Learning to Open New Doors

Ellen Klingbeil, Ashutosh Saxena, Andrew Y. Ng
Computer Science Department,
Stanford University, Stanford, CA 94305
{ellenrk,asaxena,ang}cs.stanford.edu

***Abstract*— As robots enter novel, uncertain home and office environments, they are able to navigate these environments successfully. However, to be practically deployed, robots should be able to manipulate their environment to gain access to new spaces, such as by opening a door and operating an elevator. This, however, remains a challenging problem because a robot will likely encounter doors (and elevators) it has never seen before.**

Objects such as door handles are very different in appearance, yet similar function implies similar form. These general, shared visual features can be extracted to provide a robot with the necessary information to manipulate the specific object and carry out a task. For example, opening a door requires the robot to identify the following properties: (a) location of the door handle axis of rotation, (b) size of the handle, and (c) type of handle (left-turn or right-turn). Given these keypoints, the robot can plan the sequence of control actions required to successfully open the door. We identify these "visual keypoints" using vision-based learning algorithms. Our system assumes no prior knowledge of the 3D location or shape of the door handle. By experimentally verifying our algorithms on doors not seen in the training set, we advance our work towards being the first to enable a robot to navigate to more spaces in a new building by opening doors and elevators, even ones it has not seen before.

I. Introduction

Recently, there is growing interest in using robots not only in controlled factory environments but also in unstructured home and office environments. In the past, successful navigation algorithms have been developed for robots in these environments. However, to be practically deployed, robots must also be able to manipulate their environment to gain access to new spaces. In this paper, we will discuss our work towards enabling a robot to autonomously navigate anywhere in a building by opening doors and elevators, even those it has never seen before.

Most prior work in door opening (e.g., [1, 2]) assumes that a detailed 3D model of the door (and door handle) is available, and focuses on developing the control actions required to open one specific door. In practice, a robot must rely on only its sensors to perform manipulation in a new environment. However, most modern 3D sensors, such as a laser range finder, swissranger depth camera, or stereo camera, often provide sparse and noisy point clouds. In grasping, some recent works (e.g., [3, 4]) use learning to address this problem. Saxena et al. [3] use a vision-based learning approach to choose a point at which to grasp an object. However, a task such as opening a door is more involved in that it requires a series of manipulation tasks; the robot must first plan a path to reach the handle and then apply a series of forces/torques (which may vary in magnitude and direction) to open the door.

Fig. 1. Variety of manipulation tasks required for a robot to navigate in the environment.

The vision algorithm must be able to infer more information than a single grasp point to allow the robot to plan and execute such a manipulation task.

In this paper, we focus on the problem of manipulation in novel environments where a detailed 3D model of the object is not available. We note that objects, such as door handles, vary significantly in appearance, yet similar function implies similar form. Therefore, we will design vision-based learning algorithms that attempt to capture the visual features shared across different objects that have similar functions. To perform a manipulation task, such as opening a door, the robot end-effector must move through a series of way-points in cartesian space, while achieving the desired orientation at each way-point, in order to turn the handle and open the door. A small number of keypoints such as the handle's axis of rotation and size provides sufficient information to compute such a trajectory. We use vision to identify these "visual keypoints," which are required to infer the actions needed to perform the task. To open doors or elevators, there are various types of actions a robot can perform; the appropriate set of actions depends on the type of control object the robot must manipulate, e.g., left/right turn door handle, spherical doorknob, push-bar door handle, elevator button, etc. Our algorithm learns the visual features that indicate the appropriate type of control action to use.

For a robot to successfully open a door or elevator, it also needs to plan a collision-free path to turn and push or pull the handle while moving the robot base. For this purpose, we use a motion planning algorithm. We test our algorithm on a mobile manipulation platform, where we integrate different components—vision, navigation, planning, control, etc., to perform the task of opening the door.

Finally, to demonstrate the robustness of our algorithms, we provide results from extensive experiments on 20 different doors in which the robot was able to reliably open new doors in new buildings, even ones which were seen for the first time by the robot (and the researchers working on the algorithm).

II. Related Work

Our work draws ideas from a variety of fields, such as computer vision, grasping, planning, control, etc.; we will briefly discuss some of the related work in these areas.

There has been a significant amount of work done in robot navigation [5]. Many of these use a SLAM-like algorithm with a laser scanner for robot navigation. Some of these works have, in fact, even identified doors [6, 7, 8, 9, 10]. However, all of these works assumed a *known* map of the environment (where they could annotate doors); and more importantly none of them considered the problem of enabling a robot to autonomously open doors.

In robotic manipulation, most work has focused on developing control actions for different tasks, such as grasping objects [11], assuming a perfect knowledge of the environment (in the form of a detailed 3D model). Recently, some researchers have started using vision-based algorithms for some applications, e.g. [12]. Although some researchers consider using vision or other sensors to perform tasks such as grasping [13, 14], these algorithms do not apply to manipulation problems where one needs to estimate a full trajectory of the robot and also consider interactions with the object being manipulated.

There has been some recent work in opening doors using manipulators [15, 16, 17, 1, 2]; however, these works focused on developing control actions assuming a pre-surveyed location of a known door handle. In addition, these works implicitly assumed some knowledge of the type of door handle, since a turn lever door handle must be grasped and manipulated differently than a spherical door knob or a push-bar door.

Little work has been done in designing autonomous elevator-operating robots. Notably, [18] demonstrated their robot navigating to different floors using an elevator, but their training phase (which requires that a human must point out where the appropriate buttons are and the actions to take for a given context) used the same elevator as the one used in their test demonstration. Other researchers have addressed the problem of robots navigating in elevators by simply having the robot stand and wait until the door opens and then ask a human to press the correct floor button [19, 20]. Kemp et al. [21] used human assistance ("point and click" interface) for grasping objects.

In the application of elevator-operating robots, some robots have been deployed in places such as hospitals [22, 23]. However, expensive modifications must be made to the elevators, so that the robot can use a wireless communication to command the elevator. For opening doors, one can also envision installing automatic doors, but our work removes the need to make these expensive changes to the many elevators and doors in a typical building.

TABLE I

Visual keypoints for some manipulation tasks.

Manipulation task	Visual Keypoints
Turn a door handle	1. Location of the handle 2. Its axis of rotation 3. Length of the handle 4. Type (left-turn, right-turn, etc.)
Press an elevator button	1. Location of the button 2. Normal to the surface
Open a dishwasher tray	1. Location of the tray 2. Direction to pull or push it

In contrast to many of these previous works, our work does not assume existence of a known model of the object (such as the door, door handle, or elevator button) or a precise knowledge of the location of the object. Instead, we focus on the problem of manipulation in novel environments, in which a model of the objects is not available, and one needs to rely on noisy sensor data to identify visual keypoints. Some of these keypoints need to be determined with high accuracy for successful manipulation (especially in the case of elevator buttons).

III. Algorithm

Consider the task of pressing an elevator button. If our perception algorithm is able to infer the location of the button and a direction to exert force in, then one can design a control strategy to press it. Similarly, in the task of pulling a drawer, our perception algorithm needs to infer the location of a point to grasp (e.g., a knob or a handle) and a direction to pull. In the task of turning a door handle, our perception algorithm needs to infer the size of the door handle, the location of its axis, and a direction to push, pull or rotate.

More generally, for many manipulation tasks, the perception algorithm needs to identify a set of properties, or "visual keypoints" which define the action to be taken. Given these visual keypoints, we use a planning algorithm that considers the kinematics of the robot and the obstacles in the scene, to plan a sequence of control actions for the robot to carry out the manipulation task.

Dividing a manipulation task into these two parts: (a) an algorithm to identify visual keypoints, and (b) an algorithm to plan a sequence of control actions, allows us to easily extend the algorithm to new manipulation tasks, such as opening a dishwasher. To open a dishwasher tray, the visual keypoints would be the location of the tray and the desired direction to move it. This division acts as a bridge between state of the art methods developed in computer vision and the methods developed in robotics planning and control.

A. Identifying Visual Keypoints

Objects such as door handles vary significantly in appearance, yet similar function implies similar form. Our learning algorithms will, therefore, try to capture the visual features that are shared across different objects having similar function.

In this paper, the tasks we consider require the perception algorithm to: (a) locate the object, (b) identify the particular sub-category of object (e.g., we consider door handles of left-turn or right-turn types), and (c) identify some properties such as the surface normal or door handle axis of rotation. An estimate of the surface normal helps indicate a direction to push or pull, and an estimate of the door handle's axis of rotation helps in determining the action to be performed by the arm.

In the field of computer vision, a number of algorithms have been developed that achieve good performance on tasks such as object recognition [24, 25]. Perception for robotic manipulation, however, goes beyond object recognition in that the robot not only needs to locate the object but also needs to understand what task the object can perform and how to manipulate it to perform that task. For example, if the intention of the robot is to enter a door, it must determine the type of door handle (i.e., left-turn or right-turn) and an estimate of its size and axis of rotation, in order to compute the appropriate action (i.e., to turn the door handle left and push/pull).

Manipulation tasks also typically require more accuracy than what is currently possible with most classifiers. For example, to press an elevator button, the 3D location of the button must be determined within a few millimeters (which corresponds to a few pixels in the image), or the robot will fail to press the button. Finally, another challenge in designing perception algorithms is that different sensors are suitable for different perception tasks. For example, a laser range finder is more suitable for building a map for navigation, but a 2D camera is a better sensor for finding the location of the door handle. We will first describe our image-based classifier.

1) Object Recognition: To capture the visual features that remain consistent across objects of similar function (and hence appearance), we start with a 2D sliding window classifier. We use a supervised learning algorithm that employs boosting to compute a dictionary of Haar features.

In detail, the supervised training procedure first randomly selects ten small windows to produce a dictionary of Haar features [26]. In each iteration, it trains decision trees using these features to produce a model while removing irrelevant features from the dictionary. Figure 2 shows a portion of the patch dictionary selected by the algorithm.[1] Now, when given a new image, the recognizer identifies bounding boxes of candidate locations for the object of interest.

There are a number of contextual properties that we take advantage of to improve the classification accuracy. Proximity of objects to each other and spatial cues, such as that a door handle is less likely to be found close to the floor, can be used to learn a location based prior (partly motivated by [26]).

Fig. 2. Example features found by our Haar-Boosting-Kmeans classifier.

We experimented with several techniques to capture the fact that the labels (i.e., the category of the object found) have correlation. An algorithm that simply uses non-maximal suppression of overlapping windows for choosing the best candidate locations resulted in many false positives—12.2% on the training set and 15.6% on the test set. Thus, we implemented an approach that takes advantage of the context for the particular objects we are trying to identify. For example, we know that doors (and elevator call panels) will always contain at least one handle (or button) and never more than two handles. We can also expect that if there are two objects, they will lie in close proximity to each other and they will likely be horizontally aligned (in the case of door handles) or vertically aligned (in the case of elevator call buttons). This approach resulted in much better recognition accuracy.[2]

Figure 3 shows some of the door handles identified using our algorithm. In our earlier version of the algorithm, we used Support Vector Machines on a small set of features (computed from PCA).[3] Table II shows the recognition and localization accuracies. "Localization accuracy" is computed by assigning a value of 1 to a case where the estimated location of the door handle or elevator button was within 2 cm of the correct location and 0 otherwise. An error of more than 2 cm would cause the robot arm to fail to grasp the door handle (or push the elevator button) and open the door.

Once the robot has identified the location of the object in an image, it needs to identify the object type and infer control actions from the object properties to know how to manipulate it. Given a rectangular patch containing an object, we classify what action to take. In our experiments, we considered three types of actions: turn left, turn right, and press. The accuracy

[1]Details: We trained 50 boosting iterations of weak decision trees with 2 splits using a base window size of 84 x 48 pixels. To select the optimal values of parameters, e.g., number of components used, type of kernel, etc., we used a cross-validation set. We implemented this object recognizer on left and right door handles and elevator call panel buttons. The door handle training set consisted of approximately 300 positive and 6000 negative samples, and the elevator call button training set consisted of approximately 400 positive and 1500 negative samples.

[2]In detail, we start with windows that have high probability of containing the object of interest. These candidate windows are then grouped using K-means clustering; the number of clusters are determined from the histogram of the candidate window locations. In the case of one cluster, the cluster centroid gives the best estimate for the object location. In the case of two or more clusters, the centroid of the cluster with highest probability (the one with the most candidate frames) is identified as the most likely location for an object.

[3]*SVM-PCA-Kmeans:* For locating the object, we compute features that were motivated in part by some recent work in computer vision [24, 27] and robotic grasping [13]. The features are designed to capture three different types of local visual cues: texture variations, texture gradients, and color, by convolving the intensity and color channels of the image with 15 filters (9 Laws' masks and 6 oriented edge filters). We compute the sum of energies of each of these filter outputs, resulting in an initial feature vector of dimension 45. To capture more global properties, we append the features computed from neighboring patches (in a 4x4 grid around the point of interest). We then use PCA to extract the most relevant features from this set. Finally, we use the Support Vector Machines (SVM) [28] learning algorithm to predict whether or not an image patch contains a door handle or elevator button. This gave an accuracy of 91.2% in localization of door handles.

Fig. 3. Results on test set. The green rectangles show the raw output from the classifiers, and the blue rectangle is the one after applying context.

TABLE II
ACCURACIES FOR RECOGNITION AND LOCALIZATION.

	RECOGNITION	LOCALIZATION
DOOR HANDLE	94.5%	93.2%
ELEVATOR BUTTONS	92.1%	91.5%

of the classifier that distinguishes left-turn from right-turn handles was 97.3%.

2) Estimates from 3D data: Our object recognition algorithms give a 2D location in the image for the visual keypoints. However, we need their corresponding 3D locations to be able to plan a path for the robot.

In particular, once an approximate location of the door handle and its type is identified, we use 3D data from the stereo camera to estimate the axis of rotation of the door handle. Since, the axis of a right- (left-) turn door handle is the left- (right-) most 3D point on the handle, we build a logistic classifier for door-axis using two features—distance of the point from the door and its distance from the center (towards left or right). Figure 4 shows an example of the door-axis found from the 3D point-cloud. Similarly, we use PCA on the local 3D point cloud to estimate the orientation of the surface—required in cases such as elevator buttons and doors for identifying the direction to apply force.

However, the data obtained from a stereo sensor is often noisy and sparse in that the stereo sensor fails to give depth measurements when the areas considered are textureless, e.g., blank elevator walls [29]. Therefore, we also present a method to fuse the 2D image location (inferred by our object recognition algorithm), with a horizontal laser scanner (available on many mobile robots) to obtain the 3D location of the object. Here, we make a ground-vertical assumption—that every door is vertical to a ground-plane [30].[4] This enables our approach to be used on robots that do not have a 3D sensor such as a stereo camera (that are often more expensive).

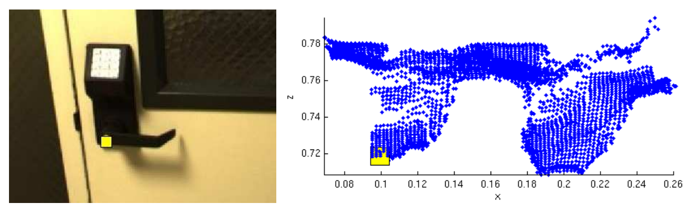

Fig. 4. The rotation axis of the door handle, shown by the yellow rectangle in the image (left) and in the point-cloud (right, showing top-view). Notice the missing points in the center of the handle.

B. Planning and Control

Given the visual keypoints and the goal, we need to design motion planning and control algorithms to allow the robot to

[4]In detail, a location in the image corresponds to a ray in 3D, which would intersect the plane in which the door lies. Let the planar laser readings be denoted as $l_i = (x(\theta_i), y(\theta_i))$. Let the origin of the camera be at $c \in \mathbb{R}^3$ in arm's frame, and let $r \in \mathbb{R}^3$ be the unit ray passing from the camera center through the predicted location of the door handle in the image plane. I.e., in the robot frame, the door handle lies on a line connecting c and $c + r$.

Let $T \in \mathbb{R}^{2\times 3}$ be a projection matrix that projects the 3D points in the arm frame into the plane of the laser. In the laser plane, therefore, the door handle is likely to lie on a line passing through Tc and $T(c + r)$.

$$t^* = \min_t \sum_{i\in\Psi} ||T(c + rt) - l_i||_2^2 \quad (1)$$

where Ψ is a small neighborhood around the ray r. Now the location of the 3D point to move the end-effector to is given by $s = c + rt^*$.

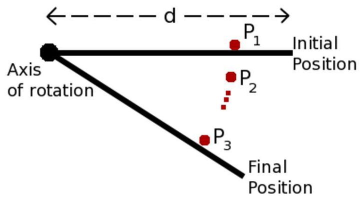

Fig. 5. An illustration showing how to obtain the locations of the end-effector from the visual keypoints.

successfully execute the task. The planning algorithm should consider the kinematics of the robot and also criterion such as obstacle avoidance (e.g., opening a dishwasher tray without hitting the objects in the tray).

For example, to turn a door handle the robot needs to move the end-effector in an arc centered at the axis of rotation of the door handle. (See Figure 5.) The visual keypoints such as length of the door handle d and the axis of rotation were estimated from the vision-based learning algorithms. Using these keypoints, we can compute the desired locations $P_i \in \mathbb{R}^3$ of the end-effector during the manipulation task.

To determine the correct control commands, we find the joint angles of the robot that will take the end-effector through the locations P_i. The robot must pass through these landmarks in configuration space; however, the problem of computing joint angle configurations from end-effector locations is ill-posed. Therefore, we use additional criterion such as keeping the wrist aligned with the axis of rotation and preventing the joints from reaching their limits or the arm from hitting any obstacles. To plan such paths, we build upon a Probabilistic RoadMap (PRM) [31] motion planning algorithm for obtaining a smooth, collision-free path for the robot to execute.

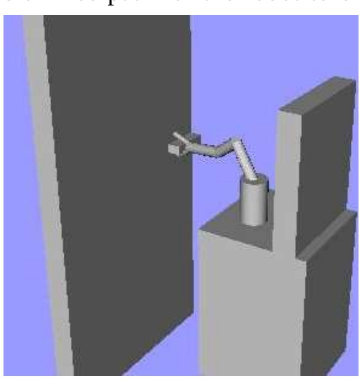

Fig. 6. Planning a path to open the door.

IV. Experiments

A. Robot

Our robotic platform (which we call STAIR 1) consists of a harmonic arm (Katana, by Neuronics) mounted on a Segway robotic mobility platform. The 5-dof arm is position-controlled and has a parallel-plate gripper. Our vision system uses a Point Grey Research stereo camera (Bumblebee XB3) and a laser scanner (Hokuyo) mounted on a frame behind the robotic arm.

B. Experiments

We used a Voronoi-based global planner for navigation [32]; this enabled the robot to localize itself in front of and facing a door (or elevator panel) within 20cm and ± 20 degrees. An experiment began with the robot starting at a random location within 3m of the door. It used lasers to navigate to the door, and our vision-based classifiers to find the handle.

TABLE III

Error rates obtained for the robot opening the door in a total number of 34 trials.

Door Type	Num of trials	Recog. (%)	Class. (%)	Localization (cm)	Success-rate
Left	19	89.5%	94.7%	2.3	84.2%
Right	15	100%	100%	2.0	100%
Total	34	94.1%	97.1%	2.2	91.2%

Fig. 7. Some experimental snapshots showing our robot opening different types of doors.

In the experiments, our robot saw all of our test locations for the first time. The training images for our vision-based learning algorithm were collected in completely separate buildings, with different doors and door handle shapes, structure, decoration, ambient lighting, etc. We tested our algorithm on two different buildings on a total of five different floors (about *20 different* doors). Many of the test cases were also run where the robot localized at different angles, typically between -30 and +30 degrees with respect to the door, to verify the robustness of our algorithms.

In a total of 34 experiments, our robot was able to successfully open the doors *31 out of 34* times. Table III details the results; we achieved an average recognition accuracy of 94.1% and a classification accuracy of 97.1%. We define the localization error as the mean error (in cm) between the predicted and actual location of the door handle. This led to a success-rate (fraction of times the robot actually opened the door) of 91.2%. Notable failures among the test cases included glass doors (erroneous laser readings), doors with numeric keypads, and very dim/poor lighting conditions. These failure cases have been reduced significantly (in simulation) with the new classifier. (The current experiments were run using our earlier svm-pca-kmeans classifier.)

For elevator button pushing and door pulling experiments, we have only performed single demonstrations on the robot. Due to the small size of the elevator buttons (2 cm diameter) and the challenge of obtaining very accurate arm-vision system calibration, reliably pushing the buttons is much more difficult, even if the simulations show high performance. Our robot has a fairly weak gripper, and therefore pulling the door open is

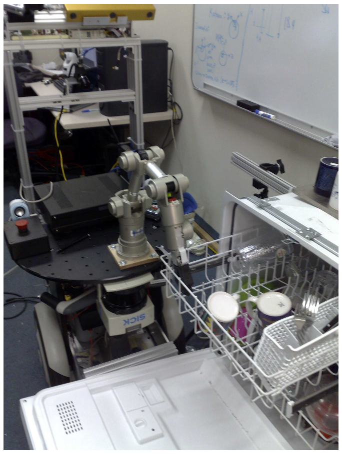

Fig. 8. Snapshots showing our robot opening a dishwasher tray.

also difficult because of the very small effective workspace in which it can exert enough torque to open a door. Also many of the doors are spring-loaded, making it impossible for this particular arm to pull them open. In future work, we plan to use active vision [33], which takes visual feedback into account while objects are being manipulated and thus provides complementary information that would hopefully improve the performance on these tasks.

Videos of the robot opening new doors and elevators are available at:

http://ai.stanford.edu/∼asaxena/openingnewdoors

To demonstrate how our ideas can be extended to more manipulation tasks, we also tested our algorithms on the task of opening a dishwasher tray in a kitchen. Using our 3D classifiers, we identified the location of the tray and the visual keypoints, i.e., the direction in which to pull the tray open. Here, training and testing was done on same dishwasher but test cases had different objects/position of the tray as compared to the training set. By executing the planned path, the robot was able to pull out the dishwasher tray (Figure 8).

V. CONCLUSION

To navigate and perform tasks in unstructured environments, robots must be able to perceive their environments to identify what objects to manipulate and how they can be manipulated to perform the desired tasks. We presented a framework that identifies some visual keypoints using our vision-based learning algorithms. Our robot was then able to use these keypoints to plan and execute a path to perform the desired task. This strategy enabled our robot to navigate to new places in a new building by opening doors and elevators, even ones it had not seen before. In the future, we hope this framework will aid us in developing algorithms for performing a variety of manipulation tasks.

Acknowledgments: We thank Andrei Iancu, Srinivasa Rangan, Morgan Quigley and Stephen Gould for useful discussions and for their help in the experiments.

REFERENCES

[1] M. Prats, P. Sanz, and A. P. del Pobil, "Task planning for intelligent robot manipulation," in *IASTED Artificial Intel App*, 2007.

[2] A. Petrovskaya and A. Y. Ng, "Probabilistic mobile manipulation in dynamic environments, with application to opening doors," in *IJCAI*, 2007.

[3] A. Saxena, J. Driemeyer, and A. Y. Ng, "Robotic grasping of novel objects using vision," *IJRR*, vol. 27, no. 2, pp. 157–173, 2008.

[4] Platt, Grupen, and Fagg, "Improving grasp skills using schema structured learning," in *ICDL*, 2006.

[5] S. Thrun and M. Montemerlo, "The graph SLAM algorithm with applications to large-scale mapping of urban structures," *IJRR*, vol. 25, no. 5-6, pp. 403–429, 2006.

[6] D. Fox, W. Burgard, and S. Thrun, "Markov localization for mobile robots in dynamic environments," *JAIR*, vol. 11, pp. 391–427, 1999.

[7] D. Wolf and G. S. Sukhatme, "Online simultaneous localization and mapping in dynamic environments," in *ICRA*, 2004.

[8] C. Stachniss and W. Burgard, "Mobile robot mapping and localization in non-static environments," in *AAAI*, 2005.

[9] P. Biber and T. Duckett, "Dynamic maps for long-term operation of mobile service robots," in *RSS*, 2005.

[10] D. Anguelov, D. Koller, E. Parker, and S. Thrun, "Detecting and modeling doors with mobile robots," in *ICRA*, 2004.

[11] A. Bicchi and V. Kumar, "Robotic grasping and contact: a review," in *ICRA*, 2000.

[12] D. Kragic and H. I. Christensen, "Robust visual servoing," *IJRR*, vol. 22, no. 10-11, pp. 923–939, 2003.

[13] A. Saxena, J. Driemeyer, J. Kearns, and A. Y. Ng, "Robotic grasping of novel objects," in *NIPS*, 2006.

[14] A. Saxena, L. Wong, and A. Y. Ng, "Learning grasp strategies with partial shape information," in *AAAI*, 2008.

[15] C. Rhee, W. Chung, M. Kim, Y. Shim, and H. Lee, "Door opening control using the multi-fingered robotic hand for the indoor service robot," in *ICRA*, 2004.

[16] L. Petersson, D. Austin, and D. Kragic, "High-level control of a mobile manipulator for door opening," in *IROS*, 2000.

[17] D. Kim, J.-H. Kang, C. Hwang, and G.-T. Park, "Mobile robot for door opening in a house," *LNAI*, vol. 3215, pp. 596–602, 2004.

[18] J. Muira, K. Iwase, and Y. Shirai, "Interactive teaching of a mobile robot," in *ICRA*, 2005.

[19] R. Simmons and et al., "Grace: An autonomous robot for the AAAI robot challenge," *AI Magazine*, vol. 24, no. 2, pp. 51–72, 2003.

[20] F. Michaud, J. Audet, and D. Letourneau, "Having a robot attend AAAI 2000," *IEEE Intelligent Systems*, vol. 15, no. 6, 2000.

[21] C. C. Kemp, C. Anderson, H. Nguyen, A. Trevor, and Z. Xu, "A point-and-click interface for the real world: Laser designation of objects for mobile manipulation," in *HRI*, 2008.

[22] T. Sakai, H. Nakajima, D. Nishimura, H. Uematsu, and K. Yukihiko, "Autonomous mobile robot system for delivery in hospital," in *MEW Technical Report*, 2005.

[23] J. Evans, "Helpmate: an autonomous mobile robot courier for hospitals," in *IROS*, 1992.

[24] T. Serre, L. Wolf, and T. Poggio, "Object recognition with features inspired by visual cortex," in *CVPR*, 2005.

[25] B. Sapp, A. Saxena, and A. Y. Ng, "A fast data collection and augmentation procedure for object recognition," in *AAAI*, 2008.

[26] A. Torralba, "Contextual priming for object detection," *IJCV*, vol. 53, no. 2, pp. 169–191, 2003.

[27] A. Saxena, M. Sun, and A. Y. Ng, "Learning 3-d scene structure from a single still image," in *ICCV workshop on 3D Representation for Recognition (3dRR)*, 2007.

[28] V. N. Vapnik, *The Nature of Statistical Learning Theory*. Springer, 1995.

[29] A. Saxena, J. Schulte, and A. Y. Ng, "Depth estimation using monocular and stereo cues," in *IJCAI*, 2007.

[30] E. Delage, H. Lee, and A. Y. Ng, "A dynamic bayesian network model for autonomous 3d reconstruction from a single indoor image," in *CVPR*, 2006.

[31] F. Schwarzer, M. Saha, and J.-C. Latombe, "Adaptive dynamic collision checking for single and multiple articulated robots in complex environments," *IEEE Tr Robotics Auto*, vol. 21, no. 3, pp. 338–353, 2005.

[32] H. Choset and J. Burdick, "Sensor based planning. I. The generalized voronoi graph," in *ICRA*, 1995.

[33] D. Katz and O. Brock, "Extracting planar kinematic models using interactive perception," in *RSS Robot Manipulation workshop*, 2007.