Simulation and Weights of Multiple Cues for Robust Object Recognition*

Sarah Aboutalib
Carnegie Mellon University
Computer Science Department
Pittsburgh, Pennsylvania
saboutal@cs.cmu.edu

Manuela Veloso
Carnegie Mellon University
Computer Science Department
Pittsburgh, Pennsylvania
veloso@cmu.edu

ABSTRACT

Reliable recognition of objects is an important capabaility in the progress towards getting agents to accomplish and assist in a variety of useful tasks such as search and rescue or office assistance. Numerous approaches attempt to recognize objects based only on the robot's vision. However, the same type of object can have very different visual appearances, such as shape, size, pose, color. Although such approaches are widely studied with relative success, the general object recognition task still remains difficult. In previous work, we introduced MCOR (Multiple-Cue Object Recognition), a flexible object recognition approach which can use *any multiple cues*, whether they are visual cues intrinsic to the object or provided by observation of a human. As part of the framework, weights were provided to reflect the variation in the strength of the association between a particular cue and an object. In this paper, we demonstrate how the probabilistic relational framework used to determine the weights can be used in complex scenarios with numerous objects, cues and the relationship between them. We develop a simulator that can generate these complex scenarios using cues based on real recognition systems.

Categories and Subject Descriptors

I.2.10 [**Vision and Scene Understanding**]: Perceptual reasoning, Video analysis

General Terms

Algorithms

Keywords

Computer Vision, Object Recognition

*(Produces the permission block, and copyright information). For use with SIG-ALTERNATE.CLS. Supported by ACM.

IROSS '07 San Diego, California, USA

1. INTRODUCTION

The complexity of real world data has made it a difficult challenge to give robots the ability to recognize objects, a property that can prove quite vital when that robot is attempting to assist humans. 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 make object recognition a difficult problem.

In addressing this issue, we have introduced a Multiple-cue object recognition (MCOR) framework and algorithm [1] which builds upon the fact that robots can observe humans interacting with the objects in their environment, and thus are provided numerous non-visual cues.

MCOR is a flexible object recognition approach which can use any multiple cues, whether they are visual cues intrinsic to the object or provided by observation of a human. MCOR takes into account that multiple cues can have different weight in their association with an object.

In this paper, we provide a probabilistic relational framework that can be used to determine the weights of cues in complex scenarios with numerous objects, other cues and the relationship between them. We develop a simulator that can generate these complex scenarios using cues based on real recognition systems.

2. RELATED WORK

There have been numerous visual-based object recognition systems [5, 2, 14, 8, 4]. Although fast and accurate results have been demonstrated by these techniques, the dependence of these approaches on visual cues alone make them susceptable 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[7] and activities[6, 13].

Encouraged by the general success of these approaches in integrating a non-visual cue for more robust object recognition, MCOR provides a general framework for flexibly including multiple cues of any number and any type, so that all the cues mentioned above such as activities, visual features, and context, in addition to any other possible cues available now or in the future, can be used to provide evidence for the presences of the object.

In order to aid in the representation of this framework, we use probablistic relational models (explained in detail later sections) to determine the weight of the association between a cue and an object. Although PRMs have been used in

a number of other domains such as the web [11], movies [**?**], and genes [3], we attempt to represent the complicated world of objects and object recognition.

3. MULTIPLE CUE OBJECT RECOGNITION (MCOR)

3.1 Object Dictionary

An object in other object recognition methods is usually described using particular types of information defined at the outset, although the content of the information is allowed to change, the actual structure of the definition is not. An object is defined as a set of cues, C_{o_i} (o_i represents the i^{th} object to be recognized), where the cues can be of any type and the set can be of any size. The cues follow a standard format described in [1] which allows them to be used generically in the algorithm, thus removing the necessity of having to define and limit the type of information that can belong to a an object.

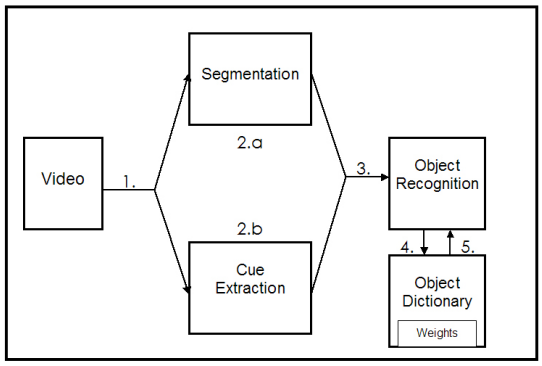

Figure 1: Flow of MCOR framework: (1.) Get image, (2.a) segment the image, while at the same time (2.b) extract cues from the image, then (3.) associate extracted cues with a segment, (4.) recognize objects based on dictionary, and (5.) update object dictionary based on the recognized objects

Segmented regions in the image are then be recognized as objects by comparing the cues extracted from the scene with cues in the object definition using properties associated with each cue. All cues must define a cue type, cue value, spatial association, temporal association, weight and simularity function, so that they can all be treated in a standard manner by the algorithm (see [1] for details).

4. MCOR ALGORITHM

The algorithm for object recognition with multiple cues, i.e. Multiple Cue Object Recognition (MCOR), then proceeds as follows(see figure 2 for pseudocode of the algorithm.).

4.1 Object Recognition

Region, r_k, is then recognized as the object with the greatest evidence, if it is above a given threshold, θ, i.e.,

$$label_k \leftarrow \text{argmax}_{o_i}\, e_{k,o_i}, \text{ if } \max e_{k,o_i} > \theta$$

4.2 Empirical Validation using Real Data

Given a set of cues for each object:

- For each object, o_i, in the set of possible objects to be recognized, O:
  - There should be a set of cues, C_{o_i}.
  - Each cue, c_l, in C_{o_i} represents a cue that is associated (i.e. indicates) object o_i and which has:
    - a cue value, cue_value_l
    - a temporal association, Δf_l
    - a spatial association, Δp_l
    - a weight, w_{o_i,c_l}
    - a similarity measure to calculate the similarity, s_{c_j,c_l} between cue c_l and another cue c_j

Analyze the video:

- For each frame of the video, F_t:
  - Extract all cues that belong to $\bigcup_i C_{o_i}$
  - For each new cue extracted, c_j, with cue_value_j:
    - Get current position, P_j.
    - If $P_j - \Delta p_j$ at $F_{t-\Delta f_j}$ is within any region $r_k \in R$, where R is the set of segmented regions to be recognized as objects:
      - Store c_j in C_k, the set of cues attached to that region.
    - Else:
      - Extract a new region, r_k, at position $P_j - \Delta p_j$ and frame $F_{t-\Delta f_j}$ and store it in R.
      - Store c_j in the currently empty C_k
  - For each region, $r_k \in R$:
    - For each object, $o_i \in O$:
      - Calculate the evidence, e_{k,o_i}, that region r_k is object o_i as follows:

$$e_{k,o_i} = \sum_{c_l \in C_{o_i}} \sum_{c_j \in C_k} w_{o_i,c_l} s_{c_j,c_l}$$

        if the cue type of c_l is not the same as c_j, then $s_{c_j,c_l} = 0$
    - Region r_k is then recognized as the object with the greatest evidence, if it is above a threshold, θ, i.e.

$$label_k \leftarrow \text{argmax}_{o_i}\, e_{k,o_i}, \text{ if } \max e_{k,o_i} > \theta$$

    - Add all cues, $c_j \in C_k$, to the set of cues in the object definition, C_{label_k}, if $\vee c_l \in C_{label_k}, s_{c_j,c_l} \neq 1$
    - If the current label, i.e. $label_k$ at F_t is different from $label_k$ at F_{t-1} and $label_k$ at F_{t-1} exists:
      - Remove all $c_j \in C_k$ added before F_t from $C_{o_{old}}$, where $o_{old} = label_k$ at F_{t-1}.

Figure 2: Algorithm for Multiple-Cue Object Recognition

The feasibility of the MCOR framework was demonstrated on real video data. Object recognition tasks were given to test various capabilities of the algorithm. For example, in one scenario, we demonstrated how the algorithm can deal with an unreliable cue type.

All the tasks started off with object definitions which contained cues that could be observed from the interaction of a human with the object and a weight value corresponding to the strength of the association between the cue and the object (currently human determined). The definitions are begun with cues that involve human interaction since those cues tend to be more obviously associated with the

object since humans usually interact with the object in a manner implicit to its definition. For instance, most interactions tend to portray the function of the object which is usually an important part of its definition. Additional cues are learned by the algorithm later.

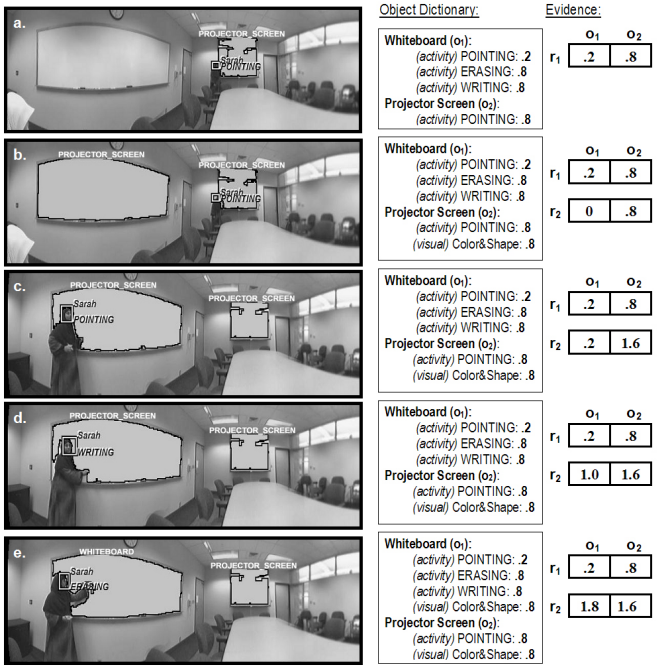

Figure 3: The use of evidence to correct a mislabeled object due to an unreliable visual cue.

4.3 Weight Learning with Synthetic Data

In order to illustrate how the weight values for each cue and object could be learned, a simulator was created to generate synthetic data. Given a set of objects, the simulator generated cues based on predetermined model which represents the probabilities of a cue being produced given the presence of an object. It is this model which the weights attempt to learn using a simple probabilistic relational model consisting of only two classes, object and cue, and one relationship between them.

The simulator was set up using predefined weights as the model in order to demonstrate how the weights used could have been learned in the real data scenarios. The weights generated by the PRM learning are then compared with the true model values as shown in figures 4. The simulation generated 100 runs for each scenario in order to learn the weights.

For exampe in one scenario, there were two objects, a whiteboard and a projector screen, where the related cues, i.e. POINTING, ERASING, WRITING, and their probabilities are represented in 4 in addition to the learned weights.

With an average error of .003 between the true and the learned weights, one can see that the PRM learning technique was able to successfully learn the true model used by the simulator.

5. WEIGHTING OF MULTIPLE CUES

Object and Cues	True Weight Value	Learned Weight Value
Whiteboard		
POINTING	.2	.198
ERASING	.8	.799
WRITING	.8	.798
Projector Screen		
POINTING	.8	.801

Figure 4: Comparison of learned weights to true weights in first scenario.

Although it is true that multiple cues can contribute to the evidence of the presence of an object, it would be too simplistic to believe that the information provided by those cues would all be equally valuable. Thus, it is necessary to have some means of determining how much the evidence should depend on any particular cue, i.e. its weight. Since we are attempting to recognize an object, this weight should necessarily depend on the strength of the association between the cue and the object.

In other words, the weight should be determined by the probability of the object o_i being present given a cue, cue_j, i.e. $P(o_i|cue_j)$.

By increasing or decreasing the value, a greater or lesser dependence on the cue in the calculation of evidence for a particular object label can be enforced. In a previous paper [1], only a simple representation of the cue and objects was utilized in calculating the weights. In this paper, we would like to demonstrate a more complex representation that is more reflective of the real world.

5.1 Probability Model of Objects and Cues

5.1.1 *Probabilistic Relational Models*

There are a number of probabilistic techniques that could have possibly been used to calculate the weights. We chose a Probabilistic Relational Model (PRM) [3] because it can include the relation information of data as an extension of standard Bayesian networks. PRMs can learn associations between classes, attributes within a classes, and attributes related to another class rather than the flat, attribute-value [3] data that Bayes nets must use. Thus, PRM allow for future growth in learning weights that reflect the relationship of the various properties of the cues and objects in determining the weights.

Within the relation model, a schema is defined containing several components: The set of classes,

$$\mathbf{X} = X_1, ..., X_n$$

each of which has a set of attributes,

$$A(X_i) = X_i.a_1, ..., X_i.a_2$$

. The attributes can be either 'fixed' or 'probabilistic'. 'Fixed' attributes are there to identify instances of the class (referred to as *entities*) and thus their value does not change. The value of 'probabilistic' attributes however can vary based on the other attributes of the entity or of related entities. It is this affect that we attempt to model and learn the parameters of.

The second component is the set of relations,

$$\mathbf{R} = R_1, ..., R_m$$

which defines the relationship between two classes. Relationships are significant in that the value of attributes in one class can depend not only on the other attributes of that class, but on the attributes of any related class.

PRMs learn the dependency structure, S, between the attributes of the classes using heuristic structure search and an adapatation of Bayesian model selection [3].

PRMs then describe a probability model over instances of a relational schema. An instance, I, of the relational schema consists of the set of entities of each class,

$$O^{\sigma}(X_i) = e_1, ..., e_p$$

, where the attributes are defined and which relationships exist between them. A skeleton, σ, is when only the fixed attributes of the entities are defined. In our case, an instance would consist of all the objects in a scene, all the cues in the scene, and the association between any of the cues or objects. Some of the attributes however are not easily defined such as the object label and thus it is necessary to determine the probability distribution of its values. A relational skeleton σ is then a partial instance where the probabilistic attributes are undefined.

PRM then defines the distribution of instantiations of attributes as:

$$P(I|\sigma, S, \delta_S) = \prod_{X_i \in \mathbf{X}} \prod_{a_j \in A(X_i)} \prod_{e_k \in O^{\sigma}(X_i)} P(I_{e_k.a_j}|I_{pa(e_k.a_j)}) \tag{1}$$

Given a training set, the parameters δ_S can be learned according to the following equation:

$$l(\delta_S|I, \sigma, S) = logP(I|\sigma, S, \delta_S) \tag{2}$$

$$l(\delta_S|I, \sigma, S) = \sum_{X_i} \sum_{A \in A(X_i)} \left[\sum_{x \in O^{\sigma}(X_i)} logP(I_{x.a}|I_{pa(x.a)}) \right] \tag{3}$$

Standard maximum likelihood estimation can then be applied where δ is chosen in order to maximize l.

5.1.2 Representation of Cues and Objects

We can know describe the model used to represent the relationship between the various cues and objects in order to determine the strength of the association and thus, weight value.

As mentioned previously, we would like to calculate the probability of an object o_i being present given a cue, cue_j, i.e. $P(o_i|cue_j)$.

To begin, we must first define what classes are necessary in this model and with what attributes. An Object class is nessecary with an *obj_id* and *object_label* as its attributes. In addition, we will have a class for each of the cue types, for now we will define six such classes: an ACTIVITY class, SPEECH class, COLOR class, SHAPE class, SOUND class, and VISUAL class. Each with the attributes of *cue_id*, *cue_value*, and *cue_distance*. *cue_value* varies depending on the cue type: For activity, it consists of the set of possible activities that can be recognized, for example

ACTIVITY.*cue_value* = WALK, SIT, ERASING, POINTING

. The exact values used for each class is described in detail in later sections. In determining these values, we attempted to use real exisiting systems to demonstrate the feasibility of using such techniques.

Thus,

$$\mathbf{X} = \text{OBJECT, ACTIVITY, SPEECH, COLOR, ...}$$

and

$$A(\text{OBJECT}) = obj\_id, obj\_label,$$

$$A(\text{CUE}) = cue\_id, cue\_value, cue\_distance,$$

where the identity in each case is a fixed atribute and the rest are probabilistic. By having each object type have its own class, the probabilistic model can reflect whether a particular type is important or not (by including its attributes in the set of parents to an object label) and it can determine whether different cue attributes are important for some cue types while not for others, for example, distance may be important for an activity, while not for color, since any associated color will necessarily have a distance of zero since it is directly on the object).

In terms of relationships, it is possible for an object to be related to another object by either being

$$\mathbf{R} = \text{ON\_TOP\_OF, BELOW, NEAR}$$

. These relationships were chosen initially since a property of many objects seem to depend many time on whether and what type of objects are placed on, below, or near them, for instance a table will often have a number of objects on top of it, a whiteboard often has erasers and markers near it, etc. In addition, it is possible for a cue to be related to an object, a relationship we label ASSOCIATE_WITH.

With this scehma, we can then use the PRM equations to define the distribution of instantiations of the attributes:

$$P(I|\sigma, S, \delta_S) = \prod_{X_i \in \mathbf{X}} \prod_{a_j \in A(X_i)} \prod_{e_k \in O^{\sigma}(X_i)} P(I_{e_k.a_j}|I_{pa(e_k.a_j)}) \tag{4}$$

where in our particular case,

$$X_i \in \text{OBJECT, ACTIVITY, SPEECH, ... ,}$$

$a_j \in cue\_value, cue\_distance$ or obj_label (the fixed attributes are ignored, since it would not make sense to learn the probability of their value), and $e_k \in O^{\sigma}(X_i)$ is the partial instantiation of each entity, i.e. the objects with unspecified object labels. $pa(e_k.a_j)$ are the parents of the j^{th} attribute in the k^{th} entity which can include any of the attributes linked to that object through one of the relationships, R.

Given a training set, the parameters δ_S can be learned according to the equations defined above. A simulator like that described earlier can then be easily produced which would generate cue values given the set of objects presents and a predefined set of parameters.

5.2 Simulation

5.2.1 Cues

ACTIVITY The activity class consists of the set of possible activities that can be recognized. The activity class provides important information as to the function of an object. Since in many cases, while visual attributes may vary, the functional characteristics often do not.

The possible cue values are based off of two different recognition systems the S-SEER system [9] which can recognize Presentation, Nobody Present, and Distant Conversation activities and the system developed by Rybski and Veloso [**?**] which provides Walking, Standing, and Sitting activity recognition. One can imagine adding additional values with input from additional activity recognition systems as is available.

SPEECH Speech consists of words or phrases that could be extracted by a speech recognizer. One can imagine that if the word "table" could be a good indicator for the presence of a table in the scene. There are actually numerous speech recognition systems [10] which can recognize a large portion of the English vocabulary, thus to limit the amount of variables, we focus just on the words pretaining to the objects being recognized.

COLOR For Color, we will use a discretized version of the HSV (Hue Saturation Value) color space where there will be 9 categories of colors: Red, Yellow, Orange, Green, Cyan, Blue, Magenta, Black and White. Each hue, all colors except black and white, can be determined according to the hue color scale which ranges from 0 to 360, and where White and Black is determined by the Value (or brightness), i.e. if the value is above 90 percent, the color will be labeled White. If it is below 10 percent, it is labeled Black.

SHAPE The Shape cue value will be determine by the shape aspect ratio of a tight bounding box around the object/segmented region i.e.

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

We can then discretize the range from 0 to Image_Height to get our categories. For now we will assume an image of size 1085x260 pixels and where we will want 8 categories. We will start with an even division of the range, but future cut off points can be chosen by looking at a histogram of the objects and their aspect ratios, so more categories can be placed in higher density regions for better distinction.

SOUND For sound, a list of possible sounds that could be made by the object is used. These values are determined by the UPC AED/C (Acoustic Event Detection and Classification) System [12], which had the lowest error rate of any other system as tested by [**?**]. The system can detect 12 classs of sounds including Knock, Door Slam, Steps, Keyboard Typing, Phone Ringing, etc.

This list is by no means final. Additional classes can be added as needed or available.

By having each object type have its own class, the probabilistic model can reflect whether a particular type is important or not (by including its attributes in the set of parents to an object label) and it can determine whether different cue attributes are important for some cue types while not for others, for example, distance may be important for an activity, while not for color, since any associated color will necessarily have a distance of zero since it is directly on the object).

5.2.2 *Relationships*

In terms of relationships, it is possible for an object to be related to another object by either being ON_TOP_OF, BELOW, NEAR. These relationships were chosen initially since a property of many objects seem to depend many times on whether and what type of objects are placed on, below, or near them. For instance, a table will often have a number of objects on top of it, a whiteboard often has erasers and markers near it, etc. In addition, it is possible for a cue to be related to an object, a relationship we label ASSOCIATED_WITH. In addition, Cues can be related to other cues with the relationship RELATED_TO. This is more for a future work, as the particular cues and values used here don't seem to depend on each other, but one can image such a association as an Activity such as Talking being closely related with a Speech cue.

5.2.3 *Simulator*

Now that we have a clear definition of possible cues and their values, we can build a simulator to generate scenarios the agent may encounter. Given a set of objects, the simulator generated cues based on predetermined model which represents the probabilities of a cue being produced given the presence of an object. It is this model which the weights attempt to learn.

In order to mimic the imperfection of the real world, noise was added to the data generated by the simulator according to the error rate of the recognition systems that the cues were based on, whenever applicable. In other words, if it is known that a system has .9 accuracy, 10 percent of the time the simulator will produce an incorrect cue value.

6. EXPERIMENT AND RESULTS

Using the cues and relationships defined above, we created various model representations that could represent the real world model of the interaction and depencies between all the entities. In figure 5, an example of one of the more simple models is illustrated.

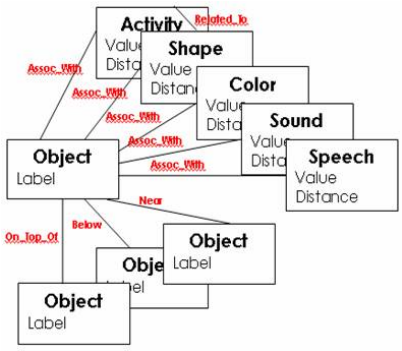

Figure 5: An example of a PRM model of the cues and objects used for determining the weights

Increasingly complex models were then tested, i.e. more links between elements were added, in order to test the accuracy of the weights in more real world situations. Two measures were then used to determine this accuracy: One measured the accuracy of the structure of the model learned, the other measured the accuracy of the parameters learned.

In the first measure, we calculated the number of errors between the true model and the learned structure, where an error is defined as either a missing edge or an extra edge. The amount of errors was then compared according the size of the data set. The result is shown in figure 6

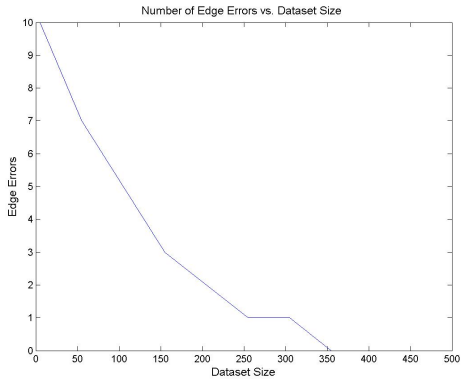

Figure 6: The number of model errors against the size of the data set

In the case of the parameters, the correct model structure was given and only the parameters had to be learned. The error rate (the sum of the absolute difference between the learned and true parameters) was then calculated for data set size ranging from 1 to 450. As can be seen from figure 7, the error rate somewhat plateus after a data set size of 300.

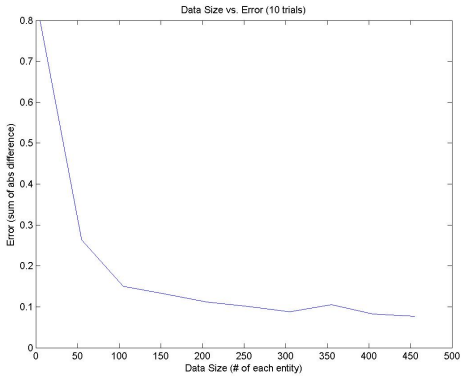

Figure 7: Error rate of learned parameters against data set size.

7. CONCLUSION AND FUTURE WORK

In this paper, we have taken further advantage of the Probabilistic Relational Model framework so that weight values can depend not only on the individual cue values of a cue and object label, but on other properties such as the influence of other cues and objects in varying relationships with each other. In addition, cues were based on actual existing recognition systems in order to demonstrate the feasibility of the system in the real world.

Further goals include applying this framework onto a mobile platform in order to more clearly demonstrate its use on robotic agents.

8. REFERENCES

[1] S. Aboutalib and M. Veloso. Towards using multiple cues for robust object recognition. In *Proceedings of AAMAS'07, the Sixth International Joint Conference on Autonomous Agents and Multi-Agent Systems*, Honolulu, Hawaii, May 2007.

[2] 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.

[3] N. Friedman, L. Getoor, D. Koller, and A. Pfeffer. Learning probabilistic relational models. In *IJCAI*, pages 1300–1309, 1999.

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

[5] 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.

[6] 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.

[7] 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.

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

[9] N. Oliver and E. Horvitz. S-seer: Selective perception in a multimodal office activity recognition system. In *MLMI*, pages 122–135, 2004.

[10] A. I. Rudnicky, A. G. Hauptmann, and K.-F. Lee. Survey of current speech technology. *Commun. ACM*, 37(3):52–57, 1994.

[11] B. Taskar, P. Abbeel, and D. Koller. Discriminative probabilistic models for relational data. In *In Eighteenth Conference on Uncertainty in Artificial Intelligence (UAI02)*, 2002.

[12] A. Temko and C. Nadeu. Classification of meeting-room acoustic events with support vector machines and variable-feature-set clustering. In *IEEE International Conference on Acoustics, Speech, and Signal Processing(ICASSP)*, volume 5, pages 505–508, 2005.

[13] 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.

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