Proceedings of the 1998 IEEE/RSJ
Intl. Conference on Intelligent Robots and Systems
Victoria, B.C., Canada • October 1998

Localization of Insulators in Electric Distribution Systems by Using 3D Template Matching from Multiple Range Images

Kentaro Kawamura[1)] Mark D. Wheeler[2)] Osamu Yamashita[1)] Yoichi Sato[3)] Katsushi Ikeuchi[3)]
1) Kyushu Electric Power Co. , Inc. Fukuoka, 815, Japan
2) Cyra Technologies Oakland, CA
3) University of Tokyo Tokyo, 106, Japan

Abstract

Kyushu Electric has developed a dual-armed mobile-robot for use in electronic distribution systems. Although some human intervention is still required, the robot greatly reduces the demands on the human operator. In order to automate some of the robot's capabilities, we have developed a 3D object-localization method for robot's positional adjustment. The method is designed to be insensitive to noise and outliers while, at the same time, it has optimal run-time efficiency. This paper first describes our algorithm, and then presents a performance evaluation.

1 Introduction

Kyushu Electric has developed a dual-armed mobile robot for maintaining hot-lines in electronic distribution systems [1]. (See Figure 1.) The robot is quite successful in reducing a number of tasks that human operators are presently required to perform under poor working conditions. The robot also reduces the number of operator hot-line accidents, thereby eliminating most of the electric service interruption periods caused by these accidents. At present, six such robots are deployed in the area serviced by the company.

This robot is controlled in the tele-operation mode: it still requires human assistance to perform such tasks as insulator recognition, positional adjustments of the robot, and guidance toward electric lines and insulators. We have begun an additional project to automate some of these required tasks. In particular, this paper describes a method for recognizing insulators for robot positional adjustment.

Generally speaking, there are two sub-problems in object recognition: object classification and localization. The first problem, classification, determines what the object is; the second problem, localization, determines the precise position of the object given its rough initial estimate. Of these two problems, we have concentrated on object localization. Our scenario allows us to assume that the system knows the object's class as well as its rough estimation, because currently a human operator drives the mobile robots from our base to the target electric pole, parks the robots near the pole, and raises the robots to a certain predetermined height. The required capability is to precisely determine the distance and rotation of the target insulator with respect to the robot.

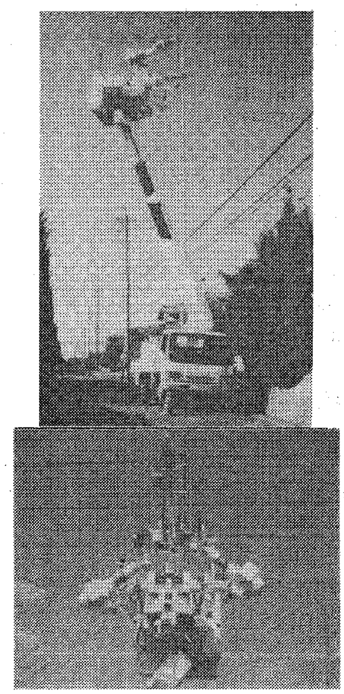

Figure 1: The mobile hot-line work robot

0-7803-4465-0/98 $10.00 © 1998 IEEE

Proceedings of the 1998 IEEE/RSJ
Intl. Conference on Intelligent Robots and Systems
Victoria, B.C., Canada • October 1998

Several localization algorithms have been proposed. Besl and McKay[2] presented the iterative closest-point(ICP) algorithm that iteratively computes nearest-neighbor correspondences between points on the model surface and points in the image data. They compute the optimal, least-squares solution of the pose. Zhang[3] presented an improvement in the ICP algorithm to reduce the outlier sensitivity of ICP. He proposed a method for dynamic threshold selection to remove outliers; however, the method reduces the outlier only when the initial pose error is small enough. Haralick[4] et al. proposed and analyzed the use of M-estimators for robust pose estimation. Their algorithm used iteratively reweighted least-squares estimation, but the pose estimate could be accurately achieved only when a sufficiently large set of point correspondences were available. Lowe's algorithm[5] may be considered as a medium ground between pose estimation and pose refinement. The random sample consensus (RANSAC) method of Fischler and Bolles[6] and the least-median-of-squares (LMedS) method of Mintz and Rosenfeld[7] are very useful for removing outliers.; unfortunately, they require an expensive combinatoric search.

We propose an algorithm which is robust and computationally inexpensive. In this paper, we present a method for localizing 3D objects in 3D range-image-data. Our approach to localization comprises the following steps:

1. Predict the visibility of model points in the image
2. Dynamically establish the correspondences between the model and the image points
3. Refine the pose estimate using the point correspondences

The first problem is how to efficiently compute the visibility of the points of the model with respect to the range-image view. We present a local approximation method for predicting the visibility of points given the pose of the object and the camera parameters. We follow up with a discussion of an efficient search technique for nearest neighbor correspondences.

The second problem is how to compute correspondences between model and image points. We describe the use of the k-d tree [8] to perform nearest-neighbor searches in order to efficiently compute these correspondences.

The third problem is how to deal with incorrect correspondences and noise. This closely resembles the classic pose estimation problem. We discuss a solution to localization that is more robust than least-squares estimation, the standard solution for 3D-3D pose-estimation problems.

This paper is organized as follows. In Section 2, we discuss how to efficiently set up the point visibility. In Section 3, we describe a method for efficiently establishing point correspondences. In Section 4, we describe our pose optimization algorithm using M-estimators. In Section 5, we describe some experimental results, in particular computational cost with respect to the number of meshes and the effect on convergence of various M-estimators. Finally, in Section 6, we offer our conclusions.

2 Point Visibility

Before matching a surface point of a model with a point in a range image, it is prudent to first determine if the model point is geometrically visible from the given pose. Since the visibility computation will be performed many times, this computation must be as efficient as possible. For a computation of the visible portions of an object model, there are two standard algorithms from the field of computer graphics: ray-casting and z-buffering. For a surface model composed of n triangles, ray-casting results in $O(n^2)$ ray-triangle intersection tests and z-buffering needs $O(n)$ operation. Since we are using an appropriate model against the actual model, having a perfect visibility computation such as z-buffering is not necessary for our localization algorithm. We can consider the problem in two cases: convex and concave surface visibility. In the following, we present an efficient approximation to solve for convex and concave surface visibility.

2.1 Convex Surface Visibility

We first discuss the simplest case, the convex case(e.g., a sphere). In this case, the visibility of a point is computed by the following Test:

$$visible_{convex}(c_i) = \begin{cases} true & n_i \cdot v_i > 0 \\ false & otherwise \end{cases}$$

where c_i denotes the ith triangle of model, n_i denotes ith triangle's outward pointing normal and v is the viewing direction vector from the camera center of projection to c_i.

2.2 Concave Surface Visibility

We compute point visibility for arbitrary shapes. We can use our previous test for convex points. Since it is inexpensive, the test is first used to check to see whether the point is visible or not under the criterion described in the previous section. Once that has been determined, we can perform more expensive tests to determine whether the point is occluded by another part of the object.

Our localization algorithm is robust to small numbers of errors. Thus we do not need an exact computation for the visibility set. We use a *lookup table* to denote visible and invisible directions with respect to the local point coordinate. The computation of the LUT for each point of the model is relatively expensive; we compute this off-line, a task which can be computed within a reasonable length of time(several minutes) and stored with the model . In on-line time, the system converts the current viewing direction to the local direction and consults the table for visibility of the model point from the current viewing direction.

3 Correspondence

Once the set of visible model points has been computed, we need to efficiently compute the correspondences between these model points and the points in the range image. Although it is difficult to find the correct correspondences without first knowing the pose of the object, our interest is not to establish the correct correspondences, but rather to determine the correct pose of the object. These correspondences are used only for efficient and correct pose estimation.

3.1 Correspondence Estimation

The closest image point (y) to a given model point (x), center of a mesh, can be defined as

$$y = arg \min_{y \in D} \| x - y \|$$

where D is the set of three-dimensional data points in the image.

For efficiently computing these correspondences, we use the k-d tree to perform nearest-neighbor searches. The cost of k-d tree operations is $O(|D|)$. If the 3D space is sparsely occupied by surfaces, we can perform this correspondence test very efficiently. Its worst case is $O(n)$; the expected number of operations is $O(\log n)$.

3.2 Robust Correspondences Estimation

Once the k-d tree is created, using closest points is sufficient for most cases of 3D localization. We discuss an extension of this nearest-neighbor search to make the correspondence search somewhat more robust to initial position errors. One such extension is surface normal similarity; the surface normal of the model point should be similar to the normal of its matching image point. When we use two normals(two unit vectors) $\hat{n_1}$ and $\hat{n_2}$, its distance(Δ_n) is

$$\Delta_n(\hat{n_1}, \hat{n_2}) = \| \hat{n_1} - \hat{n_2} \|$$

Using the angle θ between $\hat{n_1}$ and $\hat{n_2}$, we can rotate $\hat{n_1}$ and $\hat{n_2}$ such that

$$\begin{aligned} \hat{n_1}' &= \mathbf{R}\hat{\mathbf{n_1}} = \left[\begin{array}{ccc} 1 & 0 & 0 \end{array} \right]^{\mathbf{T}} \\ \hat{n_2}' &= \mathbf{R}\hat{\mathbf{n_2}} = \left[\begin{array}{ccc} \cos\theta & \sin\theta & 0 \end{array} \right]^{\mathbf{T}} \end{aligned}$$

Without loss of generality, we can use a matrix $\mathbf{R}$. Now we can simplify Δ_n to

$$\begin{aligned} \Delta_n(\hat{n_1}, \hat{n_2}) &= \Delta_n(\hat{n_1}', \hat{n_2}') \\ &= \| \hat{n_1}' - \hat{n_2}' \| \\ &= \sqrt{(\cos\theta - 1)^2 + \sin^2\theta} \\ &= \sqrt{4 \sin^2 \frac{\theta}{2}} \\ &= 2 |\sin \frac{\theta}{2}| \quad \approx |\theta| \end{aligned}$$

We can see that this is close to linear monotonic in small θ and now θ and Δ_n are directly combined.

Our purpose is to efficiently compute correspondences and proximity. We accomplish this by weighting either the point or the normal. Each data point is stored as the 6D vector($p = [x^T \omega n^T]^T$) and the distance to these points is

$$\begin{aligned} \Delta(\hat{p_1}, \hat{p_2}) &= \| \hat{p_1} - \hat{p_2} \| \\ &= \sqrt{\| x_1 - x_2 \| + \omega \| \hat{n_1} - \hat{n_2} \|} \end{aligned}$$

The weighting factor ω is used to effect the desired constraint on the correspondences. Thus no modification of the k-d tree and nearest-neighbor search technique is required.

4 Pose Optimization

We now discuss a technique for dealing with the computation of pose of a 3D object with respect to observed 3D points. Pose optimization includes the problems of pose estimation and pose refinement. Pose estimation is given a set of correspondences (correct and fixed), while pose refinement, a dynamic operation, is given an image and a rough initial pose estimate(slightly incorrect correspondences).

We begin by considering the problem of pose estimation. We first discuss a method for robust pose estimation, then show the robust pose refinement.

4.1 Pose Estimation

The pose estimation problem is to compute the pose which aligns the 3D model points x_i with their corresponding image points $y_i(i = 1, ...n)$. y_i is specified by the matrix-vector pair $< \mathbf{R}, \mathbf{t} >$ where $\mathbf{R}$ is a 3×3 rotation matrix and t is a 3D translation vector

$$y_i = \mathbf{R}\mathbf{x_i} + \mathbf{t}$$

In general, the sensed points y_i will be contaminated by noise

$$y_i = \mathbf{R}\mathbf{x_i} + \mathbf{t} + \beta$$

Assuming that β follows a normal distribution,

$$P(\beta) \propto e^{-\frac{\beta^T \beta}{2\sigma^2}}$$

then the optimal transformation is the least-squared error solution. The values$< \mathbf{R}, \mathbf{t} >$ that minimize

$$f(R,t) = \sum_i \| \mathbf{R}\mathbf{x_i} + \mathbf{t} - \mathbf{y_i} \|^2$$

The closed-form solution can be used to compute the 3D-3D pose estimation. But if the errors in the observed data are not normally distributed or self occluded, least-squares estimation may be inappropriate. Since the closed form solution is no longer valid, an iterative approach is necessary to solve this problem. And it may be necessary to consider a different objective function which is the optimal estimator with respect to the error distribution of the data. Our localization relates to the pose estimation problem. Local minima is a problem, since we will have a number of incorrect correspondences to deal with. The appropriate objective function will clarify these problems.

The next subsection will describe methods from the field of robust statistics for handling outliers.

4.2 Robust Pose Estimation

Suppose we are given a set of n observed points y_i and corresponding model points x_i, and we want to compute the pose $< \mathbf{R}, \mathbf{t} >$. The problem is that some of the n correspondences will be incorrect; worse, we do not know which ones are incorrect. The errors for these incorrect correspondences do not fit a normal distribution. For a solution to this problem, we check the field of robust statistics.

We now consider M-estimation, which is a generalization of least squares. (There are other classes of robust estimation techniques such as outlier thresholding, median/rank estimation, but these are computationally expensive.)

The general form of M-estimators is

$$E(z) = \sum_i \rho(z_i)$$

where $\rho(z)$ is an arbitrary function of the errors. The equivalent probability distribution to $E(z)$ is

$$P(z) = e^{-E(z)}$$

The M-estimate is the maximum-likelihood estimate of $P(z)$ and our choice of $\rho(z)$ determines $P(z)$.

If we use $\rho(z) = z^2$, then this is least-squares estimation.

$$P(z) = e^{-\sum_i z_i^2}$$

We can find the parameters p that minimize E by taking the derivative of E

$$\frac{\partial E}{\partial p} = \sum_i \frac{\partial \rho}{\partial z_i} \cdot \frac{\partial z_i}{\partial p} = 0$$

By substituting

$$w(z) = \frac{1}{z}\frac{\partial \rho}{\partial z}$$

we get

$$\frac{\partial E}{\partial p} = \sum_i w(z_i) z_i \frac{\partial z_i}{\partial p}$$

If we forget that w is a function of z, this is the same form as if $\rho(z) = wz^2$. And it is recognized as weighted-least squares. In this case, the term $w(z)$ measures the weight of the contribution of errors of magnitude z toward a WLS estimate.

Putting $w(z) = 1$ indicates that each error has equal confidence, regardless of how large the error. This is the pure least squares case. $w(z)$ can be defined that has the same effect as outlier thresholding by

$$w(z) = \begin{cases} 1 & |z| \leq \sigma \\ 0 & |z| > \sigma \end{cases}$$

where σ is the threshold. We believe that the least-squares and thresholding methods are sensitive to outliers. There are many other possible choices of $\rho(z)$ to reduce the sensitivity to outliers on the estimation. Table 1 lists several possible functions that are used for M-estimation in this paper. Gaussian function (least squares) is used for comparison.

Table 1.

Function Name	w(z)
Huber's function	$w = \begin{cases} 1 & \lvert z\rvert \le \sigma \\ \frac{\sigma}{\lvert z\rvert} & \lvert z\rvert > \sigma \end{cases}$
Tukey's function	$w = \begin{cases} (1-(\frac{z}{\sigma})^2)^2 & \lvert z\rvert \le \sigma \\ 0 & \lvert z\rvert > \sigma \end{cases}$
Lorentz's function	$w = (\frac{1}{1+\frac{1}{2}(\frac{z}{\sigma})^2})$
Gaussian	1

4.3 Robust Pose Refinement

Our problem is to solve for the pose p of an object given a set of correspondences between visible object points x_i and image points y_i. Since we know that we will have many outliers in our set of correspondences, we will use a robust M-estimator to solve for p. We will minimize

$$E(p) = \frac{1}{|V(p)|} \sum_{i \in V(p)} \rho(z_i(p))$$

where $V(p)$ is the set of visible model points for the model pose parameters p. $z_i(p)$ is the distance between the ith pair of correspondences. We define

$$z_i(p) = \| \mathbf{R}(\mathbf{q})\mathbf{x_i} + \mathbf{t} - \mathbf{y_i} \|$$

where $\mathbf{R}(\mathbf{q})$ is the 3×3 rotation matrix for the rotation component of p and t is the translation component.

The corresponding image points y_i are a function of p. Now y_i is

$$\begin{aligned} y_i &= y(x_i, p) \\ &= arg \min_{y \in D} \| \mathbf{R}(\mathbf{q})\mathbf{x_i} + \mathbf{t} - \mathbf{y} \| \end{aligned}$$

We use the gradient-descent search to correctly minimize the desired function $E(p)$.

5 Experimental Results

5.1 Experimental procedure

We performed experiments using range images of an insulator in electric distribution systems. The insulator's height and width were $180mm$ and $110mm$, respectively. Our range sensor consisted of a light-striping projector and a SONY 3CCD color camera [9].

The first step in our experiments was building the model for the insulator. We took 10 range images of the insulator from various viewing direction, and converted them to mesh models. Using the Zipper program [10], those mesh models were aligned manually into a standard coordinate system and merged into a uniform mesh model At each mesh, the visibility lookup table was constructed. Figure 2 shows some of the range images and the merged mesh model.

Figure 3 is the intensity view and an off-center view of the 3D range data points. Figure 4 is a typical initial starting point for the convergence tests(variations of translation error) and an overlay of the insulator model at its estimated actual location. This image contains 512×480 pixels and each pixel contains a 3D point. Each experiment consists of 100 trials of the 3D-3D localization algorithm from a randomly generated initial pose estimate. We manually verified the results of each trial and determined the number of correct trials. We evaluated the performance of the algorithm with respect to two aspects: computational cost vs. the number of image and model points and effect of M-estimator on convergence.

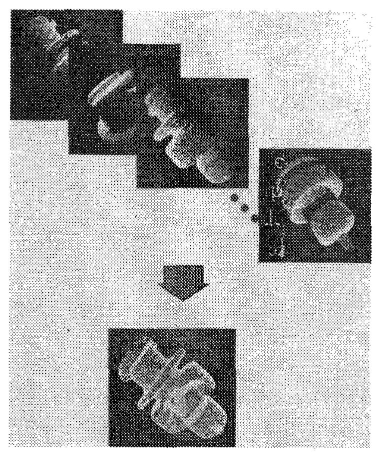

Figure 2: Range images and the merged model

5.2 Computational Cost

We evaluated the effect of the number of model and image points. Figure 5 shows the original model with 30017 meshes and one with 6975 meshes deci-

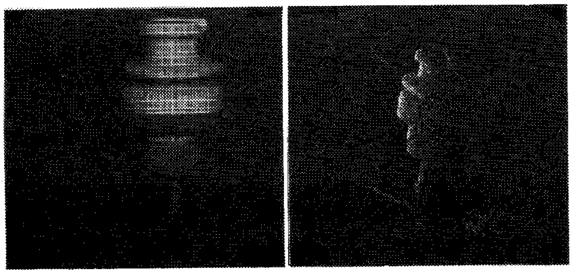

Figure 3: Intensity image view(left) and off-center view of the 3D range data points(right)

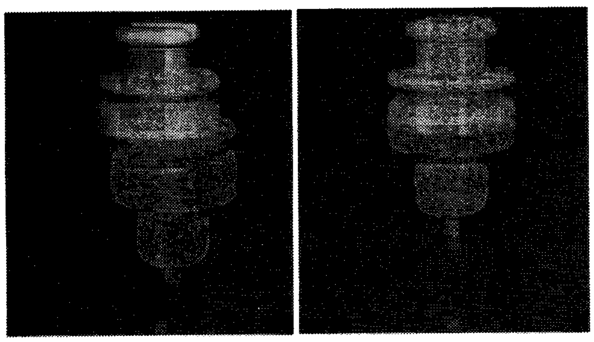

Figure 4: A typical initial starting point(translation 50 mm) and an overlay of the insulator model at the estimated location

mated from the original model using Schroeder algorithm [11]. Figure 6 shows the computational times along the number of points. As expected, the number of image points affects the computational time in a $O(\log n)$ relation; this is because the image points are structured into the k-d tree of which retrieval time is $O(\log n)$.

5.3 Effects of M-estimators

Four versions of our 3D-3D localization algorithm were tested; these Versions differ only in the M-estimators, Huber, Tukey, Lorentzian and Gaussian. Figure 7 shows that we began with $\sigma = 4mm$ and reduce it to $\sigma = 2mm$ and then to $\sigma = 1mm$ as the algorithm progressed. We also reduced σ from $12mm$ to $3mm$ in figure 8. The Lorentzian and Huber weight functions perform better than two other functions and have much better convergence properties. We were successful in obtaining more than 75% correct poses

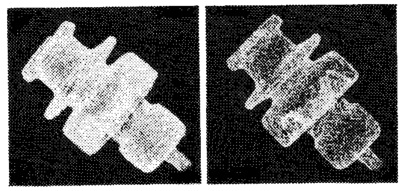

Figure 5: Original model(left) and decimated model(right)

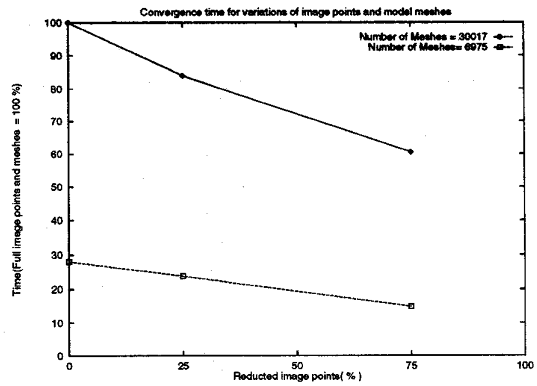

Figure 6: Convergence time for variations of image points and model meshes

for translations below $75mm$. A large number of σ were successful with large initial pose errors.

6 Conclusion

We have developed a localization algorithm for a dual-armed mobile robot. The key components of our algorithm are: a point visibility table for efficient visible computation, a k-d tree-based point correspondence algorithm, and a robust pose estimator with M-estimator. We have verified effectiveness of our algorithm under various initialization errors. We have also examined the effect of M-estimators.

Experimental results demonstrate that using dynamic correspondences within a gradient-descent search of a robust objective function is a key component in achieving this capability. In particular our algorithm achieves a wide degree of convergence for object localization in noisy image data.

Currently, we conduct our experiments indoors. We plan to conduct further experiments in outdoor envi-

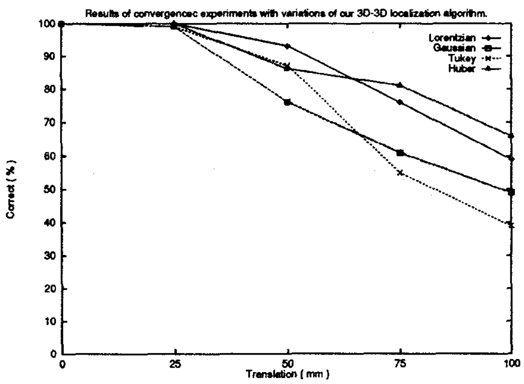

Figure 7: Results of the convergence experiments with variations of our 3D-3D localization algorithm ($\sigma = 1$)

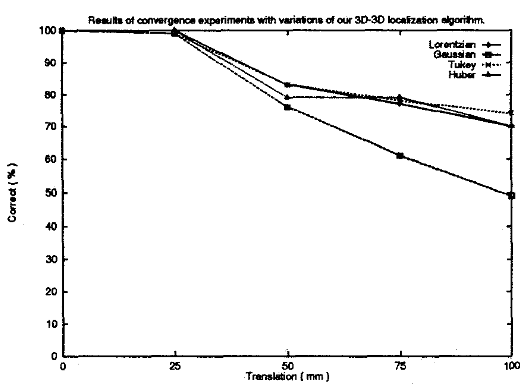

Figure 8: Results of the convergence experiments with variations of our 3D-3D localization algorithm ($\sigma = 3$)

ronments to verify the performance of our algorithm under a number of illumination conditions caused by sunlight and background clutter caused by nearby buildings.

References

[1] M. Nakashima, H. Yakabe, M. Maruyama, K. Yano, K. Morita and H. Nakagaki, "Application of Semi-Automatic Robot Technology on Hot-Line Maintenance Work," *IEEE Conf. on Robotics and Automation*, 843-850, 1995.

[2] Paul J. Besl and Neil D. McKay, "A method for registration of 3-d shapes," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 14(2):239-256, 1992.

[3] Zhengyou Zhang, "Iterative point matching for registration of free-form curves and surfaces," *International Journal of Computer Vision*, 13(2):119-152, 1994.

[4] Robert M. Haralick, Hyonam Joo, Chung-Nan Lee, Xinhua Zhuang, Vinay G. Vaidya, and Man Bae Kim, "Pose estimation from corresponding point data," *IEEE Transactions on Systems, Man, and Cybernetics*, 19(6):1426-1446, 1989.

[5] David G. Lowe, "Robust model-based motion tracking through the integration of search and estimation," *International Journal of Computer Vision*, 8(2):113-122, 1992.

[6] M. A. Fischler and R. C. Bolles, "Random sample consensus:A paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, 24(6), 1981.

[7] Peter Meer, Doron Mintz, and Azriel Rosenfeld, "Robust regression methods for computer vision:a review," *International Journal of Computer Vision*, 6(1):59-70, 1991.

[8] J. H. Friedman, J. L. Bentley, and R. A. Finkel, "An algorithm for finding best matches in logarithmic expected time," *ACM Transactions on Mathematical Software*, 3(3):209-226, 1977.

[9] K. Sato and S. Inokuchi, "Range-imaging system utilizing nematic liquid crystal mask," *Proceedings of 1st ICCV, London*, 657-661, 1987.

[10] Greg Turk and Marc Levoy, "Zippered polygon meshes from range images," *Proceedings of SIGGRAPH 94*, 311-318. ACM, 1994.

[11] William Schroeder, Jonathan Zarge, and William Lorensen, "Decimation of triangle meshes," *Proceedings of SIGGRAPH 92*, 65-70, 1992.

[12] Mark D. Wheeler, "Automatic Modeling and Localization for Object Recognition," *PhD thesis*, Carnegie Mellon University, 1996.