

The Contracting Curve Density Algorithm: Fitting Parametric Curve Models to Images Using Local Self-adapting Separation Criteria

Robert Hanek (hanek@in.tum.de) and Michael Beetz (beetzm@in.tum.de)

Image Understanding and Knowledge-Based Systems,
Munich University of Technology,
Boltzmannstr. 3, 85748 Garching b. München, Germany

Abstract.

The task of fitting parametric curve models to the boundaries of perceptually meaningful image regions is a key problem in computer vision with numerous applications, such as image segmentation, pose estimation, object tracking, and 3-D reconstruction. In this article, we propose the *Contracting Curve Density (CCD) algorithm* as a solution to the curve-fitting problem.

The CCD algorithm extends the state-of-the-art in two important ways. First, it applies a novel likelihood function for the assessment of a fit between the curve model and the image data. This likelihood function can cope with highly inhomogeneous image regions, because it is formulated in terms of *local image statistics*. The local image statistics are learned on the fly from the vicinity of the expected curve. They provide therefore locally adapted criteria for separating the adjacent image regions. These local criteria replace often used predefined fixed criteria that rely on homogeneous image regions or specific edge properties. The second contribution is the use of *blurred curve models* as efficient means for iteratively optimizing the posterior density over possible model parameters. These blurred curve models enable the algorithm to trade-off two conflicting objectives, namely heaving a large area of convergence and achieving high accuracy.

We apply the CCD algorithm to several challenging image segmentation and 3-D pose estimation problems. Our experiments with RGB images show that the CCD algorithm achieves a high level of robustness and sub-pixel accuracy even in the presence of severe texture, shading, clutter, partial occlusion, and strong changes of illumination.

Keywords: Deformable Models, Optimization, Model-based Image Segmentation, 3-D Pose Estimation, Color, Texture, Image Cue Integration, Automatic Scale Selection, Sub-pixel Accuracy

1. Introduction

A number of important image processing tasks including image segmentation, 3-D pose estimation, object tracking, and object classification can be formulated as variants of curve fitting (Sullivan and Ponce, 1998; Blake and Isard, 1998; Vijayakumar et al., 1998). Curve fitting is the computational problem of finding the parameter values for a given parametric curve model that best match given image data.

In this article, we describe the Contracting Curve Density (CCD) algorithm, a novel algorithm for parametric curve model fitting, and apply it to image segmentation and related problems such as pose estimation. *Image segmentation* is the process of locating regions in



Figure 1. The mug region and background region are inhomogeneous due to texture, clutter, shading, highlight, and spatially varying illumination.

an image that correspond to objects in the scene. *Model-based* image segmentation methods exploit a priori knowledge about the objects and/or the scene by imposing constraints on region boundaries. Model-based image segmentation can be formulated as a variant of curve fitting. First the curve fitting problem is solved and then the image regions are determined based on the fitted curve.

Unfortunately, solving image segmentation and the related curve fitting problem is often very difficult, especially in natural and unconstrained scenes. For example, suppose we want to segment the coffee mug from the background in Figure 1. This segmentation task is difficult to perform because both the mug region as well as the background region are very inhomogeneous. Both regions exhibit large variations of the pixel values (e.g. RGB values) due to clutter, shading, texture, and highlights. Furthermore, the sensed image data depend on physical conditions, such as the illumination or surface properties that are often unknown. As a consequence, algorithms that apply homogeneous segmentation criteria or predefined fitness functions have difficulties in identifying accurate image segmentations. Representative results of such algorithms are visualized in Figure 3.

The Contracting Curve Density (CCD) algorithm, which we propose in this article, considers curve fitting as a probabilistic reasoning task. The CCD algorithm gets as input data an image and a parametric curve model with a rough Gaussian a priori distribution of the model parameters. From this input, the CCD algorithm computes a Gaussian approximation of the posterior distribution. Figure 2 depicts a curve-fitting problem and the corresponding solution obtained by the CCD algorithm. In this example, the mug is modeled as a cylinder of known dimensions. By fitting the cylinder's contour to the image data, the 3-D pose of the mug is estimated with respect to the observing camera.

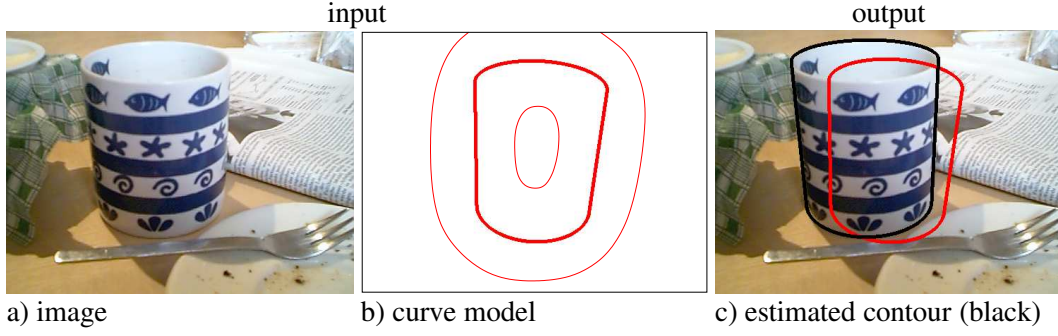


Figure 2. A curve-fitting problem: a) A mug in front of an inhomogeneous background. b) Here, the curve model describes the contour of the mug as the contour of a cylinder. The thick line is the initial contour, i.e. the contour corresponding to the mean of the a priori distribution. The two thin lines illustrate the initial uncertainty of the region boundary. c) The goal is to estimate the model parameters such that the resulting curve (black) fits to the image data and to the a priori distribution. During the fitting process, the uncertainty of the region boundary is widely reduced.

In order to develop algorithms that can robustly fit model curves in natural scenes such as the one depicted in Figure 1, two *key questions* must be addressed. In terms of a probabilistic framework, the key questions can be formulated as follows:

1. How can the likelihood of the model parameters given the image data be obtained or appropriately approximated? The likelihood has to cope with local variations of the image data caused by, e.g., texture or clutter.
2. How can we efficiently find the best parameterization of the model curve, i.e. the parameterization maximizing the posterior density. This is challenging since the posterior density may have multiple local maxima.

In the next section, we classify the body of related work according to several criteria. In particular, we discuss how related methods address the two key questions raised above. Then, in sections 1.2, we explain how we address the two key questions and we describe the main contributions of this article. In section 1.3, an overview of the remainder of this paper is given.

In this paper, by *pixel value* we denote the vector containing the local single- or multi-channel image data associated to a pixel. In our experiments, we directly use the sensed RGB values as pixel values. However, other types of local features computed in a pre-processing step may also be used, e.g. texture descriptors or color values in other color spaces.

1.1. RELATED WORK

Parametric curve models, also known as snakes, active contours, or deformable models (Kass et al., 1988), have been proven as an efficient way to incorporate application-specific a priori knowledge into computer vision algorithms. For example, in order to segment a bone in a

medical image or in order to visually track a person, models describing the possible contours of the objects of interest are used (McInerney and Terzopoulos, 1996; Leventon et al., 2000; Cootes et al., 1993; Blake and Isard, 1998). We classify the body of related work according to several dimensions.

1.1.1. *Classification According to the Segmentation Criteria*

Existing image segmentation methods can be roughly classified according to the segmentation criteria they use. These criteria correspond to assumptions that the methods make about the image data. Methods for curve-fitting employ these criteria in order to derive a fitness or likelihood function of the model parameters. We distinguish four (not disjoint) categories: edge-based, region-based, hybrid, and graph-theoretic segmentation methods.

Edge-based segmentation (also referred to as boundary-based segmentation) methods assume that region boundaries coincide with discontinuities of the pixel value or their spatial derivatives. Different edge-profiles, i.e. types of discontinuities, are used: step-edge (Baker et al., 1998; Nalwa and Binford, 1986; Canny, 1986), roof-edge (Baker et al., 1998; Nalwa and Binford, 1986), and others (Baker et al., 1998; Nalwa and Binford, 1986; Cootes et al., 1994). The problem of edge-based segmentation is that usually the edge-profile is not known in practice. Furthermore, in typical images, the profile often varies heavily along the edge caused by, e.g., shading and texture. Due to these difficulties, often a simple step-edge is assumed and edge detection is performed based on a maximum magnitude of image gradient. In Figure 1, the color values on either side of the mug's contour are not constant even within a small vicinity. Hence, methods maximizing the magnitude of the image gradient have difficulties separating the mug and the background. Figure 3a shows the result of a typical edge-based segmentation method. Three kinds of difficulties can be observed:

1. Edges not corresponding to the object boundary are detected (false positives).
2. Parts of the object boundary are not detected (false negatives).
3. Some detected edges are very inaccurate due to the inhomogeneities.

These problems may be alleviated by performing edge detection without assuming a particular edge profile. Ruzon and Tomasi (2001) maximize a distance measurement, called the earth mover's distance, between the color distributions of two adjacent windows. However, this approach is quite computationally expensive.

Region-based segmentation methods assume that the pixel values within one region satisfy a specific homogeneity criterion. Most of these methods assume that the pixel values of all pixels within one region are statistically independently distributed according to the same probability density function (Zhu and Yuille, 1996; Chesnaud et al., 1999; Mirmehdi and Petrou, 2000). Methods using Markov random fields (MRF) explicitly model the statistical dependency between neighboring pixels which allows for characterizing textures. MRF

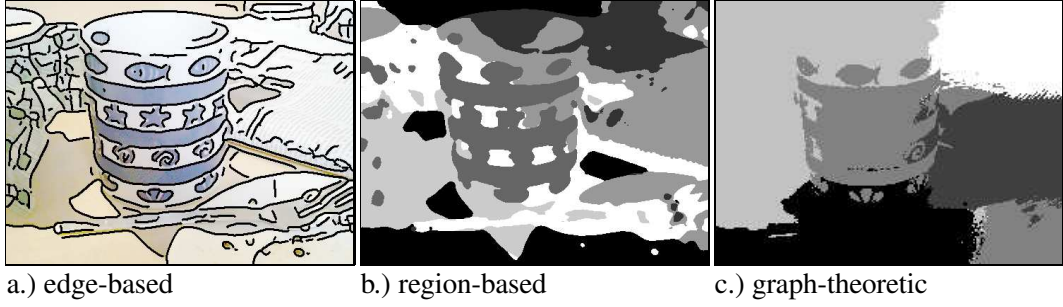


Figure 3. Three categories of image segmentation methods: a) Color edges detected by a gradient-based method (Steger, 2000). b.) Result of a region-based method (Hermes et al., 2002). c.) Result of a graph-theoretic method named *normalized cuts* (Shi and Malik, 2000) - All three approaches fail to accurately separate the mug and the background region. These methods do not exploit (nor require) model knowledge about the shape of the mug.

methods have been proposed for gray value images (Geman and Geman, 1984; Li, 2001) and color images (Bouman and Sauer, 1993; Panjwani and Healey, 1995; Bennett and Khotanzad, 1998). In section 3.2.1, we discuss the similarities and differences between MRF methods and the CCD algorithm proposed in this article.

In contrast 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. For example, regions which have the same mean pixel value but different covariance matrices (e.g. caused by texture) can be separated. However, often the underlying homogeneity assumption does not hold for the whole region. In Figure 1 both the mug region and the background region are inhomogeneous. Hence, the region-based method employed in Figure 3b fails to accurately separate the mug and the background region.

Hybrid Segmentation methods, also known as integrating segmentation methods, try to overcome the individual shortcomings of edge-based and region-based segmentation by integrating both segmentation principles (Thirion et al., 2000; Paragios and Deriche, 2000; Chakraborty and Duncan, 1999; Jones and Metaxas, 1998). Hybrid 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 doubtful 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 as in Figure 1.

Graph-theoretic segmentation methods formulate the problem of image segmentation as a graph-theoretic problem (Shi and Malik, 2000; Felzenszwalb and Huttenlocher, 1998; Cox et al., 1996; Wu and Leahy, 1993). The image is represented by a weighted undirected graph. The pixels are the nodes of the graph and each pair of neighboring nodes is connected by an edge. The edges are labeled by edge weights quantifying the similarity between the connected pixels. Image segmentation is achieved by removing edges, i.e. cutting the graph into disjoint

subgraphs. Suitable cuts are found by minimizing the cut cost or related functions evaluating the weight of the removed edges. Figure 3c depicts the result of the *normalized cuts* algorithm (Shi and Malik, 2000). This method also fails to accurately segment the mug. We suppose that this failure is caused by the predefined similarity function used for computing the weights of the edges. The edge weights only depend on the two pixels connected by the edge. The similarity function does not take the neighborhood of the pixels into account. Hence, the resulting separation criterion does not adapt to the local context of the pixels to be separated.

In order to separate adjacent regions, we do not rely on a predefined fixed edge profile, homogeneity criterion, or similarity measure. Instead, we propose locally adapted separation criteria which are iteratively learned from the vicinity of the curve.

1.1.2. Classification According to the Optimization Method

Methods for curve-fitting can be classified according to the optimization technique. Global and local optimization methods are applied.

Global optimization methods: Several global optimization methods are employed for curve fitting. These methods can be classified in *deterministic* and *stochastic* methods. Deterministic methods include *dynamic programming* and other *shortest path algorithms* (Amini et al., 1990; Geiger et al., 1995; Mortensen and Barrett, 1998; Dubuisson-Jolly and Gupta, 2001) as well as *Hough transform* (Hough, 1962; Ballard, 1981). These methods require a discretization of the search space, which usually leads to a limited accuracy or to a high computational cost, depending on the number of discretization levels.

Stochastic methods are methods such as *Monte Carlo* optimization or *simulated annealing*. The former are also known as particle filters or the *condensation* algorithm (Isard and Blake, 1998; Li et al., 2003). Particle filters have been applied successfully, e.g. for tracking. However, the computational cost of particle filters increases exponentially with respect to the dimension of the search space (the dimension of the parameter vector) (MacCormick and Isard, 2000). Usually, particle filters are fast and accurate only if the search space is of low dimension or sufficiently small. Simulated annealing (Geman and Geman, 1984; Storvik, 1994; Bongiovanni et al., 1995) is generally computationally demanding.

Local optimization methods may achieve a fast, i.e. quadratic, convergence. Approaches aiming to increase the area of convergence such as (Kass et al., 1988; Xu and Prince, 1998; Luo et al., 2000) are edge-based. For methods maximizing the image gradient, the area of convergence depends on the window size used for computing the spatial derivatives. Often a multi-scale description of the image data is used. First the curve model is fitted to a large scale description yielding a smoothed objective function. Then the blurring of the image data, i.e. the window size, is gradually reduced. Scale-space theory provides means for automatic scale selection (Lindeberg, 1998). However, blurring the image data eliminates useful high frequency information.

In this article, we propose a local optimization method based on a 'blurred model'. The method achieves a large area of convergence and a high accuracy with a relatively small number of iterations (see section 4.2).

1.1.3. *Classification According to Alternative Dimensions*

Image segmentation methods can be further classified according to the used image cues. Several methods integrate different image cues such as texture and color or brightness (Belongie et al., 1998; Malik et al., 1999; Manduchi, 1999; Thirion et al., 2000; Panjwani and Healey, 1995; Zhong et al., 2000; Zeoller et al., 2002; Mirmehdi and Petrou, 2000). We use local statistics which jointly characterize texture, color, and brightness.

Segmentation methods using a parametric curve model can be further classified according to the types of possible curve models. Some methods, e.g. (Blake and Isard, 1998), require linear curve models; that is the relation between a point on the curve and the model parameters has to be linear. Our method can cope with the linear and non-linear case.

We further distinguish *feature-extracting* and *non-feature-extracting* methods. Feature-extracting methods first, extract salient image features, e.g. edge points. In this step, the image data is substantially reduced to a relatively small set of image features. Then, they fit (or *match*) the model to the extracted features. If relevant image features can be extracted reliably, this two-step approach can be very efficient (Blake and Isard, 1998; Lowe, 1991; Xu and Prince, 1998; Luo et al., 2000; Terzopoulos and Szeliski, 1992). Non-feature-extracting methods directly fit the model to dense image data rather than to sparse image features (Kass et al., 1988; Ronfard, 1994; Pece and Worrall, 2002; Robert, 1996; Kollnig and Nagel, 1995; Ulrich et al., 2001). Unlike feature-extracting methods, non-feature-extracting methods usually do not make decisions based on local image properties only. Such methods may be applied even if suitable image features cannot be reliably extracted, e.g. in Figure 1. The CCD algorithm proposed here belongs to the class of non-feature-extracting methods.

Several methods use prior knowledge about the pixel values, e.g. intensity or color values (Cootes et al., 1993; Schmitt et al., 2002; Cootes et al., 2001). This has been proven to be very helpful in application where such prior knowledge is given with sufficient accuracy. Approaches for image matting represent this knowledge by local statistics in a manner similar to the CCD algorithm (Ruzon and Tomasi, 2000; Chuang et al., 2001). However, these methods obtain the statistics from *user-defined* parts of the image. Furthermore, these methods do not apply a curve model, and thus optimize independent pixel classifications rather than a vector of model parameters.

The CCD algorithm described in this article has been successfully applied to several challenging problems, such as model based image segmentation (Hanek, 2001) and object localization in natural scenes (Hanek et al., 2002b). Furthermore, Hanek et al. (2002a) have proposed a fast real-time version of the CCD algorithm. This method achieves a high speed-up by exploiting only a small number of pixels lying in the vicinity of the expected curve. Based

on the CCD algorithm, Hanek developed a tracking algorithm called *CCD tracker*. A detailed description of the CCD algorithm and the CCD tracker along with a comprehensive experimental evaluation is given in (Hanek, 2003). The conducted experiments show that the CCD tracker clearly outperforms other state-of-the-art trackers in terms of accuracy, robustness, and runtime.

1.2. MOTIVATION AND MAIN CONTRIBUTIONS

As described in section 1.1.1, methods for image segmentation and curve fitting usually employ predefined and fixed criteria in order to separate adjacent image regions. Figure 3 illustrates that such fixed criteria are often not flexible enough to adapt to local properties. The approach presented here does not rely on predefined fixed separation criteria; instead, it employs *locally adapted criteria*, i.e. criteria that are adapted to the local context. For example, in order to decide in Figure 1 whether a pixel belongs to the mug or to the background region, local classification rules are better suited. Pixels close to the boundary between the mug and the newspaper should be classified by different criteria than those pixels close to the boundary between the mug and the light brown part above the mug. Appropriate local criteria should depend on the local distributions of pixel values corresponding to the two sides of the curve.

Unfortunately, these local distributions are usually not known in advance. However, they can be learned in an iterative curve-fitting process that alternately performs two steps until convergence:

1. **Learn local statistics** of the pixel values from the vicinity of the expected curve. The resulting statistics define for each side of the curve a local expectation of the pixel values.
2. **Refine the estimate of the model parameter vector** by assigning the pixels in the vicinity of the expected curve to the side they fit best, according to the local statistics.

In the first step, the estimate of the model parameter vector is fixed, and based on the fixed model parameters the local statistics are computed. In the second step, the local statistics are fixed and the estimate of the model parameter vector is updated.

This process is illustrated in Figure 4. In this example, a radially distorted straight line¹ is being fit to the image data. In this simple example, the curve has just two parameters, namely the y-coordinates at the left and the right image border.² The input image and the estimated curve are depicted in row a) for different iteration steps. The image data expected according to the learned local statistics are depicted in row b). During the iteration, the estimated curve converges to the actual image curve, and the expected image data (row b) describe the actual vicinity of the image curve (row a) with increasing accuracy. During the fitting process,

¹ The line is straight in 3-D coordinates but due to the radial distortion caused by the lens it appears non-straight in the image.

² In general, the curve model may have an arbitrary number of parameters.

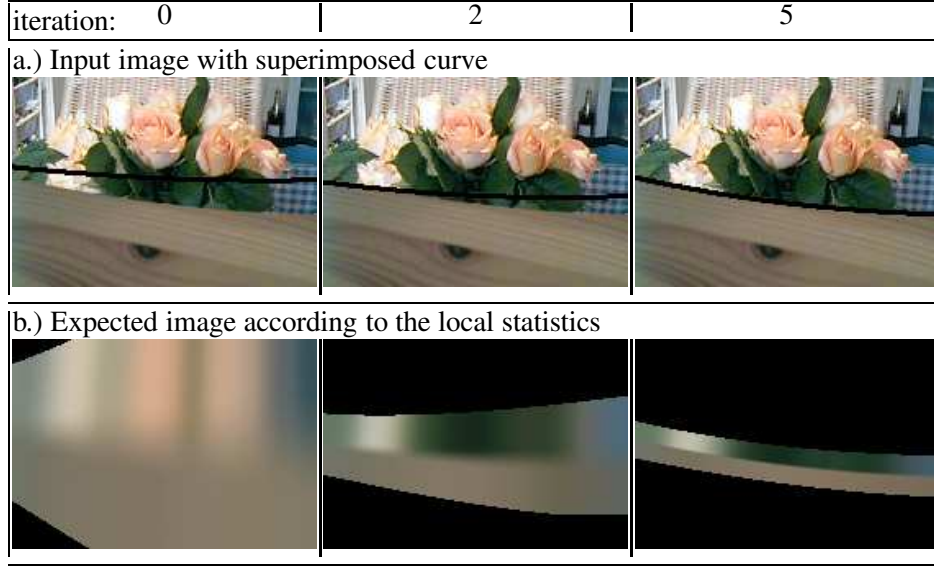


Figure 4. The CCD algorithm converges to the corresponding image curve, despite the heavy clutter and texture next to the initialization. During the iteration, the expected image data (row b) describe the actual vicinity of the image curve (row a) more and more accurately.

the probability density of the curve in the image contracts towards a single curve estimate. Therefore, we call the algorithm Contracting Curve Density (CCD) algorithm.

As main contributions of this article we propose:

1. **A likelihood function that can cope with highly inhomogeneous image regions:** The likelihood function is based on local image statistics, which are iteratively learned from the vicinity of the expected image curve. The local statistics allow for probabilistically separating two adjacent regions even in the presence of spatially changing properties, such as texture, color, shading, or illumination. The resulting locally adapted separation criteria replace predefined fixed criteria (e.g. image gradients or homogeneity criteria). An efficient method is proposed for computing the required local statistics.
2. **A ‘blurred model’ allowing for an enlarged area of convergence and efficient optimization:** In order to increase the capture range, gradient-based methods typically fit the model to a blurred image. We take the opposite approach. We use *non-blurred image data* and a *blurred model*. Instead of optimizing the relation between blurred image data and a single vector of model parameters, we optimize the relation between the non-blurred image data and a Gaussian probability distribution of model parameters. This can be thought of as fitting a blurred model to the image, where the local scale of blurring depends on the local uncertainty of the curve, see Figure 4 row b). The advantages are as follows:

- a) The capture range, i.e. the local scale, is enlarged according to the uncertainty of the model parameters. This yields, for each pixel and each iteration step, an individual compromise between the two conflicting goals, namely a large area of convergence and a high accuracy. Figure 4 row b) shows that the blurring in the direction perpendicular to the curve depends on the iteration and the position along the curve. At the beginning of the iteration, the blurring is stronger at the right side than at the left side.
- b) Optimizing the fit between an image and a blurred model is usually computationally cheaper than blurring the image. Especially if the uncertainty of the initialization is small, just a small fraction of the pixels covered by possible curve points is needed in order to refine the fit.
- c) No high frequency information of the image data is lost. This is particularly important for separating textured regions or for achieving high sub-pixel accuracy.

1.3. OVERVIEW OF THE PAPER

The remainder of this paper is organized as follows. In section 2, an overview of the proposed Contracting Curve Density (CCD) algorithm is given. Sections 3 and 4 describe the two main steps of the CCD algorithm. Section 5 contains an experimental evaluation and finally section 6 concludes the paper.

2. Overview of the Contracting Curve Density (CCD) Algorithm

The CCD algorithm fits a parametric curve model to an image. The algorithm determines the vector of model parameters, that best match the image data and a given a priori distribution of model parameters.

2.1. INPUT AND OUTPUT DATA

The input of the CCD algorithm consists of the image data and the curve model.

Image data: The image, denoted by \mathbf{I}^* , is a matrix of *pixel values*. The superscript $*$ indicates input data. The pixel value \mathbf{I}_p^* is a vector of local single- or multi-channel image data associated with pixel p . In this article, we use raw RGB values. However, other types of local features computed in a pre-processing step may also be used, e.g. color values in a different color space or texture descriptors (Malik et al., 2001).

Curve model: The curve model is composed of two parts:

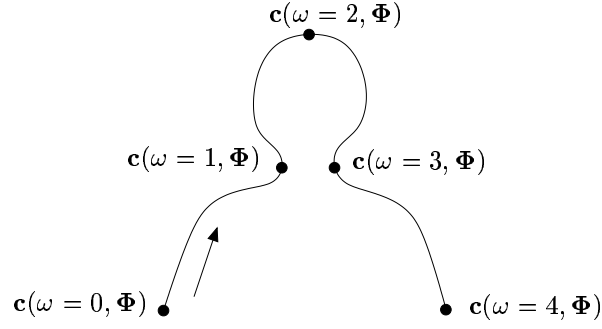


Figure 5. Curve defined by a curve function \mathbf{c} : The vector of model parameters Φ specifies the shape of a particular curve. The scalar ω specifies an individual point on this curve.

1. A *curve function* $\mathbf{c}(\omega, \Phi)$ describing a set of possible model curves in pixel coordinates. The vector of model parameters Φ specifies a particular curve, i.e. a particular shape, of the set of possible curves. The scalar ω specifies an individual point on the curve defined by Φ . The scalar ω monotonously increases as the curve $\mathbf{c}(\cdot, \Phi)$ is traversed, see Figure 5. For example, ω could correspond to the arc length of the curve between a starting point $\mathbf{c}(0, \Phi)$ and the point $\mathbf{c}(\omega, \Phi)$. Such curve functions are frequently used in computer vision (Kass et al., 1988; Blake and Isard, 1998). Models with multiple curve segments, as in Figure 17 on page 32, can be described using multiple curve functions.
2. A Gaussian *a priori distribution* $p(\Phi) = p(\Phi \mid \mathbf{m}_\Phi^*, \Sigma_\Phi^*)$ of the model parameters Φ , defined by the mean vector \mathbf{m}_Φ^* and the covariance matrix Σ_Φ^* . Depending on the application, the quantities \mathbf{m}_Φ^* and Σ_Φ^* may, for example, be given by the user or obtained from a prediction over time. Sclaroff and Liu (2001) obtain object hypotheses from an over-segmented image. Object hypotheses can also be generated by global optimization methods, e.g. Monte Carlo methods (Blake and Isard, 1998) or Hough transform (Hough, 1962).

The output of the algorithm consists of the maximum a posteriori (MAP) estimate \mathbf{m}_Φ of the model parameter vector Φ and the corresponding covariance matrix Σ_Φ . The estimate \mathbf{m}_Φ specifies the best fit of the curve to the image data \mathbf{I}^* and the a priori distribution $p(\Phi)$. The covariance matrix Σ_Φ defines the expected uncertainty of the estimate. The estimate \mathbf{m}_Φ and the covariance Σ_Φ describe a Gaussian approximation $p(\Phi \mid \mathbf{m}_\Phi, \Sigma_\Phi)$ of the posterior density $p(\Phi \mid \mathbf{I}^*)$.

2.2. STEPS OF THE CCD ALGORITHM

The CCD algorithm represents its belief of the model parameter vector by a Gaussian distribution. We denote the mean vector, the current estimate of the model parameters, by \mathbf{m}_Φ , and the corresponding covariance matrix by Σ_Φ . The quantities \mathbf{m}_Φ and Σ_Φ are initialized using

the mean \mathbf{m}_{Φ}^* and covariance Σ_{Φ}^* of the a priori distribution:

$$\mathbf{m}_{\Phi} = \mathbf{m}_{\Phi}^* \quad (1)$$

$$\Sigma_{\Phi} = c_1 \cdot \Sigma_{\Phi}^* \quad (2)$$

Here the scaling factor $c_1 > 1$ can be used to increase the initial uncertainty of the curve and thereby further enlarges the capture range of the CCD algorithm. For the moment, assume that $c_1 = 1$.

The CCD algorithm refines its belief of the model parameters, i.e. the mean vector \mathbf{m}_{Φ} and the covariance matrix Σ_{Φ} , by iterating two steps as shown in Figure 6. In the first step, the mean vector \mathbf{m}_{Φ} and the covariance matrix Σ_{Φ} are fixed. Based on the fixed \mathbf{m}_{Φ} and Σ_{Φ} , local statistics of the pixel values are learned from the vicinity of the expected curve. In the second step, the local statistics are fixed. The mean vector \mathbf{m}_{Φ} and the covariance matrix Σ_{Φ} are updated by assigning the pixels in the vicinity of the curve to the side they fit best according to the local statistics.

At the beginning, usually the uncertainty about the location and shape of the curve is high. Due to this high uncertainty, the windows used for computing the local statistics are relatively large and not close to the actual image curve. Hence, the resulting statistics describe the vicinity of the image curve only roughly, see Figure 4 on page 9. As a consequence, after the first iteration step, the uncertainty is only partially reduced.

Due to the partial reduction of uncertainty, in the next iteration step, the windows used for computing the local statistics are less wide and closer to the actual image curve. The resulting statistics thus describe the vicinity of the curve better than the previous statistics. This yields, in turn, a better estimate of the curve parameters and a further reduction of the uncertainty. The CCD algorithm iterates these two steps until convergence, i.e. until the changes of the estimate \mathbf{m}_{Φ} and the covariance matrix Σ_{Φ} are below some given thresholds. During the iterations, the probability density of the curve in the image contracts towards a single curve estimate. Therefore, we call the algorithm *Contracting Curve Density (CCD)* algorithm. Figure 7 depicts the process for two iteration steps.

Without making any assumptions about the image data \mathbf{I}^* and the curve functions \mathbf{c} , it is difficult to prove convergence or to derive the speed of convergence. However, our experiments with challenging image data show that the area of convergence is quite large and that already a small number of iterations, i.e. 5 to 20, is sufficient in order to reduce the initial uncertainty by more than 99%.

The CCD algorithm has some interesting similarities to the Expectation-Maximization (EM) algorithm (Dempster et al., 1977), which is often used for clustering-based image segmentation, a subset of region-based image segmentation (Hermes et al., 2002; Belongie et al., 1998). The first step computes local statistics defining an expectation of the pixel values (E-step). The second step maximizes this expectation (M-step). The CCD algorithm

Outline of the Contracting Curve Density (CCD) algorithm

Input: image data \mathbf{I}^* , curve function \mathbf{c} , mean vector \mathbf{m}_{Φ}^* , and covariance matrix Σ_{Φ}^*

Output: estimate \mathbf{m}_{Φ} of the model parameter vector and associated covariance matrix Σ_{Φ}

Initialization: $\mathbf{m}_{\Phi} = \mathbf{m}_{\Phi}^*$, $\Sigma_{\Phi} = c_1 \cdot \Sigma_{\Phi}^*$

repeat

1. **Learn local statistics** of the pixel values from the vicinity of the expected curve based on the current mean vector \mathbf{m}_{Φ} and the current covariance matrix Σ_{Φ} . The resulting local statistics characterize the two sides of the curve. This step consists of two substeps:

- a) Determine the pixels in the vicinity of the expected image curve and assign them probabilistically to either side of the curve, based on \mathbf{m}_{Φ} and Σ_{Φ} .
- b) Compute local image statistics for both curve sides. For each side use only the set of pixels assigned to that side in (a). These local statistics are obtained using local windows, i.e. weights which are adapted in sizes and shapes to the expected curve and its uncertainty, see Figure 7. The resulting local statistics represent an expectation of “what the two sides of the curve look like”.

2. **Refine the estimate** of the model parameter vector. This step consists of two substeps:

- a) Update the mean vector \mathbf{m}_{Φ} using a maximum a posteriori (MAP) criterion derived from the local statistics. In this step, the mean vector \mathbf{m}_{Φ} is modified such that the pixels are assigned to the side of the curve they fit best according to the local statistics computed in step 1.
- b) Update the covariance matrix Σ_{Φ} based on the Hessian of the objective function used in (a).

until changes of \mathbf{m}_{Φ} and Σ_{Φ} are small enough

Post-processing: estimate the covariance matrix Σ_{Φ} from Hessian of objective function

return mean vector \mathbf{m}_{Φ} and covariance matrix Σ_{Φ}

Figure 6. The CCD algorithm iteratively refines a Gaussian a priori density $p(\Phi) = p(\Phi \mid \mathbf{m}_{\Phi}^*, \Sigma_{\Phi}^*)$ of model parameters to a Gaussian approximation $p(\Phi \mid \mathbf{m}_{\Phi}, \Sigma_{\Phi})$ of the posterior density $p(\Phi \mid \mathbf{I}^*)$.

differs mainly by: 1.) using local statistics, 2.) exploiting a curve model and optimizing model parameters rather than pixel classifications.

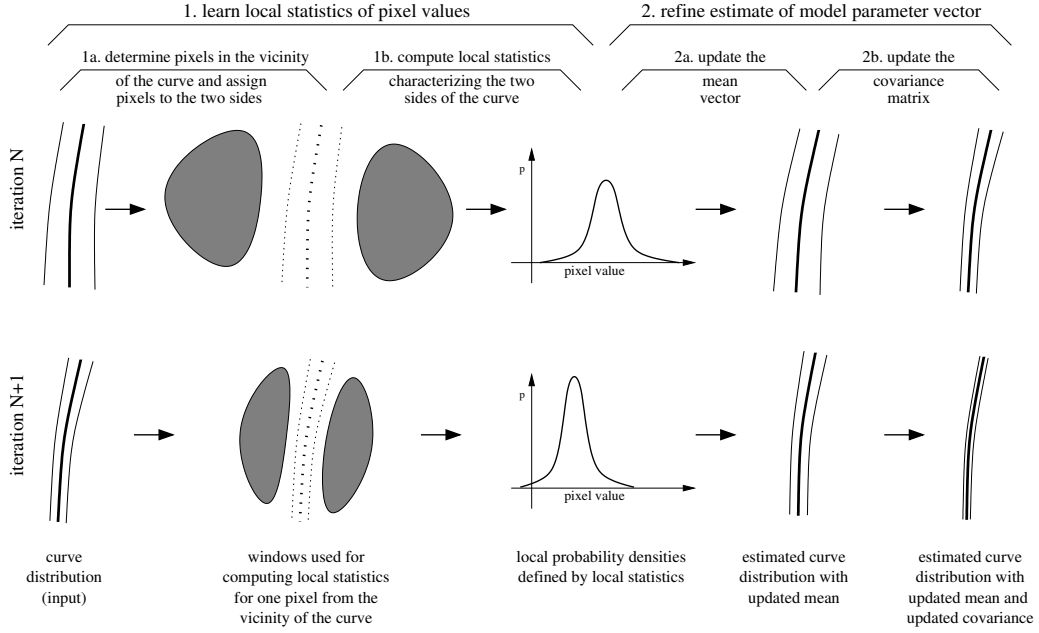


Figure 7. In each iteration of the CCD algorithm two steps are performed: 1. learn local statistics of pixel values from the vicinity of the expected curve; and 2. refine the estimate of the model parameter vector by assigning the pixels to the side they fit best according to the local statistics. Both steps consist of two substeps.

3. Learn Local Statistics (Step 1)

Let us now describe step 1 of the CCD algorithm, the learning of local statistics, in more detail. This step is performed by a sequence of two substeps. Step 1a (section 3.1) determines a set of pixels in the vicinity of the expected image curve and probabilistically assigns them to either side of the curve. Then, step 1b (section 3.2) computes local statistics of the image data for each of the two curve sides.

3.1. DETERMINE PIXELS IN THE VICINITY OF THE IMAGE CURVE (STEP 1A)

The Gaussian distribution of model parameters $p(\Phi | \mathbf{m}_\Phi, \Sigma_\Phi)$ and the model curve function \mathbf{c} define a probability distribution of the curve in the image. This curve distribution probabilistically assigns pixels in the vicinity of the curve to either side of the curve. In this section, the computation of the probabilistic assignments is derived.

The assignment $\mathbf{a}_v(\mathbf{m}_\Phi, \Sigma_\Phi) = (a_{v,1}(\mathbf{m}_\Phi, \Sigma_\Phi), a_{v,2}(\mathbf{m}_\Phi, \Sigma_\Phi))^T$ of a pixel v is a function of the mean \mathbf{m}_Φ and the covariance Σ_Φ . The first component $a_{v,1} \in [0, 1]$ describes to which extent pixel v is expected to be influenced by side 1 of the curve. The second component is the equivalent for side 2 given by $a_{v,2} = 1 - a_{v,1}$. Before we define the assignment of a pixel v for a distribution of model parameters, we first define the assignment $\tilde{a}_{v,1}(\Phi)$ for a single vector Φ of model parameters. We model a standard charge-coupled device image

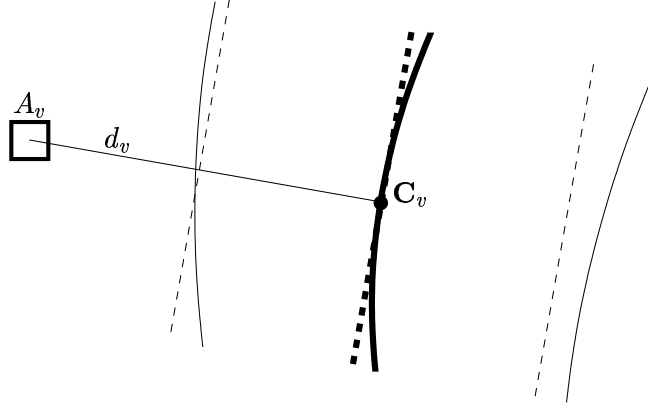


Figure 8. Local linear approximation (thick dashed line) of the curve function (thick solid line). The thin solid and thin dashed lines indicate the confidence interval of the curve and its approximations, respectively.

sensor, which integrates the radiance function over the photosensitive element of the pixel (Baker et al., 1998). The assignment $\tilde{a}_{v,1}(\Phi)$ is the fraction of the photosensitive area which belongs to side 1. It is given as an integral over the photosensitive area A_v of pixel v :

$$\tilde{a}_{v,1}(\Phi) = \frac{1}{|A_v|} \int_{A_v} l(\mathbf{x}, \Phi) d\mathbf{x} \quad (3)$$

where $|A_v|$ is the size of the photosensitive area and the label function $l(\mathbf{x}, \Phi)$ indicates on which side of the curve the point \mathbf{x} lies:

$$l(\mathbf{x}, \Phi) = \begin{cases} 1 & \text{if point } \mathbf{x} \text{ is on side 1 of the image curve } \mathbf{c}(\cdot, \Phi) \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

We define the probabilistic assignment $a_{v,1}(\mathbf{m}_\Phi, \Sigma_\Phi)$ induced by the Gaussian distribution $p(\Phi | \mathbf{m}_\Phi, \Sigma_\Phi)$ of model parameters as the expectation of $\tilde{a}_{v,1}(\Phi)$:

$$a_{v,1}(\mathbf{m}_\Phi, \Sigma_\Phi) = \mathcal{E}[\tilde{a}_{v,1}(\Phi)] \quad (5)$$

$$= \int_{\mathbb{R}^{D_\Phi}} \tilde{a}_{v,1}(\Phi) \cdot p(\Phi | \mathbf{m}_\Phi, \Sigma_\Phi) d\Phi \quad (6)$$

where D_Φ denotes the dimension of the model parameter vector Φ . For an arbitrary curve function \mathbf{c} , the probabilistic assignments cannot be computed in a closed form due to the integrations in equations (3) and (6). We next derive an efficient approximation of these assignments.

Efficient Approximation of the Assignments

For each pixel v we approximate the curve function $\mathbf{c}(\omega, \Phi)$ in the vicinity of $(\omega_v, \mathbf{m}_\Phi)$ by a function that is linear in ω and in Φ . The value ω_v is chosen such that $\mathbf{C}_v := \mathbf{c}(\omega_v, \mathbf{m}_\Phi)$ is the point on the ‘mean’ curve $\mathbf{c}(\cdot, \mathbf{m}_\Phi)$ which is closest to the center of gravity of the photosensitive area A_v , see Figure 8. For a linear curve function \mathbf{c} , the point $\mathbf{c}(\omega_v, \Phi)$ is

Gaussian distributed with mean $\mathbf{C}_v = \mathbf{c}(\omega_v, \mathbf{m}_\Phi)$ and covariance $\mathbf{J}_v \cdot \Sigma_\Phi \cdot \mathbf{J}_v^T$, where \mathbf{J}_v denotes the Jacobian of \mathbf{c} , i.e. the partial derivatives of \mathbf{c} w.r.t. the model parameters Φ in the point $(\omega_v, \mathbf{m}_\Phi)$. The displacement $D_v(\mathbf{x}, \Phi)$, i.e. the signed distance, between a point $\mathbf{x} \in A_v$ and the curve $\mathbf{c}(\cdot, \Phi)$ can be approximated by

$$D_v(\mathbf{x}, \Phi) = \mathbf{n}_v^T \cdot (\mathbf{x} - \mathbf{c}(\omega_v, \Phi)), \quad (7)$$

where \mathbf{n}_v^T is the normal vector of the curve in the point $\mathbf{c}(\omega_v, \mathbf{m}_\Phi)$. For a linear curve function \mathbf{c} , the displacement $D_v(\mathbf{x}, \Phi)$ is Gaussian distributed, $D_v(\mathbf{x}, \Phi) \sim \mathcal{N}(d_v(\mathbf{x}), \sigma_v^2)$. Its mean $d_v(\mathbf{x})$ is given by

$$d_v(\mathbf{x}) = \mathbf{n}_v^T \cdot (\mathbf{x} - \mathbf{C}_v) \quad (8)$$

and its variance σ_v^2 holds

$$\sigma_v^2 = \mathbf{n}_v^T \cdot \mathbf{J}_v \cdot \Sigma_\Phi \cdot \mathbf{J}_v^T \cdot \mathbf{n}_v. \quad (9)$$

The variance σ_v^2 describes the uncertainty of the curve along the curve normal, induced by the covariance Σ_Φ .³ The probability $p_{v,1}(\mathbf{x}, \mathbf{m}_\Phi, \Sigma_\Phi)$ that a point \mathbf{x} lies on side 1 of the curve can be obtained by integrating the Gaussian probability density function (pdf) of the displacement $D_v(\mathbf{x}, \Phi)$, which yields

$$p_{v,1}(\mathbf{x}, \mathbf{m}_\Phi, \Sigma_\Phi) = \frac{1}{2} \cdot \text{erf} \left(\frac{d_v(\mathbf{x})}{\sqrt{2} \cdot \sigma_v} \right) + \frac{1}{2}, \quad (10)$$

where $\text{erf}(\cdot)$ is the error function. Finally, the probabilistic assignment $a_{v,1}(\mathbf{m}_\Phi, \Sigma_\Phi)$ is obtained by integrating $p_{v,1}(\mathbf{x}, \mathbf{m}_\Phi, \Sigma_\Phi)$ over the photosensitive area A_v of pixel v :

$$a_{v,1}(\mathbf{m}_\Phi, \Sigma_\Phi) = \frac{1}{|A_v|} \int_{A_v} p_{v,1}(\mathbf{x}, \mathbf{m}_\Phi, \Sigma_\Phi) d\mathbf{x}. \quad (11)$$

The assignment for side 2 is simply given by

$$a_{v,2}(\mathbf{m}_\Phi, \Sigma_\Phi) = 1 - a_{v,1}(\mathbf{m}_\Phi, \Sigma_\Phi). \quad (12)$$

For the integration in equation (11), a piecewise polynomial approximation of the error function is used. Note that the accuracy of the approximations used in this section increases as the uncertainty specified by the covariance Σ_Φ decreases. In step 2b of the algorithm (see Figure 6) the covariance Σ_Φ is updated, which usually decreases the uncertainty. This is the reason why the CCD algorithm achieves high sub-pixel accuracy despite the used approximations. Higher accuracies may be possible by using higher order approximations of the curve function \mathbf{c} .

³ Additional inaccuracies, e.g. an inaccurate curve function $\mathbf{c}(\cdot)$ or image blurring, can be taken into account by adding an appropriate variance to the right hand side of equation (9).

By \mathcal{V}_u we denote the set of potential edge pixels v , i.e. pixels which are not clearly assigned to either side of the curve. For these pixels $v \in \mathcal{V}_u$, the assignments must be refined. By \mathcal{V}_s we denote the set of pixels which are assigned to one side with high certainty but which are close to pixels in \mathcal{V}_u . From pixels in \mathcal{V}_s , local statistics are learned in step 1b. In step 1a, the vicinity $\mathcal{V} = \mathcal{V}_u \cup \mathcal{V}_s$ of the expected curve is determined and for all pixels $v \in \mathcal{V}$, the probabilistic assignments \mathbf{a}_v are computed. Figure 12 row b on page 28 depicts the components $a_{v,1}$ of the assignments for different iterations. In the first iteration step, the uncertainty of the curve is higher at the right side than at the left side. Hence, the assignments are smoother at the right side.

Local Curve Coordinates

Furthermore, for all pixels $v \in \mathcal{V}$ local curve coordinates $(d_v, d_v^\perp)^T$ are determined, which are better suited to describe the vicinity of the curve than pixel coordinates. The first component d_v is the displacement of the pixel's center point \mathbf{v} to the mean curve according to function $d_v(\cdot)$, defined in equation (8):

$$d_v = d_v(\mathbf{v}). \quad (13)$$

The second component d_v^\perp denotes the perpendicular coordinate, i.e. the position of v along the curve. All pixels in \mathcal{V} are sorted according to the coordinate d_v^\perp , which allows for a more efficient computation in step 1b. Exploiting the fact that adjacent pixels have similar coordinates, this sorting can be done efficiently.

3.2. COMPUTING LOCAL STATISTICS

For the two sides separated by the curve, local statistics of the pixel values are learned from pixels that are assigned to one side of the curve with high certainty. The purpose of these statistics is to locally characterize the two sides of the curve. First, in section 3.2.1, we relate our statistical model to other context-sensitive statistical models used in computer vision. Then, in section 3.2.2, local windows (weights) describing the impact of a pixel on the local statistics are derived. Finally, in section 3.2.3, a fast recursive computation of the local statistics is proposed.

3.2.1. *Context-sensitive Statistical Models*

In order to decide to which side of the curve a pixel fits best, we consider the surrounding, i.e. the context of the pixel. A popular means of probabilistically modeling contextual constraints are Markov Random Fields (MRF). Some of the most frequently used MRFs are auto-normal models, also called Gaussian MRFs (Chellappa, 1985; Li, 2001). Gaussian MRFs and the related simultaneous auto-regression (SAR) models (Li, 2001) regard the pixel value as a Gaussian random variable with its expectation depending linearly on the neighboring pixels.

However, in these models the variance/covariance does not directly depend on the pixel's vicinity. Homogeneous Gaussian MRFs and SAR models have a single variance/covariance per image region.

We propose a statistical model which is related to Gaussian MRF and SAR models in the sense that it also employs Gaussian distributions which are conditioned on the pixels' vicinities. However, in our model the vicinity of a pixel not only affects the mean vector but also the covariance matrix. Our model allows that both the mean vectors and the covariance matrices of both sides of the curve smoothly vary along the curve. The local mean vectors and local covariance matrices define for each side of the curve and each pixel local multi-variate Gaussian approximations of the underlying distributions.⁴ The local adaption of both the mean vector and the covariance matrix is the key to the flexibility of our statistical model. Due to the local adaption of the statistics, very often the two sides of the curve can accurately be separated even if the actual distributions are non-Gaussian. For example, often a multi-modal distribution of local pixel values spans roughly just a linear sub-space within the multi-dimensional space of possible pixel values. Such linear sub-spaces can be described accurately by a Gaussian distribution even if the actual distribution is non-Gaussian.

The approach probably most similar to the CCD algorithm is the approach of Ronfard (1994). The author also uses local statistics in order to determine region boundaries. Ronfard partitions the vicinity of the curve into stripes perpendicular to the curve and assumes constant statistics within each stripe for each side of the curve. In order to avoid the spatial discretization involved in this approach, we model the statistics as a function of the position along the curve. While Ronfard uses the same variance for all stripes, we use locally adapted covariance matrixes.⁵ For each pixel, i.e. position d_v^- along the curve, we employ two sets of pixels in order to compute the local statistics $\mathbf{S}_1(d_v^-)$ and $\mathbf{S}_2(d_v^-)$ corresponding to the two sides of the curve. The pixels within such a set are scaled by weights which are described in the next section.

3.2.2. *Weighting the Pixels in the Vicinity of the Curve*

The weight $w_{p,s}(d_v^-)$ specifies to which extent a pixel p is taken into account for the computation of $\mathbf{S}_s(d_v^-)$, the local statistics for the sides $s \in \{1, 2\}$ at position d_v^- . The local statistics $\mathbf{S}_s(d_v^-)$ are used in step 2 in order to probabilistically classify pixel v to the side it fits best. From this goal, we derive six partially conflicting requirements for the weights $w_{p,s}(d_v^-)$:

1. Only pixels p which are likely to belong to the desired side s should have a positive weight; all other pixels p should have a weight of zero.

⁴ Higher order moments of the underlying distribution may be taken into account by a pre-processing step, which maps the pixel value \mathbf{I}^* to a possibly higher dimensional vector using a non-linear transformation.

⁵ Moreover, the two approaches differ in the way the statistics are obtained and in the way the optimization for the curve parameters is performed.

2. Since the statistical dependence between two pixels decreases with the distance between the pixels, only pixels which are not too far apart from the expected curve should be taken into account.
3. For the same reason the distance between pixel p and pixel v along the curve should be small.
4. The weights should allow for a fast computation of the local statistics.
5. The number of pixels with a non-zero weight should be sufficient in order to reliably estimate the local statistics.
6. For reasons of numerical stability the weights should be continuous.

Since these requirements are hard to quantify, we propose a heuristic compromise⁶. We compute the weight $w_{p,s}(d_v^\pm)$ as a product of two functions:

$$w_{p,s}(d_v^\pm) = W_s(p) \cdot W^\pm(|d_v^\pm - d_p^\pm|). \quad (14)$$

The first function, W_s , assesses the relation between pixel p and the curve. It evaluates the proximity of p to the curve and the probability of p to belong to the desired side s . It mainly finds a compromise between restriction 1 and 2. The computation of $W_s(p)$ is described in detail in appendix A. The second function, W^\pm , in equation (14) assesses the distance $|d_v^\pm - d_p^\pm|$ between pixel p and pixel v along the curve (see restriction 3). Function W^\pm is defined as an exponentially declining function:

$$W^\pm(D) = \frac{\exp(-D/\alpha)}{2\alpha}. \quad (15)$$

The multiplicative structure of equation (14) combining two independent criteria and the exponential decline of $W^\pm(D)$ allow for a fast recursive computation of the local statistics. This will be described in section 3.2.3.

Figure 9 depicts the resulting weights $w_{p,1}(d_v^\pm)$ and $w_{p,2}(d_v^\pm)$ for different pixels p and a fixed curve coordinate d_v^\pm . It shows that pixels p which are too close to the expected curve, i.e. pixels which are likely to belong to the wrong side have weights equal or close to zero. Pixels which are too far from the expected curve also have low weights. Pixels p which have a big distance $|d_v^\pm - d_p^\pm|$ along the curve also have low weight. Furthermore, the size of the window adapts to the uncertainty of the curve, i.e. for a high uncertainty the window is wider than for a small uncertainty. Compare e.g. the left and the right side in Figure 9. Row c in Figure 12 on page 28 depicts the weights of one side for different iterations. This illustrates how the local window sizes decrease during the progress of the iteration. (In contrast to Figure 9, in Figure 12 the weights are not depicted for a fixed position d_v^\pm along the curve but for the position d_p^\pm which yields the maximum weight.)

⁶ A compromise could also be learned from a set of labeled images.

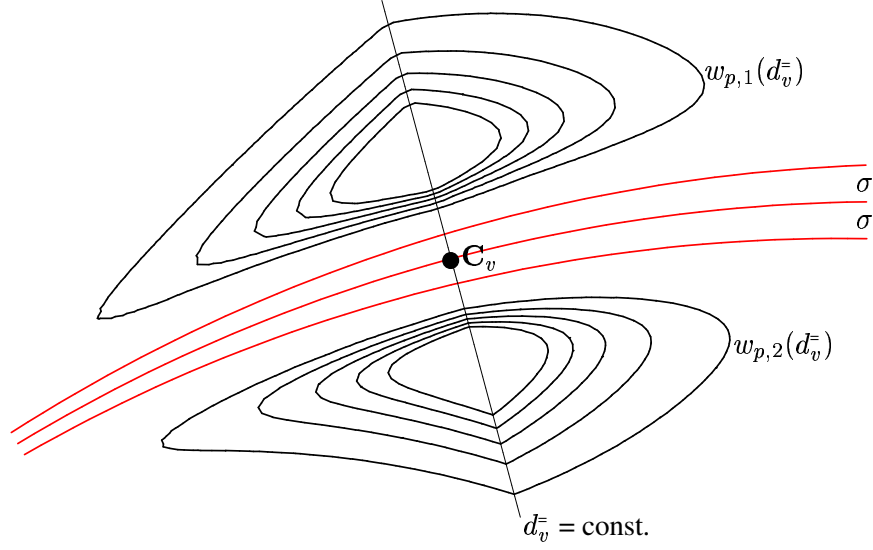


Figure 9. Contour plot of the windows (weights) used for computing local statistics: The three roughly parallel lines (red) describe the expected position and uncertainty (σ -interval) of the curve. For the pixels on the perpendicular line (straight black) local statistics are computed from the two depicted windows. The windows are adapted in size and shape to the expected curve and its uncertainty.

3.2.3. Recursive Computation of Local Statistics

Based on the weights $w_{p,s}(d_v^-)$ derived in the previous section, in this section local mean vectors $\mathbf{m}_{v,s}$ and local covariance matrices $\Sigma_{v,s}$ of the pixel values \mathbf{I}^* are derived for each side $s \in \{1, 2\}$ of the curve and each pixel v in the vicinity \mathcal{V}_u of the curve.

The local mean vectors $\mathbf{m}_{v,s}$ and local covariance matrices $\Sigma_{v,s}$ are obtained by

$$\mathbf{m}_{v,s} = \mathbf{M}_s(d_v^-) / w_s(d_v^-) \quad (16)$$

$$\Sigma_{v,s} = \mathbf{M}_s^2(d_v^-) / w_s(d_v^-) - \mathbf{m}_{v,s} \mathbf{m}_{v,s}^T + \kappa \mathbf{I} \quad (17)$$

where the sums $w_s(d_v^-)$ of weights, the first order weighted moments $\mathbf{M}_s(d_v^-)$, and the second order weighted moments $\mathbf{M}_s^2(d_v^-)$ are defined as

$$w_s(d_v^-) = \sum_{p \in \mathcal{V}} w_{p,s}(d_v^-) \quad (18)$$

$$\mathbf{M}_s(d_v^-) = \sum_{p \in \mathcal{V}} w_{p,s}(d_v^-) \mathbf{I}_p^* \quad (19)$$

$$\mathbf{M}_s^2(d_v^-) = \sum_{p \in \mathcal{V}} w_{p,s}(d_v^-) \mathbf{I}_p^* \cdot \mathbf{I}_p^{*T}. \quad (20)$$

The local weights $w_{p,s}(d_v^-)$ are specified in equation (14) and \mathbf{I}_p^* denotes the pixel values of pixel p . In equation (17) $\kappa \mathbf{I}$ is the identity matrix \mathbf{I} scaled by κ . It is used in order to avoid singularity, i.e. numerical problems, for degenerated distributions. In our experiments we choose κ to be quite small, $\kappa = 0.5$, where the pixel values are between 0 and 255.

The time complexity of the computation of the local statistics $\mathbf{S}_v := (\mathbf{m}_{v,1}, \boldsymbol{\Sigma}_{v,1}, \mathbf{m}_{v,2}, \boldsymbol{\Sigma}_{v,2})$ for a single pixel v is $O(N_v)$ where N_v denotes the number of pixels p with a non-zero weight $w_{p,s}(d_v^\pm)$. An independent computation of \mathbf{S}_v for all uncertainly assigned pixels $v \in \mathcal{V}_u$ would result in a time complexity of $O(\sum_{v \in \mathcal{V}_u} N_v)$ which is usually too expensive. However, the special structure of the weights $w_{p,s}(d_v^\pm)$ defined in equation (14) allows for a faster simultaneous computation of the local statistics \mathbf{S}_v . The function $W_s(p)$ evaluates the relation between the curve and pixel p and is independent of pixel v . The function $W^\pm(|d_v^\pm - d_p^\pm|)$ defines an exponential decline along the curve. Hence, the expressions $w_s(d_v^\pm)$, $\mathbf{M}_s(d_v^\pm)$, and $\mathbf{M}_s^2(d_v^\pm)$ defined in equations (18), (19), and (20) can be obtained as follows: First the quantities

$$W_s(p), \quad W_s(p) \cdot \mathbf{I}_p^*, \quad \text{and} \quad W_s(p) \cdot \mathbf{I}_p^* \cdot \mathbf{I}_p^{*T}$$

are computed for all pixels p in \mathcal{V}_u and then these quantities are blurred along the curve using the exponential filter defined by $W^\pm(\cdot)$. This blurring can be done efficiently in a recursive manner. The resulting time complexity for the simultaneous computations of the statistics \mathbf{S}_v for all $v \in \mathcal{V}_u$ is $O(|\mathcal{V}|)$ where $|\mathcal{V}|$ is the number of pixels in the vicinity \mathcal{V} of the curve. Due to the choice of $W^\pm(\cdot)$, the runtime does not increase with the blurring in the direction of the curve which allows for strong blurring.

4. Refine the Estimate of the Model Parameter Vector (Step 2)

In step 2 the estimate of the model parameter vector is refined using the probabilistic pixel assignments $\mathbf{a}_v(\mathbf{m}_\Phi, \boldsymbol{\Sigma}_\Phi)$ and the local statistics \mathbf{S}_v obtained in step 1. In section 4.1, we detail our observation model, specifying the assumed probabilistic relation between the model parameters and the image data. In sections 4.2 and 4.3, we show how the mean and the covariance of the posterior density can be updated using the observation model.

4.1. OBSERVATION MODEL

The observation model, i.e. likelihood function, describes the assumed probabilistic relation between the model parameters Φ and the image data \mathbf{I} . We model the pixel value \mathbf{I}_v of pixel $v \in \mathcal{V}$ as a weighted sum of two random variables $\mathbf{I}_{v,1}$ and $\mathbf{I}_{v,2}$:

$$\mathbf{I}_v = \tilde{a}_{v,1} \cdot \mathbf{I}_{v,1} + \tilde{a}_{v,2} \cdot \mathbf{I}_{v,2} \quad \text{where} \quad (21)$$

$$\tilde{\mathbf{a}}_v = (\tilde{a}_{v,1}, \tilde{a}_{v,2})^T = (\tilde{a}_{v,1}, 1 - \tilde{a}_{v,1})^T \quad (22)$$

and $\tilde{a}_{v,1} \in [0, 1]$. The random variables $\mathbf{I}_{v,1}$ and $\mathbf{I}_{v,2}$ correspond to the two sides of the curve and are assumed to be distributed according to the statistics of the corresponding side. We approximate the distributions of $\mathbf{I}_{v,1}$ and $\mathbf{I}_{v,2}$ by two Gaussians. Their mean vectors ($\mathbf{m}_{v,1}$ and

$\mathbf{m}_{v,2}$) and covariance matrices ($\Sigma_{v,1}$ and $\Sigma_{v,2}$) are obtained according to equations (16) and (17). The quantity $\tilde{a}_{v,1}$ specifies the fraction of the photosensitive area of pixel v which lies at side 1. The linear relation assumed in (21) corresponds to the sensor model used by Baker et al. (1998). It is also used in the alpha estimation literature (Ruzon and Tomasi, 2000; Chuang et al., 2001).

Due to the linear relation in (21), we obtain for the pixel value \mathbf{I}_v a Gaussian probability density $p(\mathbf{I}_v \mid \mathbf{m}_v, \Sigma_v)$. The estimates of the mean vector \mathbf{m}_v and covariance Σ_v are given by

$$\mathbf{m}_v = \tilde{a}_{v,1} \mathbf{m}_{v,1} + \tilde{a}_{v,2} \mathbf{m}_{v,2} \quad (23)$$

$$\Sigma_v = \tilde{a}_{v,1} \Sigma_{v,1} + \tilde{a}_{v,2} \Sigma_{v,2}. \quad (24)$$

To obtain an intuitive notation we define

$$p(\mathbf{I}_v \mid \tilde{\mathbf{a}}_v, \mathbf{S}_v) := p(\mathbf{I}_v \mid \mathbf{m}_v, \Sigma_v). \quad (25)$$

The notation $p(\mathbf{I}_v \mid \tilde{\mathbf{a}}_v, \mathbf{S}_v)$ indicates that the distribution of the pixel value \mathbf{I}_v depends on the side fraction $\tilde{\mathbf{a}}_v$ and the local statistics \mathbf{S}_v .

The assignments $\tilde{\mathbf{a}}_v$ depend on the model parameters Φ . We estimate the model parameters Φ by simultaneously optimizing the assignments $\tilde{\mathbf{a}}_v$ for all pixels in the vicinity \mathcal{V} of the curve. The probability density $p(\mathbf{I}_{\mathcal{V}} \mid \tilde{\mathbf{a}}_{\mathcal{V}}(\Phi), \mathbf{S}_{\mathcal{V}})$ of observing the image data $\mathbf{I}_{\mathcal{V}}$ in \mathcal{V} subjected to the model parameters Φ can be estimated by

$$p(\mathbf{I}_{\mathcal{V}} \mid \tilde{\mathbf{a}}_{\mathcal{V}}(\Phi), \mathbf{S}_{\mathcal{V}}) = \prod_{v \in \mathcal{V}} p(\mathbf{I}_v \mid \tilde{\mathbf{a}}_v(\Phi), \mathbf{S}_v). \quad (26)$$

The index v indicates quantities of a single pixel v . Analogously, the index \mathcal{V} indicates quantities of all pixels v in \mathcal{V} . For neighboring pixels similar windows are used in order to estimate the local statistics \mathbf{S}_v . Hence, in (26) pixel values of pixels on the same side of the curve are modeled as statistically dependent. The intensity of the statistical dependence is defined by the overlap of the windows. Equation (26) takes only pixels into account which are in the vicinity \mathcal{V} of the curve. Pixels outside \mathcal{V} are not used.

4.2. UPDATE THE MEAN (STEP 2A)

4.2.1. MAP Estimation

Using the observation density derived in (26) the MAP estimate $\hat{\Phi}$ of the model parameters Φ can be written as

$$\hat{\Phi} = \arg \max_{\Phi} F(\Phi) \quad \text{with} \quad (27)$$

$$F(\Phi) = p(\mathbf{I}_{\mathcal{V}} = \mathbf{I}_{\mathcal{V}}^* \mid \tilde{\mathbf{a}}_{\mathcal{V}}(\Phi), \mathbf{S}_{\mathcal{V}}) \cdot p(\Phi \mid \mathbf{m}_{\Phi}^*, \Sigma_{\Phi}^*) \quad (28)$$

where $\mathbf{I}_{\mathcal{V}}^*$ denotes the sensed image data in the curve vicinity \mathcal{V} and $p(\Phi \mid \mathbf{m}_{\Phi}^*, \Sigma_{\Phi}^*)$ denotes the Gaussian a priori density. The optimization of the product can be transformed into a

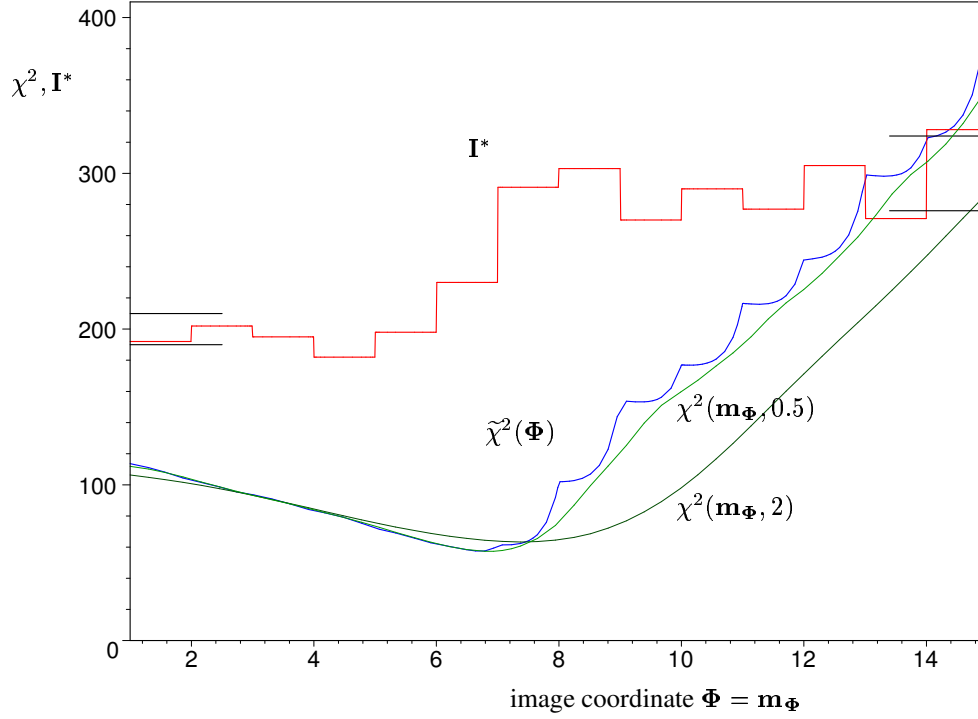


Figure 10. Detection of an edge in an 1-D array of gray values \mathbf{I}^* : the objective function $\tilde{\chi}^2(\Phi)$ which assesses the edge hypothesis Φ is non-differentiable. The smooth approximations $\chi^2(\mathbf{m}_\Phi, \Sigma_\Phi)$, depicted for $\Sigma_\Phi = 0.5$ and $\Sigma_\Phi = 2$, are differentiable. However, they have (slightly) different global minima. The horizontal lines (left and right margins) indicate the σ -interval characterizing the distributions of the two sides.

numerically more favorable optimization of a sum by

$$\hat{\Phi} = \arg \min_{\Phi} \tilde{\chi}^2(\Phi) \quad \text{with} \quad (29)$$

$$\tilde{\chi}^2(\Phi) = -2 \ln[p(\mathbf{I}_V = \mathbf{I}_V^* \mid \tilde{\mathbf{a}}_V(\Phi), \mathbf{S}_V) \cdot (\Phi \mid \mathbf{m}_\Phi^*, \Sigma_\Phi^*)]. \quad (30)$$

Due to the nature of an imaging sensor, the optimization in (29) is not trivial. An imaging sensor performs a spatial discretization of the scene which causes a non-smooth relation between the image data, e.g. RGB values, and the model parameters. For most pixels v the assignments $\tilde{a}_{v,1}$ are equal to 1 or 0. Only for the edge pixels the partial derivatives of $\tilde{a}_{v,1}$ for the model parameters Φ are not zero. The objective function $\tilde{\chi}^2$ typically has multiple points with high curvature or even points of discontinuity, see Figure 10. Hence, gradient-descent optimization methods such as Newton-Iteration have a very limited area of convergence. In the next section we show how the area of convergence can be substantially increased.

4.2.2. Fitting a ‘Blurred Model’

In order to obtain a smooth objective function, evaluating the fit between the image data and a single vector of model parameters, usually the image data are blurred, e.g. (Kass et al., 1988).

We take the opposite approach. We use non-blurred image data and a ‘blurred’ model. Our objective function takes the uncertainty of the estimated model parameters into account by evaluating the fit between the non-blurred image data and a Gaussian distribution of model parameters.

The advantages are as follows: (i) The capture range is enlarged according to the uncertainty of the model parameters. For parts of the curve with a high uncertainty the capture range is enlarged more than for parts with a smaller uncertainty. (ii) Optimizing the fit between an image and a ‘blurred model’ is in general computationally cheaper than blurring the image. (iii) Obviously, without blurring the image data high frequency information, e.g. the local covariances of the RGB values, can be used. The covariances are important especially for the characterization of texture.

The discontinuities of the derivatives of $\tilde{\chi}^2$ are caused by the non-smooth assignments $\tilde{\mathbf{a}}_v(\Phi)$. Hence, we substitute $\tilde{\mathbf{a}}_v(\Phi)$ by the probabilistic assignment $\mathbf{a}_v(\mathbf{m}_\Phi, \Sigma_\Phi)$ obtained in step 1a. While $\tilde{\mathbf{a}}_v(\Phi)$ is the assignment for a single vector of model parameters, $\mathbf{a}_v(\mathbf{m}_\Phi, \Sigma_\Phi)$ is the expectation of the assignment for a Gaussian distribution of model parameters. If the mean \mathbf{m}_Φ approaches Φ and the covariance Σ_Φ approaches the zero matrix then $\mathbf{a}_v(\mathbf{m}_\Phi, \Sigma_\Phi)$ approaches $\tilde{\mathbf{a}}_v(\Phi)$. For a non-singular covariance Σ_Φ the function $\mathbf{a}_v(\cdot, \Sigma_\Phi)$ is a smooth approximation of $\tilde{\mathbf{a}}_v(\cdot)$. In each iteration the covariance matrix is updated in step 2b and thereby the smoothing is (usually) reduced. Figure 10 illustrates the objective function for different uncertainties (levels of smoothing). The function $\chi^2(\cdot, 0.5)$ with little blurring better approximates the objective function $\tilde{\chi}^2(\cdot)$ than the function $\chi^2(\cdot, 2)$ with more blurring. However, $\chi^2(\cdot, 0.5)$ has a smaller area of convergence when using Newton-Iteration as optimization method.

We interpret the estimate $\hat{\Phi}$ of the model parameters Φ as the mean \mathbf{m}_Φ of a Gaussian approximation of the posterior distribution. Therefore, in the following we denote the estimate of the model parameters by \mathbf{m}_Φ . With this notation and the substitution of $\tilde{\mathbf{a}}_v(\cdot)$ by $\mathbf{a}_v(\cdot, \Sigma_\Phi)$ the estimate \mathbf{m}_Φ of the model parameters Φ can be written as

$$\mathbf{m}_\Phi = \arg \min_{\mathbf{m}_\Phi} \chi^2(\mathbf{m}_\Phi) \quad \text{with} \quad (31)$$

$$\chi^2(\mathbf{m}_\Phi) = -2 \ln[p(\mathbf{I}_\mathcal{V} = \mathbf{I}_\mathcal{V}^* \mid \mathbf{a}_\mathcal{V}(\mathbf{m}_\Phi, \Sigma_\Phi), \mathbf{S}_\mathcal{V}) \cdot p(\mathbf{m}_\Phi \mid \mathbf{m}_\Phi^*, \Sigma_\Phi^*)]. \quad (32)$$

The term $-2 \ln p(\mathbf{I}_\mathcal{V} = \mathbf{I}_\mathcal{V}^* \mid \mathbf{a}_\mathcal{V}(\mathbf{m}_\Phi, \Sigma_\Phi), \mathbf{S}_\mathcal{V})$ evaluates the fit between the sensed image data $\mathbf{I}_\mathcal{V}^*$ and the Gaussian distribution of model parameters with mean \mathbf{m}_Φ and covariance Σ_Φ . The term $-2 \ln p(\mathbf{m}_\Phi \mid \mathbf{m}_\Phi^*, \Sigma_\Phi^*)$ evaluates the fit between the estimate \mathbf{m}_Φ and the a priori density. In order to optimize $\chi^2(\mathbf{m}_\Phi)$, only a single Newton-Iteration step is performed. Afterwards, the steps 2b and 1 of the CCD algorithm are executed, which yield a new evaluation $-2 \ln p(\mathbf{I}_\mathcal{V} = \mathbf{I}_\mathcal{V}^* \mid \mathbf{a}_\mathcal{V}(\mathbf{m}_\Phi, \Sigma_\Phi), \mathbf{S}_\mathcal{V})$ to be optimized.

Figure 12 row d (page 28) depicts the image data expected by the blurred model, i.e. the image data $\mathbf{I}_\mathcal{V}$ optimizing the blurred observation density $p(\mathbf{I}_\mathcal{V} \mid \mathbf{a}_\mathcal{V}(\mathbf{m}_\Phi, \Sigma_\Phi), \mathbf{S}_\mathcal{V})$. The evaluation of a fit between the image data and the blurred model does not only depend on the

difference between the sensed image data (row a) and the expected image data (row d). The evaluation of a fit depends also on the covariance expected by the blurred model. Due to the high number of covariance elements, only the determinants of the covariances are illustrated in row e. The upper side of the curve is more inhomogeneous than the lower side (see row a). Hence, at the upper side the expected uncertainty is higher (see row e).

Figure 12 row f (page 28) depicts the energy $\chi_v^2 = -2 \ln p(\mathbf{I}_v = \mathbf{I}_v^* \mid \mathbf{a}_v(\mathbf{m}_\Phi, \Sigma_\Phi), \mathbf{S}_v)$ for each pixel v , i.e. the contribution of v to the objective function χ^2 . The energy is approximately two times the squared difference between the observed pixel value \mathbf{I}_v^* (row a) and the expectation according to the blurred model (row d) weighted by the expected covariances (row e).

For the estimate \mathbf{m}_Φ optimizing $\chi^2(\mathbf{m}_\Phi)$, the partial derivatives of $\chi^2(\mathbf{m}_\Phi)$ must be zero. For each pixel in the vicinity \mathcal{V} of the curve, the partial derivative of the energy χ_v^2 can be interpreted as a force. Step 2a of the CCD algorithm seeks to find the estimate \mathbf{m}_Φ which yields an equilibrium between the pixel forces. Figure 12 row g (page 28) depicts the forces acting in the direction perpendicular to the expected curve. Pixels depicted as bright act in a direction opposite to the pixels depicted as dark. During the first iterations the majority of the pixels forces the curve downward.

The area of pixels which influence the position of the expected curve depends on the uncertainty of the curve in the image. In Figure 12 the initial uncertainty of the curve is high at the right side and smaller at the left side. Consequently, at the right side the area of pixels influencing the curve estimate is wider than at the left side. However, pixels in the wider area exert on average lower forces than pixels in the narrow area. Furthermore, pixels which are close to the expected curve exert on average stronger forces than pixels which are further apart.

4.3. UPDATE THE COVARIANCE (STEP 2B)

In step 2b the covariance Σ_Φ describing the uncertainty of the estimate \mathbf{m}_Φ is updated. For the case where the relation between the model parameters and the observations is linear and the observation noise is Gaussian the covariance Σ_Φ of the χ^2 -estimate \mathbf{m}_Φ is given by

$$\Sigma_\Phi = \frac{2}{H} \quad (33)$$

where H denotes the Hessian of $\chi^2(\mathbf{m}_\Phi)$ in the point \mathbf{m}_Φ minimizing $\chi^2(\mathbf{m}_\Phi)$ (Press et al., 1996). Since in our case the relation between the model parameters and the image data is not linear, equation (33) can only be regarded as an estimate of the uncertainty. Another problem arises due to the way the estimate \mathbf{m}_Φ is obtained. In step 2a only a single iteration step is performed. Hence, the resulting estimate \mathbf{m}_Φ does not necessarily minimize the objective function χ^2 . Therefore, we update the covariance Σ_Φ by the following heuristics:

$$\Sigma_\Phi := c_2 \Sigma_\Phi + (1 - c_2) \frac{2}{H}. \quad (34)$$

Here $c_2 \Sigma_{\Phi}$ is the covariance computed in the last iteration step scaled by c_2 and $\frac{2}{H}$ is the estimate of the covariance according to (33). However, here H is the Hessian in the point \mathbf{m}_{Φ} , which does not necessarily maximize χ^2 exactly. The parameter $c_2 \in [0, 1]$, e.g. 0.25, specifies the maximum decrease of the covariance within one iteration step. If c_2 is too high the covariance declines slowly which leads to a high number of iterations. If c_2 is too small the CCD algorithm may converge to a wrong solution.

At the beginning of the iteration the covariance Σ_{Φ} usually decreases roughly exponentially. However, for an overrated estimate \mathbf{m}_{Φ} , i.e. an estimate \mathbf{m}_{Φ} which is not as good as specified by Σ_{Φ} , the curvature (Hessian) of the objective function decreases. In Figure 10 the curvature of the χ^2 function has its maximum close to the minimum of χ^2 . This slows down the reduction of Σ_{Φ} or even increases Σ_{Φ} if the quality of the estimate \mathbf{m}_{Φ} is overrated by Σ_{Φ} . Hence, the chance that the iteration still converges is increased. Steps 1 and 2 of the CCD algorithm are iterated until the changes of the estimate \mathbf{m}_{Φ} and the associated covariance Σ_{Φ} are small enough.

Post-Processing: After the last iteration the covariance of the estimate \mathbf{m}_{Φ} is estimated by $\Sigma_{\Phi} := \frac{2}{H}$. Finally the estimate \mathbf{m}_{Φ} of the model parameters and the estimate Σ_{Φ} of the corresponding covariance are returned.

Finally, we summarize the CCD algorithm in Figure 11. In contrast to the introductory overview in Figure 6, here we use the mathematical symbols introduced above.

5. Experimental Evaluation

In this section, we present our experimental results. We apply the proposed CCD algorithm to five different kinds of model fitting problems: (i) segmentation of radially distorted lines, (ii) segmentation of circles, (iii) 3-D pose estimation of cylindrical objects, (iv) 3-D pose estimation of polyhedral objects, (v) fitting a deformable models, i.e. a Point Distribution Model (Cootes et al., 1993). Hanek (2003) gives a more comprehensive experimental evaluation.

(i) Lines: We model a line as the radially distorted image of a straight line (without endpoints). Here, we assume the internal camera parameters to be known. Hence, the set of all radially distorted lines has two degrees of freedom, namely the parameters of the undistorted line. By fitting such a model to an image, the parameters of the undistorted line can be obtained without explicitly removing the radial distortion, e.g. by resampling the image.

For the sake of a ground truth we first use semi-synthetic images. 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. Figure 12 row a (page 28) shows such a semi-synthetic image. For different iterations the estimated curves are superimposed on the image. During the process the initial error is reduced by more than 99%.

Contracting Curve Density (CCD) algorithm

Input: image data \mathbf{I}^* , curve function c , mean \mathbf{m}_Φ^* and covariance Σ_Φ^*
Output: estimate \mathbf{m}_Φ of model parameters and associated covariance Σ_Φ

 Initialization: mean $\mathbf{m}_\Phi = \mathbf{m}_\Phi^*$, covariance $\Sigma_\Phi = c_1 \cdot \Sigma_\Phi^*$
repeat

 1. **learn local statistics** of image data from the vicinity of the curve

- a) determine pixels v in vicinity \mathcal{V} of the image curve from c , \mathbf{m}_Φ and Σ_Φ
 $\forall v \in \mathcal{V}$ compute probabilistic assignment $\mathbf{a}_v(\mathbf{m}_\Phi, \Sigma_\Phi)$ to the sides of the curve
- b) $\forall v \in \mathcal{V}$ compute local statistics \mathbf{S}_v of image data $\mathbf{I}_\mathcal{V}^*$
 which characterize the two sides of the curve

 2. **refine estimate** of model parameter vector

- a) update mean \mathbf{m}_Φ by performing one iteration step of MAP estimation:

$$\mathbf{m}_\Phi = \arg \min_{\mathbf{m}_\Phi} \chi^2(\mathbf{m}_\Phi) \quad \text{with}$$

$$\chi^2(\mathbf{m}_\Phi) = -2 \ln[p(\mathbf{I}_\mathcal{V} = \mathbf{I}_\mathcal{V}^* \mid \mathbf{a}_\mathcal{V}(\mathbf{m}_\Phi, \Sigma_\Phi), \mathbf{S}_\mathcal{V}) \cdot p(\mathbf{m}_\Phi \mid \mathbf{m}_\Phi^*, \Sigma_\Phi^*)]$$

- b) updated covariance Σ_Φ from the Hessian of $\chi^2(\mathbf{m}_\Phi)$

until changes of \mathbf{m}_Φ and Σ_Φ are small enough

 Post-processing: estimate covariance Σ_Φ from Hessian of $\chi^2(\mathbf{m}_\Phi)$
return mean \mathbf{m}_Φ and covariance Σ_Φ

Figure 11. The CCD algorithm iteratively refines a Gaussian a priori density $p(\Phi) = p(\Phi \mid \mathbf{m}_\Phi^*, \Sigma_\Phi^*)$ of model parameters to a Gaussian approximation $p(\Phi \mid \mathbf{m}_\Phi, \Sigma_\Phi)$ of the posterior density $p(\Phi \mid \mathbf{I}^*)$.

(ii) Circles: We apply the CCD algorithm to the segmentation of circles parameterized by the center point and the radius. Figs. 13 and 14 depict the iterations for two semi-synthetic images. In both cases the initial error is reduced by more than 99.8% and the final error is less than 5% of a pixel. However, for real images with similarly complex content we assume that the sub-pixel accuracy is lower due to different effects, such as unknown blurring caused by the imaging device.

The next experiment shows that the CCD algorithm achieves a large area of convergence even for inhomogeneous regions. Figure 15 depicts the area of convergence for the image shown in Figure 13. The CCD algorithm is started with a radius which is 5 pixels higher than the correct radius. A high initial covariance defining an uninformed a priori distribution is used, leading to a large area of convergence. The area of convergence has roughly the size of the image circle. For a small initial uncertainty the image processing operations are focused

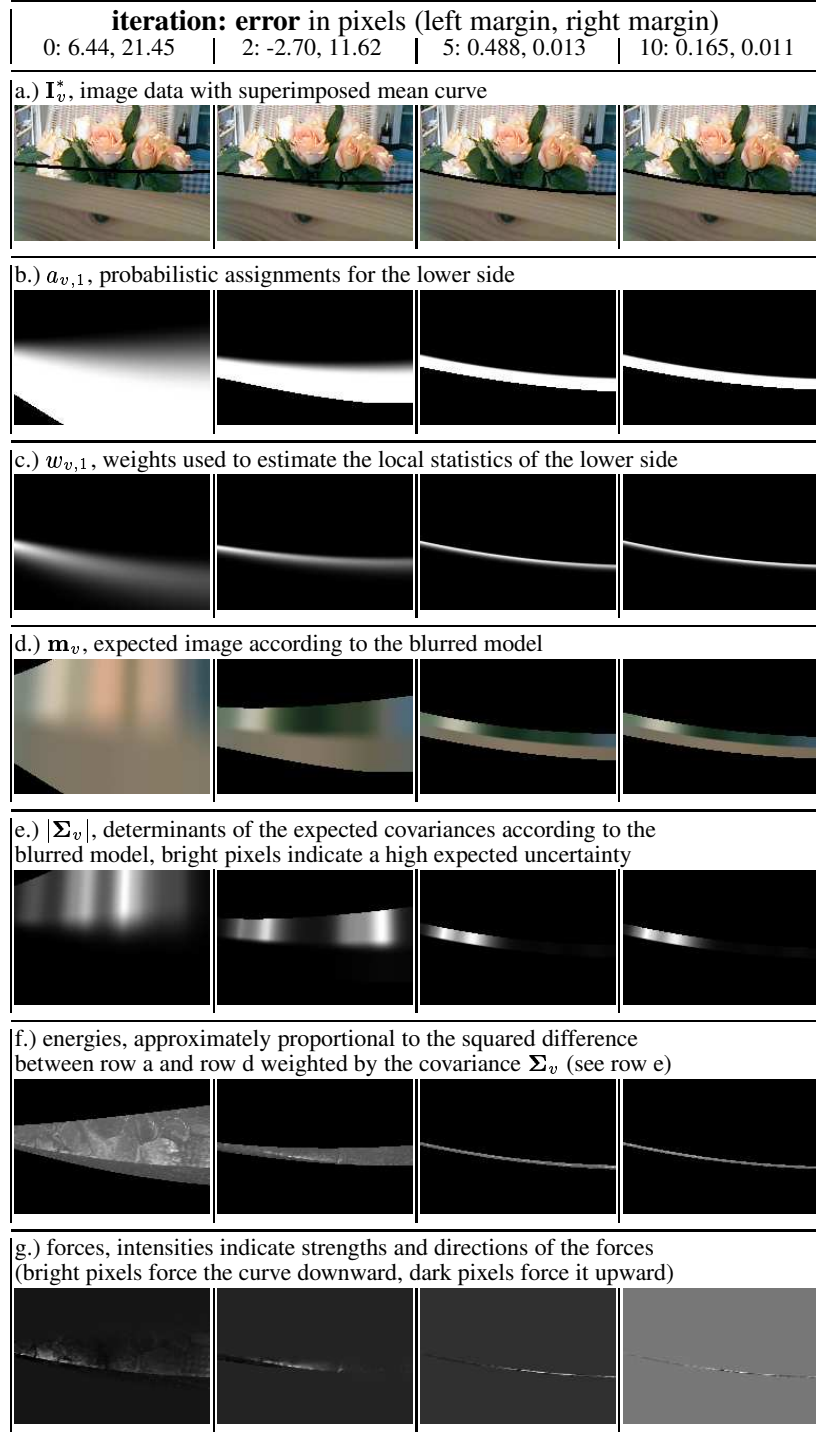


Figure 12. Semi-synthetic image showing roses behind a wooden board: the initial error is reduced by more than 99%. See text for a detailed description. (The images, except rows a and d, are individually normalized such that the grey values are within $[0,255]$.)





iteration: error in pixels (x, y, radius)			
0: 23.0, 10.0, -5.0	2: 8.39, 2.07, -6.71	6: .214, .012, -.272	10: -.025, .028, -.033
			

Figure 13. Despite the inhomogeneity of the foreground and the background the initial error is reduced by more than 99.8%.

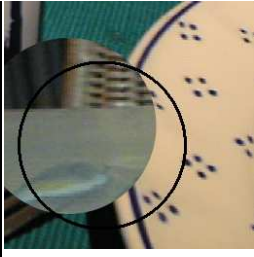
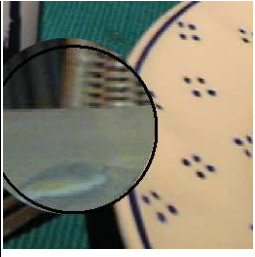
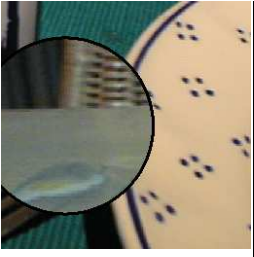
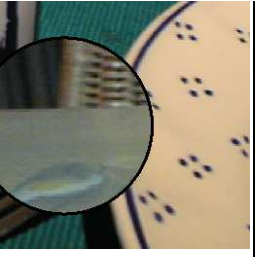
iteration: error in pixels (x, y, radius)			
0: 35.0, 20.0, -5.5	2: 6.43, 3.74, -4.93	8: -.041, -.018, -.053	13: -.040, -.013, -.044
			

Figure 14. The circle is only partially visible. Nevertheless, the final error is less than 5% of a pixel.

to a small number of pixels. Hence, the area of convergence is usually smaller. However, for a better initialization a smaller area of convergence is sufficient.

The runtime of the algorithm scales roughly linearly to the number of used pixels. Hence, the initial uncertainty has an important impact on the runtime. For example on a 500 MHz computer the first iteration step of Figure 13 takes about 4s. The initial uncertainty is big and roughly 10,000 pixels are used. After about 5 iterations the runtime is reduced to less than 1s per iteration. Hanek (2003) gives a more comprehensive analysis of the area of convergence and the runtime for many different textures.

(iii) Cylindrical objects: The 3-D pose of a cylinder (a mug of known hight and radius) is estimated with respect to the camera coordinate system. The cylinder's orientation is described by two Euler angles and its translation is described by a three-dimensional vector. Radial distortions of the lens are taken into account. The resulting contour is fitted to the image data, see Figure 2 on page 3. Despite the strong background clutter, the texture, and the shading, the CCD algorithm accurately estimates the mug's contour. Note that methods fitting the curve model not directly to the image data but to an edge map have to deal with multiple false

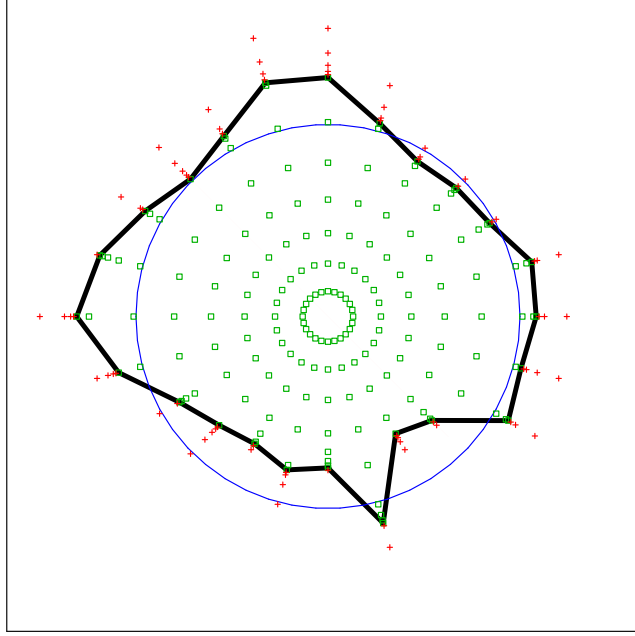


Figure 15. Area of convergence for the image depicted in Figure 13. The thick line separates all investigated converging center points (green rectangles) from the non-converging center points (red crosses). The thin blue line is the real image circle. The area of convergence has roughly the size of the image circle.

positive and false negative errors, see Figure 3a (page 5). Furthermore, the edge map shows a limited accuracy e.g. in the presence of inhomogeneous regions.

(iv) Polyhedral objects: The 6 pose parameters defining the 3-D pose of a polyhedral object are estimated by fitting the object contour to the image data. The contour consists of radially distorted line segments. Figure 16a shows a tea box with flower pattern lying in a flower meadow. Both the box and the background region have multiple internal edges, see Figure 16b. Nevertheless, the CCD algorithm accurately estimates the contour of the tea box.

The next experiment shows that the CCD algorithm is not only able to exploit object contours, but can also use internal model edges separating multiple possibly dependent image regions. We fit a rigid wire frame model to the image depicted in Figure 17a. The degrees of freedom are the 6 pose parameters. The CCD algorithm reliably estimates the pose of the box. Again, a gradient-based edge detector (Steger, 2000) yields multiple false positive and false negative errors, see Figure 17b.

(v) Deformable models: Finally a deformable model, i.e. a Point Distribution Model (Cootes et al., 1993), with 12 degrees of freedom is fitted to the image data. Figure 18 depicts the results for different iteration steps. Only four iterations are needed in order to largely reduce the initial error. Note: (i) The number of necessary iterations is similar as for low-dimensional optimization problems, e.g. Figure 12 on page 28 with two dimensions. Monte Carlo optimization methods would yield a strong increase of runtime if the dimension of the



Figure 16. Tea box with flower pattern lying in a flower meadow: a.) Despite the inhomogeneities of the foreground and the background, the CCD algorithm accurately fits the model of the tea box to the image contour (red = initialization, yellow = estimated contour). b.) Color edges detected by a gradient-based approach (Steger, 2000): The huge majority of the detected edges does not correspond to the contour of the box. Furthermore the result of the edge detection depends heavily on the illumination. In the bright part, bottom right, clearly more edges are detected.



Figure 17. Partially occluded box in front of an inhomogeneous background: a.) Due to the directed illumination, the contrast between the background and the box is heavily varying. Nevertheless, the CCD algorithm accurately fits a wire frame model to the image data (red = initialization, blue = estimate). b.) Here again a gradient-based edge detector (Steger, 2000) yields multiple false positives and false negative errors.

search space increases from 2 to 12. (ii) While region-based and graph-theoretic methods



Figure 18. Fitting a deformable model with 12 degrees of freedom: Already after 4 iterations the initial error is largely reduced.

usually require closed region boundaries, the CCD algorithm can also deal with non-closed curves as depicted in Figure 18.

6. Conclusion

We have proposed the Contracting Curve Density (CCD) algorithm, a novel method for fitting parametric curve models to boundaries of image regions. The method iteratively refines an a priori distribution of the model parameters to a Gaussian approximation of the posterior distribution.

The algorithm applies locally adapted criteria in order to separate the two sides of the curve. The separation criteria are iteratively obtained from local statistics of the curve's vicinity. Pixels, which are intersected by the curve are modeled by a mixture of two local distributions corresponding to the two sides of the curve. This locally adapted statistical modeling allows for separating the two sides of the curve with high sub-pixel accuracy even in the presence of severe texture, shading, clutter, partial occlusion, and strong changes of illumination.

The CCD algorithm employs a *blurred curve model* as a means for iteratively optimizing the fit. The algorithm optimizes the MAP criteria not only for a single vector of model parameters but for a distribution of model parameter vectors defining a blurred curve. During the iterative process not only the model parameters are refined but also the associated covariance matrix. From the covariance matrix the local uncertainties of the curve are obtained, which provide the basis for the automatic and local scale selection.

Our experiments show that the CCD algorithm achieves high sub-pixel accuracy and a large area of convergence even for challenging scenes. Hanek (2003) proposed an extension of the CCD algorithm called *CCD tracker*. The CCD tracker fits a curve to a sequence of images. The conducted experiments show that the CCD tracker outperforms other state-of-the-art trackers in terms of accuracy, robustness, and runtime (Hanek, 2003).

Appendix

A. Evaluating the Distance to the Mean Curve

In this appendix, we detail the function $W_s(p)$ which is used in equation (14) in order to obtain the weights $w_{p,s}(d_v^-)$. The function $W_s(p)$ assesses the relation between the pixel p and the current curve distribution. It mainly computes a compromise between requirements 1 and 2 described on page 18. We compute $W_s(p)$ as a product of two functions corresponding to the two requirements:

$$W_s(p) = W(a_{p,s}) \cdot W_p(d_p, \sigma_p). \quad (35)$$

The first function $W(a_{p,s})$ assesses the probability of pixel p to belong to the desired side s . The quantity $a_{p,s}$ is the probabilistic side assignment for pixel p according to equations (11) and (12). The function $W(a_{p,s})$ is defined by

$$W(a_{p,s}) = \max(0, [(a_{p,s} - \gamma_1)/(1 - \gamma_1)]^6) \quad (36)$$

with $\gamma_1 \in [0, 1[$. The function W is monotonously increasing and holds $\forall a_{p,s} \in [0, \gamma_1] : W(a_{p,s}) = 0$ and $W(1) = 1$. For the parameter γ_1 , we recommend values of about 0.5. This means roughly, pixels which belong to the desired side with a confidence of 50% or less are not used in order to compute the local statistics.

The second function, $W_p(d_p, \sigma_p)$, at the right hand side of equation (35) evaluates the proximity of pixel p to the curve. The quantity d_p denotes the expected signed distance between pixel p and the curve according to equation (13). The quantity d_p is evaluated relative to the uncertainty σ_p of the curve. The second parameter, σ_p , denotes the standard deviation, of d_p according to equation (9). For $W_p(d_p, \sigma_p)$ we choose a zero mean truncated Gaussian density:

$$W_p(d_p, \sigma_p) = c \cdot \max[0, \exp(-d_p^2/2\hat{\sigma}_p^2) - \exp(-\gamma_2)] \quad \text{where} \quad (37)$$

$$\hat{\sigma}_p = \gamma_3 \cdot \sigma_p + \gamma_4. \quad (38)$$

Here c is a normalization constant ensuring $W_p(d_p, \sigma_p)$ to be a probability density function. The parameter $\gamma_2 > 0$ defines the truncation of the Gaussian. It is used in order to limit the number of pixels with a non-zero weight, i.e. pixels which are taken into account. The standard deviation $\hat{\sigma}_p$ of the Gaussian is a linear function of the curve's uncertainty, i.e. σ_p . Hence, the width of the window automatically adapts to the uncertainty of the curve. The parameter $\gamma_4 > 0$ defines a lower boundary of the window's width. For all experiments we choose $\gamma_2 = 4$, $\gamma_3 = 4$, and $\gamma_4 = 3$.

Acknowledgments: The authors would like to thank Carsten Steger for providing the DLL of the color edge detector; Jianbo Shi and Jitendra Malik for providing the executable of the *Normalized Cuts* algorithm, and Lothar Hermes for running the PDC algorithm on our test image, see Figure 3. Furthermore, we thank Christoph Hansen for providing the PDM depicted in Figure 18.

References

- Amini, A., T. Weymouth, and R. Jain: 1990, 'Using Dynamic Programming for Solving Variational Problems in Vision'. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **12**(9), 855–867.
- Baker, S., S. Nayar, and H. Murase: 1998, 'Parametric Feature Detection'. *International Journal of Computer Vision* **27**(1), 27–50.
- Ballard, D. H.: 1981, 'Generalizing the Hough transform to detect arbitrary shapes'. *Pattern Recognition* **13**(2), 111–122.
- Belongie, S., C. Carson, H. Greenspan, and J. Malik: 1998, 'Color- and Texture-Based Image Segmentation Using the Expectation-Maximization Algorithm and its Application to Content-Based Image Retrieval'. In: *Proc. of the IEEE Int. Conf. on Computer Vision*. pp. 675–682.
- Bennett, J. and A. Khotanzad: 1998, 'Multispectral Random Field Models for Synthesis and Analysis of Color Images'. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **20**(3), 327–332.
- Blake, A. and M. Isard: 1998, *Active Contours*. Berlin Heidelberg New York: Springer-Verlag.
- Bongiovanni, G., P. Crescenzi, and C. Guerra: 1995, 'Parallel Simulated Annealing for Shape Detection'. *Computer Vision and Image Understanding* **61**(1), 60–69.
- Bouman, C. and K. Sauer: 1993, 'A generalized Gaussian image model for edge-preserving MAP estimation'. *IEEE Transactions on Image Processing* **2**(3), 296–310.
- Canny, J.: 1986, 'A Computational Approach to Edge Detection'. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **8**(6), 679–698.
- Chakraborty, A. and J. Duncan: 1999, 'Game-Theoretic Integration for Image Segmentation'. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **21**(1), 12–30.
- Chellappa, R.: 1985, 'Two-Dimensional Discrete Gauss Markovian Random Field Models for Image Processing'. *Progress in Pattern Recognition* **2**, 79–112.
- Chesnaud, C., P. Refregier, and V. Boulet: 1999, 'Statistical Region Snake-Based Segmentation Adapted to Different Physical Noise Models'. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **21**(11), 1145–1157.
- Chuang, Y., B. Curless, D. Salesin, and R. Szeliski: 2001, 'A Bayesian Approach to Digital Matting'. In: *Proc. of the IEEE Conf. Computer Vision and Pattern Recognition*. pp. II:264–271.

- Cootes, T., G. Edwards, and C. Taylor: 2001, 'Active Appearance Models'. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **23**(6), 681–684.
- Cootes, T., A. Hill, C. Taylor, and J. Haslam: 1994, 'The Use of Active Shape Models For Locating Structure In Medical Images'. *Image and Vision Computing* **12**(6), 355–365.
- Cootes, T., C. Taylor, A. Lanitis, D. Cooper, and J. Graham: 1993, 'Building and Using Flexible Models Incorporating Grey-Level Information'. In: *Proc. of the IEEE Int. Conf. on Computer Vision*. pp. 242–246.
- Cox, I., Y. Zhong, and S. Rao: 1996, 'Ratio Regions: A Technique for Image Segmentation'. In: *Proc. of the IEEE Int. Conf. on Pattern Recognition*. pp. B: 557–564.
- Dempster, A., N. Laird, and D. Rubin: 1977, 'Maximum likelihood from incomplete data via the EM algorithm'. *J. R. Statist. Soc. B* **39**, 1–38.
- Dubuisson-Jolly, M. and A. Gupta: 2001, 'Tracking Deformable Templates Using a Shortest Path Algorithm'. *Computer Vision and Image Understanding* **81**(1), 26–45.
- Felzenszwalb, P. and D. Huttenlocher: 1998, 'Image Segmentation Using Local Variation'. In: *Proc. of the IEEE Conf. Computer Vision and Pattern Recognition*. pp. 98–104.
- Geiger, D., A. Gupta, L. Costa, and J. Vlontzos: 1995, 'Dynamic-Programming for Detecting, Tracking, and Matching Deformable Contours'. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **17**(3), 294–302.
- Geman, S. and D. Geman: 1984, 'Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images'. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **6**(6), 721–741.
- Hanek, R.: 2001, 'The Contracting Curve Density Algorithm and its Application to Model-based Image Segmentation'. In: *Proc. of the IEEE Conf. Computer Vision and Pattern Recognition*. pp. I:797–804.
- Hanek, R.: 2003, 'Fitting Parametric Curve Models to Images Using Local Self-adapting Separation Criteria'. Phd thesis, Department of Informatics, Technische Universität München.
- Hanek, R., T. Schmitt, S. Buck, and M. Beetz: 2002a, 'Fast Image-based Object Localization in Natural Scenes'. In: *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*. pp. 116–122.
- Hanek, R., T. Schmitt, S. Buck, and M. Beetz: 2002b, 'Towards RoboCup without Color Labeling'. In: *RoboCup International Symposium*.
- Hermes, L., T. Zoller, and J. Buhmann: 2002, 'Parametric Distributional Clustering for Image Segmentation'. In: *Proc. of the European Conf. on Computer Vision*, Vol. 3. pp. 577–591.
- Hough, P. V.: 1962, 'Method and means for recognizing complex patterns'. U.S. Patent 3069654.
- Isard, M. and A. Blake: 1998, 'CONDENSATION – Conditional Density Propagation for Visual Tracking'. *International Journal of Computer Vision* **29**(1), 5–28.
- Jones, T. and D. Metaxas: 1998, 'Image Segmentation Based on the Integration of Pixel Affinity and Deformable Models'. In: *Proc. of the IEEE Conf. Computer Vision and Pattern Recognition*. pp. 330–337.
- Kass, M., A. Witkin, and D. Terzopoulos: 1988, 'Snakes: Active Contour Models'. *International Journal of Computer Vision* **1**(4), 321–331.
- Kollnig, H. and H. Nagel: 1995, '3d Pose Estimation By Fitting Image Gradients Directly To Polyhedral Models'. In: *Proc. of the IEEE Int. Conf. on Computer Vision*. pp. 569–574.
- Leventon, M., W. Grimson, and O. Faugeras: 2000, 'Statistical Shape Influence in Geodesic Active Contours'. In: *Proc. of the IEEE Conf. Computer Vision and Pattern Recognition*. pp. I:316–323.
- Li, P., T. Zhang, and A. Pece: 2003, 'Visual contour tracking based on particle filters'. *Image and Vision Computing* **21**(1), 111–123.
- Li, S.: 2001, *Markov Random Field Modeling in Image Analysis*. Berlin Heidelberg New York: Springer-Verlag.
- Lindeberg, T.: 1998, 'Feature Detection with Automatic Scale Selection'. *International Journal of Computer Vision* **30**(2), 79–116.

- Lowe, D. G.: 1991, 'Fitting Parameterized 3-D Models to Images'. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **13**(5), 441–450.
- Luo, H., Q. Lu, R. Acharya, and R. Gaborski: 2000, 'Robust Snake Model'. In: *Proc. of the IEEE Conf. Computer Vision and Pattern Recognition*. pp. I:452–457.
- MacCormick, J. and M. Isard: 2000, 'Partitioned Sampling, Articulated Objects, and Interface-Quality Hand Tracking'. In: *Proc. of the European Conf. on Computer Vision*. pp. II: 3–19.
- Malik, J., S. Belongie, T. Leung, and J. Shi: 2001, 'Contour and Texture Analysis for Image Segmentation'. *International Journal of Computer Vision* **43**(1), 7–27.
- Malik, J., S. Belongie, J. Shi, and T. Leung: 1999, 'Textons, Contours and Regions: Cue Integration in Image Segmentation'. In: *Proc. of the IEEE Int. Conf. on Computer Vision*. pp. 918–925.
- Manduchi, R.: 1999, 'Bayesian Fusion of Color and Texture Segmentations'. In: *Proc. of the IEEE Int. Conf. on Computer Vision*. pp. 956–962.
- McInerney, T. and D. Terzopoulos: 1996, 'Deformable models in medical image analysis: a survey'. *Medical Image Analysis* **1**(2), 91–108.
- Mirmehdi, M. and M. Petrou: 2000, 'Segmentation of Color Textures'. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22**(2), 142–159.
- Mortensen, E. and W. Barrett: 1998, 'Interactive Segmentation with Intelligent Scissors'. *Graphical Models and Image Processing* **60**(5), 349–384.
- Nalwa, V. and T. Binford: 1986, 'On Detecting Edges'. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **8**(6), 699–714.
- Panjwani, D. and G. Healey: 1995, 'Markov Random-Field Models for Unsupervised Segmentation of Textured Color Images'. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **17**(10), 939–954.
- Paragios, N. and R. Deriche: 2000, 'Coupled Geodesic Active Regions for Image Segmentation: A Level Set Approach'. In: *Proc. of the European Conf. on Computer Vision*. pp. 224–240.
- Pece, A. and A. Worrall: 2002, 'Tracking with the EM Contour Algorithm'. In: *Proc. of the European Conf. on Computer Vision*. pp. I: 3–17.
- Press, W. H., S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery: 1996, *Numerical Recipes in C*. Cambridge: Cambridge University Press.
- Robert, L.: 1996, 'Camera Calibration without Feature Extraction'. *Computer Vision and Image Understanding* **63**(2), 314–325.
- Ronfard, R.: 1994, 'Region-Based Strategies for Active Contour Models'. *International Journal of Computer Vision* **13**(2), 229–251.
- Ruzon, M. and C. Tomasi: 2000, 'Alpha Estimation in Natural Images'. In: *Proc. of the IEEE Conf. Computer Vision and Pattern Recognition*. pp. I:18–25.
- Ruzon, M. and C. Tomasi: 2001, 'Edge, Junction, and Corner Detection Using Color Distributions'. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **23**(11), 1281–1295.
- Schmitt, T., R. Hanek, M. Beetz, S. Buck, and B. Radig: 2002, 'Cooperative Probabilistic State Estimation for Vision-based Autonomous Mobile Robots'. *IEEE Transactions on Robotics and Automation* **18**(5), 670–684.
- Scalaroff, S. and L. Liu: 2001, 'Deformable Shape Detection and Description via Model-Based Region Grouping'. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **23**(5), 475–489.
- Shi, J. and J. Malik: 2000, 'Normalized Cuts and Image Segmentation'. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22**(8), 888–905.
- Steger, C.: 2000, 'Subpixel-Precise Extraction of Lines and Edges'. *International Archives of Photogrammetry and Remote Sensing XXXIII, part B3*, 141–156.
- Storvik, G.: 1994, 'A Bayesian-Approach to Dynamic Contours Through Stochastic Sampling and Simulated Annealing'. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **16**(10), 976–986.

- Sullivan, S. and J. Ponce: 1998, 'Automatic Model Construction and Pose Estimation From Photographs Using Triangular Splines'. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **20**(10), 1091–1096.
- Terzopoulos, D. and R. Szeliski: 1992, 'Tracking with Kalman Snakes'. In: *Active*. pp. 3–20.
- Thirion, B., B. Bascle, V. Ramesh, and N. Navab: 2000, 'Fusion of Color, Shading and Boundary Information for Factory Pipe Segmentation'. In: *Proc. of the IEEE Conf. Computer Vision and Pattern Recognition*. pp. II:349–356.
- Ulrich, M., C. Steger, A. Baumgartner, and H. Ebner: 2001, 'Real-Time Object Recognition Using a Modified Generalized Hough Transform'. In: E. Seyfert (ed.): *Photogrammetrie — Fernerkundung — Geoinformation: Geodaten schaffen Verbindungen*. Berlin, pp. 571–578.
- Vijayakumar, B., D. Kriegman, and J. Ponce: 1998, 'Invariant-Based Recognition of Complex Curved 3-D Objects from Image Contours'. *Computer Vision and Image Understanding* **72**(3), 287–303.
- Wu, Z. and R. Leahy: 1993, 'An Optimal Graph Theoretic Approach to Data Clustering: Theory and Its Application to Image Segmentation'. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **15**(11), 1101–1113.
- Xu, C. and J. Prince: 1998, 'Snakes, Shapes, and Gradient Vector Flow'. *IEEE Transactions on Image Processing* **7**(3), 359–369.
- Zeoller, T., L. Hermes, and J. M. Buhmann: 2002, 'Combined Color and Texture Segmentation by Parametric Distributional Clustering'. In: *Proc. of the IEEE Int. Conf. on Pattern Recognition*, Vol. 2. pp. 627–630.
- Zhong, Y., A. Jain, and M. Dubuisson-Jolly: 2000, 'Object Tracking Using Deformable Templates'. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22**(5), 544–549.
- Zhu, S. and A. Yuille: 1996, 'Region Competition: Unifying Snakes, Region Growing, and Bayes/MDL for Multiband Image Segmentation'. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **18**(9), 884–900.