Interactive 3D Model Acquisition and Registration

Yushuang Liu and Wolfgang Heidrich
The University of British Columbia

Abstract

The acquisition of geometric information from real-world objects has become a major way of modeling complex scenes and environments. Unfortunately, most optical methods for geometric model acquisition require the combination of partial information from different view points in order to obtain a single, coherent model. This, in turn, requires the registration of partial models into a common coordinate frame, a process that is usually done offline. As a consequence, holes due to undersampling and missing information often cannot be detected until after the registration.

In this paper we introduce a fast, hardware-accelerated method for registering a new view to an existing partial geometric model in a volumetric representation. This method currently performs roughly one registration every second, and is therefore fast enough for on-the-fly evaluation by the user. Given more time, the same method is also capable of producing full geometric models at very high quality.

1. Introduction

Over the past few years, geometry acquisition has become an invaluable tool for modeling complex environments. Acquisition methods that are based on optics (as opposed to contact measurement) usually need to merge partial geometric information that is obtained with the sensor in different positions relative to the object. This can be achieved either through a precisely manufactured, calibrated mechanical setup that moves the sensor and the object relative to each other (for example the commercial 3D scanners from Cyberware [3]), or via registration of largely uncalibrated setups (such as handheld devices) after the fact. It should be noted that the calibrated motion platforms usually only have limited possibilities to move. As a consequence it may be necessary to combine multiple scans that are not initially aligned in order to cover the whole surface of the object to be scanned [7]. This, again, is a registration problem from uncalibrated initial data.

Unfortunately, the registration is usually a slow offline process that is performed after all partial scans are acquired. This practice, however, is problematic; holes and other problems with missing information are only revealed after the registration phase. As a consequence it has in the past often turned out to be necessary to go back to the scanning phase and iterate until sufficient object coverage has been achieved [7]. In many cases this process may not be feasible, for example, because the object to be scanned is only available for a limited amount of time, or because it is in a remote place. Some researchers try to improve the level of control and automation for 3D registration process by view planning or by conducting an exhaustive search of all views [9, 16, 8]. Those approaches either require intensive computation work which is not feasible for real-time systems or need specific hardware to control the scanner positions. It is therefore highly desirable to be able to *quickly* quickly evaluate the quality of the acquired data during the scanning process. The registration for this evaluation does have to be good but not perfect - the alignment can later be refined in a post-processing step where necessary.

In this paper we propose an efficient, hardware-accelerated algorithm for registering partial 3D volumetric models with each other. The models can originate from any source, although in our system we use disparity images acquired with a commercial trinocular system [13]. This has the advantage of providing color information in addition to geometry. Our method can register partial geometric information at a rate of 1-3 frames per second, depending on the resolution of the disparity images. This allows the user to verify the model and detect holes during the acquisition process. If desired, the same algorithm can later be used offline to produce highly accurate registration. At that stage it is also possible to eliminate bogus information with a voting scheme.

The remainder of this paper is organized as follows: Section 2 briefly reviews the relevant previous work. In Section 3 we give an overview of the components of our system. In Sections 4 and 5 we then discuss the stereo acquisition process and the on-the-fly registration process, respectively. Section 6 deals with improvements for the high-quality offline process, before we conclude in Sections 7 and 8 by presenting results and evaluating the performance of our method.

2. Related Work

The literature provides an abundance of different methods for obtaining 3D geometry from real world objects. Rocchini et al [14] provide a very nice taxonomy of those. In short: 3D model acquisition methods can be grouped into contact vs. non-contact methods, and the non-contact methods can themselves be subdivided into methods based on transmission (like computed tomography) or reflection, using sound waves (like ultrasound) or electromagnetic waves that are either in the visible spectrum or not.

Of these categories, reflective approaches based on visible light are of particular interest, since relatively inexpensive sources (i.e. lights) and sensors (i.e. cameras) are widely available. Specific algorithms in this category include but are not limited to shape from stereo, shape from shading, shape from photometric stereo, as well as active lighting approaches. One downside common to all these methods is that, due to occlusion, only parts of the target geometry can be acquired during any single measurement. This makes it necessary to register multiple partial models into a single, coherent model of the whole object.

As to 3D range registration, there is a well established method which is the Iterated Closest Point (ICP) algorithm, in which category our method resides. Since its original introduction by Besl and McKay [1], researchers have tried to make the method more automatic and to improve its efficiency (e.g. [11]). An important variant of ICP was proposed by Blais and Levine [2]: rather than using the Euclidean distance between points on the different partial models, they suggest using projected distances along parallel rays as being cheaper to compute. Based on this suggestion and mostly in parallel to our work, Rusinkiewicz et al. [15] developed a system for interactively acquiring 3D models based on an active lighting approach for obtaining range maps from one view, and then registering these on the fly with the modified ICP algorithm. In order to achieve interactive rates for the ICP algorithm, they had to rely on a random subset of the points on the partial model for the registration.

Our system uses a variation of the ICP algorithm, that makes heavy use of graphics hardware to evaluate the alignment error between partial models. The actual optimization is similar to the work by Lensch et al. [5, 6] on registering 3D models with images.

In summary, our work is similar in spirit to the work by Rusinkiewicz et al., but with the following differences:

- We use graphics hardware to accelerate the evaluation of the alignment error for the ICP algorithm. This allows us to use all points from the projected model in all iterations, rather than only a subset.
- Our registration approach is essentially decoupled from the acquisition of the range maps, i.e. we could work with range maps from any source.
- In particular, we show that we can deal with very sparse and sometimes noisy data coming from stereo algorithms. For our specific implementation, we use a Digiclops trinocular vision system [13] from Point Grey Research together with their software library to extract range data at interactive rates. In principle, it would be possible to use other calibrated stereo setups with our method at the cost of incleased efforts to maintain calibration.
- By using stereo or other sources for the partial models, we can acquire color information together with the geometry.

3. Overview and Contributions

Our 3D model acquisition system consists of three modules: the range scanning module, the 3D-3D registration module and the model integration module. It is aimed at an interactive working mode, so that speed is the major concern for each of the three modules. For the range scanning, we choose a commercialized stereo system, the Digiclops, as the range finder, which works in real time. For the 3D-3D registration process, we adopt a depth based ICP algorithm, which takes advantage of the modern graphics hardware for geometric computations and thus runs faster than common ICP algorithms. The fast converging downhill simplex method is used for numerical optimization. Also for this module, we proposed a hardware accelerated approach which combines two images into one and thus greatly reduces the necessary amount of memory access for registration purpose. For the model integration, we use a volumetric model, which is fast to add new data to and fast to render as well.

The major effort has been in the acceleration of the 3D-3D registration process using graphics hardware.

4. Stereo Acquisition

To make the system work at interactive rates, the first step is to choose a suitable range acquisition device. In our system, a commercial color Digiclops stereo vision camera system provides real-time 3D digital image capture. A picture of it is shown in Figure 1, its size is $15.5 \times 15.5 \times 5.0$cm, and it can be fixed onto a tripod or held by hand.

There are several practical advantages for choosing this equipment. Firstly, the Digiclops consists of three on board progressive scan CCDs that allow for full 3D ranging of moving objects without interlacing problems found in standard NTSC CCDs. Secondly, the system is calibrated to high precision prior to shipping, both lens distortions and

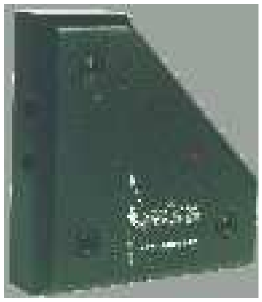

Figure 1. a picture of the Digiclops

misalignment are compensated and known as system parameters. This way we can avoid having to perform extensive calibration algorithms ourselves. Thirdly, the combined software API, the Triclops SDK, provides real-time range images using stereo vision technology, which allows users to accurately measure the distance to every valid pixel in an image and generate depth map fast and accurately. Most outliers obtained in the range scanning process, which is a known problem with 3D range scanners, can be removed by using strict surface validation functions provided by the SDK. Finally, the device acquires color at the same time as range information.

The Digiclops operates at a frame rate of about 16Hz at a resolution of 640x480 from each camera. Including the CPU time required to do stereo matching, the frame rate for generating new point clouds is in the order of 10Hz on a Pentium 4 with 1.6 GHz. Although this rate cannot be considered true real-time, yet, an interactive model acquisition can be achieved if the subject is moved slowly.

Like other approaches based on stereo matching, the Digiclops often fails to produce correct 3D points along the discontinuities on a surface. This problem mainly manifests itself along silhouette edges. When scanning the subject, a solid colored background can be used and the errors eliminated using a color segmentation algorithms.

5. Range-Based Iterated Closest Point

The 3D-3D registration is the central part for our geometric model acquisition system. With the exception of the work by Rusinkiewicz et al. [15] the performance of this task has so far been too low for interactive applications. Rusinkiewicz and his coauthors achieve their interactive rates by only using a random subset of point on the model for the registration.

We go a different route: contemporary graphics hardware is extremely efficient for geometric computations. Considering the fact that the registration problem involves mostly computations that the graphics hardware is optimized for, we adopt a depth based ICP method that takes advantage of the rendering pipeline by offloading the geometric computations to the dedicated graphics board. In this way, all the geometric computations are carried out implicitly during the rendering process, and the registration resolution is closely related to the size of the rendering window.

5.1. Registration Algorithm

Our algorithm makes use of graphics hardware for both transforming the geometry from one coordinate system to another, and for computing the quality of the alignment between two partial meshes using the rasterization part of the hardware.

We can categorize our method using Rusinkiewicz' framework for ICP algorithms [15]:

1. Point selection. All points in the range image obtained from the stereo algorithms are used for computing the alignment error.
2. Matching each point to a point in the depth image of the reference data set. We match each control point with the point in the other range image along the same orthographic projection ray, as shown in Figure 2. The major reason why we chose an orthographic projection over a perspective one is that in this way we avoid perspective foreshortening effect, so that the resolution is independent of the distance from the viewer.

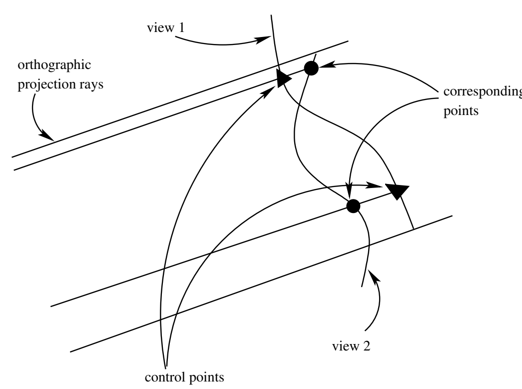

Figure 2. matching points in different images along the same orthographic projection ray.

3. Assigning a weight to each corresponding point pair and rejecting erroneous pairs. We look at each point pair equally. Note that not all points from one range map have a corresponding point in the other, since the object may not cover the whole image, and the stereo

algorithm may not be able to recover the depth at every pixel.

4. Evaluating surface consistency. A surface distance function based on the point matches is used as our consistency measure. Details are covered in the following subsection.

5. Optimization. The downhill-simplex method [12] is used as a multidimensional optimization algorithm to find the local minimum around an initial starting point in the parameter space.

5.2. Surface Alignment Metric

Throughout the range data registration process, it is necessary to compare two surface parts with varying pose to evaluate to which degree they could refer to the same part of a surface. A numerical alignment metric is used for this purpose, which acts as the objective function to be minimized by the optimization algorithm. Different local surface alignment metrics have been explored by researchers [4]. We use a overlapping region based surface distance in our approach.

For the registration process, we store the partial model that we have acquired so far in a voxel grid. From this partial model we can easily compute a dense range map by rendering a single OpenGL point for every occupied voxel. Similarly, every input range image can be converted into a range map seen from another view by simply reprojecting the individual points.

Since we match points along orthographic projection rays, the pair wise distance is simply the distance between two corresponding points in z direction (in camera space). An intuitive alignment metric is the sum of absolute pairwise distances over the overlapping region in the range images, normalized by the number of overlapping pixels. In order to avoid the registration process getting stuck in local minima, which is a common problem confronted by ICP algorithms, we adopt a segmented alignment metric function:

$$error(T) = \begin{cases} C_1 & \text{if } \|S_o\| = 0; \\ \frac{C_2}{\|S_o(T)\|} & \text{if } \|S_o\| < \varepsilon; \\ \frac{\sum_{\vec{p}\in S_o} d(\vec{p},C(\vec{p},T))}{\|S_o(T)\|} & \text{if } \|S_o\| \geq \varepsilon. \end{cases} \quad (1)$$

where T denotes the transformation to be evaluated, and S_o (a function of the transformation T) denotes the overlapping area between the two depth images. The function C describes the pairing function that yields the point $\vec{q}$ on the second surface that is closest to point $\vec{p}$ under the current transformation T. C_1 and C_2 are simply two constants, and d is the pair wise distance function $d(\vec{p}_1 - \vec{p}_2) = \|\vec{p}_1 - \vec{p}_2\|$.

The two constants and the threshold have to be chosen so that the condition

$$C_1 > \frac{C_2}{\|S_o(T)\|} > \frac{\sum_{\vec{p}\in S_o} d(\vec{p},C(\vec{p},T))}{\|S_o(T)\|} \quad (2)$$

is always true.

The employment of constants C_1, C_2 and the threshold ε is to honor surface alignments with large overlapping areas. By adding the second constraint to the optimal alignments, which is good alignments have to share large overlapping region with the reference, the possibility that the ICP stuck to local minima is greatly reduced.

5.3. Multidimensional Optimization

The registration process consists of minimizing the alignment metric as a function of the six dimensional rigid body transformation. We adopt the downhill simplex method [12] for this purpose, since it only requires point evaluations of the metric, but not its derivatives. As explained above, our acquisition pipeline starts from 3D point clouds, so our registration goal is to optimize the six degree rigid motion parameters, among which three are for the translation and three are for the rotation. So the transformation can be defined as $T = R\vec{p} + \vec{t}$. However, for surface samples not very close to the origin, small changes in rotation angles may result in large changes in translations. We have therefore decided to adopt a slightly different transformation scheme, given as $T = R(\vec{p} + \vec{t})$. We offload these matrix transformations onto the graphics hardware using the OpenGL API. For each range image we acquire, the six parameters have to be recovered in order for the data to be aligned to the model in the global coordinate system.

The downhill simplex method starts from an original simplex in parameter space, following its internal mechanism making a series of trials down through the complexity of the N-dimensional parameter space until it converges at a local minimum. For our interaction acquisition application, the subject is moved slowly and in a consistent manner, so that we can use the position obtained for the previous frame as a starting point and usually converge to the correct solution. If the motions are too fast, the registration may fail, and the algorithm has to be reset to a starting position that is known approximately (for example by roughly aligning the partial model by hand to the next image).

5.4. Hardware-Acceleration

The current generation of graphics hardware is quite flexible in its programmability, and its use is no longer limited to the traditional rendering and animation purposes.

Our work of using graphics hardware for registration purpose is inspired by the fact that graphics hardware is specifically designed to out-strip the performance of conventional CPU hardware for the class of computations that graphics hardware is dedicated to.

The most time consuming process for our algorithm is the numerical optimization which includes several tens of evaluations of the alignment metric. For each such evaluation, we need not only the depth image rendered from the current range image but also the depth image for the reference geometry, which means we need to access two pieces of memory for the most demanding operation. Note that we already use graphics hardware to generate the two range images in the first place (see Section 5.2). However, we can go further. If the two images can be somehow combined into one, that would help speeding up the registration process.

This can be achieved with the help of Register Combiners [10] or a similarly programmable rasterization stage. Our method starts by encoding the depth information into color images by applying a one dimensional linear texture with z mapped to the texture coordinate during rendering. It then subtracts the depth values between corresponding points of the two range maps using Register Combiners. Two general Register Combiner stages and one global constant color are used in our implementation.

The mathematical expression of the distance function (see Section 5.2) is the absolute difference between the two depth values stored in the same position. To calculate the absolute value, we use the "mux" operation for which the resulting pixel value is determined by a special alpha channel value:

$$mux(AB,CD) := (Spare0[\alpha] \geq 0.5)?AB:CD \qquad (3)$$

In combiner stage 1, an intermediate image is obtained by programming the combiners carefully. The pixel values in the intermediate image are

$$(tex1+(1-tex2))/2, \qquad (4)$$

which will then act as the special alpha values mentioned above. Since we have

$$0 < tex1, tex2 < 1 \qquad (5)$$

the resulted pixel values fall into this range as well.

Then in general combiner stage 2, the intermediate image is mapped to

$$\begin{aligned} A &= 2*(tex1+(1-tex2))/2-1; \\ C &= -2*(tex1+(1-tex2))/2+1 \end{aligned} \qquad (6)$$

and both B and D are assigned constant 1.

Finally, the mux operation is applied, and the pixel-wise absolute difference is thus obtained. Now, the surface consistency measure can be evaluated by reading back this image and summing up the per-pixel errors.

6. High-Quality Offline Optimization

Usually, for multiple view registration systems a gap may appear between the first and the last views due to the error accumulation between successive images. This is caused by the local nature of both the registration process and the geometric information available in one view. In our interactive registration system, we are able to trade off accuracy for speed. So a slower offline post-process using the same strategies may be applied to reconstruct a final 3D model of higher quality.

After the interactive registration pass, we have gained more preliminary information, which include

1. the user is confident that enough data has been acquired from the interactive mode;
2. results from the interactive registration are available as very good initial guesses for numerical optimization algorithm;
3. a preliminary voxel model is available.

Based on this information, the post-process can be run completely automatically.

The improvements of the result originate from several small differences approaches: one is to use a better prediction for the starting point for the numerical optimization, which not only decreases the number of iterations for registering one image speeding the process up, but also improves the registration accuracy. The second is instead of registering the current range image in sequence, we can do a global optimization step by taking the full model, deleting all point contributed from one view, and then re-registering that view. As we randomly pick views to be treated in this way, we gradually improve the registration of all views. Finally, we can use a voting scheme to determine outliers in the geometric model (e.g. erroneous data provided to us by the stereo algorithm). We do this by requiring that every voxel contains surface points contributed from a certain minimum number of different views (2-4 usually) yields good results for us.

7. Experimental Results

Before describing the test results, some practical issues concerning the implementation are discussed.

Range Data Acquisition: we use the Triclops library which is the software development kit shipped with the Digiclops. The output from this process is the corresponding 3D point cloud ready for registration using rendering pipeline.

Test system: our model acquisition system is implemented on a system running Redhat Linux 7.3, with Intel Pentium 4 1.6GHz processor and a GeForce 3 GPU. The first part which is the scanning process including the generation of 3D point cloud, operates at 3-4Hz, and the later one for registration and integration operates at about 1Hz on average. This could be accelerated further by using newer hardware or a dual-processor machine (since the stereo matching and the registration part are independent of each other).

Model Integration: for a quick rendering, we use a voxel model stored in an oct-tree model for the interactive mode. After a range image has been registered to the global coordinates system its points are merged into this oct-tree.

7.1. Synthetic Range Image Test for Registration

We tested our range based 3D-3D registration algorithm on a pair of synthetic range images created from a polygonal geometric model orthogonally mapped on the x-y plane. The second range image is generated from the same model transformed by a rotation vector of $(-5.0, 15.0, 5.0)$ (rotation parameters are in degrees) followed by a translation of 5 percent of its bounding box size along all three major axes. Both original range data sets consist about 17,000 points. The merged partial model is shown in Figure 3. Here, we use different colors to distinguish between pixels coming from different scans. The interpenetration of the two scans indicates a good alignment. For the registration process, a rendering window of size 128×128 is used. For the computation of per-pixel difference, both implementations using hardware acceleration and calculating in software directly have been tested and compared. To demonstrate the algorithms' ability to converge, we adopted various starting guesses which are all not very close to the true value. The standard deviation for all the six parameters have been calculated, among which the standard deviations for rotations are less than 0.5 degree and for translations are less than 2 percent of the bounding box size of the subject. Considering the window size used for calculations, the results are satisfying.

From these experiments, we can make the following observations: (1) the algorithm is capable of aligning range images accurately. (2) the fully hardware accelerated implementation has lower accuracy relative to the the implementation that computes the per-pixel difference in software. This is because the GPU we use has only eight bits for each color channel - the most recent generation of graphics hardware should not have these problems. (3) the fully hardware accelerated implementation is faster although there is not too much difference. As mentioned above, the starting position for this test is not close to the true value, the registration would be faster for better initial guesses.

7.2. Real Model Acquisition Result

We applied the model acquisition system to a textured ceramics model (Figure 4). During the acquisition process, the subject is moved slowly relative to the Digiclops, the newly scanned data set is registered to the partial model, and the updated model is rendered to the screen at the same time. For better control without occluding the object, we actually put the subject onto a plate and rotate the plate instead of the subject itself or the camera. After it is rotated for a whole circle, all the side views have been captured, the subject is then rotated about the axis that is roughly normal to the previous axis to get the data at the bottom of the model which can not be acquired by the circular scanning. By watching the immediately updated model, the user can find holes where no data has been acquired, and do more scans of that area to fill the holes. The movements have to be slow enough that successive scans overlap each other in a not too small area so that the registration result from previous views can be used as the starting point for the registration of the current view.

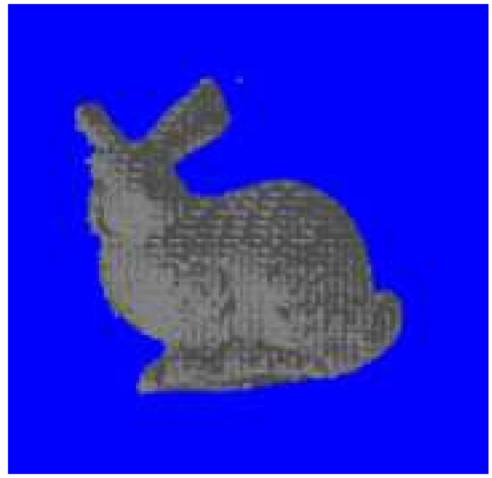

Figure 3. surface points from two different views registered to the global coordinate system.

Figure 5 shows several results of this method. The left image shows the result for a voxel grid of resolution $100 \times 100 \times 100$, while the center image has been generated with a resolution of $200 \times 200 \times 200$. The rightmost image includes the global optimization step. As can be seen, the stereo algorithm combined with our registration method generates 3D models of a ceramic unicorn successfully. There are, however, problems with very sharp features, such as the horn of the unicorn. Here the stereo data is very sparse, so that it is hard to capture the horn completely. In addition, the sparseness of the data also causes problems with the alignment. Since we adopt an averaged distance as the similarity metric, the fewer the data that are taken into consideration, the more the outliers affect the registration result.

Figure 5. Results of the model acquisition. Left and center: a low and a high resolution result without global optimization. Right: high resolution model with global optimization.

Figure 4. a picture of the subject

8. Conclusion and Future Work

In summary, we have presented an efficient, hardware based method for interactive 3D model registration. The results that we obtain with our system are quite encouraging. In the future, we would like to further improve the quality of the final model, for example by performing a better color filtering to reduce shading irregularities. We also hope to make use of a multi-processor machine to improve the performance to about 5 frames per second. At that point, the initial guesses for the new positions will be more accurate (due to faster update rates), so that the increase in performance will also manifest itself in an increased precision.

9. Acknowldegement

We would like to thank Point Grey Research for providing financial support and the Digiclops camera system for this project.

References

[1] P. Besl and N. McKay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 14(2):239–256, 1992.

[2] G. Blais and M. Levine. Registering multiview range data to create 3D computer objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 17(8):820–824, 1995.

[3] Cyberware. http://www.cyberware.com.

[4] D. F. Huber. Automatic 3d modeling using range images obtained from unknown viewpoints. In *Proceedings of the Third International Conference on 3-D Digital Imaging and Modeling*, pages 153–160, May 2001.

[5] H. Lensch, W. Heidrich, and H.-P. Seidel. Automated texture registration and stitching for real world models. In *Proceedings of Pacific Graphics '00*, Oct. 2000.

[6] H. Lensch, W. Heidrich, and H.-P. Seidel. Silhouette-based algorithm for texture registration and stitching. *Graphical Models*, 63(4):245–262, July 2001.

[7] M. Levoy, K. Pulli, B. Curless, S. Rusinkiewicz, D. Koller, L. Pereira, M. Ginzton, S. Anderson, J. Davis, J. Ginsberg, J. Shade, and D. Fulk. The Digital Michelangelo Project: 3D Scanning of Large Statues. In *Proceedings of Siggraph*, pages 131–144, July 2000.

[8] P. H. Lewis and M. S. Nixon. Viewpoint selection for complete surface coverage of three dimensional objects. In *Proceedings of the British Machine Vision Conference*, pages 740–750, Sept. 1998.

[9] J. Maver and R. Bajcsy. Occlusions as a guide for planning the next view. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 15(5):417–433, 1993.

[10] NVIDIA Corporation. *NVIDIA OpenGL Extension Specifications*, 2002. Available from http://www.nvidia.com.

[11] V. H. P. Krsek, T Pajdla and R. R. Martin. Range image registration driven by a hierarchy of surface differential features. *22nd Workshop of the Austrian Association for Patten Recognition*, pages 175–183, 1998.

[12] W. Press, S. Teukolsky, W. Vetterling, and N. Flannery. *Numerical recipes in C : the art of scientific computing*. Cambridge Univ. Press, 2nd ed. edition, 1994.

[13] P. G. Research. http://www.ptgrey.com/.

[14] C. Rocchini, P. Cignoni, C. Montani, P. Pingi, and R. Scopigno. A low cost 3d scanner based on structured light. In *Proceedings of Eurographics*, Sept. 2001.

[15] S. Rusinkiewicz, O. Hall-Holt, and M. Levoy. Real-time 3d model acquisition. In *Proceedings of Siggraph*, pages 438–446, July 2002.

[16] I. Stamos and P. Allen. Interactive sensor planning. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 489–494, June 1998.