Extraction of fine motion through multiple observations of human demonstration by DP matching and combined template matching

Koichi Ogawara Jun Takamatsu Hiroshi Kimura[†] Katsushi Ikeuchi

Institute of Industrial Science, The Univ. of Tokyo
Tokyo, 153-8505, Japan
E-mail {ogawara, j-taka, ki}@iis.u-tokyo.ac.jp

[†)]Univ. of Electro-Communications
Tokyo, 182-8585, Japan
hiroshi@kimura.is.uec.ac.jp

Abstract

This paper describes our current research on how a robot, through observation of human demonstrations, can learn task level representations of human hand-work tasks. Representations proposed so far typically handle a human hand trajectory as it is or segment the entire task into a discrete pre-determined symbol sequence; but to make a generalized model of human hand-work tasks, both types of information must be incorporated into the model appropriately. We propose a technique for segmenting an observed hand-work task into pieces which are composed of fine motion or coarse motion. Fine motion means delicate manipulation and holds the relative trajectory between the grasped object and the target object, while coarse motion is a symbol which connects each fine motion. During coarse motion, a trajectory can be adjusted according to the environment or the structure of a robot when the robot performs the same task performed by a human.

To extract essential fine motion automatically, we propose a technique for integrating and aligning multiple observations of different demonstrations, which are virtually the same task, by using data gloves and multi-dimensional Dynamic Programming (DP) matching. Along each fine motion, the relative trajectory (position and orientation) is calculated by tracking the manipulated object using a stereo vision. To localize and track the manipulated object efficiently, we propose a model-based localization technique, which combines 2D and 3D template matching.

We have implemented those techniques on our humanform robot and present an experimental result which analyzed and performed a non-contact hand-work task.

1 Introduction

Our research goal is automatic acquisition of robot behavior, in particular, hand-actions, from an observation based on automatic programming approach[1, 2]. In this framework, a robot obtains knowledge of various types of human behavior mainly from observation and then efficiently constructs a reusable model of a human task.

So far, many techniques for acquiring human hand tasks have been proposed[3, 2, 5]. They are classified into two types: one is for handling the entire trajectory of the arm directly and realizing the imitated robot arm motion[3]; the other is for generating a symbolic representation of the hand behavior[2, 5]. But direct imitation of the entire trajectory is undesirable because of the difference in mechanism of the body between a human and a robot and also the difference between the situation at the demonstration time and that at the performing time. On the other hand, segmentation and labelling of robot motion into the corresponding symbols (ex. grasp) do not mean that manipulation skills have been obtained.

To acquire the manipulation skills automatically, we have been analyzing contact-state relationship between the manipulated objects[4], but this technique cannot be applied to non-contact behavior which typically appears in everyday tasks. Toward this end, we previously presented a technique for constructing a symbolic task model. This technique is composed of hand-motion primitives and object primitives in a specified task domain[5]; however, this model describes nothing about dextrous manipulation motion.

In this paper, we propose a technique for incorporating trajectory information into the symbolic task model. We segment each grasping phase into two kinds of motion: fine motion and coarse motion. Fine motion means a delicate behavior, and the relative motion between the manipulated objects must be carefully maintained. Coarse motion is defined as an intermediate state between fine motions; a trajectory during coarse motion can be reconstructed to fit the structure of a robot or the environment.

In Section 2, we present the definition of fine motion and how to incorporate it into the task model. In Section 3, we propose a technique to automatically segment a human hand-work task into fine motion and coarse motion by integrating multiple observations with multi-dimensional Dynamic Programming (DP) matching. In Section 4, we

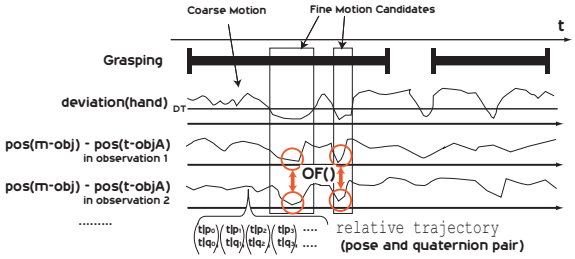

Figure 1: Fine motion and coarse motion.

propose a technique for robustly localizing and tracking the object in the scene to obtain the relative trajectory. In Section 5, we describe how we implemented these techniques on our human-form robot; we then present results of an experiment which analyzed and performed a non-contact hand-work task. We present our conclusions in Section 6.

2 Task representation

2.1 Fine motion and coarse motion

Figure.1 shows a typical flow of a hand-work sequence. We define fine motion and coarse motion as shown below.

$$\begin{aligned} \text{Fine Motion} := \quad & grasping \wedge \\ & deviation(\text{hand}) < \text{DT} \wedge \\ & OF(pos(\text{m-obj}) - pos(\exists \text{t-obj})) \\ \text{Coarse Motion} := \quad & grasping \wedge \\ & \neg(\text{Fine Motion}) \end{aligned}$$

DT is the deviation threshold of a successive hand position which is used to determine whether the hand is stationary or moving. OF() is the function which estimates the likeliness between the corresponding relative trajectories ($pos(\text{m-obj}) - pos(\exists \text{t-obj})$ among all the observations. The target of this analysis is hand-work tasks, which are sequential mutual actions between the manipulated object and a target object, and each mutual action is assumed to be the same among all the demonstrations.

All motion except fine motion during grasping is classified as coarse motion.

2.2 Relative trajectory

As described above, fine motion represents interaction between two objects and, during that time, maintaining the mutual relations is very important to the success of the task. So, during fine motion, we calculate the relative position and orientation of the manipulated object in the target object's coordinates $({}^t p_i, {}^t q_i)$ $(0 \leq i \leq T)$ and store the trajectory in the model (Figure.1). We use quarternion ${}^t q_i = [u, v, w, s]^T$ to represent the orientation.

When the robot performs fine motion in the task,: (I) the robot calculates the pose of the target object in the world coordinates $({}^w p_r, {}^w q_r)$; (II) the robot calculates the pose of the manipulated object in the target object's coordinates $({}^t p_i + {}^t q_i \cdot {}^w p_r, {}^t q_i \cdot {}^w q_r)$ $(0 \leq i \leq T)$ and moves the already grasped object to that pose.

3 Segmentation of hand-work tasks

3.1 Initial segmentation

We adopted the Polhemus 3D tracker to acquire the position of the hand during a demonstration. We capture data in about 15 fps and calculate the standard deviation of each successive three position data. By hysteresis thresholding (lower=5mm, higher=10mm; the value was determined empirically), we can obtain an initial segmented motion. For each segmented fine motion, we calculate the list of average distance between the manipulated object and the other detected objects in the scene. We also calculate the normalized vector (from the hand to each detected object) list (Figure.2). To recognize and identify the objects in the scene, we adopted the model based recognition technique presented in [5].

Distance list and vector list is defined as follows.

$$\text{distance\_list(i,j)} := (dist_0^{(i,j)}, dist_1^{(i,j)}, \cdots, dist_{num\_of\_objs}^{(i,j)})$$

$$\text{vector\_list(i,j)} := (vec_0^{(i,j)}, vec_1^{(i,j)}, \cdots, vec_{num\_of\_objs}^{(i,j)})$$

Where i is an order of a motion sequence, and j is an order in a motion sequence, $dist$ means Euclidean distance between the hand and each object. vec means normalized vector from the hand to each object.

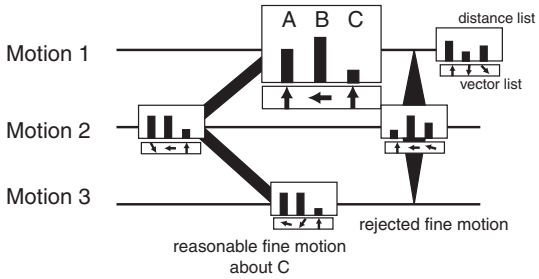

Figure 2: Matching between multiple motion sequences. A, B and C are the detected objects in the scene and the upper vertical bars mean the average distance to the manipulated object. The lower arrows mean the direction from the hand.

3.2 Multi-dimensional DP matching

A single observation will easily lead to misunderstanding of the demonstration, because the obtained relationship

between fine motion and the detected event (e.g., "hand is stationary") is groundless: this means that one cannot determine whether the obtained relationship is essential to the task or is just an accidental factor in a demonstration. This causes the generation of massive fine motion which should be regarded as noise behavior (coarse motion).

To remove those non-essential fine motions, we propose a technique for integrating multiple observations of virtually the same task by multi-dimensional DP matching.

Multiple sequence alignment based on DP matching is extensively studied in the field of biological computing[6]. For a small scale problem, natural extension of pair-wise DP matching to higher dimensions can be adopted, but its computing cost is $O(2^k N^k)$ so it is impractical to apply it to long or a large number of sequences. For a medium scale problem, minimization of search area in a multi-dimensional lattice by using constraints from possible combinations of pair-wise DP matching has an effect. For a large scale problem, progressive pair-wise alignment algorithm using a likelihood tree has a great advantage. But the latter two methods are not guaranteed to find the optimal solution.

In this study, the number of sequences (demonstrations) is relatively small (5-10 sequences) and, because alignment is processed on each grasping period, the length of each sequence is also relatively short. So we adopted the simplest method described first.

3.3 Integration of multiple observations

To apply DP matching, each hand-work sequence must be discretely segmented in advance and also we have to decide which evaluation function to use for calculating the likelihood between nodes on different sequences. In this study, the initial segmentation is done as described in the previous section and each sequence is rearranged to be composed of discrete distance lists and vector lists, i.e., fine motion.

Multi-dimensional DP matching is a natural extension of pair wise DP matching and can be calculated by iteratively computing the following recursion formula.

$$
\begin{aligned}
g(i_1, i_2, \quad & \cdots, i_N) = d(i_1, i_2, \cdots, i_N) \\
& + \min\{\cup g(ii_1, ii_2, \cdots, ii_N) | \\
& i_1 - S \leq ii_1 \leq i_1, \cdots, i_N - S \leq ii_N \leq i_N, \\
& \neg((ii_1 = i_1) \wedge \cdots \wedge (ii_N = i_N)\}
\end{aligned}
$$

Where N is the dimension (the number of the sequences): i_j is the current node in the jth sequence. $g(i_1, i_2, \cdots, i_N)$ is the accumulated likelihood from the origin to this node $(i_1, \cdots, i_N)$. $d(i_1, i_2, \cdots, i_N)$ is the evaluation function to calculate the likelihood between nodes. S is the limit factor to suppress the amount of transition, and we set S to 2.

We define an evaluation function $d(a_1, a_2, \cdots, a_N)$ as follows. This function returns the smallest variance among the detected objects. weighted_deviation in the function calculates $weight \times \Sigma(1 - \overrightarrow{vec_{avg}} \cdot \overrightarrow{vec_i})$.

function evaluationFunction: real;
var $a_1, a_2, \cdots, a_N$: integer;
$dev_1, dev_2, \cdots, dev_{number\_of\_objects}$: real;
i,k: integer;
begin
  k := 1;
  repeat
    dev_k := standard_deviation ($dist_k^{(i,a_i)}$
      $|i \in \{1, 2, \cdots, N\}$);
    dev_k := dev_k
      + weighted_deviation ($vec_k^{(i,a_i)}$
      $|i \in \{1, 2, \cdots, N\}$);
  until k > $number\_of\_objects$;
  i := argmin$_k\{dev_k|$
    $1 \leq k \leq number\_of\_objects\}$;
  evaluationFunction := dev_i;
end;

Multi-dimensional DP matching finds reasonable correspondences among all motion sequences and we can remove non-essential fine motion which resides in some of the sequences(Figure.2).

4 Object localization method

In this section, we present a robust object localization technique which combines error from edge (2D constraint) and error from depth (3D constraint), and converges the total amount of error between the estimated object pose and the real pose by iterative computing.

4.1 3D constraint

The 3D-3D pose-estimation problem is to compute the pose which aligns the 3D model points m_i, with their corresponding image points (3D depth data) r_i where $i = 1, \ldots, n$. The rigid transformation is specified by the matrix-vector pair $< \mathbf{R}, t >$ where $\mathbf{R}$ is a 3×3 rotation matrix and t is a 3D translation vector (Figure.3 (left)). In general, the image (3D) points r_i will be contaminated by noise:

$$r_i = \mathbf{R}m_i + t + \beta$$

.
where β is a random 3D variable. Assume that β follows gaussian distribution; then $< \mathbf{R}, t >$ is obtained by minimizing the following equation by the least-squares method.

$$f(\mathbf{R}, \mathbf{t}) = \mathbf{\Sigma} \parallel \mathbf{R}\mathbf{m_i} + \mathbf{t} - \mathbf{r_i} \parallel^{\mathbf{2}}$$

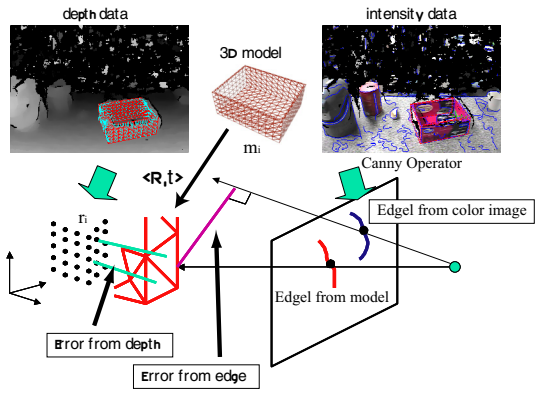

Figure 3: Object localization method which makes use of error from edge and error from depth simultaneously. Error is calculated from sum of weighted distance between each model point and input data (edgel or depth point).Blue line in the intensity image indicates extracted edgels.

However, in real applications, errors in the data are not normally distributed and, in that case, least-squares estimation is inappropriate.

Wheeler et al. proposed 3D template matching technique as a robust pose estimator based on M-estimation[7], which is used in the field of robust statistics, and which can eliminate noise belonging to outliers. M-estimator is a generalization of least-squares and is defined in the form of $E(z) = \Sigma_i \rho(z_i)$, where $\rho(z_i)$ is an arbitrary function of the errors, z_i is the observation. Kawamura el al. tried several error distribution functions as $\rho(z_i)$ in the task of localizing real electric facilities on an electric pole and found that, when the initial pose error is within 50mm, Lorentzian function is the best function among them[8].

In our experiment, the system could generally estimate the initial position error within 50mm, so we adopted the 3D template matching (3DTM) technique with Lorentzian function as an error distribution function.

$$\psi(z) = \frac{z}{1 + \frac{1}{2}z^2} \cdots \text{Lorentzian weight function}$$

For each piece of captured depth data, 3DTM first constructs a KD-tree from the dense depth points(x, y, z). Then it searches in the KD-tree for the closest point to each model triangle and estimates the average of the weighted distance between the model pose and the data. To minimize the matching cost, we use the gradient-descent search.

4.2 2D constraint

2D template matching (2DTM)[7] is much the same technique as described above, but it estimates the 3D points from the extracted edges on an intensity image.

2DTM first extracts edgels from an intensity image by use of Canny Operator and then constructs a KD-tree from the image-points and their differentials (4 dimensions) for each edgel. 2DTM then searches for the closest point in the KD-tree to each edgel of the model projected on the image. Edgels of the model are determined in two ways: some are determined off-line as explicit corners formed by two adjacent meshes, while the others are determined online from the edges whose normal is perpendicular to the viewing direction. As a result, 2DTM determines the correspondence between the 2D image point in the input image and the 3D model point. 2DTM estimates the 3D position of each 2D edgel point as shown in Figure.3 (right).

The correspondence between the 3D model points and the 3D image points is calculated in just the same way as 3DTM.

4.3 2DTM & 3DTM combined localization

2DTM is sensitive to the edges that appear in the image background and does not offer a good guess about z position (parallel to the viewing direction) of the model because of the approximation of z position of the 3D edgels. But, at the final stage of the localization, 2DTM offers a good guess about the position and orientation perpendicular to the viewing direction.

So, we first adopt the 3DTM only to localize the object to the approximate position; then we adopt the 2DTM & 3DTM combined method to localize the object to the exact position as shown in Table1.

Table 1: 2DTM & 3DTM combined localization

	Method	Sigma[mm]
1	3DTM	20.0
2	3DTM	5.0
3	3DTM	2.0
4	3DTM & 2DTM	2.0
5	2DTM	1.0

2DTM and 3DTM are calculated in the same 3D space by M-estimator (Lorentzian) with different weights. Sigma is the parameter for determining the width of the distribution function to reduce the effect of outliers.

We utilize a 9-eye stereo vision system to produce the intensity and disparity images. This vision system also provides the least and the second least Sum of Sum of Absolute Difference (SSAD) value for each pixel. We can regard pixels in a disparity image as noise or as being out of the measurement range when the difference between the above two SSADs is too small. By excluding those pixels, we can reduce the computation time for building and searching the KD-tree and miss-matching.

5 Experiment

5.1 Platform

We developed a human-form robot as a test-bed to verify our algorithms, to test the validity of a constructed task model and to realize human-robot cooperative behavior.

This robot has similar capabilities to those of the human upper torso and is equipped with a real-time 9-eye stereo vision system, 7 D.O.F. dual robot arms and delicate hands which have 4 fingers and 3 fingers, each equipped with force/torque sensor on its finger tips. This stereo vision system was utilized to observe human demonstration.

5.2 Integration of multiple observations

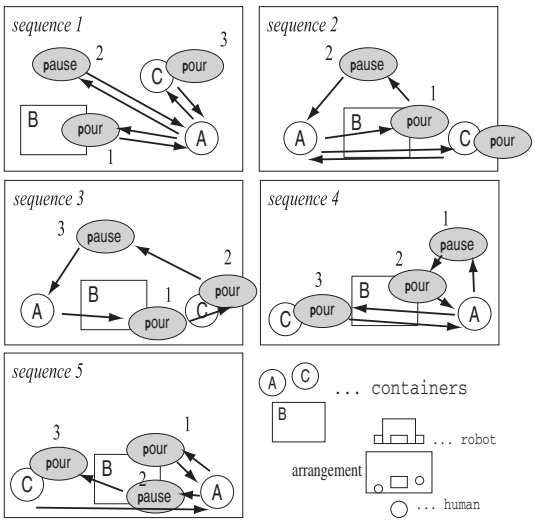

Figure 4: Configuration of multiple motion sequences.

In this experiment, a human demonstrated the same task five times with minor deviations, i.e., the arrangement of the objects and the hand motion path in each demonstration was different. The nature of this task was to pour the contents of container A into container B and then pour the contents of the container A into container C as shown in Figure.4; this was achieved by maintaining a specific relative motion between the two objects when pouring.

Figure.5 shows an example of the segmented hand trajectory. From the initial segmentation, we obtained the data set as shown in Figure.6. Clearly we see that non-essential fine motion was detected and, to disregard that useless motion, we have to relate the sequences to one another. We applied the technique introduced in Section 3 and, as a result of DP matching, the system correctly made a correspondence between all of the five sequences as a thick line box shown in Figure.6 and removed non-essential motion.

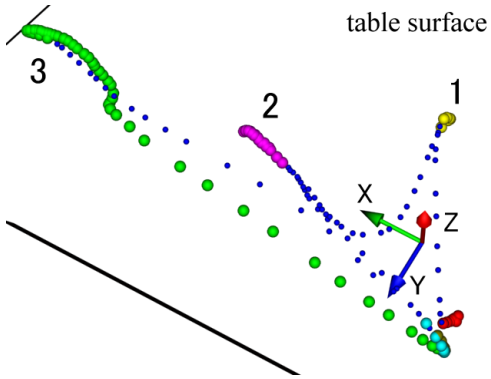

Figure 5: Trajectory of the motion sequence 4. The big spheres indicate stationary state and the small spheres indicate moving state. The numbers in the figure correspond to the numbers in the next figure.

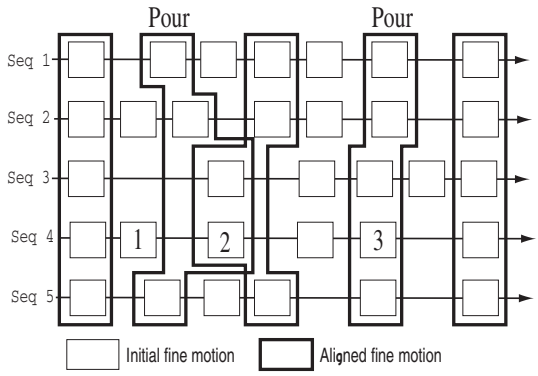

Figure 6: Result of integration by multi-dimensional DP matching.

5.3 Tracking of fine motion

Tracking was performed off-line. All the disparity and intensity images from the stereo vision were recorded in about 15fps during the demonstration. After the fine motion was integrated, the system fetched the recorded images corresponding to the fine motion period and tracked the manipulated object in the successive scenes by using the technique presented in the previous section. The initial position is determined by the data from polhemus sensor attached on the data glove. Figure.7 shows the tracking result and we see that the object is correctly localized.

At the time of tracking, the position and the orientation of the manipulated object in the target object's coordinates are stored in the task model for each image.

5.4 Performance by the robot

After the task model was successfully constructed, the robot performed the same task (in this case, pouring mo-

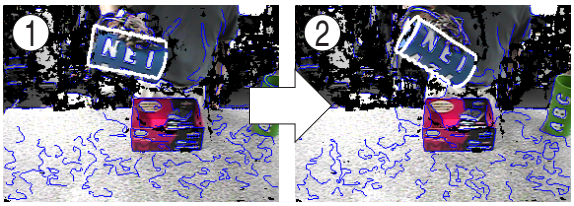

Figure 7: Tracking of container A in “Pour” motion. Thick white wire-frame indicates the projected geometric model. Winding line fragments all over the image are the detected edges. Clutter in the background is caused by the limitation of measurement range and is almost removed by SSAD thresholding.

Figure 8: Human demonstration and robot performance.The pose of the robot arm in the right figure is calculated from the relative pose of the object in the left figure.

tion). The arrangement of the objects differs from the situation at the demonstration, but the robot successfully localized container B in the scene and estimated its position and the orientation. Then the robot recomputed the trajectory of the manipulated object (Container A) to maintain the relative relationship as written in the task model during fine motion.

In this experiment, the reference relative trajectory was selected from sequence 5, but the averaged trajectory derived from all the sequences will be much more reliable.

6 Conclusions

In this paper, we have proposed a technique to automatically segment non-contact hand-work tasks into fine motion and coarse motion by integrating multiple observations using multi-dimensional DP matching. We have also presented a robust object localization technique, which is necessary to extract trajectory during fine motion. This trajectory information can make up for the previously proposed symbolic task model.

Multi-dimensional DP matching is applied to align five demonstration sequences, but this method is computationally expensive when the number of sequences increases. If the dimension or the length of the sequences are to be larger, we can apply other efficient algorithms introduced in Section 3.

We also proposed 2D & 3D combined template matching technique which can eliminate the effects of outliers to enable us to register the pose of the manipulated objects in the images obtained from vision. Matching calculation is processed so as to first utilize 3D information only to estimate the approximate pose and then to fine tune the pose using 2D edge correspondences.

In this study, we focused only on fine motion. As for coarse motion, some optimization of the trajectory fitted to the robot body structure rather than to human body structure is necessary for a robot to be skillful in a certain task. The contemplated future work is to solve the problem of detecting and recovering from errors when the robot fails to perform a task.

Acknowledgment

This work is supported, in part, by Japan Society for the Promotion of Science (JSPS) under the grant RFTF 96P00501, and, in part, by Japan Science and Technology Corporation (JST) under Ikeuchi CREST project.

References

[1] K. Ikeuchi and T. Suehiro: “Toward an Assembly Plan from Observation Part I: Task Recognition With Polyhedral Objects,” *IEEE Trans. on Robotics and Automation*, 10(3):368–384, 1994.

[2] Y. Kuniyoshi, M. Inaba and H. Inoue: “Learning by watching,” *IEEE Trans. on Robotics and Automation*, 10(6):799–822, 1994.

[3] H. Asada and Y. Asari, “The direct teaching of tool manipulation skills via the impedance identification of human motions,” *Inter. Conference on Robotics and Automation*, 1269-1274, 1988.

[4] J. Takamatsu, H. Tominaga, K. Ogawara, H. Kimura and K. Ikeuchi: “Extracting Manipulation Skills from Observation,” *IEEE/RSJ IROS*, 1:584-589, 2000.

[5] K. Ogawara, S. Iba, T. Tanuki, H. Kimura and K. Ikeuchi: “Recognition of Human Task by Attention Point Analysis,” *IEEE/RSJ IROS*, 3:2121-2126, 2000.

[6] G. Fuellen: “A Gentle Guide to Multiple Alignment,” *http://www.techfak.uni-bielefeld.de/bcd/Curric/MulAli/mulali.html*, 1997.

[7] M. D. Wheeler: “Automatic Modeling and Localization for Object Recognition,” *Ph.D Thesis*, CMU, 1996.

[8] K. Kawamura, K. Hasegawa, Y. Someya, Y. Sato and K. Ikeuchi: “Robust Localization for 3D Object Recognition Using Local EGI and 3D Template Matching with M-Estimators,” *IEEE ICRA*, 2:1848–1855, 2000.