2008 IEEE International Conference on
Robotics and Automation
Pasadena, CA, USA, May 19-23, 2008

4D Scan Registration with the SR-3000 LIDAR

Jason A. Stipes, *Member, IEEE*, John G. P. Cole and John Humphreys

Abstract— **An algorithm and data management scheme is presented to utilize the data output of the SR-3000, a 3D LIDAR sensor. The SR-3000 generates a 4D point cloud at video frame rates where each point is described by its 3D coordinates and its optical intensity. The implementation of each major component of the algorithm is described, including a projective ICP algorithm, reverse calibration equations for the SR-3000 sensor, the processing of the intensity data, and a point selection algorithm which includes the use of a 3D volume feature extraction algorithm. This ICP based alignment algorithm is used in the context of a data management scheme that stores and retrieves scan data and intermediate data products to realize pseudo-global scan alignment, data compression, and real time data display.**

I. Introduction

There have been a number of research efforts focused on the utilization of sensed 3D range data for the creation of 3D models and for other applications such as robotic localization and mapping [2][3][8][9]. Commonly, the sensors used have been based on structured light, stereo vision, or laser scanning technology. These technologies suffer from either a slow frame rate in the case of scanning sensors, or limited depth accuracy in the case of the stereo vision sensors. Recently, several companies have introduced new sensors that are commonly referred to as flash LIDAR (though actual measurements are often integrated over a finite period). These new sensors offer an interesting compromise between accuracy and frame rate. The SR-3000, manufactured by CSEM offers a sensor that can output a 176 x 144 array of four dimensional data (each point is comprised of an x, y, and z location and intensity value) at video frame rates. This paper examines the development of an algorithm to utilize the data captured by this new family of sensors. This algorithm addresses the conditioning, alignment and management of the 4D data. While the proposed algorithm has been developed and tested with output from the SR-3000, the approach is directly applicable to the output of other sensors as long as 3D point cloud data is available with the corresponding intensity data per point.

The proposed approach utilizes some common image processing techniques, a customized variant of the iterative closest point (ICP) algorithm, and a custom data management scheme to allow efficient storage and recall of previously acquired and processed data. Since it is possible to acquire the 4D scan data at up to 30 frames per second, it is desirable for the algorithm to be as efficient and computationally inexpensive as possible while still retaining overall robustness. Additionally, at such high frame rates, managing the sheer volume of data is a difficult challenge itself. As a result, the data management system must compress the acquired data by identifying and retaining only those points and scans that contain the most unique information about the scene. This sparse collection of data improves the algorithm speed and reduces the computational load required to render the acquired 3D graphical data.

In the next section, the implementation of each major component of the algorithm is described, starting with an overview of the core ICP algorithm and its derivation from other published approaches[1][10][11]. The unique and important aspects of the algorithm will be described in further detail including the derivation of the reverse calibration equations for the SR-3000 sensor and the pre-processing of the intensity data necessary to provide a more stable algorithm. The point selection algorithm, used to down sample to a sparse data set is described. This algorithm uses the processed image data and a volume feature extraction algorithm to select those scan points that are most unique. Finally, an overview of the data management scheme is described with specific attention to its utility in realization of pseudo-global scan alignment and data display.

II. IMPLEMENTATION

A. ICP Algorithm

The ICP algorithm has commonly been used for the alignment of 3D point cloud data, and numerous variations of the algorithm have been presented. Rusinkiewicz [10] summarized those variants of the ICP algorithm that were most computationally efficient. He described the ICP algorithm as a series of six stages: selection of points in one scan, matching those points to the second scan, weighting of the corresponding point pairs, rejecting certain pairs based

Manuscript received September 14, 2007. This work was supported through internal research funds at the Johns Hopkins University Applied Physics Laboratory.

J. A. Stipes is a member of the Senior Staff at The Johns Hopkins University Applied Physics Laboratory in Laurel, MD 20723 USA. (phone: 443-778-4851; fax: 443-778-1821; email: jason.stipes@jhuapl.edu)

J. G. P. Cole S. is a member of the Associate Staff at The Johns Hopkins University Applied Physics Laboratory in Laurel, MD 20723 USA. (phone: 443-778-2473; fax: 443-778-1821; email: john.cole@jhuapl.edu)

J. C. Humphreys is a research assistant at The Johns Hopkins University Applied Physics Laboratory in Laurel, MD 20723 USA. (phone: 443-778-2228; fax: 443-778-1821; email: jason.stipes@jhuapl.edu)

on distance, assigning an error metric and minimization of the error metric. Our algorithm focuses on developing the logic necessary to select and match points between scans generated by the 3D camera sensor.

In order to improve the speed of the scan alignment algorithm, it becomes desirable to select the fewest number of points for matching that will still provide for a good alignment of scans from the ICP algorithm. For this reason, selection of points is an important aspect of the ICP algorithm and approaches have included the use of all available points [1], uniform sampling[2][3][4][8], normal space sampling[10] and sampling based on image gradient or color[6][11]. Our approach is similar to that proposed by Weik [11] in which the image gradient is used to select points. Since we have intensity information available, it is desirable to use this data to enable a more robust and reliable convergence of the ICP algorithm. In addition to using data derived from the intensity of the scene, we calculate a simplified volume feature inspired by the work of Gelfand[5] for use in both selection and matching of point pairs. In the next sections, the image processing and volume feature extraction algorithms are described in more detail, and their use for the selection of points is described.

Once points have been selected from one scan, corresponding points in the scan with which the first is to be aligned must be identified. Many ICP algorithms perform an exhaustive search of this second scan to find the closest point, often using a k-d tree to accelerate the search. Even with this acceleration, this method is far too slow for our application. The efficient variants of the ICP algorithm described by Rusinkiewicz [10] use a projective method to either identify a corresponding point directly, or to locate the neighborhood in which to perform a local search for a compatible point. It is this latter approach that forms the basis for our technique. A projection from one scan to the other is performed and a local search using a simple hill climbing strategy is used to identify the point that is most compatible. In other approaches, point compatibility has been determined based on image gradient[11], color and normals[7]. Our approach uses the value of the scene image after having been processed to extract edges as well and the compatibility of our volume feature metric.

In order to accomplish the projective ICP algorithm, a reverse calibration of the system sensor is necessary. Reverse calibration is the process by which the pixel location (i.e. pixel row and column) may be determined by using the x, y and z coordinates of a point in the scan point cloud. This information is needed to perform the geometric calculations that allow projection of one acquired scan onto another for the purpose of point correlation between the scans. The details of how reverse calibration was accomplished for the SR-3000 are presented in the next section.

B. Reverse Calibration

In order for the projective version of the ICP algorithm to function properly, reverse calibration of the sensor is necessary. The sensor used in this research was the Swiss Ranger 3000 (SR-3000) by CSEM, shown in Fig. 1. This sensor is 50 x 67 x 42 mm in size and has a nom-ambiguity range of 7.5m. It has a pixel array resolution of 176 x 144 pixels which covers a field of view of about 47.5 by 39.6 degrees. In our testing the field of view was found to be a few degrees wider on each axis.

Fig. 1. Photo of the SR-3000 3D Camera

This sensor's configurable parameters allow access to the integration period, sensor modulation frequency, amplitude threshold and saturation thresholds. For our tests we used the following settings:

Modulation Frequency:	20MHz
Signal Amplitude Threshold:	128
Integration Period:	64
Saturation Threshold:	2700

These settings result in a frame rate of about 15 fps while rejecting both signals that are too weak or too strong that would result in distortion of the sensed geometry. In order to perform the reverse calibration of this sensor, the following procedure was executed:

1) Configure the sensor for the desired mode of operation using the settings previously listed.
2) Capture 100 frames of data from the sensor while mounted approximately 2 meters from a flat, off-white surface.
3) Average the captured data at each pixel to average out the depth noise of the sensor.
4) Perform a 2D regression of the data to determine both the row and column pixel of the image given the ratios x/z and y/z of the Cartesian coordinates for each point.

The form of the equation used for the 2D regression was a simple linear fit of the form:

$$P = a + bx + cy \quad (1)$$

In this equation P is the x or y pixel location x is the x/z coordinate ratio and y is the y/z coordinate ratio. The reverse calibration results for our sensor were:

$$P_x = 90.9 + 200(x/z) + 1.06E^{-7}*(y/z) \quad (2)$$

$$P_y = 70.0 - 2.80E^{-7}(x/z) + 200*(y/z) \quad (3)$$

The coefficients associated with the (y/z) variable in equation 2 and the (x/z) variables in equation 3 are sufficiently small to be omitted for the reverse calibration calculations. Fig. 2 shows the results of the regression for the Pixels in the x axis of the image, P_x. Note that all pixel locations are well resolved, with the errors being on the order of 10-4 pixels.

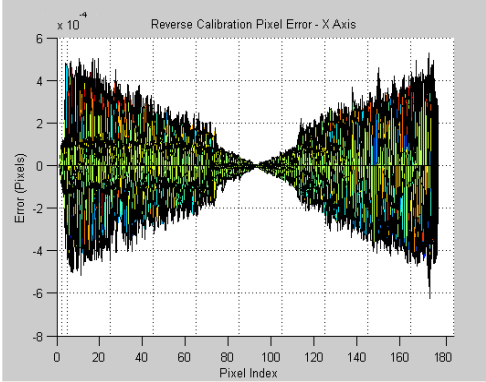

Fig. 2. Reverse Calibration Equation Error

C. *Image Processing*

The SR-3000 operates by flooding the scene with modulated infrared light. The scene illumination is not uniform and the resulting intensity image is typically brighter in the center and darker toward the edges as shown in Fig. 3 This variation in image intensity along with natural intensity variations in the scene due to light and shadow make the use of intensity values for frame to frame correlations subject to significant errors. To combat these intensity variations, some image processing is performed.

Fig. 3. Intensity Output of the SR-3000 3D Camera

The first derivative of the image in the x and y direction are taken and combined to yield an intermediate image that highlights those areas with higher image gradient as shown in Fig. 4. Then a dynamic threshold is applied to convert the image to a binary image that extracts the most prominent edges in the scene as shown in Fig. 5.

Fig. 4. First Derivative of Image Intensity

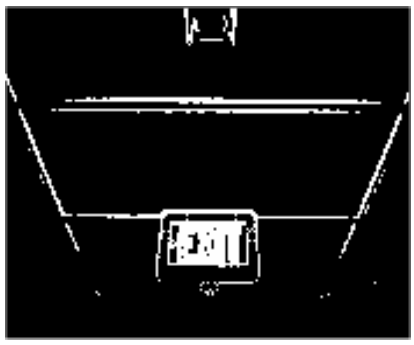

Fig. 5. Edges Extracted from the Processed Intensity Image

A series of Gaussian filters, decreasing in kernel width are applied to the binary images to yield a pyramid of images that are used in successive iterations of the ICP algorithm to guide alignment of the scene based on the locations of the extracted edges. The x and y gradients of each image in the Gaussian pyramid are calculated and used in the local search for compatible points for the projective portion of the ICP algorithm.

The intensity thresholds of the SR-3000 causes regions of the image to have zero range and intensity values if the intensity values are less than or greater than the two configurable intensity thresholds. This causes false edges to be extracted by the image processing algorithm near regions with zero intensity. To prevent this, a mask is applied to ignore those points that border regions with zero range and intensity.

D. *Volume Feature Extraction*

Analogous to edge detection in the 2D intensity domain, volume feature extraction identifies those edges, corners and areas of interest in the 3D geometric domain. Gelfand[5] suggests the use of a spherical volume feature in which spheres of varying diameters are applied at each point in the 3D data set. A 3D occupancy grid is created to estimate the intersection of the 3D point set with the sphere. This approach is used to estimate the portion of the sphere volume occupied by the local region of the 3D point set. This simple volume feature is able to differentiate scan geometries including inside and outside corners, edges and planar surfaces. Unfortunately the process of calculating the feature with differing sphere sizes and building a 3D occupancy grid for each calculation is computationally

expensive.

Our approach simplifies this process at the expense of fidelity since we intend only comparison of points, not exact matching of the volume feature among data sets. The volume feature calculation is simplified by using a cube shaped volume and method of calculation that does not require occupancy grids. Additionally, our volume feature is only evaluated for every second or third point in the data set since the filtering effect of the feature calculations yields a metric that changes rather slowly from pixel to pixel. The calculation estimates the portion of the cube filled by the 3D data when the cube is located with its center at the given data point. This estimated volume of the cube occupied by the 3D data points is the resultant volume metric. Equation 4 is evaluated at each point p_{xy} in the image where x and y are the indexes into the pixel array at the current point.

$$\sum_{i=x-n}^{x+n}\sum_{j=y-n}^{y+n}\left(d_{ij}-d_{xy}\right) \tag{4}$$

In this equation i and j are indexes into the pixel array, d_{ij} is the distance to each point p_{ij} within the cube and d_{xy} is the range to the current point p_{xy}. The interval n is determined by equation 5.

$$n=\frac{l}{2d_{xy}\sin\theta} \tag{5}$$

In this equation θ is the angle between pixels from the range sensor and l is the size of the desired volume cube. For the SR-3000 this value is 0.297 radians. Fig. 6 shows a graphical depiction of the volume feature for one scan.

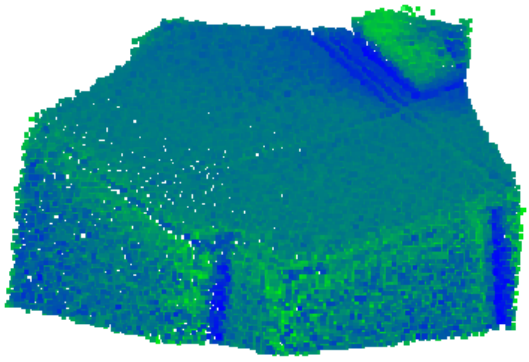

Fig. 6. Image Showing Output of Volume Feature Calculations

This image depicting the volume feature is then processed to extract the gradients in the x and y directions so that it may be used for the local point compatibility search in the ICP algorithm along with the processed intensity data.

E. Point Selection

In addition to being useful for the correlation of points from one scan to the next, the processed intensity and volume feature data may be used to select an initial subset points from the larger scene. This is accomplished through a probabilistic sampling process in which those points most unique in the scene have a higher probability of being sampled. In our algorithm, a fixed number of subsampled points are specified (i.e. 2500 which is approximately 10% of the number of points in a scan).

The points are then selected partially from the volume feature data and partially from the processed intensity data. The ratio of points selected using the volume feature and the intensity data is configurable, but is currently fixed throughout a data run. For future implementations, we aim to make this ratio an adaptive parameter that changes based on the scene geometry, selecting more points using the volume feature data when the scene is rich in geometric features, and favoring the intensity based sampling when geometric features are sparse.

Points are randomly selected without replacement from the set of points that represent the extracted edges in the binary image previously generated in the image processing step of the algorithm. These edges correlate to higher intensity gradients within the scene. These edges are landmarks within the scene that provide a reference for alignment of the two scans.

Once the required number of points have been selected using this method, the remaining points are acquired by probabilistically sampling the volume feature data. The volume features are formed into a histogram and points are pulled from the bins in inverse proportion to the number of samples within each bin. This results in points with feature values that are most rare in the scene being selected with a higher probability than those features that are more common. Invariably these rare features tend to come from regions of the image rich in geometric features that are beneficial to the alignment process of the ICP algorithm. Fig. 7 shows a two correlated scans with 2500 sampled points.

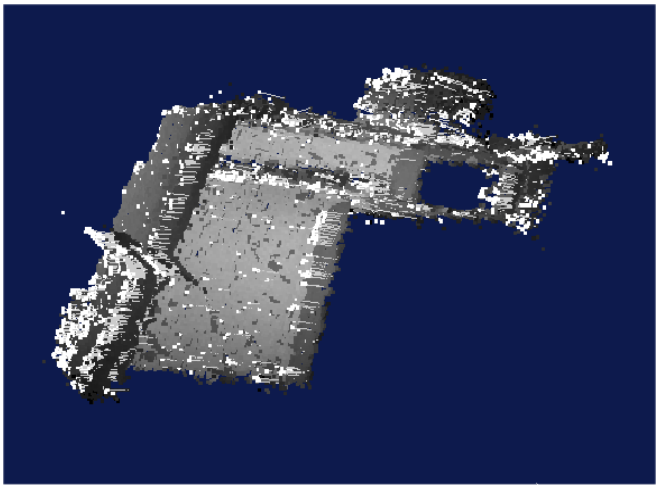

Fig. 7. An image showing two scans, the points selected for the ICP algorithm and lines drawn between the scans showing point correlations between the two images.

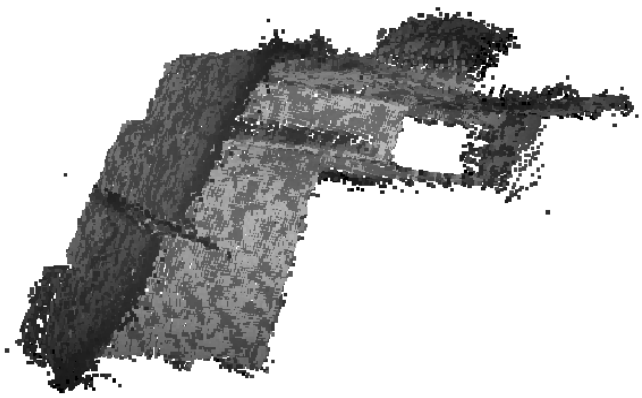

Fig. 8 The two scans of Fig. 7 shown after alignment

F. Pseudo-Global Alignment

In the discussion of scan alignment thus far, the general assumption has been that two scans, each acquired by the SR-3000 at different points in space and time, are being aligned to one another. In any alignment algorithm, some finite alignment error exits. This error may accumulate as one scan is aligned to the next in a daisy chain fashion. Since the SR-3000 acquires many frames of data per second, this cumulative error may build up over a relatively short period of time. To remedy this situation, a strategy to perform pseudo-global alignment of the acquired scans has been implemented.

Acquired scans are stored to a local scan repository. The repository stores the location of the scan and the regions of space that the scan intersects. New scans are only added to the repository if they contribute a specified amount of new information to the scene. For example, the repository can be configured to only store new scans based on how much they overlap the last scan stored in the repository, as well has how many points have been acquired in the area for all previously stored scans. In addition to archiving the complete scan data, intermediate data products such as the volume features, sampled points, and 4x4 rotation matrix that specifies the position and orientation of the scan in world coordinates, are also stored in the repository.

When a new scan is acquired, the previous scan may be used as an input to the alignment algorithm, or sampled points from all previously acquired scans that overlap the new scan may be used. Since the previously sampled points are stored for each scan in the repository, points may be randomly sampled from each overlapping scan in proportion to the amount of overlap that exists with the current scan. This sampling results in a composite scan that is assembled from all the previously acquired overlapping scans. The 4x4 translation and rotation matrix may then be used with the current scan's matrix to project the location of the points sampled from each previously acquired scan into the new scan.

With these correlations, the ICP algorithm may then proceed to align the current scan to the global population of collected scans. The data management scheme that makes this possible is rather complex. An overview of the data management approach is presented in the next section.

G. Data Management

The data management system stores acquired scans to disk and RAM when specified criteria are met, indexes them spatially, and provides access to the logged data upon request. This functionality is summarized below:

1) Logging Scans: The process of logging a newly registered scan involves two steps. First, the repository must decide whether this new scan contains new information. If, for example, scans are repeatedly being taken over a confined area, logging the redundant scans is unnecessary. Once a scan has been determined to have new, non-redundant information, it then must be logged to local memory as well as to disk.

2) Retrieving Samples of Interest: In order to globally register a new scan to all previously logged scans, it becomes desirable to query the repository for previous scans and sampled points within a given bounding box. This allows the algorithm to align the newest scan using points from several previous scans, thereby helping to reduce alignment error.

When the Scan repository is provided a bounding box and a number of points to collect, it divides the work among a hierarchical collection of spatial divisions, called scan cubes, which overlap the given bounding box. Each scan cube is queried to provide a percentage of the total number of samples, based on that scan cube's current number of samples.

H. Data Presentation

The scan repository also makes retrieval of the scans for display straightforward. Since the scans stored in the repository are a small subset of the total number of scans acquired, the rendering task is simplified. We take this step of data reduction step further in the graphics display itself, decimating and converting the point clouds stored in the repository to polygon meshes and stitching the scans together at a boundary to reduce redundant data. As a first step, those points with low intensity values are removed from each scan since these values are usually less accurate and contain more measurement noise.

Once this has been done, the pixel data can be meshed into a grid, allowing smooth surfaces to recreate the model. To do the scan meshing, triangles are constructed from the point cloud data to form a mesh. Choosing the first two pixels of a possible triangle involves traversing the scan from left to right along a single row, finding the first two valid points (non zero range).

The algorithm will attempt to find a third point for the triangle in the row below the first two valid points. The row is scanned for a valid pixel and the resulting triangle is created. After performing some sanity checks on the normal vector, the proximity of the triangles vertices, and that the

triangle has not been previously formed, the triangle is added to the mesh. This process is repeated until all points have been converted to polygons. An additional pass merges compatible adjacent triangles and removes the redundant vertices, thus reducing the size of the data. Fig. 9 shows the decimated vertices of a single scan point cloud and Fig. 10 shows the rendered polygons associated with these vertices.

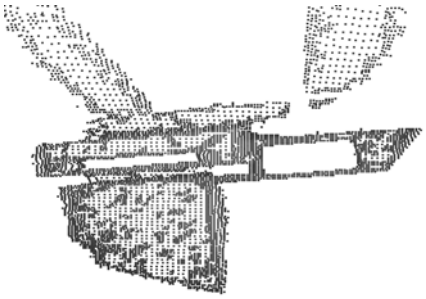

Fig. 9 A single scan with points decimated for polygon mesh formation

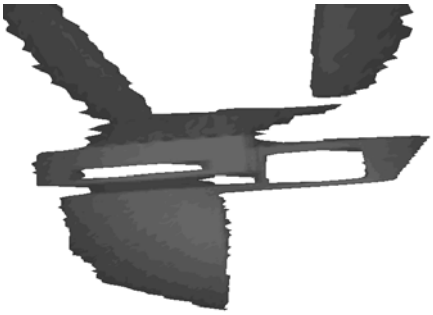

Fig. 10 The polygon mesh formed from the decimated points of Figure 8 is indistinguishable from the polygon mesh that uses all scan points

Since multiple overlapping scans must be rendered in a coherent fashion, it is necessary to stitch the scans together. The first scan received from the scan repository for rendering is used as a base scan to which the next scans will be appended. A scan to be added to the base scan must first undergo a harsh trim process that eliminates all data that overlaps data from the base scan. By traversing the new scan from all directions, a hole is cut out of the new scan mirrors the geometry of the base scan. Then, the triangle forming process is executed on the new scan.

At this point in the process the two scans still have a thin gap between them. This gap is filled by a stitching algorithm. During the process of trimming the new scan, its resultant borders and the locations where it intersects the valid edges of the previous scan are saved. Each border point of the new scan has at least one corresponding border point on the previous scan. These points are used as the basis to create a chain of polygons that close the gap between the two scans. The end result is shown in Fig. 11, where eleven scans have been stitched together to form a three-dimensional scene.

The end result of the scan meshing and stitching process is an accurate model of the real-life scene, providing details at any viewing position. In addition, memory storage is optimized. In the final product, the total number of stored pixels is reduced from a possible 278,784 (11 x 25344) to an efficient 34,950 – a reduction of 87%. The triangles that are used to represent the already efficient scene were then reduced by an additional 30% by the stitching algorithm.

Fig. 11 Multiple scans merged together into a single scene. The banded appearance shows the boundaries between scans that have been stitched together

III. Conclusion

This paper has presented the initial results of an effort to develop a robust real time scan alignment algorithm for use with the 4D high speed flash LIDAR sensors currently available. Future work in this area will focus on improvement of the algorithm speed, efficiency, robustness and adaptability for improved robotic navigation and mapping.

References

[1] Besl, P., McKay,N.1992."A Method for Registration of 3-D Shapes," Trans. PAMI, Vol.14,No.2.

[2] Blais, G., Levine ,M. 1995. "Registering Multiview Range Data to Create 3D Computer Objects," Trans. PAMI, Vol.17, No.8.

[3] Chen,Y., Medioni, G.1991."Object Modeling by Registration of Multiple Range Images," Proc. IEEE Conf. on Robotics and Automation 1991.

[4] Chen C., Hung, Y., Cheng, J. "A Fast Automatic Method for Registration of Partially-Overlapping Range Images," Proc. ICCV, 1998.

[5] Gelfand, N., Mitra, N., Guibas, L., Pottmann, H. "Robust Global Registration," Eurographics Symposium on Geometry Processing, 2005.

[6] Johnson, A. and Kang, S. "Registration and Integration of Textured 3-D Data," Proc. 3DIM, 1997.

[7] Levoy, M., et. al. "The Digital Michelangelo Project: 3D Scanning of Large Statues," Proc. SIGGRAPH, 2000.

[8] Neugebauer, P. "Geometrical Cloning of 3D Objects via Simultaneous Registration of Multiple Range Images," Proc. SMA, 1997.

[9] Rusinkiewicz, S. 2001."Real-Time Acquisition and Rendering of Large 3D Models," Ph.D.Dissertation, StanfordUniversity.

[10] Rusinkiewicz, S. Levoy, M. 2001. "Efficient Variants of the ICP Algorithm," Proc. 3DIM 2001.

[11] Weik, S. "Registration of 3-D Partial Surface Models Using Luminance and Depth Information,"Proc.3DIM,1997.