Extracting Planar Kinematic Models Using Interactive Perception

Dov Katz and Oliver Brock
Robotics and Biology Laboratory
Department of Computer Science
University of Massachusetts Amherst

Abstract— **Interactive perception augments the process of perception with physical interactions. By adding interactions into the perceptual process, manipulating the environment becomes part of the effort to learn task-relevant information, leading to more reliable task execution. Interactions include obstruction removal, object repositioning, and object manipulation. In this paper, we show how to extract kinematic properties from novel objects. Many objects in human environments, such as doors, drawers, and hand tools, contain inherent kinematic degrees of freedom. Knowledge of these degrees of freedom is required to use the objects in their intended manner. We demonstrate how a simple algorithm enables the construction of kinematic models for such objects, resulting in knowledge necessary for the correct operation of those objects. The simplicity of the framework and its effectiveness, demonstrated in our experimental results, indicate that interactive perception is a promising perceptual paradigm for autonomous mobile manipulation.**

I. Introduction

Roboticists are working towards the deployment of autonomous mobile manipulators in unstructured and dynamic environments [2], [5], [6], [10], [12], [17], [19], [24], [25], [26], [29].

Adequate autonomy and competency in such environments would open up a variety of important applications for robotics, ranging from planetary exploration to elder care and from the disposal of improvised explosive devices to flexible manufacturing and construction in collaboration with human experts. For these applications, it is not possible to provide detailed *a priori* models of the environment. Consequently, an autonomous robot has to continuously acquire perceptual information about the world to successfully execute manipulation tasks in unstructured environments.

During task execution, the value of perceptual information can be maximized by interpreting sensor streams in a manner that is tailored to the task. Focusing on task-specific aspects during the interpretation of the sensor stream will reveal the most task-relevant information, while reducing the computational cost of perception. Both of these advantages can improve the robustness of task execution, particularly in the presence of significant uncertainty. In spite of the advantages of integrating perception and manipulation, research towards integrated perceptual paradigms for autonomous manipulation is still in its early stages.

In this paper, we argue that interactive perception, a framework that exploits forceful interactions with the environment to uncover perceptual information required for the robust execution of specific tasks, can serve as an adequate perceptual framework for autonomous manipulation. We will show that the combination of forceful interactions with visual perception reveals perceptual information unobtainable by forceful interactions or visual perception alone. Blurring the boundaries of manipulation and perception leads to novel perceptual capabilities, even when the manipulation and perception capabilities are very basic.

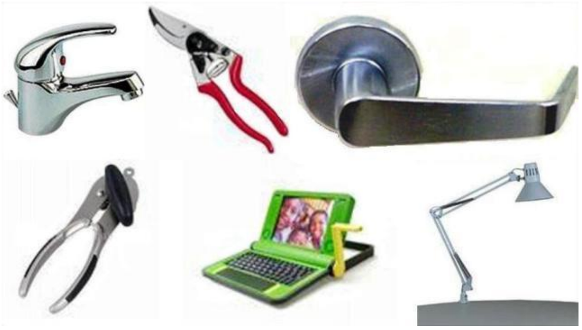

Fig. 1. Objects that possess inherent degrees of freedom; these degrees of freedom cannot be extracted from visual information alone, they have to be discovered through physical interaction

To illustrate the promise of interactive perception as a perceptual paradigm for autonomous robots operating in unstructured environments, we have developed a perceptual skill to extract kinematic models from unknown objects. Many objects in everyday environment possess inherent degrees of freedom that have to be actuated to perform their function. Such objects include door handles, doors, drawers, and a large number of tools such as scissors and pliers (Figure 1). In our experiments, UMan (Figure 2), our experimental platform for autonomous mobile manipulation, employs interactive perception to extract kinematic models from tools such as pliers or shears. These models are then employed to compute an action that transforms the kinematic state of the tool into a desired goal state, mimicking tool use. Note that kinematic models of objects cannot be extracted using visual information alone. They are also very hard to obtain from tactile feedback. We believe that the relative ease with which we are able to address this task makes a convincing case for the use of interactive perception as a perceptual paradigm for autonomous robotics.

II. Related Work

In the absence of a model, autonomous manipulation in unstructured environments depends on sensor streams to

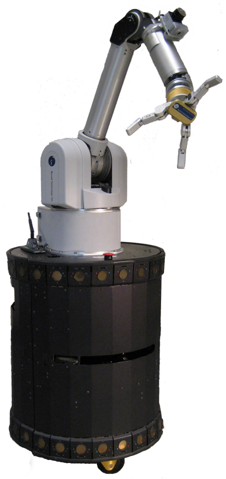

Fig. 2. UMan (UMass Mobile Manipulator)

assess the state of the world. The sensor streams should be interpreted and the resulting information can then be used to guide manipulation. In this section, we will discuss perceptual techniques that were developed independently of specific manipulation objectives as well as approaches that closely integrate perception and manipulation.

Computer vision researchers extensively explored object segmentation and labeling from static images [14]. These problems, which seem to be solved effortlessly by humans, were found to be quite challenging.

Active vision [1], [3], [4] represents a paradigm shift relative to computer vision based on static images. Now, the agent is no longer a passive observer but instead can control the motion of the sensor to actively extract relevant information. Active vision simplified the extraction of structure from visual input [22], [28] and facilitated depth estimation based on information about the camera's motion [20].

Visual servoing provides closed-loop position control for a robotic mechanism [15]. It is an example of how position control, one of the fundamental primitives of manipulation, can be greatly improved through integration with vision.

Although active vision greatly improves data acquisition, in some cases this process cannot generate the data required to support a specific task. For example, object segmentation and predicting kinematic and dynamic properties of rigid or articulated bodies remain great challenges even when the camera's position can be controlled. Prior work has shown that physical interaction with the world can remedy many of these difficulties.

Object segmentation can be solved by actively poking objects using a robotic manipulator [13], [23]. The generated optical flow allows the identification of moving objects and separates them from their background.

Tool use can be performed by treating tools in the context of their task. Instead of recovering the entire state of the world from sensor stream, Edsinger and Kemp [12], [11] focus on information that is task-relevant. This simplifies the perceptual process, and allows successful operation in environments that were not adapted to the robot. The work of Fitzpatrick, Metta, Edsinger and Kemp can be characterized as interactive perception.

Predicting the movement of objects in the plane can also be simplified by interaction. Christiansen, Mason, and Mitchell addressed this problem by placing objects on a tray which could be tilted by a robotic manipulator [7]. The robot actively tilted the tray to increase its knowledge about the objects' motion. This knowledge then facilitated successful task execution which required object displacements. Stoytchev used a predefined set of interactions with rigid objects (tools) to explore their affordances [27]. He extracted the results of tool use by the robot by visually observing the motion of rigid bodies in the workspace of the robot. This knowledge was then applied during task execution by selecting the most appropriate tool.

The last three examples demonstrate the positive effects that deliberate action has on the successful completion of tasks and on the difficulty of the perception problem. They represent a natural development from the active vision paradigm towards the interactive perception paradigm, in which robots can actively change the world to increase sensor range. The following section presents this paradigm, and explains how it can dramatically improve the capabilities of robots in unstructured and dynamic environments.

III. Interactive Perception

A robot can enhance its perceptual capabilities by including physical interactions with the environment in its perceptual repertoire. Such interactions can remove obstructions, provide an easy and controlled way of exposing multiple views of an object, or can alleviate the negative effects of lighting conditions by moving objects in the field of view. Other perceptual tasks are difficult or even impossible to accomplish without interacting with the environment. For example, reading the text in a closed book, checking whether a door is locked, and finding out the purpose of a switch mounted on the wall. Physical interactions augment the sensor stream with force feedback and allow to evoke and observe behaviors in the world that can reveal physical properties of objects. Such information would otherwise remain inaccessible for non-interactive sensors. Physical interactions thus can make traditional perceptual tasks easier. Moreover, they make a new class of perceptual information accessible to a robotic agent.

The promise of interactive perception [17] is supported by examples from the development of physical and mental skills in humans. During the acquisition of physical skills by infants, for example, physical interactions with the environment are necessary to bootstrap the cognitive process of learning the connection between action and effect, the kinematics of one's own body, and the properties and functions of objects in the environment.

Interacting with the environment as part of the perceptual process poses a challenge: selecting the most adequate interaction for a perceptual task. The need to choose the right exploratory skill while balancing between exploration and exploitation is not new. Active learning [8], a branch of machine learning, addresses the very same problem and has been shown to be highly effective. While in this paper we will focus on a single perceptual primitive to demonstrate the effectiveness of interactive perception, our future work will integrate this and other primitives into a perceptual framework that can actively select when and which interactive perceptual primitive to invoke.

IV. Obtaining Kinematic Models Through Forceful Interactions

In this section, we will present one instantiation of the interactive perception framework. We will demonstrate how a robotic manipulator can extract the kinematic properties of a tool lying on a table. No *a priori* knowledge about the tool is assumed. The robot can subsequently construct a model of the tool which will allow it to determine the appropriate interaction for using the tool. In this early work on interactive perception, we will restrict ourselves to revolute joints.

A. Algorithm

The key insight behind our algorithm is that the relative distance between two points on a rigid body does not change as the body is pushed. However, the distance between points on different rigid bodies connected by a revolute joint does change as the bodies rotate relative to each other.

First, we describe our algorithm for objects composed of two links connected by a single revolute joint. The robot interacts with a tool on the table by sweeping its end-effector across the surface. Tracking a set of features of the object throughout the interaction allows us to measure the distance between these features as the object is being moved. The features can be separated into three groups: features on the first link, features on the second link, and features on the joint connecting them. Features in the same group must maintain constant distance to each other, irrespective of the planar motion the object performs. However, the distance between features in the first group and features in the second group will change significantly as the object is being moved. The joint features are simply features that belong to both the first and the second group. This algorithm works also in the general case of multiple revolute joints. To identify the groups, a robotic manipulator interacts with the object to generate motions that will allow distinguishing between the different rigid bodies.

In order to determine the spatial extent of the links of the object, we construct a convex hull around the features in each group. Tracking enough features increases the match between the convex hull and the actual shape of the link. The length of each link is taken to be the distance between the furthest point in each group and the joint. We use this knowledge to create a kinematic model for planar kinematic chains. This model is later used to predict the actions required to manipulate the object in a meaningful fashion.

The following subsections describe in details the implementation of the kinematic model building algorithm. It is worth noting that the specific way in which we choose features, track them or analyze their relative motion does not affect the algorithm.

B. Tracking objects

Since our primary goal is to show the promise of interactive perception as a perceptual paradigm, we place objects on a plain white background, facilitating feature tracking. The white background assumption can be removed using ideas from active segmentation (similarly to [13] and [23]). This includes an initial random phase were the manipulator sweeps the environment in an attempt to segment objects. The interaction may provide interesting objects, for which we might want to construct a kinematic model.

We use the open source computer vision library OpenCV [16] to capture, record, and process images. OpenCV implements feature selection by finding corners with big eigenvalues, and feature tracking based on the optical flow algorithm of Lucas and Kanade [21]. We store the position of the automatically generated set of features in every frame during the interactive session.

The tracked features are selected before the interactive session begins. Some features may be obstructed by the manipulator's motion during the interaction. Those features will be very noisy, and therefore easily discarded. Moreover, no feature will be associated with the manipulator itself because all features are selected prior to the appearance of the arm in the scene.

C. Constructing a graphical representation

Every planar kinematic chain is composed of links and joints. Therefore, the first task we perform is joint and link identification. We build a graph based on the maximal change in distance between two features observed throughout the entire interaction. Every node $v \in V$ in the graph represents a tracked feature in the image. An edge $e \in E$ connects nodes (v_i, v_j) if and only if the distance between v_i and v_j remains constant (in practice, we allow the distance to vary up to a threshold). The resulting graph will be analyzed by the algorithm described in the following section.

D. Graph analysis

Figure 3 shows a schematic depiction of a graph constructed for an object with two joints. We can detect in this graph three groups of nodes; each group is very highly interconnected. The groups represent links, and high interconnectivity is the result of no motion between features on the same rigid body (link). The nodes that connect two groups represent joints, and therefore are highly connected to two groups.

We use the min-cut algorithm [9] to identify different groups in a graph. Min-cut will separate a graph into two sub-graphs by removing as few edges as possible. In the simple case of one joint and two links, min-cut will remove the edges that connect between the two links, resulting in two

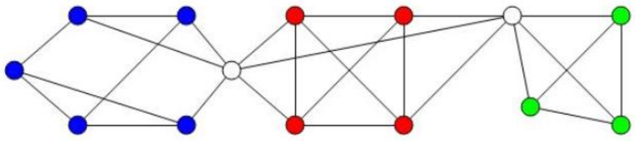

Fig. 3. Generated graph for an object with two degrees of freedom. Highly connected components (colored in blue, red and green) represent the links. Nodes that connect between components represent joints (colored in white).

highly connected sub-graphs, each representing a different link. In the general case, a graph may contain multiple highly connected components (each component represents a different link). Identifying components in this case is done simply by recursively breaking the graph into sub-graphs. The process stops when the input graph is highly connected, therefore represents one rigid component. Finally, nodes that belong to two different highly connected components are nodes that represent a joint connecting two links.

It should be noted that the above procedure automatically rejects errors in tracking features. If a feature "jumps" during tracking, the graph construction described above will lead to a disconnected component consisting of a single vertex. Disconnected nodes, and more generally small disconnected components, can be discarded during the graph analysis. The proposed procedure thus is inherently robust to errors in feature tracking.

E. Building a kinematic model

The graph provides us with information about the basic kinematic structure of the object. We can construct an approximate contour of the object by computing the convex hull of all features in a component of the graph. The extent of the visual hull gives us an approximate geometric description of each link. By combining all the information, we construct a kinematic model of the kinematic chain. This model enables the robot to reason about the effects that its interactions with the object will have. This is a prerequisite for purposeful tool use.

V. Experimental Results

We validate the method described above in experiments on our robotic platform for autonomous mobile manipulation, called *UMan* (see Figure 2, [18]). *UMan* consists of a holonomic mobile base with three degrees of freedom, a seven-degrees-of-freedom Barrett Technologies manipulator arm, and a four-degrees-of-freedom Barrett hand. The vision system is an overhead web camera, mounted above a desk. The camera's resolution is 640X480. The platform provides adequate end-effector capabilities for a wide range of dexterous manipulation tasks.

UMan is tasked to extract a kinematic model of four different tools, shown in Figure 5. To demonstrate that *UMan* can use the kinematic model for purposeful interactions with those tools, it is required to push the tool until the two rigid links form a right angle. *UMan* first uses its end-effector to sweep the table in front of it, while observing the scene. Features are tracked in the resulting video sequence and the algorithm described above is used to extract a kinematic model of the tool. Using this model, *UMan* determines the appropriate pushing motion to achieve the desired angle between the two links and performs this motion. An example of such an experiment and the corresponding visual observation is shown in Figure 4.

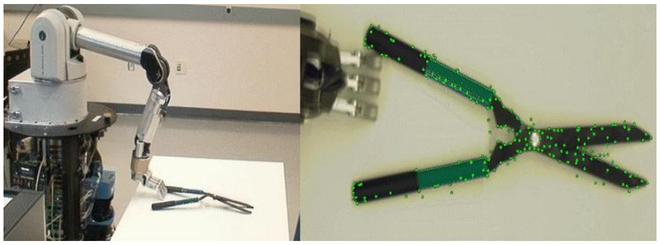

Fig. 4. UMan interacts with a tool by reaching its arm towards the tool. The right image shows the tool as seen by the robot, with dots marking the tracked features. The left image shows the experimental setting

Four tools were used in the experimental phase: scissors, shears, plier, and a stapler. All four tools have a single revolute joint, with the exception of the pliers which also have a prismatic joint that was ignored. The tools are off-the-shelf products and have not been modified for our experiments. They vary in scale, shape, and color. Despite the differences in appearance, all four tool belong to the family of two-link kinematic chains with a single revolute joint.

Figure 5 shows in each row snapshots of the experiment with one of the four tools. Each column shows a particular phase of the experiment. First, the tools are in their initial pose (before the interaction begins). Next, we see the tools in their final pose (after the interaction). The third column shows the location of the joints, as detected by interacting with the tools. In the fourth column, two green lines mark the position of the parts of the links that will be used for the purposeful interaction. A third red line indicates where one of the links needs to be moved to in order to create a right angle between the links. Finally, the last column shows the tools after the execution of the plan from the previous column—each tool was manipulated to form an angle of 90°.

The experimental results show that the detection of the revolute joint is very accurate. Moreover, the length and position of the links are also discovered correctly. The algorithm uses the information collected in the interactive process to create kinematic models for the tools. The high accuracy and usefulness of these models is demonstrated by the successful manipulation of the tools to form an angle of 90° between the links.

Figure 6 show an additional experiment with an object that possesses multiple degrees of freedom. Without any modification, the algorithm described above successfully identifies the two degrees of freedom. However, in our experiment, the first interaction only revealed a single degree of freedom. The second degree of freedom had to be extracted using another interaction. This illustrates the need to embed

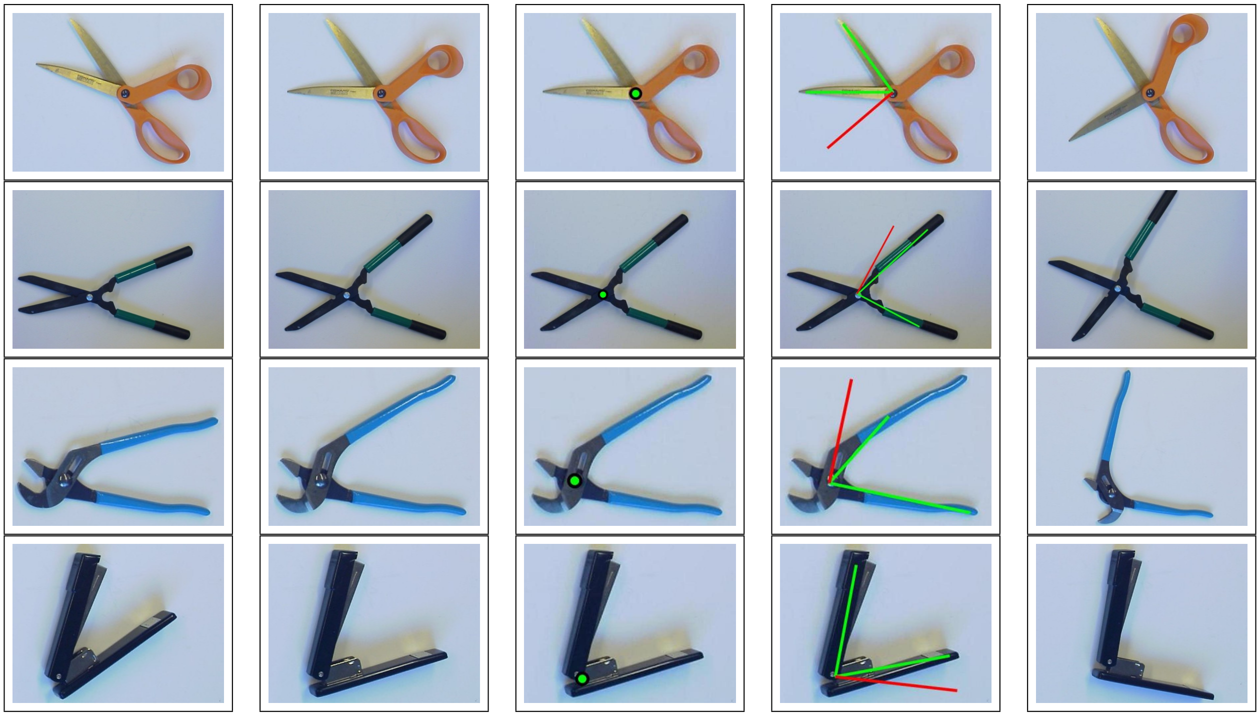

Fig. 5. Experimental results showing the use of interactive perception in extracting the kinematic properties of different objects. The first column of images shows the four objects (scissors, shears, plier, and stapler) in their initial pose. The second column shows the final pose of the four objects after the robot has interacted with them. The third column shows the revolute joint that was detected using the methods described in this paper; the revolute joint is marked with a green circle. The fourth column of images shows the links of the obtained kinematic model and the manipulation plan to form a right angle between the two links of the tools. Putting the two links into a 90° angle here serves as an example of tool use. The links of the tools are shown as green lines, and the orientation of one of the links to achieve the goal configuration of the tool is marked by a red line. The last column of images shows the results of executing the manipulation plan as presented in the previous column: the two links of the tools have been arranged in a 90° angle.

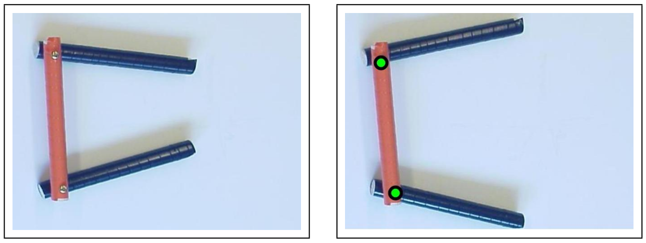

Fig. 6. Experimental results showing the use of interactive perception in extracting the kinematic properties of objects with two degrees of freedom.

the perceptual primitive described here into a higher-level perceptual process.

In all of our experiments, the proposed algorithm was able to extract the kinematic axes of the tools with great precision, despite the cheap off-the-shelf web camera that was used. Only a small displacement of the object was required. The algorithm does not have any parameters that need to be tuned. The performance of the algorithm was extremely robust, all of our experiments “just worked.” No changes were necessary to the algorithm to deal with the five objects, even though their size, visual appearance, and kinematic properties varied. Furthermore, our experiments have shown that the algorithm is insensitive to the distance of the camera to the object. The algorithm also performs without errors for a broad range of viewing angles. Even though we initially assumed that the view direction would be orthogonal to the surface of the table, the algorithm tolerates deviations of up to 30°. We have not explicitly tested this parameter but suspect that even higher deviations will continue to give good results.

The experiments discussed here demonstrate that the combination of two very fundamental capabilities, namely feature tracking and object pushing, yields a highly robust and accurate perceptual primitive. This primitive is able to extract perceptual information that neither of its two components could extract by themselves. The experiments thus demonstrate that interactive perception can increase the perceptual capabilities of a robot while at the same time improving the robustness of the perceptual process. As stated in the introduction, we believe that this is the consequence of combining manipulation and perception to develop a task-related perceptual process. We are convinced that interactive perception represents an important step towards the robust execution of autonomous manipulation tasks in unstructured environments.

VI. Conclusion

This paper explores interactive perception as an adequate perceptual paradigm for autonomous robots. Interactive perception tightly couples interaction and perception to enable the robust and efficient extraction of task-relevant information from sensor streams. The inclusion of interaction into the repertoire of perceptual primitives not only facilitates many conventional perception tasks, but also allows an autonomous agent to uncover information about the environment that would otherwise remain hidden. Such information includes, for example, the kinematic and dynamic properties of objects in the environment, or views of the environment that can only be obtained after visual obstructions have been removed.

We employed the principle of interactive perception to show that a robot can easily extract the kinematic properties of novel objects from a visual sensor stream if it is able to physically interact with these objects. We have further demonstrated how the extracted knowledge about the object can be used to determine appropriate use of the object. Our experimental results on a real-world platform for mobile manipulation show that interactive perception results in highly robust and effective perceptual algorithms.

There are many possible directions for future research and extensions of the presented work. We plan to improve our feature tracking and contour detection algorithms by using active segmentation techniques [13], [23]. We also will generalize the types of kinematic properties that can be extracted. We would like to include other types of joints, such as prismatic or spherical joints, and joints that have joint axes with arbitrary orientations. Finally, we intend to extend our framework to support additional sensor modalities, such as force sensors and laser scanners.

VII. Acknowledgments

This work is supported by the National Science Foundation (NSF) under grants CNS-0454074, IIS-0545934, CNS-0552319 CNS-0647132, and by QNX Software Systems Ltd. in the form of a software grant. We are grateful for this support.

References

[1] Aloimonos J. and Weiss I. and Bandyopadhyay A. Active Vision. In *1st International Conference On Computer Vision*, pages 35–54, 1987.

[2] P. Azad, T. Asfour, and R. Dillmann. Toward an Unified Representation for Imitation of Human Motion on Humanoids. In *ICRA*, 2007.

[3] R. Bajcsy. Active perception. *IEEE Proceedings*, 76(8):996–1006, 1988.

[4] A. Blake and A. Yuille. *Active Vision*. The MIT Press, 1992.

[5] O. Brock, A. Fagg, R. Grupen, R. Platt, M. Rosenstein, and J. Sweeney. A Framework for Learning and Control in Intelligent Humanoid Robots. *International Journal of Humanoid Robotics*, 2(3):301–336, 2005.

[6] R. Brooks, L. Aryananda, A. Edsinger, P. Fitzpatrick, C. Kemp, U.-M. O'Reilly, E. Torres-Jara, P. Varshavskaya, and J. Weber. Sensing and manipulating built-for-human environments. *International Journal of Humanoid Robotics*, 1(1):1–28, 2004.

[7] A. D. Christiansen, M. Mason, and T. Mitchell. Learning reliable manipulation strategies without initial physical models. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, volume 2, pages 1224–1230, 1990.

[8] D. A. Cohn, Z. Ghahramani, and M. I. Jordan. Active learning with statistical methods. *Journal of Artificial Intelligence Research*, 4:129–145, 1996.

[9] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. MIT Press and McGraw-Hill, 2001.

[10] P. Deegan, B. Thibodeau, and R. Grupen. Designing a Self-Stabilizing Robot For Dynamic Mobile Manipulation. In *Robotics: Science and Systems - Workshop on Manipulation for Human Environments*, 2006.

[11] A. Edsinger. *Robot Manipulation in Human Environments*. PhD thesis, Massachusetts Institute of Technology, 2007.

[12] A. Edsinger and C. C. Kemp. Manipulation in Human Environments. In *IEEE/RSJ International Conference on Humanoid Robotics*, 2006.

[13] P. Fitzpatrick and G. Metta. Grounding vision through experimental manipulation. *Philosophical Transactions of the Royal Society: Mathematical, Physical, and Engineering Sciences*, 361(1811):2165–2185, 2003.

[14] D. A. Forsyth and J. Ponce. *Computer Vision: A Modern Approach*. Prentice Hall Professional Technical Reference, 2002.

[15] S. A. Hutchinson, G. D. Hager, and P. I. Corke. A tutorial on visual servo control. *IEEE Trans. Robotics and Automation*, 12(5):651–670, October 1996.

[16] Intel. http://www.intel.com/technology/computing/opencv/.

[17] D. Katz and O. Brock. Interactive Perception: Closing the Gap Between Action and Perception. In *ICRA 2007 Workshop: From features to actions - Unifying perspectives in computational and robot vision*, 2007.

[18] D. Katz, E. Horrell, Y. Yang, B. Burns, T. Buckley, A. Grishkan, V. Zhylkovskyy, O. Brock, and E. Learned-Miller. The UMass Mobile Manipulator UMan: An Experimental Platform for Autonomous Mobile Manipulation. In *Workshop on Manipulation in Human Environments at Robotics: Science and Systems*, 2006.

[19] O. Khatib, K. Yokoi, O. Brock, K.-S. Chang, and A. Casal. Robots in Human Environments: Basic Autonomous Capabilities. *International Journal of Robotics Research*, 18(7):684–696, 1999.

[20] J. J. Koederink and A. J. Van Doorn. Invariant properties of the motion parallax field due to the movement of rigid bodies relative to an observer. *Optica Acta.*, 22:773–791, Sept. 1975.

[21] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision (darpa). In *Proceedings of the 1981 DARPA Image Understanding Workshop*, pages 121–130, April 1981.

[22] S. Maybank. The angular velocity associated with the optical flowfield arising from motion through a rigid environment. In *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, volume 401, pages 317–326, Oct 1985.

[23] G. Metta and P. Fitzpatrick. Early integration of vision and manipulation. *Adaptive Behavior*, 11(2):109–128, 2003.

[24] E. S. Neo, T. Sakaguchi, K. Yokoi, Y. Kawai, and K. Maruyama. Operating Humanoid Robots in Human Environments. In *Workshop on Manipulation for Human Environments, Robotics: Science and Systems*, 2006.

[25] K. Nishiwaki, J. Kuffner, S. Kagami, M. Inaba, and H. Inoue. The experimental humanoid robot H7: a research platform for autonomous behaviour. *Philosophical Transactions of the Royal Society*, 365:79–108, 2007.

[26] A. Saxena, J. Driemeyer, J. Kearns, and A. Y. Ng. Robotic Grasping of Novel Objects. In *Neural Information Processing Systems*, 2006.

[27] A. Stoytchev. Behavior-grounded representation of tool affordances. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 3071–3076, 2005.

[28] A. M. Waxman and S. Ullman. Surface Structure and Three-Dimensional Motion from Image Flow Kinematics. *The International Journal of Robotics Research*, 4:72–94, 1985.

[29] T. Wimboeck, C. Ott, and G. Hirzinger. Impedance Behaviors for Two-Handed Manipulation: Design and Experiments. In *ICRA*, 2007.