In *Autonomous Robots and Multirobot Systems (ARMS) Workshop* at AAMAS, Saint Paul, USA, May 7, 2013.

Learning Object Models on a Robot using Visual Context and Appearance Cues

Xiang Li[1], Mohan Sridharan[1], and Catie Meador[2]

[1] Department of Computer Science, Texas Tech University, Lubbock, TX 79409, USA, {xiang.li,mohan.sridharan}@ttu.edu

[2] Department of Computer Science, Swarthmore College, Swarthmore, PA 19081, USA, catie.meador@gmail.com

Abstract. Visual object recognition is an important challenge to widespread deployment of mobile robots in real-world domains characterized by partial observability and unforeseen dynamic changes. This paper describes an algorithm that enables robots to use motion cues to identify (and focus on) a set of interesting objects, automatically extracting appearance-based and contextual cues from a small number of images to efficiently learn representative models of these objects. Object models learned from relevant image regions consist of: (a) relative spatial arrangement of gradient features; (b) graph-based models of neighborhoods of gradient features; (c) parts-based models of image segments; (d) color distribution statistics; and (e) probabilistic models of local context. An energy minimization algorithm and a generative model of information fusion use the learned models to reliably and efficiently recognize these objects in novel scenes. All algorithms are evaluated on wheeled robots in indoor and outdoor domains.

Keywords: Visual learning, Object Recognition, Mobile robots

1 Introduction

Sophisticated algorithms have been developed for representing and recognizing objects using different visual cues [8, 10, 15]. The computational complexity of these algorithms, and the sensitivity of visual inputs to changes in object configurations and environmental factors, make it difficult for robots to reliably and efficiently model and recognize objects. Existing algorithms typically require considerable training, human supervision or prior knowledge to learn good object models, However, robot application domains are typically characterized by partial observability and unforeseen dynamic changes, making it a challenge to obtain accurate domain knowledge, human feedback or many training examples of relevant domain objects. Learning object models and object recognition therefore continue to be open problems in robotics.

The above-mentioned challenges are partially offset by some observations. First, many objects possess unique characteristics (e.g., color and shape) and distinguishable motion patterns, although these characteristics and patterns are not known in advance and may change over time. Second, images encode information about objects in the form of complementary appearance-based and contextual cues. Third, robots typically do not need to model all domain objects; if robots automatically learn and revise the

Fig. 1. Local, global and temporal cues extracted from pixels within the yellow boundary represent appearance, while mixture models and relative positions (e.g., "on" and "under") of regions within the red rectangle (outside the yellow polygon) represent context.

domain map, many tasks require robots to pay attention to objects that move. This paper describes an algorithm that exploits these observations to make the following key contributions:

- Investigates learning of object models from a small $(3-8)$ number of images. Robots learn the domain map and consider objects that move to be *interesting*, efficiently identifying corresponding image regions using motion cues.
- Exploits complementary properties of appearance-based and contextual visual cues to efficiently learn representative models of these interesting objects from relevant image regions.
- Uses learned object models in generative models of information fusion and energy minimization algorithms for reliable and efficient recognition of stationary and moving objects in novel scenes.

These contributions build on our prior work on efficiently learning object models using visual cues [13], promoting automatic and incremental learning on mobile robots. Although visual features included in our algorithm have been used in vision research, our representation fully utilizes them by learning: spatial arrangements of gradient features, graph-based models of neighborhoods of gradient features, parts-based models of image segments, color distributions, and local context models. Experiments show reliable and efficient learning and recognition on robots in indoor and outdoor domains.

The remainder of this paper is organized as follows. We first discuss related work in Section 2, and describe our approach in Section 3. Experimental results are presented in Section 4, followed by conclusions in Section 5.

2 Related Work

Many algorithms have been developed for representing and recognizing objects using scale, rotation and affine-invariant image gradient features [2, 15], appearance and shape features [5], hierarchical decompositions of parts [6] and visual code-books [16]. However, these algorithms require extensive training or human supervision, and are computationally expensive for use on mobile robots.

Context is an important cue for vision-based object recognition by humans and machines [18, 20]. Object recognition algorithms have modeled global context at the level of the entire image [26, 29], and learned models of local context from image regions surrounding the object of interest [7, 25]. Recent research has focused on extracting adaptive (and different) contextual cues from images [10, 14]. Research shows that the

importance of contextual cues varies with the quality of appearance information [20]. Research also shows that motion cues (especially relative motion) can be used for visual recognition [27] and for augmenting the recognition capability provided by other visual (or non-visual) cues [17].

Many algorithms are being developed for unsupervised learning of object models. Researchers have used the stability of some features extracted from sensor inputs for initial unsupervised classification of images [22]. Existing algorithms also enable unsupervised learning of hierarchical spatial structures from images using rule-based models [19]. Using multiple visual cues and interactions with objects, researchers are developing algorithms for learning spatial relationships between objects [23], and for automatic discovery of groups of related objects [11]. Joint representations of perception-grasping systems have been learned using reinforcement learning and hierarchical Markov models [21], and a robot's interactions with objects have been used to distinguish objects from the background [24]. However, these algorithms fail to fully exploit different visual cues, and require accurate domain knowledge and considerable training data. Our prior research has shown that regions of interest can be automatically identified in images using motion cues, modeling objects using different visual cues [12, 13]. This paper further investigates learning of object models from a small number of images, building rich representations that fully exploit complementary properties of local, global, temporal and contextual visual cues.

3 Proposed Approach

In the proposed approach, robots learn the domain map using range data and consider objects that move to be interesting. Based on the observation that characteristic features of an object have similar relative motion between consecutive images, robots track local gradient features in short sequences ($3 - 8$ images), identifying regions of interest (ROIs) corresponding to moving objects by clustering features with similar relative motion. One underlying assumption (that works well in practice) is that object motion has a non-trivial linear component. Object models are learned from candidate ROIs using complementary properties of local, global, temporal and contextual cues extracted from the ROIs. This section describes the components of the object model, and the use of learned object models for reliable and efficient object recognition in novel scenes.

3.1 Object Model Learning

Consider the process of learning an object model from a specific ROI. As shown in Figure 2, the corresponding object model includes appearance-based and contextual cues: (1) gradient features and their relative spatial arrangement; (2) connection potentials between gradient features and a graph-based model of neighboring potentials; (3) image segments and a parts-based model of their spatial arrangement; (4) color distributions and second-order image statistics; and (5) Gaussian mixture models and relative positions of image segments neighboring the ROI.

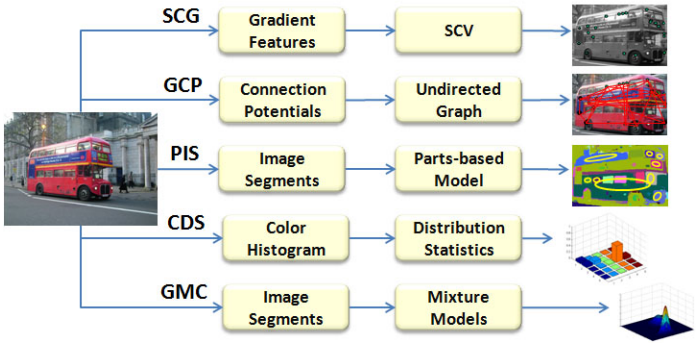

Fig. 2. Learned model uses contextual and appearance-based cues to characterize objects.

Spatial Coherence of Gradient Features (SCG): Gradient features extracted from the image ROI may not be unique. Our prior work used a *spatial coherence vector* (SCV) to model the relative spatial arrangement of gradient features, which is difficult to duplicate [12]. The SCV is computed along x and y axes for each of the N gradient features in the ROI:

$$SCV_{x,i} = \{d^x_{i,1}, d^x_{i,2}, \ldots, d^x_{i,N}\} \quad (1)$$
$$SCV_{y,i} = \{d^y_{i,1}, d^y_{i,2}, \ldots, d^y_{i,N}\}$$

where $d^x_{i,j}$ and $d^y_{i,j}$ are the relative positions of feature i w.r.t feature j along the x and y axes respectively, e.g., if x_i and x_j are the x-coordinates of feature i and j in the image, $d^x_{i,j} = 1, 0$ or -1 for $x_i >, =$ or $< x_j$ respectively; $d^y_{i,j}$ is defined similarly. The object model hence extracts N gradient features from the ROI (each feature is a 128D vector) and a $2(N-1)$-dimensional SCV for each feature.

Graph-Based Model of Connection Potentials (GCP): The second component of the object model captures the relationships between neighboring gradient features in the ROI. For any two gradient features, the *connection potential* is defined as the distribution of pixels on the line joining the features. The distance between the features is normalized and pixel values are collected in a histogram of 100 bins, which is smoothed along each dimension:

$$C^{new}_n = \alpha C_n + (1-\alpha) C_{n-1} \quad (2)$$

where the smoothed value in n^{th} bin is a function of the value in previous bin (C_{n-1}) and raw value in the bin (C_n). The effect of raw data is controlled by α, while the coarse representation (100 bins) provides computational efficiency. The N gradient features in the ROI are also sorted based on distance from the center of the ROI: $\{d_1, ..., d_{k-1}, d_k, d_{k+1}, ...d_N\}, \forall i < j,\ d_i < d_j$. The local neighborhood of each feature includes the four closest neighbors. The object model includes the connection potentials and a undirected graph [9] of local neighborhoods of connection potentials.

Parts-based Models of Image Segments (PIS): The third component uses a graph-based segmentation algorithm [4] to extract segments from the ROI such that RGB values within a segment are similar and significantly different from pixels in neighboring segments. Valid segments are modeled as 2D Gaussians that represent spatial locations in the ROI: $\mathcal{N}(\mu_k, \Sigma_k), k = 1, ..., M$ and constitute "parts" of the object. Each pixel n in the ROI is assigned membership in one of M parts based on Gaussian density functions of the parts: $\mathrm{argmax}_j\, p(n\,|\mu_j, \Sigma_j\,)$. Then, each pixel's similarity with pixels in the same part and dissimilarity with pixels in neighboring parts are computed, weighted by the probability that these pixels belong to the same part or different parts. Similarity and dissimilarity measures for each part ($PartSimM_k$, $PartDiffM_k$) are defined as the logarithm of sum of contributions of all pixels in that part. To capture local variations in positions of parts, the envelope around the extracted parts is displaced a few times and the corresponding values of $PartSimM$ and $PartDiffM$ are modeled as gamma (Γ) distributions for each part. The object model includes image segments, parts-based model and these measures of similarity and dissimilarity.

Color Distribution Statistics (CDS): The fourth component of the object model captures color information, based on our prior research [12, 13]. The ROI's pixels are used to learn normalized histograms (i.e., probability density functions) in the HSV color space. Each pdf is learned in (h, v) with ten bins in each dimension. Since color distributions are not a stable or unique representation of an object, second order statistics are computed in the form of distances between every pair of pdfs, using the Jensen-Shannon (JS) measure [3]. The fourth component consists of the color-space pdfs and incrementally-learned distribution of distances between the pdfs.

Gaussian Mixture Model of Context (GMC): The fifth component models the object's *local context* using image segments (extracted for PIS above) that share a boundary with the ROI. These segments lie within the red rectangle but outside the yellow boundary in Figure 1. The pixels in each such segment are used to learn a 2D Gaussian in the normalized HSV color space (using h, v). The relative spatial arrangement of each segment with respect to the ROI is used to assign labels "on", "under" and "beside" to the segment; image segments can have more than one label. Image segments that have the same label are used to learn a Gaussian mixture model (GMM), e.g., each of the K 2D Gaussians with label "on" is assigned a mixing factor π_k that is the ratio of number of pixels in the corresponding segment divided by the number of pixels in all K segments. Each GMM is also assigned a weight that is the ratio of number of pixels in segments with the corresponding label to the number of pixels in all segments used to model context. The object model includes GMMs and their relative positions and sizes with respect to the center and size of the ROI.

3.2 Information Fusion for Recognition

The learned models are used for object recognition in images of novel scenes, *irrespective of whether the objects are stationary or moving*. Energy minimization is used to iteratively select ROIs in test images, and generative models merge evidence from

components of learned models to compute probability of occurrence of objects in the ROIs. The analysis of a specific test image ROI is described first.

SCG-Based Matching: The SCVs of gradient features in a learned model and the matched features in the test image ROI are used to obtain p_{scg}, the corresponding object's probability of occurrence:

$$p_{scg} = \frac{x_{correct} + y_{correct}}{2 * M}, \quad p_{scg} \in [0, 1] \tag{3}$$
$$x_{correct} = \sum_{m=1}^{M} \frac{Nx_{m\_correct}}{N-1}, \; y_{correct} = \sum_{m=1}^{M} \frac{Ny_{m\_correct}}{N-1}$$

where $Nx_{m\_correct}$ and $Ny_{m\_correct}$ are the number of values in the ROI's SCV that match the learned model's SCV along x and y axes respectively; M and N are the number of gradient features in the learned model and ROI respectively. The probability distribution of occurrence of learned objects in the ROI is obtained by considering all object models.

GCP-Based Matching: The probability of occurrence of a learned object (in the ROI) is also computed by comparing the neighborhood of connection potentials in the learned model to the neighborhood of connection potentials between matched ROI features. The similarity between two connection potentials i and j is:

$$con(i,j) = \sum_{n=1}^{100} f(C_n^i, C_n^j), \;\; f(a,b) = \begin{cases} 1 & |a-b| > \beta \\ 0 & otherwise \end{cases}$$

where parameter β is used to identify significant changes in entries of connection potentials. The probability of occurrence of the learned object is:

$$p_{gcp} = \frac{1}{Z} \sum_{k \in \{1,\ldots,M\}} \sum_{i \in N_k, j \in N_{k_m}} con(i,j) \tag{4}$$

where M gradient features in the object model match features in the ROI, N_{k_m} is the connected neighborhood of feature k_m in the object model and N_k is the connected neighborhood of the corresponding matched feature k in the ROI, and Z is a normalizer. A similar computation with other object models provides the probability distribution of occurrence of learned objects in the ROI.

PIS-based Matching: To compute the probability of occurrence of a learned object using parts-based models, different relative arrangements of the learned model's parts are compared with pixels in the test image ROI. For pixels in the overlapping regions (for any arrangement), the similarity of pixels that lie within a learned model part and the dissimilarity of pixels that lie in neighboring parts are computed. The learned Γ distributions of these measures (for each part) help compute likelihood of this arrangement:

$$p_{pis} = \sum_j \{w_j \cdot f(PartSimM_j) \cdot f(PartDiffM_j)\}$$
$$f(x_j) = \Gamma\Big(|\overline{x_j} - x_j| - (k-1)\theta, k, \theta\Big) \tag{5}$$

where, for the learned object's j^{th} part, $(k-1)\theta$ is the stationary point of the learned Γ pdf, x_j is the similarity or dissimilarity computed using ROI pixels in the part ($PartSimM_j$, $PartDiffM_j$), and $\overline{x_j}$ is the mean of the Γ pdf. The match probability of this arrangement is the sum of product of these measures for each part, weighted (w_j) by the ratio of number of ROI pixels in a part divided by number of ROI pixels in all parts of object model. The arrangement that maximizes p_{pis} is chosen. A similar computation with other object models provides the probability distribution of occurrence of learned objects in the ROI.

CDS-Based Matching: To compute the probability of occurrence of the a learned object in the ROI (p_{cds}), the average distance d_{avg} is computed between the ROI's color space pdf and the pdfs in the learned object model, using the JS measure. A comparison with the expected (Gaussian) distribution of distances (for the object model) provides the value of p_{cds}. Performing this computation with all learned models provides the probability distribution of occurrence of the learned objects in the ROI. When second-order statistics of object models are being learned, relative values of average distances between the ROI's color space pdf and learned pdfs of object models are used to obtain the probability of occurrence of the learned objects in the ROI.

GMC-Based Matching: The probability of occurrence of a learned object in the ROI is also computed by comparing local context information. Each GMM in the learned model (labels: *on, under, beside*) is scaled and positioned with respect to the ROI. A matching score is computed using each GMM, considering the pixels around the convex boundary of the ROI that fall within the spatial scope of the GMM (N_{lbc}). The probability of occurrence of learned object is then the weighted sum of individual scores:

$$p_{gmc} = \sum_{lbc \in \{on, under, beside\}} w_{lbc} \cdot \Gamma\Big(f(\boldsymbol{x}_{lbc}), k, \theta\Big)$$
$$f(\boldsymbol{x}_{lbc}) = \frac{1}{N_{lbc}} \sum_{l=1}^{N_{lbc}} \sum_{j=1}^{N_{lbc}^{gmm}} \pi_j \, e^{-\frac{1}{2}(\boldsymbol{x}_l - \boldsymbol{\mu}_j)^T \Sigma_j^{-1} (\boldsymbol{x}_l - \boldsymbol{\mu}_j)} \tag{6}$$

where N_{lbc}^{gmm} is the number of 2D Gaussians in the Gaussian mixture model with label $lbc \in \{on, under, beside\}$. Each ROI pixel $\boldsymbol{x}$ is a 2D vector in the normalized (h, v) color space. The value of $f(\boldsymbol{x}_{lbc})$ is scaled by a Γ distribution and weighted (w_{lbc}) by the ratio of number of pixels that fall within the corresponding GMM divided by number of pixels that fall within all GMMs in the learned model. Values of π_j, $\boldsymbol{\mu}_j$ and Σ_j are obtained from the learned model. A similar computation with other object models provides the probability distribution of occurrence of learned objects in the ROI.

Information Fusion: Consider: (a) the identification of ROIs in test images; and (b) fusion of evidence from components of learned object models regarding the presence of corresponding objects in these ROIs. For ease of explanation, assume that any ROI contains no more than one learned object—*the algorithm can detect multiple objects in an image or ROI*. If a test image sequence contains a moving object, the corresponding ROI is identified by (as during learning) tracking and clustering gradient features in the sequence; the probability of occurrence of a learned object in this ROI is then the product of probabilities provided by components of the object model.

When test images are snapshots of objects in different scenes, ROIs are identified by matching gradient features in test images with gradient features in the learned object models. For instance, to compute the probability of occurrence of the i^{th} learned object in a test image, the K nearest neighbors are found in the test image for each of the M local gradient features in the learned model. Each of the (at most) $K * M$ features in the test image is considered for further analysis. Candidate ROIs are created by iteratively selecting M matched features in the test image using the iterated conditional modes (ICM) energy minimization algorithm [28]. Since this algorithm can be sensitive to the choice of initial estimates in high-dimensional spaces, the nearest neighbors of the learned object's gradient features are used as the initial ROI estimate. Each ROI is analyzed using generative models of information fusion. For a set of M matched (test image) features, the probability of occurrence of the i^{th} learned object (p_{O_i}) considers the evidence provided by each feature:

$$\begin{aligned} p_{O_i} &= \prod_{j \in \{1,...,M\}} p\Big(g_j | O_i, \{g_n | n = 1, ..., M, n \neq j\}\Big) \\ &= \prod_{j \in \{1,...,M\}} p(g_j | O_i) \end{aligned} \tag{7}$$

where $\{g_n | n = 1, ..., M, n \neq j\}$ is the subset of M matched test image gradient features excluding the j^{th} feature under consideration. Since $\{g_n | n = 1, ..., M, n \neq j\}$ is always given, the term is ignored in the following equations. The probability that each matched feature comes from learned object O_i is formulated as a generative model over the individual components of the object model:

$$p(g_j | O_i) = \sum_{Lbg_j \in \{fg, bg\}} p(g_j | Lb_{g_j}, O_i) \cdot p(Lb_{g_j} | O_i) \tag{8}$$

where $Lb_{g_j} \in \{fg, bg\}$ indicates whether the j^{th} feature belongs to the foreground, i.e., it is part of the target object, or to the background, i.e., it is not part of the target.

When specific labels (fg, bg) are assigned to candidate matched features, the ROI is defined by the convex hull [1] i.e., minimal convex set containing the foreground features. The idea is to identify candidate features based on feature matching and energy minimization, and use generative models to consider multiple local arrangements to refine the initial choice. Equation 8 is decomposed using the independence relationships

in the joint probability distribution:

$$p(g_j|O_i) = \sum_{Lbg_j \in \{fg,bg\}} p(g_j|Lb_{g_j}, O_i) \cdot p(Lb_{g_j}|O_i) \quad (9)$$

$$= \sum_{Lb_{g_j} \in \{fg,bg\}} p(g_j|Lb_{g_j}, scg_{O_i}) \cdot p(g_j|Lb_{g_j}, gcp_{O_i}) \cdot$$

$$p(Lb_{g_j}|pis_{O_i}) \cdot p(Lb_{g_j}|cds_{O_i}) \cdot p(Lb_{g_j}|gmc_{O_i})$$

The underlying observation is that parts-based models (PIS), color statistics (CDS) and context-based models (GMC) capture visual cues that are more global and are not evaluated based on relative arrangements of local cues. These models can hence be used to evaluate the relative likelihoods of (foreground or background) labels for the feature under consideration. The other components of the object model, i.e., those based on gradient features (SCG) and connection potentials (GCP) are used to evaluate the probability of occurrence of the gradient features given the specific labels. The individual probabilities in Equation 9 are computed using Equations 3-6 and the underlying independence assumptions work well in practice. The ROI (among candidates generated by matching gradient features) that maximizes Equation 9 and thus Equation 7 is the best estimate of the corresponding object's location in the test image.

Finally, the net probability distribution of occurrence of the L learned objects in a test image ROI is normalized: $\overline{p}_{O_i}, i \in [1, L]$. This distribution is used to recognize objects and detect novel objects when none of the learned objects has a match probability significantly larger than others. The robot concurrently learns object models and recognizes objects while revising the domain map and planning navigation. Initially, if candidate ROIs are identified corresponding to moving objects, the robot learns object models. Learned models are used to recognize these objects in subsequent images and to identify new objects. Furthermore, image ROIs corresponding to recognized objects are used to incrementally revise existing object models.

4 Experimental Setup and Results

This section describes the robot test platform and the experimental results of evaluating the proposed algorithms.

4.1 Test Platform

The test platform is a wheeled robot from Videre Design equipped with a stereo camera, monocular camera, laser range finder and pan-tilt unit on a $40cm \times 41cm \times 15cm$ base. The experiments used one of the cameras of the stereo unit that provides 640×480 images. Input from the laser range finder is used to learn the domain map. Although the robot has Wi-Fi capability, all experiments were performed on-board using a 2GHz processor and 1GB RAM. Trials were conducted in indoor offices and outdoor settings.

4.2 Experimental Results

We experimentally evaluated the robot's ability to: (a) learn representative object models from a small number of images by exploiting complementary properties of different visual cues; and (b) reliably and efficiently recognize objects in novel scenes. In comparison with our prior work [13], this paper evaluates additional components (in the object model) and includes a much more complex database of images. Twenty object categories were used in the experimental trials, e.g., *car, human, book, box, robot* and *bus*; Figure 4 shows some examples. Separate models were learned for different objects in a category, e.g., different boxes, books or humans, resulting in 60 subcategories. Objects were considered in complex backgrounds that made learning and recognition challenging. During experiments, some objects (e.g., humans and cars) moved on their own, while some (e.g., boxes) were moved on trolleys.

It is a challenge to obtain an image database of objects with well-defined motion. Experiments used ≈ 2000 images, including short sequences and individual snapshots, ≈ 700 of which were captured by the robot. To establish applicability to different domains, ≈ 1300 images of motorbikes, buses, some cars and airplanes were chosen from the *Pascal VOC2006* and *Caltech-256* benchmark datasets. The benchmark datasets include ROIs for objects in the images—the robot selected suitable ROIs when any of these images were used for learning object models, and automatically learned context models (GMC) from image segments neighboring the ROIs. To make learning challenging, each object model is learned from $\approx 3-8$ images, with ≈ 250 images used for learning all object models; remaining images are used for evaluation. Test images consist of short sequences of objects in motion and individual snapshots of objects in indoor and outdoor scenes. The robot processed $3-5$ frames/second to identify moving objects, learn models and recognize objects while performing other actions. The images used for learning and recognition were chosen randomly (in repeated trials) to obtain the results below.

	Box	Car	Human	Robot	Book	Airplane	Bus	Bike	Firetruck	Firehydrant
Box	**0.941**	0	0.017	0.025	0	0	0	0	0	0.017
Car	0.010	**0.917**	0	0.021	0	0	0	0.042	0	0.010
Human	0.080	0.024	**0.820**	0.060	0.016	0	0	0	0	0
Robot	0.027	0	0.042	**0.899**	0.027	0	0	0.005	0	0
Book	0.016	0	0	0.042	**0.942**	0	0	0	0	0
Airplane	0.029	0.051	0	0.023	0.009	**0.888**	0	0	0	0
Bus	0	0	0	0	0	0	**0.856**	0.036	0.108	0
Bike	0	0.073	0	0.010	0.016	0	0.062	**0.839**	0	0
Firetruck	0	0.032	0	0	0	0	0.080	0.016	**0.872**	0
Firehydrant	0.029	0.029	0	0	0	0	0	0	0.058	**0.884**

Table 1. Recognition accuracy averaged over different models (i.e., subcategories) in a subset of (ten) object categories.

The average classification accuracy over all 60 subcategories in 20 object categories is: 0.8860 ± 0.0432, which is very appealing given the small number of images used for

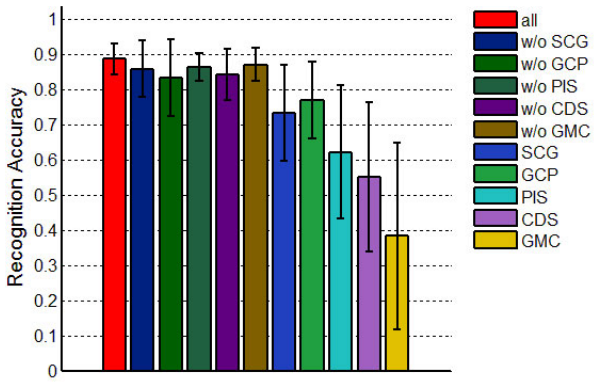

Fig. 3. Our algorithm provides higher accuracy than any individual component and any four of the components; *results are statistically significant*.

learning. Table 1 shows classification accuracy for a subset of (ten) object categories, averaged over object models in each category; off-diagonal terms represent errors. Accurate classification requires an object to be matched to the correct model; matching an object in *car-class1* to model *car-class2* is an error. One reason for errors is the learning of object models with non-unique features, e.g., long shots of humans cause features to be extracted from clothes, resulting in non-unique object models and lower recognition accuracy. Some errors correspond to an insufficient number of features in the test image being matched with the learned object models due to occlusions, motion blur or a large difference in viewpoint. Incremental revision of object models further improves recognition accuracy. Some errors also occur when test image ROIs are assigned the label of the object model with the maximum match probability, even if this value is similar to match probabilities of other objects. These errors are eliminated by requiring that the maximum match probability be substantially higher than match probabilities of other object classes. Furthermore, errors are less frequent in image sequences of objects in motion because correctly identifying the ROI enables some subset of components to provide high match probabilities for the appropriate object.

Our algorithm and existing vision algorithms have disparate objectives; our algorithm efficiently learns models of a subset of objects using $3 - 8$ images (each), while existing algorithms typically use a large database (e.g., ≥ 1000 images) for training (or learning) models of each object and focus on modeling a large number of objects. Although it is challenge to find a common frame of reference, the following comparisons were conducted.

When we increase the number of images used of learning object models, the recognition accuracy increases, e.g., 0.90 ± 0.05 with 400 images (total) for learning, and slowly approaches reported accuracies of state of the art algorithms on the benchmark datasets. However, these algorithms are much more (computationally) expensive for learning or recognition. Furthermore, it is difficult for these algorithms to learn good models from a much smaller number of images because they do not fully exploit the complementary properties of (and dependencies between) different cues.

Fig. 4. Robot recognizes objects from different categories, multiple objects and multiple instances of an object in cluttered backgrounds. Last column shows an incorrect envelope (top) and an incorrect classification due to occlusion (bottom).

Next, Figure 3 compares the average recognition accuracy of our algorithm with that of each component and any four of the components. Note that each component uses popular visual cues, although our representation better exploits their benefits. Any individual component cannot provide high accuracy and there is large variance, especially with components that primarily use color. At the same time, each component does contribute to the overall accuracy, and variance is larger when spatial arrangements of local features are not considered. The combination of all components provides the highest accuracy by learning models that exploit complementary properties of different visual cues. Figure 4 shows examples of the robot accurately recognizing objects from different categories, and recognizing multiple objects or multiple instances of objects in different scenes. The last column of Figure 4 also shows an instance where (a) top: the object boundary is incorrect (although object label is correct) due to incorrectly matched features; and (b) bottom: occlusion leads to incorrect classification, e.g., object of *bus-class1* matched with *car-class2*. We hypothesize that including a component in the object model that matches partial shapes will minimize these errors, and the computational efficiency of our algorithm supports the addition of such components.

5 Conclusions and Future Work

This paper described an algorithm for mobile robots to identify interesting objects based on motion cues, automatically learning representative models of these objects using appearance-based and contextual visual cues from a small number of images. The learned models enable reliable and efficient object recognition in novel indoor and outdoor scenes. Future research will consider image sequences with many moving objects and further improve computational efficiency using sampling and efficient energy minimization algorithms. Furthermore, we will investigate the inclusion of other components (e.g., shape) and design algorithms for automatically determining the most informative components to represent each object. The long-term goal is to enable robots to automatically and incrementally learn object models with minimal human supervision in real-world domains.

References

1. C.B. Barber, D.P. Dobkin, and H.T. Huhdanpaa. The quickhull algorithm for convex hulls. *ACM Trans. on Mathematical Software*, 22:469–483, Dec 1996.

2. M. Calonder, V. Lepetit, C.Strecha, and P. Fua. BRIEF: Binary Robust Independent Elementary Features. In *European Conference on Computer Vision*, 2010.
3. Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory, 2nd Edition*. Wiley-Interscience, 2006.
4. P. F. Felzenszwalb and D. P. Huttenlocher. Efficient Graph-Based Image Segmentation. *International Journal of Computer Vision*, 59(2):167–181, 2004.
5. R. Fergus, P. Perona, and A. Zisserman. Object Class Recognition by Unsupervised Scale-Invariant Learning. In *International Conference on Computer Vision and Pattern Recognition*, pages 264–271, 2003.
6. S. Fidler, M. Boben, and A. Leonardis. Similarity-based Cross-Layered Hierarchical Representation for Object Categorization. In *International Conference on Computer Vision and Pattern Recognition*, 2008.
7. M. Fink and P. Perona. Mutual Boosting for Contextual Inference. In *Neural Information Processing Systems*, 2003.
8. H. Kang, M. Hebert, and T. Kanade. Discovering Object Instances from Scenes of Daily Living. In *International Conference on Computer Vision*, 2011.
9. D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2010.
10. Congcong Li, Devi Parikh, and Tsuhan Chen. Extracting Adaptive Contextual Cues from Unlabeled Regions. In *International Conference on Computer Vision*, 2011.
11. Congcong Li, Devi Parikh, and Tsuhan Chen. Automatic Discovery of Groups of Objects for Scene Understanding. In *IEEE Conference on Computer Vision and Pattern Recognition*, Providence, USA, June 16-21, 2012.
12. Xiang Li and Mohan Sridharan. Autonomous Learning of Object Models on a Mobile Robot using Visual Cues. In *International Conference on Robotics and Automation*, 2011.
13. Xiang Li and Mohan Sridharan. Vision-based Autonomous Learning of Object Models on a Mobile Robot. In *AAMAS Workshop on Autonomous Robots and Multirobot Systems*, June 5 2012.
14. R. Luo, S. Piao, and H. Min. Simultaneous Place and Object Recognition with Mobile Robot using Pose Encoded Contextual Information. In *International Conference on Robotics and Automation*, 2011.
15. K. Mikolajczyk and C. Schmid. Scale and Affine Invariant Interest Point Detectors. *International Journal of Computer Vision*, 60(1):63–86, 2004.
16. F. Moosmann, B. Triggs, and F. Jurie. Fast Discriminative Visual Codebooks using Randomized Clustering Forests. In *Neural Information Processing Systems*, 2006.
17. Aniket Murarka, Mohan Sridharan, and Benjamin Kuipers. Detecting Obstacles and Drop-offs using Stereo and Motion Cues for Safe Local Motion. In *International Conference on Intelligent Robots and Systems*, 2008.
18. A. Oliva and A. Torralba. Building the Gist of a Scene: The Role of Global Image Features in Recognition. In *Visual Perception, Progress in Brain Research*, 2006.
19. D. Parikh, C. L. Zitnick, and T. Chen. Unsupervised Learning of Hierarchical Spatial Structures in Images. In *Computer Vision and Pattern Recognition*, 2009.
20. D. Parikh, L. Zitnick, and T. Chen. Exploring Tiny Images: The Roles of Appearance and Contextual Information for Machine and Human Object Recognition. *Pattern Analysis and Machine Intelligence*, 34:1978–1991, 2012.
21. J. Piater, S. Jodogne, R. Detry, D. Kraft, N. Kruger, O. Kroemer, and J. Peters. Learning Visual Representations for Perception-Action Systems. *International Journal of Robotics Research*, 30:294–307, 2011.
22. K. Roman, N. Juan, N. Eduardo, and D. Bertrand. Track-based Self-supervised Classification of Dynamic Obstacles. *Autonomous Robots*, 29(2):219–233, 2010.

23. B. Rosman and S. Ramamoorthy. Learning Spatial Relationships Between Objects. *International Journal of Robotics Research, Semantic Perception for Robots in Indoor Environments, Part 2*, 30(11):1328–1342, September 2011.
24. D. Schiebener, A. Ude, J. Morimotot, T. Asfour, and R. Dillmann. Segmentation and Learning of Unknown Objects through Physical Interaction. In *International Conference on Humanoid Robots*, pages 500–506, 2011.
25. J. Shotton, J. M. Winn, C. Rother, and A. Criminisi. TextonBoost: Joint Appearance, Shape and Context Modeling for Multi-class Object Recognition and Segmentation. In *European Conference on Computer Vision*, pages 1–15, 2006.
26. B. Siddiquie and A. Gupta. Beyond Active Noun Tagging: Modeling Contextual Interactions for Multi-class Active Learning. In *Computer Vision and Pattern Recognition*, pages 2979–2986, 2010.
27. Andrew Stein and Martial Hebert. Combining Local Appearance and Motion Cues for Occlusion Boundary Detection. In *The British Machine Vision Conference (BMVC)*, 2007.
28. R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother. A Comparative Study of Energy Minimization Methods for Markov Random Fields with Smoothness-Based Priors. *Pattern Analysis and Machine Intelligence*, 30(6), 2008.
29. A. Torralba and P. Sinha. Statistical Context Priming for Object Detection. In *International Conference on Computer Vision*, pages 763–770, 2001.