Bayesian Color Estimation for Adaptive Vision-based Robot Localization

Dirk Schulz
Department of Computer Science & Engineering
University of Washington
Seattle, WA 98195, USA
Email: schulz@cs.washington.edu

Dieter Fox
Department of Computer Science & Engineering
University of Washington
Seattle, WA 98195, USA
Email: fox@cs.washington.edu

***Abstract*— In this article we introduce a hierarchical Bayesian model to estimate a set of colors with a mobile robot. Estimating colors is particularly important if objects in an environment can only be distinguished by their color. Since the appearance of colors can change due to variations in the lighting condition, a robot needs to adapt its color model to such changes. We propose a two level Gaussian model in which the lighting conditions are estimated at the upper level using a switching Kalman filter. A hierarchical Bayesian technique learns Gaussian priors from data collected in other environments. Furthermore, since estimation of the color model depends on knowledge of the robot's location, we employ a Rao-Blackwellised particle filter to maintain a joint posterior over robot positions and lighting conditions. We evaluate the technique in the context of the RoboCup AIBO league, where a legged AIBO robot has to localize itself in an environment similar to a soccer field. Our experiments show that the robot can localize under different lighting conditions and adapt to changes in the lighting condition, for example, due to a light being turned on or off.**

I. Introduction

Estimating the state of a robot and its environment is a fundamental task in mobile robotics. Most of the existing approaches to state estimation in robotics rely on proximity sensors such as ultrasound sensors or laser range-finders. This is mostly due to the fact that vision-based techniques require more complex world models and sophisticated processing of the raw image data. Recently, vision-based state estimation has gained increased interest in the robotics community. A major reason for this increased interest is the RoboCup challenge, which requires mobile robots to autonomously play soccer (see `www.robocup.org`). In the context of RoboCup, vision is the most important sensor, since camera information is needed to localize the robot, detect the ball, and distinguish between friendly and opponent robots (cf. Fig. 1).

The RoboCup domain poses some challenging problems for state estimation, including moving objects, limited computational power, and low resolution camera information. Fortunately, the environment in RoboCup is highly constrained in that an accurate map of the environment is known in advance and objects can be identified by their uniform colors. To make use of colors, a robot has to map the raw color values observed with its camera to the colors in the map. The key problem in this context is that the appearance of these colors can change drastically under different lighting conditions. For this reason, most vision systems used in RoboCup rely on color classification approaches which are manually adjusted to the pertinent lighting conditions before a game.

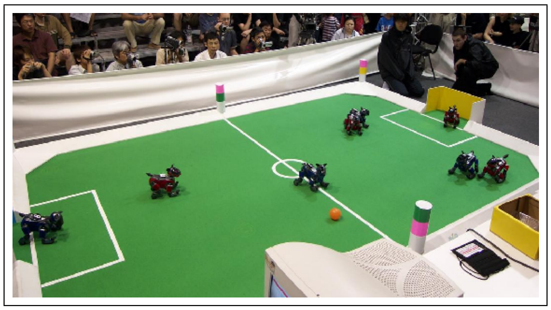

Fig. 1. Legged AIBO robots during a RoboCup match. Color information is extremely important since all robots have the same shape and the outline of the environment is completely symmetric.

The main contribution of this article is a hierarchical Gaussian color model which enables a robot to quickly adapt to different lighting conditions. The lower level of the color model represents each color in the map by a Gaussian distribution in the raw YUV color space. This level assumes that the lighting conditions in the environment are known and fixed. Uncertainty in lighting conditions is represented at the upper level of the model by a joint Gaussian distribution over the means of all lower level Gaussians. The estimates are updated using a switching Kalman filter, which makes the system robust even to sudden changes in the lighting conditions. Hierarchical Bayesian learning is used to extract adequate priors distributions from data collected in other environments. In contrast to the previous approaches, our model takes the dependencies between different colors into account, which allows to adapt even unobserved colors. We apply our technique in the context of vision-based localization. The approach uses a Rao-Blackwellised particle filter to estimate the joint posterior over robot positions and lighting conditions. Our experiments demonstrate that an AIBO robot can effectively localize in its environment using our approach without color calibration and even under rapidly changing lighting conditions.

In the following section we introduce the hierarchical color model. We first show how the color model can be updated given the robot's position and describe how a Bayesian prior over lighting conditions can be learned. The next section then describes the Rao-Blackwellised particle filter to simultaneously localize the robot and update the color model.

Before concluding, we present experiments that demonstrate the advantages of our technique.

II. Related Work

Dealing with changing lighting conditions is an important problem in computer vision. The most general approach to solve the problem is known as color constancy, where the aim is to determine the physical properties of the illuminant from images (see, *e.g.*, [10]). In principal, such an approach allows to compensate for any change in the lighting condition, if the appearance of the colors under some reference illuminant is known. However, the algorithms make simplifying assumptions about properties of surfaces in the images, which can lead to failure to determine the true illuminant. Most adaptive vision applications rely on highly specialized solutions. In the surveillance domain, for example, Gaussian mixtures are used to learn and adapt color models of individual pixels of camera images, in order to reliably distinguish between moving objects and static background [11].

Techniques for dealing with varying lighting conditions have gained increased interest in RoboCup as well. [3] discretize the color space using a grid and adapt independent multinomials over the color class membership for the individual cells. While this technique allows for efficient color classification and updates using lookup tables, it ignores dependencies between color cells. [9] use independent thresholds on the U and V channels of the colors to facilitate edge detection. Their technique makes very restrictive assumptions and it is not clear how it can be applied beyond RoboCup. Finally, [8] propose an active contour approach to ball tracking and robot localization. Instead of relying on color classification, expectation maximization is employed to fit a contour to objects by maximizing local image statistics at the object boundary. Their approach does not take varying lighting conditions into account and could be combined with our technique to achieve adaptivity. In contrast to existing techniques, our approach performs a joint estimate over all colors in an environment and performs a joint estimate over robot locations and lighting conditions. Conditioning on the robot's location allows us to learn the lighting condition from the correct pixel labels obtained from a 3d map of the environment.

III. Adaptive, Hierarchical Color Model

In this section we present our two level Gaussian color model. The model is based on the assumption that the location of the robot is known for each camera image. We will show how to concurrently estimate the robot location in the subsequent section. The task of the color model is to determine the likelihood that a color observed by the robot's camera corresponds to a certain color in the map of the environment. In our application, raw color values, denoted c, are given in YUV-space. The map describes the geometry and colors of the soccer field, the robots, and the ball.

The lower level of the color model represents the map colors independently, each by a Gaussian distribution in YUV-space. More specifically, for known and non-changing lighting conditions, we assume that the YUV values c corresponding to map color m, with $1 \leq m \leq M$, are distributed according to a Gaussian:

$$p(c \mid m) \quad = \quad \mathcal{N}(c\,;\theta[m],\nu) \tag{1}$$

Here $\theta[m] = \langle y_m, u_m, v_m \rangle$ is the mean of the Gaussian representing map color m. The covariance matrix ν is assumed to be diagonal and identical for all map colors. Under these assumptions, the low level color model is completely specified by a $3M$ dimensional mean vector $\theta = \langle \theta[1], \ldots, \theta[M] \rangle$. We will now describe the upper level, which estimates lighting conditions, *i.e.*, distributions over mean vectors θ.

A. Estimating lighting conditions

1) Linear Gaussian model: Whenever a robot is placed into a new environment, it has to estimate the lighting conditions of this environment. In Bayesian filtering, this is done by estimating the posterior over the lighting conditions (color mean vector θ) conditioned on the data observed in the environment. The data up to time k consists of a sequence of camera images $z_{1:k}$. For now, let us assume that we additionally know the locations $x_{1:k}$ at which these images were taken. Using the 3d map of the environment along with these locations allows us to determine the map color m for each pixel in each camera image. By averaging over the YUV values of the pixels assigned to each map color, it is possible to extract a color mean vector $\bar{\iota}_j$ from each image z_j [1]. These color mean vectors are used to estimate the posterior over lighting conditions. Under the assumption that the prior distribution over mean vectors is Gaussian and that the change in lighting conditions is approximately linear, we can estimate this posterior using a linear Kalman filter.

To see, let μ_0 and Σ_0 denote the mean and covariance of the Gaussian prior over color mean vectors θ. The $3M \times 3M$ covariance matrix represents the uncertainty in the lighting condition as a robot enters a new environment. Under the linearity assumptions, the posterior over lighting conditions at time k, denoted θ_k, is Gaussian:

$$p(\theta_k \mid \bar{\iota}_{1:k}) \quad = \quad \mathcal{N}(\theta_k;\, \mu_k, \Sigma_k) \tag{2}$$

The posterior parameters can be estimated recursively as

$$\mu_k \quad = \quad \mu_{k-1} + K_k(\bar{\iota}_k - \mu_{k-1}) \tag{3}$$

$$\Sigma_k \quad = \quad (I - K_k)\,\bar{\Sigma}_k, \tag{4}$$

where K_k and $\bar{\Sigma}_k$ are the so-called Kalman gain and predicted state covariance:

$$K_k \quad = \quad \bar{\Sigma}_k\,(\bar{\Sigma}_k + \nu)^{-1} \tag{5}$$

$$\bar{\Sigma}_k \quad = \quad \Sigma_{k-1} + R \tag{6}$$

This recursive update scheme follows directly from the general Kalman filter updates [2] under the additional assumptions that the lighting conditions have a random drift over time and that the color vector $\bar{\iota}_k$ can be treated as a direct observation of the current lighting condition (*i.e.*,

[1] Not every camera image contains all M map colors. However, missing values can be filled in with the most likely value using linear regression conditioned on the observed colors and the current estimate of the lighting condition.

no transformation from state space to observation space is needed). The $3M \times 3M$ matrix R in (6) models the drift of lighting conditions by adding uncertainty at each time step. This allows us to assume rather constant lighting conditions over the complete field, smooth changes are tracked by the drift of the Kalman filter.

2) Likelihood function: In our hierarchical model, the likelihood function (1) has to be modified so as to consider the uncertainty in lighting conditions. Let $\langle \mu, \Sigma \rangle$ denote the current estimate of the lighting conditions, then the likelihood of observing a YUV value c given the map color m can be computed in closed form:

$$\begin{aligned} p(c \mid m, \langle \mu, \Sigma \rangle) &= \int p(c \mid m, \theta)\, p(\theta \mid \langle \mu, \Sigma \rangle)\, \mathrm{d}\theta && (7) \\ &= \int \mathcal{N}(c\,;\, \theta[m], \nu)\, \mathcal{N}(\theta\,;\, \mu, \Sigma)\, \mathrm{d}\theta && (8) \\ &= \mathcal{N}(c\,;\, \mu|_m, \Sigma|_m + \nu). && (9) \end{aligned}$$

(7) integrates the original likelihood function (1) over the lighting conditions. The left term in (8) is the lower level Gaussian over color values, and the right term is the upper level Gaussian over lighting conditions. (9) follows then from the properties of Gaussians, with $\mu|_m$ and $\Sigma|_m$ denoting the mean and variance of the marginal distribution of $\langle \mu, \Sigma \rangle$ for the m-th color class. Note, that in the hierarchical model, the covariance of each color is bounded from below by ν, while in a flat model, which only uses the higher level Gaussian, the covariance can actually shrink to zero.

3) Non-linear model: So far, we made the assumption that lighting conditions can be modeled as a linear Gaussian process. Unfortunately, these assumptions are often violated, for example, when using different light sources such as artificial and natural light [5], or by sudden changes in brightness due to turning a light on or off. Such non-linearities can be accommodated by estimating the lighting condition using a switching Kalman filter [1]. Intuitively, such a technique estimates the state of a system using multiple Kalman filters, each filter tuned towards specific circumstances. At each iteration, the system is allowed to switch between filters, and the switching probabilities are governed by a first-order Markov process. In our context, for example, this model can use different filters for different light sources and it can switch between filters when a light is turned on or off. The switching filter is implemented using a Rao-Blackwellised particle filter, where the transitions between filters are sampled using a particle filter, and each particle is annotated with a Kalman filter that estimates the lighting conditions as described in the previous section. The details of this technique are beyond the scope of this paper, see [1] for further information. Later, we will apply Rao-Blackwellised particle filters to estimate camera locations.

B. Learning prior lighting conditions

Estimating lighting conditions requires the availability of Gaussian priors to initialize the different filters of the switching Kalman filter. Our technique learns these priors from data collected in previous environments. The different priors are learned independently by splitting the data into different collections, each collection containing sets of images observed in environments with similar lighting conditions. In our current system, the data is split using k-means clustering.

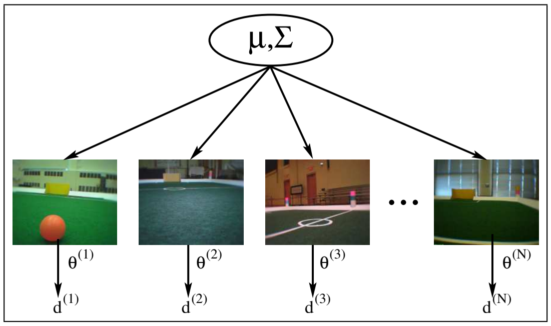

Fig. 2. Hierarchical Bayesian model: The lighting condition in each environment is represented by a Gaussian for the individual map colors, parameterized by $\langle \theta^{(j)}, \nu \rangle$. The lighting conditions are drawn from a common Gaussian distribution represented by the so-called hyperparameters $\langle \mu, \Sigma \rangle$.

We will now describe how to use a hierarchical Bayesian technique so as to learn a consistent prior $\langle \mu, \Sigma \rangle$ from one of these collections (see [7] for details). For clarity, the model is illustrated in Fig. 2. We assume that the data collection contains N data sets $d^{(j)}$, each consisting of a sequence of mean vectors $\bar{\iota}_{1:k}$ distributed according to the lighting conditions $\langle \theta^{(j)}, \nu \rangle$ in the specific environment j. The hierarchical model additionally assumes that the lighting conditions of all environments are drawn from a common Gaussian distribution, parameterized by the hyperparameters $\langle \mu, \Sigma \rangle$. Our goal is to determine the values $\langle \mu_0, \Sigma_0 \rangle$ of the hyperparameters that are optimal as prior parameters when a robot is placed in any of these environments. We can compute these values by maximizing the posterior over the hyperparameters conditioned on the data sets:

$$\langle \mu_0, \Sigma_0 \rangle \quad = \quad \underset{\langle \mu, \Sigma \rangle}{\operatorname{argmax}}\, p(\langle \mu, \Sigma \rangle \mid d^{(1)}, ..., d^{(N)}) \qquad (10)$$

In the hierarchical model, the computation of the posterior requires integration over the lighting conditions in the individual environments. Fortunately, for our Gaussian model, this integration can be done in closed form [7]. The prior values $\langle \mu_0, \Sigma_0 \rangle$ are then computed by numerically maximizing the posterior. The key advantage of this hierarchical technique is that it provides us with a statistically sound way for learning Gaussian priors. As noted above, different priors can be learned for differing lighting conditions, for example, for indoor and outdoor light, or for bright and dark environments.

To summarize, the lower level of our model uses independent Gaussians to map raw YUV values to the colors represented in a 3d map of an environment. Lighting conditions are estimated using a switching Kalman filter that maintains a joint Gaussian distribution over the color means of the lower level Gaussians. This joint estimate allows us to model dependencies between different colors. The Kalman filters are initialized by prior distributions that are learned from previous environments using a hierarchical Bayesian statistical approach. So far, we assumed that the robot and camera locations are known for each image. In the next section, we will show how to estimate the lighting conditions even when the robot's position is not known with certainty.

IV. Simultaneous Camera Localization and Lighting Condition Estimation

The estimation of lighting conditions described in the previous section requires knowledge of the camera (robot) trajectory through the environment. To estimate the camera locations, however, one needs to have an estimate of the lighting conditions in order to determine which pixel values in the images correspond to which objects in the environment [2]. Hence, since these two estimation problems are tightly coupled, it is necessary to estimate the joint posterior over lighting conditions θ_k and the camera locations $x_{1:k}$.

To deal with the highly non-linear motion of a legged robot (and especially the resulting camera motion), we estimate locations of the camera using a particle filter. The joint estimate can than be performed efficiently using a Rao-Blackwellised particle filter (RBPF) [4]. RBPFs combine the representational benefits of particle filters with the efficiency and accuracy of Kalman filters by sampling the non-linear parts of a state estimation problem and solving the linear parts using Kalman filters conditioned on the samples.

A. Rao-Blackwellised particle filter

We will first factorize the posterior over camera trajectories $x_{1:k}$ and lighting conditions θ_k as follows.

$$\begin{aligned} & p(\theta_k, x_{1:k} \mid z_{1:k}, u_{1:k-1}) \\ & \quad = p(\theta_k \mid x_{1:k}, z_{1:k}, u_{1:k-1})\, p(x_{1:k} \mid z_{1:k}, u_{1:k-1}) \quad (11) \\ & \quad = p(\theta_k \mid \bar{\iota}_{1:k})\, p(x_{1:k} \mid z_{1:k}, u_{1:k-1}) \quad (12) \end{aligned}$$

Here, $z_{1:k}$ are the camera images observed up to time k, and $u_{1:k-1}$ denote the robot motion controls; leg motion commands and head commands in our case. The key advantage of the factorization (11) is that it allows us to estimate the lighting conditions θ_k conditioned on the camera locations $x_{1:k}$. Knowing the camera locations enables us to extract the color mean vectors $\bar{\iota}_{1:k}$ from the camera images, which are sufficient statistics for estimating lighting conditions.

RBPFs estimate the factorized posterior by sampling the rightmost term in (11) using a particle filter and then solving the lighting condition estimation using Kalman filters conditioned on the samples. More specifically, RBPFs represent posteriors by sets of weighted samples, or particles:

$$S_k = \{s_k^{(i)}, w_k^{(i)} \mid 1 \le i \le N\}.$$

In our case, each particle $s_k^{(i)} = \langle \theta_k^{(i)}, x_{1:k}^{(i)} \rangle$, where $\theta_k^{(i)} = \langle \mu_k^{(i)}, \Sigma_k^{(i)} \rangle$ are the mean and covariance of the lighting condition estimate and $x_{1:k}^{(i)}$ are the histories of camera locations.

RBPFs generate samples distributed according to the posterior (11) based on samples drawn from the posterior at time $k-1$, represented by the previous sample set S_{k-1}. RBPFs generate the two components of each particle $s_k^{(i)}$ stepwise by simulating (11) from right to left. In the first step, a sample $s_{k-1}^{(i)} = \langle \theta_{k-1}^{(i)}, x_{1:k-1}^{(i)} \rangle$ is drawn from S_{k-1}. Then, a camera location is drawn conditioned on the sample $s_{k-1}^{(i)}$:

$$x_k^{(i)} \quad \sim \quad p(x_k \mid x_{1:k-1}^{(i)}, z_{1:k}, u_{1:k-1}). \qquad (13)$$

Once the camera location is sampled, it can be used to extract the mean vector $\bar{\iota}_k$ from the current camera image z_k. This mean vector is then used to update the lighting condition estimate (left term in (12)), which is done using the switching Kalman filter discussed in the previous section. It remains to be shown how to generate samples according to (13).

B. Sampling camera locations

To sample from (13), we first transform it as follows.

$$\begin{aligned} & p(x_k \mid x_{1:k-1}^{(i)}, z_{1:k}, u_{1:k-1}) \\ & \qquad = \quad p(x_k \mid x_{k-1}^{(i)}, \theta_{k-1}^{(i)}, z_k, u_{k-1}) \qquad (14) \\ & \qquad \propto \quad p(z_k \mid x_k, \theta_{k-1}^{(i)})\; p(x_k \mid x_{k-1}^{(i)}, u_{k-1}) \qquad (15) \end{aligned}$$

(14) follows from the standard Markov assumption in robot localization that the current location x_k is independent of older information given the previous location [6], and the fact that $\theta_{k-1}^{(i)}$ is a sufficient statistic for the lighting condition up to time $k-1$. (15) follows by Bayes rule and the fact that an observation only depends on the current camera location and the lighting conditions, and that the predicted camera location is independent of the lighting condition. To generate particles according to (15), we apply the standard particle filter update routine. More specifically, we predict the camera location $x_k^{(i)}$ based on the previous location $x_{k-1}^{(i)}$ along with the most recent control information u_{k-1} and the motion model $p(x_k|x_{k-1}^{(i)}, u_{k-1})$. The motion model corresponds to a noisy version of the control commands issued by the robot, including both body motion and head motion. This sampling step gives the extended trajectory $x_{1:k}^{(i)}$. The importance weight of this particle is given by the likelihood $p(z_k \mid x_k^{(i)}, \theta_{k-1}^{(i)})$ of the most recent camera image. In standard RBPF, this likelihood is given by the innovation of the Kalman filter update of the left term in (12). However, since the lighting condition model is based on average colors in the image only, it does not provide fine-grained information about the camera location. Instead, we compute the likelihood of a camera image z_k on a pixel by pixel basis:

$$p(z_k \mid x_k^{(i)}, \theta_{k-1}^{(i)}) \quad = \quad \prod_{c \in z_k} p(c \mid x_k^{(i)}, \theta_{k-1}^{(i)}), \qquad (16)$$

where the c's are the raw YUV pixel values. The likelihood of a pixel value is computed by extracting the expected map color m from the 3d map of the environment using the camera location $x_k^{(i)}$ (background pixels are not considered). To get an estimate of the lighting conditions at time k, we predict $\bar{\theta}_k^{(i)} = \langle \bar{\mu}_k^{(i)}, \bar{\Sigma}_k^{(i)} \rangle$ using a Kalman prediction step on the previous lighting condition $\theta_{k-1}^{(i)}$. Given the map color m and the predicted estimate of the lighting condition $\bar{\theta}_k^{(i)}$, the likelihood of a raw YUV pixel value c is then given by the likelihood model (9).

[2] Camera locations x_k are given in six degree-of-freedom coordinates. Estimating these coordinates consists of two sub-problems: The first is to estimate the location of the robot and the second is to estimate the camera location relative to the robot's body.

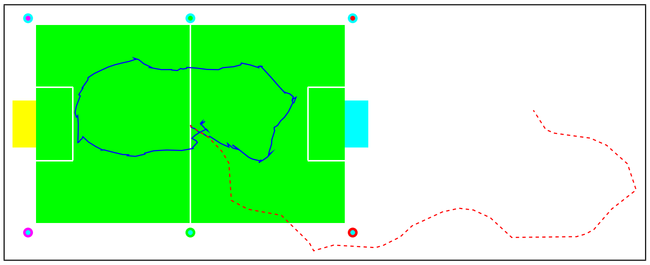

Fig. 3. Path of the robot during one experiment. (*solid*) path computed by our localization approach, (*dashed*) path computed from odometry information only.

To summarize, the RBPF algorithm works as follows. Each sample of the particle filter contains a camera location $x_{k-1}^{(i)}$ along with a Kalman filter $\theta_{k-1}^{(i)}$ representing an estimate of the lighting condition (it is not necessary to keep the complete history of camera locations). At each iteration, a sample $s_{k-1}^{(i)}$ is drawn from the previous sample set. Then, the next camera location $x_k^{(i)}$ is sampled according to the motion model (rightmost term in (15)). Next, the sample is weighted by the likelihood of the camera image given the lighting condition estimate $\bar{\theta}_k^{(i)}$ and the expected image extracted from the location $x_k^{(i)}$ and the 3d map of the environment. Finally, the switching Kalman filter for the lighting condition is updated with the mean color vector $\bar{\iota}_k$ generated using $x_k^{(i)}$. By coupling camera locations and lighting conditions, this sampling approach favors particles with correct estimates of both camera locations and lighting conditions.

V. Experiments

To evaluate our technique we performed several experiments on an AIBO league robot soccer field (see Fig. 1). The experiments are based on data logs recorded with an AIBO robot. For all experiments, the prior over lighting conditions was trained based on 44 sets of hand labeled images taken from different soccer field setups during past RoboCup tournaments and in our lab. We learned two priors for the jump Markov approach. The data sets were clustered into two groups using k-means clustering on the Y (intensity) channels of the colors. Fig. 5 (a) and (b) show a projection of the individual color Gaussians of the dark and bright prior models onto the UV plane. Each ellipse indicates the 1σ Mahalanobis distance for one of the Gaussians.

Our current implementation of the Rao-Blackwellised particle takes approximately 0.8 seconds per frame on a 1.2 GHz PC, using 100 particles. Even though this is not efficient enough to run on-board the AIBO robot in a competition setting, we expect that a speed up by a factor of 10 can easily be achieved by an improved implementation and by using better proposal distributions for the particle filter.

A. Estimating uniform lighting conditions

In the first experiment we navigated the robot across the field, taking several images of each goal and landmark (see Fig. 3). We recorded two data sets in different lighting conditions: the normal everyday lab illumination and a considerably darker illumination generated by turning off all but one light. Our Rao-Blackwellised particle filter was able to localize the robot and to estimate the lighting conditions in both cases. The overall rate of correctly classified pixels was 0.95±0.02 for the dark and 0.99±0.01 for the bright environment. The first two rows of Table 1 show the mean classification rates, *i.e.* the average percentage of correctly classified pixels, and their 95% confidence intervals for individual colors. The ground truth was generated by manually labeling 12 images of each script.

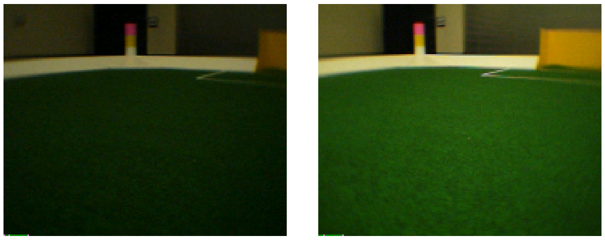

Fig. 4. Example image for the dark and the bright lighting condition used during the experiments.

Lighting	*pink*	*cyan*	*yellow*	*white*
D1 post.	0.78 ± 0.17	0.93 ± 0.05	0.99 ± 0.01	0.99 ± 0.01
D2 post.	0.72 ± 0.05	0.94 ± 0.03	0.97 ± 0.03	0.98 ± 0.01
D1 ind.	0.13 ± 0.17	0.01 ± 0.001	0.77 ± 0.20	0.29 ± 0.17
D1 joint	0.70 ± 0.18	0.82 ± 0.12	0.99 ± 0.01	0.99 ± 0.01
D2 ind.	0.48 ± 0.07	0.30 ± 0.10	0.59 ± 0.13	0.87 ± 0.08
D2 joint	0.66 ± 0.06	0.94 ± 0.02	0.97 ± 0.03	0.98 ± 0.01

Table 1: Avg. classification rates by color. Data set D1 was collected in dark lighting condition, set D2 in bright condition. Top two rows: Results achieved by our approach. Bottom rows: Comparison of classification rates on unobserved colors between an *ind*ependent Gaussian and our *joint* Gaussian model.

The two-level Gaussian color model takes dependencies between colors into account. This enables us to estimate the appearance of colors which have not yet been observed based on the colors observed so far. Therefore, the model can achieve better classification results on colors which are observed for the first time than a model that estimates colors independently, such as [3]. To demonstrate this effect, we compared our model to an independent Gaussian model constructed by setting all correlations between colors in the prior to zero. We run the two algorithms on the data logs several times. On each run we disabled the integration of observations for one of the colors. The bottom four rows of Table 1 summarize the classification results achieved with the two models on the unobserved color. In all cases, the color classification rate based on predictions by our joint Gaussian model is significantly better than the classification obtained with the independent Gaussian model. This result shows that taking dependencies between colors into account indeed improves the classification.

B. Switching lighting conditions

The second experiment demonstrates the benefit of a jump Markov model to quickly adapt to sudden changes in the lighting condition. In this experiment we changed the brightness several times by turning on and off some of the lights while moving the robot across the field. Fig. 4 illustrates the “dark” and “bright” lighting conditions in this experiment. We then compared the classification rate achieved using localization with the jump Markov model to the classification rate achieved with a single Kalman filter. For the jump

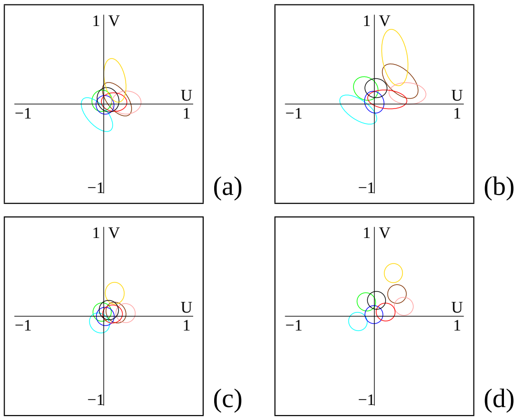

Fig. 5. Covariances of the different colors in the U-V plane: (a) prior for dark lighting conditions; (b) prior for bright lighting conditions; (c) and (d) posteriors after integrating several bright and the first dark image.

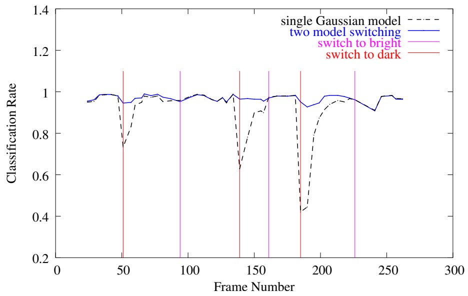

Fig. 6. Color classification rate under two lighting conditions. The classification rate of the single Gaussian model drops considerably whenever lights are switched off.

Markov model we used the dark and bright Gaussian prior models depicted in Fig. 5 (a) and (b). The jump Markov approach automatically selects the model which best matches the lighting condition in the current image and only this model is updated. The effect is visible in Fig. 5 (c) and (d). These images show the posterior UV covariances of the models after the first dark image has been integrated. The dark model has been updated only once and it is therefore still very similar to the initial prior, while the bright model is already adapted to the pertinent bright lighting conditions. Fig. 6 compares the overall color classification achieved with the jump Markov approach to the single Gaussian case. Model switching obtains high color classification rates throughout the experiment, while the classification rate of the single Gaussian model drops substantially whenever lights are switched off.

VI. Conclusions and Future Work

We proposed a Bayesian approach to maintain a two-level Gaussian color model for color classification. The top level of the model represents lighting conditions by a joint Gaussian distribution over color means, while the lower level represents individual colors by independent Gaussians in YUV space. Priors over lighting conditions can be extracted from data collected in different environments and we showed that the model can then be efficiently updated using standard Kalman filter techniques. To do so, the adaptive color model is integrated into a Rao-Blackwellised particle filter for vision-based robot localization where camera positions are sampled and a color model is maintained for each sample. By conditioning on sampled camera trajectories we are able to estimate the lighting condition as if the camera locations were known and we can use a 3d map of the environment to determine the color labels of the image pixels to maintain the color model.

A key advantage of the joint Gaussian representation of lighting conditions is that it takes dependencies between colors into account. This enables us to adapt unobserved colors conditioned on the observed ones. Our experiments show that this leads to substantially improved color classification results, when compared to a model that updates colors independently. However, Gaussians only represent linear dependencies, and the appearance of the colors can change non-linearly, for example, if light sources are changed or major changes in the brightness level occur. To deal with such situations, we maintain a set of models for qualitatively different lighting conditions and use a state switching Kalman filter to automatically adapt to major changes in lighting conditions. We showed in experiments that this approach can efficiently handle even drastic changes in brightness levels.

We demonstrated the advantages of our technique in the context of pixel-based matching between the measured color values and the colors expected from a 3d model of the environment. However, the technique is applicable to a large set of vision algorithms. Currently we are working on the integration into a much faster feature-based localization approach, such as [8]. Additionally, we experiment with compiling the Gaussian models into efficient lookup tables for the color classification. The lookup tables are replaced whenever significant changes in the lighting condition are detected.

Acknowledgments: We would like to thank the GermanTeam and the CM-Pack team for providing the training images for computing the prior models. This research is funded in part by the NSF under grant number IIS-0093406 and by DARPA's SDR Programme (grant number NBCHC020073).

References

[1] C. Andrieu, M. Davy, and A. Doucet. Efficient particle filtering for jump markov systems. In *IEEE International Conference on Acoustics Speech and Signal Processing*, 2002.

[2] Y. Bar-Shalom, X.-R. Li, and T. Kirubarajan. *Estimation with Applications to Tracking and Navigation*. John Wiley, 2001.

[3] D. Cameron and N. Barnes. Knowledge-based autonomous dynamic colour calibration. In *Proc.of RoboCup International Symposium*, 2003.

[4] A. Doucet, J.F.G. de Freitas, K. Murphy, and S. Russell. Rao-Blackwellised particle filtering for dynamic bayesian networks. In *Proc. of the Conference on Uncertainty in Artificial Intelligence*, 2000.

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

[6] D. Fox. Adapting the sample size in particle filters through KLD-sampling. *International Journal of Robotics Research (IJRR)*, 22(12), 2003.

[7] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin. *Bayesian Data Analysis*. Chapman and Hall/CRC, 2nd edition, 2003.

[8] R. Hanek, T. Schmitt, S. Buck, and M. Beetz. Towards robocup without color labeling. In *Proc. of RoboCup International Symposium*, 2002.

[9] M. Jüngel, J. Hoffmann, and M. Lötzsch. A real-time auto-adjusting vision system for robotic soccer. In *Proc. of RoboCup International Symposium*, 2003.

[10] Charles Rosenberg, Thomas Minka, and Alok Ladsariya. Bayesian color constancy with non-gaussian models. In *Advances in Neural Information Processing Systems (NIPS)*, 2003.

[11] C. Stauffer and W. E. L. Grimson. Adaptive background mixture models for real-time tracking. In *Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1999.