Rigid and Non-Rigid Classification Using Interactive Perception

Bryan Willimon, Stan Birchfield, and Ian Walker
Department of Electrical and Computer Engineering
Clemson University, Clemson, SC 29634
{rwillim,stb}@clemson.edu, ianw@ces.clemson.edu

***Abstract*— Robotics research tends to focus upon either non-contact sensing or machine manipulation, but not both. This paper explores the benefits of combining the two by addressing the problem of classifying unknown objects, such as found in service robot applications. In the proposed approach, an object lies on a flat background, and the goal of the robot is to interact with and classify each object so that it can be studied further. The algorithm considers each object to be classified using color, shape, and flexibility. Experiments on a number of different objects demonstrate the ability of efficiently classifying and labeling each item through interaction.**

I. INTRODUCTION

Visual sensing and machine manipulation are well-studied topics within robotics research. Most of this effort, however, concentrates on only one topic or the other without considering the significant coupling of the two. To be sure, an important body of work has been aimed at using remote sensing to assist in real time with manipulation, e.g., visually-guided manipulation [1][2]. However, there has been relatively little work aimed at the reverse problem, namely, using manipulation to guide non-contact sensing in meaningful ways [3] [4].

Yet, humans routinely adopt this latter approach of "manipulation-guided sensing." For example, we routinely shuffle through papers on a desk or sift through objects in a drawer to more quickly and efficiently identify items of interest. In such cases, it is our interaction with the environment that increases our understanding of the surroundings, in order to more effectively guide our actions to achieve the desired goal. In a similar manner, animals such as raccoons [5] and cats use their front paws to poke, swat, and rummage to better understand their surroundings.

As a first step in addressing this problem, Katz and Brock [3] describe a system in which a manipulator learns about the environment by interacting with it. Video available from an overhead camera is analyzed by tracking feature points on an object in order to determine the number, location, and type (revolute or prismatic) of joints. In later work, Brock and colleagues [6] use video to locate and track objects. To describe this new approach toward autonomous manipulation, they introduce the term "interactive perception." Rather than solving action and perception separately, interactive perception (also known as manipulated-guided sensing) argues that both should be addressed simultaneously.

Inspired by the above work, this paper introduces a new approach to interactive perception, in which successive manipulations of objects in an environment are used to increase vision-based understanding of that environment, and vice versa (see Figure 1). We show that deliberate actions can alter the environment in a way that simplifies perception and consequently future interactions. Our work differs from that of Katz and Brock [3] in its purpose and scope. Our system is applicable to both rigid and non-rigid objects, and it produces a richer description of the object including a skeleton and appearance model, both of which are used to guide future interactions.

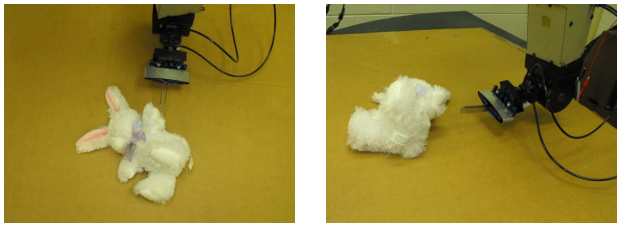

Fig. 1. The proposed setting for manipulation-guided sensing. The robotic system automatically learns the characteristics of an object by interacting with it. An overhead camera (not shown) is used for sensing the object.

Another piece of related work is that of Saxena et al. [7], in which information about a scene is gathered to generate a 3D model of each object in the scene which is then compared against a database of previously created models whose grasping locations have already been determined. Other work on grasping is presented in [8] [9]. Our method is different in that the objects being examined are unknown *a priori*.

Our work is also related to affordance learning [10] [11] [12]. In [10] [11], a robot learns the properties of an object (e.g., whether it rolls when tapped), as well as the association of properties (color, shape) and words spoken audibly by a trainer to their meaning. The work of [12] is similar in that it addresses the problem of learning about visual properties and spatial relations. Though related, our approach differs from these in that our goal is not to learn semantic associations with a tutor but rather to autonomously learn low-level properties for classification and manipulation.

II. APPROACH

A. Overview

The purpose of this work is to automatically learn the properties of an object for the purpose of classification and future manipulation. Figure 2 presents an overview of our classification process. First, the object is located in the image, and a color histogram model [13] is captured in order to model the object. Then, a 2D skeleton of the object

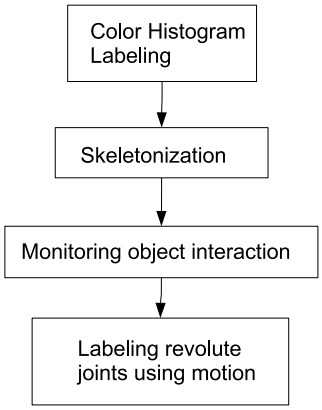

Fig. 2. Overview of our system for manipulation-guided classification of an unknown object.

is determined using a standard image-based skeletonization algorithm. The robotic arm then interacts with the object by prodding it from different directions. By monitoring the object's response to these movements, the revolute joints of the object are computed, as well as potential grasp points. We focus in this work on revolute joints because they are common in everyday situations (e.g., stapler, scissors, pliers, hedge trimmers, etc.) and because they more closely model the behavior of non-rigid objects containing stiffness (e.g., stuffed animals). Each of the boxes in the flowchart are now described in more detail.

B. Color histogram labeling

A color histogram is a representation of the distribution of the colors in a region of an image, derived by counting the number of pixels with a given set of color values [13]. Color histograms are chosen in this work because they are invariant to translation and rotation about the viewing axis, and for most objects they remain stable despite changes in viewing direction, scale, and 3D rotation. Objects are matched by comparing their color histograms with models of previously encountered objects using the technique of histogram intersection [13], which is conveniently affected by subtle differences in small areas of color while at the same time being guided by the dominant colors. We use eight bins for each (red, green, blue) color channel, leading to 512 total bins. The histogram intersection is normalized by the number of pixels in the region, leading to a value between 0 and 1 that can be interpreted as the probability of a match.

C. Skeletonization

Skeletonization is the process of determining the internal structure of a 2D image region. One way to describe a skeleton uses the analogy of a prairie fire: The boundary of the region is set on fire, and the skeleton is the loci of pixels where two or more fronts meet and quench each other [14]. The skeleton is therefore a single-pixel-wide

Fig. 3. LEFT: An isolated object to be classified. MIDDLE: The binary mask of the object. RIGHT: The image-based skeleton.

representation of the object's 2D shape. From the skeleton, it is possible to estimate candidate grasp points by noting the end points of the skeleton (where a branch terminates), while candidate revolute joints are given by intersection points of the skeleton (where two branches meet). It is widely known that the skeletonization process is extremely susceptible to noise in the image; therefore, an additional interactive step is necessary to refine these estimates. Figure 3 gives an example overhead image of a stuffed bunny on a table, along with its binary mask (obtained by thresholding) and skeleton.

D. Monitoring object interaction

To improve upon the noisy skeletonization model, the robot interacts with the object by repeatedly pushing it. The end effector is placed two inches away from an end point of the object, and the end effector is moved in the direction of the vertical or horizontal axis of the image plane (depending on the distance of the end point to the top and left image borders). As the robot interacts with the object, Kanade-Lucas-Tomasi (KLT) features [15] are tracked between successive image frames to monitor the scene changes that result from the object motion. These features are detected and tracked in the largest image region resulting from graph-based segmentation [16] that does not touch the image border. We have found it necessary to first dilate this region by one pixel to ensure that features along the boundary of the object are included. See Figure 4 for example features found on an object.

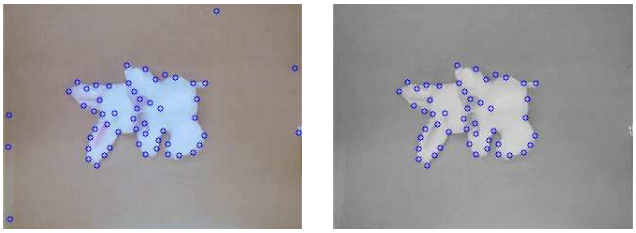

Fig. 4. LEFT: KLT features detected in the whole image. RIGHT: The subset of features that are located within the foreground region found by graph-based segmentation.

Tracked features are automatically clustered based on their Euclidean distance and motion vectors in the image plane. Features that are near each other and moving similarly are grouped together, while those that are far apart and/or moving differently are separated into distinct groups. The clustering algorithm is run every five frames to allow sufficient motion to accumulate. In contrast to the work of [3], in which small groups with three or fewer features

are discarded from the image, we have found that such groups are important when the object contains small regions, and some of the features have been lost. Therefore, in our approach all groups with at least two features are retained; while groups with a single feature point are attached to the nearest group using Euclidean image distance. Figure 5 illustrates the clustering of feature points.

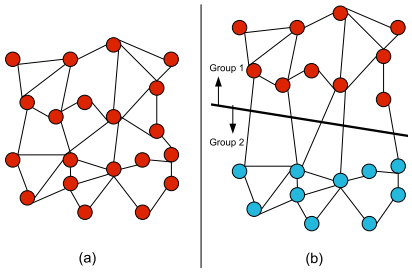

Fig. 5. Example of clustering feature points according to inter-distance values, in Euclidean space: (a) Before clustering and (b) after clustering with decision boundary.

E. Labeling revolute joints using motion

After the features in the region have been grouped, any group whose computed motion is greater than a prespecified threshold is determined to be movable and hence connected to the rest of the object via a revolute joint. The assumption is that the region with which the robot is interacting moves, while the other areas remain relatively stationary. In the case of a rigid, non-articulated object, of course there is just one region since the entire region moves together. The surrounding ellipse of the group is computed using principal component analysis (PCA) [17], and the revolute joint is considered to be the intersection point closest to the point of maximum curvature (along the major axis) of the ellipse toward the interior of the object. Figure 6 gives an example of the ellipse fitting.

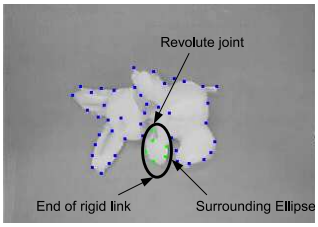

Fig. 6. Example of grouping feature points to locate revolute points near the endpoints of the major axis.

Figure 7 illustrates the initial skeleton labeled with intersection points and end points, along with the revised skeleton showing revolute joints labeled after several interactions with the robotic arm. In the revised skeleton, the end points that are considered noise in the skeleton are removed, where this determination is made based on whether the nearest intersection point (traversed along the skeleton) to the end point is a revolute joint. That is, the only branches in the skeleton that are considered extremities of the object (and therefore retained) are those whose intersection point is a revolute joint.

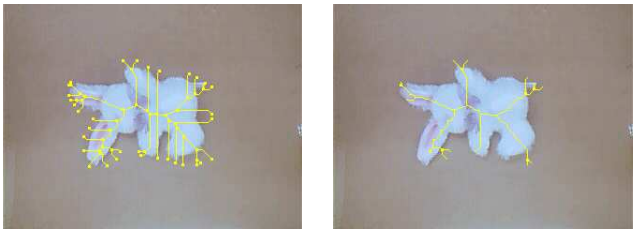

Fig. 7. LEFT: Original image with initial skeleton overlaid. RIGHT: Original image with revised skeleton overlaid after multiple interactions.

III. EXPERIMENTAL RESULTS

The proposed approach was applied in a number of different scenarios to test its ability to perform practical interactive perception. A PUMA 500 robotic arm was used to interact with the objects, which rested upon a flat table with uniform appearance. The objects themselves and their type were unknown to the system. The entire system, from image input to manipulation to classification, is automatic.

A. Articulated rigid object

In [3], revolute and prismatic joints on a rigid object were categorized using a similar technique of grouping feature points within a video sequence. One scenario shown is that of determining the revolute joint of a pair of hedge clippers. To demonstrate that our approach can calculate similar information, Figure 8 presents the result of our algorithm on a pair of pliers, along with the steps taken by the algorithm. For comparison, the result of [3] on the pair of hedge clippers is shown in Figure 9.

B. Classification experiment

We conducted an experiment with a set of eight unknown non-rigid objects to demonstrate the classification process and the possible uses of labeling individual objects for further learning. The system captured an image of each isolated object, from which the color histogram and final skeleton were computed. The images and skeletons are shown in Figure 10.

After the database of histograms and skeletons was built, the objects were randomly rearranged in a new order to test the classification performance of the system. The probability that the test and training objects were the same was computing using the color histogram, the number of revolute joints, and the number of extremities. Two versions of the algorithm were compared, one using only information available from vision, the other using information from both vision and the final skeleton resulting from interaction. Figure 11 shows the images gathered in the second run along with the best matching image from the first run. These results demonstrate that the color histogram and skeleton are fairly robust to orientation and non-rigid deformations of the objects.

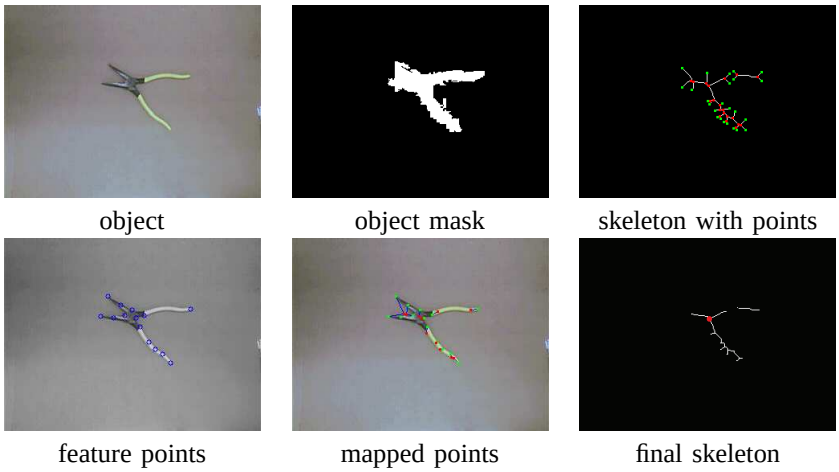

Fig. 8. Example of our approach on a pair of pliers. In lexicographic order: The original image of the object, the binary mask of the object, the skeleton with the intersection points (red dots) and end points (green dots) labeled, the feature points gathered from the object, the image after mapping the feature points to the intersection points, and the final skeleton with the revolute joint (red point) automatically labeled. The red dots represent the intersection points (possible revolute joints) of the skeleton. The green dots represent the end points (interaction points) of the skeleton.

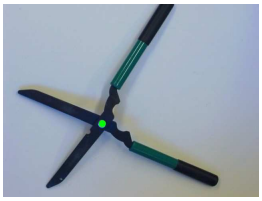

Fig. 9. Results of [3] on a pair of hedge clippers, with the green dot representing the revolute joint.

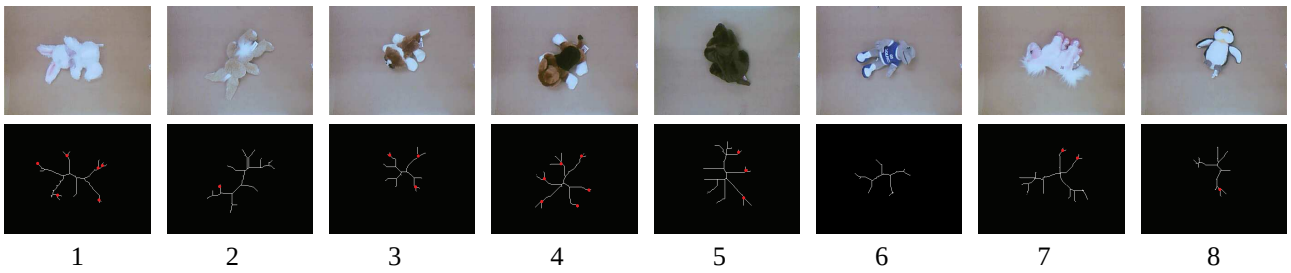

Fig. 10. TOP: Images of the individual objects used for creating a database of previously encountered items. BOTTOM: The final skeletons of the objects with revolute joints automatically labeled (red dots).

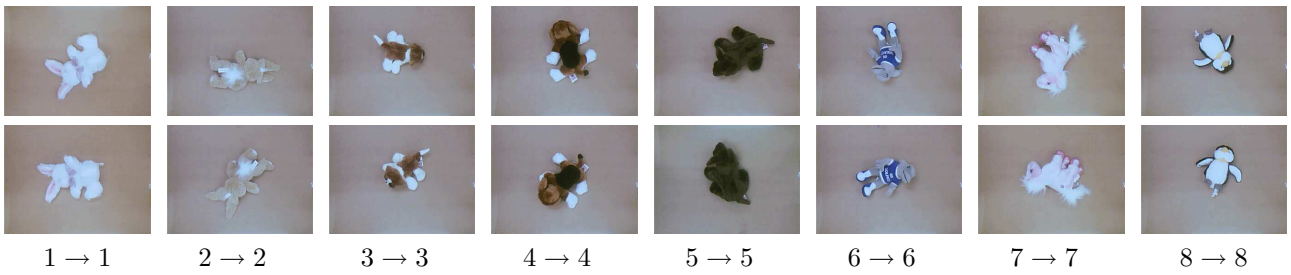

Fig. 11. Results from matching query images obtained during a second run of the system (top) with database images gathered during the first run (bottom). The numbers indicate the ground truth identity of the object and the matched identity. All of the matches are correct.

Tables I and II display the comparison matrix indicating the probability of each query image matching each database image using vision only and using vision plus interaction, respectively. The higher the value, the more likely the two images match. Bold is used to indicate, for each query image, the database image that contains the highest match value. Note that Item #1 is correctly classified only when interaction information is used.

TABLE I

EVALUATING PROBABILITIES OF STUFFED ANIMALS USING VISION ONLY: THE ROWS REPRESENT QUERY IMAGES AND THE COLUMNS REPRESENT DATABASE IMAGES.

#	1	2	3	4	5	6	7	8
1	0.84	0.27	0.25	0.22	0.15	0.27	**0.92**	0.18
2	0.33	**0.81**	0.35	0.39	0.26	0.38	0.46	0.33
3	0.54	0.50	**0.70**	0.67	0.45	0.40	0.56	0.36
4	0.41	0.45	0.60	**0.88**	0.56	0.41	0.39	0.48
5	0.19	0.19	0.25	0.41	**0.90**	0.20	0.15	0.42
6	0.39	0.51	0.32	0.33	0.27	**0.69**	0.47	0.35
7	0.78	0.36	0.28	0.24	0.16	0.33	**0.97**	0.25
8	0.29	0.33	0.37	0.61	0.51	0.33	0.24	**0.83**

TABLE II

EVALUATING PROBABILITIES OF STUFFED ANIMALS USING VISION AND THE SKELETON: THE ROWS REPRESENT QUERY IMAGES AND THE COLUMNS REPRESENT DATABASE IMAGES.

#	1	2	3	4	5	6	7	8
1	**0.78**	0.76	0.55	0.54	0.48	0.62	0.77	0.53
2	0.68	**0.80**	0.72	0.66	0.65	0.73	0.75	0.71
3	0.75	0.70	**0.83**	0.76	0.72	0.73	0.79	0.72
4	0.67	0.65	0.77	**0.79**	0.78	0.70	0.76	0.73
5	0.56	0.53	0.68	0.60	**0.93**	0.60	0.65	0.67
6	0.60	0.60	0.61	0.54	0.62	**0.73**	0.72	0.62
7	0.69	0.52	0.63	0.48	0.69	0.58	**0.92**	0.55
8	0.66	0.51	0.72	0.80	0.67	0.64	0.61	**0.88**

C. Sorting using socks and shoes

Another practical scenario of interactive sensing is that of sorting socks in a pile of laundry, or organizing shoes by pairing them. We used typical socks and shoes of different colors and sizes for this experiment, for which the results are shown in Figures 12 and 13.

The comparison matrix is shown in Tables III and IV, indicating the probability of each query image matching each database image using vision only and vision plus interaction, respectively. Again, interaction is necessary to correctly classify all the objects (in this case Item #5).

TABLE III

EVALUATING PROBABILITIES OF SOCKS AND SHOES USING VISION ONLY: THE ROWS REPRESENT QUERY IMAGES AND THE COLUMNS REPRESENT DATABASE IMAGES.

#	1	2	3	4	5
1	**0.87**	0.69	0.26	0.29	0.16
2	0.62	**0.90**	0.24	0.38	0.18
3	0.29	0.25	**0.86**	0.20	0.12
4	0.25	0.26	0.17	**0.93**	0.38
5	0.26	0.24	0.12	**0.99**	0.56

TABLE IV

EVALUATING PROBABILITIES OF SOCKS AND SHOES USING VISION AND THE SKELETON: THE ROWS REPRESENT QUERY IMAGES AND THE COLUMNS REPRESENT DATABASE IMAGES.

#	1	2	3	4	5
1	**0.82**	0.43	0.69	0.63	0.59
2	0.47	**0.77**	0.41	0.39	0.53
3	0.50	0.42	**0.75**	0.47	0.64
4	0.58	0.32	0.62	**0.81**	0.56
5	0.49	0.41	0.51	0.73	**0.79**

IV. CONCLUSION

We have proposed an approach to interactive perception in which an autonomous robot system is able to classify and label an unknown object. The proposed approach has been found to be effective over a wide range of environmental conditions. Monitoring the interaction of the object builds upon the approach in [3] to group different feature points together that share similar characteristics. Like [3], the approach is also able to determine the locations of revolute joints for planar rigid objects, but it is also applicable to non-rigid objects.

The proposed approach only begins to address the challenging long-term problem of interactive perception. Other avenues can be explored regarding improving the classification algorithm and learning strategy. When looking for a target item, one must consider the orientation of the object along with the angle from which it is viewed. Additional interaction and labeling techniques could be used to improve the ability of the system to determine which characteristics of an object make it distinguishable from other objects.

Currently, the system only allows interactions from two directions. Using the camera as a mode of reference, the robot is able to interact with the top part and the left part of the object in the classification images, but the and bottom parts of the object are out of reach from the robotic arm, because it would occlude the object from the camera's view if it tried to interact with these other parts of the object. One possible solution to this problem would be to place the isolated objects on a turntable so that the robot would be able to interact with all directions of the object without occluding any part of the camera's viewing area.

Another improvement in the modeling of the object would be to incorporate a 3D model instead of a 2D model. The 3D model would provide a more accurate representation of how each revolute joint moves and give a more detailed skeleton that describes the overall shape of the object. In the case of giving the system a round single colored ball, after viewing and interacting with the ball, the cameras would only see a circle that does not roll, in a 2D world. The system would disregard information vital to discovering the dynamics of each object if the object did something in the 3D world and looks like another in the 2D world, just like the ball scenario. We believe these are all fruitful areas for future extensions of our research.

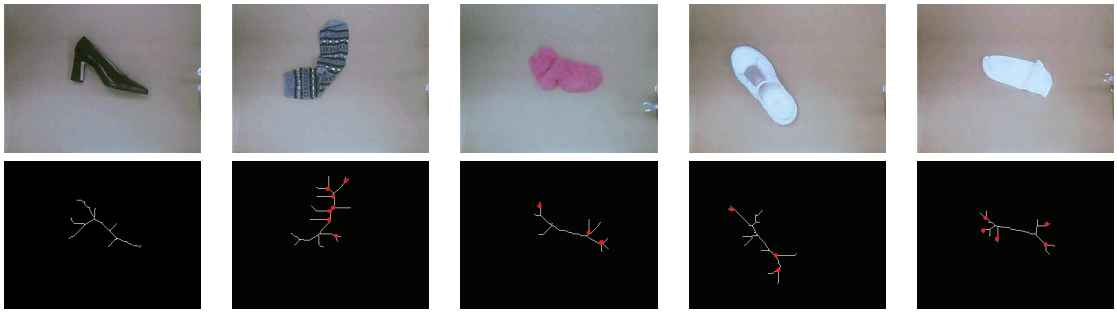

Fig. 12. TOP: Images of the individual objects gathered automatically by the system for the purpose of creating a database of objects previously encountered. BOTTOM: The final skeletons with revolute joints labeled.

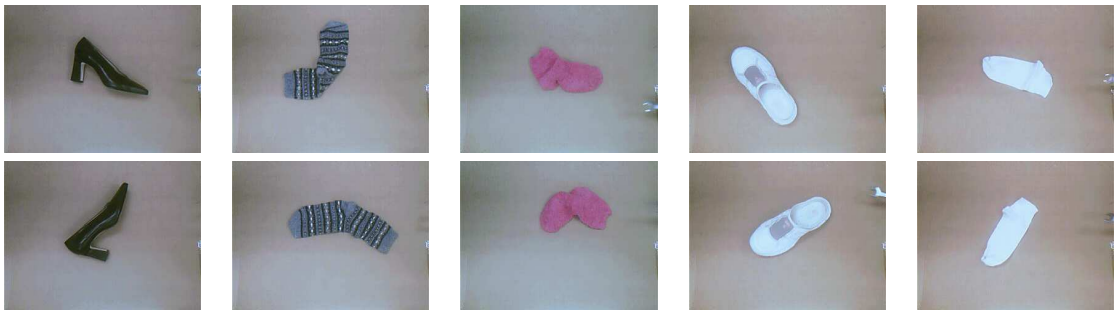

Fig. 13. Results from matching query images obtained during a second run of the system (top) with database images gathered during the first run (bottom) for the sorting experiment. There is one mistake.

V. ACKNOWLEDGMENTS

This research was supported by the U.S. National Science Foundation under grants IIS-0844954 and IIS-0904116.

REFERENCES

[1] F. Chaumette and S. Hutchinson, "Visual servoing and visual tracking," in *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. Springer, 2008, pp. 563–584.

[2] D. Kragic, M. Björkman, H. I. Christensen, and J.-O. Eklundh, "Vision for robotic object manipulation in domestic settings," *Robotics and Autonomous Systems*, vol. 52, no. 1, pp. 85–100, Jul. 2005.

[3] D. Katz and O. Brock, "Manipulating articulated objects with interactive perception," in *Proceedings of the International Conference on Robotics and Automation*, May 2008, pp. 272–277.

[4] P. Fitzpatrick, "First contact: An active vision approach to segmentation," in *International Conference on Intelligent Robots and Systems (IROS)*, 2003.

[5] I. Walker, "A successful multifingered hand design — The case of the raccoon," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Aug. 1995, pp. 186–193.

[6] J. Kenney, T. Buckley, and O. Brock, "Interactive segmentation for manipulation in unstructured environments," in *International Conference on Robotics and Automation (ICRA)*, 2009, pp. 1377–1382.

[7] A. Saxena, J. Driemeyer, and A. Y. Ng, "Robotic grasping of novel objects using vision," *International Journal of Robotics Research*, vol. 27, pp. 157–173, Feb. 2008.

[8] A. Bicchi, "Hands for dexterous manipulation and robust grasping: A difficult road toward simplicity," *IEEE Transactions on Robotics and Automation*, vol. 16, no. 6, pp. 652–662, 2000.

[9] P. Gibbons, P. Culverhouse, and G. Bugmann, "Visual identification of grasp locations on clothing for a personal robot," in *Towards Autonomous Robotic Systems (TAROS)*, Aug. 2009, pp. 78–81.

[10] V. Krunic, G. Salvi, A. Bernardino, L. Montesano, and J. Santos-Victor, "Affordance based word-to-meaning association," in *International Conference on Robotics and Automation (ICRA)*, 2009, pp. 4138–4143.

[11] L. Montesano, M. Lopes, A. Bernardino, and J. Santos-Victor, "Learning object affordances: From sensory-motor coordination to imitation," *IEEE Transactions on Robotics*, vol. 24, no. 1, pp. 15–26, 2007.

[12] D. Skočaj, G. Berginc, B. Ridge, A. Štimec, M. Jogan, O. Vanek, A. Leonardis, M. Hutter, and N. Hawes, "A system for continuous learning of visual concepts," in *International Conference on Computer Vision Systems*, 2007.

[13] M. Swain and D. Ballard, "Color indexing," *International Journal of Computer Vision*, vol. 7, no. 1, pp. 11–32, 1991.

[14] G. Bertrand, "A parallel thinning algorithm for medial surfaces," *Pattern Recognition Letters*, vol. 16, no. 9, pp. 979–986, 1995.

[15] C. Tomasi and T. Kanade, "Detection and tracking of point features," Carnegie Mellon University, Tech. Rep. CMU-CS-91-132, Apr. 1991.

[16] P. Felzenszwalb and D. Huttenlocher, "Efficient graph-based image segmentation," *International Journal of Computer Vision*, vol. 59, no. 2, pp. 167–181, 2004.

[17] I. Jolliffe, *Principal Component Analysis*. Springer, 1986.