Manipulating Articulated Objects With Interactive Perception

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

***Abstract*— Robust robotic manipulation and perception remains a difficult challenge, in particular in unstructured environments. To address this challenge, we propose to couple manipulation and perception. The robot observes its own deliberate interactions with the world. These interactions reveal sensory information that would otherwise remain hidden and also facilitate the interpretation of perceptual data. To demonstrate the effectiveness of *interactive perception* we present a skill for the manipulation of an articulated object. Using this skill, we show how *UMan*, our mobile manipulation platform, obtains a kinematic model of an unknown object. The model then enables the robot to perform purposeful manipulation of that object. Our algorithm is extremely robust, and does not require prior knowledge of the object; it is insensitive to lighting, texture, color, specularities, and is computationally highly efficient.**

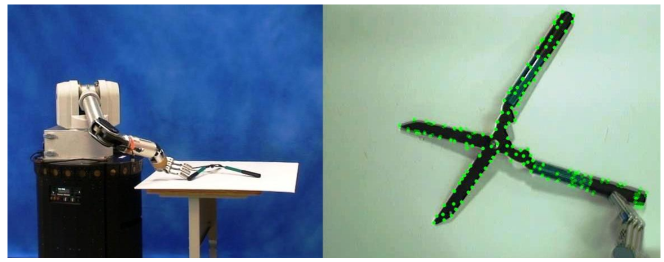


Fig. 1. The mobile manipulator UMan interacts with a tool, extracting the tool's kinematic model to enable purposeful manipulation. The right image shows the scene as seen by the robot through an overhead camera; dots mark tracked visual features.

I. Introduction

Already today mobile robots play an important role in applications ranging from planetary exploration to household robotics. Endowing these robots with significant manipulation capabilities could extend their use to many additional applications, including elder care, house-hold assistance, cooperative manufacturing, and supply chain logistics. But achieving the required level of competency in manipulation and perception has proven difficult. The dynamic and unstructured environments associated with those applications cannot be easily modeled or controlled. As a result, many existing perceptual and manipulation skills are too brittle to provide adequate manipulation capabilities.

We develop adequate capabilities for perception and manipulation in unstructured environments by closely coupling the perceptual process with the interactions of manipulation. The robot manipulates the environment specifically to assist the perceptual process. Perception, in turn, provides information necessary to manipulate successfully. The robot is watching itself manipulating the environment and interprets the sensor stream in the context of this deliberate interaction.

Coupling perception and manipulation has two main advantages: First, perception can be directed at those aspects of the sensor stream that are relevant in the context of a specific manipulation task. Second, physical interactions can reveal properties of the environment that would otherwise remain hidden to sensors. For example, by interacting with objects, their kinematic and dynamic properties become perceivable.

The proposed coupling of perception and manipulation naturally extends the concept of active vision [1], [2]. Active vision allows an observer to change its vantage point so as to obtain information most relevant to a specific task [6]. The work presented in this paper goes one step further: it allows the observer to manipulate the environment to obtain task-relevant information. Due to this coupling of perception and physical interaction, we refer to the general approach as *interactive perception*.

In interactive perception, the emphasis of perception shifts from object appearance to object function and the relationship between cause and effect. Irrespective of the color, texture, and shape of an object, a robot has to determine if this object is suited to accomplish a given task. Interactive perception thus represents a shift from the dominant paradigm in computer vision and manipulation. This shift is necessary to enable the robust execution of manipulation tasks in unstructured environments. Interestingly, there is evidence from psychology that humans use the functional affordances of an object for its categorization [3], [8].

In this paper, we develop interactive perception skills for the manipulation of unknown objects that possess inherent degrees of freedom (see Figure 1). This category of objects includes many tools, such as scissors, pliers, but also door handles, drawers, etc. To manipulate articulated objects successfully without an *a priori* model, the robot has to be able to acquire a model of the object's kinematic structure. The robot then has to be able to leverage this model for purposeful manipulation.

We demonstrate that interactive perception enables highly effective and robust manipulation of arbitrary, planar kinematic chains. Our main contribution is the interactive perception algorithm to extract kinematic models. This algorithm works reliably, does not require prior knowledge of the object, is insensitive to lighting, texture, color, specularities, and is computationally highly efficient. It is thus ideally suited as a perception and manipulation skill for unstructured environments.

II. Related Work

Interactive perception is related to an extensive body of work in robotics and computer vision; it is impossible to provide an inclusive review of that work. Instead, we will first differentiate interactive perception from entire areas of related work: feedback control, perception, and manipulation. We then discuss related research efforts that also leverage the concept of interactive perception.

Interactive perception is not feedback control. A controller affects a variable measured by a sensor to achieve a reference value. Instead, interactive perception affects the environment to enrich and disambiguate the sensor stream. The same argument differentiates interactive perception from visual servoing [14].

Interactive perception simplifies the perception problem by taking into account task constraints and by enriching the sensor stream through deliberate interaction with the environment. In contrast, most research in computer vision attempts to solve an unconstrained, passive perception problem from image data alone [12]. This is generally accomplished by either interpreting primitive image features, e.g. edges or corners, or by learning those features from many examples [4], [17], [26], [31]. Most of these methods cannot be directly employed for manipulation. Moreover, tasks such as extracting a kinematic model from images of a static object, are impossible to accomplish using these methods.

Interactive perception includes the acquisition of environmental models during manipulation. This stands in contrast with most research in manipulation [21], which addresses model acquisition in three different ways. The largest body of prior work assumes that accurate models are provided as part of the input [5], [20], [24], [27], [30]. This is not a viable option in unstructured environments. A second solution is teleoperation [10]. Here, the operator provides the cognitive capabilities to solve the model acquisition task. This solution is not feasible in autonomous manipulation. Lastly, some manipulation work explicitly includes model acquisition as part of manipulation [16], [25]. However, this work separates model acquisition from the manipulation itself and performs them sequentially. In this approach, the perceptual component is an independent component, therefore subject to the limitations discussed above.

Other prior work combines perception with deliberate physical actions [7], [13], [18], [28], [29]. To the best of our knowledge, there are only two examples of interactive perception in the literature. Christiansen, Mason, and Mitchell determine a model of an object's dynamics by observing its motion in response to deliberate interactions [9]. Fitzpatrick and Metta [11], [23] visually segment objects in the scene by pushing them. This research has motivated the work presented in this paper.

III. Obtaining Kinematic Models Through Interaction

Successful manipulation of articulated objects must be informed by knowledge of the object's kinematics. Such knowledge cannot be obtained from visual inspection alone and would be extremely difficult to determine by manipulation. However, when perception and manipulation are combined, the robot can generate visual stimuli that reveal this kinematic structure.

A. Obtaining Feature Trajectories

To extract the kinematic model of an object in the scene, we observe its motion caused by the robot's interaction. We identify the motion that occurs in the scene during this interaction by tracking point features in the entire scene. The resulting feature trajectories capture the movements of objects in the scene and allow us to infer a kinematic model of the scene.

We make no specific assumptions about the objects or the background. We simply use OpenCV's implementation [15] of optical flow-based tracking [19] to track the motion of f most distinctive features in the scene (in our experiments $f = 500$). Our algorithm requires that at least a few features are tracked on all rigid bodies throughout the interaction. As long as this requirement is satisfied, our algorithm is insensitive to lighting conditions, shadows, the texture and color of the object, and specularities. This requirement is very weak, since a scene without features does not provide substantial visual information.

Simple feature tracking in unstructured scenes is highly inaccurate. Features move relative to objects in the scene, disappear, or jump to other parts of the image. We will explain how our method of extracting kinematic models is inherently robust to this type of noise.

B. Identifying Rigid Bodies

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 does change as the bodies rotate and translate relative to each other (see Figure 2). Consequently, interacting with an object while observing changes in relative distance between points on the object will uncover clusters of points, where each cluster represents a different rigid body.

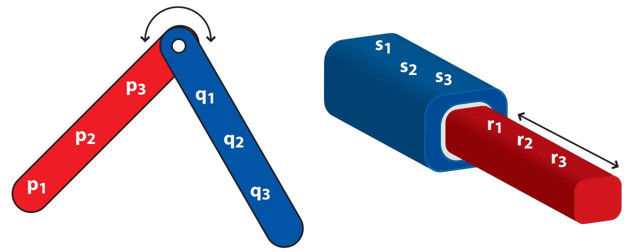


Fig. 2. Planar degrees of freedom: revolute (left) and prismatic (right). Points p_i, q_i, r_i, s_i on each rigid link do not change relative distances.

The first step of our algorithm serves to identify all rigid bodies observed in the scene. To achieve this, we build a graph $G(V, E)$ from the feature trajectories obtained throughout the interaction. Every vertex $v \in V$ in the graph represents a tracked image feature. An edge $e \in E$ connects vertices (v_i, v_j) if and only if the distance between the corresponding features remains smaller than some threshold throughout the observed interaction. Features on the same rigid body are expected to maintain approximately constant distance between them. In the resulting graph, all features on a single rigid body form a highly connected sub-graph. Identifying the highly connected sub-graphs is analogous to identifying the object's different rigid bodies.

In order to separate the graph into highly connected sub-graphs we use the min-cut algorithm, which separates a graph into two sub-graphs by removing as few edges as possible. Min-cut can be invoked recursively to handle graphs with more than two highly connected sub-graphs. The recursion terminates when breaking a graph into two sub-graphs requires removing more than half of its edges.

Our min-cut algorithm has worst case complexity of $O(nm)$, where n represents the number of nodes in the graph and m represents the number of clusters [22]. Most objects possess only few joints, making $m \ll n$. We can therefore conclude that for practical purposes clustering is linear in the number of tracked features.

This procedure of identifying rigid bodies is robust to the noise present in the feature trajectories. Unreliable features randomly change their relative distance to other features. This behavior places such features in small clusters, most often of size one. In our algorithm, we discard connected components with three of fewer features. The remaining connected components consist of features that were tracked reliably throughout the entire interaction. Each of these components corresponds to either the background or to a rigid body in the scene.

C. Identifying Joints

Two rigid bodies can be connected by either a revolute or a prismatic joint. Observing the relative motion that two rigid bodies undergo while interacting with them reveals the type of joint that connects them. Two bodies that are connected by a revolute joint share an axis of rotation. Two bodies that are connected by a prismatic joint can only translate with respect to one another. We examine all pairs of rigid bodies identified in the previous step of our algorithm to identify by which of the two joint types they are connected. If two rigid bodies experience relative motion that cannot be explained by either joint type, we infer that the bodies are not connected and belong to different articulated objects. This for example, is the case of the background, which we regard as another object.

For an object composed of n rigid bodies, there are $\binom{n}{2}$ pairs to analyze. In practice n is very small, as most objects possess only few joints.

To find revolute joints, we exploit the information captured in the graph G. Vertices that belong to two different connected components must have maintained constant distance to two distinct clusters. This property holds for features on or near revolute joints connecting two or more rigid bodies. To find all revolute joints present in the scene, we search the entire graph for vertices that belong to two clusters (see Figure 3).

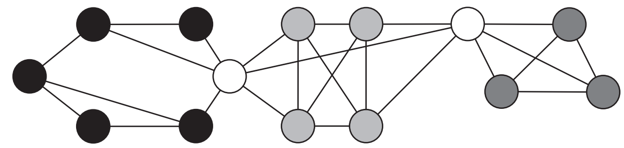


Fig. 3. Graph for an object with two revolute degrees of freedom. Highly-connected components (shades of gray) represent the links. Vertices of the graph that are part of two components represent revolute joints (white).

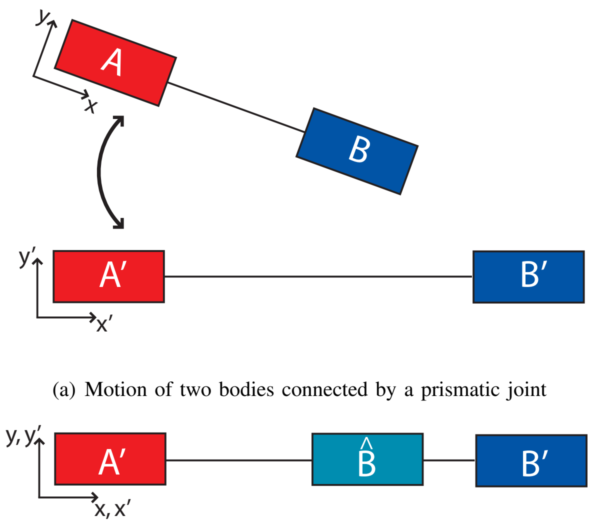
To determine if two rigid bodies are connected by a prismatic joint, we exploit distance information for features on those bodies from two different time instances. For every pair of bodies we examine the feature trajectories to determine when the two bodies experience their maximum relative displacement. Those instances will provide the strongest evidence and result in maximum robustness. We denote the corresponding positions of the two bodies by A, B and by A', B' at the beginning and the end of the motion, respectively (see Figure 4(a)). If the two bodies are connected by a prismatic joint, they may have rotated and translated together, as well as translated with respect to each other.

To confirm the presence of a prismatic joint, we compute the transformation $\mathbf{T}$ that maps features from A to A': $A' = \mathbf{T} \cdot A$. We then apply the same transformation to the second body to get its expected position, $\hat{B}$, at the second position (see Figure 4(b)). If $\hat{B} = B$ we have no evidence for a prismatic joint and in fact A and B at this point appear to be the same rigid body. If $\hat{B}$ is different than B's observed position and the displacement between B' and $\hat{B}$ is a pure translation, we conclude that the two bodies are connected by a prismatic joint. If neither prismatic nor revolute joint was detected, the two bodies must be disconnected.

After all pairs of rigid bodies represented in the graph have been considered, our algorithm has categorized their observed relative motions into prismatic joints and revolute joints, or it has determined that two bodies are not connected. The background, for example, always falls into the latter category. Using this information we build a kinematic model of the object using Denavit-Hartenberg parameters.

It is possible that two objects coincidentally perform a relative motion that would be indicative of a prismatic or revolute joint between them. Our algorithm would detect a joint between those bodies, as it has no evidence to the contrary. However, additional interactions with the objects will eventually provide evidence for removing such joints.

Note that this algorithm makes no assumptions about the kinematic structure of objects in the scene. It simply examines the relative motion of pairs of bodies. As a result, the algorithm naturally applies to planar serial chains as well as to planar branching mechanisms. Furthermore, our



(a) Motion of two bodies connected by a prismatic joint

(b) Predicted new pose $\hat{B}$ of body B

Fig. 4. Identification of prismatic joints: Based on the transformation between A and A', we anticipate B's new position $\hat{B}$. The translation between B' and $\hat{B}$ can be explained by a prismatic joint.

algorithm does not make any assumptions about the number of articulated or rigid bodies in the scene. As long as a sufficient number of features are tracked and as long as the interaction has resulted in evidence of the kinematic structure, our algorithm will identify the correct kinematic model for all objects.

D. Manipulating Articulated Bodies

Once the kinematic structure of an articulated body has been identified, manipulation planning becomes a trivial task. We define the manipulation task as moving the articulated body to a particular configuration q. Based on the kinematic model, we can determine the displacements of the joints required to achieve this configuration. During the interaction, we can track and update the kinematic model until the desired configuration is attained.

IV. Experimental Results

We validate the proposed method in real-world experiments. In those experiments, a robot interacts with various articulated objects to extract their kinematic structure. The resulting kinematic model is then used to perform purposeful manipulation.

The experiments were conducted with our robotic platform for autonomous mobile manipulation, called UMan (see Figure 5). UMan consists of a holonomic mobile base with three degrees of freedom, a seven degree-of-freedom Whole Arm Manipulator (WAM) by Barrett Technologies, and a three-fingered Barrett hand. The robot interacts with and manipulates articulated objects placed on a table in front of it (see Figure 1). The tabletop has a wood-like texture. In some of our experiments, we placed a white poster board on top of the table to provide a plain background. (We later determined that this has no effect.) An overhead off-the-shelf web camera with a resolution of 640 by 480 pixels provides a video stream of the scene. Note that the camera could have also been mounted directly on the robot, as long as it provides a good, roughly downwards view of the table. The camera mount is uncalibrated, but we ensure that the table was within the field of view. Experiments were performed right next to a window, thus lighting conditions vary significantly throughout our experiments.

UMan was tasked with extracting kinematic models of five objects: scissors, shears, pliers, a stapler, and a wooden toy, shown in Figures 6 and 7. The first four objects have a single degree of freedom (revolute joint). The wooden toy has three degrees of freedom (two revolute joints and one prismatic joint). The first four objects are off-the-shelf products and have not been modified for our experiments. They vary in scale, shape, color, and texture. For example, the scissors are much smaller than the shears, have different handles and different colors. The pliers have very long handles compared to the size of their teeth. And finally, the stapler's links do not extend to both sides of the joint, unlike the other three tools. The wooden toy was custom-made to have texture similar to that of the table, to test identification of prismatic joints, and to experiment with more complex kinematic chains.

In our experiments we tracked the 500 most distinctive features in the scene. During the interaction, which was performed using a pre-recorded motion, about half of these features are lost. Among the remaining ones, about half are very noisy and unreliable; they are discarded by our algorithm. Many of the lost and noisy features can be explained by the motion of the manipulator during the interaction. Shadows also result in unreliable features. Lost features are discarded before the graph representation is being constructed. Noisy features are automatically discarded by our method, as explained above.

Figure 6 shows the results of interacting with the objects that possess one revolute joint (scissors, shears, pliers, and stapler). The objects were placed on top of the white poster board. Experiments were repeated at least 30 times. In each experiment the objects were placed in different initial poses. In every single one of the experiments, we were able to identify the accurate kinematic structure of the object. The positions of the joints were detected with accuracy (revolute joints are marked by green dots).

The extracted kinematic model was then used to plan and execute purposeful interactions with the object. To demonstrate that, we tasked UMan with forming a 90° angle between the links of the four objects. This task simulates tool use, since using any of the four objects would require to achieve or alternate between specific configurations. The bottom two rows of Figure 6 show the results of executing a manipulation plan based on the detected kinematic structure for the four objects. In all of our experiments the results were very accurate, indicating that the extracted models can be applied for tool use.

Figure 7 shows the results of interacting with the wooden toy. The toy was placed on top of the table. In this experiment the texture of the background and the object were very similar. As a result, tracked features were distributed across the entire image. Also for this object we repeated experiments at least 30 times. In each experiment, the object was placed in a different initial pose. In all of our experiments, our algorithm identified the correct kinematic structure of the object, consisting of two revolute joints and one prismatic joint. The object was correctly separated from the background and no erroneous joints were detected. The positions of the joints were detected with accuracy (joints are marked by a green line and green dots).

During the experiments, our algorithm identified two objects. The first corresponds to the wooden toy and is composed of four links and three joints. The second is composed of one link and no joints; it corresponds to the background. This experiment required several interactions to generate perceptual evidence for each of the joints.

Fig. 5. UMan (UMass Mobile Manipulator)

In all of our experiments, the proposed algorithm was able to extract the kinematic model of the object with high accuracy. This robustness is achieved using a low-quality, low-resolution web camera. Small displacement of the object suffice for reliable joint detection. The algorithm does not require parameter tuning. The experiments were performed under uncontrolled lighting conditions, using different camera positions and orientations (for some experiments, the camera position and orientation varied throughout the interaction), and for different initial poses of the object. Our algorithm accurately recovered the kinematic structure of the object in every single one of our experiments.

The robustness and effectiveness of the proposed algorithm provides strong evidence that interactive perception can serve as a framework for manipulation and perception in unstructured environments. By combining very fundamental capabilities from perception and manipulation, we were able to create a robust skill that could not have been achieved by manipulation or vision alone. We are currently investigating additional applications of interactive perception, including object segmentation, object tracking, and grasping.

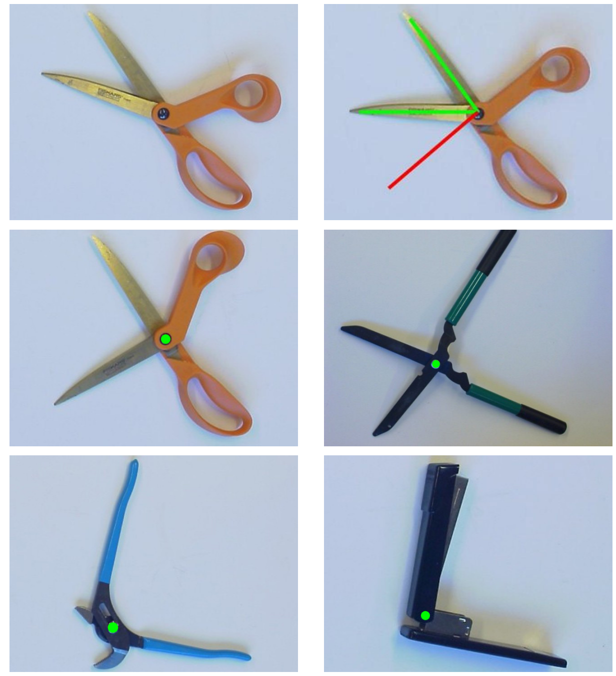


Fig. 6. Experimental results showing the manipulation of articulated objects based on interactive perception. The first row shows the scissors before the interaction (left) and after the interaction (right). The top right image also illustrates the generated manipulation plan for forming a 90° angle between the blades. The second and third row show the detected revolute joints (marked with a green dot), and the results of executing the respective manipulation plans for forming a 90° angle between the objects' links.

V. Conclusion

The deployment of robots with manipulation capabilities remains a substantial challenge, in particular in dynamic and unstructured environments. In such environments, robots cannot rely on accurate *a priori* models and are generally unable to acquire such models with high accuracy. As a result, the reliance on such models renders manipulation and perception skills brittle and unreliable in these environments, even if they work well under highly controlled conditions.

We demonstrate that it is possible to achieve robustness in perception and manipulation, even in unstructured environments, when perception and manipulation are coupled in a task-specific manner. By watching its own deliberate interaction with the world, a robot is able to improve perception by revealing information about the environment that would otherwise remain hidden. Such information includes, for example, the kinematic and dynamic properties of objects. Coupling manipulation and perception also enables the robot to interpret its sensor stream taking into account the specific task and the known, deliberate interaction. This significantly improves the robustness of perception, as the task and the interaction constrain the space of consistent interpretations of the perceptual data. We refer to this approach of coupling perception and manipulation as *interactive perception*.

In this paper, we describe an interactive perceptual algorithm to extract planar kinematic models from unknown objects by purposeful interaction. The robot pushes on an

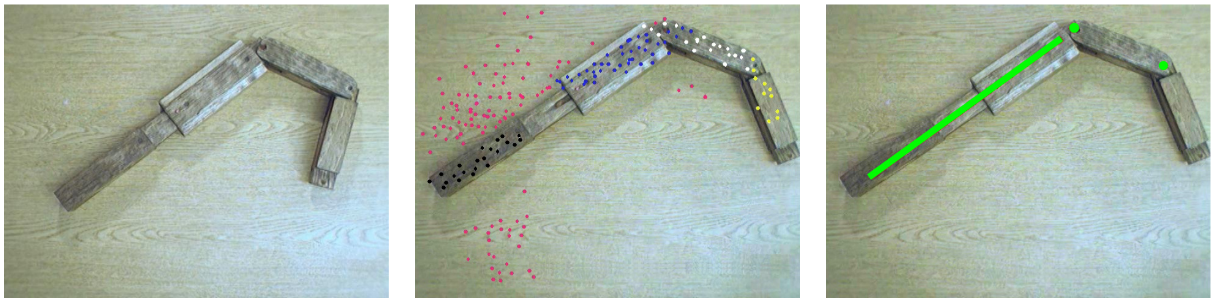


Fig. 7. Experimental results showing the extraction of the kinematic properties of a wooden toy using interactive perception. The left image shows the object in its initial pose. The middle image shows the object after the interaction. The detected clusters corresponding to rigid bodies are displayed. The right image shows the detected kinematic structure (green line marks the prismatic joint, green dots mark the revolute joints).

object in its field of view while observing the object's motion. From this observation, the robot can compute the object's kinematic model. This model is then used to perform purposeful manipulation, i.e. to achieve a specific configuration of the object. The algorithm for the extraction of kinematic models does not require prior knowledge of the object, is insensitive to lighting, texture, color, specularities, and is computationally highly efficient. It is thus ideally suited as a perception and manipulation skill for unstructured environments.

ACKNOWLEDGMENTS

We gratefully acknowledge support 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.

REFERENCES

[1] J. Aloimonos, I. Weiss, and A. Bandyopadhyay. Active vision. *International Journal of Computer Vision*, 1:333–356, 1988.

[2] R. Bajcsy. Active Perception. *IEEE Proceedings*, 76(8):996–1006, Aug. 1988.

[3] M. E. Barton and L. K. Komatsu. Defining features of natural kinds and artifacts. *Journal of Psycholinguistic Research*, 18(5):433–447, Sept. 1989.

[4] A. C. Berg, T. L. Berg, and J. Malik. Shape Matching and Object Recognition using Low Distortion Correspondence. In *Computer Vision and Pattern Recognition*, volume 1, pages 26–33, San Diego, CA, USA, June 2005.

[5] A. Bicchi and V. Kumar. Robotic grasping and contact: a review. In *International Conference on Robotics and Automation*, volume 1, pages 348–353, San Francisco, CA, USA, 2000.

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

[7] L. Bogoni and R. Bajcsy. Interactive recognition and representation of functionality. *Computer Vision and Image Understanding*, 62(2):194–214, Sept. 1995. Special issue of function-based vision.

[8] S. E. Chaigneau and L. W. Barsalou. The role of function in categories. *Theoria et Historia Scientiarum*, 2001.

[9] A. D. Christiansen, M. Mason, and T. Mitchell. Learning reliable manipulation strategies without initial physical models. In *International Conference on Robotics and Automation*, volume 2, pages 1224–1230, Cincinnati, Ohio, USA, May 1990.

[10] M. Diffler, F. Huber, C. Culbert, R. Ambrose, and W. Bluethmann. Human-robot control strategies for the NASA/DARPA Robonaut. In *IEEE Transactions on Aerospace and Electronic Systems*, volume 8, pages 939–3947, Mar. 2003.

[11] 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.

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

[13] K. Green, D. Eggert, L. Stark, and K. Bowyer. Generic Recognition of Articulated Objects Through Reasoning About Potential Function. *Computer Vision and Image Understanding*, 62(2):177–193, Sept. 1995. Special issue of function-based vision.

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

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

[16] M. Jägersand, O. Fuentes, and R. Nelson. Experimental Evaluation of Uncalibrated Visual Servoing for Precision Manipulation. In *International Conference on Robotics and Automation*, volume 4, pages 2874–2880, Albuquerque, NM, USA, Apr. 1997.

[17] V. Jain, A. Ferencz, and E. Learned-Miller. Discriminative training of hyper-feature models for object identification. In *Proceedings of the British Machine Vision Conference*, pages 357–366, Edinburgh, UK, 2006.

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

[19] B. D. Lucas and T. Kanade. An Iterative Image Registration Technique with an Application to Stereo Vision. In *Proc. Int. Joint Conference on Artificial Intelligence*, pages 674–679, Vancouver, BC, Canada, Aug. 1981.

[20] K. M. Lynch and M. T. Mason. Stable pushing: Mechanics, controllability, and planning. *International Journal of Robotics Research*, 15(6):533–556, Dec. 1996.

[21] M. T. Mason. *Mechanics of Robotic Manipulation*. MIT Press, 2001.

[22] D. W. Matula. Determining edge connectivity in O(mn). *Proceedings of the 28th Symp. on Foundations of Computer Science*, pages 249–251, 1987.

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

[24] A. Okamura, N. Smaby, and M. Cutkosky. An overview of dexterous manipulation. In *International Conference on Robotics and Automation*, volume 1, pages 255–262, San Francisco, CA, USA, Apr. 2000.

[25] R. Platt. *Learning and Generalizing Control Based Grasping and Manipulation Skills*. PhD thesis, University of Massachusetts Amherst, 2006.

[26] A. Saxena, J. Driemeyer, J. Kearns, and A. Y. Ng. Robotic Grasping of Novel Objects. In *Neural Information Processing Systems*, pages 1209–1216, Dec. 2006.

[27] S. Srinivasa, M. Erdmann, and M. Mason. Using projected dynamics to plan dynamic contact manipulation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3618–3623, Edmonton, Alberta, Canada, Aug. 2005.

[28] A. Stoytchev. Behavior-Grounded Representation of Tool Affordances. In *International Conference on Robotics and Automation*, pages 3071–3076, Barcelona, Spain, 2005.

[29] M. Sutton, L. Stark, and K. Bowyer. Function from visual analysis and physical interaction: a methodology for recognition of generic classes of objects. *Image and Vision Computing*, 16:746–763, 1998.

[30] J. C. Trinkle, R. C. Ram, A. O. Farahat, and P. F. Stiller. Dexterous Manipulation Planning and Execution of an Enveloped Slippery Workpiece. In *International Conference on Robotics and Automation*, pages 442–448, Atlanta, Georgia, USA, May 1993.

[31] P. Viola and M. J. Jones. Robust Real-Time Face Detection. *International Journal of Computer Vision*, 57(2):137–154, 2004.