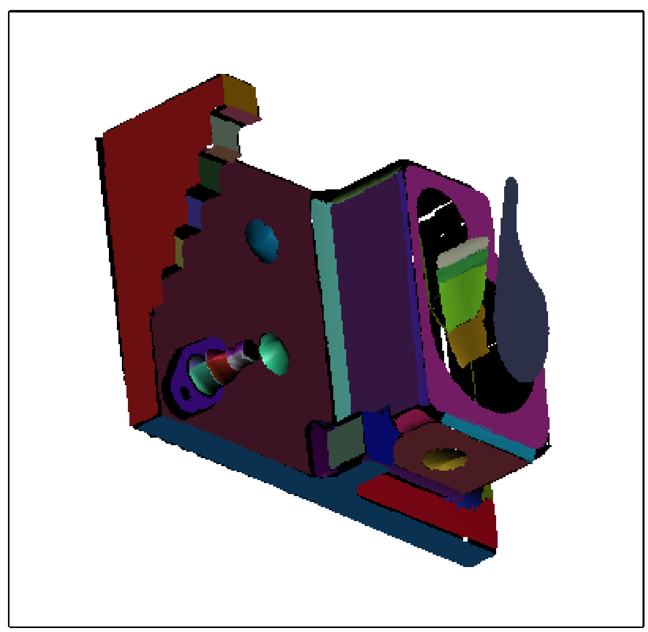


Figure 4: The partial CAD model of the mechanical part built to this point.

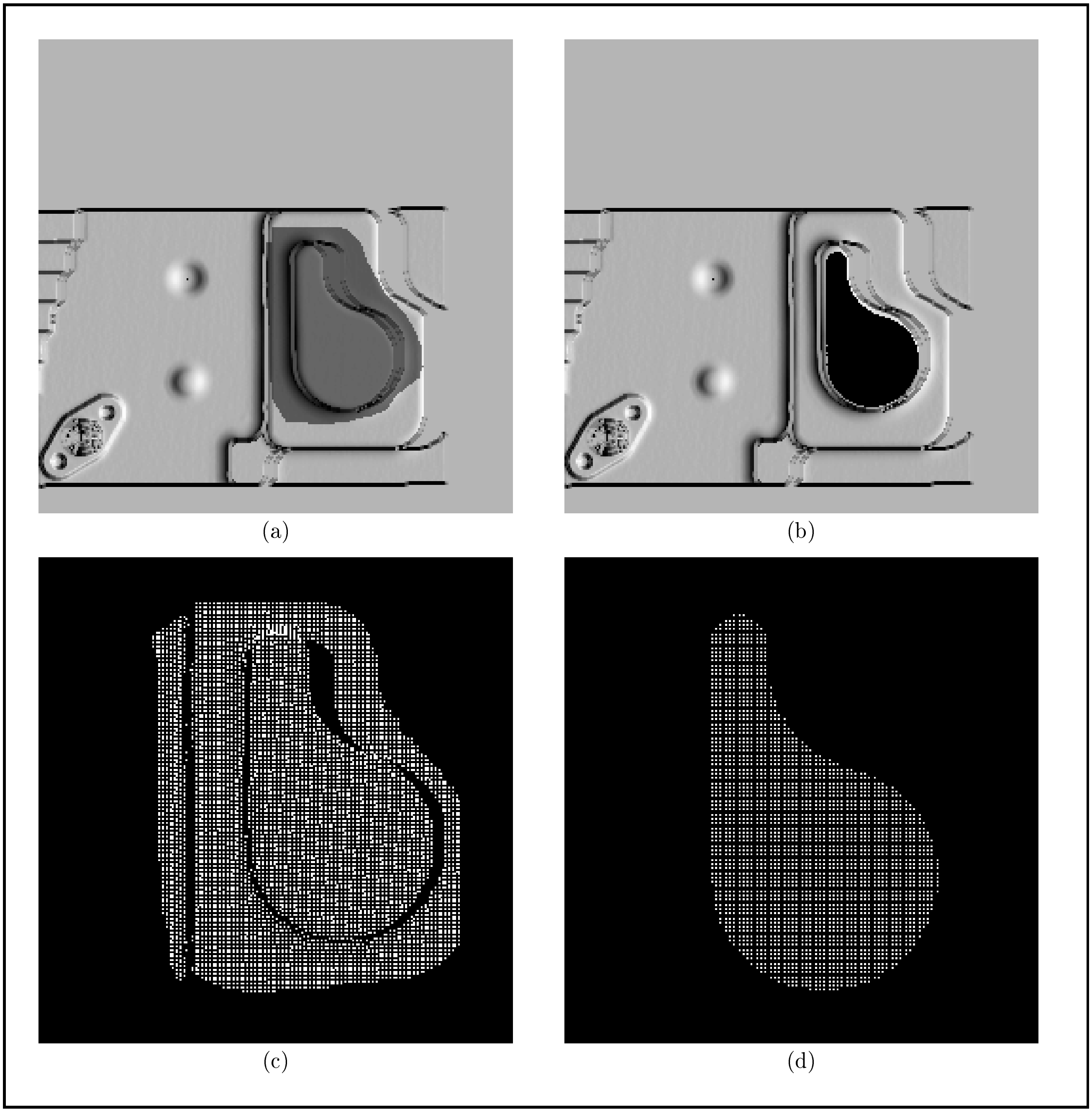


Figure 2: (a) The selected image region is highlighted. (b) The extracted planar points in this region are drawn in black. (c) The selected cloud region is highlighted. (d) The extracted planar points from the cloud are highlighted.

In the next example we will show how to build a CAD model completely from cloud data. The original cloud data is shown in Figure 5, parts (a) and (b). In parts (c) and (d) of the figure we have shown two different views of the final model patches. This model was built by running our extraction algorithm directly on the data cloud. The process of creating the model was very efficient, and took a total of about ten minutes.

In all these examples the CAD models represent a considerable data reduction from the original range data. All the CAD models were created in a span of time ranging from ten minutes to a few hours, which is much faster than can be accomplished with any other model-building system of which we are aware. The MAD values for the examples in this section were set to either 0.1mm or 0.15mm. The patches for all our models have been converted to a trimmed IGES 144 file, and have been successfully read by a number of CAD systems.

4 Conclusions

In this paper we have discussed a method for building a CAD model from multiple range images. We successfully deal with the problem of data redundancy in both image and cloud data. To create the model patches we iteratively use an extraction algorithm or a segmentation algorithm. When operating on range images, when the patch boundaries are not completely visible in any image, we create a pseudo-image by projecting the original range data. In the case of cloud data we operate directly on the cloud to create the patches.

In all the models that have been built so far there are slight gaps at the boundaries of the patches. They exist because range data is a discrete sampling of the surface geometry. Since we compute the trim curves of each patch from the range data, the implication of this discrete sampling is that at the patch boundaries there will sometimes be gaps. We are currently closing these gaps by growing the surface slightly, intersecting the grown surfaces, and removing the overlapping parts [30]. While his method has been successful in a number of cases, it still needs further testing.

References

[1] M. Rioux, "Laser rangefinders based on synchronized scanning," *Applied Optics*, vol. 23, pp. 3837–3844, 1985.

[2] P. Besl, "Active, optical range imaging sensors," *Machine Vision and Applications*, vol. 1, no. 1, pp. 127–152, 1988.

[3] R. Fisher, A. Fitzgibbon, A. Gionis, M. Wright, and D. Egger, "A hand-held optical surface scanner for environment modeling and virtual reality," Tech. Rep. DAI No.778, University of Edinburgh, Dec. 1995.

[4] G. Farin, *Curves and surfaces for computer-aided geometric design*. Boston: Academic Press, 1993.

[5] J. Murray and W. vanRyper, *Encyclopedia of graphics file formats*. O'Reilly, 1996.

[6] W. Grimson and J. Mundy, "Computer vision applications," *Communications of the ACM*, vol. 37, pp. 45–51, Mar. 1994.

[7] M. Soucy and D. Laurendeau, "Multi-resolution surface modelling from multiple-range views," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, (Champlaign, Illinois), pp. 348–353, 1992.

[8] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, "Surface reconstruction from unorganized data points," in *Computer Graphics 26: Siggraph'92 Conference Proceedings*, vol. 26, pp. 71–78, July 1992.

[9] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, "Mesh optimization," in *Computer Graphics: Siggraph '93 Proceedings*, pp. 19–25, 1993.

[10] Y. Chen and G. Medioni, "Surface description of complex objects from multiple range images," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, (Seattle, Washington), June 1994.

[11] G. Roth and E. Wibowo, "A fast algorithm for making mesh models from multi-view range data," in *Proceedings of the DND/CSA Robotics and Knowledge Based Systems Workshop*, (St. Hubert, Quebec), pp. 349–356, Oct. 1995.

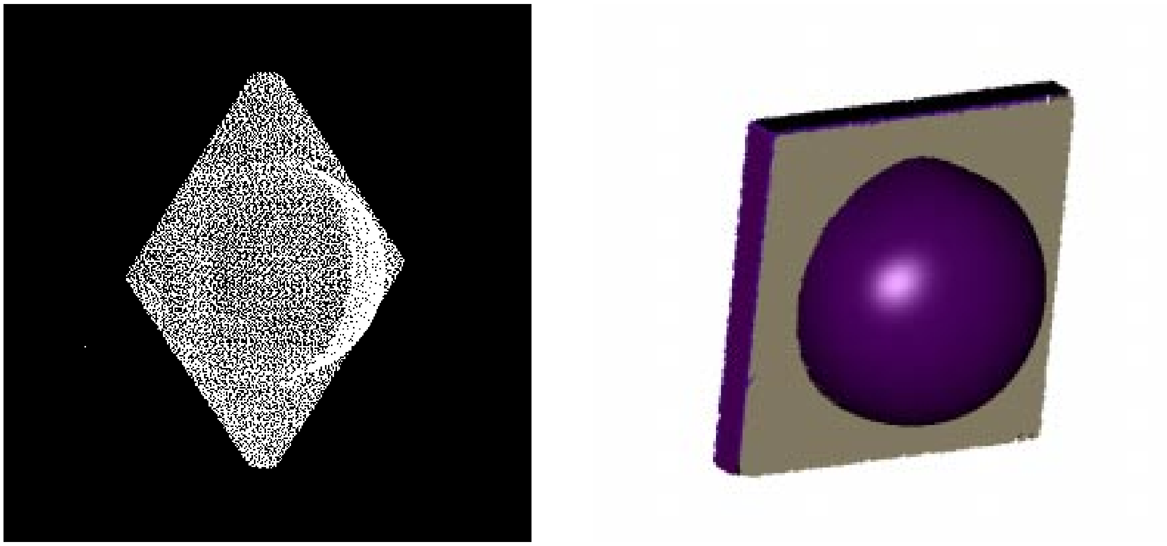
[12] B. Curless and M. Levoy, "A volumetric method for building complex models from range images," in *Computer Graphics: Siggraph '96 Proceedings*, pp. 221–227, 1996.

[13] S. Sinha and P. Seneviratne, "Single valuedness, parameterization, and approximating 3d surfaces using b-splines," in *Geometric Methods in Computer Vision II*, vol. 2031, pp. 193–204, SPIE - The International Society for Optical Engineering, 1993.

[14] C. Bradley and G. W. Vickers, "Automated rapid prototyping utilizing laser scanning and free-form matching," in *CIRP Annals '92, Manufacturing Technology*, pp. 435–439, 1992.

[15] V. Koivunen and R. Bajcsy, "Geometric methods for building cad models from range data," in *Geometric Methods in Computer Vision II*, vol. 2031, pp. 205–216, SPIE - The International Society for Optical Engineering, July 1993.

[16] V. Koivunen and J. Vezien, "Multiple representation approach to geometric model construction from range data," in *Second CAD-Based Vision*



(a) (b)

Figure 5: (a) Cloud data consisting of five registered images of a block with a sphere. (b) The CAD model built from this data shown from two different viewpoints.

Workshop, (Champion, Pennsylvania), pp. 132–139, Feb. 1994.

[17] H. Shum, K. Ikeuchi, and R. Reddy, "Principal component analysis with missing data and its application to object modelling," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, (Seattle, Washington), pp. 560–565, 1994.

[18] H. Gagnon, M. Soucy, R. Bergevin, and D. Laurendeau, "Registration of multiple range views for automatic 3-d model building," in *Proceedings of IEEE Computer Vision and Pattern Recognition Conference*, (Seattle, Washington), pp. 581–586, June 1994.

[19] Y. Chen and G. Medioni, "Object modelling by registraion of multiple range images," *Image and Vision Computing*, vol. 10, pp. 145–155, Apr. 1992.

[20] P. Chivate and A. Jablokow, "Solid-model generation of measured point data," *Computer-Aided Design*, vol. 25, pp. 587–600, Sep. 1993.

[21] Y. F. Wang and J. Wang, "On 3d model construction by fusing heterogeneous sensor data," *CVGIP: Image Understanding*, vol. 60, pp. 210–229, Sep. 1994.

[22] L. Piegel, "The sphere as a rational bezier surface," *Computer Aided Geometric Design*, vol. 3, no. 1, pp. 45–52, 1986.

[23] S. Lodha and J. Warren, "Bezier representation for quadric surface patches," *Computer Aided Design*, vol. 22, p. 574, 579 1990.

[24] G. Roth and M. D. Levine, "Extracting geometric primitives," *Computer Vision, Graphics and Image Processing: Image Understanding*, vol. 58, pp. 1–22, July 1993.

[25] I. Sekita, P. Boulanger, and G. Godin, "Extraction and approximation of range image data using a rational bezier surface," in *Vision Interface '93*, (Toronto, Ontario), June 1993.

[26] G. Roth, "Extraction of parametrically defined geometric primitives," in *Sensor Fusion VI*, vol. 2059, pp. 79–90, SPIE - The International Society for Optical Engineering, 1993.

[27] L. Piegl, "On nurbs: a survey," *IEEE Computer Graphics and Applications*, pp. 55–91, Jan. 1991.

[28] P. Boulanger and G. Godin, "Multiresolution segmentation of range images based on bayesian decision theory," in *Intelligent Robotics and Computer Vison XI: Algorithms, Techniques and Active Vision*, vol. 1825, pp. 16–18, SPIE - The International Society for Optical Engineering, Nov. 1992.

[29] P. Boulanger, G. Roth, and G. Godin, "Applications of 3-d active vision to rapid product development," in *Proceedings of the Intelligent Manufacturing Systems International Conference on Rapid Prototyping*, (Stuttgart, Germany), Feb. 1994.

[30] J. Basran, P. Boulanger, and G. Roth, "Building closed models from range data," in *Proceedings of the 28th ISATA, Dedicated Conference on Rapid Prototyping in the Automotive Industries*, (Stuttgart, Germany), pp. 25–32, Sep. 1995.