Building Models From Sensor Data: An Application Shared by the Computer Vision and the Computer Graphics Community

Roth, Gerhard

National Research Council Canada
Conseil national de recherches Canada

Institute for Information Technology
Institut de Technologie de l'information

NRC·CNRC

*Building Models from Sensor Data, an Application Shared by the Vision and Graphics Community**

G. Roth
August 1999

***published in** Proceedings of the NATO Meeting : Confluence of Computer Vision and Computer Graphics, Ljubljana, Slovenia. August 1999. 9 pages. NRC 43639.

Building models from sensor data: an application shared by the computer vision and the computer graphics community

Gerhard Roth
Visual Information Technology Group
National Research Council of Canada
Ottawa, Canada K1A OR6
Gerhard.Roth@iit.nrc.ca

Abstract

The problem of building virtual models from sensor data increases in importance as powerful graphics rendering hardware becomes widespread. Model building stands at the interface between computer vision and computer graphics, and researchers from both areas have made contributions. We believe that only by a systematic review of the remaining open research question can further progress be made. This paper is an attempt at providing such a review. First, we describe the basic steps in the model building pipeline. Then we discuss the open problems that remain in each step. Finally, we describe some overall research themes that we believe should guide further work in this area.

1 Introduction

In the past the fields of computer graphics and computer vision have been considered to be at opposite ends of the spectrum. The graphics field involves the display of virtual representations, and their manipulation [1]. The vision field involves the processing of sensor data for the purpose of image understanding [2]. A number of recent changes have occurred that make this dichotomy less accurate.

First, the basic technology of 3D graphics display systems as embodied by the applications interfaces like OpenGL, Direct3D, etc. have been widely distributed commercially. The graphics community has been very successful industrially in terms of creating the necessary infrastructure for the display of 3D models. It is now possible to display complex simulations of reality on inexpensive personal computer systems. For this reason the question of realistic content, that is what should be displayed on the graphics hardware, is becoming a more important issue.

Traditionally the graphics community has subscribed to the idea that an animator using a complex piece of animation software should create the 3D content synthetically. This paradigm has been successfully demonstrated in the ever-increasing series of complex digital animations. Yet, even though these animations have been successful commercially, it is clear that in order to create more realistic virtual environments it will be necessary to incorporate sensor data of actual physical environments, not simply to use only synthetic environments.

There are a number of reasons for this. First of all, people are familiar with their current physical world. Even if some of the 3D content is synthetic, they relate better if the virtual environment has a connection to the physical world. This is not in complete contradiction to the traditional graphics path of using only synthetic content. It simply says that to widen the impact of a virtual experience it is desirable to use sensor data of an actual physical environment. Second, it is less costly to make a 3D model of a complex geometric object directly from sensor data than to have an animator construct such a model by hand.

In order to create more realistic virtual environments it will be necessary to use sensor data. This problem of model building has long been a topic of research by some members of the vision community. However, it is a relatively small part of a much broader effort that has been directed towards the general problem of image understanding [3]. In the vision community this is beginning to change, and more vision researchers are working directly on the problem of model building [4, 5]. One of the reasons is that the investment in basic vision research has lead to a broader understanding of images and their geometric relationships [6, 7, 8]. The results are a suite of vastly improved algorithms for dealing with such basic vision problems as correspondence, stereo, and structure from motion. It seems that the capacity to obtain 3D structure from image sequences does exist. While this is a significant step in the model building process, it is, as we shall see, only one step.

There has also been ongoing work in model building by those who use active projection techniques to obtain 3D data. In the past few years these research activities have focused on both the problems of model building and inspection. Model building goes under many names, and is sometimes called reverse engineering [9, 10, 11, 12, 13]. The work done by many groups in this field has resulted in a more complete understanding of the basic steps in the model building sequence.

For the above reasons building 3D models from sensor data is an activity that is pursued by both the graphics and vision community. For example, there has recently been an effort to make models of some of the statues of Michelangelo [14], and to build models of environments [15]; both by researchers from the graphics community. In this paper we discuss the problem of model building and suggest some future research directions. One caveat is that because of space limitations our reference list is far from being complete.

Our thesis is that only by understanding and improving the entire model building process is it possible to make significant further progress. We believe that to accomplish this goal it will be necessary to have more interaction between the computer vision and computer graphics community.

2 Model building pipeline

In this section we will list the steps in the model building process. Regardless of the type of sensor used the model building pipeline proceeds in a number of distinct steps, where the output of one step is the input of the next step. We will describe each of these steps in a quick survey, which is not meant to be exhaustive. For each step we will also discuss some open problems, along with their importance and difficulty.

The input to the model building process is some sensor data, and the output is a geometric model. In practice, 3D triangles are the most commonly used geometric data representation in the graphics world. The current generation of graphics hardware has been optimized to display such textured triangles efficiently. For this reason the output of a model creation process is normally a set of possibly textured 3D triangles.

The model building process consists of the following sequential steps.

1. Calibration: the sensor characteristics and configuration are determined.
2. Acquisition: the sensor is moved to a number of different viewpoints and the data is acquired.
3. Registration: the data from different sensor positions is registered to be in the same co-ordinate frame.
4. Point Creation: a set of 3D data points are created from the sensor data.
5. Model Creation: a geometric model consisting of a number of triangular meshes is created from the 3D data points.
6. Model Compression: this triangular mesh model may be compressed to a more manageable size.
7. Texture creation: if possible 2D textures are mapped onto the 3D triangles of the mesh model.

We will now describe each of these steps in more detail, concentrating on what we think are the open problems that still remain to be solved.

2.1 Calibration

There are many different sensors and sensor geometries used during the data acquisition process. A calibration step is necessary in order to accurately find the sensor parameters. For a single camera the standard calibration parameters are the intrinsic (or internal) parameters, or the extrinsic (or external parameters).

In many model building systems there are often multiple sensors, and even different types of sensors (ie. active sensors and passive sensors). This means that the calibration process is necessary more complex than with a single passive sensor. However, a good calibration is essential if an accurate geometric model is to be produced. There are still open problems in terms of creating simple and efficient calibration processes, but progress has been made [16].

Note also that while traditionally calibration is done once, in a laboratory, it may be necessary to do the calibration on-site. The reason is that the sensors may be dissambled during transit, and only reassambled in in their final configuration at the acquisition site. On-site calibration is an area in which little work has been done. However, there has been considerable progress in self calibration for standard cameras [17, 18], so it may be that research in this area will have further applications to the problem of on-site calibration.

2.2 Acquisition

A sensor must be moved to different locations in order to acquire data. This is currently done manually in a rather ad-hoc process. In certain situations, where a sensor is mounted on a programmable motion device such as a co-ordinate measuring machine (CMM) or robot there is also the added issue of avoiding collisions with obstacles. There has been work done in the automation of the acquisition process [19, 20, 21], but some basic questions remain:

- Can we perform both view planning and obstacle avoidance at the same time? This is important when dealing with sensors that have a very small field of view. They must be close to the object in order to obtain 3D data, but must still avoid collisions.
- Can we integrate knowledge of the sensor accuracy into the planning process?
- For the registration step we would like to maintain a certain minimum overlap in the sensor data. Can we incorporate this goal into a viewpoint-planning algorithm?

2.3 Registration

Here the goal is to place all the sensor data into a common co-ordinate reference frame. This process is currently performed manually by choosing corresponding feature points [22], or by accurate sensor motion devices such as turntables or CMMs. Manual registration of the sensor data is time consuming, and automatic registration using accurate positioning devices is expensive. An alternative is to use the 3D data itself to perform data-based registration. In practice there are two kinds of data-based registration algorithms. Those which refine an already approximately known registration are called pose refinement algorithms. They are usually based on an iterative closest point (ICP) strategy [23, 24]. While these algorithms work, there are still some open questions:

- What is the best way to perform a multi-image ICP, where we must register multiple sets of 3D points at once?
- Assuming that each data point has an uncertainty estimate, we would like these estimates to be used by the registration algorithm. What is the best way of propagating such uncertainty estimates into the registration process?

If there is no prior estimate of the registration available we face the more difficult problem of pose determination. There has been less progress on the problem of data-based pose determination since it is computationally difficult [25, 26, 27]. There are many open questions:

- To what degree can the process of pose determination be automated?
- Which approach to the problem of pose determination is computationally tractable?
- Can the problem of pose determination be solved using only the sensor data itself, or must targets be manually placed to aid in the registration process?

The problem of pose determination is strongly related to the traditional vision problem of finding correspondence. As we have stated, a manual registration process requires that the corresponding points be chosen by the user in the different sensor views [6, 8]. This manual process can take a number of hours for a significant number of images, and therefore needs to be automated. Attempting to automate pose determination is equivalent to attempting to solve the correspondence problem.

2.4 3D Point Creation

Assuming that the sensor data has been acquired, it is then necessary to extract 3D points from this data. In practice, there are two types of sensors used in model building. Active sensors project light onto the object using a source such as a laser beam. There are a number of different technologies for active sensors: time of flight, triangulation and structured light being the most common [28]. For any type of active sensor 3D points are acquired efficiently and reliably by the sensing process.

Passive sensors, which do not project an illumination pattern, rely totally on the texture of the object. Traditionally depth from passive sensors is extracted using stereo algorithms [7]. However, these algorithms assume that the epipolar geometry of the two cameras is known. When a sensor is moved around an object this epipolar geometry is not known beforehand. Finding the epipolar geometry requires that we find correspondences between features in different sensor views so that again the correspondence problem is at the core. There are a number of important issues that need further study.

- When using passive sensors it is necessary to find corresponding points among many different 3D views in order to obtain the epipolar geometry and the 3D data points. Can this correspondence process be efficiently automated?
- Is it necessary to use active sensors to get 3D data, or are passive sensors sufficient? If not, what type of active projection technology should be used?

2.5 Mesh Creation

From the 3D data points a triangular mesh must be created. There are many mesh creation algorithms, which work with different types of 3D data [29, 30, 31, 11]. When very dense 3D data is available the mesh creation process is simplified. This is because the topology of the mesh can be found easily with dense 3D data, but as the data becomes sparser, this is more difficult. Passive sensors tend to produce a much sparser set of 3D data points than active sensors. This implies that mesh creation using data from a passive sensor is likely to be more difficult than with data from an active sensor.

There are still some open problems in mesh creation.

- How dense does the 3D data have to be in order to get good results? At some point the 3D data will not be dense enough to make a good model.
- How can these algorithms handle data with significantly different accuracy. This again requires that these methods incorporate estimates of uncertainly into the mesh creation process.
- How should these algorithms deal with very large amounts of data? This situation occurs when making models of large objects or environments.

2.6 Mesh Compression

Active sensors produce a very dense sampling of the surface geometry. If all of these points are used to create a 3D mesh then the resulting mesh is often very large. For this reason a mesh created from active sensor data needs to be compressed for efficient viewing. This is not difficult to do when the final compressed mesh is at a single resolution. If we wish to display the data at multiple resolutions then we will need a different compression scheme, one based on a continuous compression of the mesh [32].

A multi-resolution, continuous compression scheme is especially useful when a large number of triangles are to be displayed. There are a number of competing continuous compression methods, and little systematic work has been done in terms of comparing them [32, 33, 34].

2.7 Texture Mapping

In order to make realistic models it is desirable to add texture to the 3D mesh triangles. This is normally done by using the data from a set of 2D images [35, 36]. This is a difficult problem, which encompasses a number of issues. First of all, the images from the 2D camera must be registered with the 3D data. This is trivial if the 3D data were created from the same set of 2D images, as is the case when a single passive sensors is used for the entire process. However, if a separate active sensor was used to get the 3D data, then it is necessary that the 2D and 3D sensors data be registered accurately [37].

Before the 2D data is textured mapped onto the 3D triangles the 2D images must be pre-processed. The goal is to remove the effect of the local lighting, and also to remove any artifacts produced by surface specularity. The textures that we map on the geometric model should be as free as possible of shadows, highlights, specularities, and colour distortions. Removing such artifacts is a difficult problem. It requires both a knowledge of the lighting conditions, and the surface characteristics. No general solution has been found, but under specific conditions it has been shown to be possible to remove certain types of specularities and ambient lighting affects [38, 39]. Once pre-processed the 2D images are mapped onto the 3D mesh by a projection process.

3 Research Themes

In the previous sections we described the model building process, and some of the basic open problems in each step of this process. In this section we will discuss the following research themes that we believe are among the most important open problems in model building:

1. Automation of the entire model building pipeline.
2. Constructing models incrementally.
3. The role of active versus passive sensors.
4. Image-based rendering versus model-based rendering.
5. Environment modelling versus object modelling.

3.1 Automation of the entire process

There are available some commercial systems for building geometric models from dense 3D data. For certain applications, such as scanning human bodies the model building process is automated. However, for model building in general one of the problems with current systems is the lack of automation. A number of steps in the 3D model building process are currently very laborious, and require a rather high degree of skill. The goal is to make the model building process more automatic. This way we can decrease the time necessary to build such models, and decrease the required skill level.

Currently the acquisition process and the subsequent registration steps are the most time consuming part of the pipeline. Therefore these steps in the model building pipeline would gain the most from automation. However, automating these two steps is difficult. Planning the acquisition process is equivalent to viewpoint planning. This is a high dimensional search for which no general solutions have been found.

Automating registration is equivalent to solving the correspondence problem. While traditionally this has been considered to be an intractable problem, recent computer vision research gives hope that the correspondence problem can be solved for certain situations. First of all, there has been some success in solving the correspondence problem for 2D images if they are not too far apart in viewpoint [40, 6]. For 3D data we believe that it much easier to automate the correspondence problem than for 2D data [41]. This is because in 3D Euclidean distances are an invariant under a rigid transformation. Finally, faster computers make it more likely that both model planning and correspondence computation can be automated because these problems are computationally difficult.

3.2 Incremental model construction

A second requirement that must be met in order for 3D models to be built efficiently is to make the model building process incremental. Currently, all the data is acquired at once, then it is registered, etc. in a sequential pipeline as we have described. This means that if there are errors in the data, or there is missing data, this will not be realized till late in the process. By this time the acquisition system may be dismantled, which means that collecting more data is impossible.

A better way is to build the models incrementally, that is to perform all the steps in the process but only on a subset of the sensor data. Then by looking at the partial model we get valuable feedback which we can use to adjust the acquisition and building process. We may notice that we need to change some of the parameters of the sensor, or may need to move closer, or to scan some area again.

To incorporate feedback into the process it is necessary to build models incrementally and to save the intermediate results. This is not trivial for some steps in the process such as mesh creation. The reason is that this step requires, for example, that the current mesh model be updated incrementally as new 3D data is acquired while still keeping the old model [11].

3.3 Active versus passive sensors

Active sensors use a light source such as a laser to project texture onto an object. The 3D data is only obtained where this light source strikes the surface of the object [28]. Passive sensors do not use artificial light, but instead extract 3D points using natural texture. Both approaches have advantages and disadvantages.

Since active sensors supply their own illumination they are not affected by the ambient illumination. They can therefore successfully obtain 3D data under a wide variety of ambient lighting conditions. They project their own texture they do not require any texture on the objects being scanned. Active sensors also produce dense 3D data, which we have argued simplifies the mesh creation process.

However, active sensors are significantly more costly than passive sensors. There is also a safety issue with active sensors because the active projection system itself is sometimes powerful enough to harm the human eye (i.e. a strong laser).

Passive systems are generally less expensive than active sensors, and there are no safety issues involved in their use. However, they have all the disadvantages for which active systems have an advantage. They are intolerant to changes in the ambient illumination, they require textures on the objects being scanned, and they produce only sparse 3D data.

We believe that active projection technology will continue to be used in many model building applications. When building geometric models there may or may not be enough texture to compute detailed 3D structure using only passive sensors. This means that we cannot really predict beforehand how well a passive system will work for a given situation. By contrast active sensors produce accurate 3D data for a wide variety of ambient lighting conditions and object texture.

There are still a number of open questions in the use of active sensors. The cost of an active sensor is dependent strongly on the speed of data acquisition. This in turn impacts the density of the data that can be acquired. What density of 3D data is sufficient to make a good model? If we can still create good quality models from sparser 3D data, then this is preferable. Active sensors that produce sparser data will be less expensive, and the data acquisition process will not take as long.

Another question is what active sensor technology is beset suited for a particular application? The major technologies are time of flight, triangulation and structured light. It seems that triangulation technology is very accurate, but it useful only for distances of ten meters or less. Time of flight technology is more expensive, but is useful for longer distances. Structured light systems tend to be less accurate, and produce fewer 3D data points than either time of flight or triangulation systems. However, structured light systems are the least expensive of the three. There has been a systematic survey of active sensors [28], but there has been little experience regarding the merits of different active sensor technology for the specific application of model building.

3.4 Image-based versus model-based technology

Traditionally only model-based technology has been used in rendering virtual worlds. In this approach the goal is to have a geometric model that can be displayed on standardized commercially available rendering hardware. Recently the field of image-based rendering has matured sufficiently to provide some competition to the model-based paradigm. The idea is to not create a geometric model, but instead to use the images directly, and therefore bypass the model creation step [42, 43, 44].

The most common image-based rendering methods use image mosaics. The technology of image mosaics has matured to the point where they can be built easily with passive sensors. Mosaic acquisition, creation and display are possible without having any 3D representation of the object [45, 46]. However, mosaics do not handle viewpoint translation unless it is the case that only a planar surface is being observed.

In order for an image based rendering system to deal with translation it is necessary to have depth data. If scaled depth is available for each 2D image then that image can be rendered from a different viewpoint using image-based rendering [47]. So for image-based rendering systems, other than mosaics, it will still be a requirement that 3D be available. What will not be necessary is the creation of a 3D model from this data. This implies that steps 4, 5 and 6 of the model creation process will be eliminated with an image-based rendering system. While this is advantagous the effectiveness of image based rendering systems relative to traditional 3D graphics systems is not yet clear. They have advantages for rendering very large models, but their practical creation and display has not yet been demonstrated. The requirement that dense depth data be available makes image-based rendering systems, other than mosaics, difficult to implement in practice.

3.5 Environment Modeling versus Object Modeling

In the past there has been a concentration of work on building models of objects. Here an object is loosely defined as a blob that we view from the outside and can walk around. Objects have the following characteristics:

- You can walk around an object, it does not enclose you.
- You can move as close as you want to any part of the object.
- You can often control the lighting conditions around the object.

Recently there has been an increase in interest in building models of environments [36, 35, 15]. For environments the situation is different than for objects:

- The environment encloses you, since you are on the inside.
- You cannot necessarily move as close as you want to certain parts of the environment (i.e. there may be a high ceiling).
- It is difficult to control the lighting for the entire environment.

These differences have significant implications when building models. Basically, the problem of making and rendering object models is much simpler than for environment models for the following reasons:

- The fact that you cannot move as close as you want to a part of the environment means that the sensor data will always be at different resolutions. This is not the case for objects. We usually have the ability to scan an object at a single stand off distance. This means that the sensor data for objects tends to all be at approximately the same resolution.
- There is likely to be much more sensor data for environments than for objects. This is because environments are large and open ended, while object are usually smaller and are closed. It is also more likely that a number of different sensors will be used for creating environment models.
- Models of environments are more likely to require multi-resolution compression and visualization methods due to their large model sizes.
- The accuracy and the quality of the data is likely to be much worse for environments than for objects. This is because the lighting conditions, and the specular characteristics of the environment are much harder to control than is the case for objects.

4 Conclusion

In this short discussion paper we have described the problem of building models from sensor data. We believe that this application is one of the main drivers in an ongoing process that will create a much closer relationship between the fields of computer vision and computer graphics. We have listed what we believe are the basic model building steps, along with the open problems in each step. The graphics community tends to concentrate more efforts on the last steps in this process, and the vision community on the first steps. In the mesh creation step, which is the middle step, there has been an equal amount of work done by both communities.

We believe that to make faster progress there should be more interaction between the graphics and vision research communities. Researchers in image based rendering are clearly dealing with many vision problems, and have initiated a wider dialogue between the two communities. This paper is an attempt to encourage more such interaction on the problem of model building in order to define the open problems and future research directions.

References

[1] J. D. Foley and A. V. Dam, *Fundamentals of Interactive Computer Graphics.* Reading, Mass.: Addison-Wesley, 1982.

[2] D. Ballard and C. Brown, *Computer vision.* Prentice Hall, 1982.

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

[4] O. Faugeras, *Three-dimensional computer vision.* The MIT Press, 1996.

[5] M. Petrov, A. Talapov, and T. Robertson, "Optical 3d digitizers: bringing life to the virtual world," *IEEE Computer Graphics and Applications*, pp. 28–37, May/June 1998.

[6] Z. Zhang, "Determining the epipolar geometry and its uncertainty: a review," *International Journal of Computer Vision*, vol. 27, March 1998.

[7] R. Klette, K. Schluns, and A. Koschan, *Computer Vision: three-dimensional data from images.* Springer, 1996.

[8] P. Torr, *Motion segmentation and outlier detection.* PhD thesis, University of Oxford, 1995.

[9] 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.

[10] R. Pito and R. Bajcsy, "Data acquisition and representation of mechanical parts and interfaces to manufacturing devices," in *International Conference on Recent Advances in 3-D Digital Imaging and Modelling*, (Ottawa, Ontario, Canada), pp. 2–9, 1997.

[11] G. Roth and E. Wibowo, "An efficient volumetric method for building closed triangular meshes from 3-d image and point data," in *Graphics Interface 97*, (Kelowna, BC, Canada), pp. 173–180, May 1997.

[12] A. Marshall and R. Martin, *Computer vision, models and inspection.* World Scientific, 1992.

[13] R. Fisher, A. Fitzgibbon, and D. Eggert, "Extracting surface patches from complete range descriptions," in *International Conference on Recent Advances in 3-D Digital Imaging and Modelling*, (Ottawa, Canada), pp. 148–157, IEEE Press, May 1997.

[14] M. Levoy, "The digital michelangelo project," Tech. Rep. (see http://graphics.stanford.firenze.it/projects/mich/), Stanford University, Computer Graphics Laboratory, 1998.

[15] L. Nyland and D. McAllister, "The impact of dense range data on computer graphics," in *Proceedings of multi-view analysis and modelling workshop*, (Fort Collins), June 1999.

[16] J. Beraldin, F. Blais, J. Cournoyer, M. Rioux, F. Blais, S. El-Hakim, and G. Godin, "Object model creation from multiple range images: acquisition, calibration, model building and verification," in *International Conference on Recent Advances in 3-D Digital Imaging and Modelling*, (Ottawa, Canada), pp. 326–333, 1997.

[17] R. I. Hartley, "Self-calibration of stationary cameras," *International Journal of Computer Vision*, vol. 22, pp. 5–23, February 1997.

[18] M. Pollefeys, R. Koch, and L. V. Gool, "Self calibration and metric reconstruction in spite of varying and unknown internal camera parameters," in *International Conference on Computer Vision*, pp. 90–96, 1998.

[19] J. Maver and R. Bajcsy, "Occlusion as a guide for planning the next best view," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, pp. 417–433, May 1993.

[20] M. Reed, P. Allen, and S. Stamos, "3-d modelling from range imagery: an incremental method with a planning component," in *International Conference on Recent Advances in 3-D Digital Imaging and Modelling*, pp. 76–83, IEEE Press, 1997.

[21] R. Pito and R. Bajcsy, "A solution to the next best view problem for automated cad model acquisition," in *SPIE*, vol. 2596, pp. 78–89, 1995.

[22] E. Gagnon, J.-F. Rivest, M. Greenspan, and N. Burtnyk, "A computer assisted range image registration system for nuclear waste cleanup," in *IEEE Instrumentation and Measurement Conference*, (Brussels, Belgium), pp. 224–229, June 1996.

[23] P. Besl and N. McKay, "A method for registration of 3-d shapes," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 14, pp. 239–256, Feb. 1992.

[24] R. Benjemaa and F. Schmitt, "A solution for the registration of multiple 3d point sets using unit quaternions," in *Computer Vision-ECCV'98*, pp. 34–50, 1998.

[25] R. Bergevin, D. Laurendeau, and D. Poussart, "Registering range views of multipart objects," *Computer Vision and Image Understanding*, vol. 61, pp. 1–16, January 1995.

[26] A. Ashbrook, R. Fisher, C. Robertson, and N. Werghi, "Finding surface correspondence for object recognition using pairwise geometric histograms," in *Computer Vision-ECCV'98*, (Freiburg, Germany), pp. 674–686, June 1998.

[27] C.-S. Cheng, Y.-P. Hung, and J.-B. Chung, "A fast automatic method for registration of partially overlapping range images," in *International Conference on Computer Vision*, (Bombay, India), pp. 242–248, 1998.

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

[29] G. Turk and M. Levoy, "Zippered polygon meshes from range images," in *Computer Graphics (Siggraph '94)*, vol. 26, pp. 311–318, 1994.

[30] 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.

[31] 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.

[32] H. Hoppe, "Progressive meshes," in *Computer Graphics: Siggraph '96 Proceedings*, pp. 225–235, Siggraph 96.

[33] A. Ciampalini, P. Cignoni, C. Montani, and R. Scopigno, "Multiresolution decimation based on global error," *Visual Computer*, vol. 13, pp. 228–246, 1997.

[34] J. Xia, J. El-Sana, and A. Varshney, "Adaptive real-time level-of-detail-based rendering for polygonal models," *IEEE Transactions on Visualization and Graphics*, vol. 3, pp. 171–183, Apr. 1997.

[35] K. Ng, V. Sequeira, S. Butterfield, D. Hogg, and J. Goncalves, "An integrated multi-sensory system for photo-realistic 3d scene reconstruction," in *ISPRS Commission V Symposium: Real-Time Imaging and Dynamic Analysis*, vol. XXXII, pp. 356–363, 1998.

[36] S. El-Hakim, C. Brenner, and G. Roth, "A multi-sensor approach to creating accurate virtual environments," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 53, pp. 379–391, December 1998.

[37] P. Jasiobedski, "Fusing and guiding range measurements with colour video images," in *Proceedinsg International Conference on Recent Advances in 3-D Digital Imaging and Modelling*, (Ottawa, Ontario), pp. 339–347, IEEE Computer Society Press, 1997.

[38] R. Baribeau, M. Rioux, and G. Godin, "Color reflectance modelling using a polychromatic laser sensor," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 263–269, 1992.

[39] Y. Sato and K. Ikeuchi, "Reflectance analysis for 3d computer graphics model generation," *Graphical models and image processing*, vol. 58, pp. 437–451, September 1996.

[40] P. Torr and D. Murray, "Outlier detection and motion segmentation," in *Sensor Fusion VI*, vol. 2059, pp. 432–443, 1993.

[41] G. Roth, "An automatic registration algorithm for two overlapping range images," in *The 8th International Conference on Computer Analysis of Images and Patterns*, vol. In press, (Ljubljana, Slovena), 1999.

[42] M. Hirose, "Image-based virtual world generation," *IEEE Multi-Media*, pp. 27–33, Jan. 1997.

[43] Y. Chen and G. Medioni, "Description of complex objects from multiple range images using an inflating ballloon model," *Computer Vision and Image Understanding*, vol. 61, pp. 325–334, May 1995.

[44] S. Kang, "A survey of image-based rendering techniques," Tech. Rep. CRL 97/4, Digital Equipment Corporation, Cambridge Research Lab., Cambridge, MA, 1997.

[45] H.-Y. Shum and R. Szelski, "Panoramic image mosaics," Tech. Rep. MSR-TR-97-23, Microsoft Research, Redmond, WA, 1997.

[46] R. Szeliski, "Video mosaics for virtual environments," *IEEE Computer Graphics and Applications*, pp. 22–30, March 1996.

[47] L. McMillan, *An image-based approach to three-dimensional computer graphics.* PhD thesis, Univ. of North Carolina at Chapel Hill, 1997.