N90-29020

Methods and Strategies of Object Localization

Lejun Shao[1] and Richard A. Volz[2]

[1]Robot Systems Division, Department of Electrical Engineering and Computer Science, The University of Michigan, Ann Arbor, Michigan 48109.

[2]Department of Computer Science, Texas A&M University, College Station, Texas 77843.

Abstract

An important property of an intelligent robot is to be able to determine the location of an object in 3-D space. This paper proposes a general object localization system structure, discusses some important issues on localization and gives an overview of current available object localization algorithms and systems. The algorithms reviewed in the paper are characterized by their feature extracting & matching strategies, the range finding methods, the types of locatable objects and the mathematical formulating methods.

1 Introduction

Since the early 1980's, the use of robots in industry has become increasingly popular or even crucial in some areas. Through practice, researchers have realized that an important property of an intelligent robot is the ability to determine the location of stationary and moving objects. For example, space station construction, repairing, maintenance, satellite refueling, etc. have been identified as the potential areas of applications of telerobotics systems. The availability of an efficient means of locating objects is one of the key factors to the success of developing such systems.

Object localization has long been defined as a part of object recognition process in computer vision research [4]. But in most instances, the emphasis of the research is on object recognition. Object localization is only a by-product. In robotic applications, however, object localization usually is the ultimate goal. And, it has many of its own problems to be solved such as real-time considerations, accuracy issues, types of locatable objects, working conditions *etc.*, which object recognition research generally does not address. In some systems, "**locate**" has been defined as one of the basic independent operations the telerobot system is to perform [27]. As a result, object localization research has attracted increasing attention recently.

This paper will give an overview of the three-dimensional object localization problem. First, it provides a closer examination of the problem and then proposes a general object localization system structure. Some important issues of object localization and the possible implementations of key components of the proposed object localization system are discussed and compared. A summary is presented in the final section.

2 The Object Localization Problem

As we have mentioned, object localization is the determination of the location of an object. What must be solved when a robot vision system is trying to locate an object? A necessary component of every intelligent robot system is the world modeling system which stores, among other things, a representation of all the object models that are relevant to the robot's operation and a definition of the sensing coordinate system. To specify an object in the world modeling system, there must exist either an implicit or explicit coordinate system which is associated with that object.

The real problem to be solved when locating an object, therefore, is to determine the relative location between the sensing coordinate system and the coordinate system for the object, which is somewhat different from what the object localization means from the point of view of object recognition. In object recognition, localization can mean a description of the location relationships among the objects to be recognized.

The relative location of two coordinate systems can be specified in any one of the following methods:

1. *Position and orientation:*

   The position can be specified by three parameters, e.g., the (x,y,z) coordinates of the origin of the object coordinate system relative to the sensing coordinate system. There are three different representations of orientation:

   - Three Eular angles α, β, γ, or angles about the coordinate axes.
   - A unit vector $\mathbf{r}$ and an angle $\boldsymbol{\theta}$.
   - A quaternion. [22, 26]

2. *A* 4×4 *homogeneous transformation matrix.*

3. *Dual number quaternion:* [37]

   This is an extension of quaternion representation in which each quantity is changed to a dual quantity [11]. The dual quaternion has a similar interpretation as the real quaternion:

   $$\hat{q} = \begin{bmatrix} \sin(\hat{\theta}/2)\hat{n} \\ \cos(\hat{\theta}/2) \end{bmatrix}$$

   where the vector $\hat{n}$ is a unit line vector about which the coordinate system has rotated and translated and $\hat{\theta}$ is the dual angle of rotation and translation.

The objective of the object localization algorithm is to compute the parameters which specify the corresponding representation.

Because there are six degrees of freedom a rigid object could have in three dimensional space (three for position, three for orientation), six independent parameters is the minimum number to be determined. The advantage of position/orientation representation is that it has minimum or near minimum number of variables. But there are disadvantages. The angular representations use trigonometric functions which are of infinite order and lead to a nonpolynomial criterion. Vectorial representations also have singularities; when the rotation angle θ is zero, the axis of rotation is arbitrary. The matrix representation is linear and has no singularity problem. But the inherent redundancy for rotation leads to a high-dimensional space constraints and will make the computation a little harder. The dual number quaternion representation has a dimension of eight, which is a little bit higher than the minimum but is still quite simple to compute.

In some applications, the localization problem can be simplified due to extra constraints imposed on the object. For example, if an object is so constrained that a planar surface of the object is always lying on a plane, the degrees of freedom of the object are reduced to three: one for the rotation and two for the translation.

How does one solve for these position/orientation parameters if sensor data and object models are given? Usually the computation is carried out by a matching process. That is, the object localization algorithm will try to find a "best" transformation which will put sensed features into its corresponding model features.

From the above description, it is not difficult to imagine that a general object localization system should contain the following components: (1) **sensing system**: to provide necessary measurements;

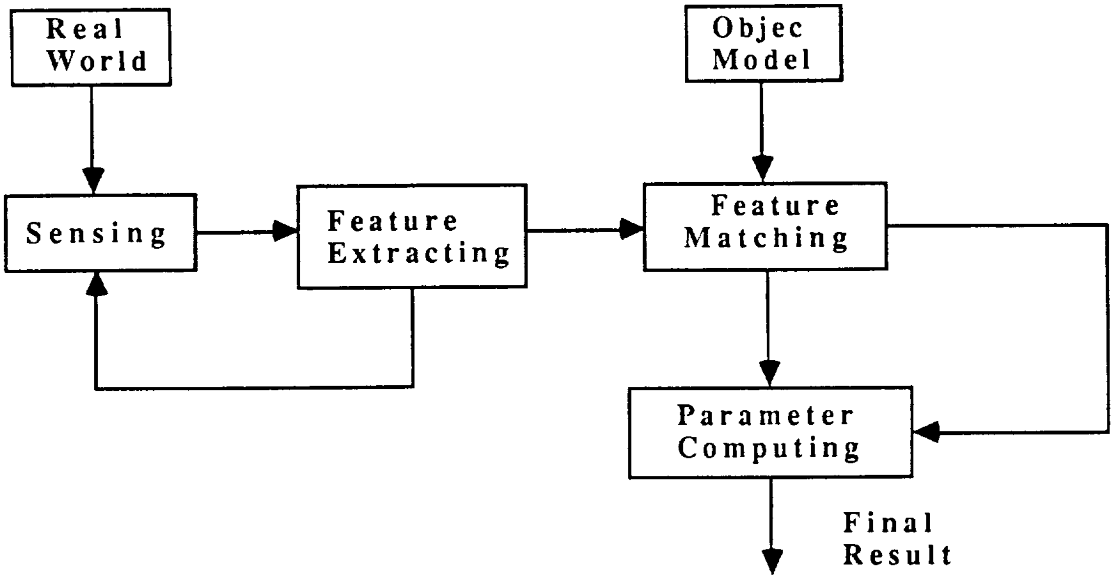

Figure 1: Object localization system organization

(2) **world model**: to represent all the objects including robot and sensors and their relationships; (3) **feature extraction**: to retrieve features which is to be used in the matching process; (4) **matching**: to try to pair the sensed features with corresponding model features; and (5) **computing**: to calculate the transformation parameters. See Fig. 1 for the configuration.

Based on the types of sensors used in the sensing system, the sensing method can be divided into serial sensing and parallel sensing. If the necessary sensor data is obtained through a series of measurements, such as the case when a spot range sensor is used, it is called as serial sensing; if the sensor data can be obtained by a single measurement, it is called as parallel sensing. In the case of serial sensing, the whole feature extraction might go through a repeated sensing-extraction process. The geometric features to be extracted and to be matched can be classified as low-level features and higher-level features. Possible low-level features include points, vectors, line segments, axes, surface patches, edges, boundaries and *etc.*. Possible high-level features include straight dihedrals, circular dihedrals [9], principle directions of surface curves, minimum, maximum and mean curvatures of surfaces, Gaussian curvatures, and *etc.*. Usually the lower the level of features is, the greater the number of features to be extracted.

The matching process is the process of finding the pairings of sensed features and the model features. Depending on the level of intelligence of the system, the matching could be done in different ways. On the lowest level, there is no matching process at all in the system. Whenever a measurement is taking place, either a default matching is assumed or a man-assisted matching is provided. On telerobotics systems, for example, the teleoperator might interactively assist the model matching by indicating with a light pen which features in the image (e.g. edges, corners) correspond to those in a stored model [1]. On higher levels, the system will be able to paring the features automatically. Table 1 shows some known feature matchings which have been used in literature to derive the location of an object. Sometimes, a combination of feature matchings are necessary to completely specify a rigid

Measured features	Matched to
Point	Point
	Planar surface
	Surface patch
Surface normal	Surface normal
Line segment	Line segment
	Planar surface
	Surface patch
Edge	Edge
Planar surface	Planar surface
Quadric surface	Quadric surface
Gaussian curvature	Gaussian curvature

Table 1: Known Matching Strategies in Object Localization

transformation.

We have just showed and discussed a general object localization mechanism. There is no common solution for the implementation now. Each component could be implemented in many different ways. Some components or relations in this mechanism may be unnecessary in certain implementations. In the next two sections , a further discussion about sensing system and feature extracting & matching strategies will be given.

3 Some Issues

We just showed a general structure of object localization systems. In practice, there are some important issues which must be considered when a real localization system is to be designed.

1. *Real-time execution*: **Hierarchical** control structure has been defined as a standard for telerobot control system architecture [1] and has been adopted by researchers to develop individual telerobot systems such as systems developed at Goddard [27], University of Michigan [36] and *etc.*. The functions of vision system are different at each level. So are the requirements for the object localization algorithms. Usually the higher the level, the slower the completion rate. See table 2 for typical completion rates at each level of telerobot control.

   At the object task planning level, for example, one of the functions of vision system is to recognize the environment. The object localization system, as a part of the vision system, is used to give an approximate measurements of the locations of the objects in the environment. The execution time is in the minute range. At the E-move level, however, the rate of completion is in the range of seconds. If a visual-feedback control strategy is used here, the localization system has to generate updated measurements for the control system to adjust the robot's movement in the same time frame. Real-time issue will become important. Based on different timing requirements, the strategies of localization might be also different.

2. *Accuracy*: Accuracy is another important issue in object localization. There are two definitions of accuracy, e.g., **absolute accuracy** ϵ and **relative accuracy** $\Delta\epsilon$.

   - *Absolute accuracy* is defined as the difference between a measured value m and it's real value s. That is, $\epsilon = m - s$;

	Average rate of change in output	Average replanning interval	Planning horizon
Servo	1 KHz	1 millisec.	15 msec.
Primitive	62Hz	16 millisec.	300 msec.
E-Move	8Hz	128 millisec.	2 sec.
Object/task	1 Hz	1 second	30 sec.
Service Bay	.1 Hz	10 second	> 10 min.
Mission	0.01 Hz	1.7 minutes	> 1 hour

Table 2: The rate of subtask completion at each level of hierarchy. ([1])

- *Relative accuracy* is defined as the difference between a measured difference and it's real difference. For example, if the real difference between two points is $\Delta\rho$ and the two measurements on the two points are s_1, s_2, then the measured difference is a function of s_1 and s_2, e.g., $\Delta\rho' = f(s_1, s_2)$, and the relative accuracy of the measurements is $\Delta\epsilon = \Delta\rho - \Delta\rho'$.

High accuracy, especially high absolute accuracy is not always required. For example, accuracy is not crucial at the beginning of an assembly task, but will be a determining factor in the final stage of operation. Even at that time, the determining factor is relative accuracy rather then absolute accuracy.

Absolute accuracy to a large extent depends on the accuracy of the sensing system. But this does not mean that we do not need a good localization algorithm. A good algorithm should be insensitive to measurement noises, object distortions and other factors which could influence the accuracy of the localization.

The achievement of high relative accuracy, on the other hand, does not necessarily depend on high accurate sensing systems. Human eyes, for example, are not good at locating objects in the absolute sense, but human have no difficulty picking up an object. Research is shown also proved this point of view [25]. Therefore, when designing an algorithm, one must evaluate its performance according to both it's absolute accuracy and relative accuracy, which has been neglected by some researchers.

3. *The type of locatable objects*: It is best if the system can locate arbitrary-shaped objects. If this is difficult, an alternative method is try to find specific detectable features for each object and store these features in that object model for feature-extraction and matching in localization process. If such features do not exist for some object, then one should try to make special marks on the object. Therefore, some guidelines should be given in the component design stage so that the design is favorable to part grasping and localization by the telerobot system. Sometimes, very simple modifications made on the part design can greatly improve part localization process.

4. *Sensing system*: What types of sensing techniques should be used in a localization system? Where should one install the sensing system? How is the dynamic range of sensing system determined? These are just some of the issues when one needs to design a sensing system. Javis [20] has presented an early overview of range finding techniques. Each technique has its advantages and disadvantages. Image-based sensing provides complete information about the environment but takes time to process it. Sparse data can be used directly for fast localization

purpose but needs a good sensing system for fast data acquisition. Multi-spot sensing is an example of such system [21]. For the installation, perhaps some sensors should be installed in fixed locations while others can be put into the robot's moving parts. Newly developed technologies should be used in localization systems. For example, motorized-zoom and auto-focus techniques can improve the dynamic range of measurements; VILS techniques can reduce the size of the whole sensing system. The use of the advanced techniques will have great impact on the design of object localization system.

4 Feature Extraction & Matching Strategies

As we said, the object localization process basically is a feature matching process. That is, finding a best estimate of transformation parameters which will align some modeled object features with certain (perhaps different type of) measured object features. Based on how feature matching is realized, the object localization algorithms can be broadly divided into two categories: the algorithms which do not involve any recognition process, and those which have more or less recognition process involved. We call these two types of algorithms as direct-localization algorithms and recognition-localization algorithms respectively.

Obviously the second type of algorithms has a higher intelligent level than that of the first ones. Even within the second type of algorithms, the intelligent levels could be different. Some of them can establish the matchings within one object, some of them can do it within a group of the same type of objects, others can match the features within a group of different types of objects. At the highest level, the algorithm could locate unmodeled objects. To do this, a set of primitive features should be specified in a database, which will form the basic frames of any object to be constructed. Before localizing the unknown object, the algorithm must explore the object and establish a model for the object using the set of primitives.

Each type of algorithm can be further classified according to their sensing methods, the types of features used for matching, mathematical formulating methods, the types of locatable objects and so on.

4.1 Direct-Localization

Direct-localization algorithms are mostly used in the situations where either the working environment is a highly-structured or the position relationships among the objects in the environment have previously been established proximately or the human beings could provide the assistance as where to take the required measurements. The telerobotics applications in most space programs meet these requirements.

Because no recognition is involved, the localization process is quite simple. The extracted features and model features can be used as inputs for direct computation. The time of localization depends on the time spent on measurements and feature extraction.

One method proposed by Gunnarsson and Prinz [18, 19] is based on their observation that if a set of points are measured and these measurements are distributed on the object surfaces, the best transformation is the one which will make the sum of distances between each measured point and it's corresponding transformed surface minimal. Their idea leads to a point-surface matching strategy. Their algorithm, when formulated in mathematical terms, becomes a least squares minimization problem and can be used to locate arbitrarily-shaped objects. Usually an iterative numerical procedure is needed to solve for the problem. The numerical procedure they used is a modified Lagrange multiplier and Newton-Raphson method. Because a good initial guess can be provided due to the fact that the object's approximate location is supposed known, the convergence of the algorithm is guaranteed in most cases.

Gordon and Seering [17] developed a system which uses striped-light and camera sensing to gather necessary range data. The system can only locate planar objects. Line-surface matching is used in their algorithm. The striped-light when projected on the planar surfaces of the object generates straight-line segments. The scene is then viewed by a camera. The equation of each line segment can be obtained by analyzing the corresponding image of that line segment viewed by the camera. Three independent line segments are needed to compute the rotation and translation parameters. The fact that the line vector is perpendicular to the rotated modeled surface normal vector can be used to derive the rotation. The algorithm uses quaternions to represent rotation and uses a numerical method to compute it. They also give a closed-form solution for the rotation when three surfaces sensed are perpendicular with each other. The calculated rotation is then used to compute the translation.

The same striped-light and camera sensing system is also used by Rutkowski, Benton and *etc.* [3, 28]. But their matching strategy is point-surface matching. In their algorithm, the measured points are from extracted line-segments, either straight or curved. Their method imposes no particular constraints on the shapes of the object surfaces, as long as the object surfaces can be partitioned into a collection of primitive surfaces, such as planes, cylinders, or spheres. The computation is carried out by a repeated location adjustment. The location adjustment is expressed by three quantities: the rotation center, rotation axis and translation vector. To guarantee a fast convergence of their algorithm, the center of mass data points is chosen as the rotation center instead of the origin of the model's coordinate system.

In above methods, if the object is a polygon, at least three surfaces need to be accessed in order to take enough measurements. Shao, Volz and *etc.* [29] have implemented an algorithm based on line-segment line-segment matching, which needs to access only one surface when localizing a planar object. Their algorithm can locate object which has planar surfaces, quadric surfaces and revolutionary surfaces. A line range sensor is used to extract line-segment parameters. The line-segments are either boundary edges or axes. The extraction of only two line-segments are enough to locate an object. Closed-form formulas are used to compute the position and orientation parameters.

When comparing with these methods, we can find out that all of them have very high measurement accuracy and fast execution speed. For example, Gordon's system has 2.5 seconds of execution time and a relative accuracy of 0.002 inches in translation and 0.1 degrees in rotation when a two inch cube is being located, and is capable of reliably assembling components with little clearance without using force controlled motion. In Gunnarsson's algorithm, the measurement error is on the same order of magnitude as the sensor error. These algorithms also have some problems. The problem associated with stripped-light sensing is that it requires extra light source with special pattern, which sometimes is inconvenient. The use of spot sensor or line sensor has the problem of multi-measurement, e.g., the sensor has to be installed on the robot's moving part and be moved together with the robot in order to take multi-measurement. This will slow down the localization process.

High-level features can also be used to locate objects. For example, Thorne and *etc.* [35] described an algorithm which uses features such as the radii or curvatures of a space curve along the curve to locate an object. The curvatures k or radii ρ of a space curve can be expressed as a function of the length s of the curve, e.g., $k = \kappa(s)$ (or $\rho = p(s)$), which is independent of the coordinates of the curve and is thus invariant under rotation and translation. The algorithm assumes that there exists a particular feature line or *fingerprint* for each object. The feature line could be a certain portion of the curved edge(s) of the object or a curve on the object surface. A curvature plot along the feature line can be drawn. In the database, the feature line is specified by a set of discrete points with each point associates with the information about its coordinate (x, y, z), radius of curvature, curvature, delta length, and total length. The total length is zero for the first point. The localization is proceeded through point-point matching. The method first measures a set of discrete points along the feature line and then finds a corresponding point for each measured point and a least squares optimization algorithm is used to find the location parameters. A similar algorithm which uses iso-gaussian (a curve connecting points of constant gaussian curvature) matching to localize an object was described by Gunnarsson [18].

4.2 Recognition-Localization

In many applications, the objects could be placed anywhere in the environment. Therefore, if a measurement is made and some sensed features are extracted, the localization system has no prior knowledge about which object or which part of the object the sensed features should belong to. In this case, in order to compute the location of an object, a recognition process is needed, which will establish the matchings between a set of sensed features and the model features.

There are two popular matching strategies: *tree searching* and *clustering*.

In tree searching strategy, if there are k sensed features $S_i, i = 1 \cdots k$ and l_m model features $M_j, j = 1 \cdots l_m$ for object $O_m, m = 1 \cdots w$, a searching tree can be constructed for each known object O_m such that the tree has l_m levels, and each intermediate node has k branches. Each path from root to leaf represents a potential matching. The total number of possible matchings, or the searching space for object O_m is l_m^k, which is very huge. To reduce the searching space, several methods have been proposed.

One algorithm proposed by Grimson, Lozano-Perez and *etc.* [15] [16] is to use the local geometrical constrains such as distance constraint, angle constraint, direction constraint, triple-product constraint and so on to reduce the searching space. Beginning from the root of the tree down, at each node, local constraint test is made to see if the sensed features up to that level are consistent with these constraints. If it is not, the entire subtree is discarded for consideration.

A similar tree searching method is used in Faugeras and Hebert's work [12, 13, 14]. Instead of local constraints, rigidity is used as the basic constraint during tree search process. Every path from the root to an intermediate node (level k for instance) represents a partial matching. The algorithm computes a best rigid transformation $\mathbf{T}_k$ up to that level (k). Then $\mathbf{T}_k$ is applied to the next unmatched model primitive M_{k+1} and only those sensed primitives that are sufficiently close to $\mathbf{T}_k M_{k+1}$ are considered. The computations are carried out by least squares optimization techniques. As each new pair of primitives adds to the partial matching list, the new estimation of transformation has to be started over again. The algorithm's underlying paradigm is "*locating while recognizing*" which is different from the paradigm of "*locating after recognizing*" used in Grimson and *etc.*'s algorithm.

Reducing the number of sensed and model features is another important method to speed up the tree searching process. The use of higher level features can effectively reduce the size of the searching tree because fewer features are usually adequate. The system developed by Bolles, Horaud and *etc.* [9, 10] is such an example. Three different types of edges are used as the primitive features. They are: straight dihedrals, circular dihedrals, and straight tangentials. They are higher level features: one pair of matched features can determine all but one of the object's six degrees of freedom.

Clustering is another technique used in recognition-localization algorithms. The principle of clustering is very simple:

```
For each element in the sensed feature list
    for each element in the model feature list
        if they are compatible, compute a transformation candidate
        put it into cluster space.
```

The cells with the largest counts are expected to represent the location.

While the principle is simple, the implementation is not so easy. The high dimensions (six) and huge space of clustering are just two difficulties. Different methods have been proposed to accommodate these problems. Three dimensional clustering, the use of proper size of cells and hierarchical clustering are some of them [2]. Several systems have been proposed by using the clustering technique. Linnainmaa *etc.* [23, 24, 25], Silberberg, Harwood, *etc.* [31], and Stockman *etc.* [32, 33, 34] are typical examples. One property of clustering is the algorithm's parallel structure, which will have an important impact on the future development of object localization algorithms.

In most algorithms, the least squares optimization is the mathematical tool to estimate the best transformation if many feature-pairs are found. But in many situations, this method is not the only tool. Bolle, Cooper [5] [6, 7, 8] presented a statistics approach of combining pieces of information to estimate 3-D complex-object position. They formulate the optimal object localization as a Bayesian probability estimation problem. The objective is to find the most likely transformation $\mathbf{T}$ that maps the model primitives onto the measured range data. The likelihood $p(Y|\mathbf{T})$ should be maximized with respect to $\mathbf{T}$, where Y is the measurement data. If κ primitives have been extracted from range data and matched to model primitives, then $p(Y|\mathbf{T}) = \prod_{k=1}^{\kappa} p(Y_k|\mathbf{T})$. That means, to arrive a global optimal solution, the maximum likelihood estimation has to be applied locally. Based on this analysis, they arrived at a different formula for minimizing the estimation error from the traditional least squares optimization formula. To arrive an optimal solution, a through analysis of measurement errors and having a good error model are needed.

5 Summary

We have discussed several object localization methods and strategies. Different levels of telerobot control have different requirements on the localization system. At the low level, the consideration of real-time execution and high accuracy is important. At higher level, the use of AI (artificial intelligence) technology becomes crucial. It seems that a lot of work has to be done in order to develop a real practical localization system. The issues discussed in section 3 are just few of those which need to be addressed by future research.

References

[1] Jamas S.Albus, Harry G. Mccain, and Ronald Lumia, *"NASA/NBS Standard Reference Model for Telerobot Control System Architecture (NASREM)"*, National Bureau of Standards, Robot System Division, March 13, 1987, pp.12.

[2] D.H. Ballard *"Parameter Networks: Towards a Theory of Low-Level Vision"*, Int'l Joint Conf. on Artificial Intelligence, Vancouver, B.C., August 1981, pp.1068-1078.

[3] R. Benton and D. Waters, *" Intelligent Task Automation Interim Technical Reports"*, AFWAL/MLTC Wright Patterson AFB, Ohio, Reps. 1-7 Oct. 1985.

[4] Paul Besl and Ramesh Jain, *"An Overview of Three-Dimensional Object Recognition"*, Univ. of Michigan, December 1984, Tech. Rep., RSD-TR-19-84.

[5] Ruud M. Bolle, *"Information Extraction About Complex Three-Dimensional Objects From Visual Data"*, Ph. D. Dissertation, Brown Univ., May 1984, Brown Univ. Tech. Rep., LEMS-6.

[6] Ruud M. Bolle and David B. Cooper, *"On Optimally Combining Pieces of Information, with Application to Estimating 3-D Complex-Object Position from Range Data"*, IBM Tech. Rep. RC 11037, Brown Univ. Tech. Rep. LEMS-8, Feb. 1985.

[7] -, *"On Parallel Bayesian Estimation and Recognition for Large Data Sets with Application to Estimating 3-D Complex-Object Position from Range Data"*, Proc. SPIE Conf. Vision for Robots, Cannes, France, Dec. 1985.

[8] -, *"On Optimally Combining Pieces of Information, with Application to Estimating 3-D Complex-Object Position from Range Data"*, IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. PAMI-8 No.5, Sept. 1986, pp. 619-638.

[9] R. B. Booles, P. Horaud, and M. J. Hannah, *"3DPO: Three Dimensional Part Orientation System"*, Robotics Research: The First Symposium, Cambridge, MA: MIT Press 1984.

[10] P. Horaud, and R. B. Booles, *"3DPO's Strategy for Matching three-Dimensional Objects in Range Data"*, Proc. IEEE 1984 Int'l Conf. Robotics, Atlanta, GA, March, 1984, pp. 78-85.

[11] W.K.Cliford, *"Preliminary Sketch of Bi-Quaternions"*, London Math. Soc., 4:381-395, 1873.

[12] Faugeras, O.D. and Herbert, M., *"A 3-D Recognition and Positioning Algorithm Using Geometrical Matching Between Primitive Surfaces"*, Proc. Eighth Int'l Joint Conf. on Artificial Intell., Aug. 1983, pp.996-1002.

[13] Faugeras, O.D., *"New Steps Toward a Flexible 3-D Vision System for Robotics"*, Proc. 8th Int'l Joint Conf. Pattern Recognition, Montreal, Canada, July-Aug., 1984, pp. 796-805.

[14] Faugeras, O.D. and Herbert, M., *"The Representation, Recognition, and, Locating of 3-D Objects"*, The Int'l Journal of Robotics Research, Fall 1986, pp.27-52.

[15] Peter C. Gaston and Tomas Lozano-Perez, *"Tactile Recognition and Localization Using Object Models: The Case of Polyhedra on a Plane"*, IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. PAMI-6, No.3 May 1984.

[16] W.Eric L. Grimson and Tomas Lozano-Perez, *"Model-Based Recognition and Localization from Sparse Range or Tactile Data"*, The Int'l Journal of Robotics Research, Fall 1984 pp.3-35.

[17] Steven J. Gordon and Warren P. Seering, *"Real-Time Part Position Sensing"*, IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. PAMI 10, No.3 May 1988.

[18] Kristjan T.Gunnarsson, *"Optimal Part Localization by Database Matching with Sparse and Dense Data"*, Ph.D. dissertation, Dept. of Mechanical Engineering, Carnegie Mellon Univ., May 1987.

[19] Kristjan T. Gunnarsson and Friedrich B. Prinz, *"CAD Model-Based Localization of Parts in Manufacturing"*, Computer, August 1987, pp.66-74.

[20] R. A. Jarvis *"A Perspective on Range Finding Techniques for Computer Vision"*, IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. PAMI-5, No.2, March 1983, pp. 122-139.

[21] Takeo Kanade, *"Three-Dimensional Machine Vision"*, Kluwer Academic Publishers, 1987.

[22] P. Kelland and P.G. Tait, *"Introduction to Quaternions"*, Macmillan and Co., 1882.

[23] S. Linnainmaa, D. Harwood, and L.S. Davis, *"Pose Determination of a Three-Dimensional Object Using Triangle Pairs"*, CAR-TR-95, Center for Automation Research, University of Maryland, 1985.

[24] -, *"Triangle-Based Pose Determination of 3-D Objects"*, Int'l Conf. on Pattern Recognition, 1986, pp. 116-118.

[25] -, *"Pose Determination of a Three-Dimensional Object Using Triangle Pairs"*, IEEE Trans. on Pattern Analysis and Machine Intelligence, Volume 10, No. 2, September, 1988, pp. 634-648.

[26] Edward Pervin and Jon A. Webb, *"Quaternions in Computer Vision and Robotics"*, Tech. Rep., Dept. of Computer Science, Carnegie-Mellon University, CMU-CS-82-150, 1982.

[27] *"Robotic Assessment Test Sets: Level 1, 2, & 3"*, NASA Gorddard, SS-GSFC-0029.

[28] W.S. Rutkowski, R. Benton, and E.W. Kent, *"Model-Driven Determination of Object Pose for a Visually Served Robot"*, IEEE Int'l Conf. Robotics and Automation, Raleigh, NC, 1987, pp.1419-1428.

[29] L. Shao, R.A. Volz, and M.W. Walker, *"3-D Object Location Determination Using Line-Segments Matching"*, IEEE Int'l Conf. Industrial Application of Machine Vision, Tokyo, Japan, 1989.

[30] S. Shekhar, O. Khatib and M. Shimojo, *"Sensor Fusion and Object Localization"*, IEEE Int'l Conf. Robotics and Automation, San Francisco, CA, 1986, pp.1623-1628.

[31] T.M.Silberberg, D. Harwood, and L.S. Davis, *"Object Recognition Using Oriented Model Points"*, Computer Vision and Image Processing, Vol. 35, 1986, pp. 47-71.

[32] G. Stockman and J.C.Esteva, *" Use of Geometrical Constrains to Determine 3-D object Pose"*, Proc. 7th Int'l Conf. on Pattern Recognition, Montreal, Canada, 1984, pp. 742-744.

[33] G. Stockman and J.C.Esteva, *" 3-D Object Pose from Clustering with Multiple Views"*, Pattern Recognition 3, 1985, pp. 279-286.

[34] George Stockman, *"Object Recognition and Localization via Pose Clustering"*, Computer Vision, Graphics and Image Processing Vol. 40, 1987, pp.261-387

[35] H. Thorne, F.B. Prinz, and O.K.Kirchner, *"Robotic Inspection by Database Matching"*, Tech. Rep., CMU-RI-TR-85-4, The Robotics Institute, Carnegie Mellon University, March, 1985.

[36] M.M. Walker, J. Dionise *"On the Simulation of Space Based Manipulators with Contact"*, NASA conf. on Space Telerobotics, January 1989.

[37] M.W. Walker, L. Shao and R. Volz *"Optimal Object Localization Using Dual Number Quaternions"*, IEEE Int'l Conf. on Applications of Artificial Intelligence, March 1989.