Towards RoboCup without Color Labeling

Robert Hanek, Thorsten Schmitt, Sebastian Buck, Michael Beetz

Munich University of Technology
Department of Computer Science IX
Orleanstr. 34, D-81667 Munich, Germany
http://www9.in.tum.de/agilo/

Abstract. *Object recognition and localization methods in RoboCup work on color segmented camera images. Unfortunately, color labeling can be applied to object recognition tasks only in very restricted environments, where different kinds of objects have different colors. To overcome these limitations we propose an algorithm named the Contracting Curve Density (CCD) algorithm for fitting parametric curves to image data. The method neither assumes object specific color distributions, nor specific edge profiles, nor does it need threshold parameters. Hence, no training phase is needed. In order to separate adjacent regions we use local criteria which are based on local image statistics. We apply the method to the problem of localizing the ball and show that the CCD algorithm reliably localizes the ball even in the presence of heavily changing illumination, strong clutter, specularity, partial occlusion, and texture.*

1 Introduction

Currently in all three real robot soccer leagues (small-size, middle-size, and Sony four-legged league) the task of visual perception is considerably simplified by two restrictions: 1.) all objects on the pitch have a distinct color, 2.) the illumination is constant and roughly homogeneous. Due to this restrictions classes of objects (e.g. robots, ball, lines, color markers, goals) can roughly be identified by their color. To the best of our knowledge, all robot soccer teams using cameras apply a color labeling step to the sensed image data, e.g. [29, 32, 15, 17, 6, 3]. In this labeling or classification step the color value of a pixel is mapped to one of the distinct colors. Labeling provides the advantage of a fast and substantial reduction of the image data while maintaining the major part of the relevant information. However even in the restricted RoboCup scenario for all color spaces the color histograms, i.e. probability density functions (pdf), of different object classes do overlap. Furthermore the pdf of an object class varies spatially. This causes an uncertainty in the labeling process. Obviously in a less restricted environment the uncertainty would be much higher.

To avoid this difficulty we use raw image data (in RGB space). Our method refines iteratively a vague interpretation of the scene by alternating two steps[1]:

---

[1] An initial vague interpretation may be obtained for example by a Monte Carlo method or by prediction over time.

Fig. 1. The CCD algorithm is able to localize the ball in natural environments (red: initial ball contour, blue: estimated ball contour).

1.) based on a vague interpretation (e.g. the ball has the 3-D position X) local pdfs are learned from the image data. These pdfs describe the local color distributions of different objects. 2.) in the second step the local pdfs are used in order to refine the interpretation of the scene. The method naturally takes the uncertainty into account caused by the overlapping pdfs. Model knowledge such as the shape of the ball is used in order to reduce the uncertainty. Pixels are not interpreted independently but based on the context and the model knowledge.

The CCD algorithm fits parametric curve models, also known as active contours, deformable models, or snakes [5, 18] into the image. Problems such as the self-localization / pose estimation problem can be addressed by curve fitting [12]. Here the position of the ball with respect to the observing robot is estimated. With our labeling-based approach [12] we experienced substantial inaccuracies under some circumstances, especially if temporal derivatives, e.g. the ball speed, are computed. The CCD algorithm achieves a substantially higher accuracy. Even more important, we think that this work could be an important contribution towards the goal of playing robot soccer in natural environments, see Fig. 1.

The reminder of this paper is organized as follows: in section 2 the contour of the ball is modeled as a function of the ball position. In section 3 an overview of the Contracting Curve Density (CCD) algorithm is given. Sections 4 describe the two main steps of the CCD algorithm. Section 5 contains an experimental evaluation. In section 6 the body of related work is briefly summarized and finally section 7 concludes the paper.
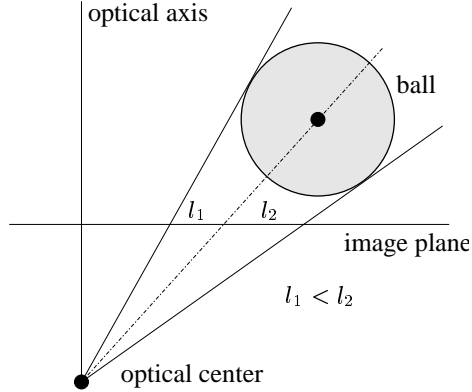
Fig. 2. Projection of the ball into the image plane: the projection of the ball's center point does not yield the center of the ball contour

2   Modeling the Ball Contour

In this section the contour of the ball is described (modeled) as a function of the ball position. In this modeling process knowledge of the ball (the object of interest) and the imaging device is incorporated. Here the ball is modeled as a sphere with known radius. We use a camera with known internal and external camera parameters. The method proposed here is developed for our non-omnidirectional vision system. However, the method can easily be adapted to omnidirectional vision systems which are quite popular in robotic soccer, e.g. [14, 24, 25].

We denote the center point of the ball in the coordinate system of the observing robot by $\mathbf{M}_r$. We distinguish two cases: 1.) in the first case the ball is assumed to lay on the floor. Hence the center point $\mathbf{M}_r = (x, y, r)^T$ of the ball has two degrees of freedom, namely x and y. The z-coordinate is given by the radius r of the ball. 2.) in the second case we assume that the ball may fly, i.e. not lay on the floor, which sometimes happens in robotic soccer. In this case the ball position has three unknown coordinates: $\mathbf{M}_r = (x, y, z)^T$. While the second case is more general, it requires an optimization for more parameters and tends to be less precise, if the ball is on the floor. By $\mathbf{\Phi}$ we denote the unknown parameters of the ball position $\mathbf{M}_r$ (for the first case $\mathbf{\Phi} = (x, y)^T$, for the second case $\mathbf{\Phi} = (x, y, z)^T$).

In the following the relation between $\mathbf{\Phi}$ and the pixel coordinates of the ball contour is derived. First the center point of the ball is expressed in camera coordinates $\mathbf{M}_c$ by

$$\mathbf{M}_c = \mathbf{R} \cdot (\mathbf{M}_r - \mathbf{t}) \tag{1}$$

where $\mathbf{R}$ and $\mathbf{t}$ specify the orientation and location of the camera with respect to the robot coordinate system. The set of tangent lines to the sphere (the ball) passing through the optical center of the camera define a cone, see Fig. 2. As

known, the intersection of the cone with the image plane yields an ellipse [28]. Note the intersection is a circle only if the center point of the sphere lies on the optical axis of the camera. In all other cases the projection of the center point is not the center point of the ball contour, see Fig. 2. The set of contour points can be described in undistorted image coordinates $\mathbf{u}$ by

$$\mathbf{u}(s) = \mathbf{m}_i + \cos(s) \cdot \mathbf{a}_1 + \sin(s) \cdot \mathbf{a}_2 \qquad (2)$$

where $\mathbf{m}$ is the center, $\mathbf{a}_1$ and $\mathbf{a}_2$ are the two axis of the ellipse in undistorted image coordinates. The angle $s \in [-\pi, ..., \pi[$ specifies a particular point on the ellipse. Our lens causes substantial radial distortions. According to Lenz et al. [19] the distorted coordinates $\mathbf{d}$ can be approximated by

$$\mathbf{d} = (d_x, d_y)^T = 2\mathbf{u}/(1 + \sqrt{1 - 4\kappa|\mathbf{u}|^2}) \qquad (3)$$

where κ is the distortion parameter of the lens. The corresponding pixel coordinates $\mathbf{c}$ can be obtained by

$$\mathbf{c} = (d_x/S_x + C_x, d_y/S_y + C_y)^T \qquad (4)$$

where S_x and S_y define the pixel size and C_x and C_y specify the center point of the camera.

3 Overview of the Contracting Curve Density (CCD) Algorithm

The CCD algorithm fits parametric curve models to image data [11]. The algorithm can roughly be characterized as an extension of the EM algorithm [10] using additional knowledge. The additional knowledge consists of: (i) a curve model, which describes the set of possible boundaries between adjacent regions, and (ii) a model of the image generation process. The CCD algorithm, depicted in Fig. 3, performs an iteration of two steps, which roughly correspond to the two steps of the EM algorithm: **1. Local statistics of image data are learned** from the vicinity of the curve. These statistics locally characterize the two sides of the edge curve. **2.** From these statistics, the **estimation of the model parameters is refined** by optimizing the separation of the two sides. This refinement in turn leads in the next iteration step to an improved statistical characterization of the two sides. During the process, the uncertainty of the model parameters decreases and the probability density of the curve in the image contracts to a single edge estimate. We therefore call the algorithm Contracting Curve Density (CCD) algorithm.

Input: The input of the CCD algorithm consists of the image data $\mathbf{I}^*$ and the curve model. The image data are local features, e.g. RGB values, given for each pixel of the image. The curve model consists of two parts: 1.) a differentiable curve function c describing the model edge curve in the image as a function of the model parameters $\mathbf{\Phi}$, 2.) a Gaussian a priori distribution $p(\mathbf{\Phi}) = p(\mathbf{\Phi} \mid \mathbf{m}_{\mathbf{\Phi}}^*, \mathbf{\Sigma}_{\mathbf{\Phi}}^*)$

Contracting Curve Density (CCD) algorithm

Input: image data $\mathbf{I}^*$, differentiable curve function c, mean $\mathbf{m}_{\boldsymbol{\Phi}}^*$ and covariance $\boldsymbol{\Sigma}_{\boldsymbol{\Phi}}^*$
Output: estimate $\mathbf{m}_{\boldsymbol{\Phi}}$ of model parameters and associated covariance $\boldsymbol{\Sigma}_{\boldsymbol{\Phi}}$

Initialization: mean $\mathbf{m}_{\boldsymbol{\Phi}} = \mathbf{m}_{\boldsymbol{\Phi}}^*$, covariance $\boldsymbol{\Sigma}_{\boldsymbol{\Phi}} = c_1 \cdot \boldsymbol{\Sigma}_{\boldsymbol{\Phi}}^*$
repeat
    1. **learn local statistics** of image data from the vicinity of the curve
        (a) compute pixels v in vicinity $\mathcal{V}$ of the image curve from c, $\mathbf{m}_{\boldsymbol{\Phi}}$ and $\boldsymbol{\Sigma}_{\boldsymbol{\Phi}}$
            $\forall v \in \mathcal{V}$ compute vague assignment $\mathbf{a}_v(\mathbf{m}_{\boldsymbol{\Phi}}, \boldsymbol{\Sigma}_{\boldsymbol{\Phi}})$ to the sides of the curve
        (b) $\forall v \in \mathcal{V}$ compute local statistics $\mathbf{S}_v$ of image data $\mathbf{I}_{\mathcal{V}}^*$
    2. **refine estimation** of model parameters
        (a) update mean $\mathbf{m}_{\boldsymbol{\Phi}}$ by performing one iteration step of MAP estimation:
$$\mathbf{m}_{\boldsymbol{\Phi}} = \arg\min_{\mathbf{m}_{\boldsymbol{\Phi}}} \chi^2(\mathbf{m}_{\boldsymbol{\Phi}}) \qquad \text{with}$$
$$\chi^2(\mathbf{m}_{\boldsymbol{\Phi}}) = -2\ln[p(\mathbf{I}_{\mathcal{V}} = \mathbf{I}_{\mathcal{V}}^* \mid \mathbf{a}_{\mathcal{V}}(\mathbf{m}_{\boldsymbol{\Phi}}, \boldsymbol{\Sigma}_{\boldsymbol{\Phi}}), \mathbf{S}_{\mathcal{V}}) \cdot p(\mathbf{m}_{\boldsymbol{\Phi}} \mid \mathbf{m}_{\boldsymbol{\Phi}}^*, \boldsymbol{\Sigma}_{\boldsymbol{\Phi}}^*)]$$
        (b) updated covariance $\boldsymbol{\Sigma}_{\boldsymbol{\Phi}}$ from Hessian of $\chi^2(\mathbf{m}_{\boldsymbol{\Phi}})$
until changes of $\mathbf{m}_{\boldsymbol{\Phi}}$ and $\boldsymbol{\Sigma}_{\boldsymbol{\Phi}}$ are small enough
Post-processing: estimate covariance $\boldsymbol{\Sigma}_{\boldsymbol{\Phi}}$ from Hessian of $\chi^2(\mathbf{m}_{\boldsymbol{\Phi}})$
return mean $\mathbf{m}_{\boldsymbol{\Phi}}$ and covariance $\boldsymbol{\Sigma}_{\boldsymbol{\Phi}}$

Fig. 3. The CCD algorithm iteratively refines a Gaussian a priori density $p(\boldsymbol{\Phi}) = p(\boldsymbol{\Phi} \mid \mathbf{m}_{\boldsymbol{\Phi}}^*, \boldsymbol{\Sigma}_{\boldsymbol{\Phi}}^*)$ of model parameters to a Gaussian approximation $p(\boldsymbol{\Phi} \mid \mathbf{m}_{\boldsymbol{\Phi}}, \boldsymbol{\Sigma}_{\boldsymbol{\Phi}})$ of the posterior density $p(\boldsymbol{\Phi} \mid \mathbf{I}^*)$.

of the model parameters $\boldsymbol{\Phi}$, defined by the mean $\mathbf{m}_{\boldsymbol{\Phi}}^*$ and the covariance $\boldsymbol{\Sigma}_{\boldsymbol{\Phi}}^*$. (The superscript $*$ indicates input data.)

    Output: The output of the algorithm consists of the estimate $\mathbf{m}_{\boldsymbol{\Phi}}$ of the model parameters $\boldsymbol{\Phi}$ and the covariance $\boldsymbol{\Sigma}_{\boldsymbol{\Phi}}$ describing the uncertainty of the estimate. The estimate $\mathbf{m}_{\boldsymbol{\Phi}}$ and the covariance $\boldsymbol{\Sigma}_{\boldsymbol{\Phi}}$ define a Gaussian approximation $p(\boldsymbol{\Phi} \mid \mathbf{m}_{\boldsymbol{\Phi}}, \boldsymbol{\Sigma}_{\boldsymbol{\Phi}})$ of the posterior density $p(\boldsymbol{\Phi} \mid \mathbf{I}^*)$.

    Initialization: The estimate $\mathbf{m}_{\boldsymbol{\Phi}}$ of the model parameters and the associated covariance $\boldsymbol{\Sigma}_{\boldsymbol{\Phi}}$ are initialized using the mean $\mathbf{m}_{\boldsymbol{\Phi}}^*$ and covariance $\boldsymbol{\Sigma}_{\boldsymbol{\Phi}}^*$ of the a priori distribution. The factor c_1 (e.g. $c_1 = 9$) increases the initial uncertainty and thereby enlarges the capture range of the CCD algorithm.

4 Steps of the CCD Algorithm

The two basic steps of the CCD algorithm, depicted in Fig. 3, are briefly summarized in this section. A more detailed description is given in [11].

4.1 Learn Local Statistics (Step 1)

The Gaussian distribution of model parameters $p(\boldsymbol{\Phi} \mid \mathbf{m}_{\boldsymbol{\Phi}}, \boldsymbol{\Sigma}_{\boldsymbol{\Phi}})$ and the model curve function c define a probability distribution of the edge curve in the image. This curve distribution vaguely assigns each pixel in the vicinity of the surmised curve to one side of the curve. In **step 1a** the set $\mathcal{V}$ of pixels v in the vicinity

iteration 0 (initialization)	iteration 2	iteration 7
a.) $\mathbf{I}_v^*$, image date with superimposed mean curve		
b.) $\mathbf{a}_v$, vague assignment to the ball region		
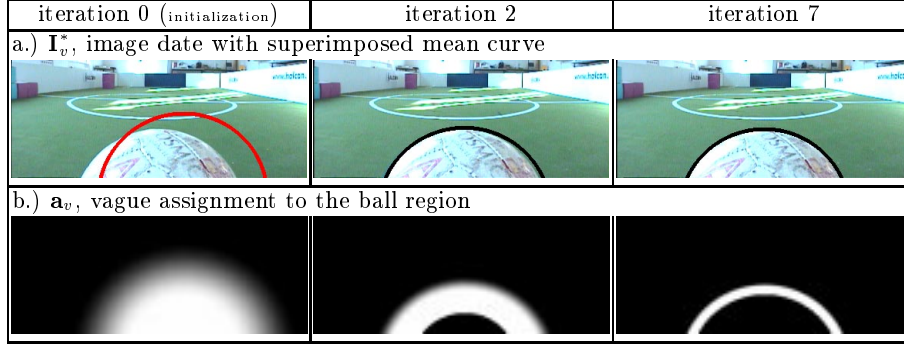
Fig. 4. a.) The initial error is iteratively reduced. **b.)** During the process also the uncertainty of the curve is reduced and the vague side assignments $\mathbf{a}_v$ become certain.

of the surmised curve is determined and for the pixels $v \in \mathcal{V}$ the vague side assignments $\mathbf{a}_v(\mathbf{m_\Phi}, \mathbf{\Sigma_\Phi})$ are computed. The components of the assignments $\mathbf{a}_v$ specify to which extent pixel v is expected to belong to the corresponding side. Fig. 4 row b.) depicts for pixels $v \in \mathcal{V}$ the assignments to the ball region. White pixels indicate a quite certain assignment to the ball region.

In **step 1b** local statistics $\mathbf{S}_v$, i.e. first and second order moments, of the image feature vectors $\mathbf{I}_v^*$ are learned from pixels which are assigned to one side with high certainty. This is done for each of the two sides separated by the curve. In order to obtain the statistics locally adapted windows (weights) are used. The windows are chosen such that the local statistics $\mathbf{S}_v$ can be computed recursively. The resulting time complexity of computing $\mathbf{S}_v$ for all pixels $v \in \mathcal{V}$ is $O(|\mathcal{V}|)$, where $|\mathcal{V}|$ is the number of pixels in the vicinity $\mathcal{V}$. Note that the time complexity is independent of the window size along the curve.

4.2 Refine the Estimation of Model Parameters (Step 2)

In the second step, the estimation of the model parameters is refined based on a MAP optimization. **Step 2a** updates the estimate $\mathbf{m_\Phi}$ such that the vague assignments $\mathbf{a}_v(\mathbf{m_\Phi}, \mathbf{\Sigma_\Phi})$ of the pixels $v \in \mathcal{V}$ fit best to the local statistics $\mathbf{S}_v$. The feature vectors $\mathbf{I}_v^*$ of pixels $v \in \mathcal{V}$ are modeled as Gaussian random variables. The mean vectors and covariances are estimated from the local statistics $\mathbf{S}_v$ obtained from the corresponding side of pixel v. The feature vectors of edge pixels are modeled as weighted linear combinations of both sides of the edge. In step 2a, only one iteration step of the resulting MAP optimization is performed. Since the vague assignments $\mathbf{a}_v(\mathbf{m_\Phi}, \mathbf{\Sigma_\Phi})$ explicitly take the uncertainty (the covariance $\mathbf{\Sigma_\Phi}$) of the estimate into account the capture range is enlarged according to the local uncertainty in the image. This leads to an individually adapted scale selection for each pixel and thereby to a big area of convergence, see [11]. In **step**

Fig. 5. The white non-RoboCup ball is precisely segmented in front of a white-grey background. This is hardly possible with color labeling. (red: initialization, black: estimated ball contour)

2b, the covariance $\mathbf{\Sigma_\Phi}$ of the estimate $\mathbf{m_\Phi}$ is updated based on the Hessian of the resulting χ^2 objective function.

5 Experiments

In our experiments we apply the proposed method to several scenes. Fig. 5 shows a mainly white non-RoboCup ball in front of a white and grey, partially textured background. Despite the lack of color the method precisely segments the ball.

Fig. 6 shows several images where the ball is just partially in the image. But this is enough in order to estimate the ball position. For close up views such as the first three images in Fig. 6 the average error is just a few millimeters, depending on the accuracy of the camera calibration.

In the next experiment we varied the background of the ball and the illumination, see Fig. 7. For the five investigated images the standard deviation of the ball estimates is 0.43 cm which is 0.62% of the estimated distance. Unfortunately, we do not have a sufficiently precise ground truth to compare our results with.

In order to evaluate the performance for a partially occluded ball, we took two images with the same ball position, one with partial occlusion and one without occlusion, see Fig. 8. The ball estimates of the two images differ by 2.0 cm, which is just 2.5% of the ball distance. However, the number of necessary iterations is about 3 times higher in the partially occluded case. The CCD algorithm not only yields an estimate of the ball position, but also a covariance matrix describing the expected uncertainty of the estimate. For the partially occluded ball the expected uncertainty is about three times higher than for the not occluded case. This allows a reasonable data fusion with other sensors data. For a partially occluded ball the area of convergence is significantly reduced, compare Fig. 9 with Fig. 1.
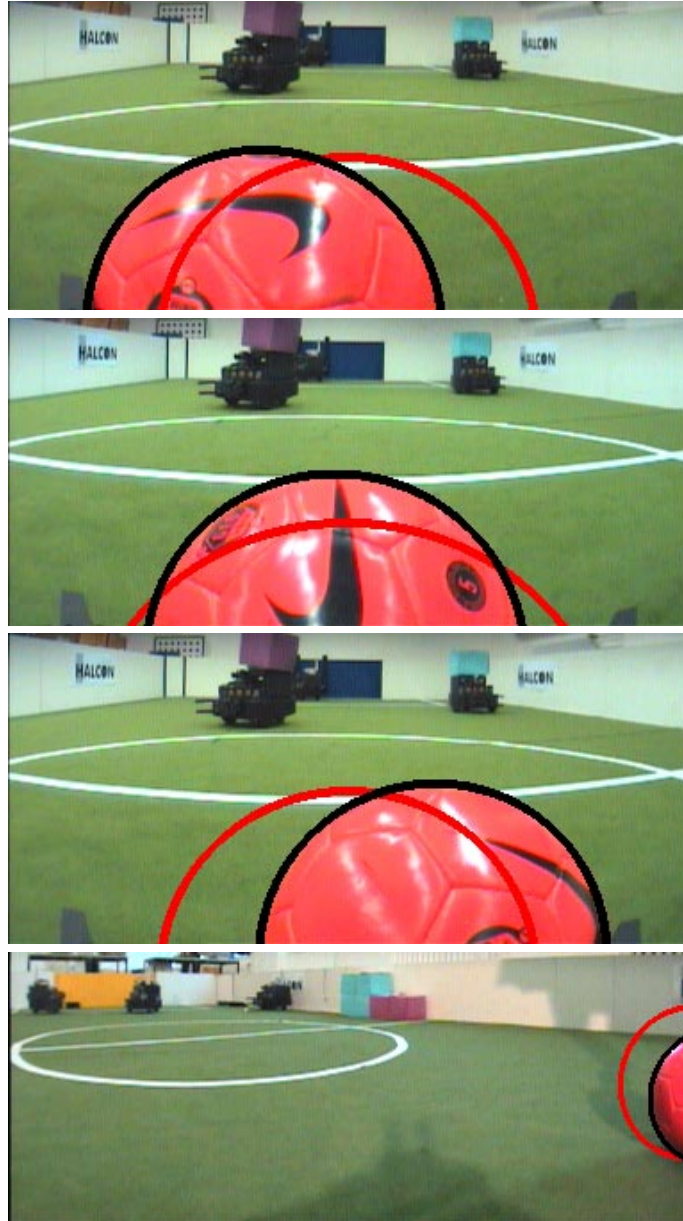
Fig. 6. Refinement of the ball position: initial ball contour (red) and estimated ball contour (black) after 7 iterations. The ball is just partially in the image. Furthermore, the ball has strong specularity and black elements. Nevertheless, the CCD algorithm reliably estimates the ball position.
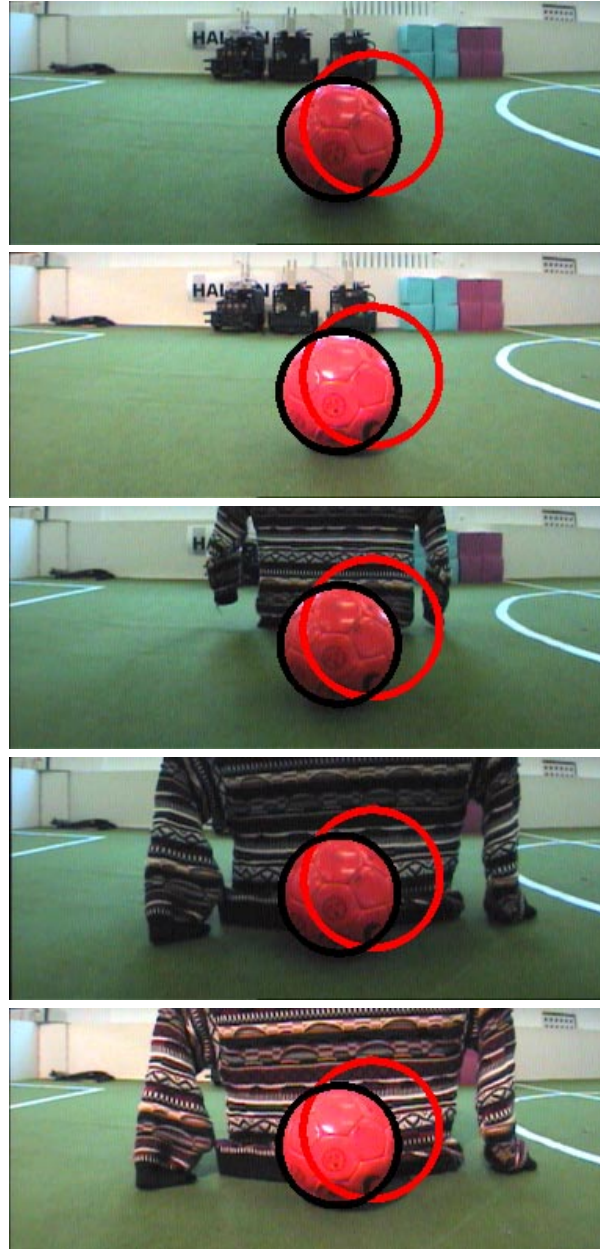
Fig. 7. The strong variations of background and illumination cause just small variations of the ball estimates. For these five images the standard deviation of the ball estimates is 0.43 cm which is 0.62% of the estimated distance.
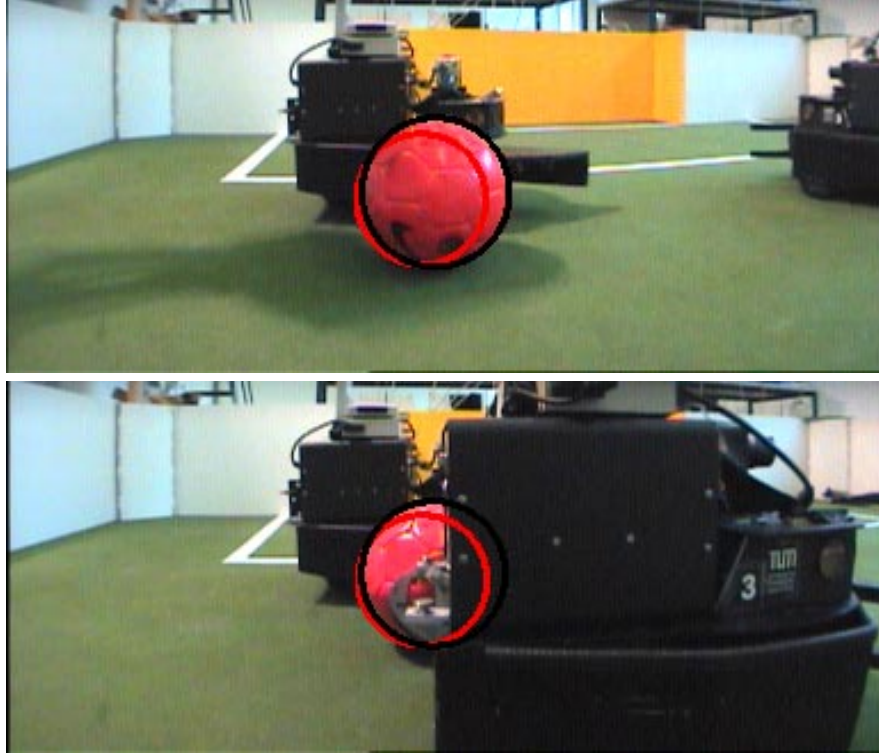
Fig. 8. The same ball position with and without occlusion: the estimates of the ball position differ by 2.0 cm which is just 2.5% of the estimated ball distance.

The next example shows that the CCD algorithm can also be successfully applied to objects of a non-spherical shapes. In Fig. 10 the 3-D pose of a cylindrical mug is estimated by fitting the cylinder contour to the image data. Such a cylindrical model could also be used for the corner posts. Examples for radially distorted straight edges are given in [11]. Straight edges could be used for polyhedral models (e.g. goals).

Finally the accuracy of the CCD algorithm is investigated. Semi-synthetic images with a known ground truth are constructed as follows: from two images one combined image is obtained by taking for one side of the curve the content of image one and for the other side of the curve the content of image two. For pixels on the curve the pixel data are interpolated. In Fig. 11 a circle is fitted to two semi-synthetic images. In both cases the errors for all three quantities (x-coordinate, y-coordinate, radius) are less than 5% of a pixel. In both cases the initial error is reduced by more than 99.8%.
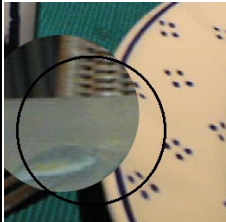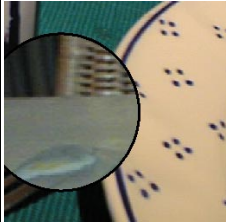
The CCD algorithm as used in this paper is designed to achieve high accuracy. The method is not optimized for speed. Hence, the runtime per image is several

magnification:

convergence to the **right** ball position   convergence to a **wrong** ball position

Fig. 9. Due to the partial occlusion of the ball and the very heterogeneous non-ball region, the area of convergence is strongly reduced. (red: initial ball contour, blue: estimated ball contour).

Fig. 10. An example of a cylindrical mug: the cylinder contour fitted to the image data matches the mug's contour. The estimated parameters of the contour are the five pose parameters of the cylinder. (red: initial contour, black: estimated contour)

errors in pixels (x-coordinate, y-coordinate, radius)			
23.0, 10.0, -5.0	-.025, .028, -.033	35.0, 20.0, -5.5	-.040, -.013, -.044
initialization	estimated contour	initialization	estimated contour

Fig. 11. Despite the inhomogeneity of the foreground and the background in both cases the final error is less than 5% of a pixel for all three coordinates.

seconds or even minutes, depending on the resolution and the initial uncertainty. In order to apply the CCD algorithm to real-time object tracking we propose a fast version of the CCD algorithm [13]. This version does not use all pixels in the vicinity of the curve. Instead it uses just a few carefully selected pixels. This method works within frame rate for simple objects such as a ball.

6   Related Work

In this section we first characterize image segmentation methods applied outside RoboCup. Afterwards we discuss briefly related work applied in RoboCup. The body of work on image segmentation methods (developed outside RoboCup) can be roughly classified into three categories: (i) **edge-based segmentation**, (ii) **region-based segmentation**, and (iii) **methods integrating edge-based and region-based segmentation**.

(i) **Edge-based segmentation** (which is also referred as boundary-based segmentation) relies on discontinuities of image data. Methods for different edge-profiles, i.e. types of discontinuities, exist (e.g. step-edge [2, 26, 7], roof-edge [2, 26], others [2, 26]). The problem of edge-based segmentation is that in practice usually the edge-profile is not known. Furthermore, the profile often varies heavily along the edge caused by e.g. shading and texture. Due to these difficulties usually a simple step-edge is assumed and the edge detection is performed based on a maximum image gradient. However, methods maximizing the image gradient have difficulties to separate regions with internal structure or texture.

(ii) **Region-based segmentation** methods such as [34, 9] rely on the homogeneity of spatially localized features (e.g. RGB values). The underlying homogeneity assumption is that the features of all pixels within one region are statistically independently distributed according to the same probability density function. Contrary to edge-based methods region-based methods do not require an edge-profile. Furthermore, they are able to exploit higher statistical moments of the distributions. Hence, regions which have the same mean feature but different covariances (e.g. caused by texture) can be separated. However often the

underlying assumption of a spatially constant probability density function per region does not hold.

(iii) **Integrating methods:** especially in recent years methods have been published which aim to overcome the individual shortcomings of edge-based and region-based segmentation by integrating both segmentation principles [30, 27, 8, 16]. These methods seek a tradeoff between an edge-based criterion, e.g. the magnitude of the image gradient, and a region-based criterion evaluating the homogeneity of the regions. However, it is questionable whether a tradeoff between the two criteria yields reasonable results when both the homogeneity assumption and the assumption regarding the edge profile do not hold. Due to these difficulties we use local criteria in order to separate adjacent regions. These separation criteria are iteratively obtained from local image statistics.

Model-based methods optimize the fit between the model and the image data. **Global optimization** methods like dynamic programming [1] and Monte Carlo optimization (particle filters, condensation algorithm [5]) are very successfully used (e.g., for tracking). However, dynamic programming requires a discretization of the search space[2], which leads to a limited accuracy, and particle filters show a very slow convergence especially if the sensor noise is low [31].

Local optimization methods may achieve a fast, i.e. quadratic, convergence. Approaches aiming to increase the area of convergence such as [21, 33] are edge-based. For methods maximizing the gradient, the area of convergence depends on the window size used to compute the spatial derivatives. Scale-space theory provides means for automatic scale selection [20]. However, blurring the image data eliminates useful high frequency information. The CCD algorithm does not blur the image data but the curve model. This yields a local and fast optimization with an enlarged area of convergence. Furthermore, high frequency information of the image data is not lost. Several segmentation methods integrate different image cues such as texture and color or brightness [4, 22, 23, 30].

To the best of our knowledge all RoboCup teams use color labeling [29, 32, 15, 17, 6, 3] which belongs to the category of region-based methods. Used color spaces are for example YUV [32, 17, 6, 3] and $\hat{H}$SY [15]. The boundaries between the classes are usually defined as rectangles, boxes, or 'pizza-slices'. Simon et al. [29] propose spatially adapted thresholds. In order to infer the ball position from the labeled image data Jonker et al. [17] use a circular Hough Transform. This method is reported to be robust against partial occlusions. Weigel et al. [32] perform a blob analysis and an inverse projection. Contrary to methods based on color labeling, our method does not need known object specific color distributions. Furthermore, knowledge on the ball contour is explicitly modeled and exploited.

---

[2] Often the contour is approximated by pixel coordinates, i.e. integers. Hence subpixel positions cannot be obtained.

7 Conclusion

We have proposed a novel method, called CCD algorithm, for fitting parametric curve models to image data and we applied this method to the problem of localizing the ball. The CCD algorithm does not depend on prior knowledge of object specific color distributions or properly adjusted threshold parameters. Instead the method starts with a vague object model that can be used to infer the expected object contour. The CCD algorithm alternately performs two steps: local statistics of RGB values are computed describing the two sides of the expected contour. Based on these statistics the model parameters are refined in a MAP step by optimizing the separation of the adjacent regions.

We have shown that the method achieves high robustness and accuracy even in the presence of heavy changes in illumination, strong texture, clutter, and specularity. Knowledge of the object of interest and the imaging sensor is explicitly modeled and exploited. This allows a straightforward adaption to other imaging devices and other problems. Since the CCD algorithm provides a confidence region for the estimates and a likelihood, a fusion with other sources of uncertain information (e.g. multiple observer) can easily be accomplished.

While other methods applied in RoboCup use color information in order to identify objects, in this paper we use shape information. However, vague knowledge on object specific (local or global) color distributions can easily be exploited and subsequently updated by the CCD algorithm. When applied to tracking this could gain additional robustness and accuracy.

References

1. AMINI, A., WEYMOUTH, T., AND JAIN, R. Using dynamic programming for solving variational problems in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence 12*, 9 (September 1990), 855–867.
2. BAKER, S., NAYAR, S., AND MURASE, H. Parametric feature detection. *International Journal of Computer Vision 27*, 1 (March 1998), 27–50.
3. BANDLOW, T., KLUPSCH, M., HANEK, R., AND SCHMITT, T. Fast image segmentation, object recognition and localization in a robocup scenario. In *Third International Workshop on RoboCup (Robot World Cup Soccer Games and Conferences)* (1999), Lecture Notes in Computer Science, Springer-Verlag.
4. BELONGIE, S., CARSON, C., GREENSPAN, H., AND MALIK, J. Color- and texture-based image segmentation using the expectation-maximization algorithm and its application to content-based image retrieval. In *Proc. International Conference on Computer Vision* (1998), pp. 675–682.
5. BLAKE, A., AND ISARD, M. *Active Contours.* Springer-Verlag, Berlin Heidelberg New York, 1998.
6. BRUCE, J., BALCH, T., AND VELOSO, M. Fast and inexpensive color image segmentation for interactive robots. In *International Conference on Intelligent Robots and Systems (IROS)* (2000).

7. CANNY, J. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence 8*, 6 (November 1986), 679–698.

8. CHAKRABORTY, A., AND DUNCAN, J. Game-theoretic integration for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence 21*, 1 (January 1999), 12–30.

9. CHESNAUD, C., REFREGIER, P., AND BOULET, V. Statistical region snake-based segmentation adapted to different physical noise models. *IEEE Transactions on Pattern Analysis and Machine Intelligence 21*, 11 (November 1999), 1145–1157.

10. DEMPSTER, A., LAIRD, N., AND RUBIN, D. Maximum likelihood from incomplete data via the EM algorithm. *J. R. Statist. Soc. B 39* (1977), 1–38.

11. HANEK, R. The Contracting Curve Density Algorithm and its Application to Model-based Image Segmentation. In *Proc. Conf. Computer Vision and Pattern Recognition* (2001), pp. I:797–804.

12. HANEK, R., AND SCHMITT, T. Vision-Based Localization and Data Fusion in a System of Cooperating Mobile Robots. In *Proc. of the IEEE Intl. Conf. on Intelligent Robots and Systems* (2000), IEEE/RSJ, pp. 1199–1204.

13. HANEK, R., SCHMITT, T., BUCK, S., AND BEETZ, M. Fast Image-based Object Localization in Natural Scenes. In *Proc. of the IEEE Intl. Conf. on Intelligent Robots and Systems (submitted)* (2002), IEEE/RSJ.

14. HUNDELSHAUSEN, F., BEHNKE, S., AND ROJAS, R. An omnidirectional vision system that finds and tracks color edges and blobs. In *5th International Workshop on RoboCup (Robot World Cup Soccer Games and Conferences)* (2001), Lecture Notes in Computer Science, Springer-Verlag.

15. JAMZAD, M., SADJAD, B., MIRROKNI, V., KAZEMI, M., CHITSAZ, H., HEYDARNOORI, A., HAJIAGHAI, M., AND CHINIFOROOSHAN, E. A fast vision system for middle size robots in RoboCup. In *5th International Workshop on RoboCup (Robot World Cup Soccer Games and Conferences)* (2001), Lecture Notes in Computer Science, Springer-Verlag.

16. JONES, T., AND METAXAS, D. Image segmentation based on the integration of pixel affinity and deformable models. In *Proc. Conf. Computer Vision and Pattern Recognition* (1998), pp. 330–337.

17. JONKER, P., CAARLS, J., AND BOKHOVE, W. Fast and Accurate Robot Vision for Vision based Motion. In *4th International Workshop on RoboCup (Robot World Cup Soccer Games and Conferences)* (2000), P. Stone, T. Balch, and G. Kraetzschmar, Eds., Lecture Notes in Computer Science, Springer-Verlag, pp. 72–82.

18. KASS, M., WITKIN, A., AND TERZOPOULOS, D. Snakes: Active contour models. *International Journal of Computer Vision 1*, 4 (January 1988), 321–331.

19. LENZ, R., AND TSAI, R. Y. Techniques for Calibration of the Scale Factor and Image Center for High Accuracy 3-D Machine Vision Metrology. *IEEE Trans. on Pattern Analysis and Machine Intelligence 10*, 5 (Sept. 1988), 713–720.

20. LINDEBERG, T. Feature detection with automatic scale selection. *International Journal of Computer Vision 30*, 2 (November 1998), 79–116.

21. LUO, H., LU, Q., ACHARYA, R., AND GABORSKI, R. Robust snake model. In *Proc. Conf. Computer Vision and Pattern Recognition* (2000), pp. I:452–457.

22. MALIK, J., BELONGIE, S., SHI, J., AND LEUNG, T. Textons, contours and regions: Cue integration in image segmentation. In *Proc. International Conference on Computer Vision* (1999), pp. 918–925.

23. MANDUCHI, R. Bayesian fusion of color and texture segmentations. In *Proc. International Conference on Computer Vision* (1999), pp. 956–962.

24. MARQUES, C., AND LIMA, P. Vision-Based Self-Localization for Soccer Robots. In *International Conference on Intelligent Robots and Systems (IROS)* (2000).

25. NAKAMURA, T., EBINA, A., IMAI, M., OGASAWARA, T., AND ISHIGURO, H. Real-time Estimating Spatial Configurations between Multiple Robots by Trinagle and Enumeration Constraints. In *International Conference on Intelligent Robots and Systems (IROS)* (2000).

26. NALWA, V., AND BINFORD, T. On detecting edges. *IEEE Transactions on Pattern Analysis and Machine Intelligence 8*, 6 (November 1986), 699–714.

27. PARAGIOS, N., AND DERICHE, R. Coupled geodesic active regions for image segmentation: A level set approach. In *Proc. European Conference on Computer Vision* (2000), pp. 224–240.

28. SEMPLE, J., AND KNEEBONE, G. *Algebraic projective geometry*. Oxford University Press, 1952.

29. SIMON, M., BEHNKE, S., AND ROJAS, R. Robust real time color tracking. In *4th International Workshop on RoboCup (Robot World Cup Soccer Games and Conferences)* (2000), P. Stone, T. Balch, and G. Kraetzschmar, Eds., Lecture Notes in Computer Science, Springer-Verlag, pp. 239–248.

30. THIRION, B., BASCLE, B., RAMESH, V., AND NAVAB, N. Fusion of color, shading and boundary information for factory pipe segmentation. In *Proc. Conf. Computer Vision and Pattern Recognition* (2000), pp. II:349–356.

31. THRUN, S., FOX, D., AND BURGARD, W. Monte carlo localization with mixture proposal distribution. In *Proc. of the AAAI National Conference on Artificial Intelligence* (2000), pp. 859–865.

32. WEIGEL, T., KLEINER, A., DIESCH, F., DIETL, M., GUTMANN, J.-S., NEBEL, B., STIEGELER, P., AND SZERBAKOWSKI, B. Cs freiburg 2001. In *5th International Workshop on RoboCup (Robot World Cup Soccer Games and Conferences)* (2001), Lecture Notes in Computer Science, Springer-Verlag.

33. XU, C., AND PRINCE, J. Snakes, shapes, and gradient vector flow. *IEEE Transactions on Image Processing 7*, 3 (March 1998), 359–369.

34. ZHU, S., AND YUILLE, A. Region competition: Unifying snakes, region growing, and bayes/mdl for multiband image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence 18*, 9 (September 1996), 884–900.