T U M

I N S T I T U T F Ü R I N F O R M A T I K

Model-Based Image Segmentation Using Local Self-Adapting Separation Criteria

Robert Hanek

TUM-I0103
September 01

T E C H N I S C H E U N I V E R S I T Ä T M Ü N C H E N

TUM-INFO-09-I0103-0/0.-FI

Druck: Institut für Informatik der
Technischen Universität München

Model-Based Image Segmentation Using Local Self-Adapting Separation Criteria

Robert Hanek

Lehrstuhl für Bildverstehen und Wissensbasierte Systeme,
Forschungsgruppe Bildverstehen (FG BV),
Technische Universität München, Germany
http://www9.in.tum.de/people/hanek/

Abstract. In this paper we address the problem of model-based image segmentation by fitting deformable models to the image data. From uncertain a priori knowledge of the model parameters an initial probability distribution of the model edge in the image is obtained. From the vicinity of the surmised edge local statistics are learned for both sides of the edge. These local statistics provide locally adapted criteria to distinguish the two sides of the edge even in the presence of spatially changing properties such as texture, shading, or color. Based on the local statistics the model parameters are iteratively refined using a MAP estimation. Experiments with RGB images show that the method is capable of achieving high subpixel accuracy and robustness even in the presence of texture, shading, clutter, and partial occlusion.

1 Introduction

Deformable models, also known as snakes or active contours [10], 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 [16, 11, 4]. The parameters of the models specify object properties such as the pose, size, and shape. The problem of estimating parameters of curve models from images not only has applications in low-level vision such as image segmentation and tracking but also in high-level vision such as 3-D pose estimation, camera calibration, 3-D reconstruction, and object recognition.

In this paper we propose a novel method for estimating the parameters of deformable edge models from image data. This method can also be applied to high-level problems such as 3-D reconstruction and pose estimation. However, here we focus on model-based image segmentation. (An example for 3-D pose estimation is also given.) Due to the high number of publications on image segmentation in the following only a few aspects of the relevant work can be reflected.

Fig. 1. Mug in front of an inhomogeneous background: **a.)** original color image, **b.)** color edges detected by a gradient-based approach (C. Steger [21]). The mug and the background are not well separated due to the edges within both regions. **c.)** cylinder contour fitted to the image data by the proposed CCD algorithm: despite the inhomogeneity of the mug and the background the estimated contour (black) matches exactly the mug's contour. (red: initialization)

1.1 Previous Work

The body of work on image segmentation can be roughly classified into three categories: (i) **edge-based segmentation**, (ii) **region-based segmentation**, and (iii) **methods integrating edge-based and region-based segmentation**.

(i) **Edge-based segmentation** (which is also referred as boundary-based segmentation) relies on discontinuities of image data. Methods for different edge-profiles, i.e. types of discontinuities, exist (e.g. step-edge [2, 17, 5], roof-edge [2, 17], others [2, 17]). The problem of edge-based segmentation is that in practice usually the edge-profile is not known. Furthermore, the profile often varies heavily along the edge caused by e.g. shading and texture. Due to these difficulties usually a simple step-edge is assumed and the edge detection is performed based on a maximum image gradient. In Fig. 1a the color values on either side of the mug's contour are not constant even within a small vicinity. Hence, methods maximizing the image gradient have difficulties to separate the mug and the background, see Fig. 1b.

(ii) **Region-based segmentation** methods such as [25, 7] rely on the homogeneity of spatially localized features (e.g. RGB values). The underlying homogeneity assumption is that the features of all pixels within one region are statistically independently distributed according to the same probability density function. Contrary to edge-based methods region-based methods do not require an edge-profile. Furthermore, they are able to exploit higher statistical moments of the distributions. Hence, regions which have the same mean feature but different covariances (e.g. caused by texture) can be separated. However often the underlying assumption of a spatially constant probability density function per region does not hold. In Fig. 1a the distributions of the RGB values of the mug and the background depend on the locations within the image.

(iii) **Integrating methods:** especially in recent years methods have been published which aim to overcome the individual shortcomings of edge-based

and region-based segmentation by integrating both segmentation principles [22, 18, 6, 9]. These methods seek a tradeoff between an edge-based criterion, e.g. the magnitude of the image gradient, and a region-based criterion evaluating the homogeneity of the regions. However, it is questionable whether a tradeoff between the two criteria yields reasonable results when both the homogeneity assumption and the assumption regarding the edge profile do not hold as in Fig. 1a.

Model-based methods optimize the fit between the model and the image data. **Global optimization** methods like dynamic programming [1] and Monte Carlo optimization (particle filters, condensation algorithm [4]) are very successfully used (e.g., for tracking). However, dynamic programming requires a discretization of the search space, which leads to a limited accuracy, and particle filters show a very slow convergence if the sensor noise is low [23].

Local optimization methods may achieve a fast, i.e. quadratic, convergence. Approaches aiming to increase the area of convergence such as [13, 24] are edge-based. For methods maximizing the gradient, the area of convergence depends on the window size used to compute the spatial derivatives. Scale-space theory provides means for automatic scale selection [12]. However, blurring the image data eliminates useful high frequency information. Several segmentation methods integrate different image cues such as texture and color or brightness [3, 14, 15, 22].

1.2 Main Contributions

The main contributions of this paper are as follows:

1.) Local self-adapting separation criteria are used in order to distinguish adjacent regions: while other methods use certain fixed criteria (e.g. image gradients, homogeneity criteria, or combinations) to separate adjacent regions, we use local self-adapting separation criteria. These criteria are based on local statistics of pixel features obtained from the vicinity of the surmised curve. These criteria allow to separate the two sides of an edge even in the presence of spatially changing properties, such as changing texture, color, or shading. For the computation of the required local statistics an efficient method is proposed.

2.) A fit between image data and a 'blurred model' is proposed: 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 probability distribution of model parameters. The advantages are as follows: (i) the capture range is enlarged according to the local uncertainties of the model curve which significantly improves the convergence. (ii) Optimizing the fit between an image and a 'blurred model' is in general computationally cheaper than blurring the image. (iii) High frequency information of the image data can be used.

Overview of the paper: the reminder of this paper is organized as follows: in section 2 an overview of the here 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 in section 6 a conclusion is given.

2 Overview of the Contracting Curve Density (CCD) Algorithm

The here proposed CCD algorithm estimates the parameters of curve models from image data. The CCD algorithm can roughly be characterized as an extension of the EM algorithm [8] using additional knowledge. The additional knowledge consists of: (i) a curve model, which describes the set of possible boundaries between adjacent regions, and (ii) a model of the imaging process. The CCD algorithm, depicted in Fig. 2, performs the iteration of two steps, which roughly correspond to the two steps of the EM algorithm:

1. Local statistics of image data are learned from the vicinity of the curve. These statistics locally characterize the two sides of the edge curve. **2.** From these statistics the **estimation of the model parameters is refined** by optimizing the separation of the two sides. This refinement in turn leads in the next iteration step to an improved statistical characterization of the two sides. During the process the uncertainty of the model parameters decreases. Thereby, the probability density of the curve in the image contracts to a single edge estimate. Therefore, we call the algorithm Contracting Curve Density (CCD) algorithm.

Input: the input of the CCD algorithm consists of the image data $\mathbf{I}^*$ and the curve model. The image data are local features, e.g. RGB values, given for each pixel of the image. The curve model consists of two parts: 1.) a differentiable curve function c describing the model edge curve in the image as a function of the model parameters $\boldsymbol{\Phi}$, 2.) a Gaussian a priori distribution $p(\boldsymbol{\Phi}) = p(\boldsymbol{\Phi} \mid \mathbf{m}_{\boldsymbol{\Phi}}^*, \boldsymbol{\Sigma}_{\boldsymbol{\Phi}}^*)$ of the model parameters $\boldsymbol{\Phi}$, defined by the mean $\mathbf{m}_{\boldsymbol{\Phi}}^*$ and the covariance $\boldsymbol{\Sigma}_{\boldsymbol{\Phi}}^*$. (The superscript $*$ indicates input data.) Depending on the application the quantities $\mathbf{m}_{\boldsymbol{\Phi}}^*$ and $\boldsymbol{\Sigma}_{\boldsymbol{\Phi}}^*$ may be obtained for example from a training set, by a human initialization, or from a prediction over time.

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

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

3 Learn Local Statistics (Step 1)

In section 3.1 the vicinity of the surmised image curve is computed. For pixels within the vicinity local statistics of the image data are computed in section 3.2

Contracting Curve Density (CCD) algorithm

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

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

repeat

1. **learn local statistics** of image data from the vicinity of the curve
   (a) compute pixels v in vicinity $\mathcal{V}$ of the image curve from c, $\mathbf{m}_{\mathbf{\Phi}}$ and $\mathbf{\Sigma}_{\mathbf{\Phi}}$
   $\forall v \in \mathcal{V}$ compute vague assignment $\mathbf{a}_v(\mathbf{m}_{\mathbf{\Phi}}, \mathbf{\Sigma}_{\mathbf{\Phi}})$ to the sides of the curve
   (b) $\forall v \in \mathcal{V}$ compute local statistics $\mathbf{S}_v$ of image data $\mathbf{I}_{\mathcal{V}}^*$
2. **refine estimation** of model parameters
   (a) update mean $\mathbf{m}_{\mathbf{\Phi}}$ by performing one iteration step of MAP estimation:

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

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

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

until changes of $\mathbf{m}_{\mathbf{\Phi}}$ and $\mathbf{\Sigma}_{\mathbf{\Phi}}$ are small enough

Post-processing: estimate covariance $\mathbf{\Sigma}_{\mathbf{\Phi}}$ from Hessian of $\chi^2(\mathbf{m}_{\mathbf{\Phi}})$

return mean $\mathbf{m}_{\mathbf{\Phi}}$ and covariance $\mathbf{\Sigma}_{\mathbf{\Phi}}$

Fig. 2. The CCD algorithm iteratively refines a Gaussian a priori density $p(\mathbf{\Phi}) = p(\mathbf{\Phi} \mid \mathbf{m}_{\mathbf{\Phi}}^*, \mathbf{\Sigma}_{\mathbf{\Phi}}^*)$ of model parameters to a Gaussian approximation $p(\mathbf{\Phi} \mid \mathbf{m}_{\mathbf{\Phi}}, \mathbf{\Sigma}_{\mathbf{\Phi}})$ of the posterior density $p(\mathbf{\Phi} \mid \mathbf{I}^*)$.

3.1 Compute Vicinity of the Image Curve (Step 1a)

The Gaussian distribution of model parameters $p(\mathbf{\Phi} \mid \mathbf{m}_{\mathbf{\Phi}}, \mathbf{\Sigma}_{\mathbf{\Phi}})$ and the model curve function c define a probability distribution of the edge curve in the image. This curve distribution vaguely assigns pixels in the vicinity of the surmised curve to one side of the curve. We define the assignment $\mathbf{a}_v(\mathbf{m}_{\mathbf{\Phi}}, \mathbf{\Sigma}_{\mathbf{\Phi}}) = (a_{v,1}(\mathbf{m}_{\mathbf{\Phi}}, \mathbf{\Sigma}_{\mathbf{\Phi}}), a_{v,1}(\mathbf{m}_{\mathbf{\Phi}}, \mathbf{\Sigma}_{\mathbf{\Phi}}))^T$ of a pixel v as a function of the mean $\mathbf{m}_{\mathbf{\Phi}}$ and covariance $\mathbf{\Sigma}_{\mathbf{\Phi}}$. 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 $a_{v,2} = 1 - a_{v,1}$. Here we model a standard CCD imagine device, which integrates the radiance function over the photosensitive elements of the pixel [2]. The value $a_{v,1}$ denotes the fraction of the photosensitive element which belongs on average to side 1.

By $\mathcal{V}_u$ we denote the set of possible edge pixels v, i.e. pixels which are not clearly assigned to one side. For these pixels $v \in \mathcal{V}_u$ the assignment must be refined. By $\mathcal{V}_s$ we denote the set of pixels which are quite surely assigned to one side 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 surmised curve is determined. For all pixels v in $\mathcal{V}$ the vague assignments $\mathbf{a}_v$ are computed. Fig. 5 row b depicts the components $a_{v,1}$ of the assignments for different iterations.

Furthermore, local curve coordinates $(d_v, d_v^\perp)^T$ are determined, which are better suited to describe the vicinity of the curve than image coordinates. The first component d_v is the displacement (signed distance) to the mean curve, i.e. the curve specified by the mean parameter vector $\mathbf{m_\Phi}$. The second component $d_v^\perp$ denotes the perpendicular coordinate, the position of v along the curve. All pixels in $\mathcal{V}$ are sorted according to the coordinate $d_v^\perp$, which allows a more efficient computation in the following step 1b. Exploiting the fact that adjacent pixels have similar coordinates this sorting can be done efficiently.

3.2 Compute Local Statistics (Step 1b)

For the two sides separated by the curve local statistics of the image features are learned from the already quite surely assigned pixels. We assume that the local statistical properties of both sides may vary along the curve. This assumption is related to the approach of Ronfard [20]. The author partitions the vicinity of the curve into stripes perpendicular to the curve and assumes within a stripe constant statistics for each side of the curve.

In order to avoid the discretization involved in this approach, we model the side statistics as a function of the position along the curve.[1] For each coordinate $d_v^\perp$ with $v \in \mathcal{V}$ two sets of pixels are used in order to compute local statistics $\mathbf{S}(d_v^\perp)$ of the two sides. The pixels p within such a set are weighted by heuristic weights $w_{p,s}(d_v^\perp)$. The index $s \in \{1,2\}$ indicates one of the two sides of the curve. Fig. 3 depicts the weights $w_{p,1}(d_v^\perp)$ and $w_{p,2}(d_v^\perp)$ for different pixels p and a fixed curve coordinate $d_v^\perp$. (Pixel v is the pixel for which characteristic statistic $\mathbf{S}(d_v^\perp)$ should be learned.) Pixels p which are too close to the surmised curve or pixels which are on the wrong side have a weight equal or close to zero. Pixels which are too far from the surmised curve also have a low weight. Furthermore, the weight decreases exponentially with the distance $\Delta d^\perp(v,p) = \mid d_v^\perp - d_p^\perp \mid$, the distance along the curve between the pixels v and p. The size of the window adapts to the uncertainty of the curve, i.e. for a big uncertainty the window is wider than for a small uncertainty. Compare e.g. the left and the right side in Fig. 3. Row c in Fig. 5 depicts the weights of one side for different iterations. Contrary to Fig. 3, in Fig. 5 the weights are not depicted for a fixed position $d_v^\perp$ along the curve but for the position $d_p^\perp$ which yields the maximum weight.

The statistics $\mathbf{S}_v = \mathbf{S}(d_v^\perp)$ contain for both sides $s \in \{1,2\}$ the first and second order moments of the image feature vectors $\mathbf{I}_p^*$ weighted by $w_{p,s}(d_v^\perp)$.[2] The time complexity of the computation of $\mathbf{S}_v$ for a single pixel v is $O(N_v)$ where N_v denotes the number of pixels p with a non-zero weight $w_{p,s}(d_v^\perp)$. An independent computation of $\mathbf{S}_v$ for all uncertainly assigned pixels $v \in \mathcal{V}_u$ would result in a time complexity of $O(\sum_{v \in \mathcal{V}_u} N_v)$. We compute simultaneously the statistics $\mathbf{S}_v$ for all $v \in \mathcal{V}_u$ with a time complexity of $O(|\mathcal{V}|)$ where $|\mathcal{V}|$ is the number of pixels in the vicinity $\mathcal{V}$ of the curve computed in step 1a. This liner time complexity is possible due to the sorting of pixels v in $\mathcal{V}$ according to $d_v^\perp$ (see step 1a)

[1] In addition, the proposed approach differs from Ronfard's approach in the statistics used to separate the two sides and in the way the statistics are obtained.

[2] In future work we might investigate the usability of other statistical models.

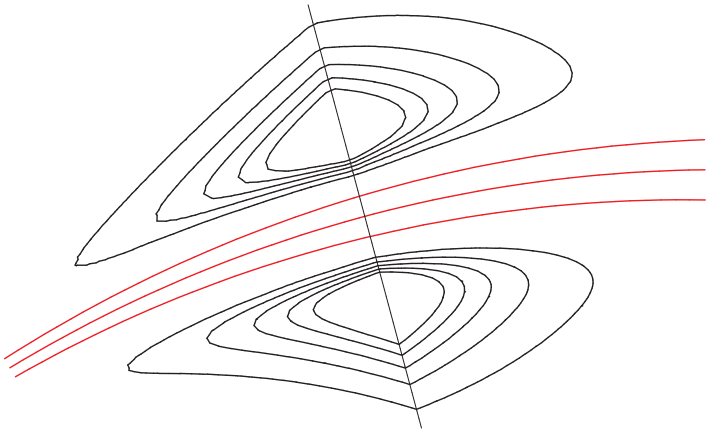

Fig. 3. Contour plot of the windows (weights) used to compute local statistics: the three roughly parallel lines (red) describe the surmised position and uncertainty (σ-interval) of the curve. For the pixels on the perpendicular line (straight black) local statistics $\mathbf{S}_v$ are computed from the two depicted windows. The windows are adapted in size and shape to the surmised curve and its uncertainty.

and the choice of the weights $w_{p,s}(d_v^\perp)$. The weights depend exponentially on the distance $\Delta d^\perp(v,p)$ along the curve which allows for a recursive computation and thereby yields a significant speedup.

4 Refine the Estimation of Model Parameters (Step 2)

In this step the estimation of the model parameters is refined using the vague pixel assignments $\mathbf{a}_v(\mathbf{m}_\mathbf{\Phi}, \mathbf{\Sigma}_\mathbf{\Phi})$ and the local statistics $\mathbf{S}_v$ obtained in step 1.

4.1 Observation Model

First we describe our observation model, the assumed probabilistic relation between the model parameters $\mathbf{\Phi}$ and the image data $\mathbf{I}$. We model the feature vector $\mathbf{I}_v$ of a pixel $v \in \mathcal{V}$ as a weighted sum of two random variables $\mathbf{I}_{v,1}$ and $\mathbf{I}_{v,2}$:

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

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

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. The quantity $\tilde{a}_{v,1}$ describes which fraction of the

photosensitive area of pixel v lies at side 1. The linear relation assumed in (1) corresponds to the sensor model used by Baker et al. [2]. The authors model the spatial discretization of the imaging device by considering a sub-rectangle of each sensor pixel. We use equation (1) for RGB values. For other features, given in different color spaces (e.g. YUV) or texture descriptors, equation (1) may be substituted by a non-linear relation.

The local statistics $\mathbf{S}_v$ obtained in step 1b contain the first and second local moments of the two sides of the curve. The first moments are the local mean vectors $\mathbf{m}_{v,1}$ and $\mathbf{m}_{v,2}$. From the mean vectors and the second moments the local covariances $\mathbf{\Sigma}_{v,1}$ and $\mathbf{\Sigma}_{v,2}$ can be obtained. We model the distributions of the random variables $\mathbf{I}_{v,1}$ and $\mathbf{I}_{v,2}$ by two Gaussian distributions. Their mean vectors and covariances are estimated by the mean vectors ($\mathbf{m}_{v,1}$ and $\mathbf{m}_{v,2}$) and covariances ($\mathbf{\Sigma}_{v,1}$ and $\mathbf{\Sigma}_{v,2}$) learned from the vicinity of pixel v. Despite Gaussian distributions are used, also regions with local multi-modal distributions can be separated, see the experiments. Due to the linear relation in (1), we obtain for the feature vector $\mathbf{I}_v$ a Gaussian probability density $p(\mathbf{I}_v \mid \mathbf{m}_v, \mathbf{\Sigma}_v)$. The estimates of the mean vector $\mathbf{m}_v$ and covariance $\mathbf{\Sigma}_v$ are given by

$$\mathbf{m}_v = \tilde{a}_{v,1}\mathbf{m}_{v,1} + \tilde{a}_{v,1}\mathbf{m}_{v,2} \tag{3}$$

$$\mathbf{\Sigma}_v = \tilde{a}_{v,1}\mathbf{\Sigma}_{v,1} + \tilde{a}_{v,1}\mathbf{\Sigma}_{v,2}. \tag{4}$$

For the sake of an intuitive notation we define

$$p(\mathbf{I}_v \mid \tilde{\mathbf{a}}_v, \mathbf{S}_v) := p(\mathbf{I}_v \mid \mathbf{m}_v, \mathbf{\Sigma}_v). \tag{5}$$

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

$$p(\mathbf{I}_\mathcal{V} \mid \tilde{\mathbf{a}}_\mathcal{V}(\mathbf{\Phi}), \mathbf{S}_\mathcal{V}) = \prod_{v \in \mathcal{V}} p(\mathbf{I}_v \mid \tilde{\mathbf{a}}_v(\mathbf{\Phi}), \mathbf{S}_v). \tag{6}$$

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 (6) feature vectors of pixels on the same side of the curve are modeled as correlated. The intensity of the correlation depends on the overlap of the windows. Equation (6) 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)

MAP Estimation: using the observation density derived in (6) the MAP estimate $\widehat{\mathbf{\Phi}}$ of the model parameters $\mathbf{\Phi}$ can be written as

$$\widehat{\mathbf{\Phi}} = \arg\max_{\mathbf{\Phi}} \; F(\mathbf{\Phi}) \quad \text{with} \tag{7}$$

$$F(\mathbf{\Phi}) = p(\mathbf{I}_{\mathcal{V}} = \mathbf{I}^*_{\mathcal{V}} \mid \tilde{\mathbf{a}}_{\mathcal{V}}(\mathbf{\Phi}), \mathbf{S}_{\mathcal{V}}) \cdot p(\mathbf{\Phi} \mid \mathbf{m}^*_{\mathbf{\Phi}}, \mathbf{\Sigma}^*_{\mathbf{\Phi}}) \tag{8}$$

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

$$\widehat{\mathbf{\Phi}} = \arg\min_{\mathbf{\Phi}} \ \tilde{\chi}^2(\mathbf{\Phi}) \quad \text{with} \tag{9}$$

$$\tilde{\chi}^2(\mathbf{\Phi}) = -2 \ln[p(\mathbf{I}_{\mathcal{V}} = \mathbf{I}^*_{\mathcal{V}} \mid \tilde{\mathbf{a}}_{\mathcal{V}}(\mathbf{\Phi}), \mathbf{S}_{\mathcal{V}}) \cdot p(\mathbf{\Phi} \mid \mathbf{m}^*_{\mathbf{\Phi}}, \mathbf{\Sigma}^*_{\mathbf{\Phi}})]. \tag{10}$$

Due to the nature of an imaging sensor the optimization in (9) 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 (influences) $\tilde{a}_{v,1}$ are equal to 1 or 0. Only for the edge pixels the partial derivatives of $\tilde{a}_{v,1}$ for the model parameters $\mathbf{\Phi}$ are not zero. The objective function $\tilde{\chi}^2$ typically has points with high curvature or even points of discontinuity, see Fig. 4. Hence, gradient-descent optimization methods such as Newton-Iteration have a very limited area of convergence.

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. [10]. 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 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 $\tilde{\mathbf{a}}_v(\mathbf{\Phi})$. We substitute $\tilde{\mathbf{a}}_v(\mathbf{\Phi})$ by the vague assignment $\mathbf{a}_v(\mathbf{m}_{\mathbf{\Phi}}, \mathbf{\Sigma}_{\mathbf{\Phi}})$ obtained in step 1a. While $\tilde{\mathbf{a}}_v(\mathbf{\Phi})$ is the assignment for a single vector of model parameters, $\mathbf{a}_v(\mathbf{m}_{\mathbf{\Phi}}, \mathbf{\Sigma}_{\mathbf{\Phi}})$ is the expectation of the assignment for a Gaussian distribution of model parameters. If the mean $\mathbf{m}_{\mathbf{\Phi}}$ approaches $\mathbf{\Phi}$ and the covariance $\mathbf{\Sigma}_{\mathbf{\Phi}}$ approaches the zero matrix then $\mathbf{a}_v(\mathbf{m}_{\mathbf{\Phi}}, \mathbf{\Sigma}_{\mathbf{\Phi}})$ approaches $\tilde{\mathbf{a}}_v(\mathbf{\Phi})$. For a non-zero covariance $\mathbf{\Sigma}_{\mathbf{\Phi}}$ the function $\mathbf{a}_v(\cdot, \mathbf{\Sigma}_{\mathbf{\Phi}})$ is a smooth approximation of $\tilde{\mathbf{a}}_v(\cdot)$. After each iteration the norm of the covariance matrix is reduced in step 2b and thereby the smoothing is reduced. Fig. 4 illustrates the resulting objective function for different uncertainties (levels of smoothing).

We interpret the estimate $\widehat{\mathbf{\Phi}}$ of the model parameters $\mathbf{\Phi}$ as the mean $\mathbf{m}_{\mathbf{\Phi}}$ of a Gaussian approximation of the posterior distribution. Therefore, in the

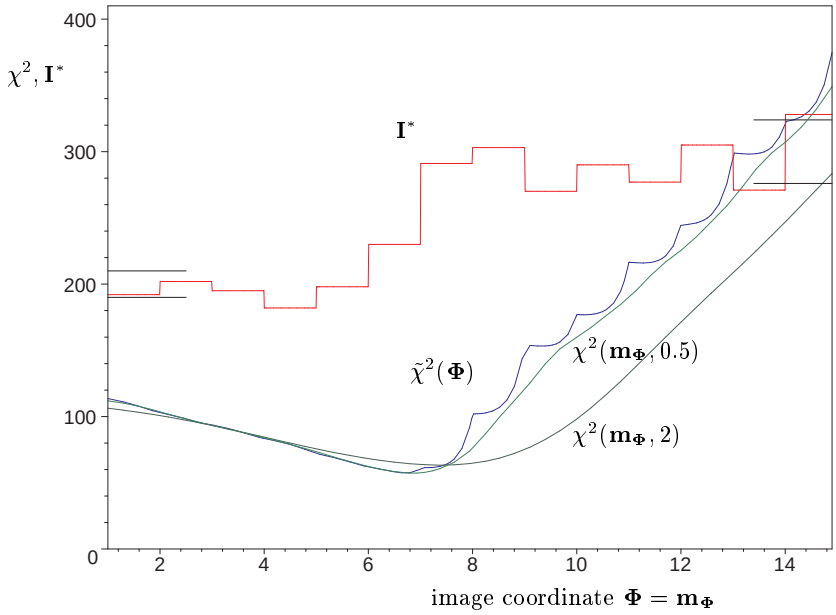

Fig. 4. Detection of an edge in an 1-D array of gray values $\mathbf{I}^*$: the objective function $\tilde{\chi}^2(\mathbf{\Phi})$ which assesses the edge hypothesis $\mathbf{\Phi}$ is non-differentiable. The smooth approximations $\chi^2(\mathbf{m_\Phi}, \mathbf{\Sigma_\Phi})$, depicted for $\mathbf{\Sigma_\Phi} = 0.5$ and $\mathbf{\Sigma_\Phi} = 2$, 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.

following we denote the estimate of the model parameters by $\mathbf{m_\Phi}$. With this denotation and the substitution of $\tilde{\mathbf{a}}_v(\cdot)$ by $\mathbf{a}_v(\cdot, \mathbf{\Sigma_\Phi})$ the estimate $\mathbf{m_\Phi}$ of the model parameters $\mathbf{\Phi}$ can be written as

$$\mathbf{m_\Phi} = \arg\min_{\mathbf{m_\Phi}} \chi^2(\mathbf{m_\Phi}) \quad \text{with} \tag{11}$$

$$\chi^2(\mathbf{m_\Phi}) = -2\ln[p(\mathbf{I}_\mathcal{V} = \mathbf{I}^*_\mathcal{V} \mid \mathbf{a}_\mathcal{V}(\mathbf{m_\Phi}, \mathbf{\Sigma_\Phi}), \mathbf{S}_\mathcal{V}) \cdot p(\mathbf{m_\Phi} \mid \mathbf{m}^*_\mathbf{\Phi}, \mathbf{\Sigma}^*_\mathbf{\Phi})]. \tag{12}$$

The term $-2\ln p(\mathbf{I}_\mathcal{V} = \mathbf{I}^*_\mathcal{V} \mid \mathbf{a}_\mathcal{V}(\mathbf{m_\Phi}, \mathbf{\Sigma_\Phi}), \mathbf{S}_\mathcal{V})$ evaluates the fit between the sensed image data $\mathbf{I}^*_\mathcal{V}$ and the Gaussian distribution of model parameters with mean $\mathbf{m_\Phi}$ and covariance $\mathbf{\Sigma_\Phi}$. The term $-2\ln p(\mathbf{m_\Phi} \mid \mathbf{m}^*_\mathbf{\Phi}, \mathbf{\Sigma}^*_\mathbf{\Phi})$ evaluates the fit between the estimate $\mathbf{m_\Phi}$ 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 $-2\ln p(\mathbf{I}_\mathcal{V} = \mathbf{I}^*_\mathcal{V} \mid \mathbf{a}_\mathcal{V}(\mathbf{m_\Phi}, \mathbf{\Sigma_\Phi}), \mathbf{S}_\mathcal{V})$ to be optimized.

Fig. 5 row d depicts the image data expected by the blurred model, i.e the image optimizing the blurred observation density $p(\mathbf{I}_\mathcal{V} \mid \mathbf{a}_\mathcal{V}(\mathbf{m_\Phi}, \mathbf{\Sigma_\Phi}), \mathbf{S}_\mathcal{V})$. 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.

Fig. 5 row f depicts the energy $\chi_v^2 = -2 \ln p(\mathbf{I}_v = \mathbf{I}_v^* \mid \mathbf{a}_v(\mathbf{m_\Phi}, \mathbf{\Sigma_\Phi}), \mathbf{S}_v)$ 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 $\mathbf{m_\Phi}$ 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 forces. Step 2a of the CCD algorithm seeks to find the estimate $\mathbf{m_\Phi}$ which yields an equilibrium between the pixel forces. Fig. 5 row g depicts the forces acting in the direction perpendicular to the surmised curve. Pixels depicted as bright act in a direction opposite to the pixels depicted as dark.

The area of pixels which influence the position of the surmised curve depends on the uncertainty of the curve in the image. In Fig. 5 the initial uncertainty of the curve is big 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 clearly lower forces than pixels in the narrow area. Furthermore, pixels which are close to the surmised curve exert on average stronger forces than pixels which are further apart.

The used assignments of pixels to sides of the curve are vague and adjacent sides compete simultaneously for all uncertainly assigned pixels. This allows to bridge gaps with insignificant or even misleading statistics and thereby increases the area of convergence.

4.3 Update the Covariance (Step 2b)

In step 2b the covariance $\mathbf{\Sigma_\Phi}$ describing the uncertainty of the estimate $\mathbf{m_\Phi}$ 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 $\mathbf{\Sigma_\Phi}$ of the χ^2-estimate $\mathbf{m_\Phi}$ is given by

$$\mathbf{\Sigma_\Phi} = \frac{2}{H} \tag{13}$$

where H denotes the Hessian of $\chi^2(\mathbf{m_\Phi})$ in the point $\mathbf{m_\Phi}$ minimizing $\chi^2(\mathbf{m_\Phi})$ [19]. Since the relation between the model parameters and the image data is not linear, equation (13) can only be regarded as an estimate of the uncertainty. Another problem arises by the way the estimate $\mathbf{m_\Phi}$ is obtained. In step 2a only a single iteration step is performed. Hence, the resulting estimate $\mathbf{m_\Phi}$ does not necessarily minimize the objective function χ^2. Therefore, we update the covariance $\mathbf{\Sigma_\Phi}$ by the following heuristics:

$$\mathbf{\Sigma_\Phi} := c_2 \mathbf{\Sigma_\Phi} + c_1 \frac{2}{H}. \tag{14}$$

Here $c_2\mathbf{\Sigma_\Phi}$ 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 (13). However, here H is the Hessian in the point $\mathbf{m_\Phi}$, which not necessarily exactly maximizes $\chi^2(\mathbf{m_\Phi})$. The parameter c_2, 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. The parameter c_1 is already used in the initialization and increases the area of convergence especially in the presence of strong non-linearities. At the beginning of the iteration the covariance $\mathbf{\Sigma_\Phi}$ usually decreases roughly exponentially. However, for a deteriorating estimate $\mathbf{m_\Phi}$, i.e. an estimate $\mathbf{m_\Phi}$ which is not as good as assumed by $\mathbf{\Sigma_\Phi}$, the curvature (Hessian) of the objective function decreases, see Fig. 4. This slows down the reduction of $\mathbf{\Sigma_\Phi}$ or even increases $\mathbf{\Sigma_\Phi}$. Hence, the chance that the iteration still converges is increased. The steps 1 and 2 of the CCD algorithm are iterated until the changes of the estimates $\mathbf{m_\Phi}$ and the associated covariance $\mathbf{\Sigma_\Phi}$ are small enough.

Post-Processing: after the iteration the covariance of the estimate $\mathbf{m_\Phi}$ is estimated by $\mathbf{\Sigma_\Phi} := \frac{2}{H}$. Finally the estimate $\mathbf{m_\Phi}$ of the model parameters and the estimate $\mathbf{\Sigma_\Phi}$ of the corresponding covariance are returned.

5 Experiments

In our experiments we apply the proposed CCD algorithm to the segmentation of two fundamental types of image features, namely (i) lines which are radially distorted and (ii) circles.

(i) Lines: a line is regarded as the radially distorted image of a straight line (without endpoints). We use a calibrated camera. Hence, the transformation from undistorted to distorted coordinates is known. The model describing the set of all possible radially distorted lines has two parameters, 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 re-sampling 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. Fig. 5 row a.) 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%.

Fig. 6 contains real image data. The mug has strong internal edges. Shading causes additional variations of the mug's color values. The background contains area of texture as well as strong variations of the illumination. After 11 iterations the estimated curve is aligned to the real curve without any visible deviation.

(ii) Circles: we also apply the CCD algorithm to the segmentation of circles parameterized by the center point and the radius. Figs. 7 and 8 depict the iteration 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

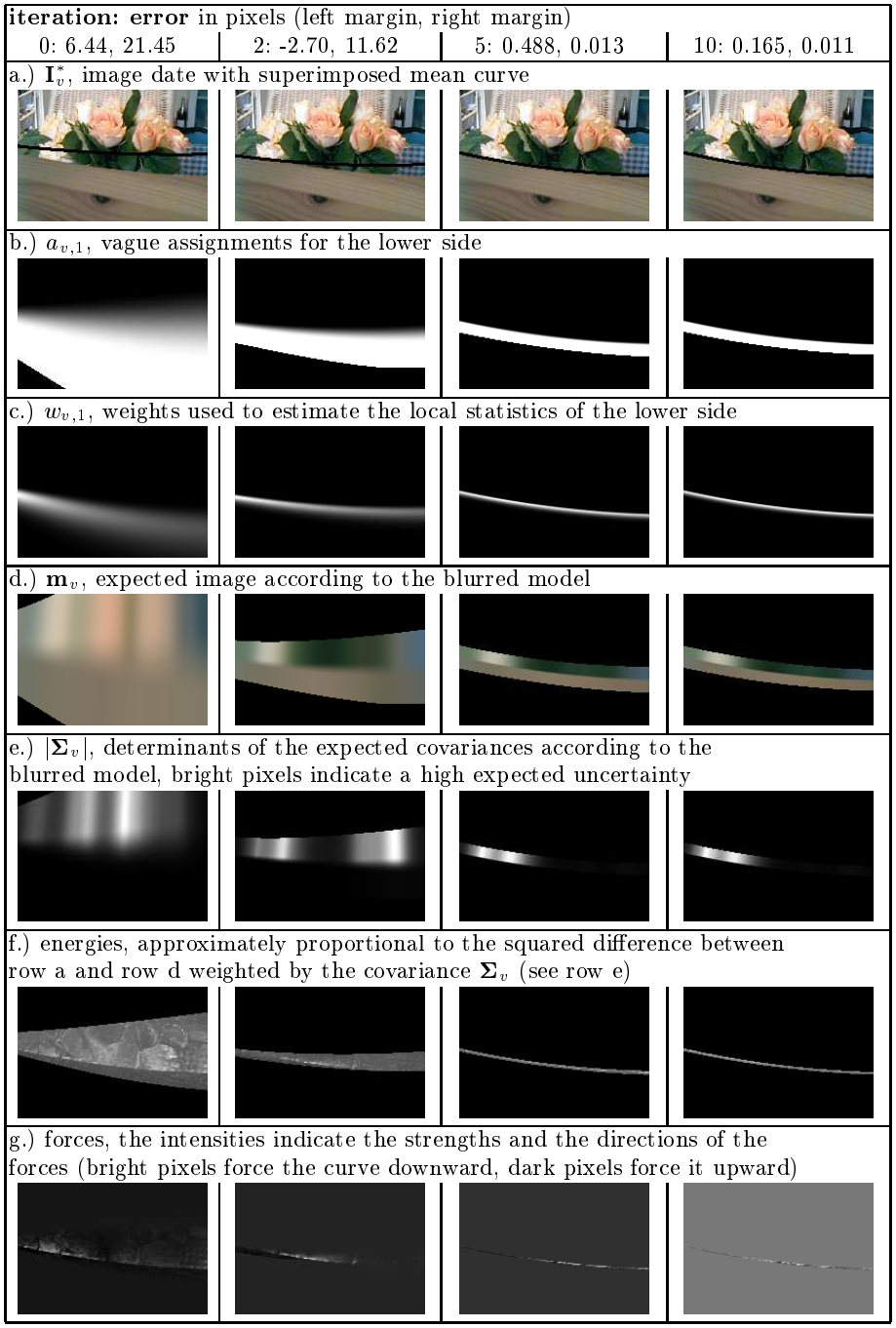

Fig. 5. Semi-synthetic image showing roses behind a wooden board: the initial error is reduced by more than 99%. See text for a detailed description. (The images, except rows a and d, are individually normalized such that the grey values are within [0,255].)

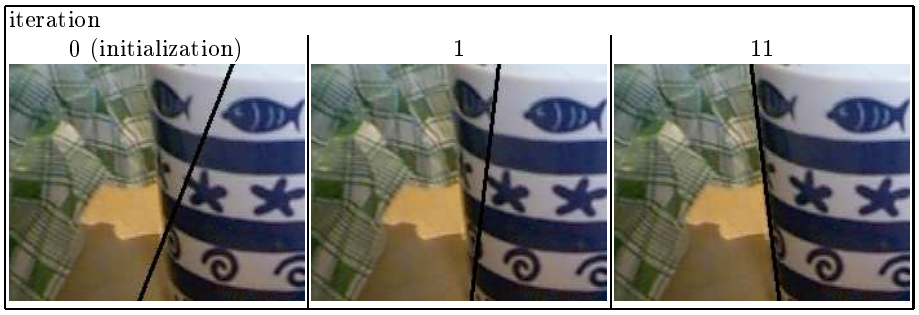

Fig. 6. Real image, (sub-image of Fig. 1a.): the initial error of the curve is more than 70 pixels (upper part). After 11 iterations the deviation between the estimated curve and the real curve is not visible.

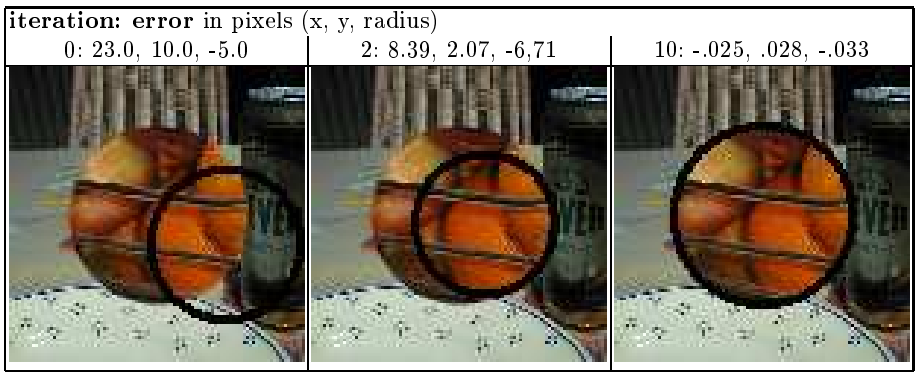

Fig. 7. Despite the inhomogeneity of the foreground and the background the initial error is reduced by more than 99.8%.

real images with similarly complex content we assume that the subpixel accuracy is lower due to different effects, such as unknown blurring caused by the imaging device.

Fig. 9 depicts the area of convergence for the image shown in Fig. 7. The CCD algorithm is started with a radius which is 5 pixels higher than the correct radius. A high initial covariance, an uninformed a priori distribution, is used leading to a big area of convergence. For a small initial uncertainty the image processing operations are focused to a small number of pixels. Hence, the area of convergence is usually smaller. However, for a better initialization only a smaller area of convergence is needed.

The run time of the algorithm scales roughly linearly to the number of used pixels. Hence, the initial uncertainty has an important impact on the run time. For example on a 500 MHz computer the first iteration step of Fig. 7 takes about

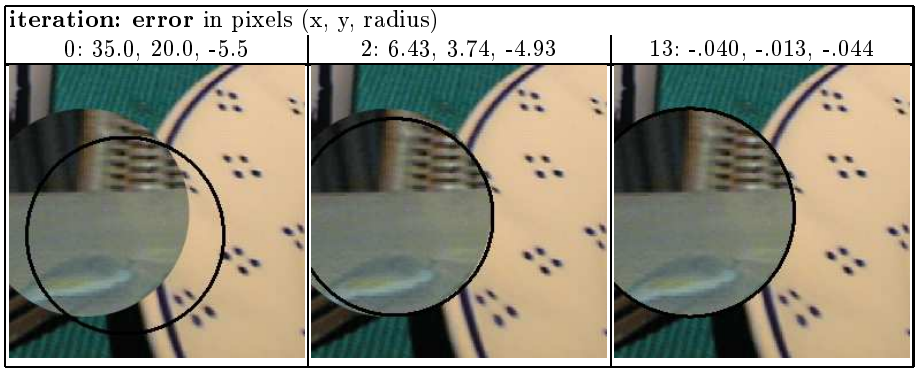

Fig. 8. The circle is only partially visible. Nevertheless, the final error is less than 5% of a pixel.

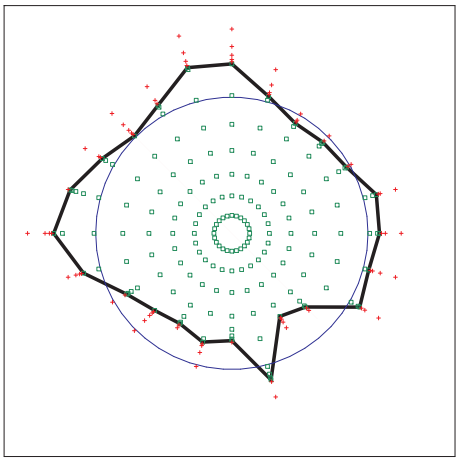

Fig. 9. Area of convergence for the image depicted in Fig. 7. The thick line separates all investigated converging center points (green) from the non-converging center points (red). The thin blue line is the real image circle. The area of convergence has roughly the size of the real circle.

4s. The initial uncertainty is big and roughly 10,000 pixels are used. After about 5 iterations the run time is reduced to less than 1s per iteration.

Cylinder contours: finally we apply the proposed CCD algorithm to the pose estimation problem. The 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. These five pose parameters specify the contour of the cylinder which is fitted to the image data. Fig. 1c depicts the initial

contour and the estimated contour. Despite the inhomogeneity of the mug and the background the estimated contour matches exactly the mug's contour.

6 Conclusion

We have proposed a novel method for fitting deformable models to image data. The method iteratively refines the a priori distribution of the model parameters to a Gaussian approximation of the posterior distribution.

Locally adapted criteria are used in order to distinguish the two sides of the curve. The separation criteria are learned 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 to separate the two sides of the curve with high subpixel accuracy even in the presence of texture, shading, clutter, and partial occlusion.

High robustness, a big area of convergence, is achieved by optimizing the resulting MAP criteria not only for a single vector of model parameters but for a distribution of model parameters. During the process not only the model parameters are refined but also the associated covariance. From the covariance the local uncertainties of the curve are obtained which provide the basis for the automatic and local scale selection.

In future work we will apply the proposed CCD algorithm to other model-fitting problems such as 3-D pose estimation, 3-D reconstruction, and object tracking.

Acknowledgments: the author would like to thank Michael Beetz, Carsten Steger, and Viswanathan Ramesh for useful discussions during the process of this work.

References

1. Amini, A., Weymouth, T., and Jain, R. Using dynamic programming for solving variational problems in vision. *PAMI 12*, 9 (September 1990), 855–867.
2. Baker, S., Nayar, S., and Murase, H. Parametric feature detection. *IJCV 27*, 1 (March 1998), 27–50.
3. Belongie, S., Carson, C., Greenspan, H., and Malik, J. Color- and texture-based image segmentation using the expectation-maximization algorithm and its application to content-based image retrieval. In *ICCV98* (1998), pp. 675–682.
4. Blake, A., and Isard, M. *Active Contours*. Springer-Verlag, Berlin Heidelberg New York, 1998.
5. Canny, J. A computational approach to edge detection. *PAMI 8*, 6 (November 1986), 679–698.
6. Chakraborty, A., and Duncan, J. Game-theoretic integration for image segmentation. *PAMI 21*, 1 (January 1999), 12–30.
7. Chesnaud, C., Refregier, P., and Boulet, V. Statistical region snake-based segmentation adapted to different physical noise models. *PAMI 21*, 11 (November 1999), 1145–1157.

8. Dempster, A., Laird, N., and Rubin, D. Maximum likelihood from incomplete data via the EM algorithm. *J. R. Statist. Soc. B 39* (1977), 1–38.
9. Jones, T., and Metaxas, D. Image segmentation based on the integration of pixel affinity and deformable models. In *CVPR98* (1998), pp. 330–337.
10. Kass, M., Witkin, A., and Terzopoulos, D. Snakes: Active contour models. *IJCV 1*, 4 (January 1988), 321–331.
11. Leventon, M., Grimson, W., and Faugeras, O. Statistical shape influence in geodesic active contours. In *CVPR00* (2000), pp. I:316–323.
12. Lindeberg, T. Feature detection with automatic scale selection. *IJCV 30*, 2 (November 1998), 79–116.
13. Luo, H., Lu, Q., Acharya, R., and Gaborski, R. Robust snake model. In *CVPR00* (2000), pp. I:452–457.
14. Malik, J., Belongie, S., Shi, J., and Leung, T. Textons, contours and regions: Cue integration in image segmentation. In *ICCV99* (1999), pp. 918–925.
15. Manduchi, R. Bayesian fusion of color and texture segmentations. In *ICCV99* (1999), pp. 956–962.
16. McInerney, T., and Terzopoulos, D. Deformable models in medical image analysis: a survey. *Medical Image Analysis 1*, 2 (1996), 91–108.
17. Nalwa, V., and Binford, T. On detecting edges. *PAMI 8*, 6 (November 1986), 699–714.
18. Paragios, N., and Deriche, R. Coupled geodesic active regions for image segmentation: A level set approach. In *ECCV00* (2000), pp. 224–240.
19. Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. *Numerical Recipes in C*. Cambridge University Press, Cambridge, 1996.
20. Ronfard, R. Region-based strategies for active contour models. *IJCV 13*, 2 (October 1994), 229–251.
21. Steger, C. Subpixel-precise extraction of lines and edges. *International Archives of Photogrammetry and Remote Sensing XXXIII, part B3* (2000), 141–156.
22. Thirion, B., Bascle, B., Ramesh, V., and Navab, N. Fusion of color, shading and boundary information for factory pipe segmentation. In *CVPR00* (2000), pp. II:349–356.
23. Thrun, S., Fox, D., and Burgard, W. Monte carlo localization with mixture proposal distribution. In *Proc. of the AAAI National Conference on Artificial Intelligence* (2000), pp. 859–865.
24. Xu, C., and Prince, J. Snakes, shapes, and gradient vector flow. *IP 7*, 3 (March 1998), 359–369.
25. Zhu, S., and Yuille, A. Region competition: Unifying snakes, region growing, and bayes/mdl for multiband image segmentation. *PAMI 18*, 9 (September 1996), 884–900.