Registration and Integration of Multiple Range Images for 3-D Model Construction

Takeshi MASUDA[1] tmasuda@etl.go.jp
Katsuhiko SAKAUE[1] sakaue@etl.go.jp
Naokazu YOKOYA[2] yokoya@is.aist-nara.ac.jp

1: Image Understanding Section, Electrotechnical Laboratory, Tsukuba, Ibaraki 305 Japan.
2: Nara Institute of Science and Technology, Ikoma, Nara 630-01, Japan.

Abstract

Registration and integration of measured data of real objects are becoming important in 3-D modeling for computer graphics and computer-aided design. We propose a new algorithm of registration and integration of multiple range images for producing a geometric object surface model. The registration algorithm determines a set of rigid motion parameters that register a range image to a given mesh-based geometric model. The algorithm is an integration of the *iterative closest point* (ICP) algorithm with the *least median of squares* (LMS or LMedS) estimator. After registration, points in the input range image are classified into inliers and outliers according to registration errors between each data point and the model. The outliers are appended to the surface model to be used by registration with the following range images. The parts classified as inlier by at least one registration result is segmented out to be integrated. This process consisting of registration and integration is iterated until all views are integrated. We successfully experimented with the proposed method on real range image sequences taken by a rangefinder. The method does not need any preliminary processes, like smoothing or trimming of isolated points, because of its robustness.

1 Introduction

The problem of registration and integration of multiple range images is one of the most important problems in 3-D shape analysis, especially in the applications of computer graphics, computer-aided design and numerical simulation. A range image usually lacks data of the points hidden behind objects or of the points that are out of the field of view of the sensor. It is usually difficult to measure the whole surface of an object at one time, thus the multiple measurement is necessary. The multiple views of an object are acquired by moving (rotating) the sensor around the object or by moving the object in front of the fixed sensor. Even though the motion is controlled, it may not be so accurate as the range measurement. The coordinate system of each set of measured data is sensor-centered. Therefore, in order to integrate multiple range images, we first have to represent all data in the object-centered coordinate system. This is what we call registration, and a number of researches have been done on this topic [1].

Besl and McKay[2] proposed the *iterative closest point* (ICP) algorithm. This algorithm estimates a set of rigid motion parameters that register a data shape to a model shape. This method works well as long as all data point has its corresponding model point. This method is affected by outliers like noise and occlusion, especially when we apply the method to multiple range images. Masuda and Yokoya [3] have proposed a robust method for registering a pair of dense range images, which is an integration of the ICP algorithm with random sampling and the *least median of squares* (LMS or LMedS) estimator. The LMS estimator gives an estimation that minimizes the median of squared residuals, while the standard *least squares* (LS) estimator gives an estimation that minimizes the sum of squared residuals. The LMS estimator is more robust than the LS estimator in that the LMS estimator is not affected by outliers of up to 50%.

This paper proposes a new method of integrating multiple range images to construct a geometric surface model and of registering a range image to the model. Comparing to the other published works [4, 5], the proposed method employs general and strong statistical concept of robust regression and outlier detection. The registration method (Section 2) determines a set of rigid motion parameters that registers a range image to a triangulated model. The method iterates the process of random sampling, estimation of motion parameters by the ICP algorithm (Section 2.2) and de-

1015-4651/96 $5.00 © 1996 IEEE
Proceedings of ICPR '96

termination of the best estimation (Section 2.4). Our integration method first classifies the range data into *inliers* and *outliers*(Section 3.1). Then the classification result is used to integrate two models (Section 3.2). One model is an accumulation of all the input range images to be used in registration. The other is an integrated model that contains only inliers. This model is common to at least two range images. We experimented our algorithm on a sequence of real range images (Section 4).

2 Registration of Range Data and Geometric Model

A range image is a set of 3-D coordinates $R = \{\boldsymbol{r}(\boldsymbol{u}) | \boldsymbol{u} \in U\}$ where each element is given by $\boldsymbol{r}(\boldsymbol{u}) = (r_1(\boldsymbol{u}), r_2(\boldsymbol{u}), r_3(\boldsymbol{u}))$ and a point in the 2-D image plane is denoted as $\boldsymbol{u} = (u_1, u_2) \in U$. The set U denotes the set of regular pixels on which 3-D coordinates of measured points are given.

We represent a set of 3-D rigid motion parameters by a 3-D congruent transformation that consists of a 3×3 rotation matrix $\boldsymbol{R}$ and a 3-D translation vector $\boldsymbol{t}$ as $\boldsymbol{T} = \{\boldsymbol{R}, \boldsymbol{t}\}$. A rigid motion of a 3-D point $\boldsymbol{x}$ is given by a linear transformation as $\boldsymbol{T}(\boldsymbol{x}) = \boldsymbol{R}\boldsymbol{x} + \boldsymbol{t}$. The transformation of a set of points $X = \{\boldsymbol{x}\}$ is represented by $\boldsymbol{T}(X) = \{\boldsymbol{R}\boldsymbol{x} + \boldsymbol{t}\}$.

We use triangulation as a geometric model to be registered with range images. Triangulation represents a surface as a set of triangles, and it is one of the most popular surface models. The registration problem addressed is to estimate a 3-D rigid motion parameters which matches a range image to a triangulated model.

2.1 Overview of the algorithm

The registration uses the LMS estimation to determine the rigid motion parameters $\boldsymbol{T}_{\mathrm{LMS}}$ between the range image R and the triangulated model Δ. The motion $\boldsymbol{T}_{\mathrm{LMS}}$ is initialized and then updated to the motion best so far evaluated in each trial. At the end of all trials, $\boldsymbol{T}_{\mathrm{LMS}}$ satisfies the LMS condition. Our registration process $\mathsf{LMS}(R, \boldsymbol{T}_{\mathrm{LMS}}, \Delta)$ is carried out as follows:

1. Initialization of $\boldsymbol{T}_{\mathrm{LMS}}$, $\tilde{m} \leftarrow \mathsf{MS}(R, \boldsymbol{T}_{\mathrm{LMS}}, \Delta)$.
2. Steps 3–6 are repeated taking n from 1 through N_{T}, where N_{T} signifies the number of trials.
3. A set of N_{S} points P is extracted from R at random: $P \leftarrow \mathsf{RS}(R, N_{\mathrm{S}})$.
4. The point set P is used by the ICP algorithm with the triangulated model Δ to estimate the motion parameters: $\boldsymbol{T}_{\mathrm{ICP}} \leftarrow \mathsf{ICP}(P, \boldsymbol{T}_{\mathrm{LMS}}, \Delta)$.
5. The estimated motion is evaluated by $m \leftarrow \mathsf{MS}(R, \boldsymbol{T}_{\mathrm{ICP}}, \Delta)$.
6. If $m < \tilde{m}$, update $\boldsymbol{T}_{\mathrm{LMS}} \leftarrow \boldsymbol{T}_{\mathrm{ICP}}$ and $\tilde{m} \leftarrow m$.

The motion $\boldsymbol{T}_{\mathrm{LMS}}$ finally minimizes the median of squared residuals $\mathsf{MS}(R, \boldsymbol{T}_{\mathrm{ICP}}, \Delta)$ for all $\boldsymbol{T}_{\mathrm{ICP}}$ estimated by the ICP algorithm at each iteration. The following sections describe the detail of the algorithm.

2.2 ICP algorithm

The ICP algorithm [2] estimates rigid motion parameters between paired 3-D shapes. This algorithm is composed of two procedures: one generates temporary correspondences and the other estimates the motion using a unit quaternion from the point correspondences. These two procedures are iterated until two given shapes are registered by the estimated transformation.

We apply the ICP algorithm using the set of N_{S} points P selected by the random sampling as the data shape and the triangulation Δ as the model shape. The motion estimate of the ICP algorithm $\boldsymbol{T}_{\mathrm{ICP}}$ is initialized by the motion $\boldsymbol{T}_{\mathrm{LMS}}$ in Section 2.1 that is evaluated best so far during preceding trials. Each point $\boldsymbol{p} \in P$ is transformed to $\boldsymbol{T}_{\mathrm{ICP}}(\boldsymbol{p})$, then it is temporarily paired to its closest point in the model $\boldsymbol{c} \in \Delta$, and a set of N_{S} points $C = \{\boldsymbol{c}\} = \mathsf{C}(\boldsymbol{T}_{\mathrm{ICP}}(P), \Delta)$ is determined. The point sets P and C are used to estimate the new rigid motion parameters $\boldsymbol{T}_{\mathrm{ICP}}$ using the quaternion representation, then the motion parameters $\boldsymbol{T}_{\mathrm{ICP}}$ is updated. These procedures are iterated until $\boldsymbol{T}_{\mathrm{ICP}}(P)$ converges to C. The ICP algorithm $\mathsf{ICP}(P, \boldsymbol{T}_{\mathrm{ICP}}, \Delta)$ is summarized as follows:

1. Initialization: $k \leftarrow 1$, $\boldsymbol{T}_{\mathrm{ICP}} \leftarrow \boldsymbol{T}_{\mathrm{LMS}}$, $d_0 \leftarrow \infty$.
2. The point correspondence is established: $C \leftarrow \mathsf{C}(\boldsymbol{T}_{\mathrm{ICP}}(P), \Delta)$.
3. The motion between paired points is determined using the quaternion representation: $\boldsymbol{T}_{\mathrm{ICP}} \leftarrow \mathsf{Q}(P, C)$.
4. $d_k \leftarrow d(\boldsymbol{T}_{\mathrm{ICP}}(P), C)$.
5. Steps 2–4 are repeated until $d_{k-1} - d_k < \tau\sigma$ increasing $k \leftarrow k + 1$.

It is proven that the sequence of mean squared errors $d_k = d(\boldsymbol{T}_{\mathrm{ICP}}(P), C)$ decreases monotonically[2]. The iteration is terminated if d_k does not change more than the threshold value τ which is made dimensionless by the approximate data size σ.

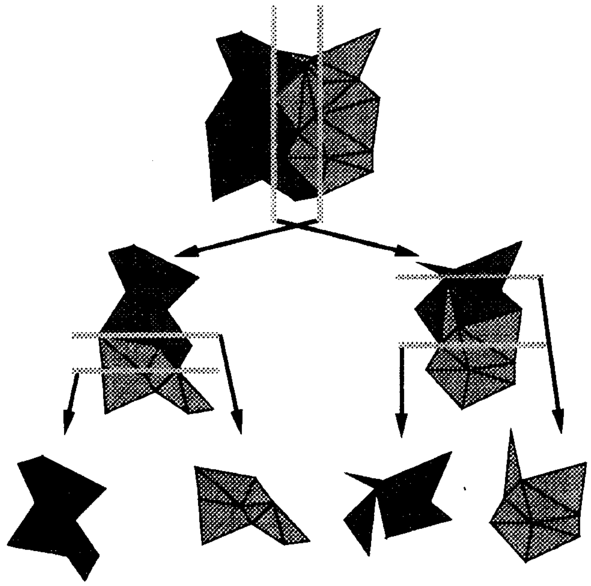

Figure 1: Modified k-d tree to search the closest triangle. Each node represents a set of triangles, which is divided in two sets along one of k coordinates. The two boundaries are determined: the maximum coordinate of the lower set and the minimum of the higher set. The boundaries hierarchically determine the circumscribing rectangular space of containing triangles.

2.3 Closest triangle search

The closest point on the triangulation is determined by searching the closest triangle using k-d tree (Figure 1). The k-d tree is a binary tree which is efficiently used in the algorithm to search the closest point in the k-dimensional space [6]. Each node in this tree has two subnodes whose boundary bisects the space along one of the k axes, thus selecting one side of the boundary induces the reduction of the rectangular space to be searched. We extended this tree structure to search the closest triangle. A triangle occupies its own size in space, thus we store two boundaries: maximum boundary of lower subset and minimum boundary of higher subset. To find the closest triangle from a point, we first search the triangle surrounded by the same boundaries where the point exists by selecting the more appropriate subnode. Then the subnode not searched yet is tested recursively as long as its boundaries are within the minimum distance so far encountered.

The distance from a point to a triangle is uniquely determined. Let the point be denoted by $\boldsymbol{p}$ and three vertices of the triangle be represented by $\boldsymbol{v}_1$, $\boldsymbol{v}_2$, $\boldsymbol{v}_3$. The condition that the closest point $\boldsymbol{c}$ is within the triangle is $\boldsymbol{c} = t_1\boldsymbol{v}_1 + t_2\boldsymbol{v}_2 + t_3\boldsymbol{v}_3$, where t_1, t_2, t_3 are weights satisfying $t_1 + t_2 + t_3 = 1$ and $t_1 \geq 0, t_2 \geq 0, t_3 \geq 0$. These weights are calculated as:

$$\begin{aligned} t_1 &= [\boldsymbol{n}, \boldsymbol{p} - \boldsymbol{v}_2, \boldsymbol{p} - \boldsymbol{v}_3]/||\boldsymbol{n}||^2, \\ t_2 &= [\boldsymbol{n}, \boldsymbol{p} - \boldsymbol{v}_3, \boldsymbol{p} - \boldsymbol{v}_1]/||\boldsymbol{n}||^2, \\ t_3 &= [\boldsymbol{n}, \boldsymbol{p} - \boldsymbol{v}_1, \boldsymbol{p} - \boldsymbol{v}_2]/||\boldsymbol{n}||^2, \end{aligned}$$

where $\boldsymbol{n} = (\boldsymbol{v}_1 - \boldsymbol{v}_2) \times (\boldsymbol{v}_2 - \boldsymbol{v}_3) = \boldsymbol{v}_1 \times \boldsymbol{v}_2 + \boldsymbol{v}_2 \times \boldsymbol{v}_3 + \boldsymbol{v}_3 \times \boldsymbol{v}_1$, and $[\cdot,\cdot,\cdot]$ signifies the scalar triple product. If at least one of the three weights is negative, the closest point locates on the side of the triangle. For example, if $t_1 < 0$, the closest point is on the segment between the vertices $\boldsymbol{v}_2$ and $\boldsymbol{v}_3$, and the weights are determined as:

$$\begin{aligned} t_1 &= 0, \\ t_2 &= (\boldsymbol{v}_2 - \boldsymbol{v}_3) \cdot (\boldsymbol{p} - \boldsymbol{v}_3)/||\boldsymbol{v}_2 - \boldsymbol{v}_3||^2, \\ t_3 &= (\boldsymbol{v}_2 - \boldsymbol{v}_3) \cdot (\boldsymbol{v}_2 - \boldsymbol{p})/||\boldsymbol{v}_2 - \boldsymbol{v}_3||^2. \end{aligned}$$

If $t_2 < 0$ in addition, the closest point coincides with the vertex $\boldsymbol{v}_3$, and the weights are reconsidered as $t_1 = t_2 = 0, t_3 = 1$. We determine the distance from a point to the triangulated model as the distance to the closest point on the closest triangle.

2.4 Evaluation of estimated motion

We have described the motion estimation by the ICP algorithm, which uses the points selected by the random sampling. Then we evaluate the estimated motion parameters and find the best.

Each point $\boldsymbol{r}$ of the range image R is transformed by the motion $\boldsymbol{T}$ to $\boldsymbol{T}(\boldsymbol{r})$. The closest point $\boldsymbol{c}$ from a transformed point $\boldsymbol{T}(\boldsymbol{r})$ is searched on the triangulated model $\boldsymbol{c} = \mathsf{C}(\boldsymbol{T}(\boldsymbol{r}), \Delta)$, then the distance from $\boldsymbol{T}(\boldsymbol{r})$ to $\boldsymbol{c}$ is determined. We calculate the distance from each transformed point in R to its closest point $\boldsymbol{c}$ in the model Δ, and then evaluate the goodness of the motion $\boldsymbol{T}$ by the median of those distances, which is represented as

$$\mathsf{MS}(R, \boldsymbol{T}, \Delta) = \sqrt{\underset{\boldsymbol{r} \in R}{\mathrm{med}} ||\boldsymbol{T}(\boldsymbol{r}) - \boldsymbol{c}||^2}. \quad (1)$$

A motion $\boldsymbol{T}_{\mathrm{ICP}}$ is estimated in each trial, and we find the motion $\boldsymbol{T}_{\mathrm{LMS}}$ from one of the $\boldsymbol{T}_{\mathrm{ICP}}$s that gives the minimum MS through N_T trials.

Using all the points in the evaluation is computationally expensive. We implemented a two-stage evaluation to reduce the computation cost. At first, we do an approximate evaluation with sampled points from the range image (for example, 256 points from the original points were used in our experiments), and only if this approximate evaluate is better than the best evaluate so far, we invoke the complete evaluation with all the points. The update of the best evaluate is determined only by the complete evaluate.

3 Integration of Range Images

3.1 Classification of range data into inliers and outliers

We classify the points in the range image R into two categories: *inliers* and *outliers*. A point $\boldsymbol{r}$ is classified as *inlier* if the transformed point $\boldsymbol{T}(\boldsymbol{r})$ is closer to the triangulation model Δ than the distance threshold θ. The remainders are *outliers*. The outliers may contain measurement errors, or they appear by the motion from the occluded region or from outside of the sensing area, or they lost the correct correspondence that were occluded by the motion, or they move differently from the object we are segmenting out as inliers. We represent the classification results as $R_{\cap\Delta}$ for inliers and $R_{\cap\overline{\Delta}}$ for outliers. The threshold value θ is determined by $2.5\hat{\sigma}$, where $\hat{\sigma}$ signifies the standard deviation of residuals estimated in a robust way as $\hat{\sigma} = 1.4826\mathsf{MS}(R, \boldsymbol{T}_{\mathrm{LMS}}, \Delta)$[7].

3.2 Integration of multiple-view range images

By applying the registration and segmentation algorithm to multiple-view range images $R^t(1 \leq t \leq N_{\mathrm{R}})$, we construct a geometric model using only reliable points extracted from these range images as inliers.

We first generate a initial triangulated model from the first range image R^1, and we determine the motion parameters $\boldsymbol{T}_t(2 \leq t \leq N_{\mathrm{R}})$ that make match the range image R^t to the model, then we update the model every time we register a new range image. We keep two models internally. One is the integrated model Ω that contains only inliers, whose matching error is guaranteed. The other is the accumulated model $\mho$ that contains all the points once appeared including outliers. This model is used by registration, and the part once classified as outliers can be re-classified as inliers in the registration with a new view. These models are constructed incrementally as below.

1. Initialization: $\mho \leftarrow \Delta(R^1)$, $\Omega \leftarrow \emptyset$, $\boldsymbol{T}_1 \leftarrow \{\boldsymbol{I}, \boldsymbol{O}\}$.
2. Steps 3–5 are repeated taking t from 2 to N_{R}.
3. Registration: $\boldsymbol{T}_t \leftarrow \mathsf{LMS}(R^t, \boldsymbol{T}_t, \mho)$.
4. Update Integrated Model: $\Omega \leftarrow \Omega \cup \Delta((R^t_{\cap\mho})_{\cap\overline{\Omega}})$.

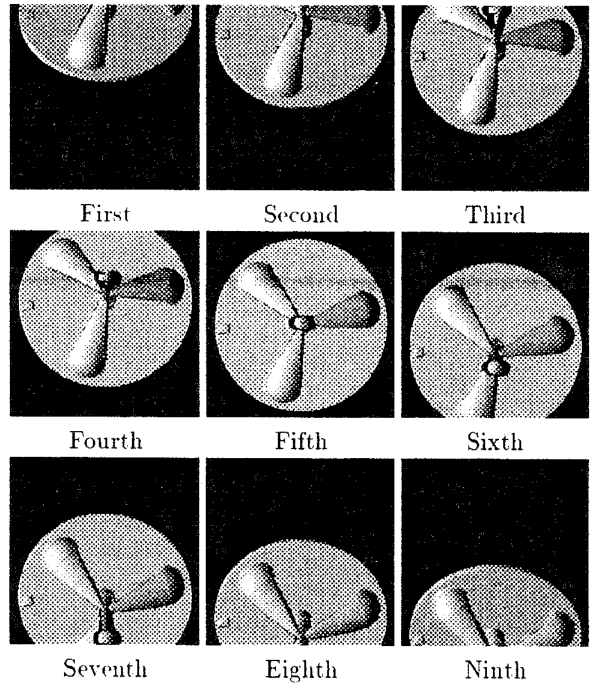

Figure 2: Nine frames of the NRCC range image sequence.

5. Update Accumulated Model: $\mho \leftarrow \mho \cup \Delta(R^t_{\cap\overline{\mho}})$.

4 Experimental Results

We applied the proposed method to a sequence of real range images. This data set was measured by using the NRCC (National Research Council of Canada) rangefinder. The NRCC range image is a scalar-valued 256×256 array containing vertical height values. Each pixel has width of 0.5mm in both horizontal and vertical coordinate directions, and the height resolution is 50μm [8]. Figure 2 shows nine sequential input range images with shading. The object is a model of a grip for a space robot arm that is a construct of a circular plane, three supports, and a rod with a spherical head. Occlusion is caused by the projections such as the rod and the supports, and truncation happens at the upper limit of the measurable domain.

The final integration result is illustrated in Figure 3. The registration and segmentation successfully extracted the inliers from the accumulated model $\mho$. The occluded part under the support and behind the rod was reconstructed from multiple views. The integrated model finally consists of 101153 triangles.

This result was obtained with $N_{\mathrm{S}} = 10$ and $N_{\mathrm{T}} =$

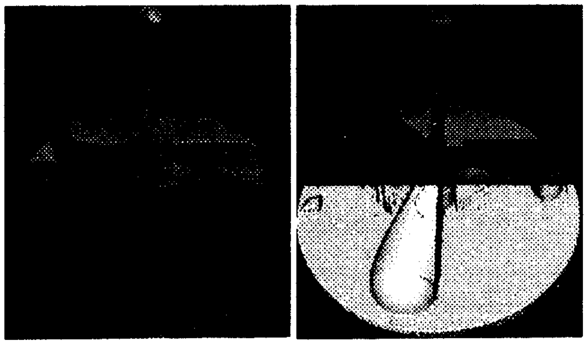

Figure 3: Integration of a sequence of NRCC range images. The left image illustrates the integrated model Ω. Each gray scale signifies the index of the source range image whose inliers are detected and appended to the model. The right image illustrates the accumulated model $\mho$. The white part is the original model generated from the first range image and the parts with gray scale are appended outliers detected by registration. The integrated model Ω does not include outliers one can find in the accumulated model $\mho$.

200, which were fixed for all the registration. The final residuals $\mathsf{MS}(R^t, \boldsymbol{T}_{\mathrm{LMS}}, \mho)(2 \leq t \leq 9)$ for each range images were 69.1μm, 68.9μm, 77.9μm, 66.4μm, 60.2μm, 60.0μm, 57.0μm and 57.1μm respectively. These residuals are comparable to the measurement accuracy of the rangefinder. The median of residuals drops fast at the beginning of trials, but more trials are necessary to make the registration more accurate. If there is the desired accuracy, the trials can be terminated if it is satisfied.

The computation time required by each registration varied from 9 minutes to 17 minutes, and 11 minutes on the average. Total computational time to integrate all 9 images was about 2 hours by a single-processor workstation (HP Apollo 9000 Model 735) whose performance is reported as 40 MFLOPS on the double-precision Linpack benchmark.

5 Conclusion

We have proposed a new method for registration and integration of multiple-view range images for constructing a 3-D geometric model. The registration method is an integration of the ICP algorithm with the random sampling and LMS estimator. The segmentation classifies each point in an input range image into inliers and outliers. Finally, we constructed a data set of the object using only the inliers from multiple views. To reduce the computational cost for searching the closest triangle, we introduced the modified k-d tree representation. We have successfully experimented with the proposed method on a sequence of real range images. The experiments showed that the method could determine motion parameters in the accuracy nearly the same as that of the measurement. In future work, we intend to develop an integration method that generates more sophisticated models like optimized triangular mesh or higher order geometric models.

Acknowledgments

This work was partially carried out under the collaboration program "Image Understanding Systems" between ETL and NRCC. The authors would like to thank Nobuyuki Otsu and Jacques Domey for making efforts to this project possible. This work was also supported in part by the Real World Computing (RWC) Program.

References

[1] B. Sabata, J. K. Aggarwal, "Estimation of motion from a pair of range images: A review," *CVGIP: Image Understanding*, Vol. 54, No. 3, pp. 309–324, Nov. 1991.

[2] P. J. Besl, N. D. McKay, "A method for registration of 3-D shapes," *IEEE Trans. PAMI*, Vol. 14, No. 2, pp. 239–256, Feb. 1992.

[3] T. Masuda, N. Yokoya, "A robust method for registration and segmentation of multiple range images," *Computer Vision and Image Understanding*, Vol. 61, No. 3, pp. 295–307, 1995.

[4] M. Soucy, D. Laurendeau, "Multi-resolution surface modeling from multiple range views," *Proc. IEEE Computer Vision and Pattern Recognition*, pp. 348–353, 1992.

[5] G. Turk, M. Levoy, "Zippered polygon meshes from range images", *Proc. SIGGRAPH 94*, pp. 311–318, 1994.

[6] D. A. Simon, M. Hebert, T. Kanade, "Real-time 3-D pose estimation using a high-speed range sensor," *Proc. IEEE Int. Conf. Robotics and Automation*, pp. 2235–2241, 1994.

[7] P. J. Rousseeuw, A. M. Leroy, *Robust Regression & Outlier Detection*, John Wiley, New York, 1987.

[8] M. Rioux, F. Blais, J.-A. Beraldin, P. Boulanger, "Range imaging sensors development at NRC laboratories", *Proc. IEEE Workshop on Interpretation of 3D Scenes*, pp. 154–160, Nov. 1989.