Robotics and Autonomous Systems 57 (2009) 629–644

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot

Color learning and illumination invariance on mobile robots: A survey

Mohan Sridharan[a,*], Peter Stone[b]

[a] Texas Tech University, United States

[b] The University of Texas at Austin, United States

ARTICLE INFO

Article history:
Received 8 May 2008
Received in revised form
3 January 2009
Accepted 20 January 2009
Available online 31 January 2009

Keywords:
Real-time robot vision
Autonomous mobile robots
Color learning
Illumination adaptation

ABSTRACT

Recent developments in sensor technology have made it feasible to use mobile robots in several fields, but robots still lack the ability to accurately sense the environment. A major challenge to the widespread deployment of mobile robots is the ability to function autonomously, learning useful models of environmental features, recognizing environmental changes, and adapting the learned models in response to such changes. This article focuses on such learning and adaptation in the context of color segmentation on mobile robots in the presence of illumination changes. The main contribution of this article is a survey of vision algorithms that are potentially applicable to color-based mobile robot vision. We therefore look at algorithms for color segmentation, color learning and illumination invariance on mobile robot platforms, including approaches that tackle just the underlying vision problems. Furthermore, we investigate how the inter-dependencies between these modules and high-level action planning can be exploited to achieve autonomous learning and adaptation. The goal is to determine the suitability of the state-of-the-art vision algorithms for mobile robot domains, and to identify the challenges that still need to be addressed to enable mobile robots to learn and adapt models for color, so as to operate autonomously in natural conditions.

Published by Elsevier B.V.

1. Introduction

Mobile robots are dependent on sensory information for their operation, but they still lack the ability to accurately *sense* the environmental input. Recent developments in sensor technology have provided a range of high-fidelity sensors at moderate costs, such as laser range finders, color cameras, and tactile sensors, thereby making it feasible to use mobile robots in real-world applications [1–3]. However, the sensors typically need extensive manual calibration, which has to be repeated when environmental conditions change. This sensitivity to environmental changes is more pronounced in the case of color cameras — they require extensive manual calibration and are sensitive to illumination changes. Mobile robots however frequently have to operate in environments with non-uniform illuminations. Therefore, even though color is a rich source of information, mobile robot applications have, until recently, relied primarily on non-visual sensors [4]. Even when a robot is equipped with a camera, the focus has largely been on using local edge-based gradients computed from the intensity image [5]. Current sensory processing methods on robots therefore fail to exploit the information encoded in color images and fail to eliminate the sensitivity to environmental changes.

The need for autonomous operation is a major challenge to the widespread use of mobile robots. In order to be suitable for real-world applications, it is not sufficient for the robot to be equipped with good sensors — it is important for the robot to exploit all the available information. The robot should be able to autonomously *learn* appropriate models for environmental features, to *recognize* environmental changes, and *adapt* the learned models in response to such changes. Mobile robots are typically characterized by constrained computational resources, but they need to operate in real-time in order to respond to the dynamic changes in their environment. Autonomous learning and adaptation on mobile robots is hence a challenging problem.

In this article the focus is on learning and adaptation in the context of visual input from a color camera. We specifically focus on: (a) color segmentation i.e. the process of clustering image regions with similar color and/or mapping individual image pixels to color labels such as *red, yellow* and *purple*, and on (b) color constancy i.e. the ability to assign the same color labels to objects under different illuminations. Mobile robot applications that do use color information, typically have some form of color segmentation as the first step, but this segmentation is sensitive to illumination changes. Fig. 1 shows an example flowchart of visual processing using color information.

[*] Corresponding address: Texas Tech University, 302 Pine Street, Abilene, TX 79601, United States.

E-mail addresses: mohan.sridharan@ttu.edu (M. Sridharan), pstone@cs.utexas.edu (P. Stone).

0921-8890/$ – see front matter. Published by Elsevier B.V.
doi:10.1016/j.robot.2009.01.004

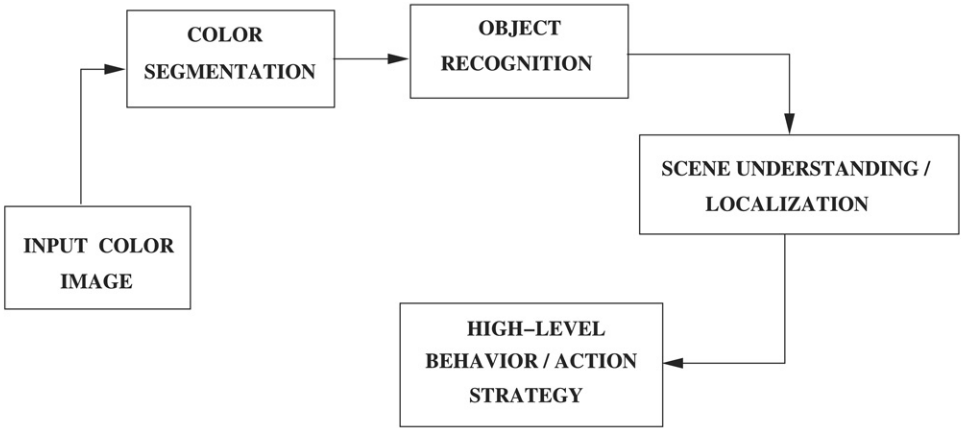
COLOR SEGMENTATION
OBJECT RECOGNITION
SCENE UNDERSTANDING / LOCALIZATION
INPUT COLOR IMAGE
HIGH–LEVEL BEHAVIOR / ACTION STRATEGY

Fig. 1. Typical vision-based operation flowchart.

The first stage of visual processing is *color segmentation*, which takes as input a color image in one of many color spaces (e.g. RGB or YCbCr) depending on the camera being used and the application domain. Color segmentation involves clustering contiguous image regions into meaningful groups based on color, and/or mapping image pixels to one of a predefined set of color labels. The output of the segmentation module is hence either a list of clustered image regions superimposed on the original image, or an image of the same size as the original image with each pixel labeled with the numerical index of the corresponding color label (e.g. “1” for *red*, “2” for *blue*). This clustering and mapping is typically computationally expensive, requires extensive human supervision, and is sensitive to illumination changes.

The color-segmented image regions are the input to the *object recognition* module, where the segmented regions are used to find “objects” and other structures with well-defined properties such as size, shape, and color. These known properties of objects are used to determine heuristics and constraints that help choose the segmented regions that best match the constraints. Object recognition using color or other features is an extensively researched field [6–8] that is not the focus of this article. Object recognition typically provides a list of detected objects and their locations in the image.

The output of the object recognition module serves as the input to the next stage of visual processing, which could involve some form of 3D scene understanding, or could serve as input to a localization module on mobile robots. Objects and their relative arrangements are used to determine additional information about the 3D scene being observed. If a stereo camera is being used, for instance, the depth information can be combined with the information about the objects to reconstruct the 3D scene. On mobile robots the known map of the world (object sizes and locations in global coordinates) can be exploited to use the computed distances and angles to the detected objects to localize the robot i.e. find its pose (position and orientation) in global coordinates. The robot can also perform simultaneous localization and mapping (SLAM) [9] or determine the structure of the world based on the motion (SFM) [10].

The reconstructed 3D scene and estimated robot pose can be used by a high-level module to choose the sequence of actions to perform in order to achieve a specific goal. For instance, if the robot is to navigate to a certain location or retrieve a target object from a specific location, the robot can use the available information to plan its actions so as to maximize the likelihood of success and minimize the use of the available resources.

In this article we are primarily interested in the problem of color-based mobile robot operation, and we are specifically focused on the task of robust color segmentation — the first module in Fig. 1. Color segmentation is typically computationally expensive and sensitive to illumination. In order to make color-based mobile robot operation feasible, we need to enable real-time autonomous segmentation of input images into meaningful color labels, and enable adaptation to illumination changes in a dynamically changing environment. We therefore survey a set of representative algorithms that have been proposed to tackle color segmentation, color learning, illumination invariance, and the associated high-level action planning, on mobile robots. We also look at approaches that just address the underlying vision problems. The goal is to determine the suitability of the state-of-the-art vision algorithms for mobile robot domains, and to identify the challenges that still need to be addressed in order to enable autonomous color-based mobile robot operation. Furthermore, we also identify key results that can be used to achieve autonomous operation with other visual features, thereby moving towards autonomous mobile robot operation under natural conditions.

2. Survey overview

The topics we aim to survey in this article – color segmentation, color learning and illumination invariance – continue to be extensively researched in the computer vision community. However, many state-of-the-art vision algorithms are not applicable to mobile robots for a variety of reasons. Algorithms for mobile robot vision problems need a few essential characteristics:

- Real-time operation, since mobile robots typically need to respond rapidly to changes in the surroundings.
- Deployment without extensive manual calibration (*training*) or extensive prior information, since mobile robots frequently need to be deployed in previously unseen environments.
- Adaptation to previously unknown environmental conditions without human supervision, since the operating conditions in mobile robot domains do not remain constant — for instance the illumination changes.

In this survey we therefore use the above-mentioned criteria to evaluate a set of vision algorithms regarding the extent to which they could be applicable to color-based mobile robot vision. Though only a subset of the algorithms are directly applicable to mobile robot domains, even those that are currently unsuitable for robot applications provide interesting research ideas and a better understanding of the underlying problems. Fig. 2 presents an overview of the survey — it lists the four distinct modules that we discuss. Section 3 discusses some standard approaches for color segmentation of static scenes, followed by methods specific to mobile robots. Section 4 presents a brief description of color spaces, and human perception and naming of colors. It also

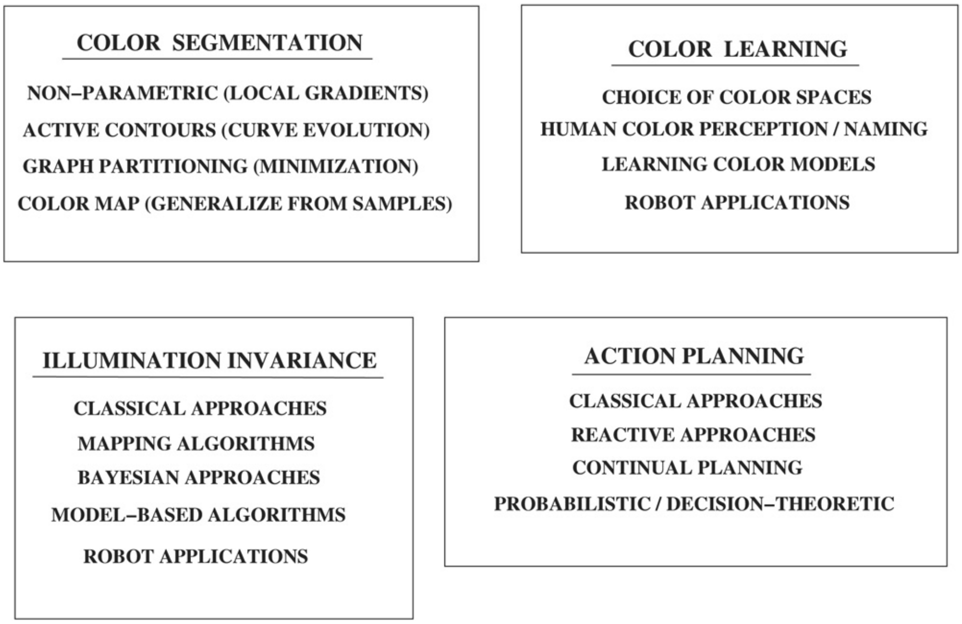

Fig. 2. Overview of the modules surveyed.

discusses algorithms for autonomous segmentation and learning of models of color distributions. Section 5 discusses approaches for illumination invariance, from the early studies to current approaches on mobile robots. Section 6 presents an overview of some planning approaches for autonomous control of sensing and high-level behavior. The "Color Segmentation", "Color Learning" and "Illumination Invariance" modules in Fig. 2 all fall primarily within the "Color Segmentation" box in Fig. 1, while "Action Planning" refers to the box labeled "High-level behavior" — it plans the actions that enable autonomous learning and adaptation. Section 7 briefly discusses how these modules are interlinked, and how the inter-dependencies can be exploited to create a real-time color-based mobile robot vision system that autonomously learns color models and adapts to illumination changes. Finally we summarize and discuss directions of further research in Section 8.

There have been other surveys on the use of vision for tasks such as robot control and navigation [11,12]. This survey aims to provide a different perspective by identifying the vision challenges on the path to autonomous mobile robot operation with color visual input. Though the survey is mainly for vision researchers interested in working on robot domains, it is also intended to be useful for roboticists exploring vision as an information source, while newcomers to both vision and robotics can get a flavor of the existing approaches for vision-based mobile robot operation.

3. Static color segmentation

Color segmentation is typically the first step in a robot vision system that uses a color camera. The goal here is to either cluster image regions into *similar* groups and/or to create a mapping from each possible pixel value to a discrete color label, i.e. to create a *color map*:

$$\Pi_E : \{m_{1,i}, m_{2,j}, m_{3,k}\} \mapsto l\,|_{l\in[0,N-1]}, \quad \forall i, j, k \in [0, 255] \qquad (1)$$

where m_1, m_2, m_3 are the values along the color channels (e.g. R, G, B), E represents the dependence on illumination, and l refers to the numerical indices of the color labels (e.g. *blue* $= 1$, *orange* $= 2$). Color segmentation of individual images is a heavily researched field in computer vision, with several good algorithms [13–15]. We first discuss a few of these approaches.

The mean-shift algorithm [13] is a very general non-parametric technique for the analysis of complex multi-modal feature spaces (e.g. texture) and the detection of arbitrarily shaped clusters in such spaces. The feature space is modeled as an empirical probability density function (pdf). Dense regions in the feature space correspond to local maxima, i.e. the modes of the unknown pdf, which are located at the zeros of the gradient of the density function. Unlike the classical approaches for kernel density estimation [16], the mean-shift procedure locates the local maxima i.e. zeros without estimating the density function. Iterative computation of the mean-shift vector that always points in the direction of maximum increase in density, and translation of the kernel in the direction of the mean-shift vector, is guaranteed to converge to a local mode. Mean-shift vectors therefore define a path to the desired modes. Once the modes are found, the associated clusters can be separated based on the local structure of the feature space. Though the technique performs well on several vision tasks such as segmentation and tracking, its quadratic computational complexity makes it expensive to perform on mobile robots with computational constraints. Fig. 3 shows image results obtained using mean-shift code [13].

An entirely different approach to image segmentation is the use of "active contours" (curve evolution), where initial contours are defined and deformed towards the object boundaries using partial differential equations. The active contour methods typically follow one of two representation schemes: parametric active contours (PAC) and geometric active contours (GAC). PAC use a parametric representation for the curves while GAC use level sets. Level sets and hence the GAC methods readily handle changes in topology as the curve evolves, for example merging and splitting. Active contour methods are typically classified into three groups: edge-based [17], region-based [18,15] and hybrid [19], based on whether they primarily operate on image edges, regions or a combination of both. Sumengen et al. [15] describe a region-based GAC method that segments images into multiple regions. Their approach uses edgeflow image segmentation, where a vector field is defined on pixels in the image grid and the vector's direction at each pixel is towards the closest image discontinuity at a predefined scale. An edge function based on this vector field is used to segment precise region boundaries. Their method allows the user to specify the similarity measure for region merging and splitting

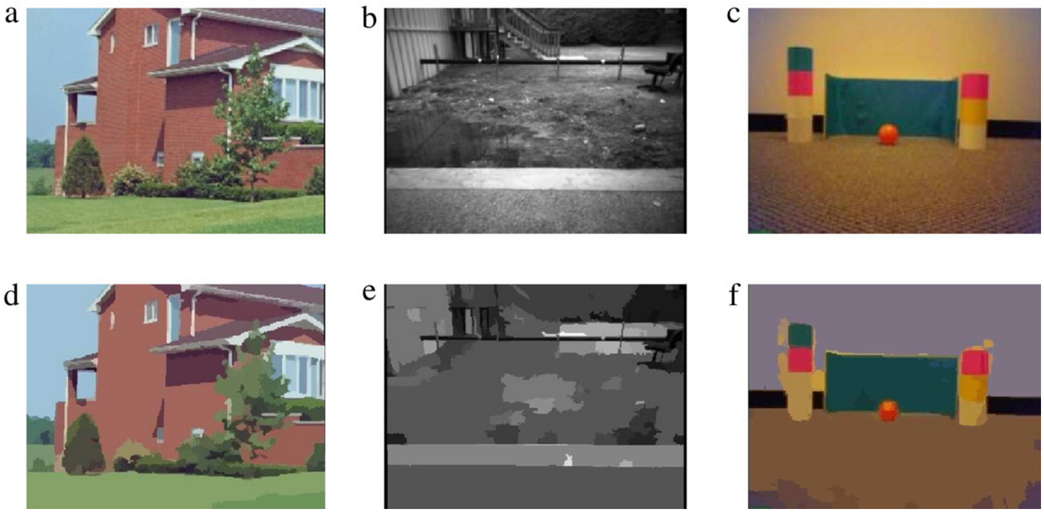

Fig. 3. Mean shift using [13]. (a) House [13], (b) Outdoor, (c) Robot soccer, (d)–(f) Segmented images.

based on different image characteristics such as color or texture. The algorithm is reasonably robust to the initial curve estimates and provides good segmentation results on a variety of images. However, it involves an iterative computation for finding the final contours and takes seconds of off-board processing per image. These algorithms are hence unsuitable for real-time operation on mobile robots with constrained resources.

Image segmentation can also be posed as a graph-partitioning problem, and more specifically as a weighted undirected graph $G = (V, E)$, where each node $v_i \in V$ represents a pixel in the image, and the edges E connect certain pairs of neighboring pixels. Each edge is assigned a weight based on some property of the pixels that it connects. In recent times, graph-based methods [20,14] have proven to be powerful segmentation algorithms. One popular class of graph-based segmentation methods find minimum cuts in the graph, where a *cut* measures the degree of dissimilarity between point sets by computing the weights of the graph edges that have to be removed to separate the two sets (say A, B):

$$cut(A, B) = \sum_{u \in A,\, v \in B} w(u, v) \tag{2}$$

and the goal is to minimize this *cost*. Since the minimum cut favors cutting small sets of isolated nodes in the graph [21], Shi and Malik [14] proposed the idea of considering pixel clusters instead of looking at pixels in isolation. Their *normalized cut* is a robust global criterion that simultaneously maximizes the similarity within a cluster and the dissimilarity between clusters. Since the normalized cut for a graph with regular grids is NP-complete, it is approximated and solved using principles drawn from the generalized eigenvalue problem [14]. In addition to providing good segmentation and object recognition, normalized cuts have also been used for other computer vision tasks such as motion tracking [22] and 3D view reconstruction [23]. However, the graph-optimization process is time-consuming. Even with the current linear approximations to the original quadratic-time implementation it involves an iterative optimization phase that takes seconds of off-board processing per image. Graph-based algorithms are hence computationally expensive to perform in real-time (20–30 Hz) on mobile robot platforms. Other recent graph-based image segmentation techniques (e.g. Felzenswalb and Huttenlocher [20]) provide improved measures for graph partitioning but still have similar computational limitations, even after incorporating simplifying assumptions that increase processing speed. Pantofaru and Hebert [24] compare the mean-shift and graph-based image segmentation algorithms. Fig. 4 shows sample results obtained with the NCuts code [25].

Table 1
Standard segmentation methods — very few are suitable for mobile robot vision.

Algorithms	Computational complexity	Manual training, prior information	Adaptation to new situations
Non-parametric [13]	High	No	No
Active contours [17,19,15]	High	Yes	No
Graph partitioning [20,23,14]	High	No	No
Color maps [29,31,26]	Low	Yes	No

Many mobile robot applications require real-time operation (20–30 Hz) and a robust mapping from pixels to color labels (the color map in Eq. (1)). In addition to being clustered, similar image regions may hence need specific color labels and may need to be tracked across several images even when environmental conditions change. This color map is typically created through an elaborate offline training phase — images are manually labeled over a period of hours to obtain the initial training samples [26]. Since the extensive labeling only labels 5%–10% of the color space, the color map is produced by generalizing from these training samples using machine learning techniques such as neural networks [27], decision trees [28], axis-parallel rectangle creation [29] and nearest neighbors [30]. Fig. 5 provides segmentation results on some sample images captured on-board a legged robot. The main disadvantage of the hand-labeled color map is that this entire training process has to be repeated when the environmental conditions such as the illumination change. Even methods that provide some robustness to illumination changes by using a different color space [31] do not eliminate the need for extensive manual calibration, or the sensitivity to illumination.

Table 1 uses the criteria mentioned in Section 2 to evaluate the segmentation algorithms discussed above. The "adaptation" column represents the ability to detect environmental changes, and to consistently hold the appropriate belief over various colored regions when such changes occur — *red* regions for instance should be clustered appropriately and labeled *red* even when the illumination changes. In summary, classical approaches to color segmentation are either computationally expensive to implement on mobile robots and/or require extensive manual calibration that is sensitive to environmental changes.

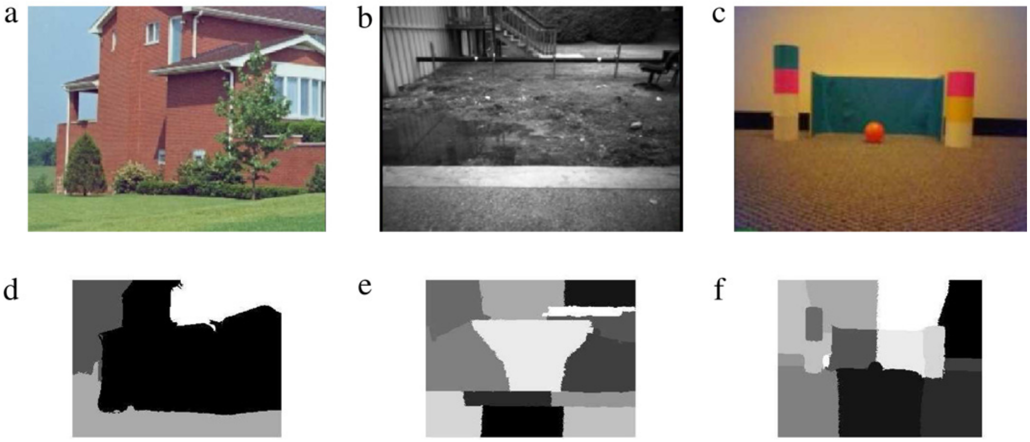

Fig. 4. Normalized cuts using [25]. (a) House [13], (b) Outdoor, (c) Robot soccer, (d)–(f) Segmented images.

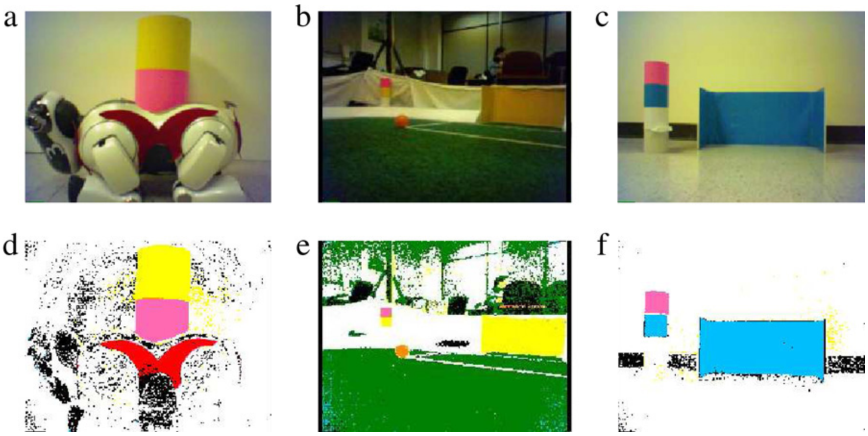

Fig. 5. Hand-labeled color map on legged robots [31]. (a)–(c) Original images, (d)–(f) Segmented output.

4. Color learning

An alternative to extensive manual color calibration is to enable the robot to autonomously segment images using learned models of color distributions. Attempts to learn color models or make them independent to illumination changes have produced reasonable success [32–34] but the methods either require the spectral reflectances of valid objects to be pre-computed, or require additional transformations. A mobile robot deployed in real-world domains is frequently required to operate in new environments where the surroundings, including objects, may change dynamically. It is therefore not feasible to measure the properties of the environment in advance. The additional transformations between color spaces or for image transformation compensation, operate at the level of pixels or small image regions. Such transformations are expensive to perform on mobile robots that require real-time operation.

One question that arises in color learning is the choice of color spaces. Gevers and Smeulders [32] evaluate several color spaces to determine their suitability for recognizing multicolored objects invariant to significant changes in viewpoint, object geometry and illumination. They present a theoretical and experimental analysis of several color models. In addition to RGB, Intensity I, normalized RGB (*rgb*), saturation S, and Hue H, they propose three models:

$$c_1 = \tan^{-1}\frac{R}{\max\{G,B\}}, \qquad c_2 = \tan^{-1}\frac{G}{\max\{R,B\}},$$
$$c_3 = \tan^{-1}\frac{B}{\max\{R,G\}} \tag{3}$$

$$l_1 = \frac{(R-G)^2}{(R-G)^2+(R-B)^2+(G-B)^2},$$
$$l_2 = \frac{(R-B)^2}{(R-G)^2+(R-B)^2+(G-B)^2}$$
$$l_3 = \frac{(G-B)^2}{(R-G)^2+(R-B)^2+(G-B)^2}$$
$$m_1 = \frac{R^{\mathbf{x}_1}G^{\mathbf{x}_2}}{R^{\mathbf{x}_2}G^{\mathbf{x}_1}}, \qquad m_2 = \frac{R^{\mathbf{x}_1}B^{\mathbf{x}_2}}{R^{\mathbf{x}_2}B^{\mathbf{x}_1}}, \qquad m_3 = \frac{G^{\mathbf{x}_1}B^{\mathbf{x}_2}}{G^{\mathbf{x}_2}B^{\mathbf{x}_1}}.$$

Each of the three channels of $c_1c_2c_3$ represents a ratio of color channel values, while each component of $l_1l_2l_3$ represents normalized square color differences. The model $m_1m_2m_3$ has each channel representing a ratio of color values of neighboring pixels $(\mathbf{x}_1, \mathbf{x}_2)$ — see [32] for details. Assuming dichromatic reflection and white illumination, the authors show that *rgb*, saturation S, Hue H, and the newly proposed models $c_1c_2c_3$, $l_1l_2l_3$ and $m_1m_2m_3$ are all invariant to the viewing direction, object geometry and illumination. Hue H and $l_1l_2l_3$ are also invariant to highlights, while $m_1m_2m_3$ is independent of the illumination color and inter-reflections under the narrow-band filter assumption. This analysis serves as a reference on the choice of color space, based on the requirements of the specific application.

In parallel to the computer and robot vision research of color, human perception and representation of color have been extensively studied over the last several decades. In his thesis, Lammens [35] presents a nice analysis of the physiology of human color perception and its representation in language. He analyzes how humans perceive colors and represent them in language, and he proposes appropriate color models based on neurophysiological data and psychophysical findings in color perception. More recent work by Mojsilovic [36] provides a computational model for color naming based on experiments where human subjects are asked to name and distinguish between colored patches. She proposes a color naming metric that measures distances between colors analogous to the human perception of color. The metric, based on distances and angular measurements in the LAB and HSL color spaces [37], is used to determine the color decomposition of images. Such results from the analysis of the human visual system can be exploited by the robot vision researchers. For instance, in their work on mobile robots, Sridharan and Stone [31] use the LAB color space [38], that organizes colors similar to the human visual system, in order to provide robustness to minor illumination changes.

In an approach representative of computer vision methods for robust color learning, Lauziere et al. [34] use the prior knowledge of the spectral reflectances of the objects under consideration for learning color models and recognizing objects under varying illumination. The color camera sensitivity curves are measured offline and used to recognize objects better under daylight illumination conditions. However, in addition to being computationally expensive, their method requires prior measurement of the camera characteristics and the spectral properties of the environment. Mobile robots are however frequently required to operate in new environments.

On mobile robots, attempts to autonomously learn the color map have rarely been successful. Cameron and Barnes [39] present a technique based on scene geometry, where edges are detected in the image and closed figures are constructed to find image regions that correspond to certain known environmental features. The color information extracted from these regions is used to build the color classifiers, using the Earth Mover's distance (EMD) [40] as the cluster similarity metric. Illumination changes are tracked by associating the current classifiers with the previous ones based on the symbolic color labels. The edge detection, formation of closed figures, and clustering are computationally expensive (order of a few seconds per image) even with off-board processing. However, the technique introduces the promising idea of exploiting the structure inherent in robot application domains to learn color models.

Chernova et al. [41] propose another approach that exploits the known structure of the application environment to learn color models. The robot moves along a pre-specified sequence of poses. At each pose, edges are detected and objects are found using domain knowledge, and image regions within the detected object were used as training samples to generate the color map. The technique works within a controlled lab setting and the appropriate motion sequence is manually specified. For true autonomous behavior the robot should be able to autonomously plan action sequences suitable for color learning and adaptation to environmental changes.

Jungel [42] modeled colors using three layers of color maps — the lowest layer represents a small subset of easily distinguishable colors, while the higher layers represent a greater number of colors with increasing precision levels. Colors in each level are represented as cuboids relative to a reference color. In order to increase the processing speed, very little computation is performed to distinguish between colors whose distributions are similar or overlap in the color space. Instead, domain-specific constraints based on the known object properties such as colors and sizes are used during object recognition to disambiguate between object colors. The reference color is tracked with minor illumination changes, and the distributions of other colors in the color space (the cuboids) are displaced by the same amount as the reference color. Though it is an innovative solution for real-time robot applications, cuboids are not a good model for color distributions and different color distributions do not change by the same amount with illumination changes. As a result, the generated map is reported to be not as accurate as the hand-labeled one.

The recent DARPA challenges (Grand Challenges 2005, 2006; Learning Applied to Ground Robots 2004–2008; Urban Challenge 2007) have emphasized the need for using visual information in real-world applications [43,44,3,45,46]. In order to navigate autonomously outdoors, issues such as color segmentation and illumination invariance had to be addressed. Mathematical models such as Mixture of Gaussians (MoG) and histograms were used to model color distributions. For instance, the MoG model for the a priori probability density of pixel values ($\mathbf{m}$) for a particular color label (l) is given by:

$$p(\mathbf{m}|l) = \sum_{i=1}^{k} p(i) \cdot f_i(\mathbf{m}), \qquad f_i(\mathbf{m}) \sim \mathcal{N}(\mu_i, \Sigma_i) \tag{4}$$

where there are k weighted components in the model (weight $= p(i)$), each being a 3D Gaussian with a mean (μ) and a covariance matrix (Σ). However, none of these techniques exploit the domain knowledge significantly, and the modeling of colors is limited to distinguishing between "safe" and "unsafe" road regions. The visual input is used only to find obstacles or image edges, while high-level decisions are predominantly based on non-visual sensors such as laser and GPS. However, the colored objects in mobile robot domains encode a lot of useful information. The challenge is to find an appropriate representation for color distributions that may overlap in the color space.

Sridharan and Stone [47,48] address the challenge of autonomous real-time modeling of overlapping color distributions. The known structure of the robot's environment (positions, sizes and color labels of objects) is exploited to plan an action sequence suitable for learning the desired colors. Color distributions are modeled using a hybrid model of 3D Gaussians and 3D histograms:

$$\text{if Gaussian } (l) \text{ then } p(\mathbf{m}|l) \sim N(\boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l) \tag{5}$$
$$\text{else } p(\mathbf{m}|l) \equiv \frac{\mathrm{Hist}_l(b_1, b_2, b_3)}{\sum \mathrm{Hist}_l}$$

where $N(\boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l)$ is the 3D Gaussian for color l, while (b_1, b_2, b_3) are the histogram bin indices corresponding to the pixel values (m_1, m_2, m_3). The robot chooses the best representation for each color distribution based on autonomously collected image statistics. The learned color map provides segmentation and localization accuracies comparable to that obtained from a manually-labeled map. The learning scheme however takes ≤ 6 min of robot time instead of hours of human effort. Overlapping colors (e.g. *red, pink, orange*) are modeled efficiently but the sensitivity to illumination still remains. Segmentation performance with learned color maps is shown in Fig. 6. Table 2 evaluates the color learning algorithms based on the criteria essential for mobile robot applications. Algorithms developed for color learning on mobile robots eliminate the need for manual training and do not need prior knowledge of color distributions, irrespective of whether color is used as a primary or secondary input source. However, when color is the main sensory input some robot applications exploit domain knowledge to model colors whose distributions may overlap in the color space (the "*(Yes)*" in the third row). Such domain knowledge is much easier to provide than hand labeling images or measuring properties of each new scene and the objects in them. Furthermore, existing algorithms can be used to learn most of this domain knowledge (e.g. SLAM algorithms [49] for learning the objects' locations). To summarize, standard color modeling algorithms are computationally expensive, and even when implemented successfully on mobile robots, they do not address the problem of detecting and adapting to environmental (illumination) changes.

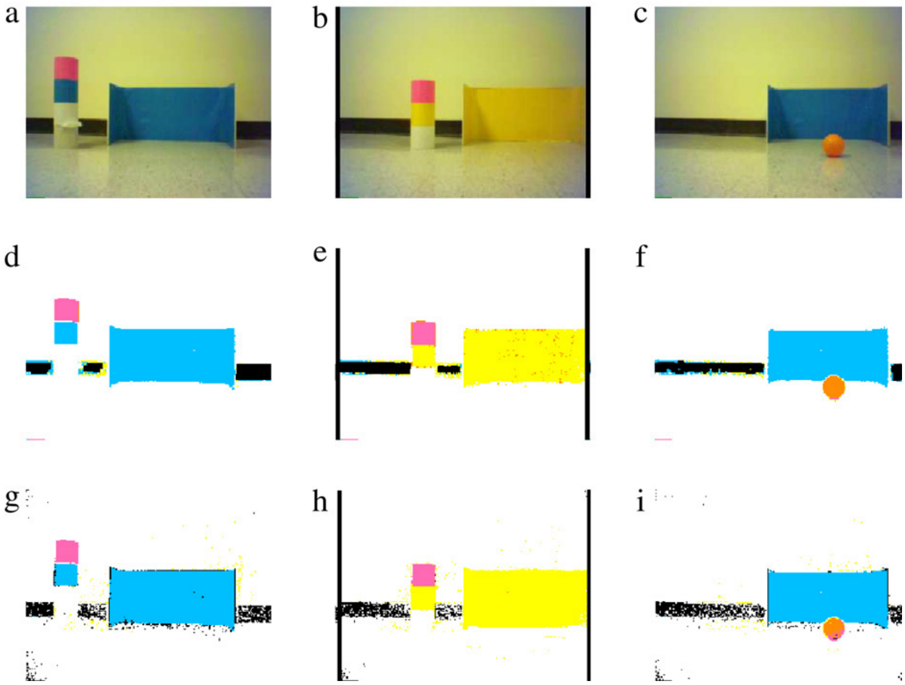

Fig. 6. Segmentation on mobile robots with learned color maps. (a)–(c) Original, (d)–(f) Segmented (Mixture of Gaussians), (g)–(i) Segmented (Hybrid model) [47]. Hybrid model works in real-time and with overlapping colors.

Table 2
Color modeling methods — eliminate manual calibration but cannot adapt.

Algorithms	Computational complexity	Manual training, prior information	Adaptation to new situations
Color modeling [32,34,36]	High	Yes	No
Robot apps: color secondary [44,3]	Low	No	No
Robot apps: color primary [39,41,42,48]	Low	No (Yes)	No

5. Illumination invariance

Irrespective of the method used for color segmentation or color learning, one common problem is the sensitivity to illumination. Mobile robot application environments are characterized by illumination changes and it is essential for the robot to be able to map the image pixels to the correct color labels, i.e. we need color constancy. This section surveys some popular color constancy techniques proposed over the last few decades. The following equation illustrates the factors that influence color perception [50,51]:

$$m_j^p = \int E(\lambda) S^{\mathbf{x}}(\lambda) R_j(\lambda)\, \mathrm{d}\lambda \qquad (6)$$

where $E(\lambda)$ is the illumination, $S^{\mathbf{x}}(\lambda)$ is the surface reflectance at a 3D scene point $\mathbf{x}$ that projects to the image pixel p, while $R_j(\lambda)$ is the channel sensitivity of the imaging device's jth sensor. The sensor value of the jth sensor of the imaging device at pixel p, m_j^p, is the integral of the product of these three terms over the range of wavelengths of light. The robot needs to perceive the same color (l) of the object under different illuminations. Changing either the surface reflectance or the illumination can modify the sensor values, assuming that the imaging device and hence the channel sensitivities remain unchanged. Fig. 7 presents an example of the effect of illumination change on a trained color map. Most color constancy methods focus on static images, are computationally expensive, and predominantly estimate the current illumination $E(\lambda)$ and surface reflectance $S(\lambda)$ as a function of a known illumination.

The Retinex theory [52] was one of the first attempts to explain human color constancy. Based on the assumption that white reflection induces maximal *rgb* camera responses (since light incident on a white patch is spectrally unchanged after reflection), Land suggested that measuring the maximum r, g, and b responses can serve as an estimate of the scene illuminant color. When it was determined that the maximum *rgb* in an image is not the correct estimate for white, the technique was modified to be based on global or local image color averages. The "Gray World" algorithm by Buchsbaum [53] is also based on the same principle. However the image average, either local or global, has been shown to correlate poorly with the actual scene illuminant [54]. In addition, this method excludes the possibility of distinguishing between the actual changes in illumination and those as a result of a change in the collection of surfaces in the environment. However, the environmental objects change dynamically in a mobile robot application domain.

Another classic approach to color constancy was proposed by Maloney and Wandell [51]. They try to recover the surface reflectance functions of objects in the scene with incomplete knowledge of the spectral distribution of the illumination. They provide a mathematical analysis of color constancy and decompose both the illumination and the surface reflectance functions as a weighted linear combination of a small set of basis functions:

$$S^{\mathbf{x}}(\lambda) = \sum_{j=1}^{n} \sigma_j^{\mathbf{x}} S_j(\lambda), \qquad E(\lambda) = \sum_{i=1}^{m} \epsilon_i E_i(\lambda) \qquad (7)$$

leading to a relationship between sensor responses and surface reflectances:

$$m^p = \Lambda_\epsilon \sigma^{\mathbf{x}} \qquad (8)$$

where Λ_ϵ is a p (dimensions of m^p, the sensor vector at a pixel) by n matrix whose (k, j)th entry corresponds to $E(\lambda)S_j(\lambda)R_k(\lambda)$. They show that a unique solution is possible *iff* the number of sensory channels, three in a color camera, is at least one more than the

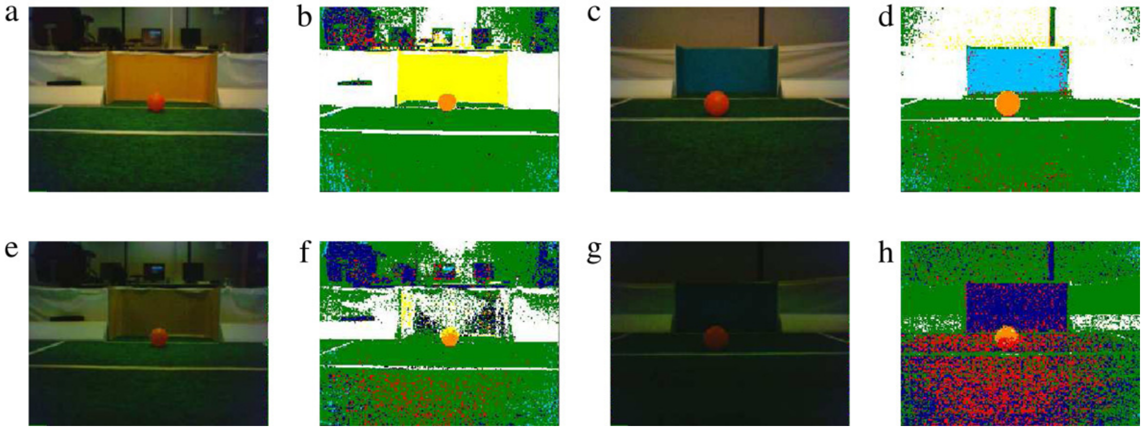

Fig. 7. Illumination sensitivity. (a)–(d) Color map trained under an illumination, (e)–(h) Ceases to work when illumination changes significantly.

number of basis functions that determine the surface reflectances. Assuming that the illumination is constant over an image region, they use the pixel values to estimate the illumination and the surface reflectance. The paper also presents some interesting conclusions about human color perception. However, neither surface reflectances nor illuminations can be modeled well with just two basis functions. In addition, the illumination does not have to remain constant over large image regions.

Forsyth proposed the *gamut mapping* algorithm for color constancy [55], which estimates the current illumination as a function of a known illumination. Since surfaces can reflect no more light than is cast on them, he concluded that the illuminant color is constrained by the colors observed in the image and can hence be estimated using image measurements alone. The algorithm generates a set of mappings that transform image colors (sensor values) under an unknown illuminant to the gamut of colors observed under a standard (canonical) illuminant, using 3D diagonal matrices. Then a single mapping is chosen from the feasible set of mappings in order to estimate the actual sensor values under the canonical illuminant. Though the method requires advance knowledge of the possible illuminants to generate the mappings, it emphasizes the point that image statistics can provide a good estimate of the illumination.

Realizing that the scene illuminant intensity cannot be recovered in Forsyth's approach, Finlayson modified it to work in the 2D chromaticity space [56]. He proved that the feasible set calculated by his algorithm was the same as that calculated by Forsyth's original algorithm when projected into 2D [57]. He then proposed the *median selection* method, which includes a constraint on the possible color of the illuminant in the gamut mapping algorithm. More recently he proposed a correlation framework [50], where the range of colors that can occur under each of a possible set of illuminants is computed in advance. Correlating this range with the colors in a test image estimates the likelihood that each of a possible set of illuminants is the scene illuminant.

Instead of estimating the illumination intensity, Klinker et al. [33] focus on the surface reflectance properties of objects in the scene. They propose an approach for color image understanding based on the Dichromatic Reflection Model (DRM). The color of reflected light is described as a mixture of the light from the surface reflection (highlights) and body reflection (actual object color). DRM can be used to separate a color image into an image based on just the highlights, and the original image with the highlights removed. They present a color image understanding system that can describe the reflection processes occurring in the scene. Their work requires prior knowledge of the possible illuminations and properties of objects in the scene, while robots are typically deployed in environments where it is not feasible to measure these properties in advance. However, their technique provides a better understanding of how the illumination and surface reflectances interact to produce the sensor responses i.e. the camera image.

In contrast to the classical approaches to color constancy, several Bayesian approaches have also been proposed. Brainard and Freeman [58] present a Bayesian decision-theoretic framework which combines all available statistics such as gray world, subspace and physical realizability constraints. They model the relation among illuminations, surfaces and photosensor responses and generate a priori distributions to describe the probability of existence of certain illuminations and surfaces. They present a local mass loss function:

$$L(\tilde{\mathbf{x}}, \mathbf{x}) = -\exp\{-|\mathbf{K}_L^{-1/2}(\tilde{\mathbf{x}} - \mathbf{x})|^2\} \tag{9}$$

where $\tilde{\mathbf{x}}$ is the estimate of the parameter vector $\mathbf{x}$, and $|\mathbf{K}_L^{-1/2}\mathbf{x}|^2 = \mathbf{x}^{\mathrm{T}}\mathbf{K}^{-1}\mathbf{x}$. The matrix $\mathbf{K}_L$ is chosen with sufficiently small eigenvalues so that the local mass loss function rewards approximately correct estimates and assigns an almost equal penalty for grossly incorrect estimates. The maximum local mass (MLM) estimator is obtained by minimizing the expected loss for the local mass loss function. The MLM estimator integrates local probabilities and uses Bayes' rule to compute the posterior distributions for surfaces and illuminants for a given set of photosensor responses. However, their approach assumes a priori knowledge of the various illuminations and surfaces, which is not feasible to obtain in many robot application domains.

Along similar lines, Tsin et al. [59] present a Bayesian MAP (*maximum a posteriori*) approach to achieve color constancy for outdoor object recognition with a static surveillance camera. Static overhead high-definition color images are collected over several days, and are used to learn statistical distributions for surface reflectance and the illumination. Then a linear iterative updating scheme is used to converge to the classification of the illumination and hence the objects in the test images. Most mobile robots operate in a dynamically changing world with motion induced image distortion and a limited view of its surroundings. Though collecting statistics as proposed by their approach may not be feasible in most robot applications, defining the space of operation based on the task is applicable to robot domains as well.

In contrast to the Bayesian methods, Rosenberg et al. [60] develop models for sensor noise, canonical (i.e. "true") color and illumination. The pixel values are considered to be proportional to the sensor values under the canonical illuminant, and are transformed to a log-space, normalized and discretized to obtain a 2D color space that factors out the pixel intensities. Based on the assumption that the global scene illuminant parameters are independent of the canonical color of an object, a single distribution is maintained over the canonical colors and used to compute the a priori likelihood of observing a particular color measurement given a set of illumination parameters. Given an input image under an unknown illumination within the set of illuminations the system is trained for, the *a posteriori* global scene illumination parameters are determined by an exhaustive search over the space defined by the learned models, using the KL-divergence measure. Experimental results show that the performance is better than that obtained by using a Maximum Likelihood Estimate approach. This approach assumes a priori knowledge of the illuminations and sensor noise, and involves a computationally expensive search process. Mobile robots however frequently move into regions with previously unknown illuminations and sensor noise.

In the domain of mobile robots, the problem of color constancy has often been avoided by using non-visual sensors such as laser range finders and sonar [4]. Even when visual input is considered, the focus has been on recognizing just a few well-separated colors [38,1] or to use intensity-based features [5]. Until recently, there has been relatively little work on illumination invariance with a moving camera in the presence of shadows and artifacts caused by rapid motion in real-world domains. Furthermore, even the few approaches that do function within the computational constraints of mobile robots (e.g. [61,62,3]) do not autonomously model illuminations and overlapping colors to smoothly detect and adapt to a range of illuminations.

Lenser and Veloso [62,63] present a tree-based state description and identification technique which they use for detecting illumination changes on Aibos (four-legged robots). They incorporate a time series of average screen illuminance to distinguish between illuminations, using the absolute value distance metric to determine the similarity between distributions. Prior work has shown that average scene illuminant is not a good representation for illumination [54]. In addition, their method does not run in real-time on-board a robot. However, their work provides additional support to the hypothesis that temporal image statistics can be used to model illuminations — the goal is to enable mobile robots to collect these statistics autonomously.

Since both sensing and actuation on mobile robots involve uncertainty, probabilistic (i.e. Bayesian) methods are becoming popular for mobile robot tasks. For instance, Schulz and Fox [64] estimate colors using a hierarchical Bayesian model with Gaussian priors and a joint posterior on position and environmental illumination. In their approach, the mean color vector obtained from an image captured at a known location is treated as a direct observation of the current illumination. The posterior over illuminations at time k, denoted by θ_k is modeled as a Gaussian: $p(\theta_k|\bar{i}_{1:k}) = \mathcal{N}(\theta_k; \mu_k, \Sigma_k)$, where $\bar{i}_k$ is the mean color vector and the Gaussian parameters (mean, covariance) are estimated using a standard Kalman filter update [65]. The joint posterior over robot poses and illuminations is factorized:

$$\begin{aligned} p(\theta_k, x_{1:k}|z_{1:k}, u_{1:k-1}) &= p(\theta_k|x_{1:k}, z_{1:k}, u_{1:k-1})\ p(x_{1:k}|z_{1:k}, u_{1:k-1}) \\ &= p(\theta_k|\bar{i}_{1:k})p(x_{1:k}|z_{1:k}, u_{1:k-1}) \qquad (10) \end{aligned}$$

where $z_{1:k}$ are input images up to time k and $u_{1:k-1}$ are the robot motion commands. This decomposition allows the use of Rao-Blackwellised Particle Filters [65] such that illumination (θ_k) can be determined based on the estimated robot pose. The method however requires significant prior information to construct the a priori Gaussian density functions for color distributions over known illuminations — several images are hand labeled. The results reported in the paper indicate that the algorithm requires significant human supervision for experiments with just two distinct illuminations and a small set of colors. The Bayesian framework can however be exploited if the models for color distributions and illuminations are learned autonomously.

Anzani et al. [61] describe another Bayesian approach for illumination invariance on wheeled robots. They use a *Mixture of Gaussians* (MoG) to model the distribution of each color Eq. (4). The labeling of color classes and association with mixture components is done by human supervision, and the Bayesian decision rule is used during classification to determine the color label. In order to adapt the model parameters and number of mixture components to changing illuminations, the EM algorithm [16] is used. Their algorithm has been tested only over a few intensity variations in the lab and involves iterative computation for parameter estimation. In addition to the computationally expensive iterative process, color distributions are not always well-modeled with MoG and it may be infeasible to obtain a large set of training samples in robot domains — incremental learning may be necessary.

The recent DARPA challenges have resulted in several approaches to illumination invariance on real-world mobile robot domains [43,44,3,45,46]. Though the visual information was only used sparingly in these challenges, Thrun et al. [3] model colors as MoG (Eq. (4) in Section 4) and add additional Gaussians or modify the parameters of the existing Gaussians in response to illumination changes. As mentioned above, not all colors are modeled well using MoG given limited training data. Furthermore, they were interested only in distinguishing safe regions on the ground from the unsafe regions and did not have to model colors whose distributions overlap in the color space.

In their work on color learning on legged robots, Sridharan and Stone [47,66] enable the robot to model illuminations using autonomously collected image statistics. While the robot is building models for the color distributions (Eq. (5) in Section 4), it also collects sample environmental images to create histograms in the color space (normalized RGB). In addition, the robot computes the distance between every pair of these histograms to arrive at a distribution of distances between color histograms under a given illumination. The histograms and the distribution of distances characterize a particular illumination. Periodically, the robot captures images and computes the corresponding color space histogram. If the average distance between this histogram and the stored histograms corresponding to the current illumination is significantly different from the expected range of distances under the current illumination, a significant change in illumination is detected. The robot adapts to large illumination changes by re-learning the appropriate portions of the color map. Furthermore, the robot detects minor illumination changes by monitoring the color segmentation performance over known scene objects, and adapts by updating the color distributions in real-time using Gaussian and histogram-merging techniques. The combined strategy enables the robot to operate smoothly in a range of illuminations [67]. Fig. 8 shows sample images as the robot tracks illumination changes.

Table 3 evaluates the illumination invariance algorithms based on the criteria in Section 2. Computer vision approaches are computationally expensive and/or require extensive prior knowledge of the possible illuminations and the scene objects. Algorithms implemented for mobile robots only tackle a small subset of the problem. Methods that can exploit domain knowledge to autonomously detect and adapt color models to new illuminations are very rare [67], and even these schemes require some domain knowledge to be specified in advance (the "*(Yes)*" in the final row). As in the case of color modeling, providing minimal domain knowledge to enable adaptation to illumination changes is significantly easier than measuring all the properties of the environment. Illumination invariance is hence still an open problem, both on mobile robots and in computer vision in general.

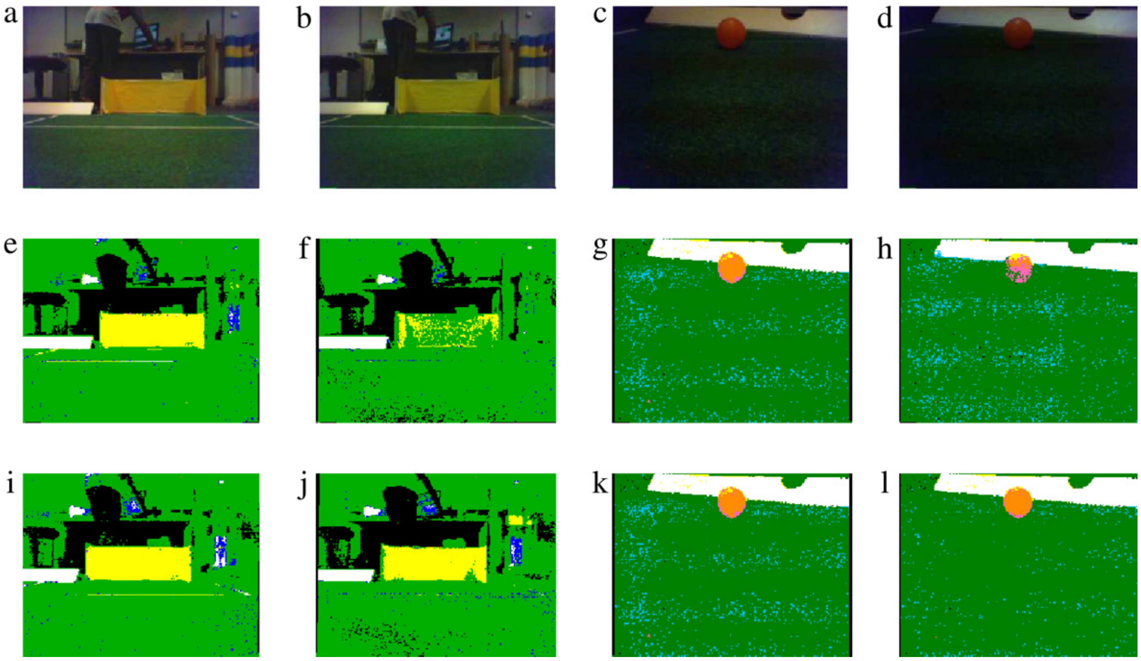

Fig. 8. Tracking illumination changes on mobile robots. (a)–(b) Before and after illumination change (Object1), (c)–(d) Before and after illumination change (Object2), (e)–(h) Without illumination adaptation, (i)–(l) with real-time adaptation.

Table 3
Illumination invariance methods — adaptation to new conditions is very rare.

Algorithms	Computational complexity	Manual training, prior information	Adapt to new situations
Classical methods [55,52,51]	High	Yes	No
Bayesian/model-based [58,60,59]	High	Yes	No
Robot apps: domain knowledge not used [61,62,64]	Low	Yes	No
Robot apps: domain knowledge exploited [67]	Low	No (Yes)	Yes

6. Action planning

Color segmentation is rarely, if ever, the end purpose of a vision algorithm, especially when used on a mobile robot. Many robot vision algorithms actively consider the purpose of object recognition when considering how to spend computational or action resources — sometimes it may be necessary to perform a sequence of actions before a particular object comes within the robot's view and can be used to model the colors. In such situations, it can be useful for a robot to *plan* its motion to facilitate color learning.

We are interested in visual processing management: given a set of information-processing and sensing actions, choose an action sequence that achieves the desired goal reliably, while using the available resources optimally. For instance, if the goal is to learn the color *blue* from a *blue box*, only the image regions known to contain a *box* need to be examined. This problem can be addressed by two different approaches: learning and planning.

In the learning approach the robot could use different action sequences to perform each task repeatedly, using techniques such as Reinforcement Learning [68] to score the action sequences and choose the best sequence for each task. Learning through repeated trials on a robot, though interesting, would however involve extensive manual effort that might be infeasible in many applications. The alternative is to jointly *plan* the sensing (*where to look*) and information-processing (*what to look for*). Challenges to be addressed include: minimizing the amount of domain knowledge to be manually encoded, conflict resolution when multiple feasible actions exist, and planning under uncertainty when the action outcomes are non-deterministic.

There exists a significant body of work in the computer vision community on using a user-specified goal to plan a pipeline of visual operators, including classical approaches for active vision [69]. However, many of these algorithms use deterministic models of the action outcomes: the pre-conditions and the effects of the operators are propositions that are required to be true a priori, or are made true by the application of the operator. Unsatisfactory results are detected by execution monitoring using hand-crafted rules, followed by re-planning the operator sequence or modifying the operator parameters [70–72]. There has also been some work on autonomous object detection and avoidance in vehicles [73], and on interpretation of 3D objects' structure [74] but extensions to more general vision tasks have proven to be difficult.

Probabilistic methods have also been used for image interpretation. Darrell [75] used memory-based reinforcement learning and Partially Observable Markov Decision Processes (POMDPs) [76] to learn what foveation actions to execute, and when to execute the terminal recognition action, in order to foveate salient body parts in an active gesture recognition system. In the learning phase, manual feedback is provided after the execution of the recognition action in order to assign a reward to the action sequence. The learned recognition policies are transformed into a compact augmented Finite State Machine for online recognition. More recently, Li et al. [77] posed image interpretation as a MDP. The reward structure is determined offline by applying all possible sequences of operators to human-labeled images. States are abstracted into a set of image features, and dynamic programming is used to compute value functions for regions in this abstracted feature space. Online image interpretation involves feature extraction and the choice of action that maximizes the value of the learned functions. In robot domains the true state of the world is not known, action outcomes are noisy, and action execution may change the state. Furthermore, computing the utilities of actions by repeated trials is often infeasible in robot domains.

AI planning, in general, is a very well researched field [78]. Several modern AI planning methods exploit the classical planning machinery by relaxing the limiting constraints on prior knowledge of the states and action outcomes [79,80]. Draper et al. [79] proposed C-BURIDAN, a planning scheme that is similar to a POMDP formulation since it models actions as probabilistic state transitions between state beliefs to choose the best action to perform. A symbolic STRIPS-like representation [81] is used for reasoning about the action effects and symbolic sub-goals of the desired goal. Action pre-conditions and effects are manually specified, and a notion of action costs does not exist. The technique cannot accumulate belief by repeated application of the same action without a manual ordering of actions.

On the other hand, Petrick and Bacchus's PKS planner [80] describes actions in terms of their effect on the agent's knowledge rather than their effect on the world, using a first order language. The model is hence non-deterministic in the sense that the true state of the world may be determined uniquely by the actions performed, but the agent's knowledge of that state is not. For example, dropping a fragile item will break it, but if the agent does not know that the item is fragile, it will not know if it is broken, and must use an observational action to determine its status. PKS captures the initial state uncertainty and constructs conditional plans based on the agent's knowledge. In the context of visual operators, PKS would plan to use the operators to examine each salient image region, branching based on what is discovered.

More recently, Brenner and Nebel [82] proposed the Continual Planning (CP) approach, which interleaves planning, plan execution and plan monitoring. It uses the PDDL syntax [83] and is based on the FF planner [84]. Unlike classical planning approaches, an agent in CP postpones reasoning about unknowable or uncertain states until more information is available. The agent allows actions to assert that the action pre-conditions will be met when the agent reaches that point in the execution of the plan, and re-planning is triggered if those pre-conditions are not met during execution or are met earlier. CP is therefore quite similar to PKS in representation, but works by re-planning rather than constructing conditional plans. However, there is *no* representation of the uncertainty in the observations and actions. In vision applications with noisy observations, the optimal behavior may be to take several images of a scene and accumulate the evidence obtained by running the visual operators more than once. This cannot be readily represented in CP, C-BURIDAN or PKS,

A classical approach for planning high-level behaviors on robots is to use a behavior-based architecture, for instance algorithms that build on the *subsumption architecture* [85]. The underlying principle is that the representation for information depends on the intended use of the information. Complicated behaviors are decomposed into layers of simple behaviors. A layer implements a particular behavior goal and higher-level goals *subsume* those of the lower layers. For instance the lowest layer could encode "avoid obstacles" or "follow the wall", followed by a layer that encodes "explore the environment", which in turn is subsumed by the overall goal: "create a map of the environment". The distributed behavior promotes modularity, but there is very little cognitive reasoning in the decision making, and there is *no* explicit state or action representation — *effects of actions are known only by executing the actions*. Such a system would be infeasible in complex robot systems. Hence, most planning systems on robots use some form of knowledge representation, and use reactive planning for simple control tasks within a larger system.

Instead of manually encoding the action outcomes, Sridharan and Stone [67,86] exploit the structure of the environment (known positions, sizes and color labels of objects) to enable a legged robot to autonomously learn probabilistic models for the motion errors and the feasibility of learning color models at different locations. The robot is then able to robustly plan action sequences that simultaneously maximize color learning opportunities while minimizing the possible localization errors [86]. The robot is able to respond to unexpected changes by *re-planning* from its current state, thereby providing support to the hypothesis that a combination of learning and probabilistic planning is currently the most appealing option for autonomous mobile robot operation. Fig. 9 shows examples of planning a sequence of actions to learn colors.

The probabilistic (i.e. decision-theoretic) scheme is currently a popular alternative for planning a sequence of actions for practical domains [87,2]. Here, the planning task is posed as an instance of probabilistic sequential decision making, or more specifically as a POMDP. POMDPs elegantly capture the incomplete knowledge of the state by maintaining a probability distribution over the true state (i.e. *belief* state). The non-deterministic action outcomes are also modeled probabilistically. The POMDP model is solved using techniques such as value-iteration [68], and the output *policy* provides an action sequence that simultaneously maximizes the reliability and minimizes the use of resources. However, practical problems formulated as POMDPs soon become intractable because the state space increases exponentially and the solutions are exponential in the state-space dimensions [88,68]. In recent times, the focus has therefore shifted to finding approximate solutions to POMDPs [89,90] (see [91,92] for surveys), and imposing a structure or hierarchy on the problem to make it more tractable [93,2, 94]. Pineau et al. [2] use a hierarchical POMDP for high-level behavior control on a nursing assistant robot. Similar to the MAXQ decomposition for MDPs [95] they impose an action hierarchy: the top-level action is a collection of simpler actions represented by smaller POMDPs. Complete solutions (policies) for the smaller POMDPs are combined to provide the total policy. Invoking the top-level policy recursively invokes policies in the hierarchy until a primitive action is reached. POMDPs at all levels are defined over the same state-action space and the relevant space for each POMDP is abstracted using a dynamic belief network. Foka and Trahanias [93] proposed a similar system for autonomous robot navigation. A significant portion of the data for hierarchy and model creation is however application-specific, and has to be hand-coded. Furthermore, even with state-of-the-art approximate POMDP solvers [90] there is an intractability inherent in POMDPs for domains with large, complex state spaces.

Recent work by Sridharan et al. [96,97] presents a novel hierarchical POMDP framework for a cognitive interaction scenario where a human and a robot jointly converse about and interact with objects on a tabletop. Given input images of the scene with several salient regions-of-interest (ROIs), many visual operators that can be applied on the images, and a query to answer, their approach decomposes the overall planning problem into two sub-problems: the problem of choosing *which* ROI to perform processing on, and the problem of deciding *how* to process the chosen ROI. One POMDP is defined at the higher level (HL) where actions consist of choosing the ROI to process, and several POMDPs are defined at the lower level (LL) with one POMDP each for planning the visual operators to apply on a particular ROI. Unlike other hierarchical POMDP approaches, autonomously collected statistics are used to learn the query-specific models at the HL and LL, which are then solved to obtain the overall policy. However, as the state space becomes more complex, a range of hierarchies in the state and action space may be required, in addition to faster POMDP solvers. Still, probabilistic (POMDP) planning methods provide the most promising results, and significant attention is being devoted towards learning the hierarchies autonomously [98].

Table 4 compares the action-planning schemes based on the criteria required for vision-based mobile robot operation. Most

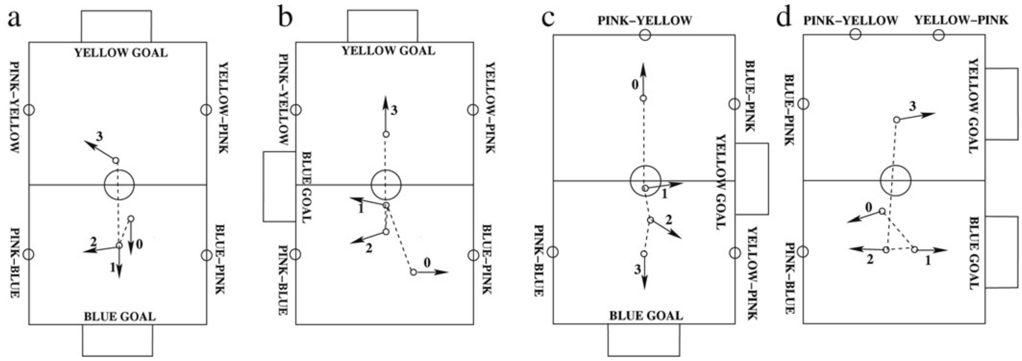

Fig. 9. Action planning for color learning [86] on mobile robots. (a)–(d) Robot plans a sequence of poses (arrow represents the orientation) depending on environmental configuration to learn colors — one color learned at each pose.

Table 4
Action-planning methods — Probabilistic autonomous schemes provide the more promising results on mobile robots.

Algorithms	Computational complexity	Manual training, prior information	Adapt to new situations
Classical methods	High	Yes	No
Reactive methods	Moderate	Very little	No
Continual planning	Low	Yes	No
Robot apps: probabilistic, extensive prior info [87,2]	High	Yes	No
Robot apps: probabilistic, limited prior info [86,97]	Low	Very little	Yes

Fig. 10. An image of the Aibo and robot soccer field.

classical and non-probabilistic planning methods require prior knowledge of states and action outcomes, which may not be feasible to provide in robot applications. Even when extensive manual encoding of information is not required (e.g. the reactive approach [85] or continual planning [82]), the techniques either require several expensive robot trials to build useful world models or fail to capture the non-deterministic action outcomes. Even a large subset of the probabilistic planning algorithms for robot domains require a lot of prior information to be manually encoded [87,2]. Very rarely do robot action-planning methods learn a majority of the required state and action models without human supervision [86,97] — even in such cases some domain knowledge needs to be encoded, though it is significantly easier to provide. To summarize, a combination of learning and probabilistic planning is currently the most appealing direction of research for real-world robot applications that require autonomous operation.

7. Bootstrapping learning and adaptation

Though color learning, illumination invariance and action planning have been discussed as separate modules, there exist interdependencies that can be exploited to create a predominantly autonomous real-time color-based robot vision system. As an example, we describe our work that enables a mobile robot to autonomously plan actions that facilitate color learning and smooth adaptation to illumination change. Our experiments were run on the Sony ERS-7 Aibos (four-legged robots) [99] and other wheeled robot platforms [47].

One standard application domain for the Aibos is the RoboCup initiative [100], where teams of four robots play a competitive game of soccer on an indoor soccer field (4 m × 6 m) — the long-term goal is to create a team of humanoid robots that can compete against a human soccer team under natural conditions by the year 2050. The robot's primary sensor is a color camera in the nose. It has three degrees-of-freedom in each leg, and three more in its head. The field has four color-coded markers spread around the boundary, in addition to the goals. The robot detects these markers in the input images captured at 30 Hz, and localizes itself in order to coordinate its activities with its teammates and score goals against the opponents. All processing for vision, localization, motion and strategy is performed on-board using a 576 MHz processor. The flow of information is similar to that described in Fig. 1 (Section 1), though the architecture is "event-driven". Fig. 10 shows an image of the Aibo and the soccer field.

When the robot is first deployed in the environment, it has no prior knowledge of the color distributions or the possible illuminations. It knows the positions, sizes and color labels of certain objects in its environment (the *structure*), and it knows its approximate starting pose. Existing approaches can be used to learn a significant portion of this structure (e.g. SLAM [9]). The robot uses the known structure to heuristically plan a motion sequence and learn models for the color distributions of interest [101], thereby obtaining the color map under the current illumination. This learning takes ≈ 6 min of robot time instead of hours of human effort. While learning color models, the robot collects random images of the surroundings to model the illumination: color space histograms and second order statistics as described in Section 5. Furthermore, as the robot moves in the environment during and after the color learning, it collects statistics to learn a model for the motion errors and the feasibility of learning colors at various locations (≈ 1 h of autonomous learning).

For all subsequent changes in object configuration or illumination, our algorithm works as follows. Sudden or large illumination changes, for instance turning on (or off) a few lamps in the room, are detected by comparing the current image statistics with the stored image statistics corresponding to the known illuminations. Once a large illumination change is detected, the robot adapts by

Table 5
Time taken to find-and-walk-to-object. The inclusion of $Adapt_{major}$ and $Adapt_{minor}$ enables the robot to operate autonomously over a range of illuminations.

Illum + Alg	Time (s)	Fail
Constant + *NoAdapt*	6.18 ± 0.24	0
Slow + $Adapt_{major}$	31.73 ± 13.88	9
Slow + $Adapt_{major,minor}$	6.24 ± 0.31	0
Sudden + $Adapt_{minor}$	45.11 ± 11.13	13
Sudden + $Adapt_{major,minor}$	9.72 ± 0.51	0
Sudden + *Slow* + $Adapt_{major,minor}$	10.32 ± 0.83	0

using the learned motion error and color learning feasibility models to plan an action sequence that simultaneously maximizes color learning opportunities while minimizing localization errors: a new color map is learned. Minor illumination changes, for instance dimming a few lamps in the room, are detected by tracking the changes in the current color distribution models using Gaussian or histogram merging.

As an example of the experiments performed to evaluate our algorithms, we describe the experiment that tested the autonomous adaptation to illumination changes. We measured the time taken by the robot to *find-and-walk-to-object*: it started out near the center of the field with the object placed near the penalty box of the opponent's goal. Table 5 tabulates the results under six different conditions, averaged over three illumination sources, with 15 trials for each test case. $Adapt_{major}$ and $Adapt_{minor}$ represent the use of the algorithms that enable the robot to detect and adapt to major and minor illumination changes respectively.

When the illumination does not change, the robot can *find-and-walk-to-object* in 6.18 ± 0.24 s. When the illumination is slowly changed as the robot performs the task, using just $Adapt_{major}$ does not help — large variance in the second row. With $Adapt_{minor}$ the results are as good as before (6.24 ± 0.31 s). Next, when the illumination is suddenly changed as the robot starts walking towards the seen object, using just $Adapt_{minor}$ does not help. The robot totally fails to perform the task most of the time, as seen by the large number of failures (fourth row, third column). With the combined strategy, i.e. with both $Adapt_{major}$ and $Adapt_{minor}$, the robot can perform the task efficiently, the additional time being used to ensure that a change in illumination did occur (9.72 ± 0.51 s). We infer that the improvement is primarily due to $Adapt_{major}$. In all these experiments, when the illumination changes significantly, the robot is put in conditions similar to the ones for which it has already learned color and illumination models. Hence, once a major change is detected, $Adapt_{major}$ consists of transitioning to the suitable models. Finally the robot is made to *find-and-walk-to-object* while the illumination is changed significantly initially, and after 3 s is changed slowly over the next 5 s. The robot is able to do the task in 10.32 ± 0.83 s *iff* both $Adapt_{major}$ and $Adapt_{minor}$ are used. We therefore conclude that a combination of the schemes is essential to operate under a range of illumination intensities (≈400 Lux to ≈1600 Lux) and color temperatures (2300 K–4000 K) [67].

Our approach has a couple of key features. First, the robot bootstraps based on the available information. During color learning the robot initially has no knowledge of color distributions and hence chooses to execute only actions that require minimal movement. As the robot learns a few colors it is able to localize better and hence plan and execute subsequent actions more robustly. Second, the robot requires minimal human supervision. The initial specification of the structure and an action sequence suitable for learning are a lot easier to provide than hand-labeling images. All subsequent detection and adaptation to change is autonomous. Furthermore, the learning and adaptation algorithms have been tested in un-engineered indoor corridors on multiple robot platforms, while the action selection algorithm can be used to learn features in addition to color. The results described here are hence a step towards autonomous color-based mobile robot operation — they show that the inter-dependencies between color learning, segmentation, illumination invariance and action planning can be exploited to create an autonomous, learning and adapting mobile robot vision system. Additional videos and images can be viewed online: www.cs.utexas.edu/~AustinVilla/?p=research/illuminvar_colorlearn, www.cs.utexas.edu/~AustinVilla/?p=research/gen_color

8. Summary and future work

Mobile robots possess immense potential in terms of their applicability to a wide range of tasks [1–3] since high-fidelity sensors such as color cameras are becoming available at moderate costs. The sensors however typically require frequent and extensive manual calibration in response to environmental changes. For instance, though vision is a rich source of information, color cameras require extensive manual calibration that is sensitive to illumination changes. The major challenge to the widespread use of mobile robots is hence the ability to learn, adapt and operate autonomously. In this article we have looked at such learning and adaptation in the context of color-based mobile robots operating in dynamic environments. We have especially focused on surveying a set of representative approaches for color segmentation, color learning, illumination invariance and action planning.

A common observation in all the surveyed modules is that many state-of-the-art computer vision and robot vision algorithms fall short of satisfying the desired characteristics that would facilitate autonomous color-based mobile robot operation: real-time operation, deployment without extensive manual training or prior information, and adaptation to previously unseen environmental conditions. The assumptions that are made while solving just the vision problems, predominantly cease to hold in the case of physical agents that interact with the real-world — this is true of several robot tasks in addition to those reviewed in this article. Instead of just modifying existing computer vision algorithms to make them work on mobile robots, robot vision and autonomous mobile robot operation in general, need to be tackled as a separate problem by clearly identifying the feasible assumptions, limiting constraints, and the information available to the robot. It is necessary to focus more on creating *complete, integrated* systems that sense and interact with their surroundings, rather than looking at just subsets of the problem in isolation. The DARPA initiatives [43,44,46], the EU Cognitive Systems (CoSy) project [102], and the RoboCup initiative [100] are all excellent examples of tasks that aim to create such integrated systems. These initiatives force the researchers to identify and tackle key problems that would not arise in the absence of such practical challenges.

In the case of mobile robots operating in the dynamic environments, the availability of high-quality visual sensors at moderate costs presents an alluring alternative to range sensors that offer relatively low-bandwidth information. There are several challenges on the path of completely autonomous operation using visual features such as color because of the inherent sensitivity of visual information to environmental factors such as illumination. However, there exist factors in robot application domains that can be exploited to offset these challenges. Our work, for instance, exploits the fact that many robot environments inherently have a moderate amount of structure. The robot uses the structure to learn models for color distributions, and to detect and adapt to illumination changes. Current algorithms that learn models for color distributions based on structure (known positions, shapes and other characteristics of the objects) [39,41,42,48] require the structure to be pre-specified. Existing algorithms can however be used by the robot to learn a significant portion of this structure. Algorithms for SLAM (Simultaneous Localization and Mapping) are able to use the visual input to simultaneously localize and build a map of the robot's surroundings [9]. In parallel, existing computer vision methods can learn object properties (shape, geometric arrangement etc.) from visual input [6]. One significant step towards autonomous color-based mobile robot operation would therefore be to enable the mobile robot to autonomously learn the required structure and use it to learn models for other environmental features such as color.

We observe from the survey that bootstrapping off of the learned or manually-specified information leads to much more autonomous and robust performance than tackling the individual problems separately. As described in Section 7, it is possible for a mobile robot to plan action sequences that facilitate the learning of models for color distributions, and to use the learned color models to monitor and modify the plan to learn models for other colors robustly. In addition, the robot is able to use the learned models to recognize and adapt to illumination changes, which in turn leads to an appropriate modification of the learned models. Incorporating such bootstrapping between planning, learning and adaptation is therefore an appealing approach to achieve efficient and robust solutions to a range of robot vision problems.

The sensitivity to illumination is a major challenge to the use to color on mobile robots. As seen in Section 5, illumination invariance (i.e. color constancy) has been a major research focus for years, but the problem is still far from being solved. Instead of making unrealistic assumptions, such as being able to measure the properties of the operating environment and/or knowing all possible illuminations ahead of time, robot vision algorithms that have modeled the effects of illumination on color on-board the robot [64,67] have led to robust performance under a range of illuminations. It would be interesting to use the experimental data obtained over years of vision research to model the effects of illuminations on the color distributions in the form of a joint color–illumination mathematical model whose parameters can then be learned autonomously by the robot.

In this article we have primarily looked at color-based mobile robot vision because it is a low-dimensional feature well understood under static conditions, and is easier to analyze under dynamic conditions. However, the lessons learned here are also applicable to challenges faced when working with other visual features and other sensory inputs (e.g. range sensors). Similar algorithms that bootstrap and learn feature models in real-time can be devised for other visual features such as texture or local gradients [5]. An even more challenging problem is to fuse the information from a set of sensors, each with a varying degree of unreliability, which may have complementary properties. A combination of a laser range finder and a color camera is an example of such a complementary set of sensors. Such sensor fusion requires a probabilistic representation of the performance of the sensory sources, which can be learned by the robot. The learned representations can then be used to learn how the information from the sensors can be combined to obtain a single robust representation of the current state of the world. In the DARPA challenges [44,3] for instance, sensor fusion continues to be a major challenge.

When there are multiple sources of information, multiple actions that can be performed by the robot, and multiple information-processing operators that can be applied on the visual input, it is useful for the robot to plan a sequence of actions to achieve the desired goal reliably and optimally. Optimal use of resources is important for robots that interact with the real world through noisy sensors and actuators, because they need to respond to dynamic changes. Classical planning methods [78] require prior knowledge of the states and action outcomes, and they do not model uncertainty effectively. Posing this planning problem as a POMDP has yielded the most promising results because POMDPs elegantly capture the features of mobile robot domains: partially observable states and non-deterministic actions. POMDPs also provide the capability to simultaneously maximize reliability and minimize resource usage. The inherent intractability of modeling practical problems as POMDPs has been partially overcome by imposing a structure or hierarchy on the problem [93,2,97] based on domain knowledge. Unlike existing work on probabilistic planning, our work in the robot soccer domain [86] and the human–robot interaction domain [97] shows the feasibility of using autonomously learned models of action outcomes to generate robust plans to achieve the desired goal. In the future even the hierarchical decomposition may be learned [98], and a range of state and action hierarchies [2] may be used to formulate complex real-world problems as POMDPs. Furthermore, a hybrid combination of deterministic and probabilistic planning schemes may be used to analyze complex human–robot interaction scenarios.

In the long term, enabling robots to operate in the real-world would require autonomous operation at a much larger scale. The robot would need capabilities such as spatial and semantic reasoning [103], analysis of 3D geometric structure, and natural language processing during human–robot interactions [104]. As the focus shifts towards developing such "cognitive" abilities on a mobile robot, including the ability to reason and bind information across different sensing modalities, it becomes necessary to design architectures that enable autonomous learning and adaptation in complex systems [102]. Ultimately, the goal is to enable widespread mobile robot deployment in a range of applications, with the robots operating autonomously and interacting with humans under completely uncontrolled natural conditions.

Acknowledgments

We thank the members of the UTAustinVilla team who contributed part of the code used to test our algorithms. This work was sponsored in part by NSF CAREER award IIS-0237699 and ONR YIP award N00014-04-1-0545.

References

[1] B.W. Minten, R.R. Murphy, J. Hyams, M. Micire, Low-order-complexity vision-based docking, IEEE Transactions on Robotics and Automation 17 (6) (2001) 922–930.

[2] J. Pineau, M. Montemerlo, M. Pollack, N. Roy, S. Thrun, Towards robotic assistants in nursing homes: Challenges and results, Robotics and Autonomous Systems, Special Issue on Socially Interactive Robots 42 (3–4) (2003) 271–281.

[3] S. Thrun, Stanley: The robot that won the DARPA grand challenge, Journal of Field Robotics 23 (9) (2006) 661–692.

[4] S. Thrun, D. Fox, W. Burgard, F. Dellaert, Robust Monte Carlo localization for mobile robots, Journal of Artificial Intelligence 128 (1–2) (2001) 99–141.

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

[6] R. Fergus, P. Perona, A. Zisserman, Object class recognition by unsupervised scale-invariant learning, in: CVPR, 2003.

[7] S. Fidler, A. Leonardis, Towards scalable representations of object categories: Learning a hierarchy of parts, in: The International Conference on Computer Vision and Pattern Recognition, CVPR, 2007.

[8] P. Viola, M. Jones, Rapid object detection using a boosted cascade of simple features, in: The International Conference on Computer Vision and Pattern Recognition, CVPR, 2001.

[9] P. Jensfelt, J. Folkesson, D. Kragic, H.I. Christensen, Exploiting distinguishable image features in robotic mapping and localization, in: The European Robotics Symposium, EUROS, 2006.

[10] D.A. Forsyth, J. Ponce, Computer Vision: A Modern Approach, Prentice Hall, 2002.

[11] G.N. Desouza, A.C. Kak, Vision for mobile robot navigation: A Survey, IEEE Transactions of Pattern Analysis and Machine Intelligence 24 (2) (2002) 237–267.

[12] D. Kortenkamp, R.P. Bonasso, R. Murphy, AI-based Mobile Robots: Case Studies of Successful Robot Systems, MIT Press, 1998.

[13] D. Comaniciu, P. Meer, Mean shift: A robust approach towards feature space analysis, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (5) (2002) 603–619.

[14] J. Shi, J. Malik, Normalized cuts and image segmentation, in: IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 22 (8).

[15] B. Sumengen, B.S. Manjunath, C. Kenney, Image segmentation using multi-region stability and edge strength, in: The IEEE International Conference on Image Processing, ICIP, 2003.

[16] R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification, 2nd ed., Wiley Publishers, 2000.

[17] V. Caselles, R. Kimmel, G. Shapiro, Geodesic active contours, International Journal of Computer Vision 22 (1) (1997) 61–79.

[18] T.F. Chan, L.A. Vese, Active contours without edges, IEEE Transactions on Image Processing 10 (2) (2001) 266–277.

[19] N. Paragios, R. Deriche, Geodesic active regions for supervised texture segmentation, in: The International Conference on Computer Vision, ICCV, 1999.

[20] P.F. Felzenswalb, D.P. Huttenlocher, Efficient graph-based image segmentation, International Journal of Computer Vision 59 (2) (2004) 167–181.

[21] Z. Wu, R. Leahy, An optimized graph-theoretic approach to data clustering, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 11 (1993) 1101–1113.

[22] J. Shi, J. Malik, Motion segmentation and tracking using normalized cuts, in: The International Conference on Computer Vision, ICCV, 1998.

[23] D. Hoiem, A. Efros, M. Hebert, Recovering surface layout from an image, International Journal of Computer Vision (IJCV) 75 (1) (2007) 151–172.

[24] C. Pantofaru, M. Hebert, A comparison of image segmentation algorithms, cmu-ri-tr-05-40, Tech. Rep., Robotics Institute, Carnegie Mellon University, September 2005.

[25] T. Cour, F. Benezit, J. Shi, Spectral segmentation with multiscale graph decomposition, in: The International Conference of Computer Vision and Pattern Recognition, CVPR, 2005.

[26] W. Uther, S. Lenser, J. Bruce, M. Hock, M. Veloso, Cm-pack'01: Fast legged robot walking, robust localization, and team behaviors, in: The Fifth International RoboCup Symposium, Seattle, USA, 2001.

[27] C.M. Bishop, Neural Networks for Pattern Recognition, Oxford University Press, 1995.

[28] S. Chen, M. Siu, T. Vogelgesang, T.F. Yik, B. Hengst, S.B. Pham, C. Sammut, RoboCup-2001: The Fifth RoboCup Competitions and Conferences, Springer-Verlag, Berlin, 2002.

[29] D. Cohen, Y.H. Ooi, P. Vernaza, D.D. Lee, RoboCup-2003: The Seventh RoboCup Competitions and Conferences, Springer-Verlag, Berlin, 2004.

[30] P. Stone, K. Dresner, P. Fidelman, N.K. Jong, N. Kohl, G. Kuhlmann, E. Lin, M. Sridharan, D. Stronger, UT Austin Villa 2004: Coming of age, TR 04-313, Tech. Rep., Department of Computer Sciences, UT-Austin, October 2004.

[31] M. Sridharan, P. Stone, Real-time vision on a mobile robot platform, in: The IEEE International Conference on Intelligent Robots and Systems, IROS, 2005.

[32] T. Gevers, A.W.M. Smeulders, Color-based object recognition, Pattern Recognition 32 (3) (1999) 453–464.

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

[34] Y.B. Lauziere, D. Gingras, F.P. Ferrie, Autonomous physics-based color learning under daylight, in: The EUROPTO Conference on Polarization and Color Techniques in Industrial Inspection, vol. 3826, 1999.

[35] J.M.G. Lammens, A computational model of color perception and color naming, Ph.D. Thesis, Computer Science Department, State University of New York at Buffalo, NY, June 1994.

[36] A. Mojsilovic, A computational model for color naming and describing color composition of images, IEEE Transactions on Image Processing 14 (5) (2005) 690–699.

[37] J.D. Foley, A.V. Dam, S.K. Feiner, J.F. Hughes, Computer Graphics: Principles and Practice, Addison-Wesley, 1995.

[38] J. Hyams, M.W. Powell, R.R. Murphy, Cooperative navigation of micro-rovers using color segmentation, Autonomous Robots 9 (1) (2000) 7–16.

[39] D. Cameron, N. Barnes, Knowledge-based autonomous dynamic color calibration, in: The International RoboCup Symposium, 2003.

[40] Y. Rubner, C. Tomasi, L.J. Guibas, The earth mover's distance as a metric for image retrieval, International Journal of Computer Vision 40 (2) (2000) 99–121.

[41] S. Chernova, E. Crawford, M. Veloso, Acquiring observation models through reverse plan monitoring, in: The 12th Portuguese Conference on Artificial Intelligence, EPIA, 2005.

[42] M. Jungel, Using layered color precision for a self-calibrating vision system, in: The Eighth International RoboCup Symposium, Lisbon, Portugal, 2004.

[43] DARPA, The DARPA Grand Challenge, http://www.grandchallenge.org/, 2005.

[44] LAGR, The DARPA learning applied to ground robots challenge, www.darpa.mil/ipto/programs/lagr/, 2005.

[45] UPenn-LAGR, The UPenn LAGR Team, http://ddlee.com/LAGR/, 2006.

[46] DARPA, The DARPA Urban Robot Challenge, http://www.darpa.mil/grandchallenge/index.asp, 2007.

[47] M. Sridharan, Robust structure-based autonomous color learning on a mobile robot, Ph.D. Thesis, Electrical and Computer Engineering, The University of Texas at Austin, USA, August 2007.

[48] M. Sridharan, P. Stone, Autonomous color learning on a mobile robot, in: The Twentieth National Conference on Artificial Intelligence, AAAI, 2005.

[49] A.J. Davison, I.D. Reid, N.D. Morton, O. Stasse, MonoSLAM: Real-time single camera SLAM, Pattern Analysis and Machine Intelligence (PAMI) 29 (6) (2007) 1052–1067.

[50] G. Finlayson, S. Hordley, P. Hubel, Color by correlation: A simple unifying framework for color constancy, IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (11) (2001) 1209–1221.

[51] L.T. Maloney, B.A. Wandell, Color constancy: A method for recovering surface spectral reflectance, Journal of Optical Society of America A 3 (1) (1986) 29–33.

[52] E.H. Land, The retinex theory of color constancy, Scientific American (1977) 108–129.

[53] G. Buchsbaum, A spatial processor model for object color perception, Journal of Franklin Institute 310 (1980) 1–26.

[54] D.H. Brainard, B.A. Wandell, Analysis of the retinex theory of color vision, Journal of Optical Society of America A 3 (10) (1986) 1651–1661.

[55] D. Forsyth, A novel algorithm for color constancy, International Journal of Computer Vision 5 (1) (1990) 5–36.

[56] G. Finlayson, Color in perspective, IEEE Transactions of Pattern Analysis and Machine Intelligence 18 (10) (1996) 1034–1038.

[57] G. Finlayson, S. Hordley, Improving gamut mapping color constancy, IEEE Transactions on Image Processing 9 (10) (2000) 1774–1783.

[58] D.H. Brainard, W.T. Freeman, Bayesian color constancy, Journal of Optical Society of America A 14 (7) (1997) 1393–1411.

[59] Y. Tsin, R.T. Collins, V. Ramesh, T. Kanade, Bayesian color constancy for outdoor object recognition, in: The International Conference of Computer Vision and Pattern Recognition, CVPR, 2001.

[60] C. Rosenberg, M. Hebert, S. Thrun, Color constancy using KL-divergence, in: The IEEE International Conference on Computer Vision, ICCV, 2001.

[61] F. Anzani, D. Bosisio, M. Matteucci, D.G. Sorrenti, On-line color calibration in non-stationary environments, in: The Ninth International RoboCup Symposium, Osaka, Japan, 2005.

[62] S. Lenser, M. Veloso, Automatic detection and response to environmental change, in: The International Conference of Robotics and Automation, ICRA, 2003.

[63] S.R. Lenser, On-line robot adaptation to environment change, Ph.D. Thesis, Computer Science Department, Carnegie Mellon University, Pittsburgh, PA, July 2005.

[64] D. Schulz, D. Fox, Bayesian color estimation for adaptive vision-based robot localization, in: The IEEE International Conference on Intelligent Robots and Systems, IROS, 2004.

[65] S. Thrun, W. Burgard, D. Fox, Probabilistic Robotics, MIT Press, Cambridge, USA, 2005.

[66] M. Sridharan, P. Stone, Structure-based color learning on a mobile robot under changing illumination, Autonomous Robots 23 (3) (2007) 161–182.

[67] M. Sridharan, P. Stone, Global action selection for illumination invariant color modeling, in: The IEEE International Conference on Intelligent Robots and Systems, IROS, 2007.

[68] R.L. Sutton, A.G. Barto, Reinforcement Learning: An Introduction, MIT Press, Cambridge, USA, MA, 1998.

[69] J.Y. Aloimonos, I. Weiss, A. Bandhopadhay, Active vision, International Journal of Computer Vision (1987) 333–356.

[70] R. Clouard, A. Elmoataz, C. Porquet, M. Revenu, Borg: A knowledge-based system for automatic generation of image processing programs, IEEE Transactions on Pattern Analysis and Machine Intelligence 21 (2) (1999) 128–144.

[71] S. Moisan, Program supervision: Yakl and pegase+ reference and user manual, Rapport de Recherche 5066, INRIA, Sophia Antipolis, France, December 2003.

[72] M. Thonnat, S. Moisan, What can program supervision do for program reuse? IEE Proceedings on Software 147 (5) (2000) 179–185.

[73] C. Shekhar, S. Moisan, M. Thonnat, Use of a real-time perception program supervisor in a driving scenario, in: Intelligent Vehicle Symposium '94, Paris, France, 1994.

[74] X. Jiang, H. Bunke, Vision planner for an intelligent multisensory vision system, Technical Report, University of Bern, Bern, Switzerland, 1994.

[75] T. Darrell, Reinforcement learning of active recognition behaviors, Tech. Rep. Interval Research Technical Report 1997-045, extended version of paper in NIPS-95, 1997, pp. 858–864.

[76] L. Kaelbling, M. Littman, A. Cassandra, Planning and acting in partially observable stochastic domains, Artificial Intelligence 101 (1998) 99–134.

[77] L. Li, V. Bulitko, R. Greiner, I. Levner, Improving an adaptive image interpretation system by leveraging, in: Australian and New Zealand Conference on Intelligent Information Systems, 2003.

[78] M. Ghallab, D. Nau, P. Traverso, Automated Planning: Theory and Practice, Morgan Kaufmann, San Francisco, CA 94111, 2004.

[79] D. Draper, S. Hanks, D. Weld, A probabilistic model of action for least-commitment planning with information gathering, in: Uncertainty in AI, UAI, 1994.

[80] R. Petrick, F. Bacchus, Extending the knowledge-based approach to planning with incomplete information and sensing, in: International Conference on Automated Planning and Scheduling, ICAPS, 2004.

[81] S.J. Russell, P. Novig, Artificial Intelligence: A Modern Approach, 2nd ed., Prentice Hall, Upper Saddle River, New Jersey, 2003.

[82] M. Brenner, B. Nebel, Continual planning and acting in dynamic multiagent environments, in: The International Symposium of Practical Cognitive Agents and Robots, 2006.

[83] D. McDermott, PDDL: The planning domain definition language, Technical Report TR-98-003/DCS TR-1165, Tech. Rep., Yale Center for Computational Vision and Control, 1998.

[84] J. Hoffmann, B. Nebel, The FF planning system: Fast plan generation through heuristic search, Journal of Artificial Intelligence Research 14 (2001) 253–302.

[85] R.A. Brooks, A robust layered control system for a mobile robot, Robotics and Automation 2 (1986) 14–23.

[86] M. Sridharan, P. Stone, Comparing two action planning approaches for color learning on a mobile robot, in: The VISAPP International Workshop on Robot Perception, 2008.

[87] C. Boutillier, T.L. Dean, S. Hanks, Decision theoretic planning: Structural assumptions and computational leverage, Journal of AI Research 11 (1999) 1–94.

[88] J. Goldsmith, M. Mundhenk, Complexity issues in Markov decision processes, in: IEEE Conference on Computational Complexity, 1998.

[89] B. Bonet, H. Geffner, Faster heuristic search algorithms for planning with uncertainty and full feedback, in: The International Joint Conference on Artificial Intelligence, IJCAI, 2003.

[90] T. Smith, R. Simmons, Point-based POMDP algorithms: Improved analysis and implementation, in: UAI, 2005.

[91] D. Aberdeen, A survey of approximate methods for solving partially observable markov decision processes, Technical Report, Australia National University, School of Information Science and Engineering, 2002.

[92] M. Hauskrecht, Value function approximations of partially observable Markov decision processes, Journal of Artificial Intelligence 13 (2000) 33–95.

[93] A.F. Foka, P.E. Trahanias, Real-time hierarchical pomdps for autonomous robot navigation, in: IJCAI Workshop on Reasoning with Uncertainty in Robotics, 2005.

[94] J.M. Porta, M.T.J. Spaan, N. Vlassis, Robot planning in partially observable continuous domains, Robotics: Science and Systems (2005).

[95] T. Dietterich, The MAXQ method for hierarchical reinforcement learning, in: International Conference on Machine Learning, ICML, 1998.

[96] M. Sridharan, J. Wyatt, R. Dearden, E-HiPPo: Extensions to hierarchical POMDP-based visual planning on a robot, in: The 27th PlanSIG Workshop, 2008.

[97] M. Sridharan, J. Wyatt, R. Dearden, HiPPo: Hierarchical POMDPs for planning information processing and sensing actions on a robot, in: International Conference on Automated Planning and Scheduling, ICAPS, 2008.

[98] M. Toussaint, L. Charlin, P. Poupart, Hierarchical POMDP controller optimization by likelihood maximization, in: Uncertainty in AI, UAI, 2008.

[99] The Sony Aibo robots, http://www.sonystyle.com, 2004.

[100] H. Kitano, M. Asada, Y. Kuniyoshi, I. Noda, E. Osawa, Robocup: The robot world cup initiative, in: International Conference on Autonomous Agents, ICAA, 1997.

[101] M. Sridharan, P. Stone, Autonomous planned color learning on a mobile robot without labeled data, in: The Ninth IEEE International Conference on Control, Automation, Robotics and Vision, ICARCV, 2006.

[102] N. Hawes, A. Sloman, J. Wyatt, M. Zillich, H. Jacobsson, G.-J. Kruiff, M. Brenner, G. Berginc, D. Skocaj, Towards an integrated robot with multiple cognitive functions, in: The Twenty-second National Conference on Artificial Intelligence, AAAI, 2007.

[103] B.J. Kuipers, in: Margaret Jefferies, Albert (Wai-Kiang) Yeap (Eds.), An Intellectual History of the Spatial Semantic Hierarchy, Springer-Verlag, 2008.

[104] G.-J.M. Kruijff, H. Zender, P. Jensfelt, H.I. Christensen, Situated dialogue and spatial organization: What, where…and why? International Journal of Advanced Robotic Systems, Special Issue on Human–Robot Interaction 4 (1) (2007) 125–138.

Mohan Sridharan is an Assistant Professor at Texas Tech University. He was a Research Fellow in the School of Computer Science at the University of Birmingham (UK) between Sep 07 and Sep 08. He received his Ph.D. in Electrical and Computer Engineering from The University of Texas at Austin in August 2007. His research interests include robotics, computer vision, cognitive systems and statistical machine learning.

Peter Stone is an Alfred P. Sloan Research Fellow and Associate Professor in the Department of Computer Sciences at The University of Texas at Austin. He received his Ph.D. in Computer Science in 1998 from Carnegie Mellon University. From 1999 to 2002 he was a Senior Technical Staff Member in the Artificial Intelligence Principles Research Department at AT&T Labs - Research. Peter's research interests include machine learning, multiagent systems, robotics, and e-commerce. In 2003, he won a CAREER award from the National Science Foundation for his research on learning agents in dynamic, collaborative, and adversarial multiagent environments. In 2004, he was named an ONR Young Investigator for his research on machine learning on physical robots. Most recently, he was awarded the prestigious IJCAI 2007 Computers and Thought award.