Multiple-Cue Object Recognition on Outside Datasets

Sarah Aboutalib* and Manuela Veloso**

Abstract— **This work builds upon the fact that robots can observe humans interacting with the objects in their environment, and that humans provide numerous non-visual cues to the identity of objects. In previous work, we outlined a Multiple-Cue Object Recognition (MCOR) algorithm which attempted to use multiple features of any type to produce more robust object recognition. All results so far reported with MCOR has been on data collected by ourselves.**

In this work, we introduce new advancements in the MCOR algorithm to increase its effectiveness and ability to deal with complex real data from outside datasets. These advancements include the integration of Scale-Invariant Feature Transform (SIFT) features and an improvement in training. To demonstrate the effectiveness of the MCOR framework, we first show a comparison of the MCOR algorithm to an outside dataset to show its basic advantages. We then demonstrate the advanced MCOR features on real cooking video datasets.

I. Introduction

There are a variety of tasks where an agent's ability to accomplish them depends heavily on the reliable recognition of the objects in the environment. Category-level Object recognition however has proven to be a significantly difficult challenge especially with the complexity of real world data, where there is great variation in both the appearance of objects within a single object class (e.g. chairs come in many shapes and colors), and in the appearance of the same object under various circumstances (e.g. the same chair can appear different with changes in lighting, view, and orientation).

Several approaches have attempted to focus on learning the visual features of an object in order to recognize it. Although great progress has been made along these lines, there is still much to be done in order to build an object recognition system that can be used under any of the various situations that must be dealt with real data.

In dealing with this complexity, we are particularly interested in the important observation that the environment and context around an object can provide numerous non-visual cues to the identity of the objects, such as the interaction of humans with those objects, which can then be utilized by an agent observing the interaction. The benefit of including non-visual information is supported by the success made by a few approaches, which have successfully integrated non-visual cues, although generally restricted to a single type of non-visual information.

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]. We defined a context-dependent subset of objects, "interactionable" objects, as objects that can interact or be interacted with. This set of objects is most applicable to our multiple-cue algorithm [3] and functional recognition algorithms.

In this work, we outline advancements in the algorithm, making improvements on two key features:

Integration of SIFT Features In order to enhance the visual description of the objects being learned, the advanced MCOR algorithm includes scale-invariant feature transform (SIFT) features [8] to the definition of each object.

Training In previous work, learning was previously done on simulated training sets, in the work we describe the method used to develop a greater training dataset and the information that it provides.

These advancements allowed the MCOR algorithm to be more easily and effectivly used on other datasets besides those collected by ourselves. In order to evaluate our framework, we first show a comparison of the MCOR algorithm to another object recognition technique and dataset that also uses activity information for object recognition. We then demonstrate the advancements of the MCOR algorithm on real cooking video datasets.

II. Related Work

There have been numerous vision-alone-based object recognition systems [9], [4], [17], [12], [8]. Although fast and accurate results have been demonstrated by these techniques, the dependence of these approaches on visual cues alone make them susceptible to variations in size, lighting, rotation, and pose, all of 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 [11] and activity cues [10], [16].

Functional recognition [15], [14], [19], [18], a technique which uses affordance properties to determine the function and finally identity of the object, is another category of algorithms, similar to MCOR, which attempts to use the way people interact with objects in order to identify an object either in place of or in conjunction with its visual attributes.

Encouraged by the general success of these approaches in integrating a non-visual cue for more robust object recognition, moving past just functional information to include other cues such as speech, MCOR [2] provides a general framework for flexibly including multiple cues of any number and

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

**M. Veloso is Faculty with the Computer Science Department, School of Computer Science, Carnegie Mellon University, PA 15213, USA veloso@cmu.edu

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. MCOR

The MCOR framework is based on object recognition using an unrestricted number of cues. It provides a flexible framework for including evidence from any type of information. It is able to use human interaction with the objects to remove ambiguity. Figure 1 shows a flow diagram of MCOR.

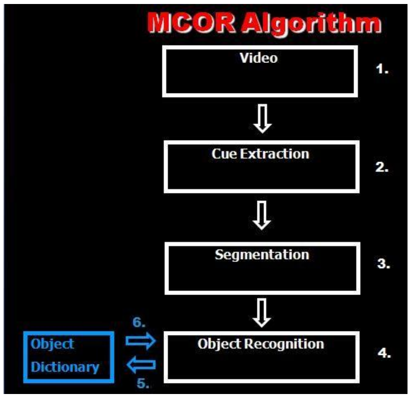


Fig. 1. Flow of MCOR framework: (From top to bottom) (1.) Get frame from video, (2.) extract cues from the image, (3.) segment the image and associate extracted cues with a segmented region, (4.) recognize objects based on dictionary, (5.&6.) update object dictionary based on the recognized objects, search image again with updated dictionary including updated visual description of object.

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. This becomes a possible object candidate.

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. It is learned using a Probabilistic Relational Model see [1] for details.) is given. The evidence, $e_{k,i}$ that the region, r_k belongs to a particular object class, i is then calculated using the equation:

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

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 gets added to its definition. In this way, new cues can be added to an objects definition in the dictionary and generalization can occur.

IV. Scale-Ivariant Feature Transform in MCOR

Previously, when a visual description of the object being identified was grabbed, the visual description consisted of color information and the aspect ratio of the bounding-box around the segment produced by a region-growing color segmentation algorithm [2]. In the advanced version of MCOR, in addition to the color and shape information, we grab SIFT (Scale-Invariant Feature Transform) features from the area within the bounding box to build a model of the object.

SIFT features [8] are well-known descriptors used widely through computer vision and object recognition tasks. These features are useful because they provide highly descriptive texture-based features which are robust to most changes in scale and rotation.

Because it would be too computationally intensive and unnecessarily repetitive to calculate a feature for every pixel location, we calculate SIFT features only at interesting points in the image such as corners. There are numerous corner detectors that can be used, we used the common Harris Corner Detector [6], defined by the equation below:

$$A = \sum_u \sum_v G(\mathbf{u}, \sigma) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} = \begin{bmatrix} < I_x^2 > & < I_x I_y > \\ < I_x I_y > & < I_y^2 > \end{bmatrix}$$

A is a Harris matrix whose eigenvectors and values can be used to determine areas of interest, i.e. if the first and second eigenvalues have a zero response, there are no features of interest. If one has a large response, an edge is present. If both have a large response, a corner is found. Brackets, $<>$, indicate averaging over the summation of all pixel locations, $\mathbf{u}$, and $G(u,v)$ corresponds to the Gaussian function used.

In SIFT, key locations are further deciphered by using the maxima and minima of the results of the difference of Gaussians are applied in scale space to a series of smoothed and sampled images in an image pyramid.

Dominant orientations are assigned to each interest point using a 128 dimensional vector formed from a histogram of image gradients in the neighborhood of the interest point. See [8] for exact details and functions. This produces key points that are more stable and robust to changes in orientation and scale.

In MCOR, we start with some initial training images from which SIFT features are taken and stored in a database in the object definition. An example of the interest points where SIFT features were extracted are shown in figure 2. Additional model images are added to the database if a segment is determined to be recognized as a particular object category by MCOR. In this way, new visual features can be learned for each object dictionary as recognition occurs.

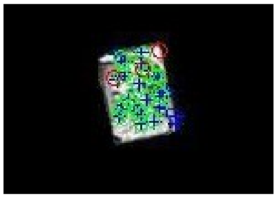


Fig. 2. Each point in the figure corresponds to the location of an extracted SIFT feature. This is a training example of a cereal box used for recognition. Note the numerous number of feature points due to the complex texture on the cereal box.

If a model for an object is already stored in a database, then we look for that object in each of the frames. In figure

3, we show an image of the SIFT features extracted during recognition. These key points are then compared with the key points saved in each object dictionary using a Hough Transform to determine the strength of the match. The similarity is then used as the similarity measure described in the evidence equation in section 3. Weight is set to 1 since there we assume that a SIFT model of an object is strongly associated with that object. It is then up to the similarity to determine the extent of the evidence provided by the SIFT response.

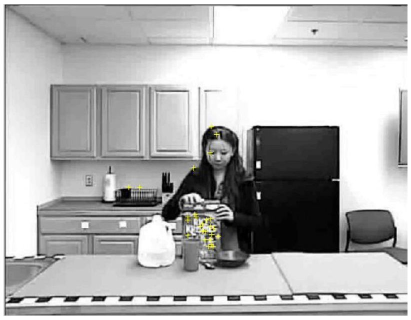


Fig. 3.

Thus, MCOR is able to create a more detailed visual descriptor of the objects being described by including scale-invariant feature transform models to each object definition.

V. Training

Another advancement made to the MCOR algorithm is an improvement in the training data. In order to train the algorithm in previous experiments, we used simulated data programmed off of a human-defined model of cue and object associations. In order to get a better reflection of real world data, we developed and used tools that could produce a useful dataset for object recognition of ‘interactionable’ objects.

Because MCOR and functional recognition algorithms are able to utilize cues provided by the interaction of humans with objects. The training mechanisms described here were based on producing datasets with those characteristics. There are two mechanisms used to produce such datasets which we describe below: (1) Scene sorting, which is used to neutralize the common camera shifting of data taken from real world datasets, and (2) Labeling using ViPER, which uses a labeling program from the University of Maryland and develops a labeling framework to provide useful information to ‘interactionable’ based algorithms.

A. Scene Sorting

Because we want training datasets which are primarily generated from real world datasets in order to show the advantages of the MCOR algorithm in the real world. The problem is most real world video datasets, except for those filmed in a restricted laboratory, consist of frequent shifts in camera angles (think of any real world TV show such as cooking shows). In order to compensate for these sudden shifts in camera angle, we outline a scene gist algorithm used to sort different camera angles or scenes into bins with similar scenes, so they can be processed using the same tracking or computer vision parameters that would need to be adjusted if the scene were constantly shifting.

In our scene sorting algorithm, we use local-intensity histograms to separate viewing angles. This is done by dividing images into four sections. A gray value histogram of each section is then taken. The image is divided into four regions in order to provide some spatial information, if a single histogram was used for the entire image all spatial information would be lost. Although dividing the image into even more sections would provide more spatial information, it also reduces the robustness. Through past experience, it has been determined that four components usually provides the best balance between the tradeoffs. See figure 4 for an example of the gist descriptor.

The histogram is then compared to a database of stored histogram values. If the histogram is similar to one already found in the database it is labeled under the same scene category. If it is different then any other histogram according to a particular histogram, it is then added to the database as a new scene category.

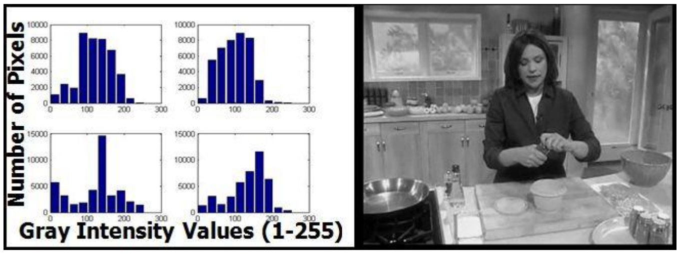


Fig. 4. Gist information extracted and used in order to separate different scenes.

Once the video clips are properly sorted into their corresponding scene/camera viewing angle categories, we then need a way to label each of the clips with all the information that would be useful for learning for an ‘interactionable’ based algorithm.

B. Labeling using ViPER

There are a number of video labeling products out there, but we chose to use ViPER: Video Performance Evaluation Resource [7] by the Language And Media Processing (LAMP) at the University of Maryland because it allows you to easily define your own schema of labeling as well as allows for duplication and interpolation when labeling an object across a large number of frames.

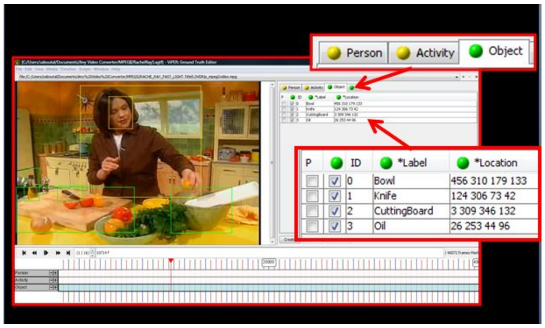
Using ViPER, we developed an ‘interactionable’ schema which can provide useful information about objects and their interactions. Thus, we labeled the following:

- Person
  - Face: Bounding box around frontal face
  - Body: Bounding box encompassing entire body seen
  - Hands: Bounding box around left and right hand
- Object
  - Label: Object category of that object

  - Location: Bounding box around object
  - Segment: Polygon around the true segmentation of the object.
- <Cue> - can be replaced with any cue type, ex: Activity.
  - Value: Object category of that object
  - Person: Person providing or related to the cue
  - Objects: Objects being affected or generating the cue

This schema (see figure 5) allows for a large variety of learning opportunities for example once could learn the average distance of an activity cue as related to the person doing the activity (for instance the location of the face) from an object. This information can then be used to reduce the search area of the segment of the object being recognized.

This then creates XML files for easy processing which can then be used for easy training and performance evaluation.



P	●	ID	● *Label	● *Location
□	☑	0	Bowl	456 310 179 133
□	☑	1	Knife	124 306 73 42
□	☑	2	CuttingBoard	3 309 346 132
□	☑	3	Oil	26 253 44 96

Fig. 5. Example of Labeling using ViPER.

The data provided by these techniques provide a good base for providing training data which could potentially be used in any real world circumstance where similar objects are present, so a robot could find training from cooking videos useful when running in a kitchen setting, for example.

VI. Experiments and Results

In order to evaluate our framework, we first show a comparison of the MCOR algorithm to another object recognition system and dataset that also uses activity information in determining the evidence of the presence of a particular object. We then demonstrate the advancements of the MCOR algorithm on cooking video datasets.

A. MCOR Comparison using Outside Dataset

In order to demonstrate the ability of the MCOR algorithm to run on datasets outside of our own, we first compare results produced by Gupta et al. [5] on their dataset.

The dataset consist of forty-six videos that were five to ten seconds long. Frame size was 640x480 pixels.

In this work, we have successfully applied the MCOR algorithm to this Gupta set, demonstrating: (i) An equivalent object recognition accuracy to the results reported by Gupta, (ii) the additional ability to generalize from previous experience, i.e., the algorithm visually recognizes objects without the additional activity recognition, after it has previously learned the association, and (iii) the successful engineering of running the algorithm in other datasets.

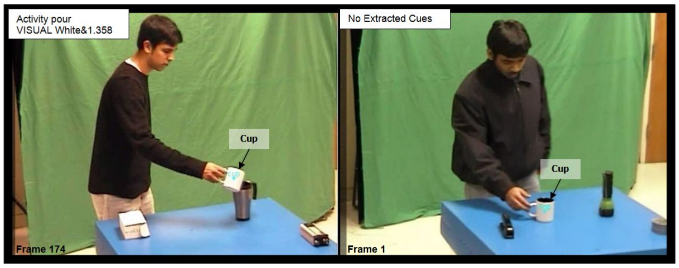


Fig. 6. Captured frames taken from two different videos from the Gupta and Davis dataset. In the left, the 150th frame was captured at the point when the activity of pouring was recognized. This allowed the cup to be recognized and the visual features to be stored in the cups dictionary. Thus, when the next video was processed with the updated definition. The cup was able to be recognized on the first frame without any other cue information.

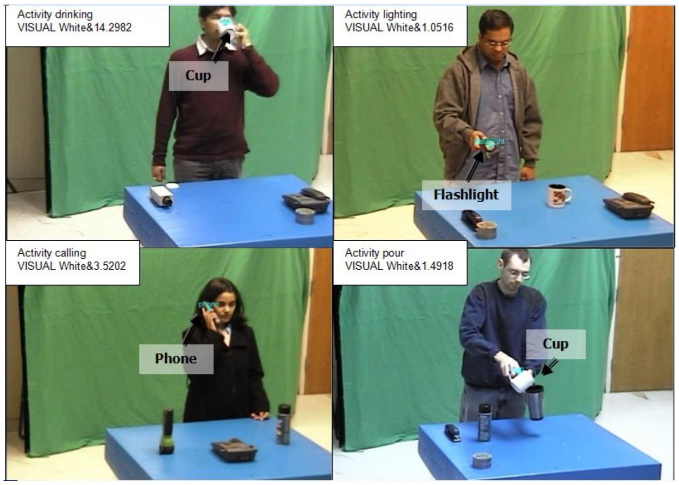


Fig. 7. Captured frames taken from results video generated after processing various video clips from the Gupta and Davis Dataset. Each frame shows the point in the video when an activity cue was recognized, which allowed for the object being used to be recognized and the color and shape information (top right of each frame) to be stored.

The Gupta dataset is an optimal choice for frameworks interested in 'interactionable' objects, i.e., objects that are interacted with and interact, such as the MCOR algorithm. The Davis's group used these videos to do activity recognition and then object recognition based on these activities. Here is an overall description of the dataset:

- Number of Videos: 46 videos each about 5-10 secs
- Objects Recognized: Spray can, Phone, Cup, Flashlight
- Activities Recognized: 5 activities: Spraying, Answering, Lighting, Pouring, Drinking

Gupta was able to recognize 98.67 percent of the objects using activity information based on a histogram of oriented gradients using an adaboost classifier, this was an improvement over the 78.33 percent recognition rate without activity information.

We did not have access to their activity recognition, so for our object recognition and for comparison with the results from Gupta, we manually created the activity recognition information for each of the videos according to the activities outlined by [5].

MCOR provides a color-based region segmentation and tracking algorithm (extending my previous algorithm and implementing it in Matlab) that can segment objects in the images using color-based region growing. It then tracks that region according to proximity, shape and color [2], [3].

We then ran the MCOR algorithm using the automated visual object segmentation and tracking and the manually annotated activity information.

Given an object dictionary with the activity information and the visual information, MCOR was able to achieve a 100 percent recognition rate, as the manual annotations on activity had no noise.

Both my 100 percent and the Gupta reported 98.67 percent object recognition excellent performance, are not surprising as this Gupta dataset has a one-to-one association of an activity to an object. Example of our results can be found in figures 6 and 7.

One of the strengths of the MCOR approach is that it allows a many-to-many weighted associations between objects and cues, which is not tested demonstrated with the Gupta dataset. Furthermore, another main contribution of MCOR is the ability to learn an association and generalize from it to similar visual situations.

Interestingly, using the Gupta dataset, after MCOR recognizes an object through the association of the visual and activity cues, it is able to generalize and recognize objects solely from their updated visual description without the need for the use of the activity information. For example, the white cup in one video (captured frame shown in left image in figure 6) was actually recognized based on the color and shape information learned from another video (captured frame shown in right image in figure 6).

In summary, with this dataset, we have shown that the MCOR algorithm can utilize datasets outside those generated by our own work. MCOR got comparable results and showed its generalization capabilities.

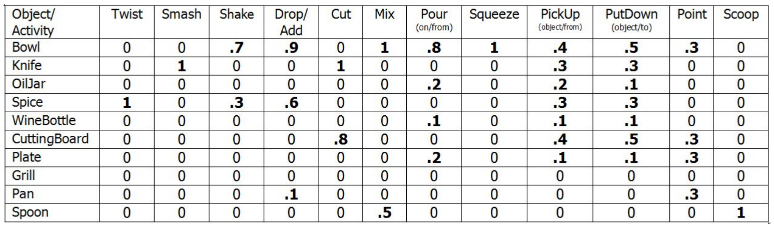
B. Advanced MCOR on Cooking Data

In this section, we show how the training and SIFT advancements were used in order to enhance the MCOR framework. First, we show the utilization of the training to learn the weights used in the object dictionary. In previous work, this had always been determined either by hand or based off of simulated data. Second, we show the benefit of the SIFT features for enhancing the visual descriptors for the MCOR algorithm.

1) Training: Training was done on two main sources:

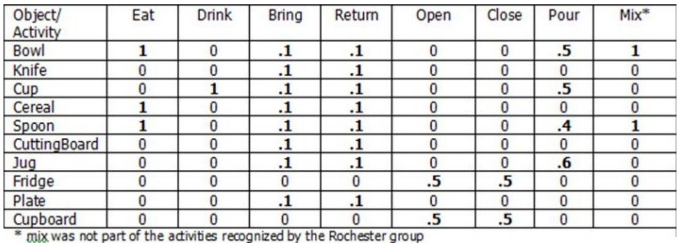
- Rachel Ray videos: Learning was done on 100 video clips from 1 to 30secs, from 3 half-hour videos (Ripped from DVD)
- LACE dataset, University of Rochester [13]: Learning was done on 32 video clips from 1 to 3mins, from 8 half-hour videos

In figures 8 and 9, we show tables showing the probabilities learned from the XML data generated by our ViPER schema. These probabilities represent $P(O|C)$, where O is the object category and C, a cue value. One can see how certain activities are more strongly associated with a particular object such as 'Pour' to 'Bowl' with a value of .8, while other activities such as 'PutDown' are more widely distributed across all the objects and thus less weighted when giving evidence for any particular object.



Object/ Activity	Twist	Smash	Shake	Drop/ Add	Cut	Mix	Pour (on/from)	Squeeze	PickUp (object/from)	PutDown (object/to)	Point	Scoop
Bowl	0	0	**.7**	**.9**	0	**1**	**.8**	**1**	**.4**	**.5**	**.3**	0
Knife	0	**1**	0	0	**1**	0	0	0	**.3**	**.3**	0	0
OilJar	0	0	0	0	0	0	**.2**	0	**.2**	**.1**	0	0
Spice	**1**	0	**.3**	**.6**	0	0	0	0	**.3**	**.3**	0	0
WineBottle	0	0	0	0	0	0	**.1**	0	**.1**	**.1**	0	0
CuttingBoard	0	0	0	0	**.8**	0	0	0	**.4**	**.5**	**.3**	0
Plate	0	0	0	0	0	0	**.2**	0	**.1**	**.1**	**.3**	0
Grill	0	0	0	0	0	0	0	0	0	0	0	0
Pan	0	0	0	**.1**	0	0	0	0	0	0	**.3**	0
Spoon	0	0	0	0	0	**.5**	0	0	0	0	0	**1**

Fig. 8. Learned weights for the Rachel Ray Cooking dataset. Each number represents the probability of an object (rows) given each cue value (columns).



Object/ Activity	Eat	Drink	Bring	Return	Open	Close	Pour	Mix*
Bowl	**1**	0	**.1**	**.1**	0	0	**.5**	**1**
Knife	0	0	**.1**	**.1**	0	0	0	0
Cup	0	**1**	**.1**	**.1**	0	0	**.5**	0
Cereal	**1**	0	**.1**	**.1**	0	0	0	0
Spoon	**1**	0	**.1**	**.1**	0	0	**.4**	**1**
CuttingBoard	0	0	**.1**	**.1**	0	0	0	0
Jug	0	0	**.1**	**.1**	0	0	**.6**	0
Fridge	0	0	0	0	**.5**	**.5**	0	0
Plate	0	0	**.1**	**.1**	0	0	0	0
Cupboard	0	0	0	0	**.5**	**.5**	0	0

* mix was not part of the activities recognized by the Rochester group

Fig. 9. Learned weights for University of Rochester (LACE Dataset). Each number represents the probability of an object (rows) given each cue value (columns).

This training data then provided the weight values that were then used in the MCOR algorithm when recognizing objects. Since the recognition results are similar to those shown in previous papers [2], we will mainly focus in the next section on the new SIFT adaptation.

2) SIFT Results: The inclusion of the SIFT features provides an added benefit to the MCOR algorithm, where it can now visually recognize objects it could not do so previously. In figure 10, we show some examples of SIFT models extracted from one training example of a knife, a cereal box and wine bottle. Note how the objects with a more interesting texture pattern (such as the cereal box) provides more interest points for the SIFT model.

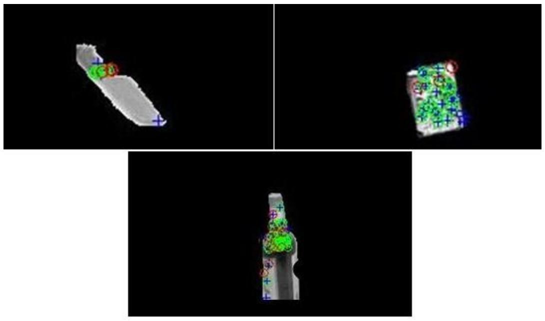


Fig. 10. Examples of SIFT models extracted from one training example of a knife (top left), a cereal box (top right), and a wine bottle (bottom).

It is these complex textured items which gave the previous MCOR algorithm difficulty, primarily because the previous version was solely dependent on getting a color segmentation to visually describe the object (see figure 12 for examples). This meant textually complex objects would provide strange and difficult segmentations and so could not be recognized as often in each frame. With the new MCOR algorithm which looks for SIFT features as well, these objects are no longer a major problem.

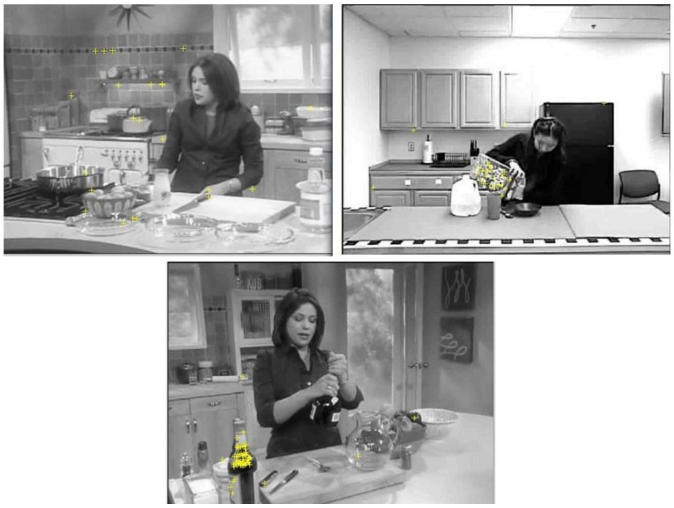


Fig. 11. SIFT features, represented by yellow + signs, found to match the model of the knife (top left), the cereal box (top right), and the wine bottle (bottom). Note the large number of matched SIFT features for the ceral box because its large amounts of texture, and the small number for the less textured knife.

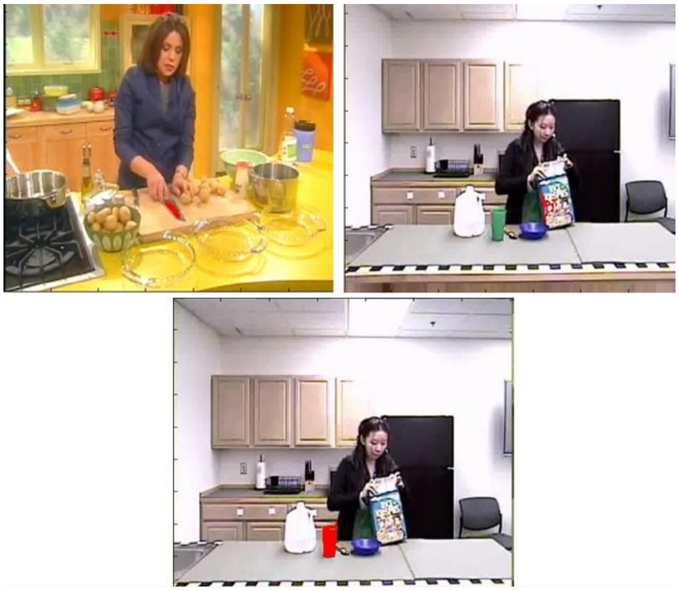


Fig. 12. Color segmentation results, colored in with red, used in color and shape features used by MCOR. Top left shows the segmentation of the knife, top right shows the segmentation of the cereal box, and bottom shows segmentation of a cup. Note the better segmentation of the homogenous colored knife and the less precise segmentation of the more textured cereal box.

It is important to note however, that using the SIFT features alone does not provide enough information to recognize all the objects that MCOR can. Take the knife for instance in figure 11. Because of the smooth texture of the knife, it could not get enough interest points to properly find it in the frame. This shows the benefit of the integration of multiple cues by the MCOR algorithm, which can find the knife (figure 12) using the color-segmentation growing as well as the cutting activity, even if the SIFT feature model cannot.

VII. Conclusion

With this paper, we have demonstrated the advantage of MCOR compared to another functional recognition object recognition system, where we showed both a match and slight improvement in the recognition rate and the added ability to generalize object visual properties to new videos. In addition, we outlined and demonstrated the advancement of new MCOR features including the use of SIFT features and a new framework for training data.

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. Aboutalib and M. Veloso. Cue-based equivalence classes and incremental discrimination for object recognition. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2009.

[4] 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.

[5] A. Gupta and L. S. Davis. Objects in action:an approach for combining action understanding and object perception. *CVPR*, 2007.

[6] C. Harris and M. Stephens. A combined corner and edge detector. *Proceedings of the 4th Alvey Vision Conference*, pages 147–151, 1988.

[7] Language and M. P. Laboratory. The video performance evaluation resource. *http://viper-toolkit.sourceforge.net/*.

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

[9] 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.

[10] 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.

[11] 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.

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

[13] U. of Rochester. Lace: Indoor activity benchmark dataset. *http://www.cs.rochester.edu/ spark/muri/*.

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

[15] M. Stark, P. Lies, M. Zillich, J. Wyatt, and B. Schiele. Functional object class detection based on learned affordance cues. *Computer Vision Systems: 6th International Conference (ICVS)*, pages 435–444, 2008.

[16] 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.

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

[18] P. Winston, B. Katz, T. Binford, and M. Lowry. Learning physical descriptions from functional definitions, examples, and precedents. *AAAI*, 1983.

[19] K. Woods, D. Cook, L. Hall, K. W. Bowyer, and L. Stark. Learning membership functions in a function-based object recognition system. *Journal of Artificial Intelligence Research*, (3):187–222, 1995.