*Full paper*

# Toward Human-Like Real-Time Manipulation: From Perception to Motion Planning

**Sukhan Lee**[a], **Hadi Moradi**[b], **Daesik Jang**[c], **Hanyoung Jang**[d], **Eunyoung Kim**[e], **Phuoc Minh Le**[a], **JeongHyun Seo**[a] **and JungHyun Han**[d,*]

[a] School of Information and Communications, Sungkyunkwan University,
Suwon 440-746, South Korea

[b] Electrical and Computer Engineering Department, Tehran University, Tehran, Iran

[c] Department of Computer Information Engineering, Kunsan National University,
Kunsan 573-701, South Korea

[d] College of Information and Communications, Korea University,
1-5 Anam Dong, Seoul 136-701, South Korea

[e] Department of Computer Science, University of Southern California, Los Angeles, CA 90089, USA

Received 7 January 2008; accepted 18 January 2008

**Abstract**

Human-like behavior is crucial for intelligent service robots that are to perform versatile tasks in day to day life. In this paper, an integrated approach to human-like manipulation is presented, which addresses real-time three-dimensional (3-D) workspace modeling and accessibility analysis for motion planning. The 3-D workspace modeling uses three main principles: identification of global geometric features, substitution of recognized known objects by corresponding solid models in the database and multi-resolution representation of unknown obstacles as required by the task at hand. Accessibility analysis is done through visibility tests. It complements and accelerates general motion planning. The experimental results demonstrate that the human-like behavior-oriented methods are sufficiently fast and robust to model 3-D workspace, and to plan and execute tasks for robotic manipulative applications.

**Keywords**

Service robot, object manipulation, workspace modeling, multi-resolution modeling, motion planning

## 1. Introduction

Human-like real-time, as well as dependable manipulation is crucial for the next generation of service robots that are to perform versatile tasks for assisting humans

[*] To whom correspondence should be addressed. E-mail: jhan@korea.ac.kr

DOI:10.1163/156855308X315136

in day-to-day life. Such service robots need to perform environment modeling, object recognition, motion planning and task execution in an integrated system in terms of the overall performance of a given manipulation task. Therefore, more serious efforts should be expended for the in-depth study of an integrated and task-oriented paradigm of modeling–recognition–planning for manipulation. To date, modeling, recognition and planning have been developed more or less independently of each other without accounting for the effect of their interaction on the overall performance as an integrated system, which often fares poorly when it comes to integration.

Prior research that is related to our approach includes grasp planning of relatively flat objects that can be treated as an extrusion of their projected silhouettes based on vision-based two-dimensional (2-D) object modeling [1], and the planning of grasp pose and stable grasp for unknown objects under the constraint of manipulator workspace based on vision-based three-dimensional (3-D) object modeling *via* structure from motion [2]. The latter combines grasp planning with object modeling as an integrated system. For human-like manipulation in a natural setting, however, further generalization is required since the work focuses the modeling only on a single object to grasp without consideration of the surroundings for collision avoidance or carry out the modeling without taking overall real-time performance into consideration.

Generally speaking, proper attention is yet to be paid either to the critical issues of modeling associated with the execution of manipulation tasks or to the opportunity of exploring various robotic functions as additional inputs to modeling. For instance, since supporting the real-time requirement of a task is more immediate and crucial than promoting higher precision in modeling for robotic manipulative tasks, the level of precision in modeling may be set relative to task accomplishment in compromise with efficiency. Furthermore, if helpful, such robotic functions as motion planning and object recognition can be integrated with 3-D modeling as a means of furnishing additional inputs to modeling.

In this paper, we address the areas where we can improve the performance of service robots by considering the robotic tasks, toward human-like behaviors, i.e., real-time 3-D environment modeling and motion planning accelerated by visibility analysis.

## 2. Related Work

Most of 3-D modeling research to date has focused on either accurately representing individual objects/artifacts for archiving/displays or representing an environment as a map for navigation or walk-through [3, 4]. In contrast, although real-time modeling of 3-D cluttered workspace has been of utmost interest to the robotics community, due to its impact on automated manipulation based on a perception referenced action paradigm, few benchmarking research results have been produced and are available today.

One of the issues crucial for establishing an approach to 3-D modeling may be the optimization of the trade-off between precision and efficiency. Several works [5–7] addressed this trade-off of precision and efficiency mainly in the context of navigation, which may differ from that in the context of manipulation. Section 3 is the core of this paper, and presents an approach to 3-D modeling as part of an integrated system of modeling, recognition and manipulation. A preliminary version of Section 3 has appeared in Ref. [8].

The construction of a 3-D model often requires the registration of multiple scans or images. The most popular algorithm for registration is Iterative Closest Points (ICP), originally proposed by Besl and McKay [9]. The major problem associated with ICP may be the convergence to local minima in the case where the initial relative pose is far apart or in the case where there are many outliers. Recently, spin-images [9] and SIFT (scale-invariant feature transform) [5, 10] have been chosen as features for use in multiple-scan registration.

In general, accessibility analysis performs spatial or geometric reasoning to determine the directions along which a tool can access a target object. The majority of the work in accessibility analysis has been conducted in the inspection field [11, 12], in which it is assumed that the environment is open for a probe's motion and only the work piece may collide with the probe. Moreover, all previous works on determining accessibility propose algorithms that run mostly off-line.

For the purpose of real-time accessibility analysis, the authors proposed to utilize the visibility query accelerated by commodity graphics hardware [13, 14]. As discussed in Section 4, the proposed approach had both strength and weakness. Section 4 of this paper aims at reporting the progress of the authors' on-going research to overcome the weakness of the visibility query-based accessibility analysis.

## 3. Workspace Modeling

This paper presents an approach to 3-D modeling as part of an integrated system for perception-guided, human-like robotic manipulation, particularly for performing services for humans. Therefore, unlike conventional approaches where no proper attention has been given to the overall performance of an integrated system, this study addresses the efficiency and performance trade-off by considering the integration of robotic capability of recognizing global geometric features and objects, and motion planning.

The proposed integrated approach to 3-D modeling explores and mimics the computational principles behind the way humans model surroundings for navigation or manipulation. In a fundamental sense, it is the perception–action paradigm that human navigation or manipulation is based on. A closer examination of the perception–action paradigm of humans reveals that humans achieve an optimal trade-off between efficiency and performance, by exploiting multiple sources of information available for efficiency in modeling and reducing uncertainties, and by controlling the level of modeling precision as required by the task at hand.

For instance, upon encountering a new environment, humans quickly approximate the geometric configuration of the environment based on known global geometric features such as parallelepiped rooms, long corridors, floors, ceilings, walls, etc. Humans quickly identify known objects in the environment and replace them by the models in memory so they do not need to be concerned about the details. Humans pay attention to modeling details only when it is necessary for accomplishing the given task at hand or avoiding collision. The above observations on the perception–action paradigm of humans coincide well with the fundamental nature of human visual perception in connection to geons [15].

Based on the above observations, we have chosen the following three fundamental computational principles in 3-D modeling:

(i) The approximate yet rapid characterization of geometric configuration of the workspace by recognizing global geometric features such as planes, walls and floors, and their spatial relations.

(ii) The quick recognition of known objects in the workspace for the *in situ* substitution of the recognized objects by their corresponding models in the database.

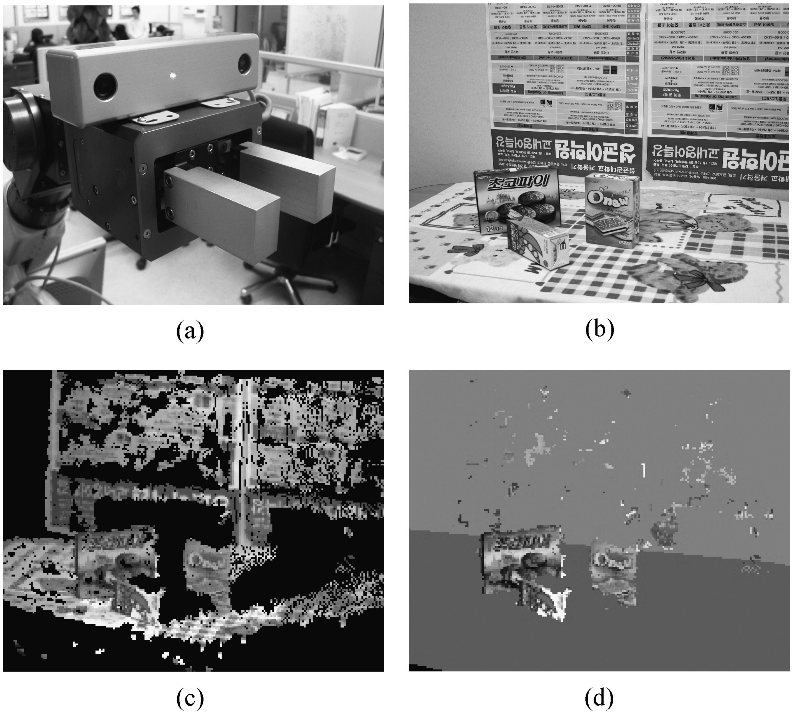
(iii) The task-oriented modeling of geometric details on the fly, where the resolution of geometric details is chosen adaptively due to the requirement of accomplishing the task at hand.

### *3.1. Planar Feature Extraction*

In our experiments, the service robot is equipped with a stereo camera, mounted on the parallel jaw gripper with an eye-on-hand configuration, as shown in Fig. 1a. Figure 1b shows the workspace and Fig. 1c shows the range data acquired from the stereo camera on the fly. The range data are in the form of 3-D point clouds.

For an approximate yet rapid characterization of geometric configuration of the workspace, the global planar features of the workspace are extracted using an extended version of the RANSAC (Random Sampling Consensus) algorithm [16] — a widely used technique for fitting a model to an experimental data set. To ensure fast and robust extraction of planar features, we resort to the combined use of the 3-D point clouds from stereo-sis and the SIFT features, the 3-D information of which is available from stereo-sis. The boundaries of a feature plane are determined using a 2-D oriented bounding box from the covariance matrix computed with 3-D points that belong to the corresponding feature plane.

Planes are extracted and replaced for the point clouds. The extracted planes, together with the remaining points that are not lying on the planes, are shown in Fig. 1d. By replacing the 3-D points that belong to the identified planar features simply by plane models, we can achieve a significant enhancement of computational efficiency through the reduction of the search space for subsequent processes such as object recognition.



(a)

(b)

(c)

(d)

**Figure 1.** Point clouds generation and plane extraction. (a) Gripper with a stereo camera. (b) Workspace. (c) Point clouds for the range data. (d) Extracted planes.

### *3.2. Object Recognition and Registration*

It is assumed that all objects to be manipulated (called target objects) have solid model representations in the object database. The object recognition algorithm is comprised of the following steps:

(i) Matching SIFT features between the measured and the modeled.

(ii) Verification of the matching result based on additional geometric constraints among SIFT features.

(iii) Integration of the solid model of the recognized object into the workspace model.

Unlike the traditional algorithms [17], we use SIFT features with their 3-D information available from stereo-sis (stereo-sis SIFT) for object recognition. The SIFT-matching procedure will output a set of matched stereo-sis SIFT-point pairs. The pairs are then verified using a simple 3-D geometric constraint, named 3-D triangle constraint. When there are $N$ pairs, ${}_{N}\mathrm{C}_3$ triangles can be generated. The 3-D triangle constraint requests that a pair of triangles based on three matched point pairs should be congruent within an error bound. Only when the triangles satisfy the constraint are their vertices (three pairs of the matched points) accepted. Each measured SIFT point listed in a set of matched stereo-sis SIFT-point pairs is subject to a triangle constraint test and is removed out from the pool of SIFT points waiting for a test, once found valid. This process continues till all the measured SIFT points are exhausted.

Figure 2 compares the matching results with and without the constraint. In Fig. 2, the solid model of a cereal box, drawn from the database, is virtually placed at the right side. In Fig. 2a, a mis-matched SIFT-point pair is found. The complexity of object recognition is $O(M_\mathrm{S}N_\mathrm{S})$, where $M_\mathrm{s}$ is the number of SIFT features in an object model and $N_\mathrm{s}$ is the number of SIFT features in the search space of a workspace.

If the total number of matched SIFT features after verification is greater than or equal to 3, the object is substituted by the corresponding solid model from the database using the quaternion-based method [18]. Figure 3 shows the result.

The search space for object recognition can be significantly reduced by using the planar features extracted at the previous stage. For example, consider the assumption that all objects lie on the extracted planar surfaces, such as table/desk top

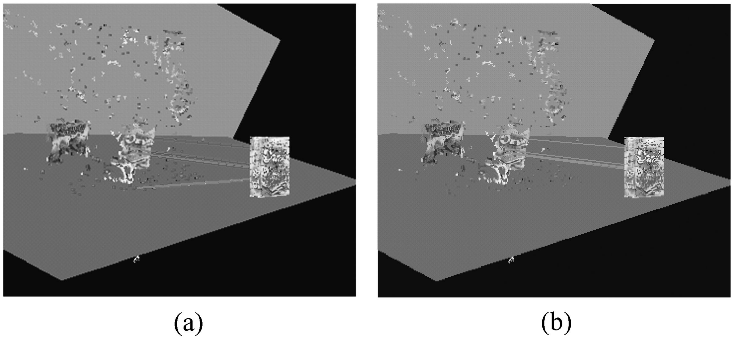


**Figure 2.** SIFT-based object recognition. (a) SIFT matching without the 3-D triangle constraint. (b) SIFT matching with the 3-D triangle constraint.

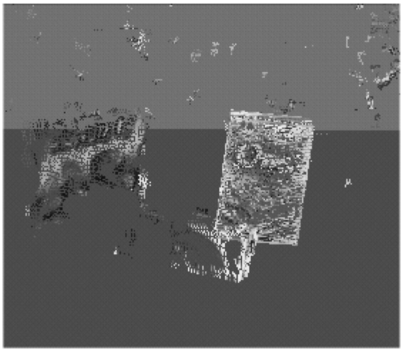


**Figure 3.** Substitution of the target object's point clouds by the corresponding solid model of the database.

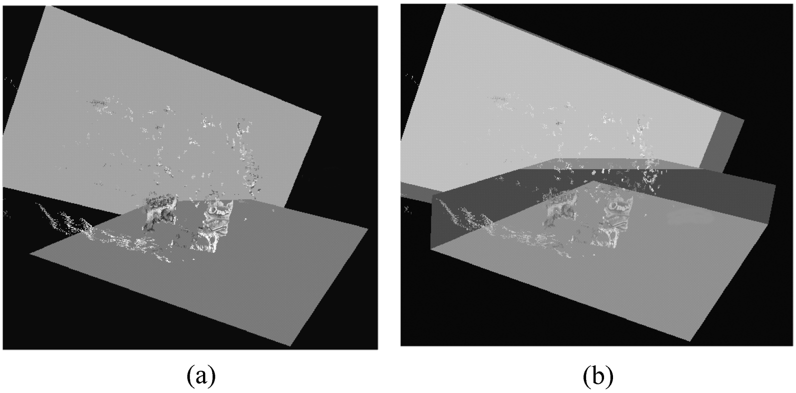


**Figure 4.** Reduced search space of a target object. (a) Extracted planes. (b) Search space.

planes, i.e., no object is placed on top of other objects. In order to utilize the assumption, the distance between a SIFT feature and an extracted plane is calculated. If the distance is larger than the target object's height for all extracted planes, the SIFT feature is discarded. The remaining SIFT features constitute the search space for object recognition. Figure 4 depicts the reduced search space of a target object. The object recognition process immediately benefits from the reduced space. In the current implementation, the recognition process in the reduced space takes on average 117 ms, while working with the original space takes 234 ms.

Note that consideration of the planar surfaces simply reduces the search space. The proposed stereo-sis SIFT-based object recognition is not limited to an object lying on planar surfaces, but can be applicable to any object as long as a geometric model of the object is available together with the SIFT features on it.

### *3.3. Task-Oriented Obstacle Modeling Using a Multi-Resolution Octree*

The remaining point clouds that belong neither to the extracted global plane features nor to the recognized objects are taken as obstacles. The number of obstacle points is usually too large and therefore approximate representation is needed. A candidate for such an approximate representation is a regular grid, where every grid cell is of a uniform size. Figure 5a shows the regular grid cells, each of which contains at least a 3-D point.

More preferred to the regular representation is a task-oriented representation, where obstacles are modeled with different levels of precision, as required by the task at hand. Note that the final task of the integrated system is to make a collision-free path for grasping the target object. In this case, a task-oriented representation implies that the resolution of individual cells be determined as required by generating a collision-free path. That is, those obstacles required to take into consideration for generating a collision-free path should be represented in a higher resolution of cells sufficient enough for an optimal collision-free path. On the other hand, those

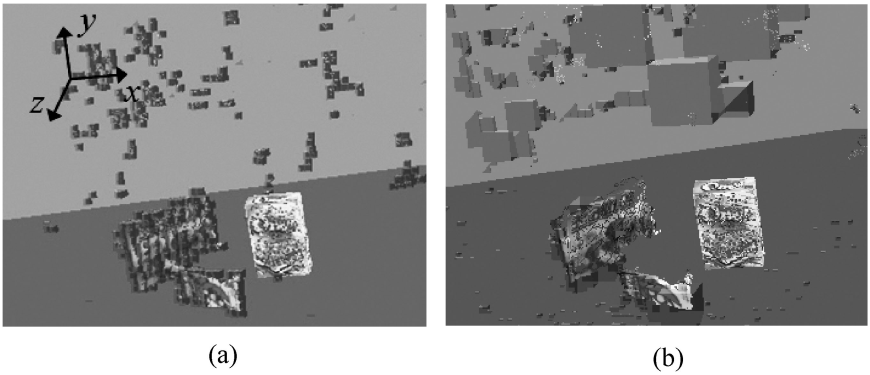


**Figure 5.** Obstacle representations. (a) Regular grid. (b) Octree.

obstacles having nothing to do with generating a collision-free path may be represented in a coarse resolution of cells.

To implement the above task-oriented representation, we adopt a multi-resolution octree. It is formed by dividing a parent cell recursively into eight equal-sized child cells, starting from the entire workspace as the initial cell, where the number of recursion in cell division determines the resolution of the corresponding cell. We propose the recursive division to be carried out based on the fill factor of an octree cell. The fill factor of a cell is defined as the following indices representing how well the captured 3-D point clouds cover the corresponding cell: (i) the weighted density of 3-D points of a cell and (ii) the uniformity of 3-D point distribution in the cell.

The density of 3-D points is determined simply by dividing the number of 3-D points in the cell by the cell size defined here as the area of the cell face. However, in order to obtain a more accurate description of density, we need to take the distance of a 3-D point from the camera into consideration. This is because, due to the nature of stereo-sis, the further a cell is located from the camera, the less the number of 3-D points generated for the cell, as shown in Fig. 6. That is, the density of 3-D points becomes lower as they are located further from the camera. To take this into consideration, we assign a weight to a 3-D point and define the weighted density as the sum of the weights of the 3-D points in a cell, where the weight of a 3-D point is determined to be proportional to the multiplication of the depth of the 3-D point (i.e., the distance along the optical axis from the camera) and the pixel size divided by the focal length of the camera. The advantage of using the weighted density is that the octree representation is less sensitive to the distance of the scene from the camera.

The uniformity of 3-D point distribution in a cell is to see if the cell has an empty or free space despite the weighted density of the cell exceeding the threshold. The uniformity can be checked based on the biases of the mean and variance of the 3-D points in the cell from the desired values.

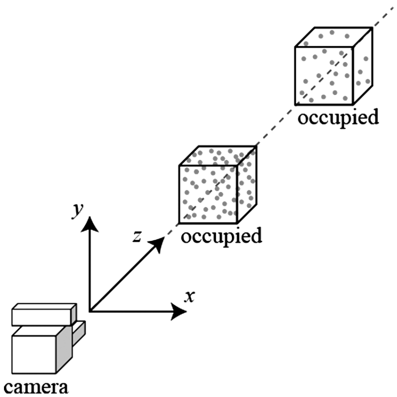


**Figure 6.** Distance-based fill factor computation.

When a cell has a weighted density greater than the predefined threshold and satisfies the uniformity criterion, the cell is determined to be 'occupied' and the recursion stops. Otherwise, recursive subdivision continues. Note that the use of the weighted density allows a cell far from the camera to be classified as 'occupied' if the 3-D points are uniformly distributed, even though the number of 3-D points in the cell is smaller, as shown in Fig. 6. The multi-resolution octree is generated by assigning the thresholds of the weighted density and the uniformity differently to individual cells, depending on their levels of precision as required by such task as the generation of a collision-free path.

In the case where a cell contains no 3-D point, it is declared as 'empty'. In addition to 'occupied' and 'empty', the leaf nodes of the octree can be 'don't know' if the corresponding cell is out of the camera's field of view. As will be discussed in Section 3.4, the workspace may not be captured fully by one shot and then multiple scenes have to be combined. In such a case, we may designate 'don't know' to those cells out of the camera's field of view.

### *3.4. SIFT and Planar Feature-Based Registration of Multiple Scenes*

The proposed technique for 3-D workspace modeling makes use of a sequence of images for fusion. Therefore, an efficient and real-time method for registering multi-view 3-D data scanned by a stereo camera is of prime necessity. Having two feature sets, $S_{t-1}$ and $S_t$, corresponding to timeframes $t-1$ and $t$, respectively, the following steps are taken to merge the two scenes:

(i) The SIFT feature set $S_{t-1}$ is reduced to $S'_{t-1}$, such that $S'_{t-1}$ contains only the SIFT features lying on, within some error bound, any of the extracted feature planes.

(ii) $S'_{t-1}$ is then matched with $S_t$, to obtain the matched point-pair set $M_{t-1,t}$.

(iii) The transformation matrix needed for registration is calculated using the matched point-pair set $M_{t-1,t}$ and the quaternion-based method of Horn [9].

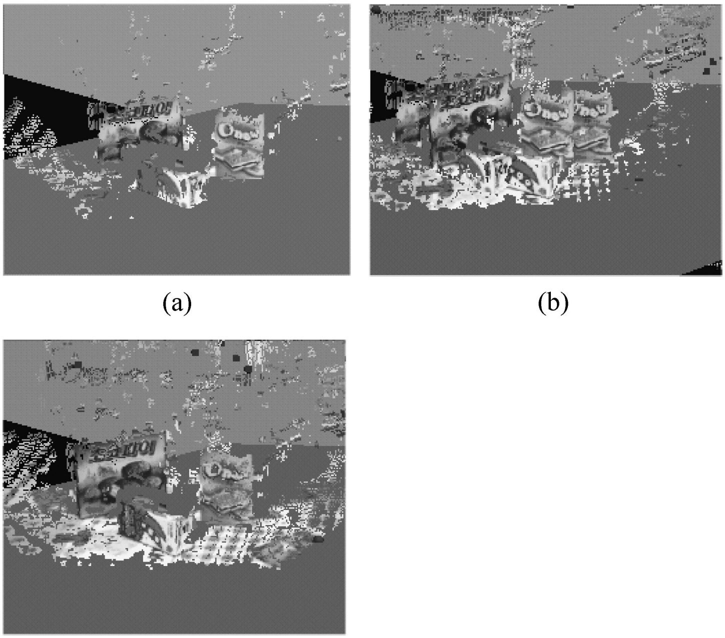


**Figure 7.** Matching multiple scenes. (a) Scene at a time frame. (b) Two consecutive scenes overlapped before registration. (c) Registered scene with the matched points overlapped.

(iv) All 3-D points at timeframe $t$ are transformed to the coordinate system of the timeframe $t-1$.

The complexity of scene registration is $O(N'_{t-1}N_t)$, where $N'_{t-1}$ and $N_t$ are the numbers of SIFT features in $S'_{t-1}$ and $S_t$, respectively. Figure 7 shows the result of scene registration based on stereo-sis SIFT and global plane features.

The cluttered neighborhood of a target object is delimited by a sphere, named the locality sphere [13, 23], the center of which is located at the target object (see Fig. 8). The radius of the locality sphere is automatically and heuristically computed, considering the population of the obstacles surrounding the target object. In the experimental workspace shown in Fig. 7, the radius is approximately 40 cm.

A simple camera motion is planned for taking a sequence of images that can entirely cover the locality sphere. However, it would be more desirable if the camera motion can be determined based on the sensor planning for capturing additional images at the best camera poses. It is subject to further research.

### *3.5. Experimental Results and Discussion*

We implemented the proposed algorithms on a mobile 7-d.o.f. manipulator, which was equipped with a parallel jaw gripper. A stereo camera, Bumblebee [18] (Point Grey Research), was installed in the eye-on-hand configuration on the gripper. The manipulator is also equipped with a PC (Pentium4 2.8 GHz CPU).

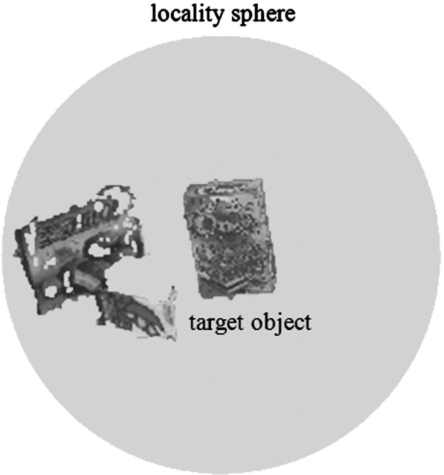


**Figure 8.** Delimited neighborhood about the target object.

**Table 1.**
Computation time for individual processes that constitute an entire cycle of modeling

| Sub-modules | Average time (ms) |
|---|---|
| SIFT feature calculation | 188 |
| Planar feature extraction | 19 |
| Object recognition | 117 |
| Multi-resolution octree construction | 60 |
| Scene registration | 180 |
| Total | 564 |

The proposed algorithms for workspace modeling have been tested for 20 workspaces, each of which is similar in complexity to the workspace of Fig. 1b. Table 1 shows the average time required for 3-D modeling in the experimental workspaces. Figure 9 shows the comparative evaluation of performance, (a) computation time and (b) depth error, among the proposed SIFT and plane-based scene registration approach (marked by 'SIFT+Plane' in the graph) the SIFT-based approach without using the planar features (marked by 'SIFT' in the graph), and the well-known ICP approach. As can be seen in the graphs, 'SIFT+Plane' and 'SIFT' show much better performance than 'ICP' in terms of the computation time (200 *versus* 800 ms) as well as of the depth error (1.5 *versus* 2.5 cm).

Although Fig. 9a does not show it clearly, 'SIFT+Plane' may have a potential to further reduce the computation time of the plain 'SIFT', especially when a larger portion of SIFT points can be pruned out by the over-the-plane constraint on SIFT features.

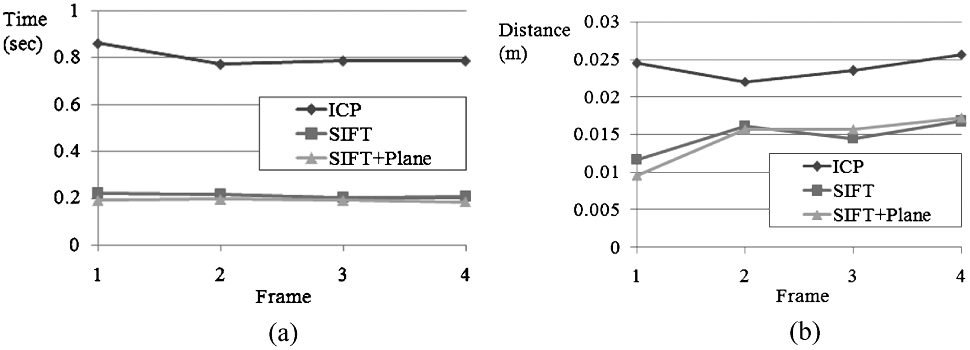


**Figure 9.** Comparative performance evaluation of the proposed approach against conventional ICP in scene registration. (a) Registration time. (b) Depth error after registration.

According to our experiments, the lateral error is much less than the depth error, often less than 30% of the depth error. In Fig. 9b, therefore, only the depth error for the estimated pose of the target object is shown. The resulting depth and lateral errors in the target pose estimate are regarded as sufficient for grasping under the tolerance of the gripper in grasping in our experimentation. Should a higher precision be required by the gripper tolerance, we may resort to fusing the multiple SIFT evidence in time from a sequence of images at different camera positions.

A stereo-sis matching often results in spurious data due to false matching or large depth errors due to off-range imaging. For workspace modeling presented in this section, we applied a procedure of pruning spurious or erroneous data based on an epipolar geometric pruning [19] and a model constraint pruning for stereo-sis SIFT as seen from the object recognition section.

## 4. Accessibility Analysis and Motion Planning

The complex nature of motion planning for high d.o.f. robots often makes it impossible to reach interactive or real-time performance [20, 21]. However, the real-time issue becomes critical for a service robot working in cluttered environments. This section provides object access and removal algorithms that can accelerate general motion planning.

As discussed earlier, the cluttered neighborhood of a target object is delimited by a the locality sphere. The motion of the gripper is restricted to be linear within the sphere, as shown in Fig. 10. More specifically, when the gripper accesses, grasps and removes the target object, the access path is linear and the removal path is the inverse of the access path. Therefore, both the access direction and the removal direction are simultaneously determined. See Refs [22, 23] for another approach where the removal path is not limited to be the inverse of the access path.

The key point in our approach is that by restricting the gripper's motion to be linear within the locality sphere, the accessibility analysis is reformulated into a visibility test. Suppose that a person sees an object and the object is fully visible along

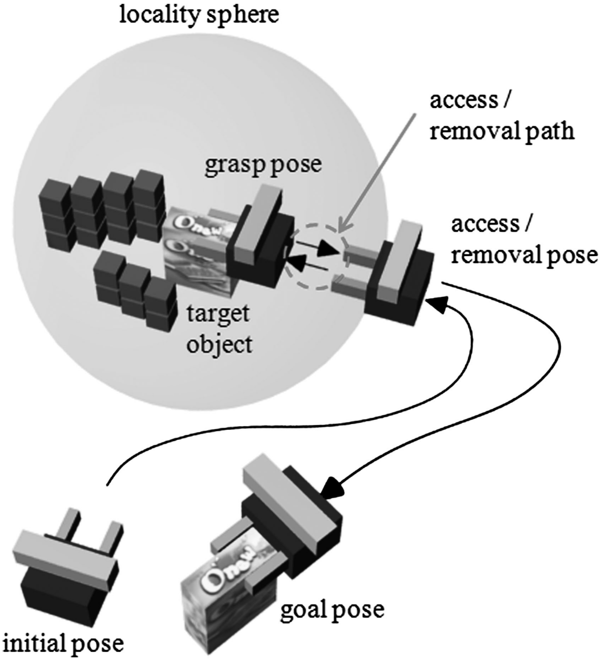


**Figure 10.** Linear paths within the locality sphere.

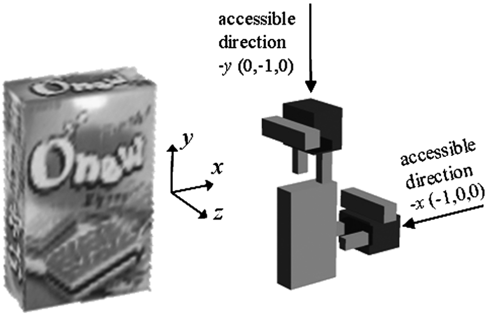

accessible direction -y (0,-1,0)
y
x
z
accessible direction -x (-1,0,0)

**Figure 11.** Accessible directions.

the viewing direction. Then, the object can be removed along the opposite of the viewing direction. Similarly, for the gripper to be able to move toward an object, grasp it and move backward to remove the object, the gripper should be fully visible at a grasp point.

### *4.1. Visibility Test*

Each target object in the database contains graspability or accessibility information. Figure 11 shows the graspability representation for the example cereal box. A robot arm of a small gripper may access the box along four directions: ±$x$ and ±$y$ with respect to the local coordinates of the box. The proposed algorithm tests the object visibility in each accessible direction.

The visibility test is done by rendering the input model and counting the number of visible pixels. Let us consider the workspace given in Fig. 5b and discuss the visibility test with the accessible direction $x$. Suppose that when only the target object is rendered with the viewing direction $x$, as shown in Fig. 12a, the number of pixels occupied by the object is $m$. At the next stage, the target object is rendered into the environment which includes octree and planes, as shown in Fig. 12b. The number of visible pixels $n$ of the object should be less than $m$, because the cereal box is just partially invisible due to the obstacles. When $m > n$, the target object is partially (when $n \neq 0$) or completely (when $n = 0$) invisible. Then, the object is determined to be *not* accessible along the viewing direction.

Figure 12c and 12d shows the same visibility test process with the accessible direction $-x$. As illustrated in the figures, $m = n$, and the box is determined to be fully visible, i.e., accessible along $-x$.

When the object is determined to be fully visible, the gripper visibility is tested. For this purpose, the object faces orthogonal to the accessible direction are sampled. By placing the gripper at a sampled point, the number of visible pixels $p$ is counted, as shown in Fig. 12e. Then, the gripper is rendered into the workspace, as shown

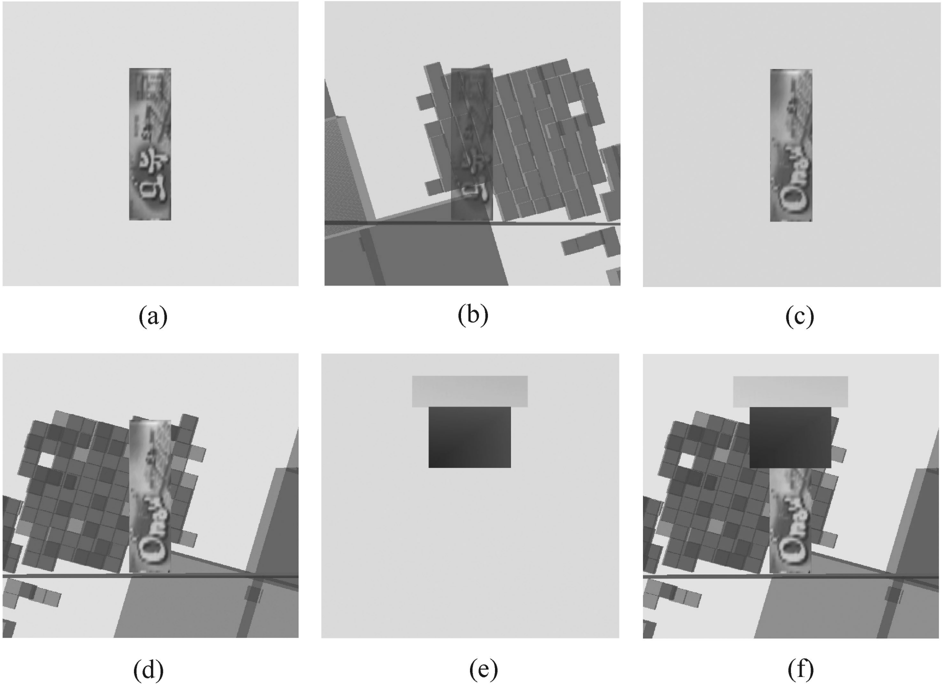


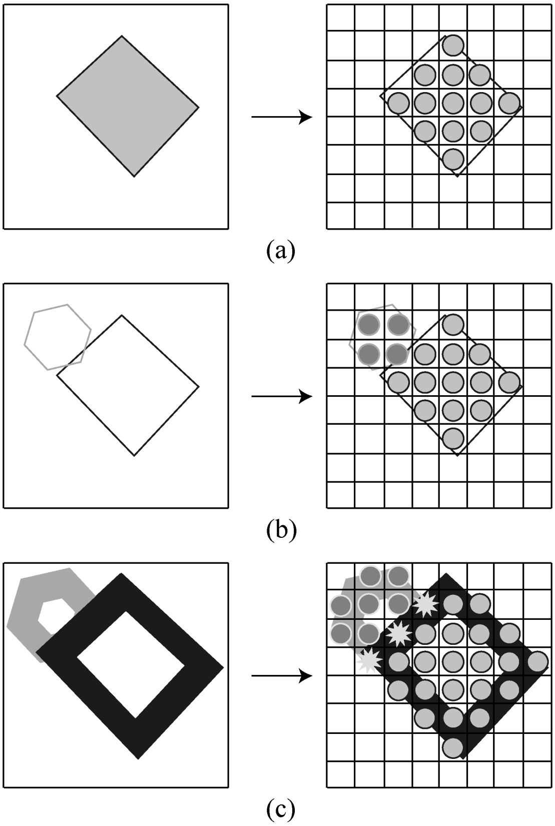
**Figure 12.** Object and gripper visibility test. (a) Object seen along $x$. (b) Partially visible object. (c) Object seen along $-x$. (d) Fully visible object. (e) Gripper seen along $-x$. (f) Fully visible gripper.

in Fig. 12f, and the number of visible pixels $q$ occupied by the gripper is counted. Only when $p = q$ is the gripper fully visible. It is the case in Fig. 12e and 12f. Therefore, the target object is determined to be accessible by the gripper along $-x$.

### *4.2. Rasterization for the Visibility Test*

The visibility test is accelerated by a powerful rasterization capability of graphics hardware. Rasterization of a primitive maps the primitive into the pixel grid, as shown in Fig. 13a. The rasterization process samples the primitive at each pixel's center to determine whether the pixel is included in the primitive or not.

The major strength of such an image-space algorithm is that it usually employs a graphics processing unit (GPU) which has been evolving at a rate faster than Moore's law and therefore it can show excellent performance. However, it also has a shortcoming: the accuracy is governed by the image-space resolution and consequently some overlapping primitives may be missed, as shown in Fig. 13b, where the two objects overlap, but do not share any pixel.



(a)

(b)

(c)

**Figure 13.** Handling of sampling error. (a) Rasterization. (b) Sampling error. (c) Rasterization with line drawing.

In order to overcome such a sampling error, the proposed algorithm adopts boundary offset for each primitive. The boundary offset operation is easily implemented using graphics hardware-supported line drawing. After rasterization, our algorithm renders each primitive one more time in the wireframe rendering mode. Figure 13c shows the primitives with lines drawn. The sum of normal rasterization and line drawing can overcome the sampling error and eventually find the intersection. In Fig. 13c, the star-shaped pixels are shared by the primitives.

### 4.3. Grasp Quality

Note that, as shown in Fig. 14, a target object can be grasped even when the gripper does not completely reach the sampled point. A simple and intuitive quality measure is devised using the gripper–object overlap in conjunction with the reachability and manipulability of the object:

$$aq_i = k_o go_i + k_d ad_i, \tag{1}$$

in which $aq_i$ is the accessibility quality of a sample point $i$, $go_i$ is the gripper–object overlap quality and $ad_i$ is the accessible direction quality. The weights associated to the qualities are denoted by $k_o$ and $k_d$.

Obviously, the more the gripper and object overlap, the higher the friction and the better the gripper–object overlap quality $go_i$. On the other hand, the accessible direction quality $ad_i$ is a representation of the manipulability and the reachability of the object. Given a robot position, the object's front faces/sides are easily reached and manipulated. In contrast, the back faces/sides of the object are more difficult to reach and harder to manipulate. The accessibility quality $aq_i$ is estimated for each grasp pose by the accessibility analysis algorithm.

There have been many works in the field of grasp analysis. For instance, Trinkle [24] provided a quality measure of 'how far a grasp is from losing form closure'. Ferrari and Canni [25] proposed a measure that gauges the largest disturbance wrench given a limited maximum contact force. In the framework proposed in this paper, the grasp analysis was not addressed. Instead, a simple and intuitive quality measure is used in order to combine the graspability and manipulability measures.

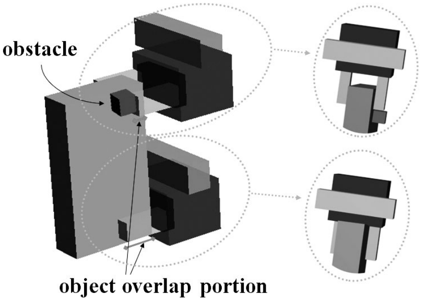


**Figure 14.** Gripper–object overlap depending on the obstacles.

An existing system such as Ref. [26] can be incorporated into the proposed framework.

### *4.4. Integration With Motion Planning*

When a grasp pose for the gripper is computed, the corresponding access/removal pose is also determined, as shown in Fig. 10. The gripper's linear path between the grasp pose and the access/removal pose is guaranteed to be collision-free. However, motion planning is still required to produce the collision-free path for the rest of the robot arm.

Among the parts of a robot, the gripper is closest to the cluttered environment around the target object and, therefore, motion planning for the gripper is usually the most time-consuming operation. The proposed approach increases the efficiency of the overall system because motion planning for the gripper is greatly simplified.

Outside the locality sphere, i.e., 'from the initial pose to the access pose' and 'from the removal pose to the goal pose' in Fig. 10, motion planning is needed for the entire arm. The proposed system can use any general motion planner, such as the Elastic Strip framework [27] or the Probabilistic Roadmap method [28]. The current implementation adopts the Elastic Strip framework. Note that the path generated by the motion planner can be input to the 3-D workspace modeler in order to provide adequate accuracy around the robot's path. As a result, the environment along the path can be gradually updated.

### *4.5. Experimental Results*

The proposed algorithms are implemented on a PC with a Pentium4 2.8 GHz CPU and NVIDIA GeForce 6600GT GPU. Figure 15 shows a series of snapshots for the manipulation task in the workspace of Fig. 1b. Figure 16 is a new experimental workspace and Fig. 17 is for the new workspace.

Table 2 shows the average time required for motion planning (Elastic Strip), with and without accessibility analysis for 20 experimental workspaces. In motion planning, the time is measured for the entire path from the initial pose to the goal pose. Note the performance gain between 4.769 ($= 0.02 + 4.749$) and 8.281 s.

## **5. Conclusion**

An integrated approach to human-like manipulation is presented. The behavior-oriented workspace modeling is not only novel but also significant, since it provides a service robot with the capability of human-like and real-time operations. Also, the proposed accessibility analysis approach follows a simplified behavior of humans, i.e., fully visible objects are determined to be retrievable without collision. The experimental results demonstrated that the proposed methods are sufficiently fast and robust to model 3-D workspace, and plan and execute a task in real-time for application to robotic manipulative tasks.

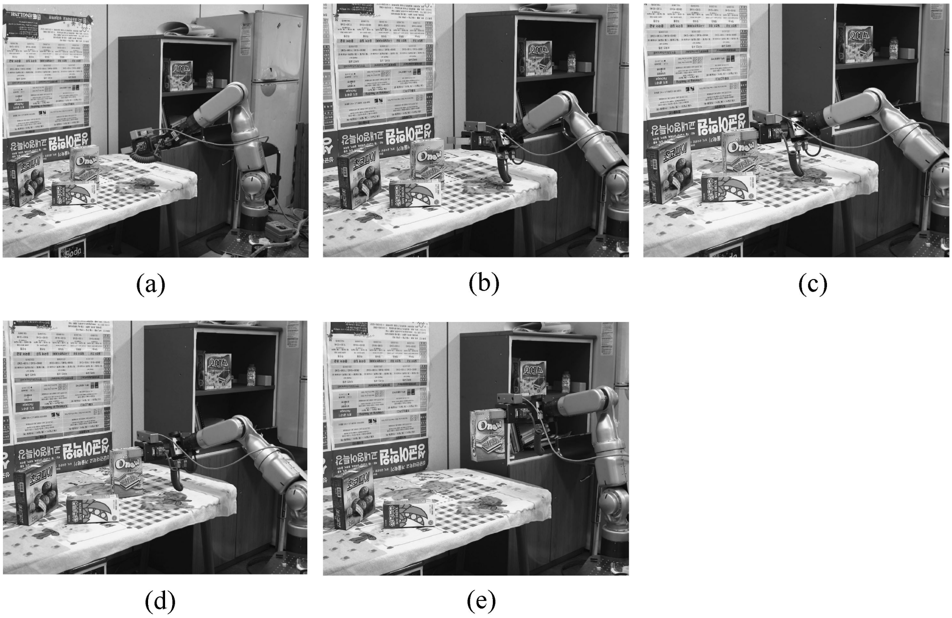


**Figure 15.** Snapshots of experiment 1. (a) Initial pose. (b) Access pose. (c) Grasp pose. (d) Removal pose. (e) Goal pose.

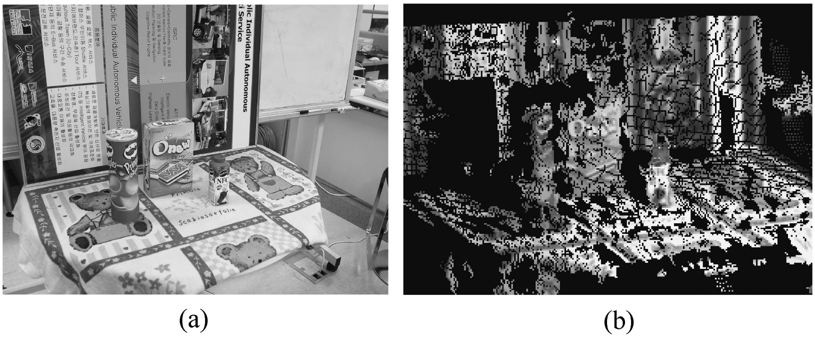


**Figure 16.** An experimental workspace. (a) Workspace. (b) Point clouds for the range data.

Note that the environment, as well as the objects used for the experimentation are assumed to have textures sufficient for obtaining 3-D point clouds as well as local features such as SIFT for recognition based on stereo cameras. However, the proposed integrated approach to visual manipulation can be implemented with other methods available for object recognition, as well as for collecting 3-D point clouds based on active sensors that do not depend on textures. Also, the proposed approach is not limited to the geometry of the objects and their layout on the table as used in this experimentation, as long as a 3-D model of the object to be grasped is given, and the access to and retrieval from the target object can be carried out by a straight

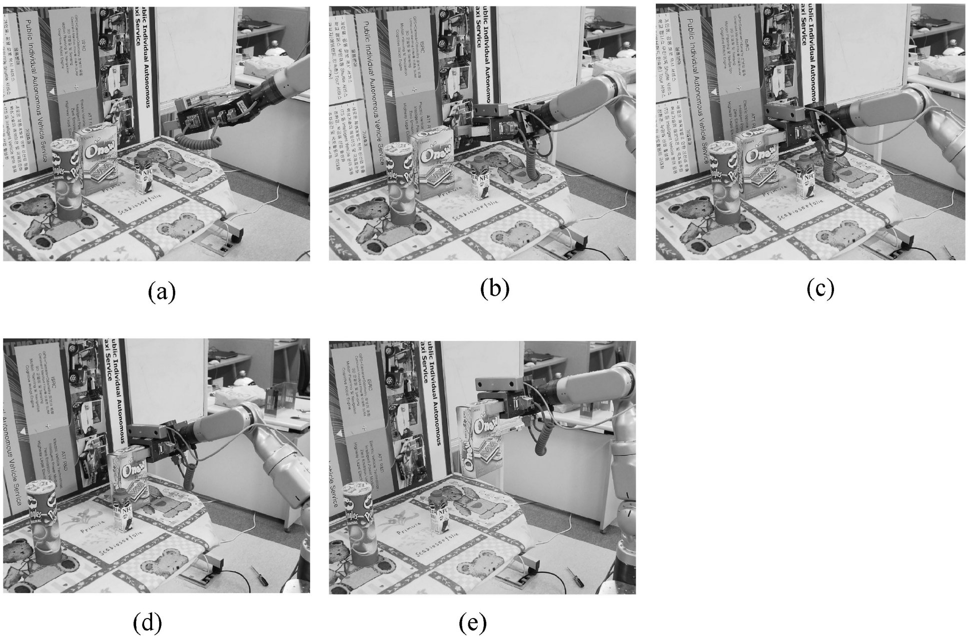


**Figure 17.** Snapshots of experiment 2. (a) Initial pose. (b) Access pose. (c) Grasp pose. (d) Removal pose. (e) Goal pose.

**Table 2.**
Computation time for accessibility analysis and motion planning

| Sub-modules | Average time (s) |
|---|---|
| Accessibility analysis | 0.02 |
| Motion planning with accessibility analysis | 4.749 |
| Motion planning without accessibility analysis | 8.281 |

line motion of the gripper. A general form of access and retrieval motions that requires non-straight line paths may be synthesized by a combination of straight-line motions by applying the proposed method to a series of intermediate target poses identified. However, how to determine intermediate target poses is a subject of further investigation.

## *Acknowledgements*

This work was performed for the 21st Century Frontier R&D Programs funded by the Ministry of Science and Technology of Korea. This work was also supported by MIC, Korea under ITRC IITA-2006-(C1090-0603-0046).

## References

1. A. Morales, G. Recatala, P. Sanz and A. Pobil, Heuristic vision-based computation of planner antipodal grasps on unknown objects, in: *Proc. IEEE Int. Conf. on Robotics and Automation*, Seoul, pp. 583–588 (2001).
2. K. Yamazaki, M. Tomono, T. Tsubouchi and S. Yuta, A grasp planning for picking up an unknown object for a mobile manipulator, in: *Proc. IEEE Int. Conf. on Robotics and Automation*, Orlando, FL, pp. 2143–2149 (2006).
3. M. Levoy, K. Pulli, B. Curless, S. Rusinkiewicz, D. Koller, L. Pereira, M. Ginzton, S. Anderson, J. Davis, J. Ginsberg, J. Shade and D. Fulk, The digital Michelangelo project: 3D scanning of large statues, in: *Proc. ACM SIGGRAPH 2000*, New Orleans, LA, pp. 131–144 (2000).
4. H. Surmann, A. Nüchter and J. Hertzberg, An autonomous mobile robot with a 3D laser range finder for 3D exploration and digitalization of indoor environments, *J. Robotics Autonomous Syst.* **45**, 181–198 (2003).
5. M. Garcia and A. Solanas, 3D simultaneous localization and modeling from stereo vision, in: *Proc. IEEE Int. Conf. on Robotics and Automation*, New Orleans, LA, pp. 847–853 (2004).
6. D. Huber, O. Carmichael and M. Hebert, 3D map reconstruction from range data, in: *Proc. IEEE Int. Conf. on Robotics and Automation*, San Francisco, CA, pp. 891–897 (2000).
7. Y. Liu, R, Emery, D. Chakrabarti, W. Burgard and S. Thrun, Using EM to learn 3D models of indoor environments with mobile robots, in: *Proc. Int. Conf. on Machine Learning*, Williamstown, MA, pp. 329–336 (2001).
8. S. Lee, D. Jang, E. Kim, S. Hong and J. Han, A real-time 3D workspace modeling with stereo camera, in: *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Edmonton, AB, pp. 2140–2147 (2005).
9. P. Besl and N. McKay, A method for registration of 3D shapes, *IEEE Trans. Pattern Anal. Machine Intell.* **14**, 239–256 (1992).
10. S. Se, D. Lowe and J. Little, Vision-based mapping with backward correction, in: *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Lausanne, pp. 153–158 (2002).
11. S. Spitz and A. Requicha, Accessibility analysis using computer graphics hardware, *IEEE Trans. Visual. Comp. Graphics* **6**, 208–219 (2000).
12. A. Spyridi and A. Requicha, Accessibility analysis for polyhedral objects, in: *Engineering Systems with Intelligence: Concepts, Tools, and Applications*, S. G. Tzafestas (Ed.), pp. 317–324. Kluwer, Dordrecht (1991).
13. H. Jang, H. Moradi, S. Lee and J. Han, A visibility-based accessibility analysis of the grasp points for real-time manipulation, in: *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Edmonton, AB, pp. 3111–3116 (2005).
14. H. Jang, S. Lee, D. Jang, E. Kim, H. Moradi and J. Han, A graphics hardware-based accessibility analysis for real-time robotic manipulation, in: *Proc. Int. Conf. Intelligent Computing*, Hefei, pp. 71–80 (2005).
15. J. Hummel and I. Biederman, Dynamic binding in a neural network for shape recognition, *Psychol. Rev.* **99**, 480–517 (1992).
16. M. Fischler and R. Bolles, Random sample consensus: a paradigm for model fitting with application to image analysis and automated cartography, *Commun. ACM* **24**, 381–395 (1981).
17. D. Lowe, Object recognition from local scale invariant features, in: *Proc. Int. Conf. on Computer Vision*, Kerkyra, 1150–1157 (1999).
18. Bumblebee. Available at http://www.ptgrey.com/
19. R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd edn. Cambridge University Press, Cambridge (2003).

20. H. Reif, Complexity of the mover's problem and generalizations, in: *Proc. 20th IEEE Symp. on Foundations of Computer Science*, San Juan, pp. 421–427 (1979).
21. J. Kuffner, K. Nishiwaki, S. Kagami, M. Inaba and H. Inoue, Motion planning for humanoid robots, in: *Proc. Int. Symp. of Robotics Research*, Siena (2003).
22. H. Jang, H. Moradi, S. Hong, S. Lee and J. Han, Spatial reasoning for real-time robotic manipulation, in: *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Beijing, pp. 2632–2637 (2006).
23. H. Jang, H. Moradi, P. Minh, S. Lee and J. Han, Visibility-based spatial reasoning for object manipulation in cluttered environments, *Computer-Aided Des.* **40**, 422–438 (2008).
24. J. Trinkle, A quantitative test for form closure grasps, in: *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Raleigh, NC, pp. 1670–1676 (1992).
25. C. Ferrari and J. Canny, Planning optimal grasps, in: *Proc. IEEE Int. Conf. on Robotics and Automation*, Nice, pp. 2290–2295 (1992).
26. A. T. Miller, GraspIt!: a versatile simulator for robotic grasping, *PhD Thesis*, Department of Computer Science, Columbia University (2001).
27. O. Brock, Generating robot motion: the integration of planning and execution, *PhD Thesis*, Stanford University (1999).
28. L. Kavraki and J.-C. Latombe, Probabilistic roadmaps for robot path planning, in: *Practical Motion Planning in Robotics*, K. Gupta and A. P. del Pobil (Eds), pp. 33–53. Wiley, Chichester (1998).

## About the Authors

**Sukhan Lee** is currently a Distinguished Professor of Information and Communications Engineering and the Director of the Intelligent Systems Research Center at Sungkyunkwan University (SKKU), South Korea. He also holds an Adjunct Professorship with the Georgia Institute of Technology as well as with the University of Southern California. Prior to his tenure with SKKU since 2003, he served as an Executive Vice President and Chief Research Officer for Samsung (SAIT), managing Intelligent Systems as well as Micro/Nano Systems Sectors. He was also with the Jet Propulsion Laboratory, California Institute of Technology, as a Senior Research Staff (1990–1997) and with the University of Southern California as a Faculty of Electrical Engineering and Computer Science (1983–1997). He received his PhD from Purdue University, West Lafayette, and MS and BS from Seoul National University. He is a Fellow of IEEE and a Fellow of Korea National Academy of Science and Technology. He currently directs the Industrial Activities Board as a Vice President of the IEEE Robotics and Automation Society. He is the Founding Editor-in-Chief for the Springer-Verlag journal Intelligent Service Robotics.

**Hadi Moradi** received his PhD degree in Computer Engineering (Robotics) in 1999 from the University of Southern California (USC). He is currently an Assistant Professor at the Electrical and Computer Engineering Department at Tehran University. Before joining Tehran University, he spent several years in manufacturing and the automation industry. He also conducted research at the Computer Science Department at USC and Intelligent Systems Research Center at Sungkyunkwan University. His research interests include motion planning, manipulation, environment modeling and educational robotics.

**Daesik Jang** received the BS, MS and PhD degrees in Computer Science from Soongsil University, South Korea, in 1994, 1996 and 1999, respectively. He worked at the research center of JulyNet Co., Ltd from 2000 to 2002. He also worked as a Research Leader at INCOM Co., Ltd until 2004. He spent the period from 2004 to 2005 in the Intelligent Systems Research Center at Sungkyunkwan University, South Korea, as a Research Leader. He is currently a Professor in the Department of Computer Information Engineering at Kunsan National University. His current research interests are computer vision, robot vision, intelligent vehicles and human–computer interaction.

**Hanyoung Jang** is currently a PhD student at Korea University, South Korea. He received the MS and BS degrees in the College of Information and Communications at Korea University. He specializes in computer graphics and is interested in real-time physics simulation. He has researched real-time collision detection and accessibility tests of robot arms since 2005.

**Eunyoung Kim** received the BS and MS degrees in Information and Communications Engineering from Sungkyunkwan University in 2003 and 2005, respectively. She has developed a module for 3-D object recognition and modeling at the Intelligent System Research Center from 2003 to 2006. She has been a PhD student in the Computer Science Department at the University of Southern California since 2006. Her research interests include non-rigid registration and range image understanding. She is currently working on 3-D object classification and segmentation in range data.

**Phuoc Minh Le** has been with Intelligent System Research Center, Sungkyunkwan University, South Korea, since 2005. He is also a Master candidate at the School of Mechanical Engineering at the same university. He received a BS degree in Mechanical Engineering at Hochiminh City University of Technology, Vietnam. His research interests include probabilistic motion planning and mobile manipulator visual servoing.

**JeongHyun Seo** is a master student in the Department of Electrical and Computer Engineering at Sungkyunkwan University, South Korea. He received the BS degree from the Department of Computer Engineering from Silla University, South Korea, in 2006. His research interests include robot vision, computer graphics, and robust object and workspace modeling based on multiple features.

**JungHyun Han** is an Associate Professor of the College of Information and Communications at Korea University, where he directs the Interactive 3-D Media Laboratory and Game Research Center supported by the Korean government. Prior to joining Korea University, he worked at the School of Information and Communications Engineering of Sungkyunkwan University, and at the Manufacturing Systems Integration Division of the US Department of Commerce National Institute of Standards and Technology. He received a BS degree in Computer Engineering at Seoul National University, an MS degree in Computer Science at the University of Cincinnati and a PhD degree in Computer Science at the University of Southern California. His research interests include geometric modeling, real-time rendering, and 3-D game design and development.