Autonomous Learning of Object Models on Mobile Robots using Visual Cues

by

Xiang Li, M.S., B.S.

Dissertation

In

Computer Science

Submitted to the Graduate Faculty
of Texas Tech University in
Partial Fulfillment of
the Requirements for the Degree of

Doctor of Philosophy

Approved

Committee:

Dr. Mohan Sridharan, Chairman

Dr. J. Nelson Rushton

Dr. Hamed Sari-Sarraf

Dr. Peter Stone

Dr. Dominick Casadonote
Interim Dean of the Graduate School

Texas Tech University

August, 2013

ACKNOWLEDGMENTS

I would like to express the deepest appreciation to my supervisor, Dr. Mohan Sridharan, whose expertise, understanding, and patience, added considerably to my graduate experience. Mohan was a great source of advice and encouragement, and gave me a lot of freedom in choosing my research focus. Without his guidance and persistent help, this dissertation would not have been possible.

I would also like to thank my friends in the Stochastic Estimation and Autonomous Robotics Lab, particularly Shiqi Zhang, for our debates, exchanges of knowledge, skills, and venting of frustration during my graduate program, which helped enrich the experience.

I must also acknowledge my committee members, Dr. J. Nelson Rushton, Dr. Hamed Sari-Sarraf and Dr. Peter Stone for their help and advice.

I would also like to thank my family for the support they provided me through my entire life, and in particular, I must acknowledge my wife, Yue, without whose love, encouragement and editing assistance, I would not have finished this dissertation.

TABLE OF CONTENTS

ABSTRACT

Mobile robots are increasingly being used in real-world application domains such as disaster rescue, surveillance, health care and navigation. These application domains are typically characterized by partial observability, non-deterministic action outcomes and unforeseen changes. A major challenge to the widespread deployment of robots in such domains is the ability to learn models of domain objects automatically and efficiently, and to adapt the learned models in response to changes. Although sophisticated algorithms have been developed for modeling and recognizing objects using different visual cues, existing algorithms are predominantly computationally expensive, and require considerable prior knowledge or many labeled training samples of desired objects to learn object models. Enabling robots to learn object models and recognize objects with minimal human supervision thus continues to be an open problem.

The above-mentioned challenges are offset by some observations. First, many objects have distinctive characteristics, locations, and motion patterns, although these parameters may not be known in advance and may change over time. Second, images encode information about objects in the form of many different visual cues. Third, any specific task performed by robots typically requires accurate models of only a small number of domain objects. This dissertation describes an algorithm that exploits these observations to achieve the following objectives:

1. Investigate learning of object models from a small $(3 - 8)$ number of images. Robots consider objects that move to be interesting, efficiently identifying corresponding image regions using motion cues.

2. Exploit complementary strengths of appearance-based and contextual visual cues to

efficiently learn representative models of these objects from relevant image regions.

3. Use 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 with minimal human supervision.

These objectives promote incremental learning, enabling robots can acquire and use sensor inputs and human feedback based on need and availability. The object models consist of: 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. Although the visual cues underlying individual components of the object model have been used in other algorithms, our representation of these cues fully exploits their complementary strengths, resulting in reliable and efficient learning and recognition in indoor and outdoor domains. All algorithms are evaluated on wheeled robots in indoor and outdoor domains and on images drawn from benchmark datasets.

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

Mobile robots are increasingly being used in real-world application domains such as disaster rescue, surveillance, health care and navigation [17, 43, 47, 116] due to the ready availability of high-fidelity sensors at moderate costs and the development of sophisticated sensory input processing algorithms. For instance, many sophisticated algorithms have been developed for object recognition using a variety of visual cues [33, 35, 50, 62, 67, 70, 86]. Real-world application domains are characterized by partial observability, non-deterministic action outcomes and unforeseen dynamic changes. A major challenge to the widespread deployment of robots in such domains is the ability to autonomously and efficiently learn models of domain objects and adapt the learned models in response to changes. Simultaneously, it is usually difficult for robots to have a large number of images to learn, especially the labeled ones. The challenge is all the more pronounced when people considering images from a color camera, due to the sensitivity of visual inputs to environmental factors (e.g., illumination) and the computational complexity of visual information processing algorithms. In addition, many visual processing algorithms require extensive human supervision during a training phase that may need to be repeated when environmental conditions or object configurations change substantially. Vision-based autonomous learning and adaptation on mobile robots hence remains an open problem.

The challenges described above are offset by the presence of a significant amount of structure in many real-world application domains. Objects with unique characteristics (e.g., color or shape) and motion patterns exist at specific locations, although these parameters are not known in advance and may change over time. This dissertation describes an approach that enables a mobile robot to exploit the structure inherent in real-world applications domains, autonomously learning models for objects that move (or are moved) based on 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. The algorithm described in this dissertation exploits these observations to achieve the following objectives:

1. Investigate learning of object models from a small $(3-8)$ number of images. Robots consider objects that move to be interesting, efficiently identifying corresponding image regions using motion cues.

2. Exploit complementary strengths of appearance-based and contextual visual cues to efficiently learn representative models of these objects from relevant image regions.

3. Use 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 with minimal human supervision.

The algorithm is based on the following underlying assumptions:

1. The interesting objects are those that move.

2. Object motion is not very fast and has a non-trivial linear component.

3. Objects with substantial overlap do not move with the same velocity.

4. The target should not be texture-less surface or object with repetitive pattern.

In this dissertation, robots consider objects that move to be interesting and automatically learn models for moving objects. In terms of autonomous learning, the object's motion should not be at high speed in order for robots to capture enough images with the moving objects. If being provided labeled samples, robot can build models for stationary objects as well. Robots use learned models to recognize the corresponding objects in novel scenarios, *irrespective of whether the objects are stationary or moving*. Learning is triggered by identifying interesting image regions corresponding to candidate objects using temporal visual cues, i.e., by tracking local image gradient features over a short sequence of 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, second-order statistics of color distributions, probabilistic models of local context and convex hull of gradient features. Our object model thus utilizes the complementary strengths of local, global and temporal visual cues to build robust models that characterize environmental objects. Belief revision and energy minimization algorithms use the learned models to recognize stationary and moving objects in novel scenarios. Furthermore, the learning method *bootstraps* off of the available information: the learned models are revised incrementally as the corresponding objects are recognized, leading to robust object recognition in subsequent frames. All algorithms are implemented on mobile robots and evaluated on benchmark computer vision datasets, and in indoor and outdoor environments.

The remainder of this dissertation is organized as follows. Chapter 2 reviews related work in computer vision and robotics to motivate our approach for autonomously learning probabilistic object models. An efficient characterization of image gradient features is presented in Section 3.1, followed by a description of the object model in Section 3.3. Section 3.3.1–Section 3.3.6 describe individual components of the object model based on image gradient features, connection potentials, graph-based image segments, color distributions, local context and convex boundaries. The belief revision and energy minimization algorithms used for recognition are described in Section 3.4. Chapter 4 presents the experimental setup and discusses experimental results. Finally, the conclusions and future research directions are described in Chapter 5.

CHAPTER 2

RELATED WORK

Object recognition continues to be a major challenge in computer vision and robotics. Visual recognition can be pursued on different levels of semantic granularity [81]. One extreme strategy is exemplar detection [62, 67], where exactly the same query object is sought in scenes with different environmental conditions such as background, lighting, occlusion, viewpoint. etc. The other extreme is category-level object recognition, where all instances of a category are to be recognized [4, 14]. The general goal is to represent objects by learning visual cues in a common model. The approach developed in this dissertation to learn object models is about exemplar detection.

Many algorithms use different visual cues to characterize objects. Section 2.1–Section 2.5 summarize the related algorithms based on those visual cues I am using in the algorithms described in this dissertation. Section 2.6–Section 2.9 discuss object recognition algorithms. All along, these methods are compared to the algorithms described in this dissertation.

2.1 Local Features-based Algorithms

Interest points are local features for which the signal changes two-dimensionally. They can be extracted reliably, are robust to partial visibility and the information content in these points is high. Image gradients can be used for robust feature and texture matching by extracting information from interest points in images. An image gradient is a directional change in the intensity or color in an image. There are several types of gradient features, like corner, region and blob. The detected corner points [74] correspond to points in the 2D image with high curvature. These do not necessarily correspond to projections of 3D corners. Corners are found at various types of junctions, on highly textured surfaces, at occlusion boundaries, etc. Corner based features extract stable features, that can be matched well in spite of changes in viewing conditions. The Harris detector was identified as the most stable one in many independent evaluations [46, 70]. It is a convenient tool for providing a large number of features. There are also multi-scale as well as scale and affine invariant extensions of this approach. For example, SUSAN [110] computes the fraction of pixels within a neighborhood which have similar intensity to the center pixel. It is more efficient but also more sensitive to noise. The FAST detector, introduced by Rosten and Drummond in [95, 96] builds on the SUSAN detector, which compares pixels only on a circle of fixed radius around the point. The local features detected with region detectors typically represent homogeneous regions. Intensity-based Regions (IBR) [121] starts from intensity extrema (detected at multiple scales), and explores the image around them in a radial way, delineating regions of arbitrary shape, which are then replaced by ellipses. IBR is more robust to small gaps in the region contour, but it may break down when the region is non-convex. IBR extracts small regions whose intensity patterns clearly stand out with respect to its immediate surroundings. However, image segments are typically relatively large too large, in fact, to be used as local features. Superpixels are typically based on segmentation methods which are computationally expensive like normalized cuts. Ren's approach moves from pixels to the piecewise linear approximations of contours and the constrained Delaunay triangulation to model continuity [92]. Because of the weighting factor measuring the self-dissimilarity over scale, the blob detector typically fires on blob-like structures in the image. The Hessian affine detector [69] simultaneously adapts location as well as scale and shape of the point neighborhood. DoG (difference-of Gaussians) [62] can also be categorized as blob detectors. Because of the weighting factor measuring the self-dissimilarity over scale, the detector typically fires on blob-like structures in the image. However, those representations have an disadvantage that the segmentation results are still unstable and inefficient for processing large amounts of images. MSER (maximally stable extremal regions) [67] successfully deals with these problems which often find blob-like structures in the image. However, apart from blob-like structures, they also detect other, more irregularly shaped patterns, which is considered as their distinctive property.

Feature detection is not the final goal, but just the first step in a processing chain, followed by feature description and matching. Many algorithms have used scale, rotation and affine-invariant local image gradient features to characterize and recognize objects [16, 62, 70]. For instance, Schmid and Mohr [101] represented objects using gray-value invariants at interest points, and used a voting algorithm and semi-local constraints to recognize objects in test images. Mikolajczyk and Schmid [70] used gradient features invariant to affine transforms to characterize and recognize objects in images. Lowe [62] developed the scale-invariant feature transform (SIFT) that uses image gradient features to characterize objects of interest. More recently, [57] represented objects using a codebook of gradient features and an implicit shape model, interleaving object categorization and foreground segmentation for recognition, while [97] generalized and optimized the corner detector for repeatability with little loss of efficiency in 3D scenes. [8] developed a descriptor (SURF) that are faster to compute and match while preserving the discriminative power of SIFT. Like SIFT, SURF relies on local gradient histograms but uses integral images to speed up the computation. Considering this 64 dimensions vector yields good recognition performance, that version has become a *de facto* standard. [15] computed a binary descriptor (BRIEF) on the basis of simple intensity difference tests. Compared with other descriptors, BRIEF is robuster to typical classes of photometric and geometric image transformations. Similar to the algorithm described in this dissertation, BRIEF is targeting real-time applications. In Chapter 4, we therefore compare our algorithm to both SURF and BRIEF.

Although algorithms based on gradient features have been used in many applications [19], they neglect the rich global information encoded in color images. Such gradients are also not discriminative enough for texture-less surfaces (e.g. walls, doors) and objects with repetitive patterns, where other visual features may prove useful. Other object recognition algorithms characterize objects using models of appearance, shape and size [33] or as a hierarchical decomposition of parts [35], and perform scene understanding using human inputs [86]. Transmission and storage of local feature descriptors are of critical importance in the context of mobile visual search applications. Chandrasekhar et al. [18] represented gradient histograms as tree structures which can be efficiently compressed. Their proposed framework offered low complexity and has significant speed-up in the matching stage. Researchers have also developed object models based on human visual cortical mechanisms [103] and visual code-books of object features [73].

Overall, many gradient feature algorithms describe the texture information in local regions, which neglect the rich global information encoded in color images. These algorithms are also computationally expensive for robot application domains.

2.2 Color-based Algorithms

Color is commonly experienced as an indispensable quality in describing the world around us. The basic approach to compute color image derivatives is to calculate separately the derivatives of the channels and add them to produce the final color gradient.

However, the derivatives of a color edge can be in opposing directions for the separate color channels. Therefore, a summation of the derivatives per channel will discard the correlation between color channels [23]. To better understand the formation of color images, the dichromatic reflection model has been introduced by Shafer [104]. The model describes how photometric changes, such as shadows and specularities, influence the RGB-values in an image. On the basis of this model, algorithms have been proposed which are invariant to different photometric phenomena such as shadows, illumination and specularities [40]. Lauziere et al. [54] describe an approach for learning color models and recognizing objects under varying illumination using the prior knowledge of the spectral reflectances of the objects under consideration. However, their method require extensive measurement of the camera characteristics and the spectral properties of the environment under consideration, while mobile robots are frequently required to operate in new environments. Histograms are a convenient tool if their inherent drawbacks are avoided. Local kernel histograms [77] can retrieve spatial information using small histograms size for real-time processing and including smoothing features to cope with small movements and camera noise. Attempts to learn color models or make them independent to illumination changes have produced reasonable success [41, 51]. Finlayson et al. compute the covariance matrix of normalized mean-subtracted color and use them as indexing numbers the three angles formed by the inverse cosine of the covariances [37]. Kobayashi et al. [52] analyze the estimation process of the color invariants from RGB images, and propose a novel invariant feature of color based on the elementary invariants to meet the circular continuity residing in the mapping between colors and their invariants. Lee [56] used a generate-and-test methodology to evaluate which simulated global illumination condition leads to the generated view that most closely matches what the robot actually sees. However, in addition to involving extensive computation, those methods have a main drawback that illumination invariant color features are less informative about the image content than original coordinates [78]. Weijer et al. [123] extend the description of local features with color information. Dalas et al. [98] presented a strategy that combines color and depth images to detect people in indoor environments. Similarity of image appearance and closeness in 3D position over time yield weights on the edges of a directed graph that they partition greedily into tracklets, sequences of chronologically ordered observations with high edge weights. Each tracklet is assigned the highest score that a Histograms-of-Oriented Gradients (HOG) person detector yields for observations in the tracklet.

Overall, many color-based algorithms are typically sensitive to illumination changes. Furthermore, besides extensive computation, many illumination invariant color features are less informative about the image content than other visual cues from original image.

2.3 Parts-based Algorithms

In [48], Hoffman and Richards noticed that the parts of an object play a key role in recognition. There has been considerable research in computer vision on representing objects as a collection of parts [1, 2, 28, 71, 127]. Parts are typically shared in a discrete fashion; for example, a single template for a wheel part may be shared across multiple view-based mixture models [118, 83] or within a compositional grammar of vehicles [130]. In particular, [83] learns coefficients which calibrate parts that are shared across sub-category mixtures. However, because parts may look different under different viewpoints and compositions, many algorithms share a linear subspace rather than a fixed template, letting a small number of basis filters generate a large, near-continuous range of part appearances. For example, Mohan et al. [71] developed an example-based framework for detecting objects in static images, using four distinct example-based detectors to find different components of the human body. The algorithm by Fergus et al. [33] learns models of object properties such as shape, size and position, and estimates model parameters using the Expectation Maximization algorithm [11]. Besides linear subspace, Cascades have also been used for object detection for many years [126, 38]. Recently, for example, cascades have been applied to kernel based methods [124] resulting in models that, while very accurate, are still orders of magnitude slower than the algorithm described in this dissertation. More recent work focused on deformable part models. Engel and Toennies [28] proposed an algorithm for localizing complex shapes in images using a part-based deformable shape representation with finite element vibration modes. Similarly, Felzenszwalb et al. [31] developed an object detection system based on mixtures of multiscale deformable part models and introduced an approach for discriminative training with partially labeled data. Pedersoli et al. [87] accelerates part based and deformable models by reducing the number of image locations where part filters must be evaluated as well. Ullman et al. [122] showed that many parts-based algorithms require manual intervention (after a few parts are learned automatically) to guide the search for further parts and constrain computational costs.

Overall, most of these algorithms based on parts require extensive prior knowledge for modeling target objects. Those algorithms also require a large number of training samples.

2.4 Context-based Algorithms

Context is believed to play an important role in recognition for humans [80, 24]. Humans use a significant amount of contextual information to recognize objects in images [85]. Object recognition algorithms have modeled global context at the level of the entire image, such as global texture [117, 109] or 3D scene information [49]. Global context is a common approach to localizing objects in images by sliding a window across all locations and scales in the image and classify each local window as containing either the target or background [89]. Oliva et al. [79] use a statistical summary of an image which provides an efficient and compact representation of the image that can be used to inform about scene properties, in addition to being used to prime local object features. Scene context can be used to restrict the set of possible objects that may be present in the scene, or to reduce the possible locations an object may be present. Torralba et al. [117] represented global context information in terms of the spatial layout of spectral components. Mruphy et al. [76] proposed to use the scene context (image as a whole) as an extra source of (global) information, to help resolve local ambiguities. However, it is a heavy task for robots to compute global context information. Context can also be modeled locally in image regions surrounding the object of interest. Shotton et al. [107] proposed an approach to learning a discriminative model of object classes, incorporating appearance, shape and context information efficiently. Fink et al. [36] introduced Mutual Boosting which is a method aimed at incorporating contextual information to augment object detection. Recent research has focused on extracting adaptive (and different kinds of) contextual cues from image regions [58, 63]. Furthermore, research shows that the importance of contextual cues varies with the quality of the appearance information [85]. However, contextual information can be used in conjunction with local approaches to improve performance, efficiency and tolerance to image degradation.

Overall, many context-based recognition approaches do not provide simple representations of context, and are computationally expensive for robot application domains.

2.5 Shape-based Algorithms

The overall shape model of the approaches is either (a) a global geometric organization of edge fragments [10, 82]; or (b) an ensemble of pairwise constraints between point features [27, 64]. Global geometric shape models are appealing because of their ability to handle deformations, which can be represented in several ways. Several earlier works for shape description are based on silhouettes [72, 106]. Since silhouettes ignore internal contours and are difficult to extract from cluttered images, more recent works represent shapes as loose collections of 2D points. Cootes et al. [20] introduced Active Shape Models, which can only deform to fit the data in ways consistent with the training set. [108] This semi-local representation allows to establish point-to-point correspondences between shapes even under nonrigid deformations. Elidan et al. [27] proposed another way to use pairwise spatial relations between landmark points. Other works propose more informative structures than individual points as features, in order to simplify matching. Belongie et al. [10] proposed the Shape Context, which captures for each point the spatial distribution of all other points relative to it on the shape. More recently, Ferrari et al. [34] proposed an approach for learning class-specific explicit shape models from images annotated by bounding boxes, and localizing the boundaries of novel class instances in the presence of extensive clutter, scale changes, and intra-class variability. Trinh et al. [119] described a top-down object detection and segmentation approach that uses a skeleton-based shape model and that works directly on real images. Ma et al. [64] proposed contour based object detection suitable for matching of edge fragments.

Overall, similar to context-based algorithms, many shape-based algorithms do not have a simple representation, and are computationally expensive.

2.6 Algorithms based on Multiple Visual Cues

Recent algorithms have used multiple visual cues and interactions with objects for different tasks, e.g., [94] learn spatial relationships between objects, [100] distinguish objects from background, and [59] automatically discover groups of related objects. Parikh et al. [84] enabled unsupervised learning of hierarchical spatial structures using rule-based models. Ommer et al. [81] described a composition system that automatically learns structured, hierarchical object representations in an unsupervised manner without requiring manual segmentation or manual object localization. But those unsupervised algorithm require many training samples. In many real-world applications, robot may not have a large database for training. Bayesian incremental algorithm [30] can learn a model based on appearance and shape from several images. Gehler et al. [39] studied several models that aim at learning the correct weighting of different features from training data. However, those algorithms need significant human supervision and domain knowledge. Du et al. [25] presented a systematic approach to integrating multiple cues in visual tracking. Their examples selectively integrate four visual cues including color, edges, motion and contours. But they manually initialized the targets of interest in the first frame of each sequence and learned the reference models. Therefore it is still a challenge to learn a model autonomously from a small number of images.

2.7 Visual Object Recognition on Robots

Most computer vision algorithms discussed above can’t be applied for real-time system considering their time consumption. What’s more, these algorithms require many training samples and/or significant human supervision and domain knowledge; it is difficult to satisfy these requirements in many robot application domains. Visual object recognition on robots is typically achieved using simplified versions of computer vision algorithms, e.g., using gradient features or heuristic constraints derived from known object properties [9, 26, 113, 114, 116]. For instance, Se et al. [102] enabled a mobile robot to use scale invariant visual landmarks to localize and build a 3D environment map. Pressigout and Marchand [91] proposed a real-time tracking framework for visual servoing applications based on the fusion of visual cues, while Spinello et al. [111] developed an approach to detect and track people and cars, using visual input and laser range data. Ess et al. [29] jointly estimated camera position and stereo depth while detecting objects and their trajectories based on visual cues. More recently, Piater et al. [88] used reinforcement learning and hierarchical Markov models to learn joint representations for perception-grasping systems. However, many of these algorithms require significant human supervision and knowledge of the task and the domain. [125] outlined an online, any-time planning framework enabling the active exploration for object detection. However, the algorithm does not fully exploit the information in the environment.

2.8 Key Mathematical Principles used in Computer Vision

Besides the research on visual cues, computer vision algorithms traditionally draw upon mathematical principles such as energy minimization, graph theory and belief propagation [12, 53, 129]. For instance, Guo et al. [45] developed an adaptive non-planar road detection and tracking algorithm, using a Markov random field (MRF) for optimization and belief propagation in segmented images. [99] used a probabilistic model for fast data collection and augmentation. More recently, Porway and Zhu [90] developed an algorithm based on generative models, energy minimization and Markov Chain Monte Carlo (MCMC) inference to outperform existing algorithms in object recognition tasks. However, those algorithms require considerable prior knowledge. Kolmogorov et al. [53]

used MRF models to build inference layers based on color, contrast and stereo matching. So it requires depth information, which means it does not work only with a monocular camera. Arbelaez et al. [5] used normalized energy of the match between images as a measure of goodness of fit. White et al. [128] defined generative models for sets of graphs, using their spectral representation to construct a dual vector space. Roux et al. [13] introduced the masked RBM, which explicitly models occlusion boundaries in image patches by factoring the appearance of any patch region from its shape. However, those algorithms are computationally expensive. Although robot application domains make it difficult to obtain a large number of labeled samples or considerable prior knowledge our approach enables efficient use of energy minimization and generative model.

2.9 Unsupervised Learning of Object Models

Given the importance of autonomous operation on robots, algorithms are being developed for unsupervised learning of object models. For instance, Roman et al. [93] used the stability of a subset of features extracted from sensory inputs for initial unsupervised classification. Compared with the algorithm described in this dissertation, we do not require any sensory inputs but the images from the camera. Prior research has also enabled a robot to use visual input to autonomously adapt visual feature models to illumination changes [112], and use temporal visual cues in addition to stereo and range inputs to achieve autonomous navigation [75]. However, they did not use complementary strengths of different visual cues. More recently, [55] built high-level class-specific feature detectors from unlabeled data using a large neural network with more than one billion connections, while [66] combined a discriminative object detector with the correspondence offered by the nearest-neighbor approach. [65] developed an algorithm for unsupervised scene classification that used the context of image features for semantic recognition of indoor scenes on a mobile robot. However, these algorithms are computationally expensive and require accurate domain knowledge. Unsupervised multiple instance learning (MIL) [105] can collect the online samples for incremental learning but needs considerable training data. [3] recently developed an objectness measure based on multiple image cues to automatically identify image windows containing objects of any learned class. Although by combining their different image cues this complementary strategy provides interesting capabilities, we show (experimentally) in Chapter 4 that the algorithm is computationally expensive and does not achieve the desired objectives (e.g., reliability) because visual cues are not fully exploited.

2.10 Summary

As described in the previous sections, most computer vision algorithms are computationally expensive, requiring many labeled training samples or extensive human supervision. On the other hand, many robot vision algorithms do not fully exploit the information in the environment. This dissertation describes and thoroughly evaluates the algorithm that support incremental learning of representative object models from a small number of images, resulting in reliable and efficient object recognition in novel scenes. Since none of the single visual cues can provide high recognition performance, our object model uses complementary strengths of appearance-based and contextual visual cues:

- Relative spatial arrangement of gradient features
- Graph-based models of neighborhoods of gradient features
- Parts-based representation of image segments
- Color distribution statistics

- Probabilistic models of local context
- Convex hull of shape representation

Based on those visual cues, the algorithm described in this dissertation makes the following significant contributions:

- Focus on a small subset of interesting domain objects identified using motion cues.
- Providing efficient implementations of algorithms for extracting features from images.
- Using energy minimization algorithm for iteratively selecting image regions of interest (ROIs) for further analysis.
- Building generative models of information fusion to make best use of all relevant local, global, temporal and contextual information.

CHAPTER 3
ALGORITHM

Many real-world application domains are characterized by partial observability, non-deterministic action outcomes and unforeseen dynamic changes. A major challenge to the widespread deployment of robots in such domains is the ability to autonomously and efficiently learn models of domain objects and adapt the learned models in response to changes. Many existing computer vision algorithms are computationally expensive and are not suitable for robots. In addition, autonomous object model learning based on a relatively small database is still a problem for mobile robots. In this dissertation, we use complementary strengths of appearance-based and contextual visual cues to build an object model autonomously based on a small sequence of images. An energy minimization algorithm and a generative model of information fusion use the learned models for reliable and efficient object recognition in novel scenes. Underlying assumptions include: (1) The interesting objects are those that move. (2) object motion is not at very high speed and has a non-trivial linear component; (3) objects with substantial overlap do not move with the same velocity; and (4) The target should not be texture-less surface or object with repetitive pattern. These assumptions work well in practice.

This chapter describes the learning of object models and the use of these models for probabilistic object recognition in novel scenes. We first describe a reliable and efficient method for detecting unique image gradient features in images (Section 3.1). Tracking these gradient features in short image sequences enables the robot to model motion cues and identify regions of interest (ROIs) corresponding to candidate objects (Section 3.2). The overall object model learned from each candidate region is described in Section 3.3, followed by a description of individual components (of the object model) in the subsequent sections. Belief revision and energy minimization algorithms use the learned models to recognize stationary and moving objects in novel scenes, as described in Section 3.4.

3.1 Salient Image Gradient Extraction

As described in Chapter 2, local image gradient features have been used extensively to characterize and recognize objects because they are robust to one or more factors such as scale, orientation, affine transforms, illumination and viewpoint [62, 67, 68]. Algorithms that extract gradient features typically consist of a *detector* and a *descriptor*. The detector uses second-order gradients to extract small image regions (called *keypoints*) that are consistent across variations in the factors of interest. The *descriptor* associates each extracted keypoint with a compact signature. Objects of interest can be represented by a database of such *feature descriptors* extracted from relevant image regions.

[120] experimentally compared many detectors and descriptors in terms of computational efficiency and recognition accuracy. These experiments indicate that the MSER detector provides good computational efficiency by identifying a small set of unique regions to characterize objects. Experiments also indicate that the SIFT descriptor, which uses a 128-dimensional vector to represent each distinctive region, provides good object recognition accuracy. The detector for SIFT is the *Difference of Gaussians* (DoG) operator implemented in scale-space, while MSER finds elliptical covariant regions on level sets of the image.

DoG is a feature enhancement algorithm that involves the subtraction of one blurred version of an original image from another, less blurred version of the original. In the simple case of grayscale images, the blurred images are obtained by convolving the original grayscale images with Gaussian kernels having differing standard deviations. Blurring an image using a Gaussian kernel suppresses only high-frequency spatial information. Subtracting one image from the other preserves spatial information that lies between the range of frequencies that are preserved in the two blurred images. Thus, the DoG is a band-pass filter that discards all but a handful of spatial frequencies that are present in the original grayscale image [22].

MSER is used as a method of blob detection in images. Because the regions are defined exclusively by the intensity function in the region and the outer border, this leads to many key characteristics of the regions which make them useful. Over a large range of thresholds, the local binarization is stable in certain regions, and have the properties listed below [67].

- Invariance to affine transformation of image intensities
- Covariance to adjacency preserving (continuous)transformation on the image domain
- Stability: only regions whose support is nearly the same over a range of thresholds is selected.
- Multi-scale detection without any smoothing involved, both fine and large structure is detected.
- Note however that detection of MSERs in a scale pyramid improves repeatability, and number of correspondences across scale changes.
- The set of all extremal regions can be enumerated in worst-case , where is the number of pixels in the image.

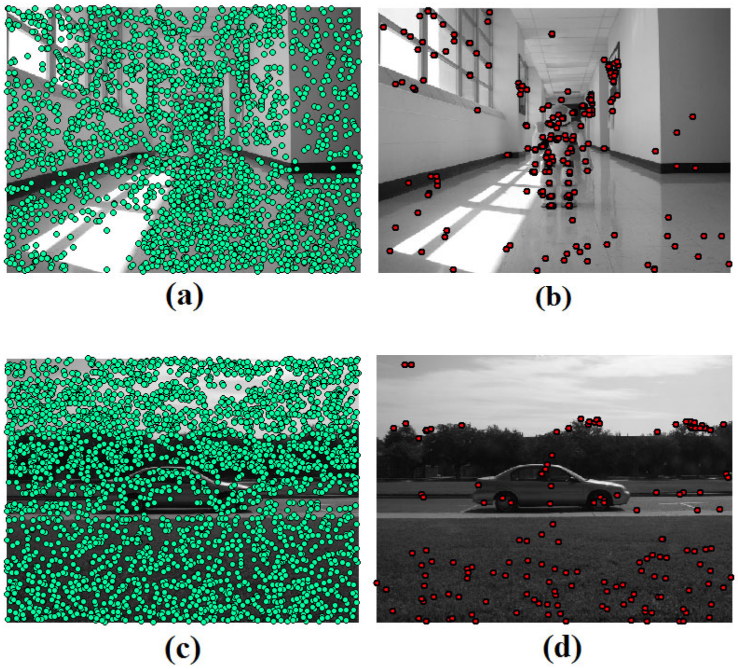


Figure 3.1: Comparison of MSER and SIFT (with DoG detector): (a),(c) keypoints detected with the DoG algorithm; (b),(d) keypoints extracted with MSER algorithm. MSER finds small set of distinctive keypoints.

As shown in Figure 3.1, MSER detector finds a much smaller number of distinctive keypoints in an image in comparison with the DoG detector. Our local gradient representation combines the MSER detector and SIFT descriptor to exploit their complementary strengths.

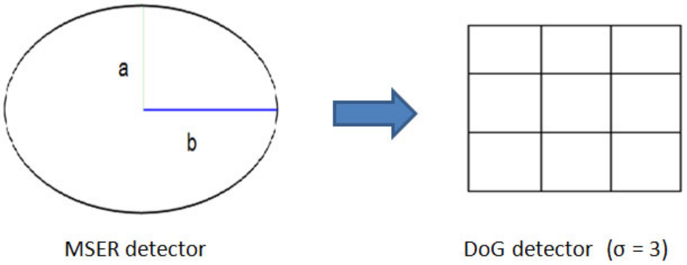


Figure 3.2: Transform MSER to DoG.

A DoG detector represents each detected image region using four parameters: (x, y, σ, θ) that denote location of the region, scale (σ) and orientation (θ). The MSER detector uses five parameters: (x, y, a, b, c) that denote the location, axes of ellipse representing the distinctive region (a, b), and orientation (c). The scale space for a DoG operator is:

$$L(x, y; \sigma) = G(x, y; \sigma) * I(x, y) \tag{3.1}$$

which convolves a variable-scale Gaussian $G(x, y; \sigma)$ with an input image $I(x, y)$. The parameter σ defines the range of the mask and hence determines the range of the detector. Figure 3.2 shows the regions of those two detectors. There are two options using the MSER representation of a keypoint to obtain the scale of the equivalent DoG detector:

$$\sigma = \begin{cases} K \cdot \sqrt{a^2 + b^2} & option1 \\ \max(a, b) & option2 \end{cases} \tag{3.2}$$

However, the orientation of MSER cannot be used for DoG which uses θ computed from an orientation histogram in the Gaussian smoothed image (Equation 3.1). The equivalent orientation in scale-space is hence computed after computing σ using $option1$ with $K = 1.3$ (obtained by automatic parameter tuning) because it provides best recognition accuracy–see [60]. To extract gradient features from a candidate image region, robots thus compute MSER keypoints, transform them to equivalent DoG keypoints and compute SIFT descriptors. In the text below, *gradient features* refers to MSER-SIFT features. These features are used to trigger autonomous learning of object models.

3.2 Candidate Image Region Selection

As stated earlier, once a domain map has been learned (with stationary objects), the interesting objects in many application domains are those that move. This dissertation investigates the automatic learning of models for such objects with the objective of promoting incremental learning with minimal human supervision. Image regions corresponding to such objects are identified using motion cues. Similar to the optical flow methods in computer vision [42], moving objects are detected by tracking the motion of gradient features in a short sequence of $(3 - 8)$ images. Consider the gradient features extracted from two consecutive images $\{I_{t-1}, I_t\}$ of a moving object:

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

where each feature ms is a $128D$ vector, pos is the feature's (x, y) position in the image, and F_{t-1} and F_t are the number of gradient features in I_{t-1} and I_t respectively. The gradient features in these two sets are matched using the Euclidean distance metric. The matched features are clustered based on their relative displacement between the images, motivated by the observation that characteristic features of an object 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 is defined by the convex hull [7] (i.e., minimal convex set) containing the matched features 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 the short sequence $(3 - 8$ images). In our object model, we also define the convex boundaries as the shape of the object. The other five components are described in the following section.

3.3 Layered Object Model

Once image regions corresponding to candidate objects have been found, local, global and contextual visual cues extracted from these regions are used to model the objects. In this work, objects are characterized by representations of gradient features, connection potentials between gradient features, image segments, color distributions, local context and shape, because they these visual cues have complementary strengths. For instance, gradient features are robust to scale, orientation, viewpoint and illumination, but neglect the global information in images and are not well-suited for some object surfaces. Connections between gradient features model the immediate neighborhood of gradient features. Color distributions provide a more global characterization of an object; they are not sensitive to surface texture but are sensitive to factors such as illumination. Similarly, a graph-based segmentation algorithm, e.g., [32], provides image segments that are substantially different from segments immediately around them; these segments can be used to define object parts and local context.

It is essential to incorporate an appropriate representation of these visual cues in order to exploit their complementary strengths. For instance, gradient features from a car's wheel may be similar to those from a wheel of another car. Color distributions of different regions may be similar and viewpoint changes can cause different features to be extracted for the same object. In this dissertation, the object model learned from a candidate image ROI consists of a representation of each of the visual cues, as shown in Figure 3.3. The object model has six components: (1) gradient features and their relative spatial arrangements; (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 arrangements; (4) color distributions and second-order image statistics; (5)

Gaussian mixture models and relative positions of image segments neighboring the ROI; and (6) convex hull of gradient features. The key aspect is that *learning is triggered by motion cues and accomplished automatically from a small number of images*. The individual components of the learned object model are described below in the context of an image ROI containing an object of interest.

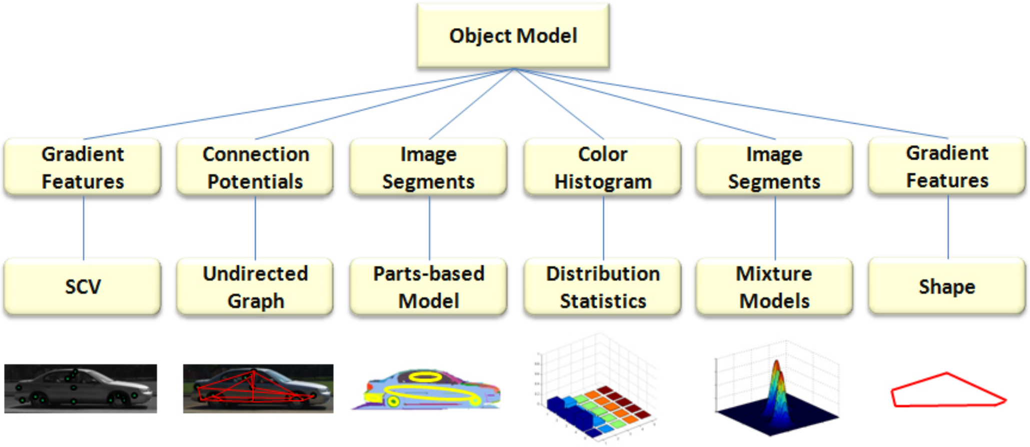


Figure 3.3: Object Model consists of five components: (1) gradient features and their relative spatial arrangement; (2) connection potentials between neighboring gradients and an undirected graph of neighborhood relationships; (3) image segments and parts-based model of relative spatial arrangement of segments; (4) color distributions and second-order image statistics; (5) Gaussian mixture model and relative positions of image segments neighboring the ROI; and (6) convex hull of gradient features.

3.3.1 Gradient-based Representation

Consider a specific ROI that is being used to build an object model. The extraction of gradient features from this ROI proceeds as described in Section 3.1. Similar to the color coherence vector for color histograms [44], the spatial arrangement of local gradient features corresponding to a specific object is captured using a *spatial coherence vector* (SCV) . The SCV computation is motivated by the observation that although the individual 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 spatial coherence of each gradient feature is defined as its position in the image relative to every other gradient feature extracted from the ROI. This relative coherence is computed separately along the x and y axes. If the object in the ROI is characterized by N gradient features, the SCV for the i^{th} feature is defined as:

$$\begin{aligned} SCV_{x,i} &= \{d^x_{i,1}, d^x_{i,2}, \dots, d^x_{i,N}\} \\ SCV_{y,i} &= \{d^y_{i,1}, d^y_{i,2}, \dots, d^y_{i,N}\} \end{aligned} \tag{3.3}$$

where $d^x_{i,j}$ is the relative position of feature i w.r.t feature j along the x-axis; $d^y_{i,j}$ is the corresponding relative position along the y-axis. For instance:

$$d^x_{i,j} = \begin{cases} 1 \quad \text{if } x_i > x_j \\ 0 \quad \text{if } x_i = x_j \\ -1 \quad \text{if } x_i < x_j \end{cases} \tag{3.4}$$

where x_i and x_j are the x coordinate values of feature i and j respectively in the image plane.

Consider the illustrative example in Figure 3.4, where three gradient features have been clustered based on velocity v. The SCVs of these features along the x and y axes are shown in Table 3.1 and Table 3.2.

	1	2	3
1	–	-1	-1
2	1	–	1
3	1	-1	–

Table 3.1: X-axis SCV

	1	2	3
1	–	1	1
2	-1	–	1
3	-1	-1	–

Table 3.2: Y-axis SCV

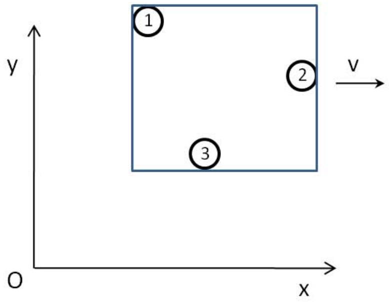


Figure 3.4: SCV computation for gradient features.

If the learned model for an object has M gradient features (each feature is a 128D vector), the model is augmented with a $2(M-1)$-dimensional vector for each feature, to represent the SCV along the x and y axes. The gradient features and SCV constitute one component of the object model, as shown in Figure 3.3.

3.3.2 Connection-based Representation

The second component of the object model captures the distribution of pixels between gradient features in the image ROI, as shown in Figure 3.5. The *connection potential* between two gradient features is computed as the distribution of pixels on the line joining the features in the image. The spread between any two features is normalized to unit distance for robustness to scale changes, and the pixel's 3D color values are collected in a histogram of 100 bins that is smoothed along each dimension using an impulse response filter:

$$C_n^{new} = \alpha C_n + (1-\alpha) C_{n-1} \tag{3.5}$$

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

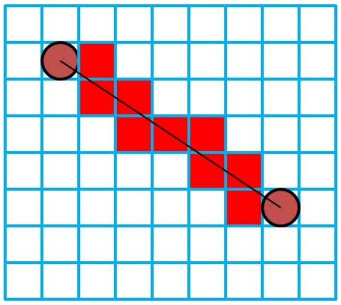


Figure 3.5: Pictorial representation of connection potential between gradient features.

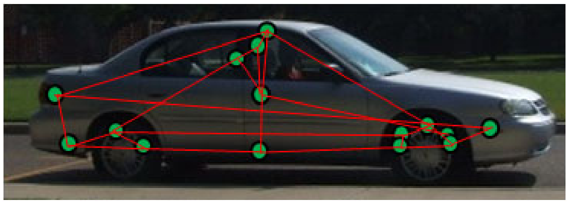
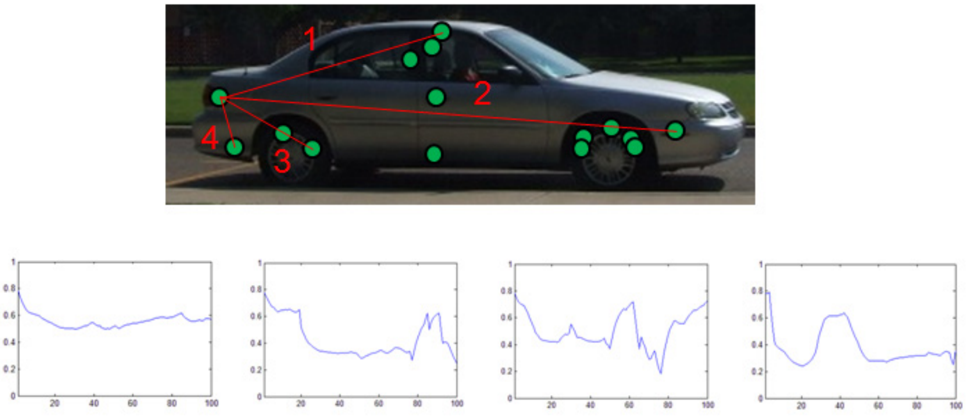


Figure 3.6: A car's undirected graph that represents the neighborhood relationships of connection potentials.

A connected neighborhood is then learned 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:

$$\{d_1, ..., d_{k-2}, d_{k-1}, d_k, d_{k+1}, d_{k+2}, ..., d_N\} \tag{3.6}$$

where $\forall i < j$, $d_i < d_j$. A connected neighborhood of a feature is defined as the four closest neighbors in the list, as shown in Figure 3.7. In Figure 3.7, we show one channel of 3D color values for each connection. In the algorithm described in this dissertation, we set α in Equation 3.5 to 0.2. The object model is augmented with an undirected graph that represents the neighborhood relationships of connection potentials between gradient features in the model.



(1) (2) (3) (4)

Figure 3.7: The connected neighborhood of a feature. $\alpha = 0.2$

3.3.3 Parts-based Representation

The third component of the object model considers the arrangement of object parts made up of image segments. Segments are extracted from the candidate ROI by applying a graph-based segmentation algorithm on the image [32]. This algorithm classifies the images into segments such that the RGB pixel values within a segment are similar to each other and dissimilar to pixels in the surrounding segments. Segments within the image ROI are then modeled as Gaussians that represent the locations of the segments within the ROI. These 2D Gaussian models constitute “parts” of the object in the ROI: $\mathcal{N}(\mu_k, \Sigma_k), k = 1, ..., P$. These parts and the list of parts connected to (i.e., that share a boundary with) each part are added to the object model. Probabilistic and heuristic constraints are used to filter spurious parts, e.g., segments that do not overlap significantly with the ROI under consideration ($\leq 40\%$) are ignored and the variance of pixel values within each part is checked to ensure part validity. In addition, significantly concave segments are ignored—other visual features extracted from these portions of the ROI will still contribute to the object model.

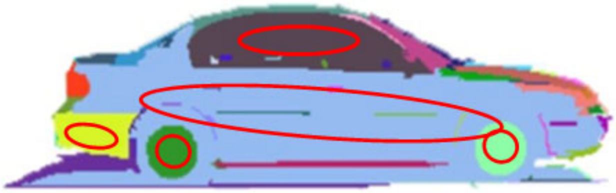


Figure 3.8: The Gaussian parts of a car.

Once the Gaussian parts of an object are extracted, two measures are computed to model the similarity of pixel values within each part ($PartSimM$) and the dissimilarity of pixels in neighboring parts ($PartDiffM$), as described in Algorithm 1 below.

Algorithm 1 considers the pixels that are in all the parts (i.e., Gaussians) computed in the ROI. First, each of the N pixels in the P parts learned from the ROI is assigned a label $lbp(\boldsymbol{x})$, i.e., membership in one of the parts, based on *a priori* probability density functions of the parts (lines 2–4). Next, for each pixel, lines 6–8 compute the similarity with other pixels within the same part, while lines 9–11 compute the dissimilarity with pixels in neighboring parts. Both $Sim(lbp(\boldsymbol{x}))$ and $Diff(lbp(\boldsymbol{x}))$ use function $RGB()$ to compute the difference in RGB values of two pixels, weighted by the probability that these pixels belong to the same part or different parts. The contributions of each pixel are summed up, and the similarity and dissimilarity measures are computed for each part ($PartSimM$, $PartDiffM$) as the logarithm of the summations (lines 14, 15).

Canonical values for the similarity and dissimilarity measures are computed for each part in the learned object model by considering local variations in the positions of the parts. As shown in Figure 3.9, the solid rectangle is an envelope around parts in the ROI,

Algorithm 1 Similarity + Dissimilarity of Object Parts.

```
Initialize $Sim$ and $Diff$ arrays.
for each ROI pixel $\boldsymbol{x}$ do
    $lbp(\boldsymbol{x}) = \arg\max_{1 \le j \le P} \; p(\boldsymbol{x}\,|\mu_j, \Sigma_j\,)$
end for
for each ROI pixel $\boldsymbol{x}$ do
    for each pixel $\boldsymbol{x}_{in}$ in part $lbp(\boldsymbol{x})$ do
        $Sim(lbp(\boldsymbol{x}))+ = \sum_{\Delta r, \Delta g, \Delta b} RGB(\boldsymbol{x}, \boldsymbol{x}_{in}) \cdot p(\boldsymbol{x}|\mu_{lbp(\boldsymbol{x})}, \Sigma_{lbp(\boldsymbol{x})}) \;\; p(\boldsymbol{x}_{in}|\mu_{lbp(\boldsymbol{x})}, \Sigma_{lbp(\boldsymbol{x})})$
    end for
    for each pixel $\boldsymbol{x}_{nhb}$ in neighboring parts of part $lbp(\boldsymbol{x})$ do
        $Diff(lbp(\boldsymbol{x}))+ = \sum_{\Delta r, \Delta g, \Delta b} RGB(\boldsymbol{x}, \boldsymbol{x}_{nhb}) \cdot p(\boldsymbol{x}|\mu_{lbp(\boldsymbol{x})}, \Sigma_{lbp(\boldsymbol{x})}) \;\; p(\boldsymbol{x}_{nhb}|\mu_{lbp(\boldsymbol{x}_{nhb})}, \Sigma_{lbp(\boldsymbol{x}_{nhb})})$
    end for
end for
for each part $j$ do
    $PartSimM_j = ln(Sim(j))$
    $PartDiffM_j = ln(Diff(j))$
end for
```

while the dotted rectangle is a local change in position. The values of $PartSimM$ and $PartDiffM$ computed over these local position changes are modeled as a Gamma distribution. Figure 3.10 shows such a Gamma pdf, which is used for recognition—see Section 3.4.3. The third component thus consists of image segments, parts-based model and measures of similarity (dissimilarity) within (between) parts.

3.3.4 Color-based Representation

The fourth component considers the color-based cues extracted from the ROI under consideration. The robot uses pixels in the ROI to build normalized histograms, i.e., color space pdfs, in the HSV color space that provides some robustness to minor illumination

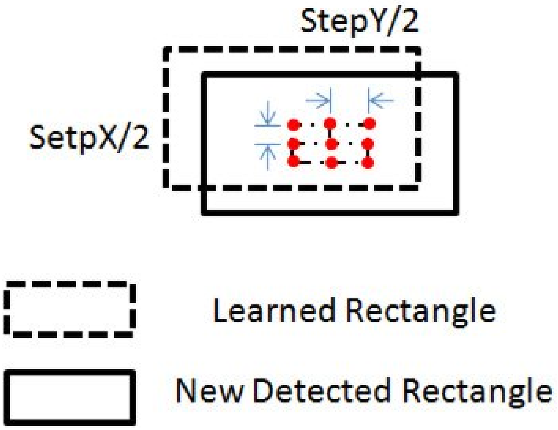


Figure 3.9: Modeling the similarity and dissimilarity between pixels in the parts-based model of an object.

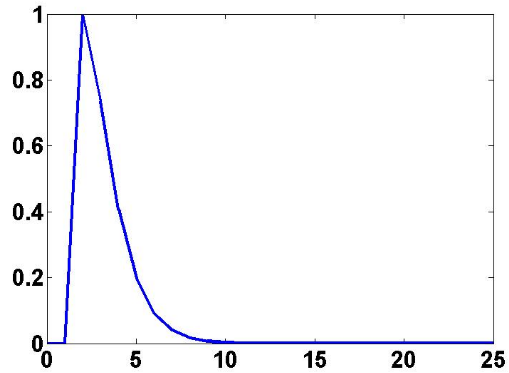


Figure 3.10: A Gamma pdf is used to represent pixel similarity (or dissimilarity) within (or between) parts.

changes. Pixel values are converted to HSV and normalized:

$$h = \frac{H/360}{H/360 + S + V} \qquad s = \frac{S}{H/360 + S + V} \tag{3.7}$$
$$v = \frac{V}{H/360 + S + V}$$

where hue (H), saturation (S) and value (V) are the dimensions of the color space. After normalization, any two of the three dimensions are a sufficient statistic for pixel values.

Each pdf is hence modeled as a normalized histogram in the (h, v) space, quantized into ten bins in each dimension.

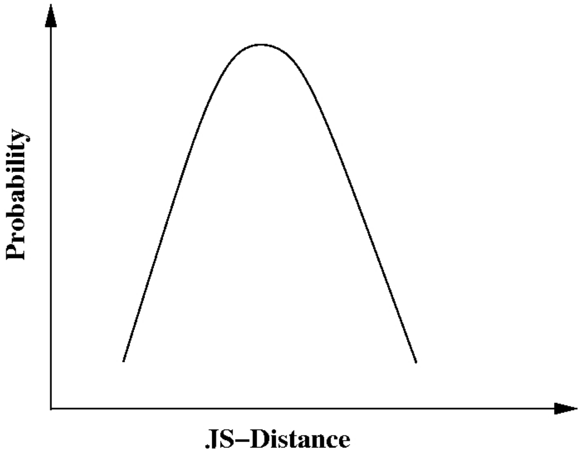


Figure 3.11: Distribution of distances between color space pdfs.

As stated earlier, color distributions are not a stable or unique representation for an object. Based on prior work by [112] on robots learning color distributions in the presence of illumination changes, the distance between every pair of pdfs is computed using the Jensen-Shannon (JS) measure—see [21]:

$$\begin{aligned} JS(a,b) &= \frac{KL(a,m)+KL(b,m)}{2} \\ KL(a,b) &= \sum_i \sum_j (a_{i,j} \cdot \ln \frac{a_{i,j}}{b_{i,j}}), \quad m = \frac{a+b}{2} \end{aligned} \tag{3.8}$$

where (a, b) are distributions (i.e., pdfs), m is a distribution obtained by averaging the two pdfs, and $KL()$ computes the KL-divergence measure between two distributions. The JS measure is robust to spurious peaks in the observed pdfs, e.g., due to large regions of a single color in the ROI. The distribution of distances models the variance in the color distributions; it is represented as a Gaussian—Figure 3.11. The learned pdfs and image statistics constitute the fourth component of the learned object model.

3.3.5 Context-based Representation

The fifth component models the object's *local context* using the subset of image segments (extracted in Section 3.3.3 for parts-based models) that share a boundary with the ROI. These segments correspond to image regions within the red rectangle but outside the yellow boundary in Figure 3.12. The pixels in each such neighboring segment are used to learn a 2D Gaussian in the normalized HSV color space (using only h, v). The relative spatial arrangement of each segment with respect to the object ROI is used to assign labels "above", "under" and "beside" to the segment, e.g., the label "under" implies that the segment is immediately below the object ROI as shown in Figure 3.13. An image segment can have more than one label, e.g., the segment for a tree may be "above" and "beside" the ROI for a car.

Segments with the same label are used to learn a Gaussian mixture model (GMM). For instance, to learn the GMM from K image segments with label "on", each of the K 2D Gaussians is assigned a mixing coefficient 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 w_{lbc} that is the ratio of number of pixels in segments with the corresponding label (lbc) to the number of pixels in all segments used to model context. The relative positions and sizes of these GMMs with respect to the ROI's center and size are also computed. The object model learned from the ROI includes the GMMs, and their relative positions and sizes.

3.3.6 Shape-based Representation

The sixth component of the object model is a convex polygons used as shape representation based on gradient features in the image ROI, as shown in Figure 3.14. The convex polygons is defined by convex hull [7], which is the smallest convex set that

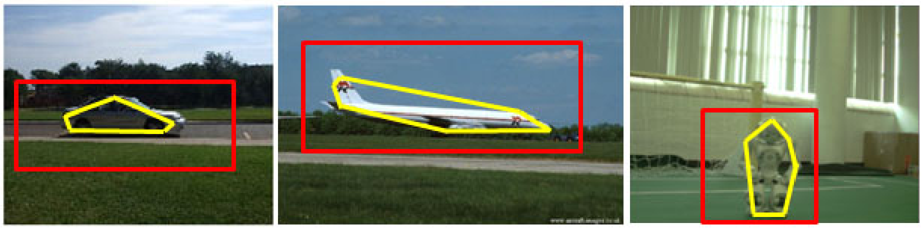
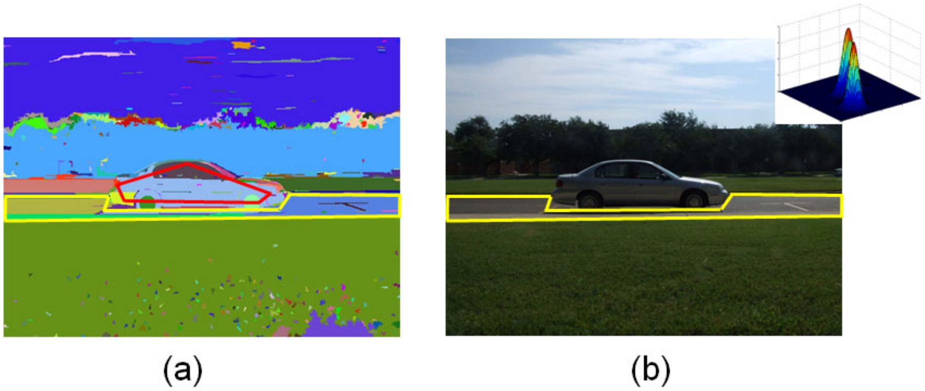


Figure 3.12: Local, global and temporal visual cues extracted from the yellow convex region represent appearance information. Mixture models and relative positions (e.g., “on” and “under”) of regions in the red rectangle (excluding the yellow polygon) represent context information.



(a)

(b)

Figure 3.13: The segments labeled ”on” for a car image.

contains gradient features in the ROI. This is a very simple representation for the shape.

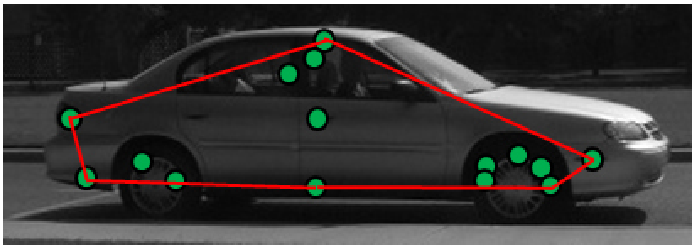


Figure 3.14: The convex boundaries for a car image.

In the model developed in this dissertation, our gradient-based representation and parts-based representation have not considered orientation changes. However, the object model, which exploits complementary strengths of different visual cues as described in this chapter, is invariant to scale and orientation changes.

3.4 Information Fusion and Matching Strategy

Consider the situation where the robot has autonomously learned models for one or more domain objects. These learned models are used to detect the corresponding objects in test images of novel scenes, *irrespective of whether the object is stationary or moving*. For a test image, the robot uses energy minimization to iteratively select ROIs for analysis, and uses belief revision (based on generative models) to merge the evidence provided by components of the learned models regarding the probability of occurrence of the corresponding objects in test image ROIs. We begin with the analysis of a specific test image ROI using components of the learned object models.

3.4.1 Gradient Feature-based Matching

Consider the use of gradient features to estimate the probability of occurrence of a learned object in the test image ROI. As stated in Section 3.1, an object model includes gradient features and the corresponding spatial coherence vector (SCV), which are used to estimate the probability of occurrence of the corresponding object in the ROI:

$$\begin{gathered} p_{ssm} = \frac{x_{correct} + y_{correct}}{2 * M}, \ p_{scg} \in [0,1] \\ x_{correct} = \sum_{m=1}^{M} \frac{x_{m,correct}}{N-1}, \quad y_{correct} = \sum_{m=1}^{M} \frac{y_{m,correct}}{N-1} \end{gathered} \tag{3.9}$$

where $x_{m,correct}$ and $y_{m,correct}$ 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 value of $p_{ssm} \in [0,1]$ is the probability of spatial match of two sets of gradient features. This computation is repeated with each learned object model to obtain the probability (distribution) of occurrence of learned objects in the ROI.

3.4.2 Connection-based Matching

As stated in Section 3.3.2, each learned object model includes connection potentials between gradient features and an undirected graph of local neighborhood relationships between potentials. The probability of occurrence of each learned object in the ROI is also computed by comparing the neighborhood of connection potentials between features in the learned model to the neighborhood of connection potentials between matched features in the ROI.Once the ROI's gradient features have 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 normalized distributions C_n^j and C_n^i that represent these connections in the ROI and learned object model respectively:

$$con(i,j) = \sum_{n=1}^{100} f(C_n^i, C_n^j) \tag{3.10}$$

$$f(a,b) = \begin{cases} 1 & |a-b| > \beta \\ 0 & otherwise \end{cases} \tag{3.11}$$

where parameter β identifies significant change in entries of the connection potentials. The probability of occurrence of the learned object is obtained by comparing the neighborhood of matched connection potentials in the test image ROI and learned object model:

$$p_{con} = \frac{1}{Z} \sum_{k \in \{1,\ldots,M\}} \sum_{i \in N_k, j \in N_{k_m}} con(i,j) \tag{3.12}$$

where M gradient features in the object model match the features in the ROI, N_{k_m} and N_k are the connected neighborhoods of feature k_m and matched feature k in the object model and ROI respectively, and Z is a normalizing factor. This computation is repeated with each learned object model to obtain the probability of occurrence of the corresponding object in the test image ROI.

3.4.3 Parts-based Matching

The parts-based representation in a learned object model can also be used to compute the probability of occurrence of the corresponding object in the test image ROI. As described in Section 3.3.3, an object model includes an arrangement of parts such that pixels within a part have similar values, while pixels in neighboring parts have dissimilar values. Unlike the gradient features and color features (below), image segments and parts are *not* extracted from the ROI in the test image. Instead, the arrangement of pixels in the parts of the learned object model is compared with pixels in the ROI.

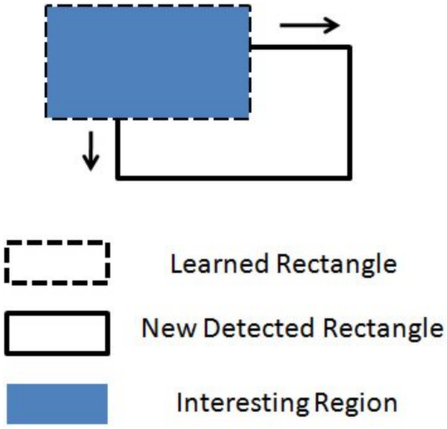


Figure 3.15: Illustration of the search for the best arrangement of learned parts-based model in the test image ROI.

Consider Figure 3.15, where the filled rectangle with dotted boundary represents the envelope around the parts in i^{th} learned object model, while the rectangle with the solid boundary represents the test image ROI. Different relative arrangements of the two rectangles are considered. For each such arrangement, the pixels in the overlapping region are extracted. If this relative arrangement corresponds to a good match, the subset of extracted pixels that lie in a learned part in the object model should have similar pixel values, and pixels in neighboring parts should have significantly different values. This similarity and dissimilarity can be evaluated using the $PartSimM$ and $PartDiffM$ measures, as described in Algorithm 1—the difference is that class labels of pixels, i.e., $lbp(\boldsymbol{x})$, are provided as input. In Section 3.3.3, the expected values of these measures were modeled as a Gamma distribution for each part of a learned object model. These Gamma distributions are used to evaluate the suitability of this arrangement:

$$\begin{aligned} p_{cdm} &= \sum_j \{w_j \cdot f(PartSimM_j) \cdot f(PartDiffM_j)\} \\ f(x_j) &= \Gamma\Big(|\overline{x_j} - x_j| - (k-1)\theta, k, \theta\Big) \end{aligned} \tag{3.13}$$

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, and $\overline{x_j}$ is the mean of the Γ distribution. 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 best arrangement of the two rectangles in Figure 3.15 is one that maximizes p_{cdm}. Repeating this computation with each learned object model provides the probability of occurrence of the corresponding object in the ROI.

3.4.4 Color-based Matching

Color space distributions extracted from the test image ROI (Section 3.3.4) are also used to compute the probability of occurrence of the learned objects in the ROI (p_{js}). As described in Section 3.3.4 in the context of learning object models, pixel values within the test image ROI are extracted to build a normalized color space histogram (i.e., a pdf). The average distance d_{avg} is computed between this pdf and the color space pdfs corresponding to the learned object model, using the JS distance measure described in Equation 3.8.

Each learned object model includes a Gaussian distribution of distances between the corresponding color space pdfs (i.e., a second order statistic), as shown in Figure 3.11 in Section 3.3.4. Comparing the computed average distance with this (Gaussian) distribution of distances for the learned object model provides p_{js}, the probability of occurrence of the corresponding object in the test image ROI. This computation is repeated with each learned object models to obtain the probability of occurrence of the corresponding learned object in the test image ROI. When the (Gaussian) distribution of distances is being learned incrementally for learned object models, it is still possible to use the relative values of average distances between the 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.

3.4.5 Context-based Matching

For a test image ROI, the probability of occurrence of each learned object is also computed by comparing the local context information in the learned model with the ROI's local context information. Each Gaussian mixture model (GMM) in the learned model (for labels: *above, under, beside*) is scaled and positioned suitably with respect to the test image ROI. A matching score is computed based on each GMM, considering the pixels around the convex boundary of test image 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 the individual scores:

$$
\begin{aligned}
p_{lc} &= \sum_{lbc \in \{above, under, beside\}} w_{lbc} \cdot \Gamma\Big(f(\boldsymbol{x}_{lbc}), k, \theta\Big) \\
f(\boldsymbol{x}_{lbc}) &= \frac{1}{N_{lbc}} \sum_{l=1}^{N_{lbc}} \sum_{j=1}^{N_{lbc}^{gmm}} \pi_j \, e^{-\frac{1}{2}(\boldsymbol{x}_l - \boldsymbol{\mu}_j)^T \Sigma_j^{-1} (\boldsymbol{x}_l - \boldsymbol{\mu}_j)}
\end{aligned}
\tag{3.14}
$$

where N_{lbc}^{gmm} is the number of 2D Gaussians in the GMM with label $lbc \in \{above, under, beside\}$. 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 lie in the corresponding GMM divided by number of pixels that lie in all GMMs in the learned model. Values of π_j, $\boldsymbol{\mu}_j$ and Σ_j are obtained from the learned model. This computation is repeated with each learned object model to compute the probability of occurrence of the corresponding object in the test image ROI.

3.4.6 Shape-based Matching

For a test image ROI, the probability of occurrence of each learned object is also computed by comparing the convex boundaries in the learned model with the ROI's convex boundaries. A matching score $score_{shape}$ is computed based on turning function [6]. The smaller value of $score_{shape}$ shows two sets of convex boundaries are matched better. The probability of occurrence of learned object is computed as below:

$$
p_{cb} = 1 - score_{shape}, \;\; score_{shape} \in [0, 1] \tag{3.15}
$$

3.4.7 Information Fusion

Sections 3.4.1-3.4.6 used the individual components of learned object models to compute the probability of occurrence of these objects in a test image ROI. This section describes energy minimization and belief revision algorithms for: (a) the identification of ROIs in test images; and (b) the fusion of evidence from individual components of learned object models regarding the presence of corresponding objects in these ROIs. For ease of explanation, assume that an ROI contains no more than one of the learned objects—the algorithm can detect multiple objects in an ROI or image.

If a test image sequence contains a moving object, the corresponding ROI is identified by tracking and clustering gradient features, as described in Section 3.2 in the context of learning object models. However, the sequence may consist of stationary objects or the test images may be snapshots of objects in different scenes. In such situations, the gradient features in a test image are compared with the gradient features in the learned object models to identify ROIs. Consider the computation of 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. In each iteration, M matched features (in test image) are selected using an energy minimization algorithm (described later in this section) and analyzed using generative models.

For a set of M matched (test image) features, the probability of occurrence of the i^{th} learned object(p_{O_i}) is computed as the product of the evidence provided by each of these features:

$$\begin{aligned} p_{O_i} &= \prod_{j \in \{1,...,M\}} p\Big(g_j | O_i, \{g_n | n = 1, ..., M, n \neq j\}\Big) \\ &= \prod_{j \in \{1,...,M\}} p(g_j | O_i) \end{aligned} \tag{3.16}$$

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

$$p(g_j | O_i) = \sum_{Lbg_j \in \{fg, bg\}} p(g_j | Lb_{g_j}, O_i) \cdot p(Lb_{g_j} | O_i) \tag{3.17}$$

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 (foreground or background) labels are assigned to candidate matched features, the test ROI is defined by the convex hull [7] (i.e., minimal convex set) containing the foreground features. The intuitive 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 3.17 is decomposed further using the independence relationships encoded by the joint probability distribution:

$$
\begin{aligned}
p(g_j|O_i) = & \sum_{Lbg_j \in \{fg,bg\}} p(g_j|Lb_{g_j}, O_i) \cdot p(Lb_{g_j}|O_i) \qquad (3.18)\\
= & \sum_{Lb_{g_j} \in \{fg,bg\}} p(g_j|Lb_{g_j}, ssm_{O_i}, con_{O_i}) \cdot \\
& p(Lb_{g_j}|cdm_{O_i}, js_{O_i}, lc_{O_i}) \\
= & \sum_{Lb_{g_j} \in \{fg,bg\}} p(g_j|Lb_{g_j}, ssm_{O_i}) \cdot p(g_j|Lb_{g_j}, con_{O_i}) \cdot \\
& p(Lb_{g_j}|cdm_{O_i}) \cdot p(Lb_{g_j}|js_{O_i}) \cdot p(Lb_{g_j}|lc_{O_i})
\end{aligned}
$$

This decomposition is based on the observation that parts-based models (CDM), color histograms (JS) and local contextual models (LC) capture visual cues that are more global in nature, and are not evaluated based on relative arrangements of local cues. These models can therefore 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., spatial arrangement of gradient features (SSM), neighborhood relationships of connection potentials (CON) and convex boundaries of gradient features (CB) are used to evaluate the probability of occurrence of the gradient feature given the specific label. These individual probabilities in the decomposed expression in Equation 3.18 are computed using Equations 3.9-3.15. The underlying independence assumptions work well in practice.

The ROI (among candidates being considered) that maximizes Equation 3.18 and hence Equation 3.16 is the best estimate of the corresponding object's location in the test image. Candidate ROIs are generated using the iterated conditional modes (ICM) energy minimization algorithm [115]. 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 to obtain the initial ROI estimate. Finally, the normalized probability distribution of occurrence of the learned objects in a test image is computed:

$$\overline{p}_{O_i} = \frac{p_{O_i}}{\sum\limits_{j=1}^{Q} p_{O_j}} \tag{3.19}$$

This distribution is used to recognize learned objects and detect novel objects when none of the learned objects has a match probability significantly larger than others.

The overall algorithm 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 of Algorithm 2, the robot considers the images captured at consecutive time-steps. The MSER-SIFT gradient features are extracted from these images and matched to arrive at the candidate object ROIs. If a valid object is detected, i.e., $validObject()$ returns $true$ in line 5, the robot extracts visual features to populate the four components of the object model. If prior learned object models exist, the robot attempts to match the new model with an existing one (line 7). If a close enough match is not found, the robot creates a new model corresponding to this object and increments the count of learned models (lines 9-12). However, if the learned model matches an existing object model with sufficiently high probability ($p_i > probThresh$ in Equation 3.19), the existing object model is augmented. Such incremental updates of the model can (in theory) cause a drift but it is not observed in practice. As long as there is a good match based on some subset of the components of learned models, the algorithm is able to recover from errors. To incrementally revise existing object models, the component CDS and GMC can be updated, and the component

Algorithm 2 Object Model Learning and Recognition

```
Require: : Ability to learn object models based on feature connections, gradient features,
    color distributions, color segment parts, local context and shape.
Require: Learned map of the surroundings for navigation.
Initialize: $numObjects = 0$ (no prior knowledge).
while true do
    if $modelLearn$ then
        Compute gradient features for $I_t$ and $I_{t-1}$.
        if $validObject()$ then
            Compute SCV, connection potentials, segment parts, color distribution statistics, GMMs and convex boundaries.
            if $(numObjects > 0)$ & $existModel()$ then
                Augment model of appropriate object.
            else
                $ComputeNewModel()$
                $numObjects = numObjects + 1$
            end if
        end if
    else
        Compute SCV, connection potentials, segment parts, color distributions, GMMs and convex boundaries from $I_t$.
        if $numObjects > 0$ then
            Compute match probabilities of learned models.
            Identify object in image.
        end if
    end if
end while
```

SCG, GCP and PIS can also be replaced based on image ROIs corresponding to recognized objects. Such revisions enable the robot to adapt to changes, e.g., update color models as the illumination changes slowly.

If the $modelLearn$ flag is turned off, the robot recognizes objects using the object models learned so far. As stated earlier, recognition can occur in an image sequence or a single image with stationary or moving objects. The robot identifies candidate ROIs and then computes probability of occurrence of different learned objects in each ROI. If a match with sufficiently high probability is found, the object is reported as being recognized in the corresponding test image ROI (lines 17-18). Although learning and recognition are separated in Algorithm 2 for ease of explanation, the robot concurrently learns object models and recognizes objects in multiple ROIs.

CHAPTER 4
EXPERIMENTAL SETUP AND RESULTS

This chapter describes the robot test platform and the results of evaluating the algorithms described in previous chapter.

4.1 Test Platform

The ERA-MOBI robot (a.k.a "erratic") from Videre Design is used as the test platform—see Figure 4.1. 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. 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, corridors and outdoor settings.

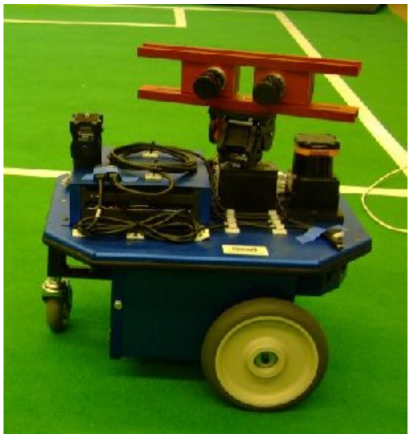


Figure 4.1: Robot test platform: “Erratic”.

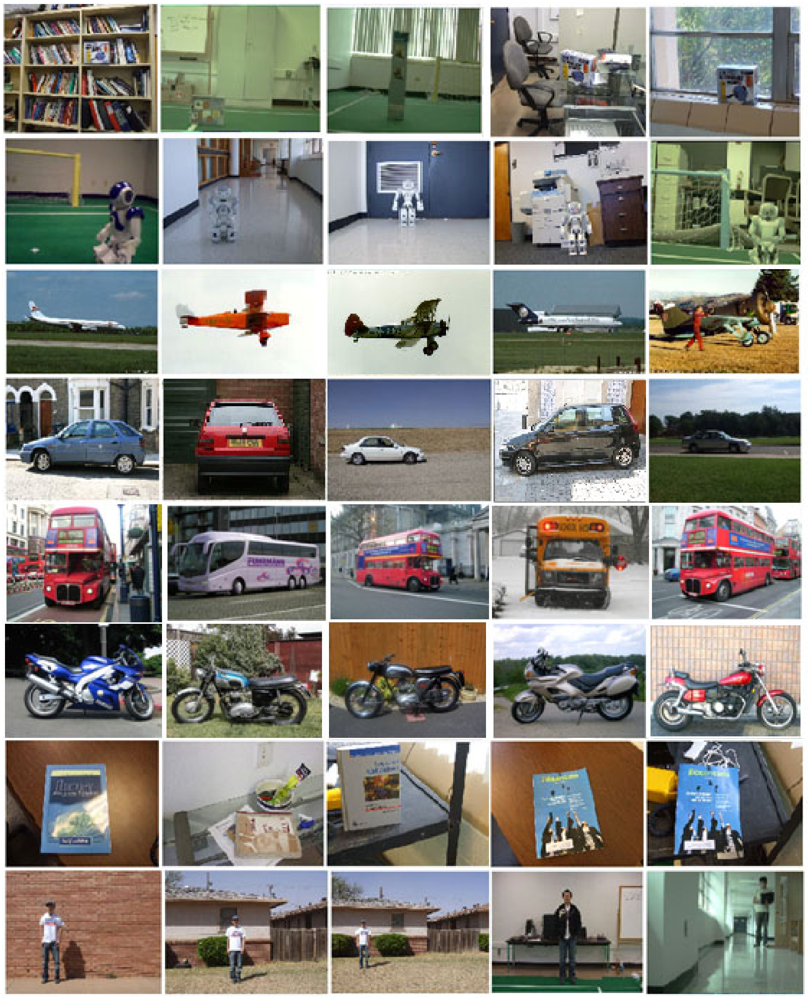


Figure 4.2: Examples of objects from eight object categories.

4.2 Experimental Setup

It is challenging 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 considered image segments neighboring the ROIs for contextual cues. To make learning challenging, each object model is learned from $3-8$ images which are randomly selected, with ≈ 250 images used for learning all object models; remaining images are used for testing. Test images consist of short sequences of objects in motion and snapshots of objects in indoor and outdoor scenes. For stationary objects, we provided labeled training samples. We have repeated the experiments for 10 times to collect the experiment data. Therefore, the results reported in this chapter are based on testing the algorithms on ≈ 20000 images. The robot processed $3-5$ frames/second to identify ROIs, learn models and recognize objects while performing other operations such as navigation and mapping. The images used for learning and recognition were chosen randomly (in repeated trials) to obtain the results below.

4.3 Experimental Results

The algorithm is successful if the robot can achieve the following objectives:

1. Learn object models from a small $(3-8)$ number of images for moving objects.

2. Exploit complementary strengths of appearance-based and contextual visual cues to efficiently learn representative models of these objects from relevant image regions.

3. Use 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 with minimal human supervision.

Learning the desired object models in both indoor and outdoor environment is tested in our experiments. In addition, the time consumption is measured. In the last, we compared the accuracy and the time consumption with existing algorithms.

Ten object categories were used in the experiments: human, box, airplane, book, car, motorbike, bus, humanoid robot, fire truck and fire hydrant—Figure 4.2 shows examples. The "box", "robot" and "book" categories were evaluated in indoor domains, "human" category was evaluated in outdoor and indoor domains, and other categories were used for outdoor trials. Separate models were learned for different objects within a category, e.g., different boxes, books or humans, resulting in 40 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 others (e.g., boxes) were moved on trolleys. Robot iteratively computes the values of the small set of parameters in the algorithm, e.g., clusters with at least 15 matched gradient features are considered to be candidate objects.

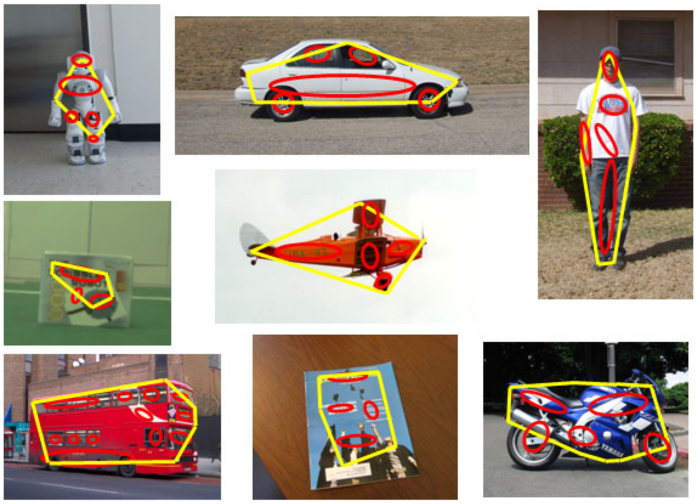
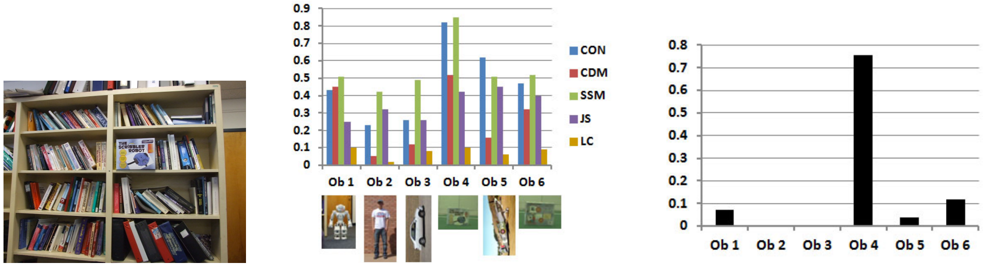


Figure 4.3: Images with ROIs, parts and local context regions.

In our experiments, we do not show the results of testing object model with the shape component. Because we found that our shape component does not significantly improve recognition accuracy. The false case happens when one or more background features in the test image are matched with the corresponding learned object model. In that case, the convex boundaries in the test images may become very different with the learned model and that may lead to a very low shape matching probability, which should be high for the same objects. Therefore, the defined shape component is proved to be unstable. We just show the results of testing the rest of five components in the object model in this chapter.

The test images consist of short sequences of objects in motion and images of objects in different indoor and outdoor scenes. Figure 4.3 shows examples of ROIs, parts and local context regions extracted from some images. Next, Figure 4.4(a) shows a challenging test image of a box on a book-shelf. Figure 4.4(b) shows the match probabilities (using components of the object models) for the top two sub-categories within the "box" category and the closest match within four other categories. As seen in Figure 4.4(c), merging match probabilities results in robust recognition of the box in the test image. Figures 4.5(a)–4.5(c) illustrate the use of local context for better disambiguation. Without contextual information, an airplane is recognized as a car, but including context-based models enables the robot to disambiguate cars and airplanes.

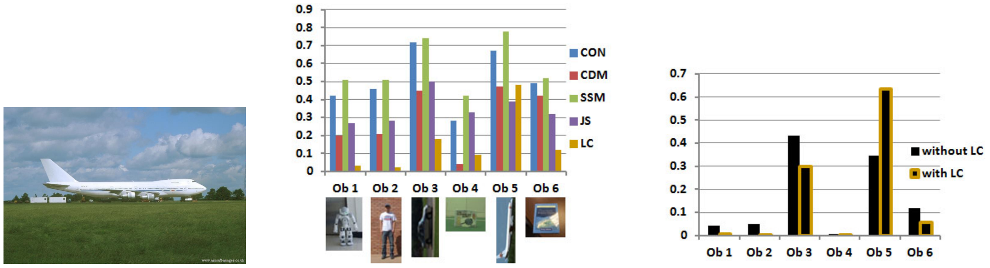


(a) Test Image.

(b) Match Probabilities.

(c) Net Match.

Figure 4.4: Test image: (a) A box in a complex background; (b) Individual match probabilities for the appropriate ROI—the top two subcategories in "box" category and best matches for four other categories are shown along x-axis; (c) Merging the individual probabilities results in robust recognition.



(a) Test Image.

(b) Match Probabilities.

(c) Net Match.

Figure 4.5: Test image: (a) An airplane on a field; (b) Individual match probabilities for the appropriate ROI; (c) Merging the individual probabilities results in robust recognition—local context information plays an important role in disambiguation.

Next, Figure 4.6 compares the average recognition accuracy of our algorithm with that of each component used individually ("Individual component") and different subsets of four components ("All components except") included in the object model. We observe that none of the individual components can provide high recognition accuracy because they are unable to fully exploit the information encoded by different visual cues. In addition, there is large variance in the recognition accuracy provided by each component, especially with components that primarily use color (and color-based) cues.

Figure 4.6 also shows that each component of the object model contributes to the overall recognition accuracy. The recognition accuracy of our algorithm is better than that of different subsets of four components. In addition, the variance is observed to be larger when spatial and local cues (e.g., spatial arrangements of local features) are not considered. These results indicate that although each component uses visual cues widely used by many other algorithms, our representation better exploits their complementary strengths to learn representative object models that provide high recognition accuracy.

The average classification accuracy over all 40 subcategories in 10 object categories is: 0.8860 ± 0.0432, which is promising given the small number of images used for learning.

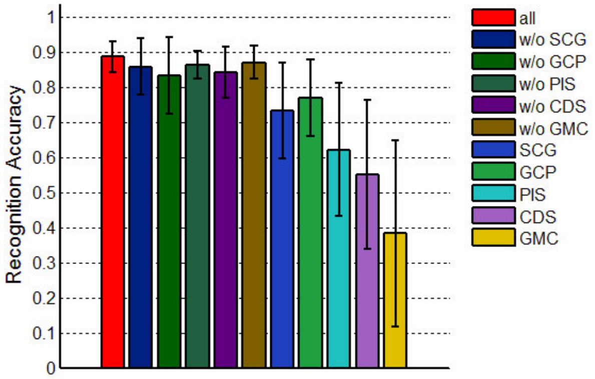


Figure 4.6: The match probabilities obtained with each component of the object model, averaged over subcategories in each object category. Exploiting complementary strengths of the components provides reliable object recognition.

Table 4.1 shows classification accuracy for the different object categories, averaged over the different object models (i.e., subcategories) in each category. The classification is correct only if an object in the test image is matched to the correct model—matching an object in *car-class1* to learned model *car-class2* is incorrect. The off-diagonal terms represent errors. Our prior experiments [61] indicated that one reason for the classification errors is the learning of object models with non-unique features, e.g., long shots of the

	Box	Car	Human	Robot	Book	Airplane	Bus	Motorbike	Fire Truck	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
Motorbike	0	0.073	0	0.010	0.016	0	0.062	**0.839**	0	0
Fire Truck	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 4.1: Recognition accuracy averaged over different models (i.e., subcategories) in ten (i.e., a subset of) object categories.

"human" category cause features to be extracted from clothes, resulting in non-unique object models and lower object recognition accuracy. However, augmenting the object model with models of context and other appearance features significantly reduces such errors. Some of the recognition errors in the current system correspond to an insufficient number of test image features being matched with the learned models due to motion blur or a substantial difference in scale or viewpoint. As stated earlier, the robot autonomously learns the object models used in these experiments from a small number of images (to make learning and recognition challenging). Revision of object models over times (as stated in Algorithm 2) further improves recognition accuracy. Another reason for errors is 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. These errors can be eliminated by requiring that the maximum match probability be substantially higher than the match probabilities of other object classes. In addition, errors are less frequent in image sequences of objects in motion because identifying the ROI properly enables one or more components to provide high match probabilities for the appropriate object. These experimental results indicate that the robot is able to autonomously learn object models using appearance-based and contextual visual cues, and use the learned models for robust recognition in novel indoor and outdoor scenes.

Our algorithm and existing vision algorithms have disparate objectives; our algorithm efficiently learns representative models of relevant objects using $3 - 8$ images (each), while existing algorithms typically focus on modeling a large number of objects and use a much larger number of images for training or learning representative models of each object. Although finding a common frame of reference is challenging, 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, existing algorithms are much more (computationally) expensive for learning and/or recognition, and very few algorithms support the incremental learning capability provided by our algorithm. Furthermore, it is difficult for existing algorithms to learn good models from a small number of images because they do not fully exploit the complementary strengths of (and dependencies between) different cues.

We also compared the recognition accuracy and efficiency of our algorithm with state of the art algorithms that use gradient features, e.g., SURF and BRIEF as discussed in Section 2.1. We provided labeled training samples (i.e., images with labeled ROIs) for SURF and BRIEF to learn object models—the learned models were then used for object recognition. During learning, these algorithms extract local image gradient features from the ROIs to create models for the corresponding objects. For recognition, features in the learned models were matched with features extracted in the test images. Table 4.2 shows that our algorithm provides much higher accuracy than these algorithms, primarily because our algorithm exploits the complementary strengths and dependencies between local, global, temporal and contextual visual cues. At the same time, the use of multiple components does increase the computational cost—Table 4.3 shows that SURF and BRIEF are more efficient. We believe that this trade-off is justified since it supports incremental learning of good object models from a small number of images.

Finally, we compared our algorithm with the algorithm developed by [3] that defines an *objectness* measure to automatically identify image windows containing objects of any learned class; the objectness-based algorithm thus shares one of the objectives of our

	Box	Car	Human	Robot	Book	Airplane	Bus	Motorbike	Fire Truck	Firehydrant
Proposed	**0.941**	**0.917**	**0.820**	**0.899**	**0.942**	**0.888**	**0.856**	**0.839**	**0.872**	**0.884**
SURF	0.804	0.784	0.706	0.822	0.832	0.742	0.713	0.772	0.754	0.793
BRIEF	0.843	0.822	0.743	0.855	0.843	0.772	0.733	0.813	0.782	0.834

Table 4.2: Our algorithm provides higher accuracy than SURF and BRIEF using the same number of image for learning the object models.

	SURF	BRIEF	Proposed	Objectness
Learning	0.1	0.005	0.3	360
Testing	0.12	0.01	0.25	5

Table 4.3: Computation time in seconds

algorithm. However, our algorithm efficiently learns representative models of relevant objects using $3 - 8$ images (each), while the objectness-based algorithm (similar to many other computer vision algorithms) typically focuses on modeling a large number of objects and use a much larger number of images for training or learning representative models of each object. We compared our algorithm and the algorithm based on objectness measure on the basis of recognition accuracy and computational efficiency. Compared with the objectness-based algorithm, our algorithm is significantly more efficient—see last column in Table 4.3. Figure 4.7 compares the recognition accuracy of the two algorithms as a function of the number of images used for learning object models. The images not used for learning object models are used for evaluation, and the experiments are repeated multiple times to obtain the results shown in Figure 4.7. The objectness measure-based algorithm does not fully exploit all visual cues and it requires objects to be much more distinct from the background. As a result, our algorithm provides a much higher recognition accuracy using a much smaller number of images for learning object models.

In summary, we have experimentally evaluated the robot's ability to:

1. Learn object models from a small $(3 - 8)$ number of images for moving objects.

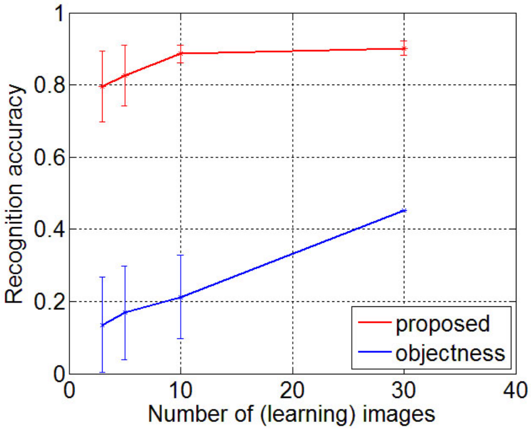


Figure 4.7: Our algorithm provides higher recognition accuracy than the *objectness*-based algorithm while using a much smaller number of image for learning the object models.

2. Exploit complementary strengths of appearance-based and contextual visual cues to efficiently learn representative models of these objects from relevant image regions.

3. Use 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 with minimal human supervision.

The algorithm described in this dissertation thus achieves the desired objectives, supporting incremental learning and enabling robots to acquire and use visual inputs and human feedback based on need and availability.

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

For widespread deployment in complex real-world application domains, mobile robots need the ability to make best use of sensor inputs and human feedback based on need and availability. In the context of visual object recognition, robots therefore need the ability to incrementally, reliably and efficiently learn representative models of relevant domain objects from a small number of images. In this dissertation, robots use motion cues extracted from short image sequences to automatically identify candidate image regions corresponding to interesting domain objects. The object models learned from these image regions exploit the complementary strengths of local, global and contextual visual cues extracted from the image regions. Specifically, object models consist of relative spatial arrangements of gradient features, graph-based models of neighborhoods of gradient features, parts-based models of image segments, color distribution statistics, and mixture models of local context. The learned object models are used by the robot in energy minimization algorithms and probabilistic generative models of information fusion, thus recognizing the corresponding objects in novel scenes. Experimental results show that our algorithm enables robots to incrementally, reliably and efficiently learn object models and recognize objects in indoor and outdoor domains. Our algorithm thus satisfies the objectives appropriate for robot application domains, and opens up multiple directions for future research.

The images used for experimental evaluation had a small number of moving objects in any given image that did not significantly occlude each other while moving. Future research will investigate the use of image sequences with multiple moving objects. The object model will also be expanded to include visual cues corresponding to shape and depth information (e.g., RGB-D cameras); this information (especially depth) will enable robots to disambiguate between partially occluded objects. In addition, our algorithm currently depends on the tracking of gradient features to identify image regions corresponding to objects of interest. Future research will also consider other image features, learning unique models for object categories with within-category similarity based on a subset of visual features.

Although the computational efficiency of our algorithm is substantially better than that of existing algorithms, future research will focus on improving computational efficiency. Currently, the computationally expensive portions of the algorithm include the learning (and use) of the parts-based models, and the use of energy minimization algorithms for iteratively analyzing image ROIs. We are investigating the use of sampling-based algorithms and more efficient energy minimization algorithms, and we are considering sampling-based methods for improving computational efficiency. Furthermore, we are developing an algorithm that will consider the dependencies between the components of the object model to incrementally and automatically determine the most informative subset of 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.

BIBLIOGRAPHY

[1] Amit Adam, Ehud Rivlin, and Ilan Shimshoni. Robust fragments-based tracking using the integral histogram. In *Computer Vision and Pattern Recognition*, pages 798–805, 2006.

[2] Shivani Agarwal, Aatif Awan, and Dan Roth. Learning to Detect Objects in Images via a Sparse, Part-Based Representation. *Pattern Analysis and Machine Intelligence*, 26:1475–1490, 2004.

[3] Bogdan Alexe, Thomas Deselaers, and Vittorio Ferrari. Measuring the Objectness of Image Windows. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2189–2202, 2012.

[4] Y. Amit and D. Geman. A computational model for visual selection. *Neural computation*, 11(7):1691–1715, 1999.

[5] 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.

[6] Esther M. Arkin, L. Paul Chewi, Daniel P. Huttenlocher, Klara Kedemt, and Joseph S. B. Mitchell. An efficiently computable metric for comparing polygonal shapes. In *ACM-SIAM Symposium on Discrete Algorithms*, pages 129–137, 1990.

[7] 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.

[8] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Computer vision and image understanding*, 110(3):346–359, 2008.

[9] E. Bayro-Corrochano and C. Lopez-Franco. Invariants and Omnidirectional Vision for Robot Object Recognition. In *International Conference on Intelligent Robots and Systems*, 2005.

[10] Serge J. Belongie, Jitendra Malik, and Jan Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:509–522, 2002.

[11] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer-Verlag New York, 2008.

[12] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast Approximate Energy Minimization via Graph Cuts. In *International Conference on Computer Vision*, pages 377–384, 1999.

[13] Timothy F Brady, Talia Konkle, and George A Alvarez. A review of visual memory capacity: Beyond individual items and toward structured representations. *Journal of Vision*, 11(5), 2011.

[14] M. Burl, M. Weber, and P. Perona. A probabilistic approach to object recognition using local photometry and global geometry. *Computer VisionECCV98*, pages 628–641, 1998.

[15] M. Calonder, V. Lepetit, M. Ozuysal, T. Trzcinski, C. Strecha, and P. Fua. BRIEF: Computing a Local Binary Descriptor Very Fast. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(7):1281–1298, 2012.

[16] 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.

[17] J. Casper and R. R. Murphy. Human-robot Interactions during Urban Search and Rescue at the WTC. In *Systems, Man and Cybernetics, Part B*, volume 33, pages 367–385, 2003.

[18] Vijay Chandrasekhar, Gabriel Takacs, David M. Chen, Sam S. Tsai, Yuriy Reznik, Radek Grzeszczuk, and Bernd Girod. Compressed Histogram of Gradients: A Low-Bitrate Descriptor. *International Journal of Computer Vision*, 96:1–16, 2012.

[19] Winston Churchill and Paul Newman. Practice makes perfect managing and leveraging visual experiences for lifelong navigation. In *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, Minnesota, USA, May 2012.

[20] Timothy F. Cootes, Christopher J. Taylor, David H. Cooper, and Jim Graham. Active Shape Models-Their Training and Application. *Computer Vision and Image Understanding*, 61:38–59, 1995.

[21] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory, 2nd Edition*. Wiley-Interscience, 2006.

[22] W Davidson and M Abramowitz. Molecular expressions microscopy primer: Digital image processing-difference of gaussians edge enhancement algorithm. *Olympus America Inc., and Florida State University*, 2006.

[23] Silvano Di Zenzo. A note on the gradient of a multi-image. *Computer Vision, Graphics, and Image Processing*, 33(1):116–125, 1986.

[24] Santosh Kumar Divvala, Derek Hoiem, James H Hays, Alexei A Efros, and Martial Hebert. An empirical study of context in object detection. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1271–1278. IEEE, 2009.

[25] Wei Du and Justus H. Piater. *A Probabilistic Approach to Integrating Multiple Cues in Visual Tracking*. 2008.

[26] S. Ekvall, P. Jensfelt, and D. Kragic. Integrating Active Mobile Robot Object Recognition and SLAM in Natural Environments. In *International Conference on Intelligent Robots and Systems*, 2006.

[27] Gal Elidan, Geremy Heitz, and Daphne Koller. Learning Object Shape: From Drawings to Images. In *Computer Vision and Pattern Recognition*, volume 2, pages 2064–2071, 2006.

[28] Karin Engel and Klaus D. Toennies. Hierarchical vibrations for part-based recognition of complex objects. *Pattern Recognition*, 43:2681–2691, 2010.

[29] A. Ess, B. Leibe, K. Schindler, and L. van Gool. Moving Obstacle Detection in Highly Dynamic Scenes. In *International Conference on Robotics and Automation*, 2009.

[30] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *Computer Vision and Image Understanding*, 106(1):59–70, 2007.

[31] Pedro F. Felzenszwalb, Ross B. Girshick, David A. McAllester, and Deva Ramanan. Object Detection with Discriminatively Trained Part-Based Models. *Pattern Analysis and Machine Intelligence*, 32:1627–1645, 2010.

[32] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181, 2004.

[33] 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.

[34] Vittorio Ferrari, Frdric Jurie, and Cordelia Schmid. From Images to Shape Models for Object Detection. *International Journal of Computer Vision*, 87:284–303, 2010.

[35] Sanja Fidler, Marko Boben, and Ales Leonardis. Similarity-based Cross-Layered Hierarchical Representation for Object Categorization. In *The International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[36] M. Fink and P. Perona. Mutual Boosting for Contextual Inference. In *Neural Information Processing Systems*, 2003.

[37] Graham D Finlayson, Subho S Chatterjee, and Brian V Funt. Color angular indexing. In *Computer VisionECCV'96*, pages 16–27. Springer, 1996.

[38] Francois Fleuret and Donald Geman. Coarse-to-fine face detection. *International Journal of computer vision*, 41(1-2):85–107, 2001.

[39] Peter V. Gehler and Sebastian Nowozin. On feature combination for multiclass object classification. In *International Conference on Computer Vision*, pages 221–228, 2009.

[40] Theo Gevers and Arnold W.M. Smeulders. Color-based object recognition. *Pattern Recognition*, 32(3):453 – 464, 1999.

[41] Theo Gevers and Harro Stokman. Robust histogram construction from color invariants for object recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(1):113–118, 2004.

[42] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Prentice Hall, 2002.

[43] Michael A. Goodrich and Alan C. Schultz. Human-Robot Interaction: A Survey. *Foundations and Trends in Human-Computer Interaction*, 1(3):203–275, 2007.

[44] P. Greg, Z. Ramin, and M. Justin. Comparing Images Using Color Coherence Vectors. In *ACM International Conference on Multimedia*, 1997.

[45] 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.

[46] Robert M Haralock and Linda G Shapiro. *Computer and robot vision*. Addison-Wesley Longman Publishing Co., Inc., 1991.

[47] Jesse Hoey, Pascal Poupart, Axel Bertoldi, Tammy Craig, Craig Boutilier, and Alex Mihailidis. Automated Handwashing Assistance for Persons with Dementia using Video and a Partially Observable Markov Decision Process. *Computer Vision and Image Understanding*, 114(5):503–519, 2010.

[48] D Hoffman and W.A. Richards. Parts of recognition. *Cognition*, 18:65–96, 1984.

[49] Derek Hoiem, Alexei A. Efros, and Martial Hebert. Putting Objects in Perspective. In *Computer Vision and Pattern Recognition*, volume 2, pages 2137–2144, 2006.

[50] Hongwen Kang, Martial Hebert, and Takeo Kanade. Discovering Object Instances from Scenes of Daily Living. In *International Conference on Computer Vision*, Barcelona, 2011.

[51] G.J. Klinker, S.A. Shafer, and T. Kanade. A physical approach to color image understanding. *International Journal of Computer Vision*, 4(1):7–38, 1990.

[52] M. Kobayashi and K. Kameyama. A composite illumination invariant color feature and its application to partial image matching. *IEICE TRANSACTIONS on Information and Systems*, 95(10):2522–2532, 2012.

[53] 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.

[54] Yves Berube Lauziere, Denis J Gingras, and Frank P Ferrie. Autonomous physics-based color learning under daylight. In *Industrial Lasers and Inspection (EUROPTO Series)*, pages 86–100. International Society for Optics and Photonics, 1999.

[55] Quoc V. Le, Marc'Aurelio Ranzato, Rajat Monga, Matthieu Devin, Kai Chen, Greg S. Corrado, Jeffrey Dean, and Andrew Y. Ng. Building High-Level Features using Large Scale Unsupervised Learning. In *The Twenty-Ninth International Conference on Machine Learning*, 2012.

[56] Juhyun Lee. *Robust Color-based Vision for Mobile Robots*. PhD thesis, Computer Science Department, The University of Texas at Austin, TX, December 2011.

[57] B. Leibe, A. Leonardis, and B. Schiele. Robust Object Detection with Interleaved Categorization and Segmentation. *International Journal of Computer Vision*, 77(1-3):259–289, 2008.

[58] Congcong Li, Devi Parikh, and Tsuhan Chen. Extracting Adaptive Contextual Cues from Unlabeled Regions. In *International Conference on Computer Vision (ICCV)*, Barcelona, Spain, November 6-13 2011.

[59] 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.

[60] 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.

[61] Xiang Li, Mohan Sridharan, and Shiqi Zhang. Autonomous Learning of Vision-based Layered Object Models on Mobile Robots. In *International Conference on Robotics and Automation*, Shanghai, China, May 9-13 2011.

[62] D. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[63] 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.

[64] 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.

[65] Hirokazu Madokoro, Yuya Utsumi, and Kazuhito Sato. Unsupervised scene classification based on context of features for a mobile robot. In Andreas Knig, Andreas Dengel, Knut Hinkelmann, Koichi Kise, RobertJ. Howlett, and LakhmiC. Jain, editors, *Knowledge-Based and Intelligent Information and Engineering Systems*, volume 6881 of *Lecture Notes in Computer Science*, pages 446–455. Springer Berlin Heidelberg, 2011.

[66] T. Malisiewicz, A. Gupta, and A.A. Efros. Ensemble of Exemplar-SVMs for Object Detection and Beyond. In *IEEE International Conference on Computer Vision*, pages 89–96, 2011.

[67] J. Matas, O. Chum, M.Urban, and T. Pajdla. Robust Wide Baseline Stereo from Maximally Stable Extremal Regions. In *British Machine Vision Conference*, 2002.

[68] K. Mikolajczyk and C. Schmid. A Performance Evaluation of Local Descriptors. In *Pattern Analysis and Machine Intelligence*, 2007.

[69] Krystian Mikolajczyk and Cordelia Schmid. An affine invariant interest point detector. In *Computer VisionECCV 2002*, pages 128–142. Springer, 2002.

[70] Krystian Mikolajczyk and Cordelia Schmid. Scale and Affine Invariant Interest Point Detectors. *International Journal of Computer Vision*, 60(1):63–86, 2004.

[71] Anuj Mohan, Constantine Papageorgiou, and Tomaso Poggio. Example-Based Object Detection in Images by Components. *Pattern Analysis and Machine Intelligence*, 23:349–361, 2001.

[72] Farzin Mokhtarian and Alan Mackworth. Scale-based description and recognition of planar curves and two-dimensional shapes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (1):34–43, 1986.

[73] Frank Moosmann, Bill Triggs, and Frederic Jurie. Fast Discriminative Visual Codebooks using Randomized Clustering Forests. In *The Neural Information Processing Systems (NIPS)*, 2006.

[74] Hans P Moravec. Obstacle avoidance and navigation in the real world by a seeing robot rover. Technical report, DTIC Document, 1980.

[75] 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.

[76] Kevin Murphy, Antonio Torralba, and William T. Freeman. Using the Forest to See the Trees: A Graphical Model Relating Features, Objects, and Scenes. In *Neural Information Processing Systems*, 2004.

[77] Philippe Noriega, Benedicte Bascle, and Olivier Bernier. Local kernel color histograms for background subtraction. In *International Conference on Computer Vision Theory and Applications*, volume 219. Setfbal, Portugal, 2006.

[78] Philippe Noriega and Olivier Bernier. Real time illumination invariant background subtraction using local kernel histograms. *British Machine Vision Association (BMVC)*, pages 567–580, 2006.

[79] 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.

[80] Aude Oliva and Antonio Torralba. The role of context in object recognition. *Trends in Cognitive Sciences*, 11:520–527, 2007.

[81] B. Ommer and J.M. Buhmann. Learning the compositional nature of visual object categories for recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(3):501–516, 2010.

[82] Andreas Opelt, Axel Pinz, and Andrew Zisserman. A Boundary-Fragment-Model for Object Detection. In *European Conference on Computer Vision*, pages 575–588, 2006.

[83] Patrick Ott and Mark Everingham. Shared parts for deformable part-based models. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1513–1520. IEEE, 2011.

[84] D. Parikh, C. L. Zitnick, and Tsuhan Chen. Unsupervised Learning of Hierarchical Spatial Structures in Images. In *International Conference on Computer Vision and Pattern Recognition*, 2009.

[85] 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.

[86] Devi Parikh and Kristen Grauman. Interactively Building a Discriminative Vocabulary of Nameable Attributes. In *International Conference on Computer Vision and Pattern Recognition*, June 20-25 2011.

[87] Marco Pedersoli, Andra Vedaldi, and Jordi Gonzalez. A coarse-to-fine approach for fast deformable object detection. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1353–1360. IEEE, 2011.

[88] Justus Piater, Sebastien Jodogne, Renaud Detry, Dirk Kraft, Norbert Kruger, Oliver Kroemer, and Jan Peters. Learning Visual Representations for Perception-Action Systems. *International Journal of Robotics Research*, pages 1–14, October 2010.

[89] Jean Ponce, David Forsyth, Equipe-projet Willow, Sophia Antipolis-Méditerranée, Rapports d'activité RAweb, Logo Inria, and Inria Alumni. Computer vision: a modern approach. *Computer*, 16:11, 2011.

[90] 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.

[91] Muriel Pressigout and Eric Marchand. Real-time Hybrid Tracking using Edge and Texture Information. *International Journal of Robotic Research*, 26:689–713, 2007.

[92] Xiaofeng Ren, Charless C Fowlkes, and Jitendra Malik. Scale-invariant contour completion using conditional random fields. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages 1214–1221. IEEE, 2005.

[93] K. Roman, N. Juan, N. Eduardo, and D. Bertrand. Track-based Self-supervised Classification of Dynamic Obstacles. *Autonomous Robots*, 29(2):219–233, 2010.

[94] 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.

[95] Edward Rosten and Tom Drummond. Fusing points and lines for high performance tracking. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages 1508–1515. IEEE, 2005.

[96] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *Computer Vision–ECCV 2006*, pages 430–443. Springer, 2006.

[97] Edward Rosten, Reid Porter, and Tom Drummond. Faster and better: A machine learning approach to corner detection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 32:105–119, 2010.

[98] J. Salas and C. Tomasi. People detection using color and depth images. *Pattern Recognition*, pages 127–135, 2011.

[99] Benjamin Sapp, Ashutosh Saxena, and Andrew Y. Ng. A fast data collection and augmentation procedure for object recognition. In *AAAI*, pages 1402–1408. AAAI Press, 2008.

[100] 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.

[101] C. Schmid and R.Mohr. Local grayvalue invariants for image retrieval. *Pattern Analysis and Machine Intelligence*, 19(5):530–535, 1997.

[102] S. Se, D. Lowe, and J. Little. Global Localization using Distinctive Visual Features. In *International Conference on Intelligent Robots and Systems*, 2002.

[103] 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.

[104] Steven A Shafer. Using color to separate reflection components. *Color Research & Application*, 10(4):210–218, 1985.

[105] Pramod Sharma, Chang Huang, and Ram Nevatia. Unsupervised incremental learning for improved object detection in a video. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3298–3305. IEEE, 2012.

[106] Daniel Sharvit, Jacky Chan, Huseyin Tek, and Benjamin B Kimia. Symmetry-based indexing of image databases. In *Content-Based Access of Image and Video Libraries, 1998. Proceedings. IEEE Workshop on*, pages 56–62. IEEE, 1998.

[107] 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.

[108] Jamie Shotton, Andrew Blake, and Roberto Cipolla. Contour-Based Learning for Object Detection. In *International Conference on Computer Vision*, volume 1, pages 503–510, 2005.

[109] 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.

[110] Stephen M Smith and J Michael Brady. Susana new approach to low level image processing. *International journal of computer vision*, 23(1):45–78, 1997.

[111] Luciano Spinello, Rudolph Triebel, and Roland Siegwart. Multiclass Multimodal Detection and Tracking in Urban Environments. *International Journal of Robotics Research*, 29:1498–1515, 2010.

[112] M. Sridharan and P. Stone. Global Action Selection for Illumination Invariant Color Modeling. In *International Conference on Intelligent Robots and Systems*, 2007.

[113] Mohan Sridharan and Xiang Li. Autonomous Information Fusion for Robust Obstacle Localization on a Humanoid Robot. In *International Conference on Humanoid Robots*, 2009.

[114] P. Stone, K. Dresner, P. Fidelman, N. K. Jong, N. Kohl, G. Kuhlmann, E. Lin, M. Sridharan, and D. Stronger. UT Austin Villa 2004: Coming of Age, AI TR 04-313. Technical report, Department of Computer Sciences, UT-Austin, October 2004.

[115] 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.

[116] S. Thrun. Stanley: The Robot that Won the DARPA Grand Challenge. *Journal of Field Robotics*, 23(9):661–692, 2006.

[117] A. Torralba and P. Sinha. Statistical Context Priming for Object Detection. In *International Conference on Computer Vision*, pages 763–770, 2001.

[118] Antonio Torralba, Kevin P Murphy, and William T Freeman. Sharing visual features for multiclass and multiview object detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(5):854–869, 2007.

[119] Nhon H. Trinh and Benjamin B. Kimia. Skeleton Search : Category-Specific Object Recognition andSegmentation Using a Skeletal Shape Model. *International Journal of Computer Vision*, 94:215–240, 2011.

[120] T. Tuytelaars and K. Mikolajczyk. Local Invariant Feature Detectors: A Survey. *Foundations and Trends in Computer Graphics and Vision*, 3(3):177–280, 2007.

[121] Tinne Tuytelaars and Luc Van Gool. Matching widely separated views based on affine invariant regions. *International journal of computer vision*, 59(1):61–85, 2004.

[122] Shimon Ullman, Michel Vidal-Naquet, and Erez Sali. Visual Features of Intermediate Complexity and their Use in Classification. *Nature*, 2002.

[123] J. Van De Weijer and C. Schmid. Coloring local feature extraction. *Computer Vision–ECCV 2006*, pages 334–348, 2006.

[124] Andrea Vedaldi, Varun Gulshan, Manik Varma, and Andrew Zisserman. Multiple kernels for object detection. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 606–613. IEEE, 2009.

[125] Javier Velez, Garrett Hemann, Albert S. Huang, Ingmar Posner, and Nicholas Roy. Active exploration for robust object detection. In *International Joint Conference on Artificial Intelligence (IJCAI)*, Barcelona, Spain, July 2011.

[126] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–511. IEEE, 2001.

[127] Markus Weber, Max Welling, and Pietro Perona. Unsupervised Learning of Models for Recognition. In *European Conference on Computer Vision*, pages 18–32, 2000.

[128] D. White and R. C. Wilson. Spectral Generative Models for Graphs. In *International Conference on Image Analysis and Processing*, 2007.

[129] Jonathan S. Yedidia, William T. Freeman, and Yair Weiss. Generalized Belief Propagation. In *Neural Information Processing Systems*, pages 689–695, 2000.

[130] Long Zhu, Yuanhao Chen, Antonio Torralba, W Freeman, and A Yuille. Part and appearance sharing: Recursive compositional models for multi-view. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1919–1926. IEEE, 2010.