

# Autonomous Color Learning in an Artificial Environment\*

Dave van Soest      Mark de Greef      Jürgen Sturm      Arnoud Visser

*Intelligent Autonomous Systems, Universiteit van Amsterdam*

## 1 Context

In our application, the RoboCup soccer competition, we are interested whether certain objects (ball, beacons, goals) are present in the field of view. RoboCup soccer is a color-coded environment, where highly saturated colors are used to ease the recognition of those objects essential for RoboCup soccer. So the color of the object, as classified by a human, is the natural feature space to be used as the first step of processing the stream of images.

Humans are able to classify the field as green in a color image under a wide variety of lighting conditions, and a robot should be able to learn the same mapping from the 3-dimensional color space to a few color classes. The robots in the 4-Legged League record images in the 3-dimensional YUV color space. A standard approach in color-invariant image processing is to convert such images to a more robust color space (see for an overview [2]). Unfortunately such transformations - which are for instance based on Gaussian convolution - are out of scope for our robotic platform. We used another machine learning technique.

To find the mapping from the 3-dimensional color space to the color classes, an analysis of the feature space is needed. Objects which humans would classify as the same color should be visible as clouds of 'similar' pixels. It is the task of the algorithm to separate those clouds into clusters. While there is a wide variety clustering techniques available (see for an overview [3]), complex algorithms are out of scope due to the limited computational resources of the RoboCup robots.

## 2 Clustering methodology

Partitional clustering algorithms are attractive for applications where computational costs are an issue. Fortunately the number of color classes at the RoboCup field is a priori knowledge, which allows us to use a mixture density model for the clusters, without the need to resolve the issue how many clusters are present in the data. The data only has to be 'fitted' against the free parameters of the density distributions in the mixture model. For the density distribution we have chosen a Mixture of Gaussian (MoG), which means that the mean  $\vec{\mu}_s$  and the covariance matrix  $\Sigma_s$  for each of the clusters  $s$  have to be estimated, together with  $w_s$  which represent the weighting factors of the clusters. The free parameters  $\vec{\mu}_s$ ,  $\Sigma_s$  and  $w_s$  are optimized in the maximization step of an expectation-maximization (EM) algorithm, while the training data is redistributed over the clusters in the expectation step.

## 3 Color class assignment

Once the clusters of the typical colors of the objects are found, it seems too trivial to assign the correct label to each of them. Yet, when different lighting conditions are encountered it is clear that the clusters do not have a fixed size or location. If the light changes then the color clusters will shift. The only constant is the relative position of the clusters with respect to each other. The relative positions are maintained for a wide variety of circumstances, although it is possible to change the lighting conditions so drastic that even this feature is no longer constant.

---

\*More details about the algorithms and results of this study can be found in M. de Greef and D. van Soest, 'Automatic Color Calibration on a Robosoccer-field', June 2006, Technical Report, Universiteit van Amsterdam

In the YUV color space it is hard to define the relations between the color clusters; e.g. these relations are vectors in the 3-dimensional space. It is easier to define the relations in the HSI color space, because the predefined colors of the soccer field (except black and white) are well saturated, and mainly differ on their hue value  $H$ .

The algorithm for our setting is as follows. Here  $\mathcal{CC}$  initially is the set of clusters. One by one clusters are removed this set based on a heuristic rule and added to the set of known color classes  $\mathcal{C}$ . The following rules are applied:

$$C_{white} = \arg \min_{C_{C_s} \in \mathcal{CC}} S_s, \mathcal{CC} := \mathcal{CC} - \{C_{white}\} \quad (1)$$

$$C_{sky-blue} = \arg \max_{C_{C_s} \in \mathcal{CC}} H_s, \mathcal{CC} := \mathcal{CC} - \{C_{sky-blue}\} \quad (2)$$

$$C_{green} = \arg \max_{C_{C_s} \in \mathcal{CC}} H_s, \mathcal{CC} := \mathcal{CC} - \{C_{green}\} \quad (3)$$

$$C_{yellow} = \arg \max_{C_{C_s} \in \mathcal{CC}} H_s, \mathcal{CC} := \mathcal{CC} - \{C_{yellow}\} \quad (4)$$

$$C_{pink} = \arg \max_{C_{C_s} \in \mathcal{CC}} H_s, \mathcal{CC} := \mathcal{CC} - \{C_{pink}\} \quad (5)$$

Adding rules for the other predefined colors is quite trivial. For instance, black can be distinguished from white by its intensity  $I_s$ . The rules are based on min- and max-functions, and not on absolute values. As can be seen by the vertical lines in figure 1, the absolute hue-value of the centers of the clusters  $H_s$  can shift considerably when the lighting conditions are altered. Yet, the order from left to right between the colors is the same for all three conditions. No other assumptions than the heuristic rules above are made, which makes this algorithm quite general applicable. Notice that the clustering process itself is executed in the the YUV space; only the association of the found clusters with the right labels is performed in the HSI space. Therefore, only the cluster centers have to be converted to HSI which is computational no problem.

## 4 Results

Having the method we tested it using different lighting conditions, not only the three lighting conditions given in figure 1. The algorithm proved to be highly robust against lighting conditions. See [1] for more details. Only in two extreme cases the limits of the method were encountered.

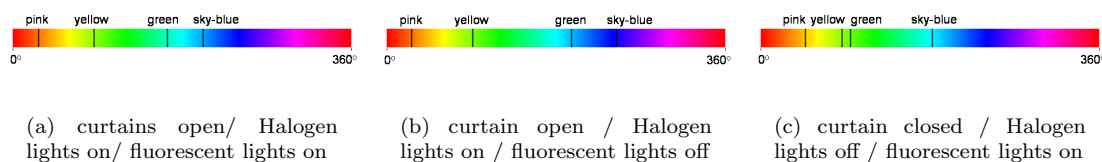


Figure 1: Hue value of the cluster centers  $H_s$  under a variety of lighting conditions

Once the mapping from the 3-dimensional color space to the color classes is learned, this color map can be stored in a lookup-table. The lookup-table is then used during soccer play to segment the images along scanlines and detect useful objects in an efficient way (e.g. the beacons, the goal).

## References

- [1] M. de Greef and D.A. van Soest. Automatic color calibration on a robosoccer-field. Technical report, Universiteit van Amsterdam, June 2006.
- [2] J.-M. Geusebroek. *Color and Geometrical Structure in Images*. PhD thesis, Universiteit van Amsterdam, November 2000.
- [3] J.J. Verbeek. *Mixture models for clustering and dimension reduction*. PhD thesis, Universiteit van Amsterdam, December 2004.