The 2009 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 11-15, 2009 St. Louis, USA

Cue-Based Equivalence Classes and Incremental Discrimination for Multi-Cue Recognition of “Interactionable” Objects

Sarah Aboutalib* and Manuela Veloso**

Abstract— **There is a subset of objects for which interaction can provide numerous cues to those objects’ identity. Robots are often in situations where they can take advantage being able to observe humans interacting with the objects. In this paper, we define this subset of ‘interactionable’ objects for which we use our Multiple-Cue Object Recognition algorithm (MCOR) to take advantage of using multiple cues. We present two main contributions: 1) the introduction of cue-driven equivalence class discrimination, and 2) the integration of this technique, the general MCOR algorithm, and a hierarchical activity recognition algorithm also presented in this paper, demonstrated on data taken from a static Sony QRIO robot observing a human interacting with objects. The hierarchical activity recognition provides an important cue for the object recognition.**

I. Introduction

Because of the complexity of real world data, providing robots with the ability to recognize objects has proven an exceedingly difficult task. The great variation both in the appearance of objects of the same class and in the appearance of the same object under various conditions combine to produce that difficulty.

In previous work, we have shown that visually similar objects can be disambiguated through the integration of information obtained from cues of various types producing a Multiple-Cue Object Recognition (MCOR) algorithm[2]. In this work, we first define a subset of objects which can be interacted with, which we term “interactionable” objects, since this is the set of objects our multiple-cue algorithm most benefits, and is a definition which many other techniques (such as Functional Recognition) can utilize.

In light of this definition, this paper makes two main contributions:

First, the introduction of cue-driven discrimination where objects in the same equivalence classes given a cue, can be distinguished. While in previous work we showed that objects visually similar can be distinguished by another cue such as activities and speech, we now present a method that allows any set of objects similar to each other given any cue to be incrementally divided through the addition of various cue types, until each set contains only one object, and thus can be recognized apart from the others. This reduces the number of comparisons necessary when calculating the evidence for different object labels and allows it to take advantage of the sequential nature of data received from a robot.

Second, the integration of this technique, the general MCOR algorithm, and an activity recognition algorithm outlined below to recognize objects in video data. When dealing with robot data and many types of video data, it is important to be able to obtain cue information in a speedy manner, we provide a hierarchical activity recognition algorithm that can provide fast and simple activity recognition for key activities. We then provide results demonstrating these concepts on data taken from a Sony QRIO robot.

Overall, this paper demonstrates the benefit of Multiple-Cue Object Recognition and cue-equivalence classes in obtaining object recognition results for interactionable objects from video in unrestricted real-world environments using simple, fast-to-use cues, when typically more complicated and extensively trained cues would be necessary.

II. Related Work

A number of efforts address object recognition in a robot [4], [8], [9], [6], [10], using techniques focused on very specific visual cues or explicit gestures from the human [8]. Although effective for their specified tasks, none provide a general framework for utilizing cues across various domains.

The weakness of depending on visual cues in terms of lighting, pose, rotation, and other factors is a well-known problem. Numerous techniques have sought to address this problem [13], [3], [19], [16], [11], [5] through the use of more descriptive and invariant features. Although fast and accurate results have been demonstrated by these techniques, the dependence of these approaches on visual cues alone requires the use of large training sets to address the issue of variablity within even a single object category, which can not be avoided in real world data.

Other approaches have attempted to compensate for the weaknesses of visual cues by including another type of information such as context[15] and activity cues [14], [18].

Encouraged by the general success of these approaches in integrating a non-visual cue for more robust object recognition, MCOR [2] provides a general framework for flexibly including multiple cues of any number and any type, so that all cues, whether activity, visual, context, or any other possible cues available now or in the future can be used to provide evidence for the presence of an object.

III. Interactionable Objects

There has been some inquiry as to when multiple-cue object recognition can be best used, i.e., to what set of

*S. Aboutalib is a PhD candidate in the Computer Science Department, School of Computer Science, Carnegie Mellon University, PA 15213, USA saboutal@cs.cmu.edu

**M. Veloso is the Herbert A. Simon Professor in the Computer Science Department, School of Computer Science,Carnegie Mellon University, PA 15213, USA veloso@cmu.edu

978-1-4244-3804-4/09/$25.00 ©2009 IEEE

objects can this recognition algorithm best be applied. Since Multiple-Cue Object Recognition can utilize information obtained through the observation of interactions with an object by a human or any other agent, the set of objects which can obtain the greatest benefit from this algorithm for recognition are those which can be interacted with or can interact. We will now call this set of objects, 'interactionable' objects for lack of an equivalent term currently existing.

For instance, the MCOR algorithm would not be best used on a Mars Rover, where the video data often contains objects (such as various rocks) which are not or cannot be interacted with. Similarly, a video of a tree or a mountain standing static would not be very useful to the MCOR algorithm. In addition, the determination of whether an object is interactionable or not is dataset-specific: Some objects may be interactionable in one dataset and non-interactionable in another depending on the set of interactions possible for each dataset. Interaction in this definition means speaking about objects, having sounds coming from the objects, being acted upon, acting as well as another possible manipulation done to or by the objects.

It should be noted however, that although our method has a special advantage for these sets of interactionable objects, it is possible to use it on the non-interactionable objects such as the tree in the video above, if it is given visual data ahead of time. It however, would not be *best* used in that case. MCOR would only be as good as the visual cues it was given and thus would be on par with other visual recognition system using those same cues. MCOR gains its advantage when used with the set of objects outlined above. The interactionable term may also be useful for other methods that depend on interaction to aid in recognition such as in Functional Recognition [7], [17].

IV. Cue-Based Equivalence Classes

In previous work, we presented a multiple-cue object recognition algorithm (MCOR)[2]. In this section, we introduce a new concept that can be applied to the algorithm in order to reduce some of the work and produce results more efficiently. Although there have been other decision-tree methods for object recognition [21], [12], our method deals with the handling of multiple cues and the introduction of these different cues for more efficient reduction of the set of possible object classes for a particular region.

A. Equivalence

An equivalence class is defined as the set of all objects that contain the same property given a specific cue. In typical object recognition, the cues are usually based on visual cues, such as color or texture. These, however, often produce classes that will encompass more than one type of object, and thus errors in recognition will occur. The goal then is to continually separate the objects into classes of smaller and smaller size until the remaining set has cardinality of one.

In terms of the algorithm, we define an equivalence class, eq_k, as the set of object labels that are applicable to the k^{th} region we would like to recognize.

An object label is determined to be applicable, if it satisfies the condition that the evidence provided by an extracted cue from the data is *very close* to the evidence given by the object label with the greatest amount of evidence, i.e., only object labels close to the highest evidence value will be placed in that equivalence class. This is where "equivalence" is determined, as given by the equation:

$$eq_k \leftarrow \left\{ x | e_{k,x} - \max_i e_{k,i} < \varepsilon \right\}$$

An equivalence class for the k^{th} region, eq_k is the set of labels x whose evidence is "close", i.e., a small given ε value away from the largest evidence value, $\max_i e_{k,i}$.

The evidence is calculated by:

$$e_{k,i} = w_{i,l} s_{j,l}$$

The evidence $e_{k,i}$ that region k is object i is given by the product of $w_{i,l}$, the weight between object i and cue l, and $s_{j,l}$, the similarity between the extracted cue j and the corresponding cue l in the dictionary (A similarity function is given as one of the properties of the cues [2]). Calculation of the weight and similarity is outlined in [2], [1].

Thus, the equivalence class for a particular region, eq_k, will consist of all the object labels which are applicable to that region given the cues seen thus far by the robot.

B. Incremental Discrimination

We now must determine how the classes can be divided. In our algorithm, we introduce the idea of separation through the addition of more cue types. In this process, the algorithm begins with the equivalence classes outlined above where every region initially starts with all possible object labels, these labels are then divided with the introduction of a cue type. Objects with the same property are then pooled together, this process continues until each object is in its own equivalence class or until all cues are utilized. See figure 3 for the pseudocode of the equivalence classes and this incremental discrimination of the object labels, and figures 1 and 2 for examples.

C. MCOR and Equivalence Classes

The introduction of the concept of cue-based equivalence classes cuts down on comparisons MCOR would have done, and allows it to take advantage of the sequential introduction of evidence characteristic of video data such as that taken from a robot. In the following section, we give a brief review of the MCOR algorithm.

1) MCOR Algorithm: The MCOR algorithm is based on the concept of utilizing any potential evidence that can lead to the identity of an object. Thus, it provides a flexible framework allowing the use of any number of cues and of any type [2].

The MCOR algorithm begins by extracting all possible cue information, c_j. It then segments the region, r_k, associated with that cue if it has not already been segmented. Such regions become a possible object candidates.

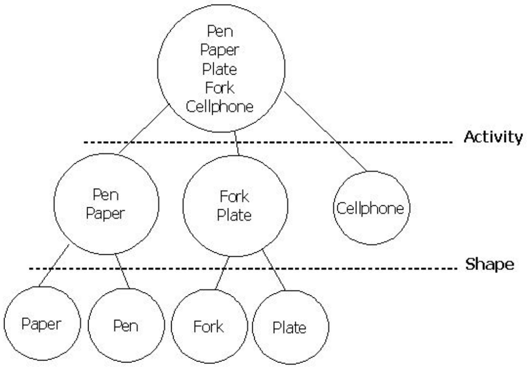


Fig. 1. Model of an Equivalence class separation starting with the application of the activity cue. In the first level, we have a circle containing the set of all objects in that scenario. The dotted line represents the application of a cue to the set; in the first case, the activity cue. The original set is then divided into three smaller sets containing objects that are similar given that cue. For instance, pen and paper are grouped together because they are both associated with the "writing" activity. The application of another cue, shape, represented by the second dotted line, further divides the sets based on those with similar shapes. Since this separates all the objects into their own individual sets allowing them to be distinguished completely from one another for identification, the algorithm stops.

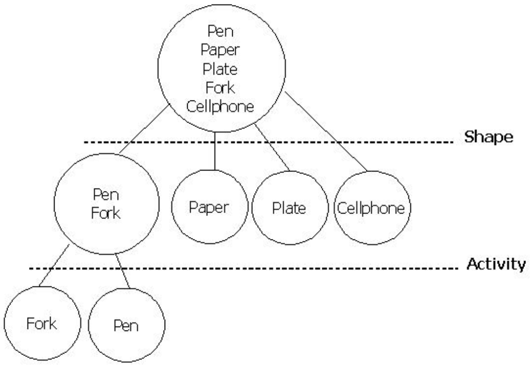


Fig. 2. Model of an Equivalence class separation starting with the application of the shape cue, contrasting with figure 1

An object dictionary, containing all the cues l associated with each particular object and their weights, $w_{i,l}$, (i.e. the strength of the association, and is learned using a Probabilistic Relational Model [1]) is given. The evidence, $e_{k,i}$, that the region, r_k, belongs to a particular object class, i is then calculated as:

$$e_{k,i} = \sum_{l \in C_i} \sum_{j \in C_k} w_{i,l} s_{j,l}$$

It is at this point where the equivalence classes will cause some change. The next section describes this in more detail. The objects are then recognized as the object class with the greatest evidence, if it is above a threshold, θ, i.e.,

$$label_k \leftarrow \text{argmax}_i \, e_{k,i}, \text{ if } \max e_{k,i} > \theta$$

Once an object is recognized, all the cues not previously associated with that object class in the object dictionary get added to its definition. In this way, new cues can be added to an object's definition in the dictionary. See figure 3 for the pseudocode of the implementation of equivalence classes in MCOR.

2) Inclusion of Equivalence Classes: With the utilization of the equivalence classes, we can reduce the number of comparisons made when determining the probability that a particular object belongs to a particular class. This reduction occurs in two primary ways:

1: Taking advantage of the temporal aspect of real robot data, we no longer need to calculate the evidence from scratch at each frame. Instead, the set of object labels applicable to each region is adjusted incrementally at each time step, based on the cue evidence given at that time.

2: Instead of comparing every object with every cue in the object definition, it is only necessary to look at cues with the equivalence classes, until there is only one object label that fits that region, once that point is reached it is unnecessary to look at any more cues.

```
For each region $k$, there is a set $eq_k$ for all object labels
applicable to that region, i.e. its equivalence class

• $\forall eq_k$ where $|eq_k| > 1$ :
    – Get next cue $l$ in video from robot
    – for each object label, $i$ in $eq_k$:
        ∗ $e_{k,i} = w_{i,l} s_{j,l}$.
        ∗ the evidence that region $k$ is object $i$, where $w_{i,l}$ is the weight between
          object $i$ and cue $l$, and $s_{j,l}$ is the similarity between the extracted cue
          $j$ and the cue $l$ in the dictionary
            · $eq_k \leftarrow \{x | e_{k,x} - \max_i e_{k,i} < \varepsilon\}$
```

Fig. 3. **Pseudocode for incremental discrimination of the objects in the equivalence classes**

V. Activity Recognition

In order to get the MCOR algorithm to work, it is necessary to have cues that can provide useful information. One of the most important cues is activity recognition information. In order to make the algorithm practical for our real-time video data, we created an efficient activity recognizer.

The standard approach to activity recognition is an HMM based on observations such as the change in the x and y position of a particular tracking point. There are several advanced activity recognition systems, but since our focus is primarily on the use of the cue, rather than developing an intricate cue recognition system, we have developed an activity recognition algorithm that can produce quick and simple results for some key activities.

We based the algorithm on HMMs and Viterbi algorithm, but doing so through a hierarchical structure. Below we first describe this model, then show how it is used for recognition.

A. Hierarchical Model

Many of the activities we are interested in tend to be higher-level activities, which we initially attempted to recognize using a single tier recognition system where large activities movements were attempted to be recognized by classifying the change in the x and y movement of the centroid of the region being tracked. In our case, the region is a bright pink wristband worn by the subject.

This however presented a problem, since for such activities as eating, there would be a large distribution in the change of activities. In order to solve this problem, we divided the activities into smaller movements whose distributions are more easily separable (see figure 4) and which will then be used as observations for the larger activities.

1) Small Movements: The smaller movements consisted of pickup, put down, move left, move right, and stay. These movements distributions can be easily separated and thus more easily recognized.

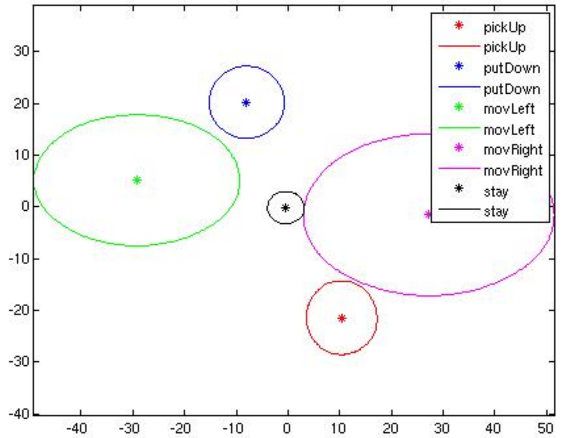
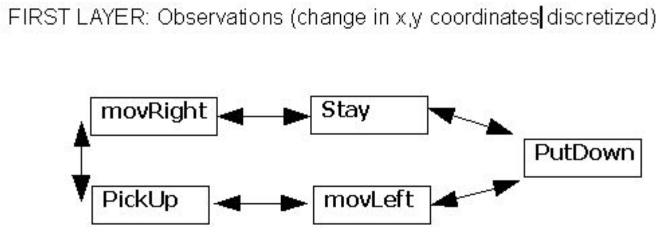


Fig. 4. Gaussian Models of the observations representing the small movements: pick up, put down, move left, move right, and stay.

The HMM model for these states is shown in figure 5. In order to do the recognition, we used the Baum-Welch algorithm to learn the parameters of the HMM, and then used the Viterbi algorithm (see figure 6) to recognize the small movements.



FIRST LAYER: Observations (change in x,y coordinates discretized)
movRight
Stay
PutDown
PickUp
movLeft

Fig. 5. HMM representing the pick up, put down, move left, move right, and stay states

In order to make the recognition online, we used a very small window for the Viterbi algorithm. Figure 6 shows an example of the recognition of a small movement. The Viterbi algorithm generated results with a .96 accuracy.

2) Large Activities: The larger activities are then recognized using the smaller movements as observations. The technique is the same for the smaller movements, i.e., the Baum-Welch algorithm to calculate the parameters of the HMM, then the Viterbi algorithm was used to recognize.

B. Algorithm for Recognition

The overall algorithm for the activity recognition then proceeds as follows:

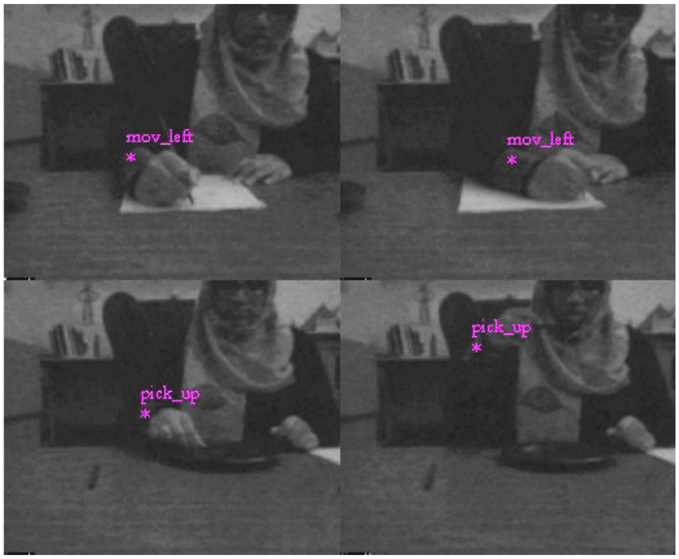


Fig. 6. An example of the activity recognition for a small movement on data from the Sony QRIO observing a human.

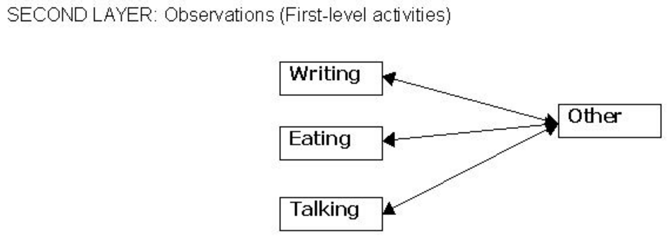


Fig. 7. HMM representing the eating, writing, speaking, and other states (other states is used to represent transition activities, usually consisting of one of the small movements).

First, training data is produced where each image retrieved from the video produced by the QRIO is given two labels: a small movement and a larger activity for training of both HMMs. The pink wristband is segmented in each image and the centroid calculated.

The parameters for both HMMs are then calculated using the Baum-Welch algorithm, where the small movements use the change in location of the centroid as observations, while the large activities use the small movements as observations.

Next, recognition is done online using an adjusted form of Viterbi algorithm, where instead of calculating the path between the states given by the HMMs, which would require the entire video, we only run the algorithm on a small window of data. In the case of the small movements, only two observations were used, for the large activities, four observations.

VI. Integration of Concepts

In this section, we describe the integration of the equivalence classes, hierarchical activity recognition, and MCOR algorithm for use on the data obtained from a Sony QRIO robot. More specifically, we outline the exact cues used.

When running an algorithm on a robot, a key component is speed, since it has to obtain results in real time. Because of this, all the cues used in the algorithm were obtained through fast and simple recognition systems described below:

ACTIVITY Activity information was obtained using the hierarchical activity recognition.

COLOR Color information was taken through segmentation using a color growing algorithm [20].

SHAPE Shape information was obtained by taking the shape aspect ratio of a tight bounding box around the object/segmented region, i.e.,

$$\text{Aspect Ratio} = \text{Bounding\_Height}/\text{Bounding\_Length}$$

SOUND Sound was used to determine whether a person was speaking or not.

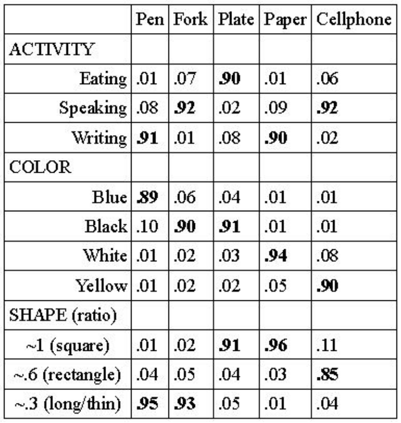
These cues were then used as data for the MCOR algorithm. As mentioned earlier, the MCOR algorithm was adjusted to include the concept of equivalence classes (see figure 3). Thus, for each frame of the video retrieved from the QRIO, the equivalence class algorithm was applied with the weights calculated according to MCOR [2], [1]. The addition of each cue allowed the equivalence classes to be narrowed down, until objects could be recognized. Thus, the equivalence classes and MCOR algorithm were combined together so information from multiple cues is utilized incrementally as the data from the QRIO robot is retrieved.

Although these are fairly simple cues, a major advantage of the MCOR algorithm and equivalence classes is the ability to combine multiple simple, fast cues that can be used to come up with results that would normally need more complicated cues, and slow extensive training.

VII. Experiments and Results

In order to demonstrate the effectiveness of these concepts, we ran the adjusted MCOR algorithm on video from a Sony QRIO robot. The robot remained static as it observed the interaction of a human with a set of 5 objects (Pen, Paper, Fork, Plate, and Cellphone) and using the presented cues.

Initially, the weights used in the online object recognition were calculated offline using simulated data [1]. The resulting object dictionary used in the recognition is shown in figure 8.



	Pen	Fork	Plate	Paper	Cellphone
ACTIVITY					
Eating	.01	.07	**.90**	.01	.06
Speaking	.08	**.92**	.02	.09	**.92**
Writing	**.91**	.01	.08	**.90**	.02
COLOR					
Blue	**.89**	.06	.04	.01	.01
Black	.10	**.90**	**.91**	.01	.01
White	.01	.02	.03	**.94**	.08
Yellow	.01	.02	.02	.05	**.90**
SHAPE (ratio)					
~1 (square)	.01	.02	**.91**	**.96**	.11
~.6 (rectangle)	.04	.05	.04	.03	**.85**
~.3 (long/thin)	**.95**	**.93**	.05	.01	.04

Fig. 8. Object dictionary used in the MCOR algorithm for recognition. Weights were determined through learning based on large amounts of data generated by a simulator.

MCOR was run on about four minutes of video. Below, we give a few examples of the results. We chose three scenarios that illustrate the concept of the equivalence classes described and the robustness of the algorithm in dealing with ambiguous cues. In the first instance, we illustrate its ability to recognize objects with similar shape. In the second, its ability to recognize objects with ambiguous activities, and in the third, its ability to recognize objects that can be recognized without ambiguity.

A. Distinguishing Ambiguous Shapes

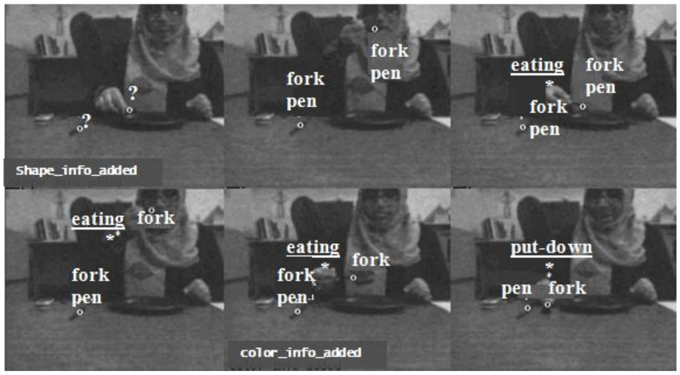


Fig. 9. Recognition results of a fork and a pen where both have the same shape. Results of the object recognition algorithm are given as labels embedded on to each frame of the video. Video is taken from the left-eye camera of a QRIO robot.

In this first scenario, we demonstrate how objects with ambiguous shapes are distinguished by the algorithm. There were two possible sets of objects that could be confused based on shape: the fork and pen (which are both long and thin) and the plate and paper (because the shape measure is based on the aspect ratio of the bounding box, the shape of the plate and paper come out to about the same, if more complicated shape recognition were used it is possible to distinguish them without having to add additional cues. This also nicely demonstrates how the algorithm can compensate for weak cues). See figure 9 for an example of the fork and pen scenario. The plate and paper yielded similar results, under the writing activity.

B. Distinguishing Ambiguous Activity

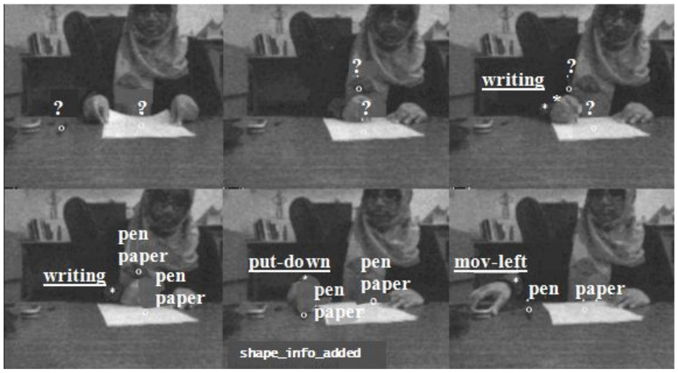


Fig. 10. Recognition of pen and paper, given that the activity of writing is associated with each.Results of the object recognition algorithm are given as labels embedded on to each frame of the video. Video is taken from the left-eye camera of a QRIO robot.

In the second scenario of ambiguous activities (see figure 10), we have two objects: pen and paper. In this scenario, activity information was provided first, in order to test the algorithm ability to recognize when visual information is not available right away. In the beginning in each of these instances, the same activity (writing) is associated strongly with both objects. Since the algorithm works by recognizing objects in sequence with the information it recieves from the video, we were able to narrow down the list of possible object to two (pen and paper).

Once visual information is added, the algorithm is able to take into account both color and shape. Because of the equivalence classes, it only applies one, i.e. color or shape, since either would be enough to put the objects in their separate classes, saving extra calculations.

C. No Ambiguity

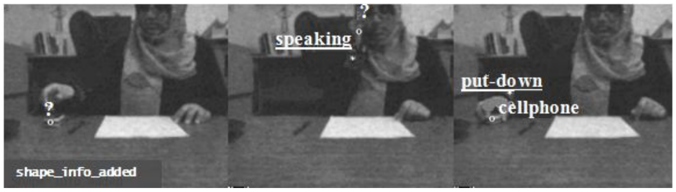


Fig. 11. Recognition results for the cellphone. Results of the object recognition algorithm are given as labels embedded on to each frame of the video. Video is taken from the left-eye camera of a QRIO robot.

In the third task, we wanted to illustrate that the MCOR algorithm can recognize objects on the first shot, without having to disambiguate between other objects.

In figure 11, the results of the algorithm are shown, where cellphone was successfully recognized without ambiguity with another object, taking advantage of the fact that it has a very distinctive property none of the other objects have, i.e., its association with talking.

Thus, the algorithm was able to demonstrate its use in real robot data, correctly identifying multiple interactionable objects.

VIII. Conclusion

In summary, we have identified and termed the subset of objects which can be utilized in multiple-cue object recognition and other methods which depend on interaction to gain information, i.e., "interactionable" objects. We have shown that the multiple cue object recognition (MCOR) algorithm can be applied to real robot data as demonstrated on the Sony QRIO. We successfully introduced the concept of equivalence classes into the MCOR algorithm, used a hierarchical activity recognition algorithm that correctly produced activity labels useful in the overall object recognition. The QRIO was able to identify all specified objects. Our approach further shows how multiple-cues can allow simple fast cues to produce recognition results that would often require much more time and complication, or unrealistic restrictions on real-world data to produce the same results.

Acknowledgments

We would like to thank SONY for their generous loan of their remarkable QRIO robots, which we very much enjoyed using in our research. This research was partly sponsored by the National Science Foundation under grant no. IIS-0637069 and by the Boeing Corporation. The views and conclusions contained in this document are those of the authors only.

References

[1] S. Aboutalib and M. Veloso. Simulation and weights of multiple cues for robust object recognition. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2007.

[2] S. Aboutalib and M. Veloso. Towards using multiple cues for robust object recognition. In *AAMAS '07: Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, pages 1–8, New York, NY, USA, 2007. ACM.

[3] S. Agarwal, A. Awan, and D. Roth. Learning to detect objects in images via a sparse, part-based representation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 26(11):1475–1490, November 2004.

[4] R. Bischoff and V. Graefe. Dependable multimodal communication and interaction with robotic assistants. In *Proceedings of the 11th International Workshop on Robot and Human Interactive*, 2002.

[5] G. Burghouts and J.-M. Geusebroek. Performance evaluation of local colour invariants. *Computer Vision and Image Understanding: CVIU*, 113:48–62, 2009.

[6] T. Freire, B. Filho, M. S. Filho, E. O. Freire, R. Alex, C. Freitas, and H. J. Schneebeli. Object recognition for an agent-based controlled mobile robot.

[7] J. Gibson. *The theory of affordance. In: Percieving, Acting, and Knowing*. Lawrence Erlbaum Associates, Hillsdale, NJ, 1977.

[8] A. Haasch, N. Hofemann, J. Fritsch, and G. Sagerer. A multi-modal object attention system for mobile robot. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005.

[9] O. Hempel, U. Bker, and G. Hartmann. agent-based object recognition. *ICEIA*, 2000.

[10] M. Higuchi, S. Aoki, A. Kojima, and K. Fukunaga. Scene recognition based on relationship between human actions and objects. *Proc. of 17th International Conference of Pattern Recognition*, 3:73–78, 2004.

[11] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[12] R. Marre, P. Geurts, J. Piater, and L. Wehenkel. Decision trees and random subwindows for object recognition. *ICML workshop on Machine Learning Techniques for Processing Multimedia Content (MLMM2005)*, 2005.

[13] B. Moghaddam and A. Pentland. Probabilistic visual learning for object detection. In *International Conference on Computer Vision (ICCV'95)*, pages 786–793, Cambridge, USA, June 1995.

[14] D. J. Moore, I. A. Essa, and M. H. Hayes. Exploiting human actions and object context for recognition tasks. In *ICCV (1)*, pages 80–86, 1999.

[15] K. Murphy, A. Torralba, and W. Freeman. Using the forest to see the trees: a graphical model realting features, objects, and scenes. *NIPS*, 16, 2003.

[16] E. Murphy-Chutorian and J. Triesch. Shared features for scalable appearance-based object recognition. *Proc. IEEE Workshop Applications of Computer Vision*, January 2005.

[17] E. Rivlin, S. J. Dickinson, and A. Rosenfeld. Recognition by functional parts. *Computer Vision and Image Understanding*, 62:164–176, 1995.

[18] M. M. Veloso, P. E. Rybski, and F. von Hundelshausen. Focus: a generalized method for object discovery for robots that observe and interact with humans. In *Proceedings of the 2006 Conference on Human-Robot Interaction*, March 2006.

[19] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2001.

[20] F. von Hundelshausen and R. Rojas. Tracking regions and edges by shrinking and growing. *In Proceedings of the RoboCup 2003 International Symposium*, 2003.

[21] D. Wilking and T. Rofer. Real-time object recognition using decision tree learning. *In. RoboCup 2004: Robot World Cup VIII, Lecture Notes in A.I.*, (3276):556–563, 2005.