In *Autonomous Robots and Multirobot Systems Workshop at AAMAS* (ARMS 12), Valencia, Spain, June 5 2012.

Vision-based Autonomous Learning of Object Models on a Mobile Robot

Xiang Li and Mohan Sridharan

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

Abstract. This paper presents a novel approach that enables a mobile robot to autonomously learn object models using local, global and temporal visual cues. Learning is triggered by motion cues—interesting image regions are identified by tracking and clustering salient (local) image gradient features across a sequence of images. Object models learned from these candidate image regions consist of: (1) gradient features and their relative spatial arrangement; (2) neighborhood relationships of connection potentials between the gradient features; (3) parts-based representation of image segments extracted from the region; and (4) color distribution statistics. Belief revision and energy minimization algorithms used the learned object models to reliably and efficiently recognize the desired objects in novel scenes. All algorithms are implemented and evaluated on a mobile robot platform deployed in indoor and outdoor domains.

Keywords: Visual learning, graphical models, autonomous mobile robots.

1 Introduction

Object recognition continues to be an open challenge in the field of robotics and computer vision despite the development of sophisticated object recognition algorithms that use a variety of visual cues [7, 8, 13–16, 24]. Real-world application domains are characterized by unforeseen dynamic changes and reliable operation in such domains requires the robot to autonomously learn and revise models of domain objects. However, the sensitivity of visual inputs to changes in environmental factors and the computational complexity of visual input processing algorithms make vision-based autonomous operation a formidable challenge. This paper describes a novel approach that enables a mobile robot to autonomously learn models for environmental objects using local, global and temporal visual cues. The approach draws inspiration from nature, where a chameleon that has camouflaged itself by taking on the color of the background can still be detected when it starts moving. We hypothesize that once a map of the world (with stationary objects and obstacles) has been learned, objects that can move are interesting and need to be tracked by the robot. The proposed approach therefore identifies interesting image regions corresponding to candidate objects using temporal visual cues, i.e., by tracking local gradient features over successive images. Each candidate object is then characterized by image gradients, connections between gradient features, image segments and color distributions extracted from the corresponding image region.

The learned models are augmented with an additional layer that models the relative spatial arrangement of gradient features, neighborhood relationships of feature connections, parts-based arrangement of image segments and second-order statistics of color distributions. The layered object model thus utilizes the complementary properties of local, global and temporal visual cues to build robust models that characterize environmental objects. A belief revision strategy uses the learned models to detect objects in subsequent frames, revising the learned models and leading to more accurate object recognition. All algorithms are evaluated on mobile robots in real-world domains.

2 Related Work

Sophisticated algorithm have been developed in computer vision research to characterize and recognize object using scale, rotation and affine-invariant gradient features [4, 13, 16, 18]. For instance, Mikolajczyk and Schmid [16] developed image gradient features invariant to affine transforms to characterize and recognize objects in images. Lowe [13] developed the scale-invariant feature transform (SIFT) that used local image gradient features to characterize objects of interest. Matas et al. [15] represented objects by using an affine-invariant set of extremal regions, called the maximally stable extremal regions (MSER). However, algorithms that use gradient features are not well-suited for representing objects with patterned or texture-less surfaces, and are computationally expensive. Other algorithms for object recognition characterize objects using models of appearance, shape and size [7], as a hierarchical decomposition of parts [8], or as a contour that identifies object boundaries tracks non-rigid shapes [30]. Researchers have also developed models based on the human visual cortical mechanisms [25] and used visual code-books to represent a wide range of objects [17]. However, these algorithms typically require extensive manual supervision during training and are computationally expensive.

Computer vision algorithms draw upon mathematical principles such as energy minimization, graph theory and belief propagation in graphical models [3, 29, 11]. For instance, Guo et al. [10] developed an adaptive non-planar road detection and tracking algorithm that uses a Markov random field (MRF) for belief propagation. Kolmogorov et al. [11] used MRF models to build inference layers based on color, contrast and stereo matching, while Arbelaez et al. [1] used the normalized energy of the established match between images as a measure of goodness of fit. More recently, Porway and Zhu [22] developed a Markov Chain Monte Carlo (MCMC) inference algorithm that outperforms existing inference algorithms in tasks such as drawing interpretation and object recognition. Piater et al. [21] learned joint representations for perception-grasping systems, using reinforcement learning and hierarchical Markov models. Although algorithms based on graphical models result in robust object recognition, obtaining labeled samples and conditional probability distributions is a challenge in robot domains.

Researchers in computer vision and robotics are increasingly focusing on developing algorithms for unsupervised learning of object models. Roman et al. [23] proposed a hierarchical approach that relies on the stability of a subset of features extracted from sensory inputs to perform an initial classification of images using unsupervised methods. Parikh et al. [20] developed an algorithm for unsupervised learning of hierarchical spatial structures from images, using a rule-based model and a graph-based represen-

tation for each rule. Prior work on robots has shown that a robot can use visual input to autonomously adapt visual feature models to illumination changes [27], and use temporal cues in addition to other visual and non-visual cues to achieve autonomous navigation![19]. Many of these algorithms fail of fully exploit the rich information in visual inputs. In this paper, we present a novel approach that enables a mobile robot to autonomously learn object models using local, global and temporal visual cues.

3 Proposed Algorithm

This section describes the learning of object models and the use of these models to recognize objects in novel scenes. Learning is triggered by motion cues and image regions corresponding to candidate objects are identified by tracking local gradient features in a sequence of images (Section 3.1). The object models are then composed of low-level and high-level representations of gradient features, connection potentials between gradient features, image segments and color distributions extracted from the candidate regions (Section 3.2). A belief revision strategy uses the learned models for object recognition, as described in Section 3.3.

3.1 Candidate Image Region Selection

Many real-world object possess unique characteristics and trace well-defined motion patterns, although these characteristics and patterns are not known in advance and may change over time. Image regions corresponding to candidate objects are hence identified by tracking MSER-SIFT gradient features [12] in consecutive images. Consider features extracted from images: $\{I_{t-1}, I_t\}$ at time $t-1$ and t:

$$MS_{t-1} = \{ms_{t-1,i}, pos_{t-1,i}\}_{i=1}^{N_{t-1}}$$
$$MS_t = \{ms_{t,i}, pos_{t,i}\}_{i=1}^{N_t}$$

where each feature ms is a $128D$ vector, pos is the feature's (x, y) position in the image, and N_{t-1} and N_t are the number of gradient features in I_{t-1} and I_t respectively. The gradient features in these two sets are matched based on the Euclidean distance metric. The matched features are clustered based on their relative displacement between the images. The underlying hypothesis is that unique features corresponding to an object are likely to have similar relative motion between consecutive images. Clusters with more than a minimum number of matched features are considered to be candidate objects in motion. Convex boundaries are found around each cluster and any cluster that includes many features from a different cluster within its boundary is removed. In addition, pair-wise feature matching is performed over $3-5$ consecutive images. This selection of candidate image regions assumes that object motion is not very fast and has a translational component. We also assume that objects with substantial overlap do not move with the same velocity.

3.2 Learning Layered Object Model

For a candidate region of interest (ROI), an object model is learned autonomously—see Figure 1. The first layer models visual features from the ROI and the second layer models higher-level abstractions for robustness. Specifically, the object model has four components based on: (1) gradient features and relative spatial arrangement of features; (2)

connection potentials between neighboring gradient features and an undirected graph of relationships between these potentials; (3) image segments and a parts-based model of relative spatial arrangement of segments; and (4) color distributions and second-order image statistics. These components are described below.

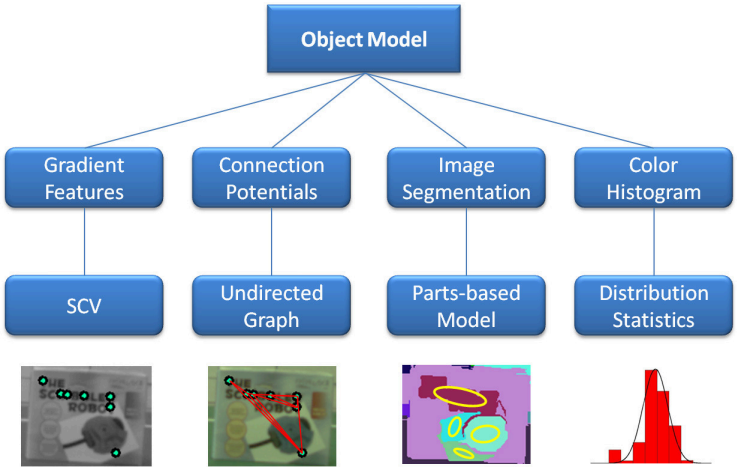


Fig. 1: Learned object models consists of four components that exploit the complementary properties of local, global and temporal visual cues.

Spatial Coherence Vector: Although the gradient features may not be unique, the spatial arrangement of features extracted from the image ROI corresponding to an object is difficult to duplicate. The object model represents the relative spatial arrangement of gradient features using a *spatial coherence vector* (SCV) similar to the coherence vector for color histograms [9]. If the object has N gradient features in the ROI, the SCV for the i^{th} feature is computed along the x and y axes:

$$\begin{aligned} SCV_{x,i} &= \{d^x_{i,1}, d^x_{i,2}, \ldots, d^x_{i,N}\} \\ SCV_{y,i} &= \{d^y_{i,1}, d^y_{i,2}, \ldots, d^y_{i,N}\} \end{aligned} \tag{1}$$

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. If x_i and x_j are the x-coordinates of features i and j in the image:

$$d^x_{i,j} = \begin{cases} 1 & \text{if } x_i > x_j \\ 0 & \text{if } x_i = x_j \\ -1 & \text{if } x_i < x_j \end{cases} \tag{2}$$

and $d^y_{i,j}$ is defined similarly. The object model hence extracts N gradient features from the ROI (each feature is a 128D vector) and a $2(N-1)$-dim SCV for each feature.

Connection Potentials: The second component of the object model captures the distribution of pixels between gradient features in the image ROI. The *connection potential*

between two gradient features is computed as the color distribution of pixels on the line joining the features in the image. The spread between any two features is normalized to unit distance and the 3D color pixel values are collected in 100 bins. This distribution is smoothed along each dimension using an impulse response filter:

$$C_n^{new} = \alpha C_n + (1 - \alpha) C_{n-1} \tag{3}$$

where the smoothed value in the n^{th} bin, i.e., C_n^{new}, is a function of the value in the previous bin (C_{n-1}) and the raw color value in bin (C_n). The parameter α controls the effect of raw data on the smoothed value. This coarse representation (100 bins) provides computational efficiency while modeling the connection potential.

A connected neighborhood is built for each gradient feature (in the learned model) by sorting the features in increasing order of distance from the center of the ROI. Consider the sorted list of N features:

$$\{d_1, ..., d_{k-2}, d_{k-1}, d_k, d_{k+1}, d_{k+2}, ..., d_N\} \tag{4}$$

where $\forall i < j,\ d_i < d_j$. A four-connected neighborhood of each feature is then defined as the four closest neighbors in the sorted list. The object model is then augmented with an undirected graph (i.e., Markov network [2]) that models the neighborhood relationships of connection potentials between gradient features in the object model.

Parts-based Representation: The third component of the object model is the spatial arrangement of object parts made up of image segments. A graph-based segmentation algorithm [6] is used to extract segments from the image ROI such that the pixel values within a segment are similar to each other and significantly different from pixels in surrounding segments. Spurious segments are filtered by rejecting significantly concave segments and segments that do not overlap substantially with the image region under consideration. Individual segments are then modeled as Gaussians that represent segment locations within the ROI. These 2D Gaussian models: $\mathcal{N}(\mu_k, \Sigma_k), k = 1, ..., M$ constitute the "parts" of the object in the ROI. These parts and the list of neighboring parts (that share a boundary) are added to the object model. Two measures are defined to compute pixel similarity within each part ($PartSimM$) and pixel dissimilarity in neighboring parts ($PartDiffM$), as described in Algorithm 1.

Algorithm 1 considers the N pixels in the M parts (i.e., Gaussians) computed in the ROI. First, each pixel is assigned a label $lb(n)$, i.e., membership in one of the M parts, using the *a priori* probability density functions of different parts (lines 2–4). Next, each pixel's contribution to the similarity and dissimilarity measures is computed (lines 5–12). The term N_{in} represents the number of pixels in the part that pixel n belongs to, while N_{nhb} is the number of pixels in all neighboring parts. The function $RGB()$ computes the difference in RGB values of two pixels. For each pixel, lines 6–8 compute the similarity with other pixels in the same part ($Sim_{lb(n)}$), and lines 9–11 compute the dissimilarity with pixels in neighboring parts ($Diff_{lb(n)}$). For both $Sim_{lb(n)}$ and $Diff_{lb(n)}$, the value added in each iteration is the difference in RGB values between the corresponding pixels, weighted by the probability that the two pixels belong to the same part ($Sim_{lb(n)}$) or different parts ($Diff_{lb(n)}$). The contributions of each pixel are summed up, and the similarity and dissimilarity measures ($PartSimM_{lb(n)}$, $PartDiffM_{lb(n)}$) are computed as the logarithm of the summations (lines 13, 14).

Algorithm 1 Similarity + Dissimilarity of Object Parts.

Initialize $Sim = 0$ and $Diff = 0$.
for $n = 1$ to N **do**
  $lb(n) = \arg\max_{1 \leq j \leq M} p(n \,|\mu_j, \Sigma_j)$
end for
for $n = 1$ to N **do**
  for $n_{in} = 1$ to N_{in} **do**
    $Sim_{lb(n)} += \frac{\sum_{\Delta r, \Delta g, \Delta b} RGB(n, n_{in})}{p(n|\mu_{lb(n)}, \Sigma_{lb(n)})\, p(n_{in}|\mu_{lb(n)}, \Sigma_{lb(n)})}$
  end for
  for $n_{nb} = 1$ to N_{nb} **do**
    $Diff_{lb(n)} += \frac{\sum_{\Delta r, \Delta g, \Delta b} RGB(n, n_{nb})}{p(n|\mu_{lb(n)}, \Sigma_{lb(n)})\, p(n_{nb}|\mu_{lb(nb)}, \Sigma_{lb(nb)})}$
  end for
end for
$PartSimM_{lb(n)} = ln(sim_{lb(n)})$
$PartDiffM_{lb(n)} = ln(diff_{lb(n)})$

Local variations in the positions of parts are modeled by artificially displacing the envelope around the extracted parts. The values of $PartSimM$ and $PartDiffM$ computed over these positions are modeled as a gamma distribution. The object model's third component thus consists of image segments, parts-based model and measures of expected similarity (dissimilarity) within (between) parts.

Color Distribution Statistics: The final component of the object model is based on the distribution of pixels extracted from the ROI. These pixels are used to learn normalized histograms, i.e., color space pdfs, in the HSV color space. Since any two of the three normalized dimensions are sufficient statistics, each pdf is learned in (h, v) with ten bins in each dimension. Since color distributions do not constitute a stable or unique representation, second-order image statistics are computed [26]. Specifically, the distance between every pair of pdfs is computed using the Jensen-Shannon (JS) measure:

$$JS(a,b) = \frac{KL(a,m) + KL(b,m)}{2} \qquad (5)$$
$$KL(a,b) = \sum_i \sum_j (a_{i,j} \cdot \ln \frac{a_{i,j}}{b_{i,j}}), \quad m = \frac{a+b}{2}$$

where (a, b) are two pdfs and KL is the KL-divergence measure. The JS measure is robust to spurious peaks in observed pdfs. The fourth component thus consists of color-space pdfs and the distribution of distances between the pdfs.

3.3 Information Fusion for Object Recognition

The learned object models are used for object recognition in subsequent images of novel scenes, irrespective of whether the object is stationary or moving. For a given test image,

this section describes the belief revision strategy to estimate probability of occurrence of different learned objects. Assume initially that a subset of gradient features in the test image have been matched with the features of a learned object model to obtain a test image ROI. The use of energy minimization algorithms to iteratively select test image ROIs is described later in this section.

Gradient Feature-based Matching: The probability of occurrence of a specific object is computed by comparing the SCV of gradient features in the test image ROI with the learned object model's SCV:

$$p_{ssm} = \frac{x_{correct} + y_{correct}}{2 * M},\ p_{ssm} \in [0, 1] \tag{6}$$

$$x_{correct} = \sum_{m=1}^{M} \frac{Nx_{m\_correct}}{N-1}$$

$$y_{correct} = \sum_{m=1}^{M} \frac{Ny_{m\_correct}}{N-1}$$

where $Nx_{m\_correct}$ and $Ny_{m\_correct}$ represent the number of values in the test image SCV that match the learned object model's SCV along the x and y axes respectively. The term M is the count of gradient features in the learned object model. The value of $p_{ssm} \in [0, 1]$ is the probability of spatial match of the two sets of gradient features. A similar computation using other learned object models provides a probability distribution of occurrence of each object in the test image ROI.

Connection Potential Matching: The probability of occurrence of a learned object is also computed by comparing the neighborhood of connection potentials between gradient features in the test image to the connection potentials and neighborhood of the corresponding matched features in the object model. Once the ROI's gradient features have already been matched with the learned object model's gradient features, a similarity measure is computed between connection j in the ROI and the corresponding (matched) connection i in the learned model. This similarity measure uses the corresponding normalized distributions C_n^j and C_n^i in the ROI and learned model:

$$con(i, j) = \sum_{n=1}^{100} f(C_n^i, C_n^j) \tag{7}$$

$$f(a, b) = \begin{cases} 1 & |a - b| > \beta \\ 0 & otherwise \end{cases} \tag{8}$$

where β is a parameter to identify significant change in entries of the connection potentials. The probability of occurrence of the learned object is obtained using the neighborhood of connection potentials in the test image ROI and learned object model:

$$p_{con} = \frac{1}{Z} \sum_{k \in \{1,\ldots,M\}} \sum_{i \in N_k, j \in N_{k_m}} con(i, j) \tag{9}$$

where M gradient features in the object model match the features in the ROI, N_k is the connected neighborhood of gradient feature k in the object model, N_{k_m} is the connected

neighborhood of (matched) gradient feature k_m in the ROI, and Z is a normalizer. A similar computation with other learned object models provides the probability distribution of occurrence of these objects in the ROI.

Parts-based Matching: For the selected test image ROI, a match probability is also computed using the parts-based component of the learned object model. Different relative arrangements of the learned model's parts are compared with pixels in the test image ROI. For pixels in the overlapping region for each arrangement, the similarity of pixels within a learned model part and the dissimilarity of pixels in neighboring parts is computed using the $PartSimM$ and $PartDiffM$ measures in Algorithm 1—the pixel class labels ($lb(n)$) are provided as input. The learned gamma distributions of values of these measures are used to compute the suitability of this arrangement:

$$\begin{aligned} f(x_j) &= gamma(|x_j^{l_i} - x_j| - (k-1)\theta, k, \theta) \\ p_{cdm}^j &= f(PartSimM_j) \times f(PartDiffM_j) \\ p_{cdm} &= \sum_j (w_j^{l_i} \times p_{cdm}^j) \end{aligned} \tag{10}$$

where $(k-1)\theta$ is the stationary point of the gamma distribution, x_j is the pixel similarity ($PartSimM$) or dissimilarity ($PartDiffM$) when considering the i^{th} learned object model's j^{th} part and $x_j^{l_i}$ is the mean of the gamma pdf. The match probability for this relative arrangement (p_{cdm}) is the weighted product of match probabilities p_{cdm}^j of each part. The weight $w_j^{l_i}$ is the ratio of number of pixels in that part divided by the number of pixels in all parts of the object model. The best arrangement is one that maximizes p_{cdm}. A similar computation is performed using other learned object models to obtain the probability distribution of occurrence of the learned objects in the ROI.

Color-based Matching: Color space distributions extracted from the test image ROI are also used to compute the probability of occurrence of the learned objects. For the i^{th} learned object model, the average distance d_{avg,l_i} is computed between the test image pdf and the color space pdfs corresponding to the learned object model, using Equation 5. A comparison with the expected (Gaussian) distribution of distances (for the learned object model) provides p_{js,l_i}, the probability of occurrence of the corresponding object in the test image ROI. A similar computation is performed using other learned object models to obtain the probability distribution of occurrence of different objects in the test image ROI. Note that it is possible (when the second-order statistics are being learned) to use relative values of the average distances between test image pdf and learned pdfs of different object models to obtain the probability of occurrence of the learned objects in the test image ROI.

Information Fusion: Finally, consider the identification of test image ROIs (for analysis) and the information fusion strategy. For ease of explanation, assume that only one object exists in a test image ROI—the algorithm can detect multiple objects in an image or ROI. If the object is moving in a test image sequence, ROIs are identified by tracking gradient features. However, when test images are snapshots of stationary objects or

objects in different scenes, ROIs are identified by matching gradient features in the images with the learned object models. Consider a test image that is being compared with the i^{th} learned object model. For each of the G local gradient features in the model, K nearest neighbors are found in the test image. Each of the possible (at most) $K * G$ combinations is a ROI in the test image that can be analyzed by individual components of the object model. Energy minimization is used for iteratively selecting ROIs from the available combinations, as described later in this section.

For a specific ROI, belief revision is used to combine the evidence provided by components of the learned object models regarding presence of the corresponding objects. First, the *predicted* estimate of the i^{th} learned object's occurrence in the ROI is:

$$p(m_{l_i}^{pre}) = \prod_{j \in \{1,\ldots,G\}} nn_j; \quad nn_j = \frac{dis\_std_j}{dis_j} \tag{11}$$

where dis_std_j is the distance from the i^{th} learned model's j^{th} gradient feature to the nearest neighbor among the ROI's features, and dis_j is the distance to the current selection among the K possible neighbors. Next, the *observations* provided by the individual components of the learned object model are merged as:

$$p(m_{l_i}^{ob}) = p_{con,l_i} \cdot p_{cdm,l_i} \cdot p_{ssm,l_i} \cdot p_{js,l_i} \tag{12}$$

where the individual probabilities (p_{ssm,l_i}, p_{con,l_i}, p_{cdm,l_i}, p_{js,l_i}) of occurrence of i^{th} learned object are computed as described above. We assume that the individual components are mutually independent, which works well in practice. The *corrected* estimate of occurrence of the i^{th} learned object in the test image ROI is then computed as:

$$p(m_{l_i}) = p(m_{l_i}^{pre}) p(m_{l_i}^{ob}) \tag{13}$$

The ROI (among the candidates generated by matching gradient features) that maximizes Equation 13 is the best estimate of the corresponding object's location. This optimization problem is solved using the iterated conditional modes (ICM) energy minimization algorithm [28]. The results with ICM can be sensitive to the choice of initial estimate in high-dimensional spaces. However, we obtain robust performance by using the nearest neighbors of the learned model's gradient features to obtain the initial ROI estimate. The normalized match probability distribution is then computed as:

$$p_{l_i} = \frac{p(m_{l_i})}{\sum_{j=1}^{M} p(m_{l_j})} \tag{14}$$

This probability distribution is used for recognizing learned objects and for detecting novel objects when the match probabilities for all learned models are low.

The overall operation is described in Algorithm 2. A mobile robot begins with a learned map of the domain but no initial knowledge of the desired objects. If the robot is to learn object models, i.e., $modelLearn$ is $true$ in line 3, the robot uses clustered gradient features extracted from consecutive images to obtain candidate ROIs. If a valid ROI exists (line 5), visual features are extracted to populate the four components of the learned object model. The robot attempts to match the new object model with existing models (if any, line 7). If a match with a sufficiently high probability is found, the

Algorithm 2 Object Model Learning and Recognition.

Require: : Ability to learn object models based on feature connections, gradient features, color distributions and color segment parts.
Require: Learned map of the surroundings for navigation.
1: Initialize: $numObjects = 0$ (no prior knowledge).
2: **while** true **do**
3: **if** $modelLearn$ **then**
4: Compute gradient features for I_t and I_{t-1}.
5: **if** $validObject()$ **then**
6: Compute SCV, connection potentials, segment parts and color distribution statistics.
7: **if** $(numObjects > 0)$ & $existModel()$ **then**
8: Augment model of appropriate object.
9: **else**
10: $ComputeNewModel()$
11: $numObjects = numObjects + 1$
12: **end if**
13: **end if**
14: **else**
15: Compute SCV, connection potentials, segment parts and color distributions for I_t.
16: **if** $numObjects > 0$ **then**
17: Compute match probabilities of learned models.
18: Identify object in image.
19: **end if**
20: **end if**
21: **end while**

existing object model is augmented. If a good match is not found, a new entry is created in the list of learned objects (lines 9-12). If the $modelLearn$ flag is not set, the robot performs object recognition using the learned object models (lines 17-18). Although learning and recognition are separated in Algorithm 2 for ease of explanation, the robot performs them concurrently on multiple ROIs.

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 ERA-MOBI robot (a.k.a "erratic") from Videre Design [5] is used as the test platform—see Figure 2. It is a $40cm \times 41cm \times 15cm$ wheeled base equipped with a stereo camera, monocular camera, laser range finder and pan-tilt unit. The experiments used one of the cameras of the stereo unit that provides 640×480 images. The laser range finder with a range of 30m is used to learn the domain map. Although the robot has Wi-Fi communication capability, all experiments were performed on-board using a 1.6GHz Core2 Duo processor and 1GB RAM. Trials were conducted in indoor offices, corridors, outdoor settings, and on images from benchmark datasets.

Fig. 2: Robot test platform: "Erratic".

4.2 Experimental Results

Six object categories were used in the experiments: humans, boxes, airplanes, books, cars and humanoid robots—Figure 3 shows examples of each category. The "car" and "airplane" categories were evaluated in outdoor settings, "human" category was evaluated in outdoor and indoor settings, while other categories were primarily used for indoor trials. Separate models were learned for different objects within a category, e.g., different boxes, books or humans, to result in 20 subcategories. The objects were considered in complex backgrounds that made learning and recognition challenging. During trials, some objects (e.g., humans and cars) moved in specific directions, while others (e.g., boxes and books) were moved on trolleys.

It is a challenge to obtain a relevant database of objects (with well-defined motion) for experimental evaluation. The experiments were conducted over a set of ≈ 1000 images. About 700 images were captured by the robot over a period of time. A total of ≈ 300 images (all "airplane" and some "car" images) are from the *Pascal VOC2006* dataset (to show applicability to benchmark datasets), which includes information about ROIs—suitable ROIs were selected manually when any of these images were used for learning. Each object model is learned autonomously using $\approx 3-5$ images, with ≈ 90 images used for learning all object models—the remaining images are used for evaluation. The goal of the experimental trials is *not* to compare our algorithms with existing computer vision algorithms (for object recognition), but to show that a mobile robot can autonomously learn object models and use these models to robustly recognize objects in novel scenes. The robot accounted for its own motion and processed $3-5$ frames/second to identify moving objects and learn desired models, and to detect objects in subsequent images. Figure 4 shows examples of ROIs and parts extracted from sample images, while Figures 5(a)-5(c) shows results for a challenging test image.

Results of experimental trials show that object models are learned autonomously and used for reliable object recognition in novel scenes. The learned model for one category rarely provides a good match for any other object category. Figure 6 analyzes the contribution of each component of the object model towards the overall classification performance. Each component's contribution depends on the object in the specific image under consideration. For instance, it is difficult to distinguish between the front and

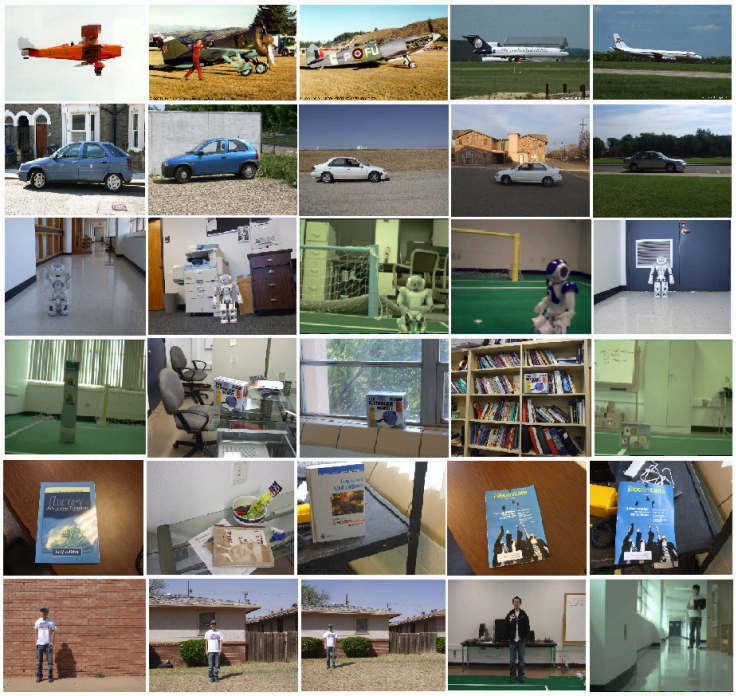


Fig. 3: Sample images of objects from six object categories.

back of a humanoid robot using color distributions (both surfaces are mostly white in color), but the gradient features on these surfaces are significantly different. Figure 6 shows that no single component is able to reliably recognize objects across the different categories and subcategories, but the combination of these components exploits their complementary properties to provide reliable and efficient recognition.

	Box	Car	Human	Robot	Book	Airplane
Box	0.927	0	0	0.033	0.04	0
Car	0.042	0.89	0	0.068	0	0
Human	0.074	0.017	0.78	0.047	0.05	0.032
Robot	0.028	0	0.017	0.863	0.012	0.02
Book	0.058	0	0	0.025	0.917	0
Airplane	0.03	0.024	0	0.016	0.024	0.906

Table 1: Object recognition accuracy averaged over different models (i.e., subcategories) in each category.

Table 1 shows the classification accuracy for the different object categories, averaged over the different object models (subcategories) within each category. The classification is considered to be correct only if the robot matches each object in the test image

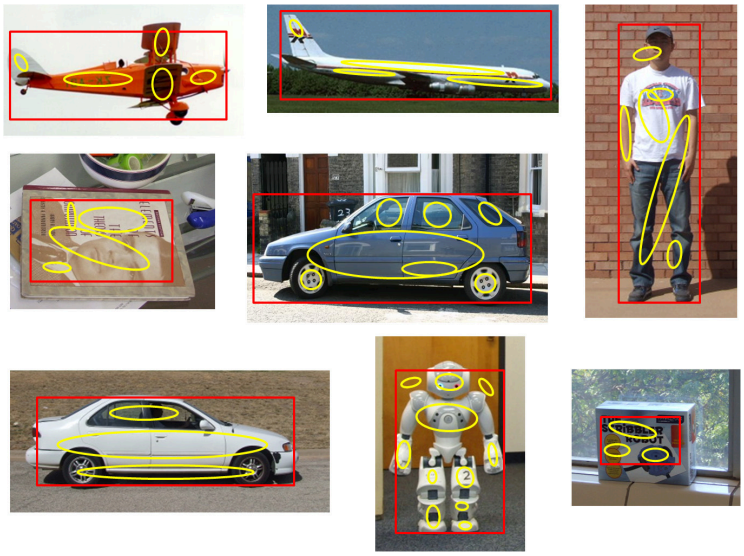
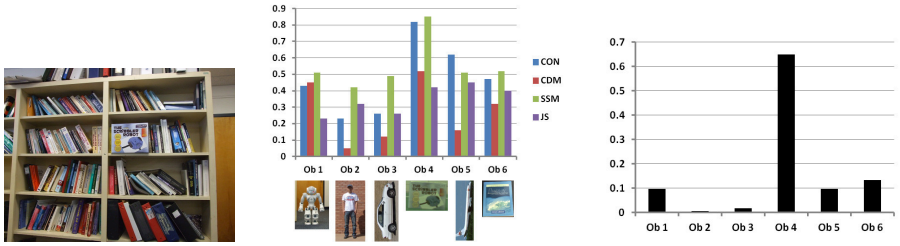


Fig. 4: Images with ROIs and parts extracted.



(a) Test Image. (b) Match Probabilities. (c) Net Match.

Fig. 5: (a) Test image of a box in a complex background; (b) Individual match probabilities— the best subcategory within each category is shown along the x-axis; (c) Net match probabilities for different categories.

to the correct model within the appropriate category, i.e., matching an object in *human-class1* to the learned model *human-class2* is incorrect. The off-diagonal terms therefore represent errors. One reason for the classification errors is the learning of object models with non-unique features, e.g., long shots of the "human" category cause gradient features to be extracted from clothes, resulting in non-unique object models and lower classification accuracy. Some classification errors correspond to situations where a sufficient number of unique features in the test image are not matched with the learned object models due to motion blur or a substantial difference in scale or viewpoint (between learning and testing). A third reason for the errors is the fact that test image ROIs are assigned the label of the object model with the maximum match probability, even if that value is not significantly higher than match probabilities of other objects. Some of the errors can hence be eliminated by assigning a threshold on the match probability for object recognition—performance is not very sensitive to the choice of this threshold. In

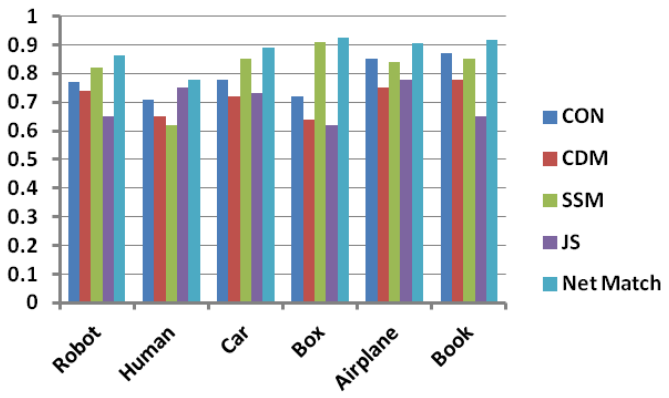


Fig. 6: The match probabilities obtained with each component of the object model, averaged over subcategories in each object category.

addition, classification errors are less frequent in image sequences of objects in motion because identifying the ROI properly enables one or more of the components to identify the object accurately. These experimental results indicate that the autonomously learned object models result in reliable recognition of objects in indoor and outdoor scenes.

5 Conclusions and Future Work

Autonomous operation is a key requirement for mobile robots in dynamic real-world domains. This paper described a novel approach for a mobile robot to autonomously learn object models by exploiting the complementary properties of local, global and temporal visual cues. Image regions corresponding to candidate objects are identified using motion cues, and the objects are modeled using local gradient features, connection potentials between the gradient features, parts-based model of image segments and color distribution statistics. Belief revision and energy minimization algorithms use the learned models for robust recognition in novel scenes.

One constraint in the current set of experiments is that only a couple of objects were moving at any point in time—future research will investigate the extension to image sequences with multiple moving objects, consider other object categories and use stereo vision to disambiguate partially occluded objects. The object model currently assumes independence between the individual components—future work will enable learning of the relationships between the components to build more robust models. In addition, stochastic sampling will be used to improve the efficiency of selecting candidate ROIs and matching parts-based models. Furthermore, the approach will be extended to a team of robots collaborating towards a shared objective.

Acknowledgments

This work was supported in part by the ONR Science of Autonomy award N00014-09-1-0658.

References

1. Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. From Contours to Regions: An Empirical Evaluation. In *Computer Vision and Pattern Recognition*, pages 2294–2301, 2009.
2. Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer-Verlag New York, 2008.
3. Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast Approximate Energy Minimization via Graph Cuts. In *International Conference on Computer Vision*, pages 377–384, 1999.
4. Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. BRIEF: Binary Robust Independent Elementary Features. In *European Conference on Computer Vision*, pages 778–792, 2010.
5. Videre Design. Videre Design Robot and Sensors, 2010. `http://www.videredesign.com/index.php?id=21`.
6. Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Efficient Graph-Based Image Segmentation. *International Journal of Computer Vision*, 59(2):167–181, 2004.
7. Robert Fergus, Pietro Perona, and Andrew Zisserman. Object Class Recognition by Unsupervised Scale-Invariant Learning. In *Computer Vision and Pattern Recognition*, pages 264–271, 2003.
8. Sanja Fidler, Marko Boben, and Ales Leonardis. Similarity-based Cross-Layered Hierarchical Representation for Object Categorization. In *International Conference on Computer Vision and Pattern Recognition*, 2008.
9. P. Greg, Z. Ramin, and M. Justin. Comparing Images Using Color Coherence Vectors. In *ACM International Conference on Multimedia*, 1997.
10. Chunzhao Guo, S. Mita, and D. McAllester. Adaptive Non-Planar Road Detection and Tracking in Challenging Environments using Segmentation-based Markov Random Field. In *International Conference on Robotics and Automation*, 2011.
11. Vladimir Kolmogorov, Antonio Criminisi, Andrew Blake, Geoffrey Cross, and Carsten Rother. Probabilistic Fusion of Stereo with Color and Contrast for Bilayer Segmentation. *Pattern Analysis and Machine Intelligence*, 28:1480–1492, 2006.
12. Xiang Li and Mohan Sridharan. Safe Navigation on a Mobile Robot using Local and Temporal Visual Cues. In *International Conference on Intelligent Autonomous Systems*, Ottawa, Canada, August 30-September 1 2010.
13. D. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.
14. Tianyang Ma and Longin Jan Latecki. From partial shape matching through local deformation to robust global shape similarity for object detection. In *Computer Vision and Pattern Recognition*, pages 1441–1448, 2011.
15. J. Matas, O. Chum, M.Urban, and T. Pajdla. Robust Wide Baseline Stereo from Maximally Stable Extremal Regions. In *British Machine Vision Conference*, 2002.
16. Krystian Mikolajczyk and Cordelia Schmid. Scale and Affine Invariant Interest Point Detectors. *International Journal of Computer Vision*, 60(1):63–86, 2004.
17. Frank Moosmann, Bill Triggs, and Frederic Jurie. Fast Discriminative Visual Codebooks using Randomized Clustering Forests. In *The Neural Information Processing Systems (NIPS)*, 2006.
18. Pierre Moreels and Pietro Perona. Evaluation of Feature Detectors and Descriptors Based on 3D Objects. In *The International Conference on Computer Vision (ICCV)*, 2005.
19. Aniket Murarka, Mohan Sridharan, and Benjamin Kuipers. Detecting Obstacles and Dropoffs using Stereo and Motion Cues for Safe Local Motion. In *International Conference on Intelligent Robots and Systems*, 2008.

20. Devi Parikh, C. Lawrence Zitnick, and Tsuhan Chen. Unsupervised learning of hierarchical spatial structures in images. In *Computer Vision and Pattern Recognition*, pages 2743–2750, 2009.
21. Justus H. Piater, Sbastien Jodogne, Renaud Detry, Dirk Kraft, Norbert Krger, Oliver Kroemer, and Jan Peters. Learning visual representations for perception-action systems. *International Journal of Robotic Research*, 30:294–307, 2011.
22. J. Porway and S. C. Zhu. C4: Computing Multiple Solutions in Graphical Models by Cluster Sampling. *Pattern Analysis and Machine Intelligence*, 33(9):1713–1727, 2011.
23. K. Roman, N. Juan, N. Eduardo, and D. Bertrand. Track-based Self-supervised Classification of Dynamic Obstacles. *Autonomous Robots*, 29(2):219–233, 2010.
24. C. Schmid and R.Mohr. Local Grayvalue Invariants for Image Retrieval. *PAMI*, 19(5):530–535, 1997.
25. Thomas Serre, Lior Wolf, Stanley Bileschi, Maximilian Riesenhuber, and Tomaso Poggio. Robust Object Recognition with Cortex-Like Mechanisms. *Pattern Analysis and Machine Intelligence*, 29(3), March 2007.
26. M. Sridharan and P. Stone. Global Action Selection for Illumination Invariant Color Modeling. In *IROS*, 2007.
27. Mohan Sridharan and Peter Stone. Color Learning and Illumination Invariance on Mobile Robots: A Survey. *Robotics and Autonomous Systems*, 75(1):1–38, 2009.
28. Richard Szeliski, Ramin Zabih, Daniel Scharstein, Olga Veksler, Vladimir Kolmogorov, Aseem Agarwala, Marshall F. Tappen, and Carsten Rother. A Comparative Study of Energy Minimization Methods for Markov Random Fields with Smoothness-Based Priors. *Pattern Analysis and Machine Intelligence*, 30:1068–1080, 2008.
29. Jonathan S. Yedidia, William T. Freeman, and Yair Weiss. Generalized Belief Propagation. In *Neural Information Processing Systems*, pages 689–695, 2000.
30. Alper Yilmaz, Xin Li, and Mubarak Shah. Contour-based object tracking with occlusion handling in video acquired using mobile cameras. 26:1531–1536, 2004.