Vickers Image Segmentation Using Active Contours

Masterarbeit

zur Erlangung des Diplomgrades
an der Naturwissenschaftlichen Fakultät der
Paris-Lodron Universität Salzburg

eingereicht von
Michael Gadermayr, B.Eng.

Betreuer:
Univ.-Prof. Mag. Dr. Andreas Uhl

Department of Computer Sciences
University of Salzburg
Jakob Haringer Str. 2
5020 Salzburg, Austria

Salzburg, im Februar 2012

Abstract

Hardness is an important characteristic of solid materials (e.g. metal), expressing its resistance to permanent deformation. In order to measure the mentioned property, several approaches exist (e.g. Vickers, Brinell).

The basic idea of all approaches is to measure the deformation of a material, produced by a standardized indenter. In a first step, a standardized indenter is pressed into the material with a defined (known) force, which leads to a deformation. To be able to provide a hardness measure, in a second step, precise gauging of respective deformation (indentation) is done. Knowing the size of the indentation as well as the pressure force, the hardness of the material can be accurately determined.

We especially concentrate on Vickers hardness testing. In this approach, a square based pyramidal indenter is pressed into the material. After mechanical deformation, a picture is taken (using a microscope), which shows an approximately square shape. The size of the shape depends on the force and the hardness of the material. In order to calculate hardness of the material, both diagonals of the generated shape must be measured.

Conventionally, measurement of diagonals is executed manually, according to standardized methods [9, 21]. This manual evaluation is not only expensive, but also interpretive and subjective. Moreover, the operators physical and psychical conditions (e.g. fatigue) potentially affect the hardness evaluation process.

Hence we focus on developing image processing algorithms which replace or at least support manual work. We start with images that are taken, which should be segmented in order to identify the four corner points. Having these corner points, diagonals can be calculated easily.

Different proposals have been made to compute results by the use of image processing software. Target of this work is to investigate active contours and level set algorithms with respect to Vickers images. Particularly, an algorithm has been implemented which segments images accurately and robustly. A measure for accuracy could be implemented as well, but this issue has not been investigated so far.

Contents

Chapter 1

Introduction

There are several proposals for image segmentation of Vickers indentations.

One group of algorithms rely on wavelet analysis [14, 23]. These methods assume that object borders are perfectly straight lines. As this constraint is not always redeemed perfectly, a precise segmentation cannot be assured. While the assumption that images meet a geometric constraint introduces robustness, accuracy decreases as the assumption is partially wrong.

Another approach [11] is based on edge detection followed by Hough transform and least squares approximation of lines. Edge finding techniques are based on the assumption that high differences between neighboring pixels imply that these pixels are part of the border. This assumption is not right at all. Highly noisy images, as well as very bright images (with low contrast), do not subject this assumption. It might occur that noise has a higher edge response than a real object boundary.

The method introduced in [15] applies thresholding followed by a Hough transform. Thresholding suffers from one big problem. If this method is applied to images, it is necessary to calculate a threshold value which depends on the current image. Of course it is not possible to use one single threshold for each image, as images differ too much. Unfortunately finding a threshold that perfectly separates objects from a background is not only very difficult, in fact, sometimes it definitely is not possible to segment images by using any threshold (assumption that object and background have different brightness is wrong). This happens especially if the object is quite bright and/or the background is dark or noisy.

Other suggested methods also binarise the image using thresholding [18, 24] (followed up by morphological closing).

Another approach is based on axis projection and Hough transform [26]. By summing up values of the same rows as well as values of the same columns, two one dimensional functions are processed. The indentation can be located, where the values are lower (as the object is darkly colored, dark colors have a lower gray value). This method suffers when images are

noisy or slightly rotated.

Methods relying on template matching ([17, 7]) are quite robust to noise. This is because big templates suppress noise as large regions are summed up (in contrast to small edge operators). In [17] square templates of different sizes and different rotations are matched with the image. The results of this approach are robust but only serve as approximations, as real Vickers are no perfect squares. In contrast, the template matching approach introduced in [7] provides robust as well as precise results. A high degree of accuracy is achieved by applying four corner templates instead of one complete square template.

Our work deals with different kinds of active contours algorithms. The aim of these approaches is that a contour with a defined initialization converges to real object boundaries within a number of iterations. The contour is forced by different energy functionals, which depend on pixel values of the image on the one hand and homogeneity criteria of the contour on the other hand.

First (chapter 2) we investigate the traditional active contours approach (snake), which is the most simple one and was introduced [12] first of all. This chapter also covers improvements like the gradient vector flow approach (GVF) [25], which avoids problems like the lack of edge propagation.

Chapter 3 deals with the level set approach [20], which is an alternative to traditional active contours which handles parametrization in an intrinsic way (i.e. by its level set).

As both, the snake and the level set methods highly depend on initialization, we identified that it is not adequate to use such an algorithm standalone for Vickers segmentation, because initialization would have to be different for different images (size, position, rotation). To deal with the initialization problem, a strict Shape-Prior gradient descent approach is proposed in chapter 4, which is highly noise resistant, but does not find points exactly (because of the strict shape). Moreover a Shape-Prior force was applied to level set algorithms, which aims in noise resistance, but still suffers from a poor initialization as well.

Chapters 5 deals with pre- and postprocessing of the images, in order to optimize segmentation results.

To get both, independence of initialization and noise resistance on the one hand and exact localization of points on the other hand, a multi-resolution approach is proposed in chapter 6. The results of our proposed approach are compared with those of the corner template matching strategy introduced in [7].

In chapter 7 we focus on collecting additional information of indentation images by regarding different focus levels. By applying a focus measure to all different focus level images, depth information of the image can be

gathered.

In chapter 8, we investigate the impact of unfocused images on the segmentation process. Especially, we identify (un)focus setups which are even beneficial for the segmentation.

In chapter 9, a gradual enhancement strategy is introduced, which is based on differently focused images. The aim is to reduce the runtime of the overall hardness test.

In chapter 10 the runtimes of different introduced algorithms are compared.

Chapter 11 concludes this thesis.

1.1 Vickers images

The images to segment approximately fit the following description:

- square geometry of object to segment
- dark object, light background
- diagonals are horizontally and vertically aligned
- object is situated close to center

Figure 1.1 shows quite perfect images.

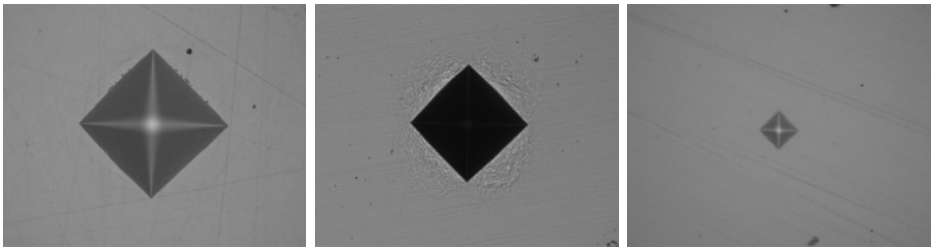


Figure 1.1: Images redeeming constraints

1.1.1 Noisy images

Images in figure 1.2 show many different kinds of noise.

1.1.2 Low contrast images

Some bright images lack of contrast, especially the diagonals' gray scale is the same as the background. Another inconvenience is the fact, that in such images, possible noise is often darker than the imprint. Examples are pictured in figure 1.3.

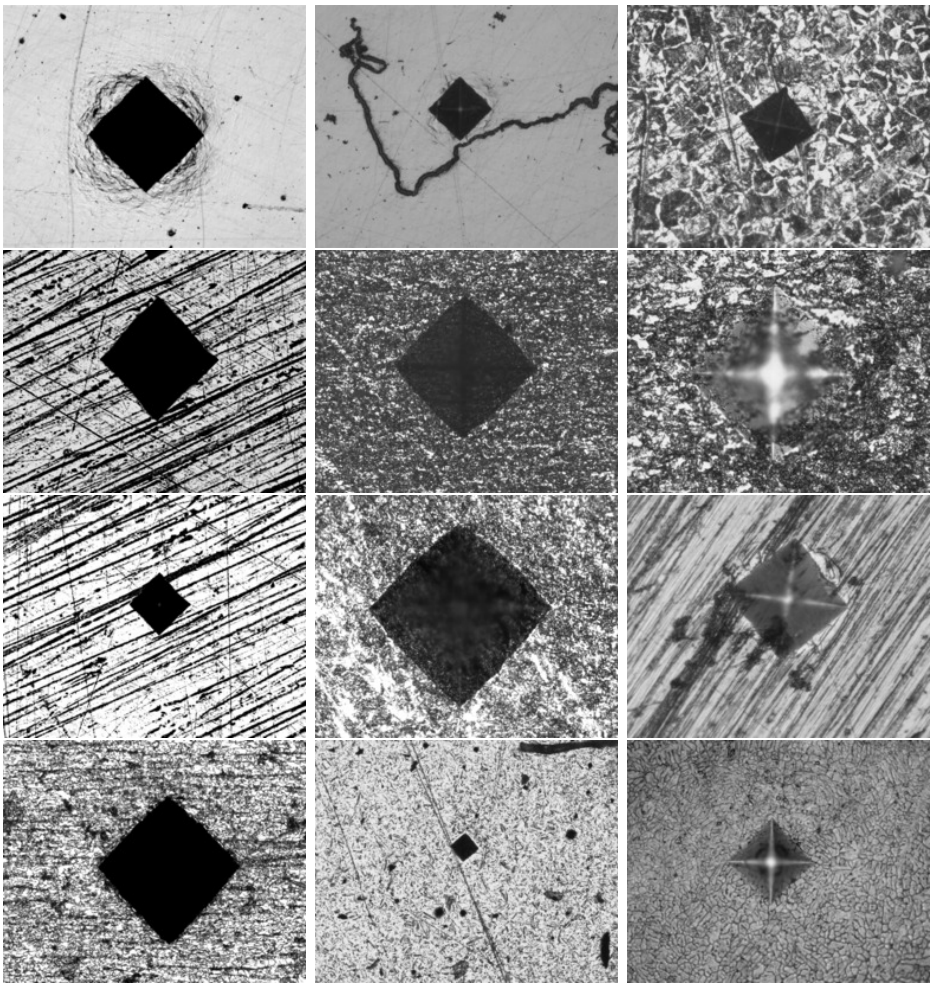


Figure 1.2: Noisy images

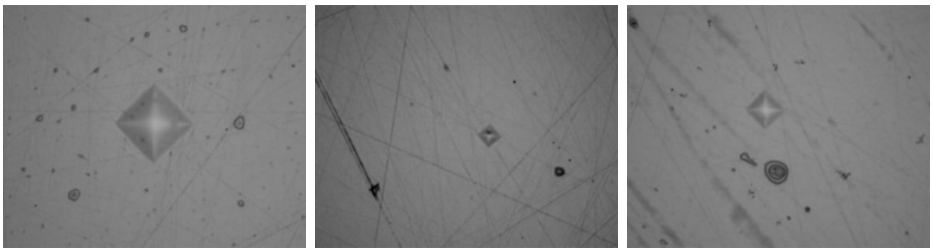


Figure 1.3: Low contrast

1.1.3 Violation of rotation

The diagonals are not always aligned horizontally and vertically. Some images are slightly rotated, like in figure 1.4.

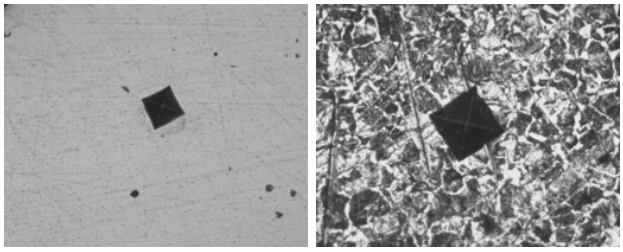


Figure 1.4: Violation of rotation constraint

1.1.4 Shape violation

Whereas the first image (figure (1.5)) shows a concave curvature, the second one shows a convex curvature. Moreover the first image does not exactly represent a square, as the right sides are longer than the left ones. The sides of the third image highly differ from straight lines. Examples are shown in figure 1.5.

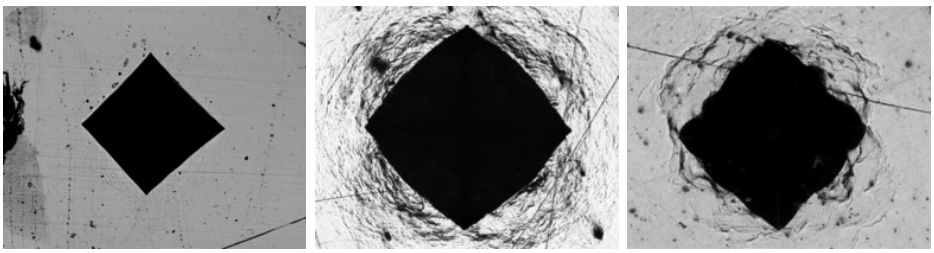


Figure 1.5: Shape violation

1.1.5 Different size

The following images represent the smallest and largest imprints in our database (shown in figure 1.6).

1.2 Database

For testing different algorithms and setups 150 test images (resolution 1280x1024 pixels) provided to us were used. In order to compare the calculated results with the ground truth, these 150 images where manually evaluated by

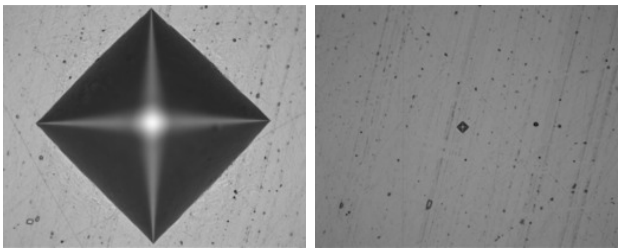


Figure 1.6: Different sizes

four people independently. The ground truth was determined by taking the mean of all four measures. For the first step of the multi-resolution approach (chapter 6), these images are downscaled by factor 10 (resolution 128x102 pixels).

1.2.1 Image series

In the chapters 7 - 9, we need image series with different focus setups. For this purpose, we have got 25 indentation series, each containing 40 images of the same indentation but with different setups. The quality of this series is considerably lower.

Chapter 2

Traditional active contours approach

Intuitively, an active contour is a closed curve which iteratively converges at the object's borders, by means of gradient descent of an energy functional. The curve is represented by a sequence of pairs containing x and y coordinates, which are markers that are connected with straight line segments.

Traditional active contours (also called snakes as introduced in [12]) is an attempt to minimize a cost (energy) function which depends on image pixels and the contour's shape by the use of gradient descent. Aim of that algorithm is that minimization of the cost function leads to an appropriate segmentation of the image. The energy function consists of an additive weighted mixture of internal energy, image energy and external energy. The weights can be adjusted in order to control the behavior of the snake. A snake is represented parametrically by $v(s) = (x(s), y(s))$, where $x(s)$ is the x coordinate and $y(s)$ is the y coordinate for a given element s of the contour.

$$E^*_{snake} = \int_{Snake} E_{snake}(v(s))ds \tag{2.1}$$

$$E^*_{snake} = \int_{Snake} E_{intern}(v(s)) + E_{image}(v(s)) + E_{extern}(v(s))ds \tag{2.2}$$

2.1 Internal energy

The internal energy consists of a linear combination of continuity (first order derivation) and curvature (second order derivation).
The continuity term prevents the marker points $v(s)$ from converging to few points with low image energy (i.e. the contour acts like a membrane). $v(s)$ should be approximately equally distributed over the contour.

The curvature term prevents the contour from developing sharp corners (i.e. the contour acts like a thin plate). However, corners or curves can be evaluated, but only if the image energy is strong enough to outweigh the curvature energy.

$$E_{intern} = \alpha \cdot |v'(s)| + \beta \cdot |v''(s)| \tag{2.3}$$

Large weighted parameters α, β (in ratio to other weighting parameters), make the contour more noise resistant, but details are segmented less exactly. Slight bendings are more evolved than sharp corners and the speed of convergence is declining. Small α and β, on the other hand, make the snake more vulnerable to noise. However, details would be segmented more exactly.

2.2 Image energy

This Energy is calculated from image pixels. Usually the image energy is calculated by applying an edge operator to the original image. Pixels near edges having a higher gradient represent a lower energy. The idea is that edges are likely to be part of the object's border. λ is a weighting parameter, ∇ is the gradient operator, $||\cdot||$ is the Euclidean norm and I is the image gray value.

$$E_{image} = \lambda \cdot E_{edge} = -\lambda \cdot ||\nabla I(v(s))|| \tag{2.4}$$

In line drawings the energy can be calculated directly from the pixels. The Energy is low at the lines and high everywhere else:

$$E_{image} = \lambda \cdot E_{gray} = \lambda \cdot I(v(s)) \tag{2.5}$$

For segmentation of Vickers images, clearly the first energy criterion in equation 2.4 is used, as object borders correspond with edges, not lines.
When increasing λ, the image energy becomes more important. That leads to faster convergence and potentially more exact segmentation, but to more noise sensitivity as well.

2.3 External energy

External forces like user interaction might be used, but this is not further investigated, as hardness image segmentation should run as batch process without any human interaction at all.

2.4 Analysis on indentation images

The OpenCV (V2.0.0) snake implementation algorithm was applied to Vickers test images. Although there exist lots of examples which can be segmented precisely, a real world test with actual Vickers images shows that this simple approach does not lead to appropriate results, especially if the starting configuration is not near to the real contour.

2.4.1 Problems

1. If the initial active contour is too far away from the actual image gradients, it has no chance to converge to the real contour. This happens because when the contour starts far away from the object's border, gradient descent does not succeed in finding the global optimum. Gradients near to the active contour are not influenced by the real border of the object. Consequently either the image energy is very low or even influenced by noise in a wrong way. Without image energy, a segmentation will never succeed. We call the region where gradient descent succeeds "capture range". This problem is shown in figure 2.1 (left).

2. Especially noisy images suffer, because the gradient image shows lots of regions with low image energy (figure 2.1, middle), where the contour potentially converges to.

3. Corners are not sufficiently detected. The active contour consists of a number of straight line segments, which connect the parametrized marker points. There is no reason for the marker points to move towards the corners, as the internal energy and the image energy would increase. The internal energy would increase because a corner introduced a high energy. The image energy increases as the gradient values at the corners are lower than edge pixels which are distant to the corners. Consequently, the corners of the object are cut (figure 2.1, right).

2.4.2 Solutions

- The gradient image can be calculated in different ways. Using the small standard operators like Sobel 3x3 etc. does not lead to sufficient results, as this operator produces very fine gradient images (figure 2.2 top row). That means, a sharp edge in the input image leads to a very thin response in the greadient image. Such thin gradients cannot be "found" by the active contour (no gradient-descent possible from far away).

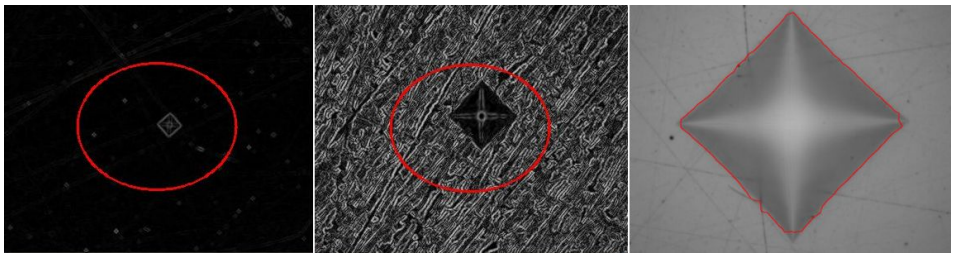


Figure 2.1: Problems: low capture range (left), noise (middle), cut corners (right)

One way is to use larger templates for edge detection (instead of a small 3x3 operator), which results in a propagation of the gradients (figure 2.2 bottom row). Consequently, the active contour's capture range is increased, as the edge information is propagated over the image.

- The first solution helps to improve on problem 1 and 2, but not on problem 3. Problem 3 occurs because the internal and the image energy increases if the contour would exactly detect the corners. The internal energy logically increases, because curvature inclines. The image energy increases, because the edge energy increases if the square partial derivation (dx or dy) would be small. At the corners, one square partial derivation is low.

$$E_{edge} = -||\nabla I(v(s))|| = -\sqrt{\frac{dI}{dx}(v(s))^2 + \frac{dI}{dy}(v(s))^2} \tag{2.6}$$

An idea to solve this problem is not to calculate the gradients dx and dy, but the diagonal gradients. While the traditional approach suffers from weaker gradients near the corners, now we have the opposite effect. Unfortunately, this method suffers if the image is not correctly rotated or no perfect square. Using the following energy criterion, gradient pixels near corners become as large as others on average:

$$E_{edge} = -max(|\frac{dI}{dx}v(s)|, |\frac{dI}{dy}v(s)|) \tag{2.7}$$

Nevertheless, the algorithm still tends to cut corners, as a corner increases internal energy. In order to reduce computational complexity, the calculation of the gradient in x direction only regards pixels in the same rows as the current pixel and calculation of the gradient in y direction only regards pixels in the same column as the current pixel. This strategy is utilized and improves the segmentation performance.

Figure 2.2 shows 4 different images from left to right and different sizes of gradient operators from top to bottom (3 pixels, 20 pixels, 100 pixels).

The contour was initialized equally, intentionally far away from the object, to show the effect of different sizes of edge operators. If the image has high contrast, low noise and sufficient size, the use of a larger gradient operators definitely is beneficial.

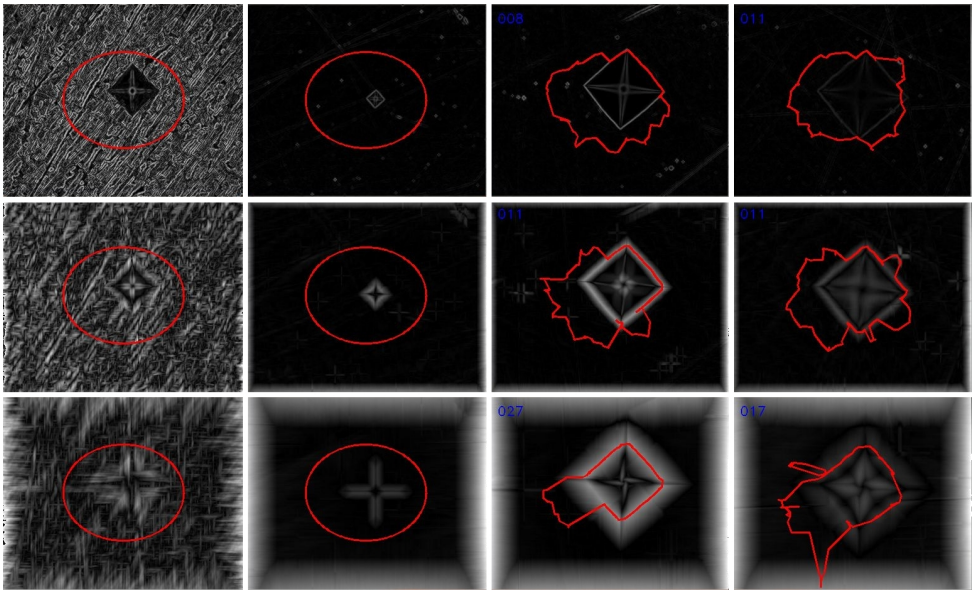


Figure 2.2: Different gradient operators

2.4.3 Remaining problems

Our solutions help segmenting perfect images of a certain size, as well es slightly noisy images. Still very problematic are images with small objects (the big gradient-operator does not work, as the size is limited by the object size), or images that are very dark (with a lot of noise) or quite bright. Such objects tend to disappear, when big gradient operators are used.
Moreover the problem of initialization of the contour is improved, but not solved completely. The convergence radius still depends on the size of the gradient operator. Consequently, one initialization for all images is still impossible, which implies that the traditional snake approach does not serve as standalone application for Vickers image segmentation, anyway it might serve as a second step of a multi-resolution algorithm if an approximation of corners is already available.
Figure 2.3 shows the dependency on initialization. The top images show the initialized contour whereas the bottom images show the results.

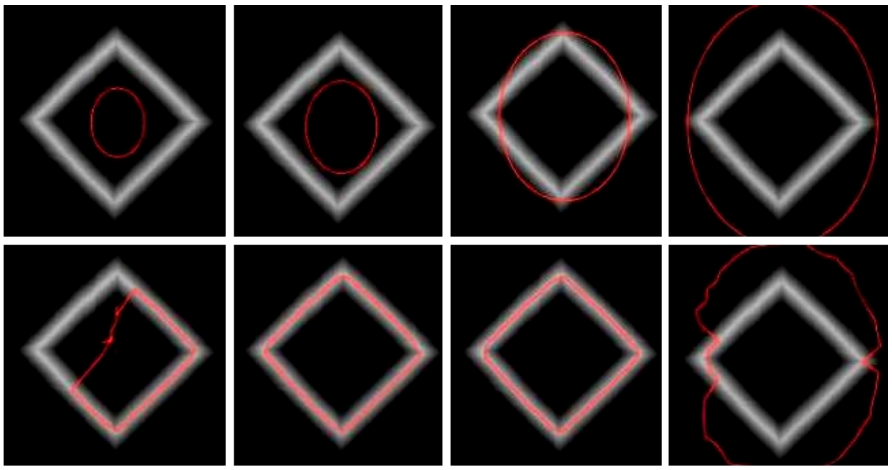


Figure 2.3: Different initializations

2.5 Gradient vector flow approach (GVF)

One problem of the traditional snake is edge propagation. Using a small edge operator for creating the edge map leads to a small zone of convergence. On the other hand, big gradient operators destroy small detail information, especially if the objects are quite small.
While a snake, based on a (small) standard gradient operator might be a good segmentation tool, if object borders are approximately known, it definitely is insufficient as a standalone application.
To avoid this inconvenience, the gradient vector flow approach (GVF) has been introduced [25]. Instead of calculating an edge map, the authors propose a vector field, where vectors are pointing towards regions with higher gradients.

In a first step an edge map $f(x, y)$ is derived from the image ($f(x, y) = ||\nabla I(x, y)||$). Consequently, the vectors of ∇f (∇ is the gradient operator) are pointing towards regions with a higher edge value f (as shown in figure 2.4). As the capture range of ∇f is still small, gradients have to be "smeared" away from the actual gradient's position.

In order to increase the capture range, the authors propose to minimize the following energy criterion:

$$E = \int_Y \int_X \mu((\frac{dr}{dx})^2 + (\frac{dr}{dy})^2 + (\frac{ds}{dx})^2 + (\frac{ds}{dy})^2) + ||\nabla f||^2 ||v - \nabla f||^2 dxdy \tag{2.8}$$

$v(x, y) = (r(x, y), s(x, y))$ is the resulting gradient vector field.

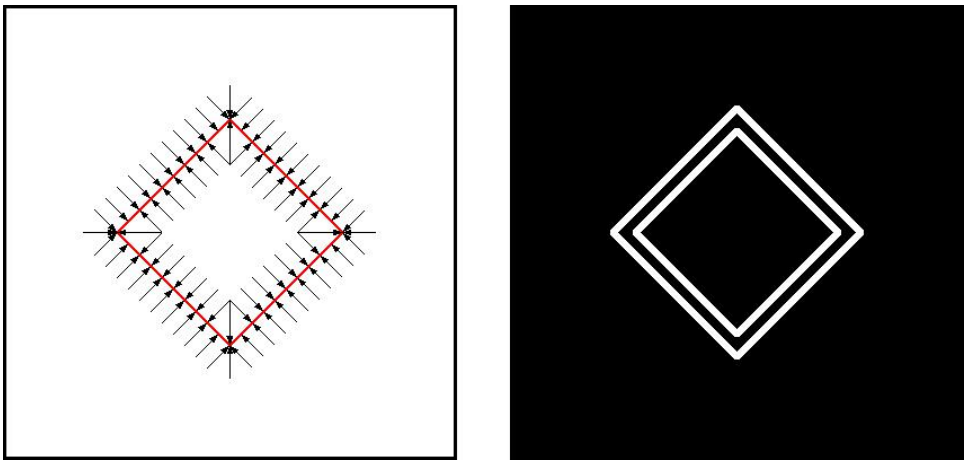


Figure 2.4: ∇f (left), $||\nabla f||$ (right)

Now we investigate two classes of vectors:

- $||\nabla f||$ is very high: In this case, the energy E is dominated by the second term and is consequenty minimized by setting $v = \nabla f$.
- $||\nabla f||$ is very low or zero: In this case, the E is dominated by the first term. Therefore, the square partial derivations of r and s should be as small as possible (i.e. neighboring vectors v and v' are pointing to similar directions). A minimization of the energy leads to a propagation of the gradient information.

The parameter μ adjusts the priorities of the two energy terms. Increasing μ leads to a smoother function v and suppresses noise but also potential important information.

The output vector field v is shown in figure 2.5.

The open source software Snake++ (`http://sourceforge.net/projects/snakecpp/`) includes an algorithm, which calculates v by iteratively decreasing the energy criterion E. We investigated this GVF implementation.

2.5.1 GVF evaluation

The main difference to the traditional active contours approach is the computation of the edge map of the image, which is a vector field in case of the GVF approach. Anyhow, the GVF vector field can easily be converted into an edge map which can be handled by the traditional algorithm. That was done, as we did not achieve acceptable results using the GVF contour evaluation algorithm, even though the vector field looks acceptable:

$$edgemap(x, y) = ||GVF(x, y)|| \tag{2.9}$$

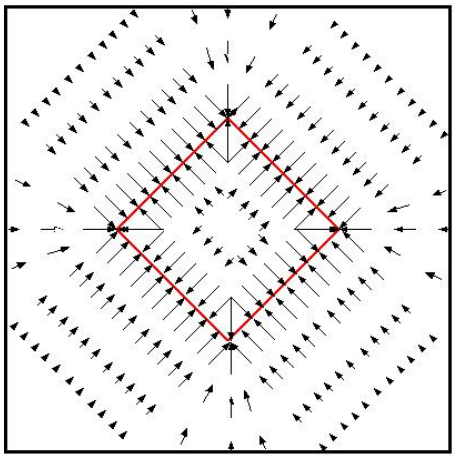


Figure 2.5: Gradient vector field v

Although there is a loss of information (direction), the snake implicitly calculates the direction by regarding the neighboring pixels. Like with gradient operators, pixel values are higher near to the actual object border.
As the vector field cannot be visualized straightforward, the following strategy has been chosen:

$$edgemap(x, y, Red) = ||GVF_x(x, y)|| \tag{2.10}$$

$$edgemap(x, y, Green) = ||GVF_y(x, y)|| \tag{2.11}$$

Now the red channel of the RGB image contains the absolute gradients in x direction, whereas the green channel contains the absolute gradients in y direction. Surely there is a loss of information applying the mentioned approach, as only the absolute values are regarded and so the direction of the vectors gets lost.

2.5.2 Advantage

The generation of the gradient vector field is advantageous to simple gradient operators as far as edge propagation is concerned. With standard operators, small and bright objects might disappear, or edges might be blurred too much. Calculation of a vector field does not blur boarders if the difference between object and background is too small. Instead, starting with an edge map (small edge operator is applied), and proceeding with propagation of gradients does not bring such negative effects. Moreover, a problem of the traditional snake is the convergence into concave regions. This inconveniences can be improved using the GVF snake.

The higher the number of iterations, the higher the degree of edge propagation (i.e. higher capture range). Figure 2.6 shows different numbers of

iterations.

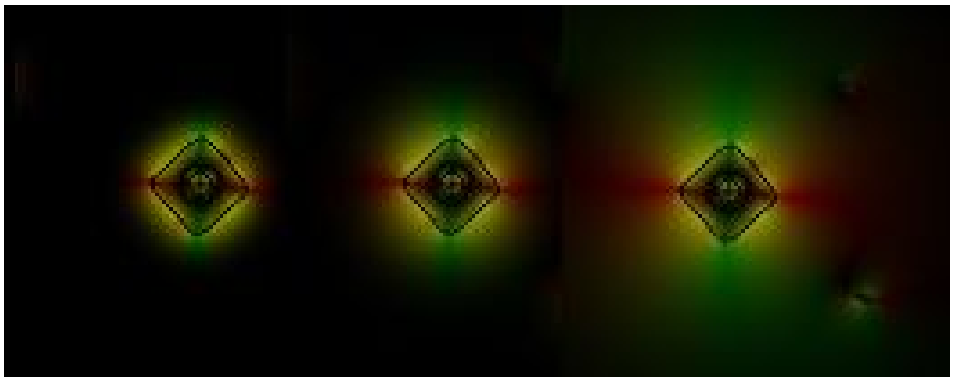


Figure 2.6: GVF different numbers of iterations, from left to right 80, 360, 3600

2.5.3 Problems

- One drawback of the GVF approach is the preservation of noise. While large gradient operators remove noise (act as low-pass filters), GVF propagates noise. That has advantages as even very small objects do not disappear (this would happen in case of large gradient operators). But one disadvantage is that the convergence to the global maximum (real image border), is unsettled by local minima, especially if images are noisy.

- Another drawback is the complexity of the calculation of a vector field. As each step just compares neighboring pixels, lots of iterations ($10^3 - 10^4$) must be applied to the original gradient image (to reach usable capture ranges). More iterations lead to a larger capture range.

  If images are downscaled, the GVF algorithm is less computationally expensive within a defined number of iterations. Moreover the capture range which is necessary decreases as distances measured in pixels are shrinking. Starting with high resolution images (1280x1024 pixels) without approximate knowledge of the placement of the object definitely is too expensive with respect to computational costs.

2.5.4 Acceleration - gradient diffusion field (GDF)

One the one hand, the GVF approach is able to generate a vector field with large zones of convergence (capture range), if the number of iterations is high enough. On the other hand computational costs are extraordinarily high (even for downscaled images).

To reduce computational costs, [13] introduced the gradient diffusion field (GDF). Like with GVF this approach aims at "smearing" gradients, in order to increase the capture range. In opposite the GDF approach is not based on a vector field, but on a scalar field.

It is defined by the following recursive scheme, where GoG is the Gaussian-of-Gradient operator, I is the image gray value and k is the total number of iterations.

$$GDF_0 = |GoG * I| \tag{2.12}$$

$$GDF_{i+1} = \max(|\alpha(i) \cdot (GoG * GDF_i)|, GDF_i) \tag{2.13}$$

$$\alpha(i) = 1 - \frac{i}{k} \tag{2.14}$$

For each iteration, the image is convolved with GoG operator, which applies edge detection and blurring.

The authors argue that the number of iterations (and the computational costs) required to reach a defined capture range is much lower.
Nevertheless, the capture range definitely is limited, as far as 8-bit gray scale images are used. The capture range is generated by descending pixel values (different pixel values are limited). Moreover operational experience showed, that GDF's limitation of the propagation zone is lower than GVF's one (if applied to Vickers images).
So like GVF, GDF will not serve as a standalone method for Vickers image segmentation, as the capture range is still not sufficient (especially for 1280x1024 pixels images).

2.6 Analysis on indentation images - starting constraints

One straightforward approach is to let the snake start from a defined starting region independent of the image. A big benefit would be that the snake would work stand-alone! We only have to ensure that the object is withing the starting contour, as otherwise the deformation reconstructs just one part of the object. To keep that constraint, a simple idea is to start from the very outside of the image. As the capture range in high resolution images does not measure up, images are downscaled to reduce computational expenses as well as to increase the capture range, in relation to the image size.
Figure 2.7 shows two different images (downscaled to 320x256 pixels) with different starting configurations, using the GDF approach.

Although the capture range would be large enough, to guide the contour to move towards the real object boundary, the GDF (and also GVF)

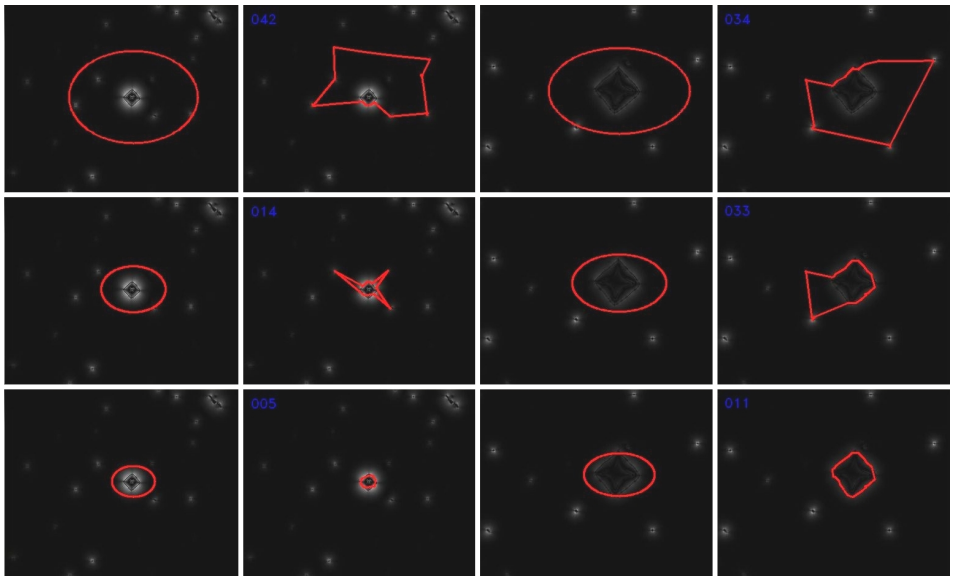


Figure 2.7: Different starting configurations

approach suffers from artefacts, induced by even little noise in the original image.
The more precise the initialization is, the better are the results. Therefore, this method is not robust enough to segment images standalone, without already approximate knowledge of the results. Accuracy definitely increases if a starting region is determined by the respective image.

2.7 Results

Active Contours algorithms are divided into 2 major steps, the calculation of a kind of edge map (or vector field) and the deformation of the contour (gradient descent). In order to give the snake a chance to deform intentionally, the surface of the edge map must be accurate.
On the one hand, the gradient descent of the contour requires gradients. Therefore, different approaches (GVF, GDF) propagate edge pixels to regions without gradients, to make the edge map more steady. Just using large gradient templates does not lead to enough propagation, as the size of the templates is limited by the size of the images.
On the other hand especially using the GVF (or the GDF) approach does not remove noise, which implies that the surface gets deceptive with many local minima if the input image is noisy.
Consequently, there are two opportunities to improve the segmentation process. One is to amend the snake deformation algorithm to be able to deal with local minima (e.g. a balloon force as introduced in [4]). The other one

is to use the snake algorithm as a second step for finding accurate results, having approximations of the objects already been given. The first idea has not been investigated with snakes but in the chapters 3 and 4 with similar approaches. The second idea has been investigated in chapter 6, where a multi-resolution approach is introduced.

Chapter 3

Level set approach

The level set method introduced in [20], is an alternative to the traditional snake model. Apart from other inconveniences, traditional active contours suffer from an explicit parametrization of the contour. The curve is given by a series of frontier (marker) points which are evaluated in each iteration. During the curve evolution significant problems may arise:

- Splitting
  The contour is not able to split into two or more parts in a natural way. To allow such a deformation, the algorithm has to be expanded.
- Regularization of parameter points
  To ensure that distances between frontier points stay approximately the same and do not collapse, the continuity regularization term is necessary. Otherwise the frontier has to be re-parameterized periodically.
- Moreover the evolution of edges may fail, as the neighboring marker points are connected by straight lines (details get lost).

In the level set formulation an explicit parametrization by frontier points is circumvented by using an intrinsic formulation. The evolving contour is given by its level set Γ.

$$\Gamma = \{(x, y) | \phi(x, y) = 0\} \tag{3.1}$$

$\phi(x, y)$ is a function which is 1 inside, -1 outside of the region and exactly 0 at the frontier of the evolved shape (figure 3.1). The points outside of the contour are given by Γ_{out}:

$$\Gamma_{out} = \{(x, y) | \phi(x, y) < 0\} \tag{3.2}$$

The points inside of the contour are given by Γ_{in}:

$$\Gamma_{in} = \{(x, y) | \phi(x, y) > 0\} \tag{3.3}$$

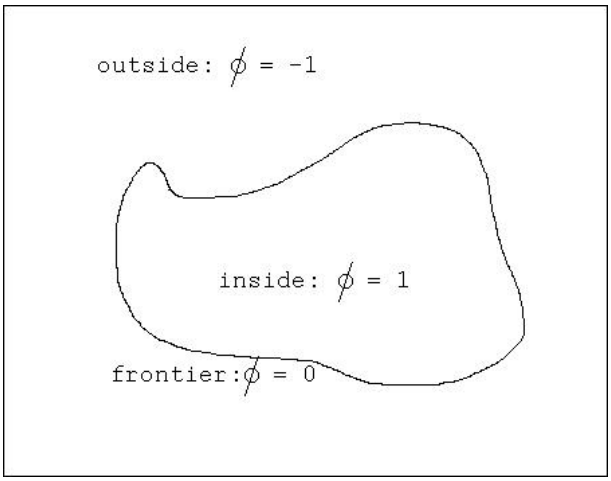


Figure 3.1: Level set function ϕ

Evolution of the frontier happens by moving the initial level set in normal direction with a specified speed v.
It is represented by the following partial differential equation (PDE):

$$\frac{d\phi}{dt} = v \cdot \frac{\nabla\phi}{||\nabla\phi||} \tag{3.4}$$

Intuitively, with progressing time t, ϕ is moving with speed v (which is variable) normally directed to the frontier ($\frac{\nabla\phi}{||\nabla\phi||}$). $\nabla\phi$ is divided by its norm ($||\nabla\phi||$) to normalize the vector.

Although the general curve evolution is defined, the level set approach is highly flexible. There exist lots of different ways of calculating the speed function v, which influences the behavior of the evolving level set.

3.1 Caselles - edge based approach

One commonly used edge based level set variation has been introduced in [1].
It is defined by following equation:

$$E = \int_0^1 g(I(\Gamma(q))) \cdot ||\Gamma'(q)||dq \tag{3.5}$$

$$g(I) = \frac{1}{1 + ||\nabla(G * I)||^2} \tag{3.6}$$

Γ is the evolving curve, G is a Gaussian operator, I is the image gray value and ∇ is the gradient operator.

$g(I)$ is high (converges to 1) if the gradient operator (∇) has no response, i.e. the image has no gradients at the current position. $g(I)$ is low in the

case of high gradients. $||\Gamma'(q)||$ is high at the evolving curve and low ($\Gamma = 0$) everywhere else. To put it in a nutshell, $g(I(\Gamma(q)))||\Gamma'(q)||$ is high if the contour is positioned where the image gradients are low. The minimization of E minimizes such occurrences.

3.2 Chan-Vese - region based approach

Caselles' level set method relies on gradient pixels, like the traditional snake algorithm. As with the snake approach, the edge based level set algorithm requires the propagation of edges to increase the capture range. Another problem is that image gradients might be weak.
To bypass this issues, in [2] a region based approach has been introduced, where the force of the contour is not based on image gradients. This method is based on the assumption that the object's surface as well as the surface outside of the object are homogeneous as far as its gray value is concerned.

A simple region based approach is given by the following equation (I is the image gray value, $\bar{I_{in}}$ and $\bar{I_{out}}$ are the average values inside and outside of the contour):

$$E = \int_{\Gamma_{in}} (I(v) - \bar{I_{in}})^2 dv + \int_{\Gamma_{out}} (I(v) - \bar{I_{out}})^2 dv \tag{3.7}$$

Intuitively, energy is low if the gray values inside of the contour are equal and the gray values outside of the contour are equal.

Whereas this criteria does not depend on a regularization (curvature) criteria of the evolved curve, the criterion proposed by Chan-Vese does:

$$E_{CV} = \int_{\Gamma_{in}} (I(v) - \bar{I_{in}})^2 dv + \int_{\Gamma_{out}} (I(v) - \bar{I_{out}})^2 dv + \lambda \int_{\Gamma} ||\nabla \phi(v)|| dv \tag{3.8}$$

I is the image gray value, ∇ is the gradient operator and λ is the curvature weighting term. The higher λ the less is the probability to evolve sharp corners.

3.3 Statistical approach

Chan-Vese's approach is based on the assumption that images do not necessarily have strong gradients at their boundaries, but the average color (gray value) has to be different. This assumption usually is a quite good idea, however it is inappropriate for some kinds of images, like these in figure 3.2.

All of these images contain noise which is darker than the object to segment. Consequently, a region based model would state that such dark

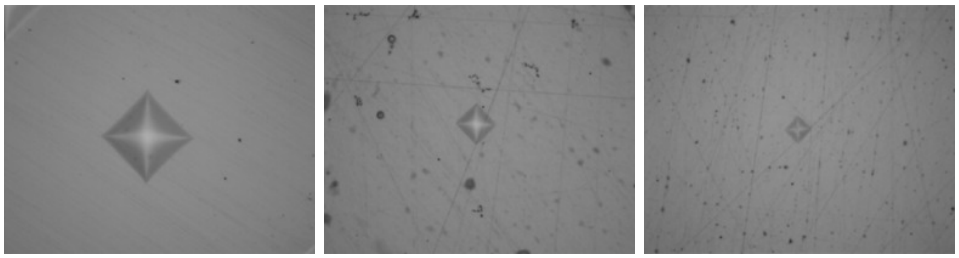


Figure 3.2: Inappropriate images

noise pixels are more likely to be part of the object than to be part of background (as background average color is brighter). Obviously, this does not match with reality.
To overcome that inconvenience, a statistical approach has been introduced in [5]:

$$E = -\sum_i \int_{\Omega_i} \log p_i(f(v))dv \tag{3.9}$$

Ω_i is one region to separate from the others and p_i is the probability density function of a feature f in the region Ω_i. While the region based approach only allows definition of two regions (outside, inside), the statistical definition allows arbitrary disjunct regions.

Transforming the energy criterion gives a more intuitive representation:

$$\begin{gathered} E = -\sum_i \int_{\Omega_i} \log p_i(f(v))dv = \\ -\sum_i \sum_{f(v)\in\Omega_i} p_i(f(v)) \log p_i(f(v)) = \sum_i H_{\Omega_i} \end{gathered} \tag{3.10}$$

Consequently, the energy can be minimized, by minimizing the sum of the entropies (H_{Ω_i}) of all regions Ω_i. While in the region based approach homogeneous colors in separate regions are assumed, the statistical approach more generally assumes homogeneity based on feature vectors $f(v)$ with (theoretically) an arbitrary number of dimensions. The declared statistical model is a generalization of the region based one.

Applying the commonly used regularization criteria (minimizing the length of the contour $|C|$), gives the following energy functional.

$$E = -\sum_i \int_{\Omega_i} \log p_i(f(v))dv + \alpha|C| \tag{3.11}$$

In the case of Vickers segmentation we target in separating one object from background, so two partitions Ω_{in} and Ω_{out} are claimed. We call them

Γ_{in} and Γ_{out}.

$$E = -\int_{\Gamma_{in}} \log p_{in}(f(v))dv - \int_{\Gamma_{out}} \log p_{out}(f(v))dv + \alpha|C| \tag{3.12}$$

We replaced the continuity term $\alpha|C|$ by the functional proposed by Chan-Vese to simplify implementation:

$$E = -\int_{\Gamma_{in}} \log p_{in}(f(v))dv - \int_{\Gamma_{out}} \log p_{out}(f(v))dv + \lambda \int_{\Gamma} ||\nabla\phi(v)||dv \tag{3.13}$$

This formulation allows not only to use gray scale as feature, but each feature which could be defined for a specific pixel might be included in a feature vector of arbitrary length. Especially any kind of gradient information is interesting. Using a gradient direction information e.g. segmentation of an image like shown in figure 3.3 would become possible.

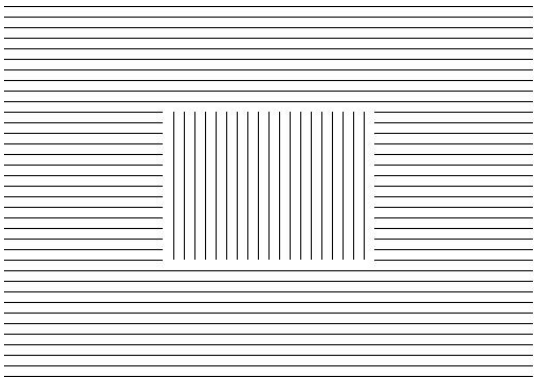


Figure 3.3: Structure image

In the case of only one single feature two one dimensional probability densities are computed (one for the object, one for the background), using sample data from the current position of the contour. In the case of a multidimensional feature vector, these densities have the same dimensionality as the feature vector has.

3.3.1 Generation of densities p_{in} and p_{out}

p_{in} and p_{out} are probability densities. Unfortunately this densities are not given, but have to be evaluated. What we have are 2 sets of feature vectors, $f(\Gamma_{in})$ and $f(\Gamma_{out})$.
There are two major ways to model a distribution by a set of samples. The

first is to evaluate the parameters of a specific distribution. Though this requires the knowledge of the distribution. As we do not want to assume a specific distribution, we decided for a non parametric method. Particularly the empirical distribution is convolved with a Gaussian Parcen window function of different sizes.
For each dimension of density, the Parcen window is applied separately. The used size depends on the feature as well as the total number of dimensions. This is because in higher dimensions, the same number of features would lead to a sparse probability density matrix (deceptive surface), while in lower dimensions an empirical distribution without any Parcen window could be smooth enough.

You have to make aware that Parcen windows approximations are computationally expensive (because of convolution) and have to be applied for each step of iteration separately. Complexity increases exponentially with respect to the number of dimensions. Consequently we mainly investigate one and two dimensions.

Empirically, we proved that the following feature vector is a good choice:

$$f(v) = (I(v), ||\nabla I(v)||)) \tag{3.14}$$

Feature vectors with a third dimension have been tested, but did not improve the results.

As the computational expense of a Parcen window approximation could not be accepted, we did not apply any windowing function, but used the empirical distribution straightforward instead.

3.4 Analysis on indentation images

While the edge based functional (Caselles) turned out to be quite vulnerable to initialization and noise, the Chan-Vese region based algorithm generates quite robust and exact results. The statistical approach does not provide better results, so the higher computational affords are not justified.

3.5 Software

Two different open source level set image processing software packages have been tested and extended.

- Creaseg
  This is a Matlab implementation of diverse level set algorithms like Chan-Vese, Caselles, etc.
  `http://www.creatis.insa-lyon.fr/~bernard/creaseg`

- Ofeli
  A very fast C++ implementation of Chan-Vese and Caselles algorithms.
  `http://code.google.com/p/ofeli`

First we investigated the Creaseg tool, as it deals with numerous different algorithms. As the runtime of the algorithms did not satisfy our demands, we later concentrated on Ofeli, which is a high performance implementation in C++.
Ofeli implements the Caselles and the Chan-Vese level set algorithms.
We have extended the software by the presented statistical approach. Further more, prior knowledge about shape has been introduced (chapter 4).

3.6 Initial configuration

Edge based algorithms definitely have to be initialized nearby the boundary (as with snake). Although region based approaches are known to be less vulnerable to initial configuration, starting with a general level set is not advantageous. On the one hand, even region based algorithms do not succeed to converge at the desired boundary, if the contour is too far away from it, because the image background often is quite dark near the image borders (caused by exposure). On the other hand if such long distances must be managed, computational costs are tremendous (runtime $> 10s$). To avoid that inconveniences, in chapter 6 a multi-resolution approach is suggested, which combines a robust shape-prior segmentation on downscaled images as a first step and a level set approach as a second step for more accurate results.

Chapter 4

Introduction of Shape-Prior

The methods mentioned so far suffer from converging to local minima. Especially in noisy images the traditional active contours as well as the level set methods lead to bad results, especially if the initialization is not appropriate.

Once again, the traditional active contours model is investigated. The proposed model is based on image energy (supplied by edges) and internal energy (curvature, connectivity).
For the general case of image segmentation this energy functional might be a good approach. On the other hand, for a special kind of segmentation (e.g. Vickers) we have some more information about the shape of the object than just homogeneity.
The following three possibilities to introduce Shape-Prior knowledge are discussed in the next chapters.

- Gradient descent method with limitation by parametrization
- Shape-Prior level set approach with gradient descent of prior level set [3]
- Shape-Prior level set approach with direct computation of prior level set [27]

4.1 Strict Shape Prior Gradient descent method

Different from other level set or snake based Shape Prior introductions, we propose a quite different way for robust segmentation of shapes which are known a priori. This approach requires a parametric description of the prior shape. The object that will be segmented will have exactly the prior shape, as not a contour (parametrized by points or level set), but the object description parameters are directly evolved by gradient descent. While the traditional active contours as well as level set algorithms allow arbitrary

deformation of the initial contour, this approach only allows evaluation of the following four parameters.

1. x_0 ... translation x axis
2. y_0 ... translation y axis
3. r_0 ... scaling
4. α_0 ... rotation

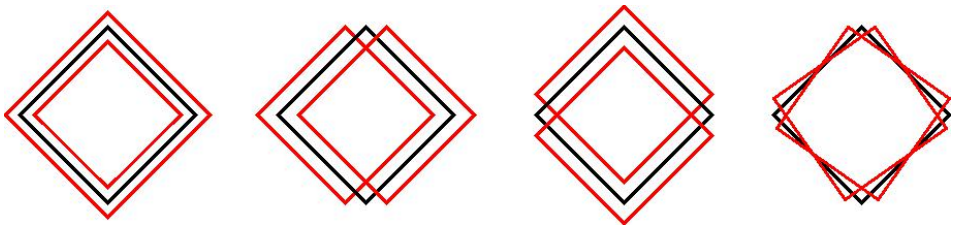


Figure 4.1: 4 degrees of freedom

- In the first equation we assume that the center of the object is given by $(0, 0)$ and it is not rotated. The separating contour is given by the set of points (x, y), which the following property:

$$|x| + |y| = r_0 \tag{4.1}$$

  That means, every point of the contour has the same Manhattan-distance from the point $(0, 0)$. The parameter r_0 has to be evolved.

- Now we assume that the image is not situated exactly in the middle, but it is not rotated:

$$|x - x_0| + |y - y_0| = r_0 \tag{4.2}$$

  Now every point of the contour has the same Manhattan-distance from the point (x_0, y_0). The parameters r_0, x_0, y_0 have to be evolved.

- The following equation additionally allows rotation:

$$\begin{aligned} |(x - x_0) \cdot cos(\alpha) + (y - y_0) \cdot sin(\alpha)| + \\ |(x - x_0) \cdot sin(\alpha) - (y - y_0) \cdot cos(\alpha)| = r_0 \end{aligned} \tag{4.3}$$

Of course, this algorithm will not be able to segment Vickers images perfectly, as Vickers' shape often cannot be described by a perfect square. Though this is not our objective, but the results could serve as good approximations for another approach (snake, level set). Hence we do not target in perfect accuracy, but in a pretty good segmentation of a very high rate of images.

4.1.1 Gradient descent based on edges

In order to allow translation, scaling and rotation, the following equation defines the contour Γ:

$$\begin{aligned}\Gamma = \{(x,y) : &|(x-x_0)\cdot cos(\alpha) + (y-y_0)\cdot sin(\alpha)| + \\ &|(x-x_0)\cdot sin(\alpha) - (y-y_0)\cdot cos(\alpha)| = r0\}\end{aligned} \tag{4.4}$$

x_0, y_0, r_0, α are parameters of the evolving shape.
In the discrete case, the contour is defined in the following way: (e.g. with $\epsilon = 1$)

$$\begin{aligned}\Gamma = \{(x,y) : &||(x-x_0)\cdot cos(\alpha) + (y-y_0)\cdot sin(\alpha)| + \\ &|(x-x_0)\cdot sin(\alpha) - (y-y_0)\cdot cos(\alpha)| - r0| < \epsilon\}\end{aligned} \tag{4.5}$$

$$E(x_0, y_0, r_0, \alpha) = -\frac{1}{|\Gamma|}\int_\Gamma ||\nabla I(c)||dc \tag{4.6}$$

$||\nabla I(c)||$ is the norm of the image gradient. The energy is minimized by gradient descent.

$$s = (x_0, y_0, r_0, \alpha) \tag{4.7}$$

The initialization is given by the vector s_0.

$$s_{n+1} = s_n + \lambda(\nabla E) \tag{4.8}$$

λ which usually is a multiplicative component, is called step size. To allow lambda to act as a signum function (one pixel left, stay, one pixel right), which can deal with numerical issues, it is more generally defined as function. We use the following definition:

$$\lambda((x_1, ..., x_n)^T) = (sign(x_1), ..., sign(x_n))^T \tag{4.9}$$

The gradient ∇E is calculated numerically.

$$\nabla E = (\frac{dE}{dx}, \frac{dE}{dy}, \frac{dE}{dr}, \frac{dE}{d\alpha})^T \tag{4.10}$$

E.g. the partial derivation of the x dimension is calculated in the following way:

$$\frac{dE}{dx}((x,y,r,\alpha)^T) = E((x+1,y,r,\alpha)^T) - E((x-1,y,r,\alpha)^T) \tag{4.11}$$

4.1.2 Gradient descent based on edges and balloon

Although this approach is better able to deal with local minima caused by noise which introduce high gradients, gradient descent still suffers from small or often no gradients far away from the object (that depends on the gradients of the object and the gradient propagation algorithm). The balloon approach, introduced in [4] for active contours, deals with this problem by adding an energy term, forcing the contour to become smaller/larger.
Our simple model allows applying a kind of balloon force in an easy but effective way: Instead of calculating r_0 by gradient descent (a part of the vector s), r_0 is simply decreased by one in each step of descent. As the contour starts near image boundaries, it necessarily has to cross objects boundaries, when getting smaller and smaller.
Unlike unforced gradient descent, the proposed method does not stop before r_0 becomes small (zero). In a second step the history of the gradient descent has to be analyzed to get the best fitting vector r_n. Several strategies are proposed later.

4.1.3 Gradient descent based on region property

On the one hand, gradient based active contours suffer from high computational costs (gradient must be propagated far, especially if no balloon force is used). On the other hand, noise or little gradients affect the algorithm as far as segmentation performance is concerned.
Although the balloon force slightly reduces this issues, the energy model proposed by Chan-Vese for level set methods can also be allied with our model:

$$E_{Region}(x_0, y_0, r_0, \alpha) = \int_{\Gamma_{in}} (I(r) - \bar{I_{in}})^2 dr + \int_{\Gamma} (I(r) - \bar{I_{out}})^2 dr \quad (4.12)$$

$$\begin{aligned} \Gamma_{in} = \{(x,y) : &|(x - x_0) \cdot cos(\alpha) + (y - y_0) \cdot sin(\alpha)| + \\ &|(x - x_0) \cdot sin(\alpha) - (y - y_0) \cdot cos(\alpha)| < r_0\} \end{aligned} \quad (4.13)$$

$$\begin{aligned} \Gamma_{out} = \{(x,y) : &|(x - x_0) \cdot cos(\alpha) + (y - y_0) \cdot sin(\alpha)| + \\ &|(x - x_0) \cdot sin(\alpha) - (y - y_0) \cdot cos(\alpha)| > r_0\} \end{aligned} \quad (4.14)$$

I is the image gray value, $\bar{I_{in}}$ and $\bar{I_{out}}$ are the average gray values of inside and outside the contour. Surely our strict Shape-Prior approach does not need any regularization term at all.

4.1.4 Gradient descent based region property and balloon

Although the problem of initialization can be improved in many cases, there are some cases, where a general initialization does not lead to a convergence to the global minimum. Especially images with small or bright foreground

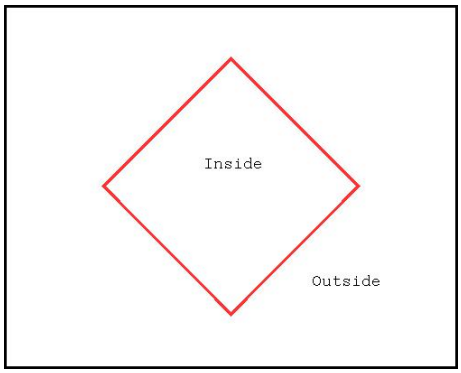


Figure 4.2: Region based approach

objects suffer. On reason is that the pixels, situated at image margins are often darker than other background pixels (which are situated more in the middle) and might be as dark or even darker than (parts of) the object to segment.
To prevent the contour from descending into false local minima, again a balloon force is implemented, similarly like above.

4.1.5 Gradient descent based directed gradients and balloon

This method has been derived from the general regional (Chan-Vese) model. Based on the knowledge that objects are darker than backgrounds, a directed edge operator is introduced into the Shape-Prior contour evolution. "Directed" means that gradients have to be orthogonal to the contour in order to generate a positive response only if the pixels inside are darker than the pixels outside. This should introduce a higher level of noise resistance. Moreover, as pixels are only regarded within a defined zone, ones very far outside (which might be quite dark, although they do not belong to the object) do not influence the segmentation process, at least when the radius r of the vector s is not too big.
Figure 4.3 shows the shape of a template. The best results were achieved with a template of thickness 3. That means pixels with a distance of maximal 3 pixels from the contour are regarded.

4.1.6 Gradient descent: statistical approach

The region based approach does not depend on edges, which implies that it is less vulnerable to poor initialization and weak edges. Moreover object and background color do not have to be known exactly. But the average color of the object has to be different from the average background color, which should be clear, when regarding the energy criterion. That is a huge restriction when segmenting images (as mentioned in chapter 3). Although

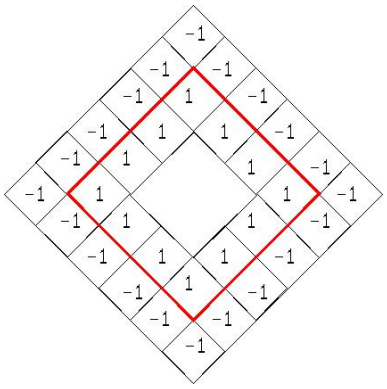


Figure 4.3: Directed edge template, thickness 1

Vickers images usually redeem that constraint, segmentation might profit from a more general approach.
This formulation depends not just on different average gray values, but on differences of the plausibility densities. Moreover not just the gray value, but even other features of the image might be used for segmentation.

$$E(x_0, y_0, r_0, alpha) = -\int_{\Gamma_{in}} \log(p_{in}(f(v)))dv - \int_{\Gamma_{out}} \log(p_{out}(f(v)))dv \tag{4.15}$$

- 1-dimensional
  The most straightforward approach uses only gray value as the single feature (v is a pixel of the image):

$$f(v) = (I(v)) \tag{4.16}$$

- 2 and more dimensional attempts:

$$f(v) = (I(v), ||\nabla I(v)||)) \tag{4.17}$$

$$f(v) = (I(v), \frac{dI}{dx}(v), \frac{dI}{dy}(v)) \tag{4.18}$$

$$f(v) = (I(v), ||\frac{dI}{dx}(v)||, ||\frac{dI}{dy}(v)||) \tag{4.19}$$

The higher the dimensionality of the feature vector f, the higher the computational costs, as for each step of the iterative gradient descent, the probability densities p_{in} and p_{out} have to be calculated, which are n dimensional. Moreover, a higher dimensionality causes the probability function (which is a matrix of n dimensions) to become a sparse matrix, as the number of matrix elements is exponentially increasing whereas the number of

features stays the same. When the elements of the matrix are rare, the empirical distribution cannot be utilized straightforward.
Consequently, it is necessary to estimate the real probability density function. This is done by applying a Gaussian Parcen window in different sizes.

Empirical tests proved, that the following feature vector produces the best results:

$$f(v) = (I(v), ||\nabla I(v)||)) \tag{4.20}$$

A Gaussian Parcen window is applied to the second dimension (edge information) with variance $\sigma = 2$. For the first dimension (gray value) the empirical density function is being used.

4.1.7 Decision rules for the balloon approach

Strategies without balloon force do not lead to acceptable results. The problem is, that simple gradient descent suffers from descending into local minima. Hence we have proposed a kind of balloon force, which ensures that the size of the contour is declining in each iteration. In contrast to the traditional approach (without balloon force), now we do not stop if the optimum is reached, but continue until a defined minimum size of the contour is reached.

As this approach does not lead to one single minimum, but several local ones, another exercise is to find out which of the local minima is the desired one.

- Global minimum decision rule:
  The simplest way is to use the parameters of the global minimum (lowest of all local minima) as result parameters.

- Highest gradient decision rule:
  On each local minimum, the gradients near the contour (edge intensity) are calculated. The winner is the configuration with the highest absolute gradients.

- Highest gradient decision rule followed by another attempt based on region property:
  As the balloon forces the contour to shrink in each iteration, when the size of the object is reached, the rotational alignment might suffer. This attempt allows the contour in a second step to rotate without the balloon force, for more exact results. However, the results cannot be improved.

- Single directed gradient operator:
  This idea is based on the knowledge that inside of the object pixels are

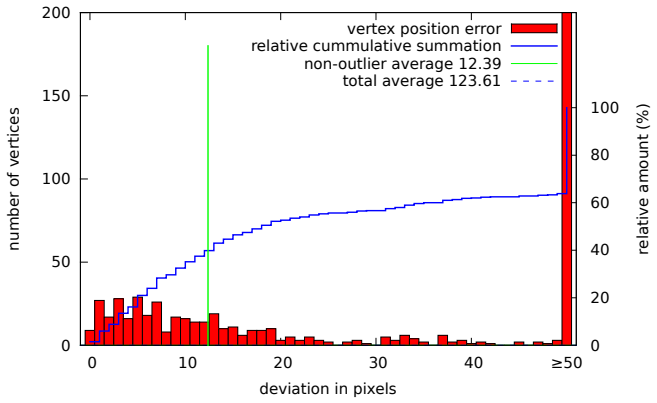


Figure 4.4: Edge based with balloon, highest gradient decision strategy

likely to be darkly colored, while background pixels should be brighter. To decide for one minimum, the template introduced in section 4.1.5 is used.

4.1.8 Analysis on indentation images

Like mentioned above, a pixel accurate segmentation of Vickers images is not possible using such a strict Shape-Prior method. But as we only want to get a good approximation of objects (especially, an approximation of the four corner points), the aim is that the number of corner points within a given distance from the actual corner points is as high as possible. In opposite, the number of corners detected farther away from their actual position than a given threshold should be as low as possible.
This threshold has been chosen to be 5 pixels in the downscaled (factor 10) image, i.e. 50 pixels in the original image. Nevertheless, most corner points that are detected within this range are much more accurate (within 0 to 2 pixels (respectively 0 to 20 pixels)). As a metric, the Euclidean norm has been used. Our database consists of 150 images, i.e. 600 edge points are segmented.

In figure 4.4 - 4.13 different strategies are evaluated. For each deviation in pixels (i.e. the distance between calculated and actual point), the number of edge points with respective distance to the real points are shown. The right most bar collects all the outliers (distance greater than 49 pixels). As we focus on reducing outliers, in the bar charts especially the right most bar is important for assessing the different segmentation strategies.

First we would like to point out that the balloon force improves performance for all different models. The results of all methods without the

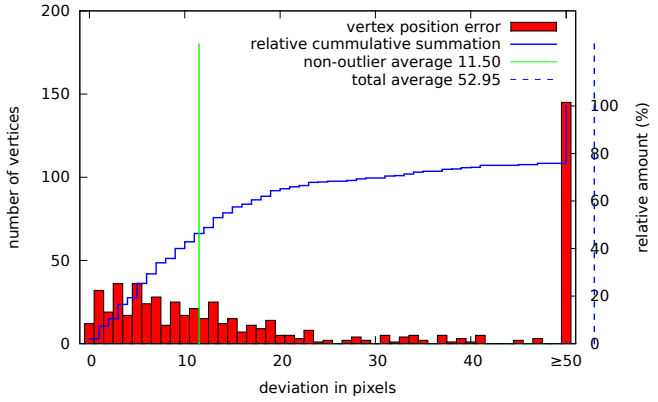


Figure 4.5: Edge based with balloon, directed edge decision strategy

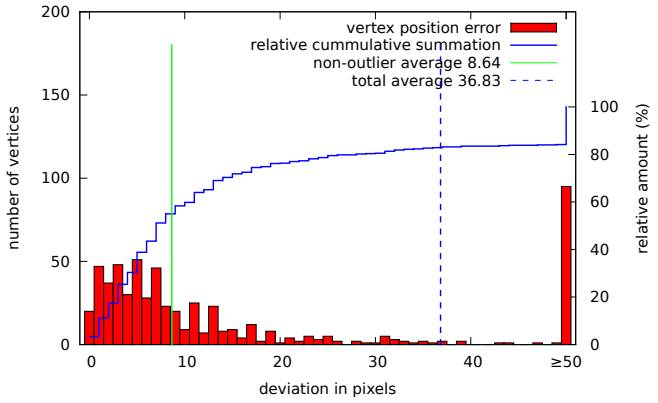


Figure 4.6: Region based with balloon, with highest gradient decision rule

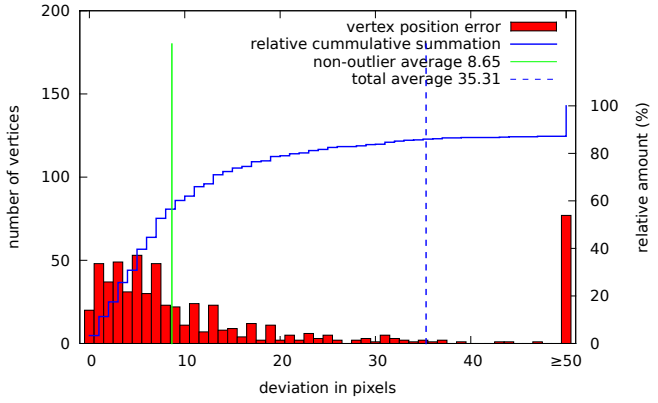


Figure 4.7: Region based with balloon, with directed edge decision rule

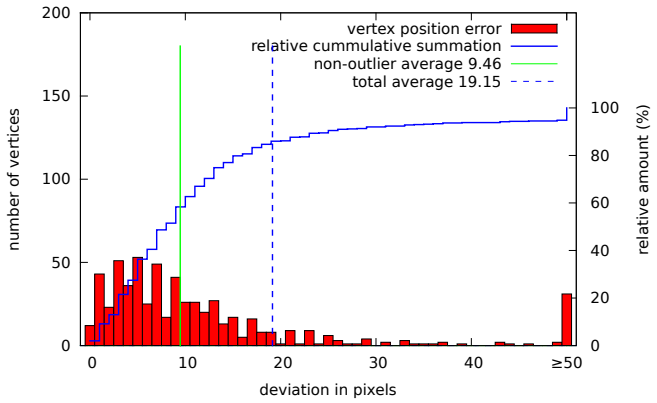


Figure 4.8: Directed edge with balloon, using global minimum

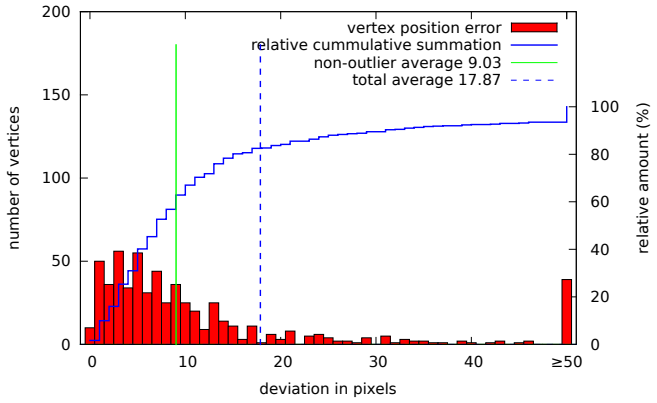


Figure 4.9: Statistical method with balloon, 1 dimensional: $f = (I(v))$, using directed edge decision strategy

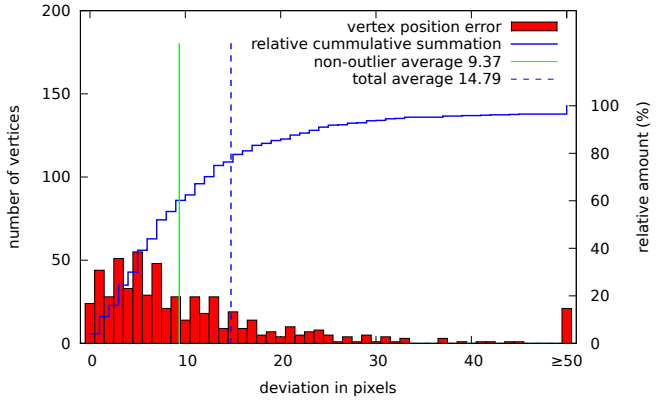


Figure 4.10: Statistical method with balloon, 2 dimensional: $f = (I, ||\nabla I||)$, no Parcen window, using directed edge decision strategy

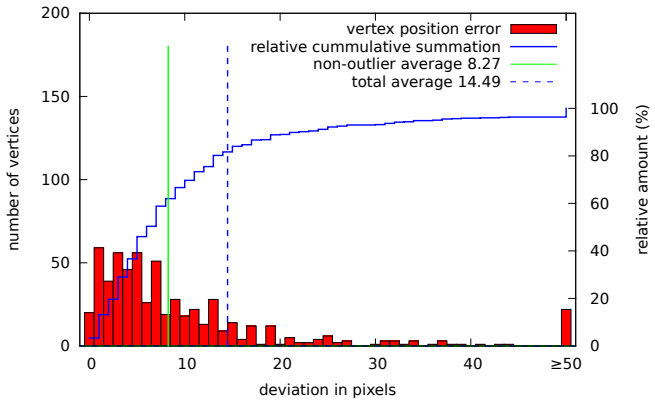


Figure 4.11: Statistical method with balloon, 2 dimensional: $f = (I, ||\nabla I||)$, Parcen window on gray value, using directed edge decision strategy

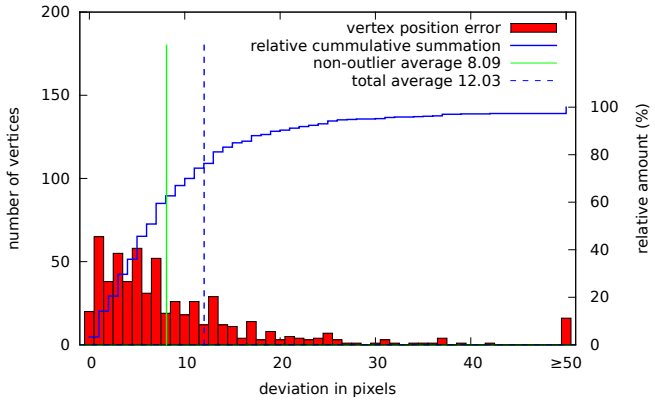


Figure 4.12: Statistical method with balloon, 2 dimensional: $f = (I, ||\nabla I||)$, Parcen window on the second dimension and directed edge decision strategy

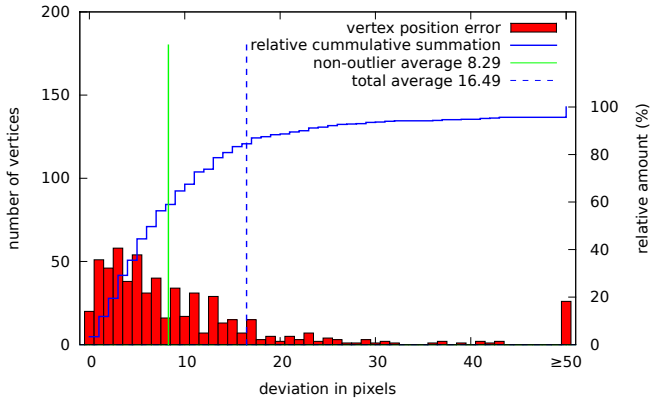


Figure 4.13: Statistical method with balloon, 3 dimensional: $f = (I(v), \frac{dI}{dx}, \frac{dI}{dy})$, using directed edge decision strategy

balloon force are inferior and are not shown. As you can see, the simple edge based (figures 4.4, 4.5) as well as the region based approaches (figures 4.6, 4.7) are not able to provide appealing results. Especially the edge based calculation suffers from very bad segmentation for at least about 25 % of all corners. But also the region based approach fails in about 15 % of all attempts.
Much more successful results are produced by the directed edge method (figure 4.8). Only 5 % of all calculations failed (distance from actual point more than 5 pixels).
Additionally we investigated the effects of different decision rules (section 4.1.7). Figures 4.4 and 4.6 show results with "highest gradient" decision strategy. Meanwhile figure 4.5 and 4.7 show results with "directed edge" strategy which are better, especially as far as outliers are concerned. The "directed edge" decision rule turned out to be most competitive for all different algorithms.

The statistical approach works with lots of adjustment parameters. First we have to decide for a feature vector, after that a Parcen window function has to be chosen for each dimension of the feature vector. While errors using the straightforward one dimensional approach (figure 4.9) are moderate, but not perfect, the two dimensional one (figures 4.10, 4.11, 4.12) delivers a quite good error rate. Especially the version using a Parcen window ($\sigma^2 = 2$) in figure 4.12 delivers the best results of all with an error rate of about 2.5 %. Hence this method is used in the multi-resolution framework (chapter 6), as a first step to get good approximative results.
Tests with the statistical approach using feature vectors with more than 2 dimensions (e.g. figure 4.13) did not lead to better results, but caused extraordinarily more computational complexity.

As a next step we compared the best configuration of our gradient descent algorithm with the template matching algorithm introduced in [17]. The template matching method relies on square templates of different sizes and rotations. As with our algorithm, the template matching is not used for a precise segmentation, but for an approximate location of the indentations. Figure 4.14 shows, that the performances (cumulative curves) of the methods are quite similar. However, our proposed gradient descent method is slightly more reliable.

4.2 Shape Prior level set with gradient descent

Although the proposed strict Shape-Prior (section 4.1), in combination with the balloon force introduces a high level of noise resistance, our scenario requires more precise results. Precise results cannot be achieved, as real objects slightly neglect the Shape-Prior restriction (i.e. the shape of the

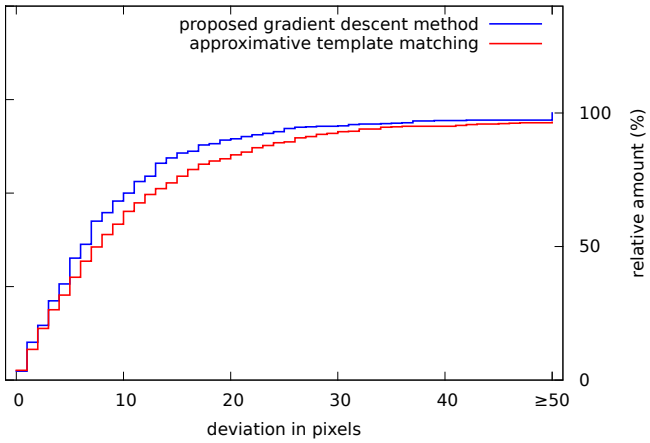


Figure 4.14: Comparison of the proposed strict shape prior gradient descent method with template matching

object can not exactly be described by a square).
Consequently it is a good idea to use a strict Shape-Prior approach as a first stage on a downscaled image to get an approximation of the result points, and to use a non Shape-Prior algorithm, which is much more liable to initialization, as a second step to get detailed results. This approach is investigated in chapter 6.
Even though that might be a good idea, using the non Shape-Prior level set algorithm has some problems:

- A small smoothing (curvature) term leads to exact segmentation, but makes the algorithm vulnerable to noise. Although accurate initialization can reduce effects, noisy images are still problematic.
- A large smoothing term decreases the vulnerability to noise, but leads to a loss of accuracy, because edges as well as small details distinguish.

In order to get both, accuracy and robustness, a Shape-Prior level set method which is scale, shift and rotation invariant is used, that has been introduced in [3].

4.2.1 Functionality

In addition to the evolution of the level set function ϕ another level set function ψ is introduced, which represents the prior shape. For most applications the prior shape cannot be represented as a simple fixed level set, as shifting in x and y direction, scaling as well as rotation should be allowed. To archive such an invariance at each evaluation step the current prior function must be evaluated. One way to evaluate this function is gradient descent (in [3]),

as brute force search is not acceptable as far as computational complexity is concerned.
After evolving the Shape-Prior parameters, the traditional energy function (e.g. Chan-Vese functional in [3]) is supplemented with the shape energy E_{shape}.

$$E = E_{traditional} + \lambda \cdot E_{shape} \tag{4.21}$$

E_{shape} is defined by following equation:

$$E_{shape}(\phi, \psi) = \int_{\Omega} (H(\phi) - H(\psi))^2 dx \tag{4.22}$$

$$H(\phi) = \begin{cases} 1, & \text{if } \phi \geq 0 \\ 0, & \text{else} \end{cases} \tag{4.23}$$

The shape energy term is low if the two level set functions ϕ (evolving curve) and ψ (Shape-Prior) are congruent and high otherwise. As the Shape-Prior should be invariant to shifting $(\bar{x}, \bar{y})$, rotation $(\bar{\alpha})$ and scaling $(\bar{r})$, ψ^* is a variable:

$$\psi = \psi^*(\bar{x}, \bar{y}, \bar{r}, \bar{\alpha}) \tag{4.24}$$

Consequently, the following term is minimized by the gradient descent algorithm:

$$E_{shape}(\phi, \psi) = \int_{\Omega} (H(\phi) - H(\psi^*(\bar{x}, \bar{y}, \bar{r}, \bar{\alpha})))^2 dx \tag{4.25}$$

Minimizing the respective term means finding the best fitting parameters for the prior shape in order to achieve the lowest shape energy.

As gradient descent is vulnerable on the one hand and computationally expensive on the other hand, we did not apply this approach but concentrated on the approach introduced in the following chapter, which is an improvement of this approach.

4.3 Shape-Prior level set: direct computation

Gradient descent of a prior level set function suffers from computational complexity (surely a brute force search is much more complex), but also the convergence to the global optimum cannot be guaranteed, as the search space might be deceptive. Consequently, an a priori calculation of the prior function would be advantageous.
In [27] a method was suggested to evaluate parameters of circles, which are known to be rotation invariant. This makes the evaluation easier, as only

scale and shift has to be known. The calculation of squares that are not rotated is presented in the following equations:

$$\bar{x} = \tilde{\pi_x}(x, y) \tag{4.26}$$

$$\bar{y} = \tilde{\pi_y}(x, y) \tag{4.27}$$

The median ($\tilde{}$) of the projection π_x ($\pi_x(x, y) = x$) is assumed to be the center of the x axis. The median of π_y is assumed to be the center of the y axis.

$$\bar{r} = \sqrt{\int_\Omega H(\phi)}/\sqrt{2} \tag{4.28}$$

The half length of a diagonal is labeled as $\bar{r}$ ("radius"), which is calculated in a way that the area of the prior shape is equal with the contour's area.

In order to allow rotation of the prior shape, gradient descent has to be applied, as rotation ($\bar{\alpha}$) cannot be calculated in an easy way like the other parameters. The following term must be minimized, with respect to $\bar{\alpha}$ (other parameters are known):

$$E_{shape}(\phi, \psi) = \int_\Omega (H(\phi) - H(\psi^*(\bar{x}, \bar{y}, \bar{r}, \bar{\alpha})))^2 dx \tag{4.29}$$

As gradient descent of $\bar{\alpha}$ is quite expensive as far as computational costs are concerned, it is recommended to ponder if rotation is really necessary. If rotation of the prior shape is not calculated, slight rotations of the contour do not cause any troubles. Only if the rotation is quite distinct, segmentation might suffer (depending on the weight of the Shape-Prior λ).

One step of gradient descent of $\bar{\alpha}$ is given by the following instruction:

$$\bar{\alpha} = \bar{\alpha}' + \iota(\frac{dE_{shape}}{d\bar{\alpha}}) \tag{4.30}$$

In our implementation we especially used the following function for ι. This ensures, that the computational costs do not expand extremely, which might happen if the step size would be very small.

$$\iota(x) = \begin{cases} 0.1, & \text{if } x > 0 \\ -0.1, & \text{if } x < 0 \\ 0, & \text{if } x = 0 \end{cases} \tag{4.31}$$

4.3.1 Weight of the Shape-Prior

When applying prior knowledge of the evolved shape, an important issue is the determination of the weight parameter λ. While a small λ leads to small influence of the prior shape, large λ effects higher influence, i.e. the evaluated shape tends to be more similar to the prior. Figure 4.16 shows the evolution of a circular shape (figure 4.15 left) using a square Shape-Prior (figure 4.15 right). Increasing the prior weight λ influences the output of

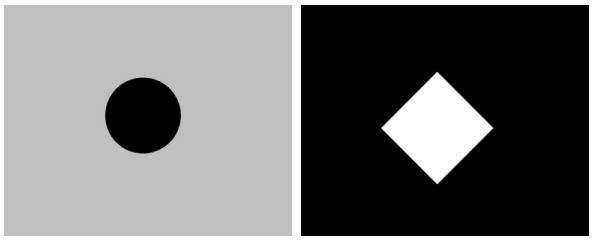


Figure 4.15: Shape to segment and prior shape

the level set method (figure 4.16). While when using very little weight, the Shape-Prior does not really influence the evolution and the result equals the real shape, heavier weight restricts segmentation more.

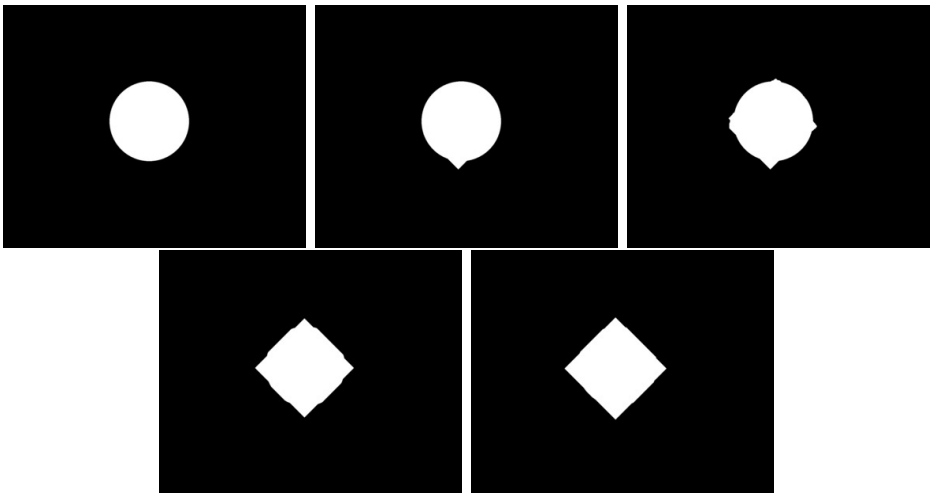


Figure 4.16: Different Shape-Prior weights

Although this example has no practical relevance, it serves as a good example. For real Vickers images, the prior shape and the actual shape do not differ that much. The prior is like a kind of smoothing term, that sustains the object corners (in opposite to the curvature term).

There are different ways to choose the weight λ:

- Strategy 1: One way to introduce a Shape-Prior into a batch segmentation algorithm (i.e. no human interaction) is to choose λ fixed for all images.

$$\lambda = \lambda_0 \tag{4.32}$$

- Strategy 2: As the traditional force (e.g. Caselles, Chan-Vese) depends on differences of gray values, λ should be adjusted so that images with smaller gradient values are equally influenced by the Shape-Prior as images with higher gradients. Otherwise low contrast images would be stronger influenced by the prior shape than high contrast images.

$$\lambda = \lambda_0 \cdot diff_{image} \tag{4.33}$$

- Strategy 3: Moreover image quality can be considered when λ is calculated, because low quality images are likely to be segmented totally wrong, whereas high quality images usually are not. The introduction of a (highly weighted) Shape-Prior leads to less tweaked results, but the results are more robust (fewer outliers).

Using the Chan-Vese approach, the following calculation for λ is used:

$$var_{intra} = \frac{1}{|\Gamma_{in}|}\int_{\Gamma_{in}} |I(v) - \bar{I_{in}}|dv + \frac{1}{|\Gamma_{out}|}\int_{\Gamma_{out}} |I(v) - \bar{I_{out}}|dv \tag{4.34}$$

$$var_{inter} = |\bar{I_{in}} - \bar{I_{out}}| \tag{4.35}$$

$$\lambda = \lambda_0 \cdot var_{intra} \cdot var_{inter} \tag{4.36}$$

Using the Ofeli C++ level set image segmentation library, $\lambda_0 = \frac{1}{2000}$ turned out to be a good configuration.

4.3.2 Exposure problem

Calculating the best fitting prior shape for a given level set has one more advantage, as Vickers images have got one challenge: Image borders often are quite dark (due to exposure). As the standard Chan-Vese model utilizes each pixel outside of the object to calculate the average gray value, we improved this inconvenience, by altering the calculation slightly. Originally, the region outside of the object is given by the following equation:

$$\Gamma_{out} = \{(x, y) : r > r_0\} \tag{4.37}$$

$$\begin{aligned} r = &|(x - x_0) \cdot cos(\alpha) + (y - y_0) \cdot sin(\alpha)| + \\ &|(x - x_0) \cdot sin(\alpha) - (y - y_0) \cdot cos(\alpha)| \end{aligned} \tag{4.38}$$

The new version only regards pixels nearby the object to be "outside" pixels:

$$\Gamma^*_{out} = \{(x, y)|r > r_0 \wedge (x - \bar{x})^2 + (y - \bar{y})^2 < 2 \cdot \bar{r}^2\} \tag{4.39}$$

As mentioned before, the prior shape has to be calculated to get $\bar{x}$, $\bar{y}$ and $\bar{r}$. We renounced to calculate the exact rotation of the object (which would be computationally expensive), because the shape of the new "outside region" is circular, and is consequently not influenced by rotation.

4.3.3 Analysis on indentation images

Without solving the exposure problem (section 4.3.2), introduction of prior knowledge definitely improves the segmentation results. Especially the number of outliers can be minimized. The weight calculation strategy 3 turned out to be the best option.
As the exposure problem has been solved, segmentation generally increased and applying Shape-Prior weight no longer improves overall results. However, for specific noisy images, applying the prior knowledge is still beneficial. But as overall result could not be improved, our analysis did not rely on Shape-Prior. Nevertheless, solving the exposure problem requires the calculation of the best fitting prior shape.

Chapter 5

Pre- and postprocessing of images

5.1 Preprocessing: Blurring as noise reduction

The algorithm with strict Shape-Prior is highly resistant to noise, because no single pixel is considered for one evaluation step, but always a set of pixels. Such a policy does not require blurring at all, as noise resistance is intrinsic as large areas are regarded instead of single pixels.
In contrast, level set algorithms do not regard a set of pixels for one step, but only border pixels. To prevent this methods from over-segmentation, level set methods contain a smoothing term (curvature term), which planes the contour. As a high curvature term leads to a bad segmentation of corners, a Shape-Prior force has been introduced, which also smoothes the contour but also helps sustaining the intended shape's characteristics.

Another approach to get a higher level of continuity and to reduce noise in order to prevent the method from over-segmentation is to blur the image. While the Shape-Prior introduces high level knowledge (of the shape), blurring only reduces high frequency information (noise). As object frontiers also represent high frequency information they are affected by blurring. But on the one hand blurring is not done excessively, and on the other hand the region based model (Chan-Vese), does not depend on gradients of the image.

In chapter 6, we propose a multi-resolution algorithm which approximately segments downscaled images (factor 10) as a first step (with gradient descent method) and exactly segments original images with precomputed results (and a level set algorithm) as a second step.

Images for the first step of the segmentation are not blurred at all. First of all, these images are downscaled (by averaging with a Lanczos filter),

which reduces high frequency information (noise) as well. Moreover, as algorithms work on large areas, averaging suppresses the residual noise.

As the second step operates on high resolution images (1280x1024) and the corresponding algorithms (level set, snake) work on single pixels, blurring should be considered. Experience proves that blurring acts similar to raising the curvature term of level set algorithm.

Tests where done using different blurring operators. Blurring operators of sizes from 3 to 20 pixels have been convolved with original images using the Image-Magick blur function. Performance tests show that blurring does not lead to better results. However, the results are quite similar. As blurring implies computational costs, we do not preprocess our images in that way.

5.2 Postprocessing and adjustment of algorithms

After applying any kind of level set or snake algorithm a decision must be made in order to identify the corners of the objects. Unfortunately, depending on the configuration, objects are not always real squares, but often incomplete or ragged.

In the following sections different solutions are discussed:

5.2.1 Extreme points

One straightforward approach is just to search for the highest, the lowest, the right most and the left most point. As the diagonals of the objects are aligned horizontally and vertically, if segmentation succeeded the calculated points are the corners of the indentation (figure 5.1, left).

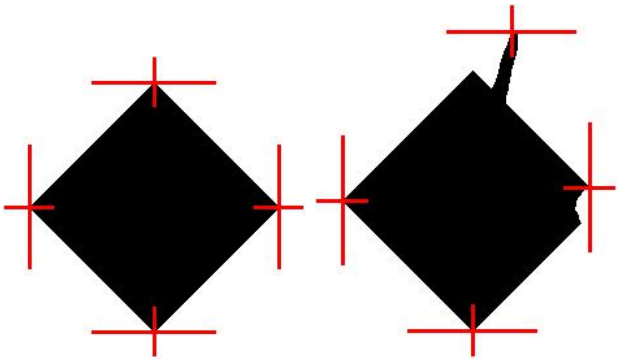


Figure 5.1: Extreme points: success (left) and failure (right)

That quite simple approach supplies acceptable results, if the image is segmented well. That means that the border must not be ragged, which

cannot always be assured. Two problematic cases has been observed (figures 5.1 (right), 5.2):

- Artefacts affect the calculation of the corner points. Such artefacts can be reduced by applying either a higher curvature term, or a Shape-Prior weight. However, doing this is known to affect the accuracy. Increasing the curvature weight especially raises the problems outlined next.
- Cutting of corners affects results. Often corners are not segmented exactly, but approximated by a curve instead of a sharp corner.

Figure 5.2: Artefacts and cut corners

5.2.2 Adjust level set algorithm - smoothing term

An adjustment can be done by configuring the algorithm: To prevent the level set from evolving into a ragged surface, the curvature term can be increased. But a consequence of a high curvature term (i.e. that curvature becomes heavier weighted than the image energy) is that corners are more likely to be cut.

5.2.3 Adding Shape-Prior weight

An improvement can be attained by the use of a Shape-Prior energy. This not only smooths, but also helps maintaining corners, in opposite to increasing the smoothing term.

5.2.4 Morphologic operations

Mathematical morphology provides erosion ($\ominus$) and dilation ($\oplus$) as basic operations on binary images.
These simple operations are not sufficient for many applications, as erosion

just removes pixels from the image which leads to a decrease of object size, whereas dilation suffers from the opposite effect. Consequently erosion and dilation is usually used in combination:

- opening:
  $I \circ Y = (I \ominus Y) \oplus Y$
  Opening removes small artefacts and objects.

- closing:
  $I \bullet Y = (I \oplus Y) \ominus Y$
  In opposite to opening, closing adds area to the foreground object, especially holes are filled.

I denotes the image and Y denotes the morphologic operator.
To achieve higher impact, the morphologic operations can be iterated in the following way:

$$I \ominus^n Y = (...(I \ominus Y)... \ominus Y) \ominus Y \tag{5.1}$$

$$I \oplus^n Y = (...(I \oplus Y)... \oplus Y) \oplus Y \tag{5.2}$$

$$I \circ^n Y = (I \ominus^n Y) \oplus^n Y = (...(I \ominus Y)... \ominus Y) \oplus Y)... \oplus Y) \tag{5.3}$$

$$I \bullet^n Y = (I \oplus^n Y) \ominus^n Y = (...(I \oplus Y)... \oplus Y) \ominus Y)... \ominus Y) \tag{5.4}$$

It is necessary to consider the morphologic operator used. The shape of Vickers objects suggests the use of a diamond operator in order to sustain edges, since (standard) circle operators would affect corners (figure 5.3). With a large number of iterations, the effect is reinforced.

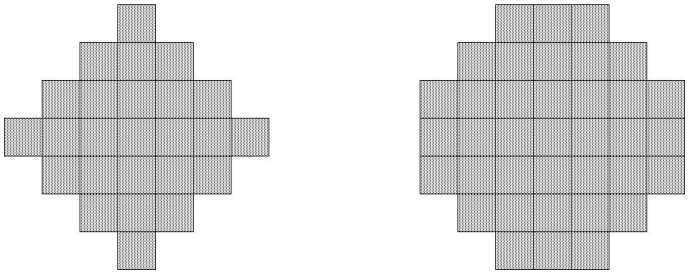


Figure 5.3: Different morphological operators (left diamond, right circle)

The following issues should be solved by morphologic operations:

- Filling cut edges (sometimes even holes):
  The closing procedure is known to fill holes. Unfortunately, in our case closing would not only have to fill a hole, but also have to reconstruct the corners of the object. As shown in figure 5.4, closing is able to

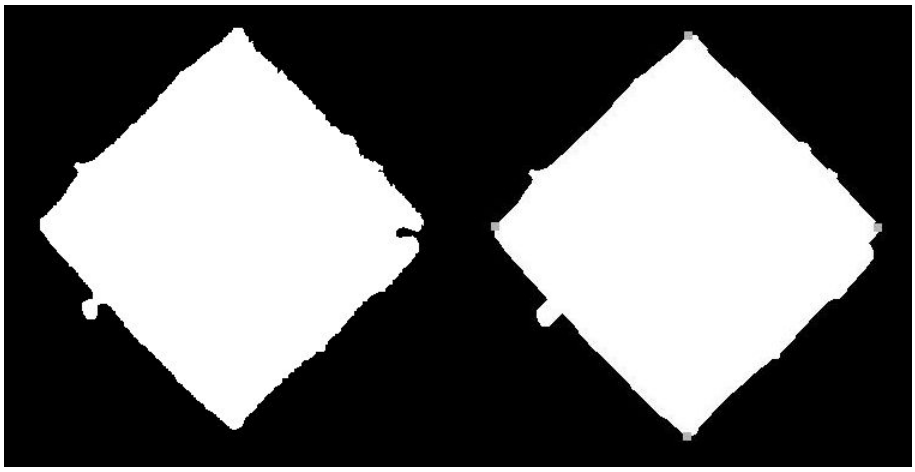


Figure 5.4: Closing with diamond operator (left original, right closed)

fill holes, but it definitely is not able to reconstruct objects corner. Although when a diamond operator is used, closing does not solve this problem.

- Another issue is removing artefacts produced by the level set algorithms:
  The opening procedure is known to remove such noise. Figure 5.5 shows, that opening actually succeeds to remove the artefacts. However,

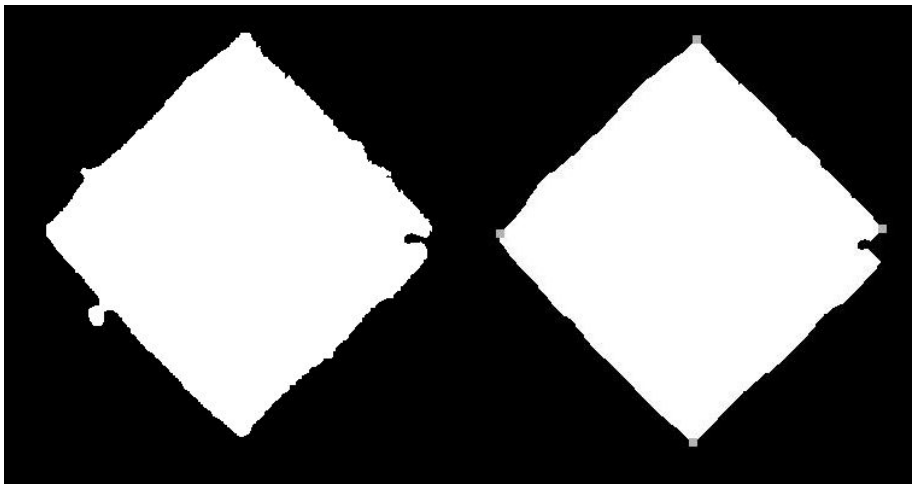


Figure 5.5: Opening with diamond operator (left original, right opened)

we have to be aware that opening not only removes noise. Also useful information might be affected, especially corner pixels, which are most important (as the locations of this points should be calculated).

Unfortunately closing does not succeed to reconstruct missing corners. On the other hand, opening is able to remove noisy artefacts. But as perfect segmentation would require both exercises, morphologic methods alone cannot deliver the desired results.

One idea is to first close the segmented image, and afterwards apply a Hough transform locally (sections 5.2.5, 5.2.6). Although this seems to be a good idea, results could not be improved significantly (compared to applying only the Hough transform). Performance stayed about the same. As morphologic operations are computationally expensive, we do not use them for Vickers segmentation.

5.2.5 Hough transform global

While some Vickers segmentation algorithms rely on Hough transform [10] alone, we investigated the method as a post processing method. Instead of improving and adjusting the shape of the object, the Hough transform calculates probable lines that cross many edge pixels. The corner points should be situated, where these lines meet each other.
One simple approach is to calculate the Hough transform of the whole image. As a result, most probable lines are delivered. In order to get the desired lines, we extract first two parallel lines and then two lines that are normally directed to the first lines. Moreover it is important to prevent getting lines which are very close.

The lines are represented by an angle θ and a distance r from the origin (figure 5.6).

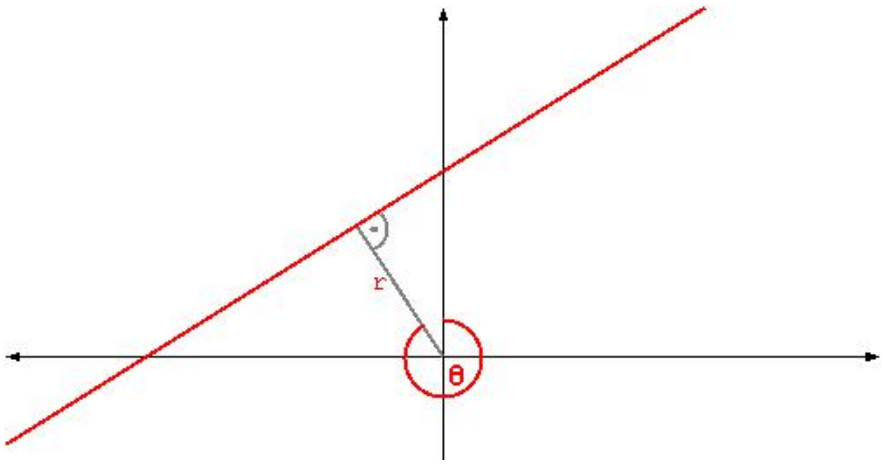


Figure 5.6: Representation of straight lines

The 4 result lines are chosen by the following strategy:

- Line 1: The line with the most edge points on it.
- Line 2: The line with the most edge points on it and with a difference of rotation smaller than 0.2 ($|\theta_1 - \theta_2| < 0.2$) and a normal distance of greater than 10 pixels compared to line 1.
- Line 3: The line with the most edge points on it and with a difference of rotation to a normal line to line 1 of smaller than 0.2.
- Line 4: The line with the most edge points on it and with a difference of rotation smaller than 0.2 and a normal distance greater than 10 pixels compared to line 3.

5.2.6 Hough transform local

Unfortunately, Vickers objects cannot be represented exactly by a square consisting of four straight lines, as most edges are slightly arced or noisy. To prevent this issue, our proposed local Hough approach just approximates lines in a defined distance from the precalculated corner points.
This allows a better approximation especially if the image borders were evolved quite exactly by a level set method.
In the first step, the Hough transform is computed separately for each corner in a range (radius) of e.g. 60 pixels. In the second step, the right two lines of the Hough transform have to be selected. The real corner is assumed to be on the intersection of the selected two Hough lines (figure 5.7). The line representation shown in figure 5.6 is utilized.

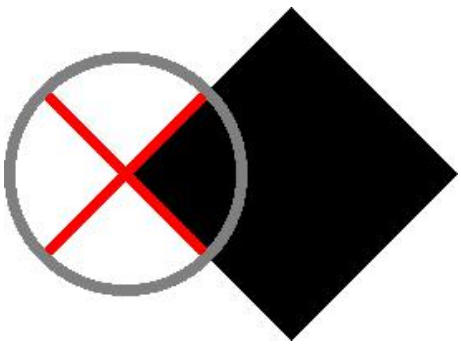


Figure 5.7: Local Hough transform applied to leftmost corner

Two different strategies of selecting the two lines where evaluated:

- The first line is the one with the best Hough rating and the second line is the one with best Hough rating that meets the following constraint:

$$||\theta_1 - \theta_2| - \pi/2| < 0.3 \tag{5.5}$$

θ_1 belongs to line 1 and θ_2 belongs to line 2. Although this method is quite simple, it works best and was chosen for our segmentation.

- Another idea is to maximise the following energy terms:

$$E_1 = \alpha \cdot \frac{hough_i}{hough_{max}} - \beta \cdot ||\theta_i| - 45^o| \tag{5.6}$$

$$E_2 = \alpha \cdot \frac{hough_i}{hough_{max}} - \beta \cdot ||\theta_i - \theta_1| - 90^o| \tag{5.7}$$

While the first term is influenced by the (relative) Hough rating $hough_i$ (the greater the better), the second term depends on the angles of the lines (difference to perfectly aligned square). Whereas the angle of the first line selected should be similar to $45^o (= \pi/4)$, the angle of the second line should be about 90^o distant from the first line's angle. Although this approach is quite flexible and was tested with different weighting parameters α and β advantages to the simple first approach could not be achieved. Consequently the first selection strategy is used in the following.

Figure 5.8 shows the impact on the global Hough transform if the object sides are not perfectly straight (green). The red lines are achieved with the local Hough transform (radius = 60 pixels).

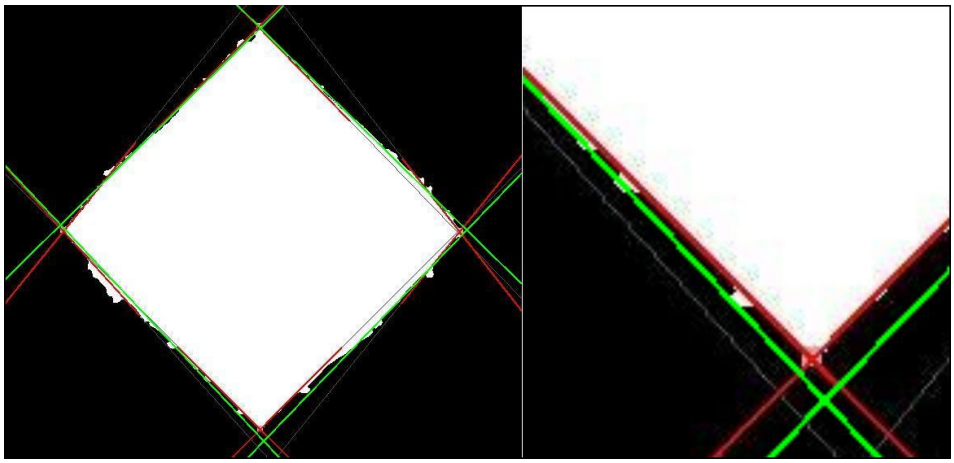


Figure 5.8: Local (red) vs. global (green) hough transform

5.2.7 Approximation of curves

Another way to reconstruct corners is to approximate curves instead of lines. Especially if one line is calculated per side (global Hough transform), approximation of curves could improve the performance. But as we use a local

Hough approach, which provides accurate results (in the case of acceptable approximation), we did not consider the use of curves.

5.2.8 Analysis on indentation images - best strategy

The local Hough transform as mentioned, in most cases is able to locate the corner points accurately. When the image is segmented appealingly, the computed output is highly acceptable. Points are located far away from actual points' position only if the active contours segmentation produces an unacceptable output. In this case mostly even humans would fail to locate the points accurately. Consequently, as postprocessing only regards the segmented image, a failure of the segmentation algorithm cannot be corrected. So we are satisfied with the local Hough strategy.

If especially very bad results should be improved, a quality measure of segmentation's output could be used to identify those images which failed the correct segmentation. In a second step, such difficult images could be segmented again, by adjusting parameters to emphasize on robustness more than on high accuracy. This might be done by increasing the Shape-Prior weight, the smoothing factor, the Hough radius or by blurring the image before segmentation. The main issue is to judge the segmentation's result, as the real results for comparison are naturally not available.

One idea is to calculate the best fitting prior shape for a given output level set and to calculate differences from that shape. If a (high) Shape-Prior weight is included, this surely does not provide any accuracy, as the level set is forced (depending on the weight) to change its shape to the prior shape. Doing this calculations only the result level set is regarded. Another way is to include the input image, and to get a quality index by the response to a template, which fits the output level set. This step has not been investigated so far.

Another way to improve the performance especially for difficult images is to calculate "difficulty" even before segmentation. In chapter 4, an attempt was done to realize a flexible algorithm. The weight of the Shape-Prior energy depends on the gray value variation inside, the gray value variation outside and the difference of the average gray values. Segmentation results of some images could be improved. However, the overall performance decreased.

Chapter 6

Iterative multi-resolution approach

On the one hand, the strict Shape-Prior image segmentation (on downscaled images) leads to a good approximation of image contours with little influence of the starting configuration (with balloon force we always start with the biggest radius). On the other hand different level set approaches provide very exact segmentation, but hugely depend on the starting configuration. Consequently a combination of those two different, complementary approaches is investigated.

We propose the following policy:

- Stage 1: Apply a strict Shape-Prior approach on the downscaled images, (factor 10) to reduce complexity, but maintain the basic image features.
- Upscale the calculated results and generate an initial level set for the second stage (level set method), by using the results from stage 1.
- Stage 2: A level set approach without (or with soft) Shape-Prior weight now starts computation with the upscaled results on the original image.

6.1 Stage 1

As stage 1 segmentation algorithm, the statistical gradient descent method has been chosen, which provides the best results (section 4.1). Best results means that the most of the detected corners where situated within a threshold distance of 5 pixels in the downscaled image. That makes a distance of up to 50 pixels in the original image. The chosen method is able to detect 97.5 % of all corners within that distance in the database declared in section 1.2. For exact results look at section 4.1.

6.2 Stage 2

For stage 2 different types of level set and snake algorithms with different setups were used:

- traditional snake
- level set Caselles
- level set Chan-Vese
  - different smoothing terms
  - with/without morphologic operations (open, close, open/close, close/open, different operators)
  - with/without Hough transform (global, local)
  - different Shape-Prior weights

6.3 Analysis on indentation images

The multi-resolution approach has been evaluated using different stage 2 algorithms (see below). The following figures show results of different strategies. For each deviation in pixels (i.e. the distance between calculated and actual point), the number of edge points with (exact) the respective distance to the real points are shown (red bars). The right most bar collects all outliers (distance greater than 19 pixels). The blue curve represents the relative cumulation, i.e. for each deviation the ratio of points with equal or less deviation.

6.3.1 Traditional active contours (snake)

Using the straightforward OpenCV snake implementation as stage 2 algorithm, the results shown in figure 6.1 can be achieved.

The best configuration of the snake approach produces the following results (using local Hough transform (radius = 60 pixels)). Especially the probability of very exact results (deviation $0 - 1$ pixels) can be increased with the Hough transform, because the traditional snake has the tendency to cut the corners of objects during segmentation. Applying the Hough transform, the cut corners can be reconstructed well. The results are shown in figure 6.2.

The snake algorithm is not always able to detect corners perfectly, but Hough transform is able to compensate this inadequacy (as shown in figure 6.3). The white dot marks the new adjusted corner point after the Hough

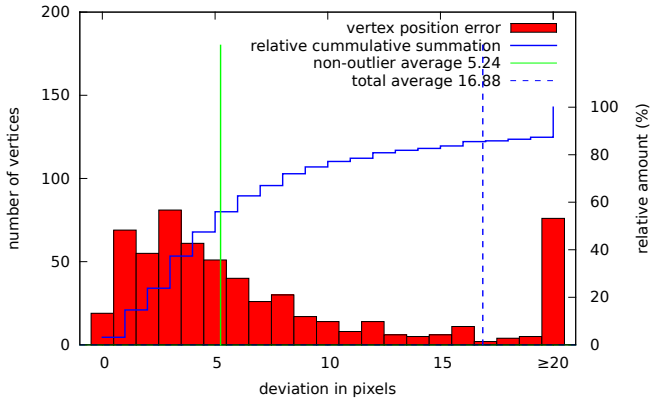


Figure 6.1: Snake results

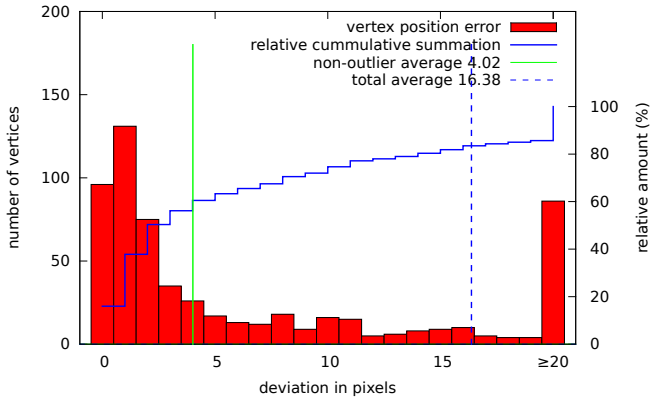


Figure 6.2: Snake results with Hough transform

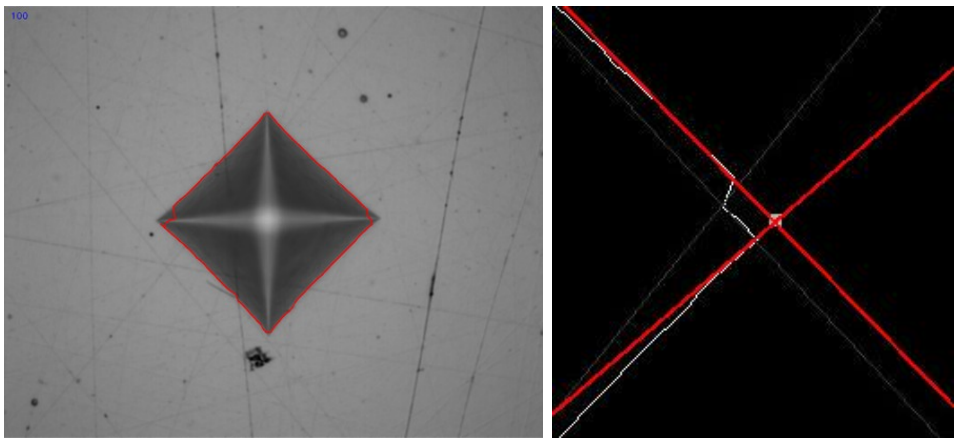


Figure 6.3: Improvements of Hough transform

transform.

Figure 6.4 shows three steps of the evaluation. In the first step (left), stage 1 of the multi-resolution algorithm has been applied, i.e. a good approximation is calculated to serve as starting configuration for the snake. In the middle you can see the output of the snake algorithm. Finally, the Hough transform is applied (right).

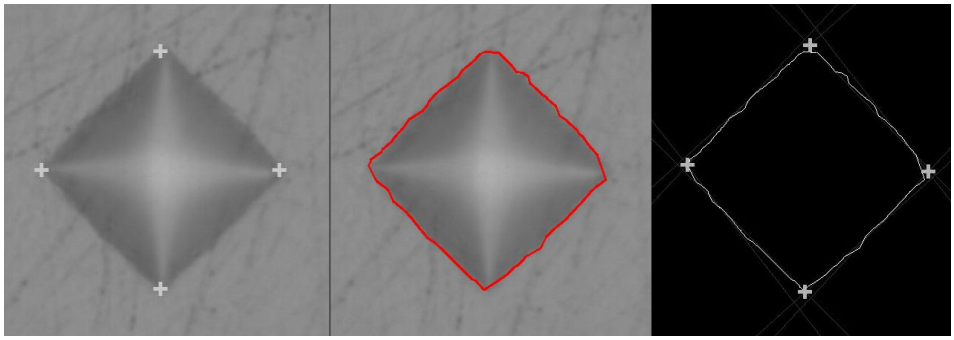


Figure 6.4: Three steps of computation

6.3.2 Level set approach

The best results are achieved using the region based Chan-Vese algorithm.

The results of the Chan-Vese level set method, without a smoothing Shape-Prior and any image pre- or postprocessing are shown in figure 6.5.

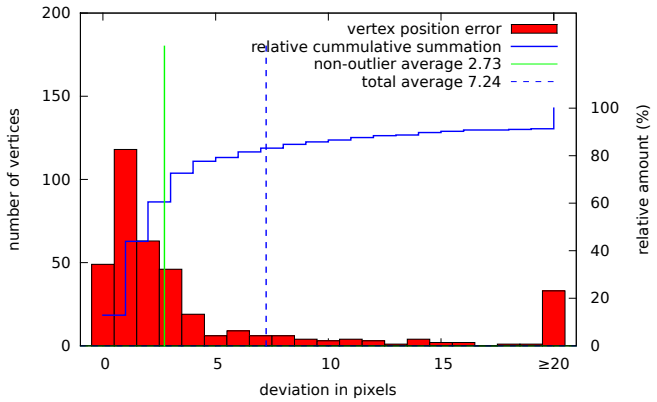


Figure 6.5: Level set approach, $\lambda = 8$

Introducing the Hough transform (local, radius = 60 pixels) robustness is inclining. This is because as with the snake, also the level set algorithm

(especially without Shape-Prior) has the tendency to cut corners. Figure 6.6 shows the results with smoothing term $\lambda = 8$.

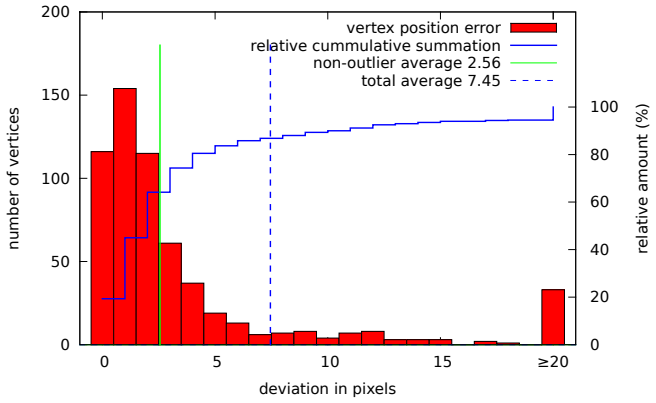


Figure 6.6: $\lambda = 8$, hough transform (60 pixels)

Further improvements can be achieved (figure 6.7) by restricting the Chan-Vese model from regarding the whole image at each iteration step. When only regarding a circle around the object to segment, the dark background at the image boundary does not affect the segmentation. Shape-Prior is not applied.

We propose using this configuration to get the best results, with limited

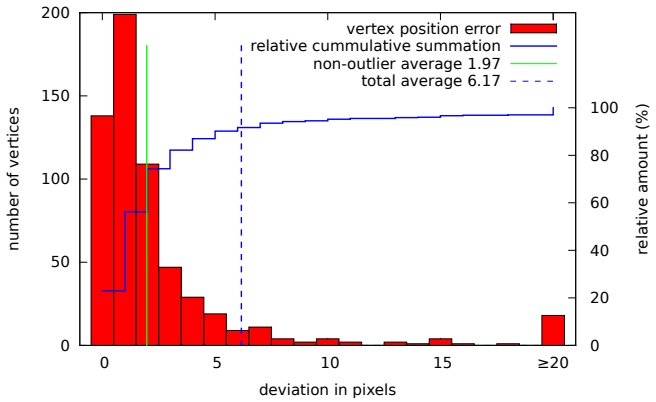


Figure 6.7: Smoothing term $\lambda = 8$, hough transform (60 pixels)

computational costs.

Applying a small Shape-Prior weight does not influence the results strongly. Without the improvement that the pixels outside are not regarded by the region based algorithm (which is based on the calculation of the best fitting prior shape), Shape-Prior definitely was advantageous, but now results are

nearly the same. Surely, if the Shape-Prior gets too strong, results suffer. However, the best fitting prior shape has to be computed, in order to ignore pixels far away from the object.

Figure 6.8 shows differences between applying no Shape-Prior (left side) and a strong Shape-Prior (right side). With a Shape-Prior, edges are more likely to sustain in the segmentation process. But we have to be aware of the risk of under-segmentation. A very strong Shape-Prior might imply that only perfect squares are evolved and peculiarities of objects get lost.

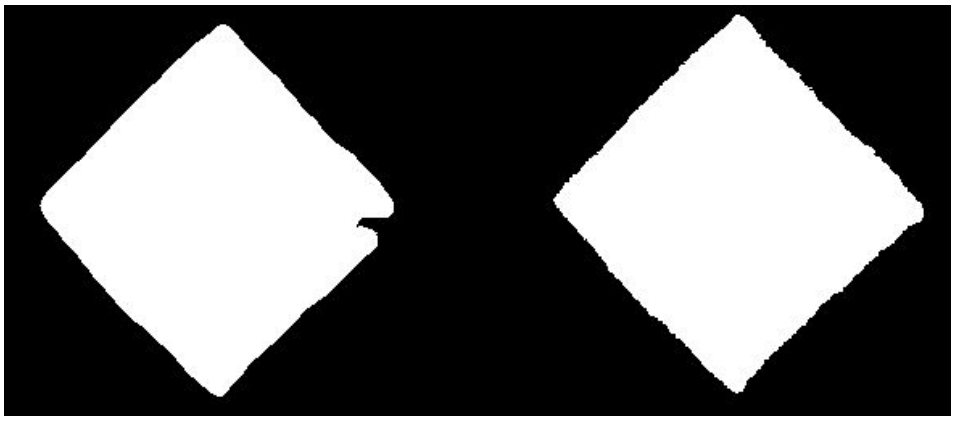


Figure 6.8: Effect of the Shape-Prior

Figure 6.9 shows the steps of the multi-resolution algorithm. First the original image is downscaled by factor 10 and the approximations of the points are calculated (left part). After that the level set algorithm is applied, which generates a binary image (middle part). The postprocessing step (local Hough transform), generates exact output points.

6.3.3 Reference results

Now we compare the performance of our proposed multi-resolution algorithm with an existent approach. The results, presented in figure 6.10 can be achieved by the multi-resolution template matching algorithm (introduced in [7]), which is known to be very robust.

In figure 6.11 you can see that the proposed multi-resolution level set algorithm combined with the local Hough transform definitely is competitive as far as segmentation performance is concerned. The chart compares the cumulative distribution of the proposed level set method (red) and the reference template matching method (blue):

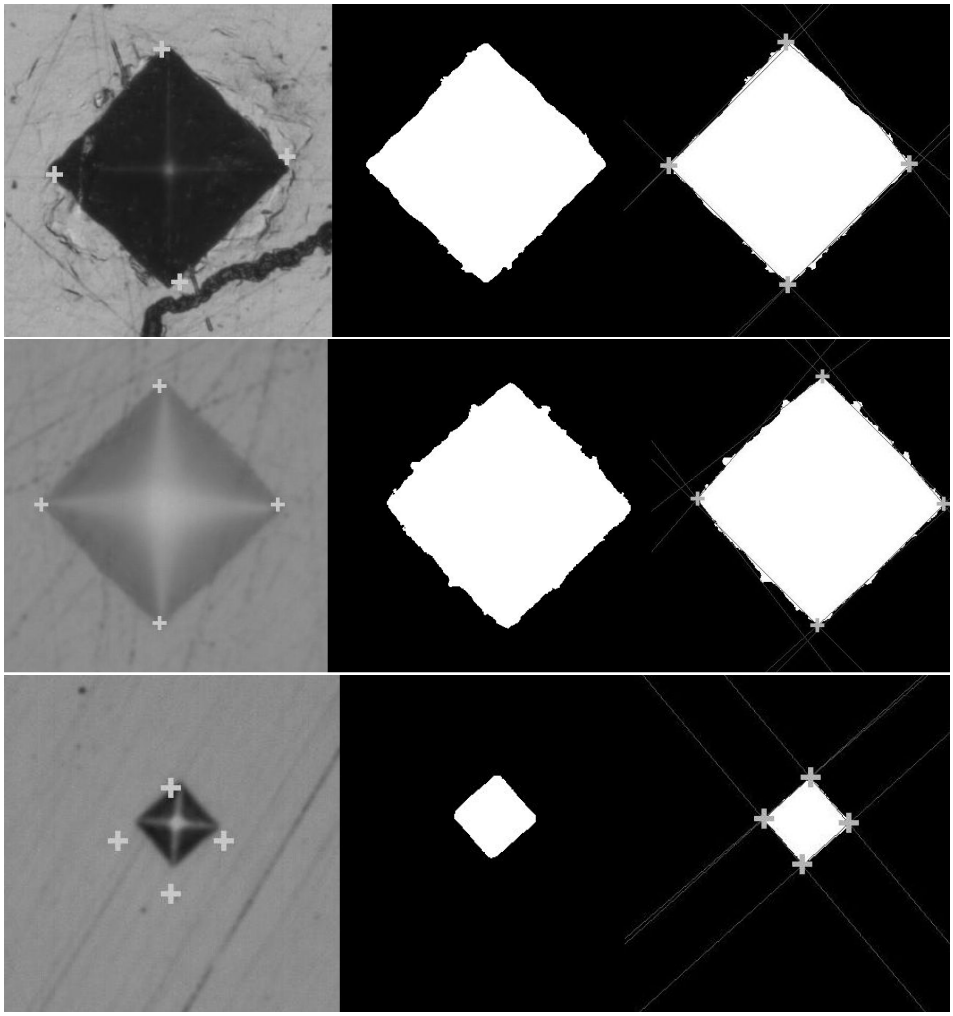


Figure 6.9: 3 stages of multi-resolution algorithm

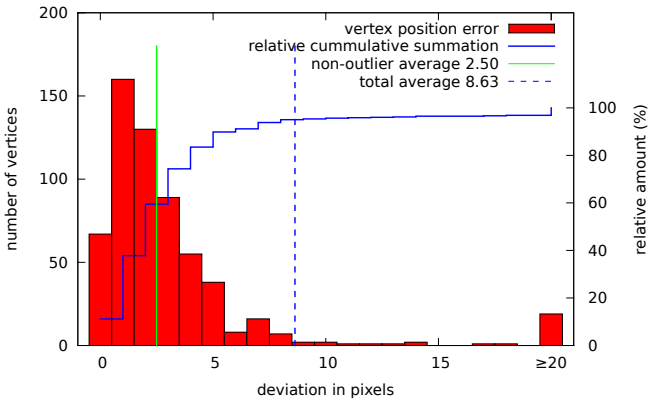


Figure 6.10: Reference results

Especially the probability of a very exact segmentation of the corner points

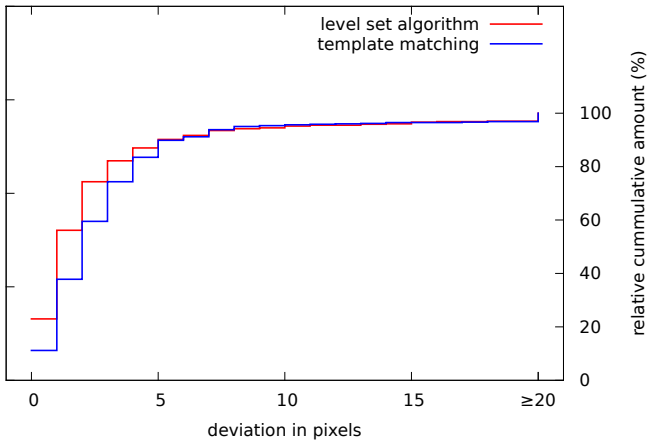


Figure 6.11: Comparison of new approach vs. multi-resolution template matching

(Euclidean distances of 0 to 5 pixels) is considerably higher with our new proposed method. The number of outliers is the same.

Moreover, we compare the results with an even more precise 3-stage segmentation method proposed in [8]. The results of this method are shown in figure 6.12.

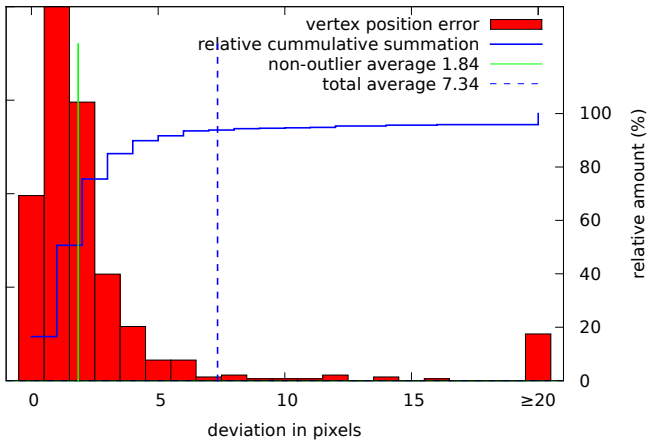


Figure 6.12: 3-stage segmentation approach

The cumulative curves of our proposed multi-resolution approach and the 3-stage approach are compared in figure 6.13. As the curves are crossing, a winner cannot be determined. The new multi-resolution level set based approach is slightly more exact (deviations < 2 pixels) and the number

of outliers is slightly lower. However, the number of corner points detected within a deviation of $3-5$ pixels is slightly higher with the 3-stage approach.

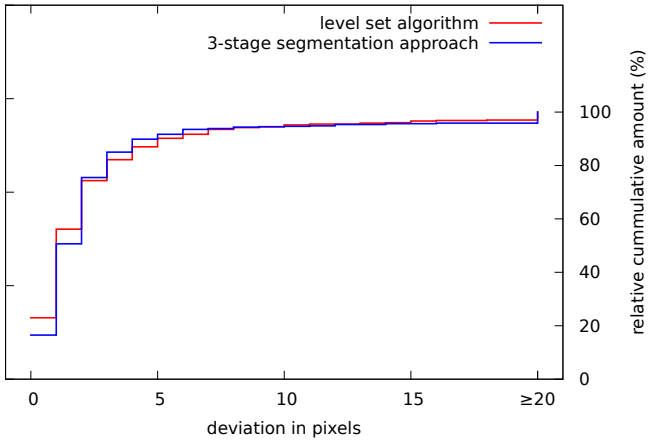


Figure 6.13: Comparison of new approach vs. 3-stage segmentation approach

6.3.4 Outliers

Now we know that the outliers ratio of our proposed multi resolution approach is the same as the outliers ratio in the referenced template matching approach [7] (outliers ratio in [8] is slightly higher). However, we still have not investigated, what types of images are likely to fail the segmentation processes. First of all, figure 6.14 shows images which cannot be segmented precisely by any of the two approaches. The left two images suffer from heavy noise. The third one is difficult to segment, because of the thick stripe in the left part of the image. The right most image borders cannot be determined, because the noise in the lower half is darker than the imprint.

Next we would like to know, if there are any images which can be segmented by one of the algorithms, but not by the other one. Actually there are such images. The segmentation of the images in figure 6.15 fails, if the template matching approach is applied, but not if the multi-resolution active contours algorithm is applied. In all of these images, the template matching method determines the corners to be placed farther outside than the actual corners are. The problem is, that only the response of the template is considered. In opposite, our proposed statistical first stage algorithm focuses on homogeneity (i.e. low entropy).

The images in figure 6.16 cannot be segmented by the proposed multi

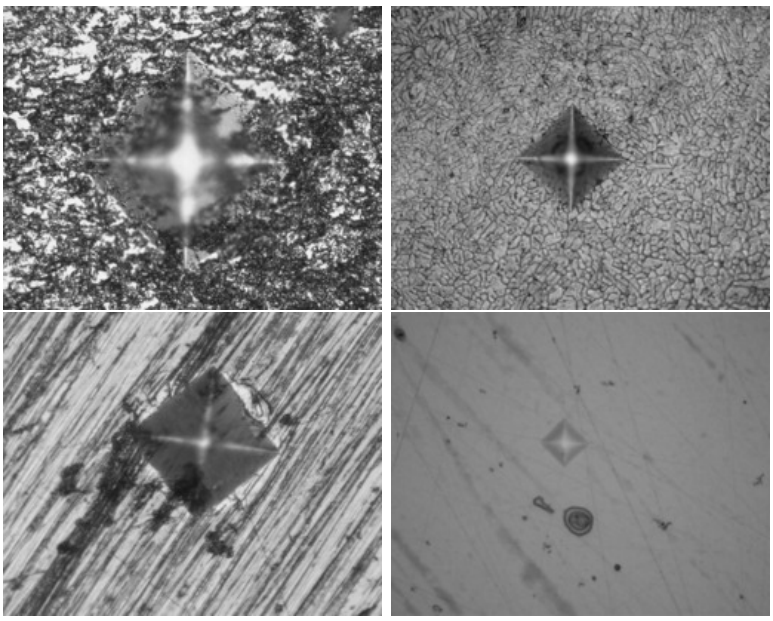


Figure 6.14: difficult images: both approaches fail

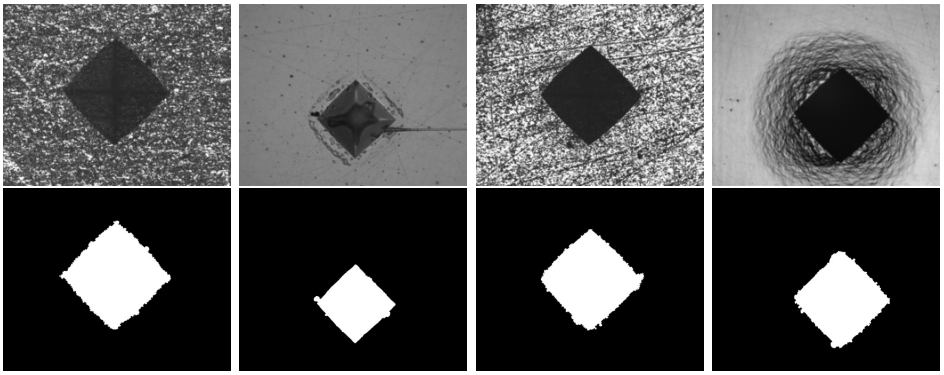


Figure 6.15: The template matching approach fails to segment the top images appropriately. The bottom level sets are successful outputs of our new proposed level set algorithm. The remaining artefacts are compensated by the Hough transform.

resolution algorithm. The left two images are highly noisy, whereas the right two image look quite nice. Actually, these two images are difficult to segment because the artefacts are similarly colored as the object. Consequently, the level set algorithm cannot distinguish the object from the artefacts. The performance could be improved by adding the introduced Shape-Prior weight. However, the superior segmentation performance of the level set algorithm for high quality images would suffer.

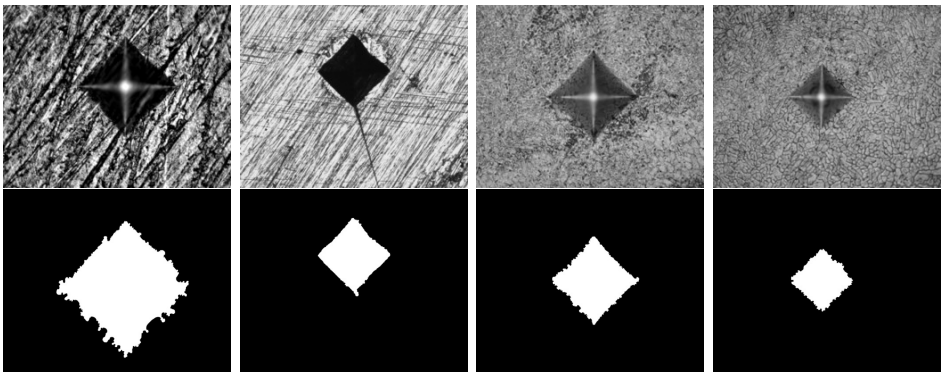


Figure 6.16: The multi-resolution active contours approach fails to segment the top images. The bottom (unsuccessful) level sets are computed.

Chapter 7

Shape from focus

Straightforward image processing usually only relies on one image, which has to be segmented. All the information must be gathered from this single two dimensional signal. However, the real world cannot be described by two dimensions, as space has got three dimensions. To overcome this inadequacy, a more general approach [19] relies not only on one (focused) image, but on a set of images, with different focus setups. This introduces a kind of depth information, although the signal is sampled with a simple 2D-camera.

Focus can only be achieved for a specific defined region. That means, it is not possible to focus an object in the foreground as well as an object in the background simultaneously. Consequently, if pictures with different focus setups of a three dimensional object are taken, information of the third dimension can be obtained. To estimate the depth of an image point, the picture with the highest focus measure (i.e. the point is in focus) at this point has to be evaluated.

In the referenced approach [19], the shape of visibly rough surfaces is determined. As focus can only be measured if differences of pixel values occur, smooth surfaces cannot be segmented in this way. However, the authors propose a special illumination strategy in order to create high frequency signals. As our hardware should stay unchanged this detail has not been investigated.

In the case of Vickers indentations, images are containing imprints, which definitely are three dimensional (the middle of the object is most far away whereas the background is nearer to the camera). The constraint of a visibly rough surface (i.e. high frequency, noisy images, low contrast) often applies, especially in case of low quality images which are likely to be mis-segmented by the (proposed) traditional segmentation method. Segmentation performance for the high quality images (i.e. low noise, high contrast) could not

be improved, as high frequency information is missing. But as such images usually can be segmented accurately by our so far proposed multi-resolution algorithm, we decided to focus on low quality images (e.g. the image in figure 7.1), with high frequencies on the one hand (necessary for this approach) and low contrast (hard to segment for traditional methods that rely on different gray values) on the other hand.

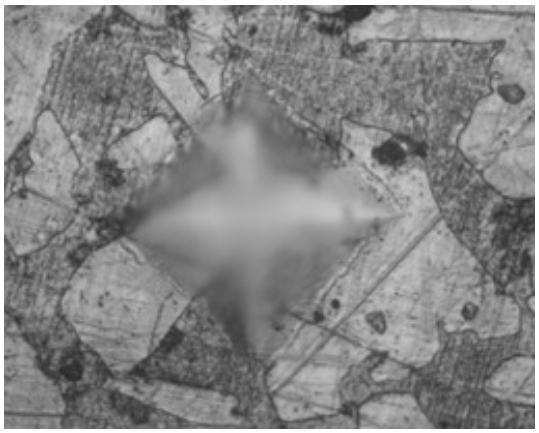


Figure 7.1: Example image with noise and low contrast - it is even hard to manually determine the object's boundaries.

To include information from focus, 40 pictures with different focus settings have been taken. Figure 7.2 shows the effect of different focus settings. While in the left image the deepest part of the imprint, in the right image the background (shallow part) is focused. Obviously, having pictures with different focus settings, the depth relative to other pixels (which is sufficient) can be estimated.

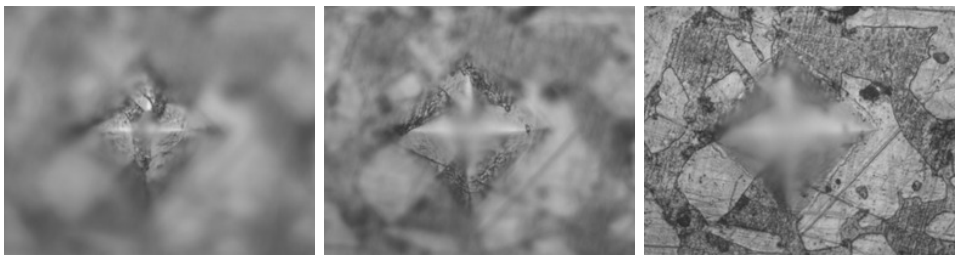


Figure 7.2: From left to right focus starts on the region far away (imprint) and is pulled nearer

In figure 7.3 (middle and right image) regions are focused, which are father away than any part of the image. As these images do not present regions which are in focus, the images do not provide any depth information. Consequently, they are not necessary for determination of the depth. Moreover

to reduce computational costs, only 8 images are used for processing.

Figure 7.3: From left to right focus starts at the background and is pulled nearer

7.1 Introduction of shape from focus

A method to determine the "shape from focus" has been introduced in [19]. First of all a focus measure is introduced to determine a degree of focus for each pixel. In the sections 7.1.1 - 7.1.3 different alternatives are discussed.

7.1.1 Sum-modified-Laplacian (SML) operator

The authors propose a modified Laplacian which is calculated for each pixel of each gathered image. It is based on the second order derivation.

$$\begin{aligned} ML(x,y) = &|2 \cdot I(x,y) - I(x-s,y) - I(x+s,y)| + \\ &|2 \cdot I(x,y) - I(x,y-s) - I(x,y+s)| \end{aligned} \tag{7.1}$$

The step size s defines the regarded distance between the pixels (e.g. $s = 1$). The following focus measure has been proposed:

$$F_{SML}(i,j) = \sum_{x=i-N}^{i+N} \sum_{y=j-N}^{j+N} ML(x,y) \quad if\ ML(x,y) \geq T \tag{7.2}$$

The threshold T should be set to a value greater than zero, in order to suppress very weak responses. N defines the size of the regarded region surrounding a pixel.

7.1.2 Tenengrad operator

Alternatively the Tenengrad focus measure could be used instead of the proposed sum-modified-Laplacian, which is based on the first order derivation.

$$S_x = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \tag{7.3}$$

$$S_y = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix} \tag{7.4}$$

$$F_T(i,j) = \sum_{x=i-N}^{i+N} \sum_{y=j-N}^{j+N} T(x,y) \quad if\ T(x,y) \geq T \tag{7.5}$$

where $T(i,j) = S_x^{*2}(i,j) + S_y^{*2}(i,j)$ and S_x^* and S_y^* are convolutions of the Sobel operators S_x and S_y with the image.

7.1.3 Range metric

Moreover, we investigated the range metric, which is based on the histogram. As we need a metric for each point separately and not for the whole image, the region r surrounding the point is regarded:

$$r(i,j) = \{(x,y)|(|x-i| + |y-j|) \leq N\} \tag{7.6}$$

The range metric is calculated in the following way:

$$F_{range}(i,j) = max(r(i,j)) - min(r(i,j)) \tag{7.7}$$

N defines the size of the regarded region.

We have compared these three focus measures in section 7.3.1.

7.1.4 Shape computation

First of all, to compute the shape of an object, a series of images I_k of this object with different focus levels $k \in L$ must be gathered (L is the set of focus levels). After that, for each point $v = (x,y)$ in each image I_k of a series, a focus measure $F(v)$ must be computed. Next, for each point v the focus level $k \in L$ with the highest focus measure $F_k(v)$ is calculated. Each focus level k represents a defined depth level d:

$$d(v) = k \in L : \forall l \in L : F_k(v) \geq F_l(v) \tag{7.8}$$

Although the depth is not measured absolutely, relative differences are sufficient to determine peaks and valleys of the surface.

Whereas this method produces regions of the same discrete level, the authors additionally introduced an approach that generates a smooth shape. As our application does not allow a perfect reconstruction of the shape (images contain regions without high frequency) and a smoother approximation is computationally more expensive, we content ourself with the simple discrete version.

7.1.5 Setup

Figure 7.4 shows the generated depth information gathered from the images in figure 7.2 with the SML focus metric. While brightly colored regions have been determined to be near to the camera, darkly colored ones are expected to be far away. In the case of a weak response (no high frequencies available), the determination is uncertain. This is the reason why even regions in the foreground are darkly colored. The higher the degree of high frequency information is, the higher is the certainty.

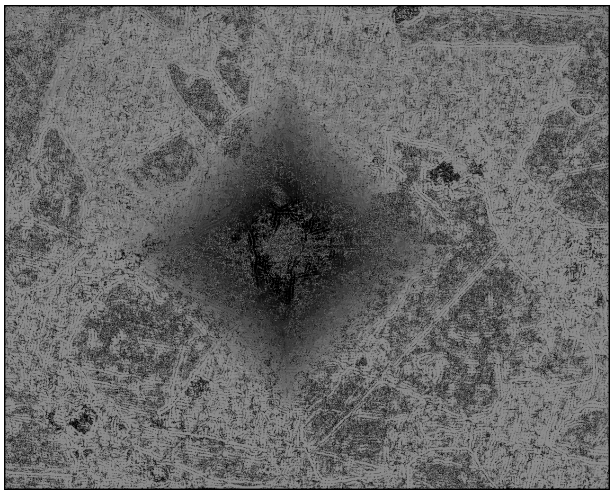


Figure 7.4: Depth measure applied to image ($T = 0$)

The threshold T was set to zero gathering the referenced image. Pursuing this strategy leads to one big trouble: Each pixel is assigned to a depth plane, independently from the focus measure's best response. The level with just a slightly higher response is the resulting depth level. To overcome this inadequacy, the threshold T has to be set greater than zero. Consequently, regions which cannot be assigned reliably are labeled to be uncertain. In Figure 7.5 unreliable pixels are marked in red.

7.2 Only use depth information for segmentation

As the shape from focus approach does not perfectly segment the images, we utilized the generated depth information in our active contours algorithms. One straightforward approach just takes the gathered depth image in order to segment the object from the background. Having images like in figure 7.5, this method might be a good idea, because regions inside the object are assigned with a focus level far away from the camera (darkly colored), whereas regions in the background are mostly determined to be near to the camera (bright colored).

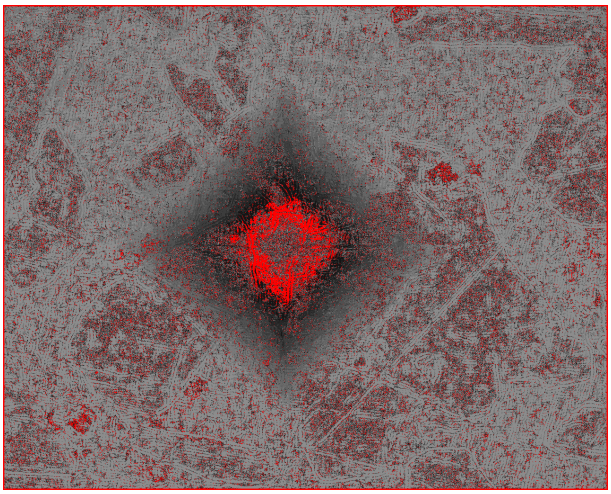


Figure 7.5: Depth measure applied to image ($T = 7$)

7.3 Include depth as feature in segmentation

As depth information alone in the case of many images surely does not provide enough information, our second approach does not only rely on the gathered new information of the third dimension, but also uses the traditional image information (gray value and edge information). Figure 7.6 shows images with large regions where the depth level cannot be reliably determined.

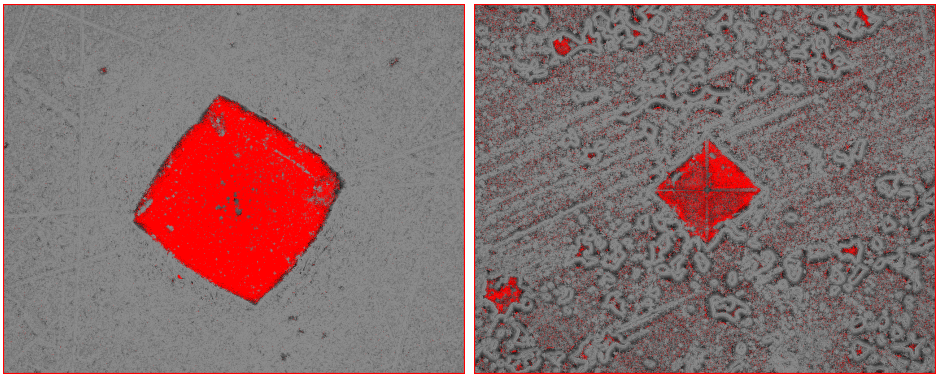


Figure 7.6: Depth determination not reliable (SML measure: $T = 7, N = 1, s = 3$)

The existing multi-resolution approach (strict Shape-Prior gradient descent followed by a level set method), has been slightly changed to allow depth information as a new feature:

7.3.1 Include focus information in gradient descent method

The first stage of our multi-resolution algorithm has been introduced in section 4. Now we especially concentrate on section 4.1.6 where the statistical approach is introduced. This method is not only able to consider just one single feature (like e.g. the region based model), but a feature vector of arbitrary length (in practice the length is highly limited, due to computational costs). Our experience has shown that the following feature vector yields the best results, with the standard image database.

$$f(v) = (I(v), ||\nabla I(v)||) \tag{7.9}$$

As we now concentrate on low quality images and we would like to add information from focus (depth map), the following feature vectors has been investigated:

$$f(v) = (I(v), ||\nabla I(v)||, depth_{focus}(v)) \tag{7.10}$$

$$f(v) = (I(v), depth_{focus}(v)) \tag{7.11}$$

We have compared the traditional version of the feature vector (shown in equation 7.9) with the new versions (equations 7.10 and 7.11) comprising gray, edge (only in equation 7.10) and additionally depth information (calculated from focus).

Our standard database does not comprise different focus levels, but only one focused image of each indentation. Consequently, the image-series database (section 1.2.1) for evaluation is used instead with 25 mostly difficult images and different focus setups (40 for each indentation). We used only 8 of the 40 images, as we do not need to perfectly reconstruct the shape of the indentation. The 8 images consist of 7 images where the focused plane is farther away than the background (figure 7.2) and the image where the background is in focus.

7.3.2 Analysis on indentation images

We succeeded in achieving better results with the additional shape information. Using the following feature vector in equation 7.11 and the SML focus measure ($T = 7, N = 1, s = 3$) turned out to be the best strategy. The edge information seems to be no longer required.

In figure 7.7 the impact of different focus metrics on the image-series database (section 1.2.1) is shown. Although, the differences are very small we identified the SML operator to be the best focus metric for our purpose, as the number of outliers is slightly lower. In the following analyses, the shape from focus approach based on the SML operator is compared with

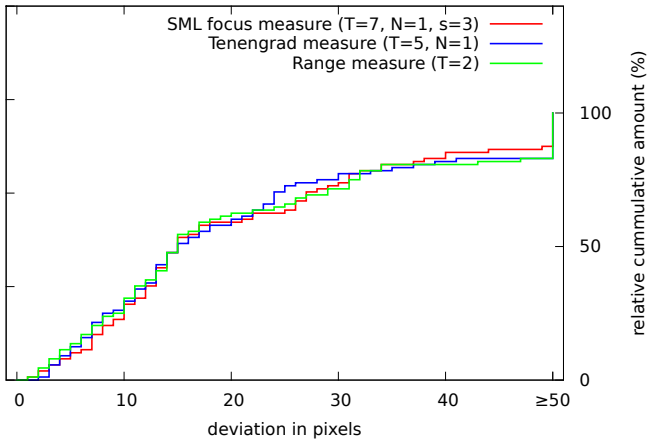


Figure 7.7: Different focus metrics

the traditional approach (section 4.1).

In figure 7.8 the results are shown, which are achieved with the mentioned statistical method, which includes gray value information as well as shape information. Figure 7.9 shows the best results that are gathered using our proposed statistical gradient descent method including gray value and edge information. In figure 7.10 these two different approaches are compared. The (cumulative) curves represent the number of edges, which are detected within ($\leq$) a given distance. Whereas the number of edges detected quite exactly (0-25 pixels) is similar, the number of outliers can be decreased significantly with the shape-from-focus information. As we concentrate on a low outliers ratio (i.e. a high degree of robustness) in the first stage, the focus information is able to improve the segmentation performance.

Unfortunately, the execution runtime is high as the generation of the depth information is computationally complex (discussed in chapter 10).

In Fig. 7.11 the impact of the initialization on the traditional region based level set segmentation is shown. If the method is initialized with the results achieved with the Shape Prior gradient descent method including the depth information, the results are clearly superior (red line) to the traditional first stage. The depth information used by the Shape Prior approach definitely increases the segmentation performance.

7.3.3 Include focus information in level set method

The second stage of the multi-resolution algorithm comprises an active contours algorithm (section 3). In case of our standard database, the region based level set approach produced best results, followed by the statistical level set approach (section 3.3). The statistical approach is able to deal with

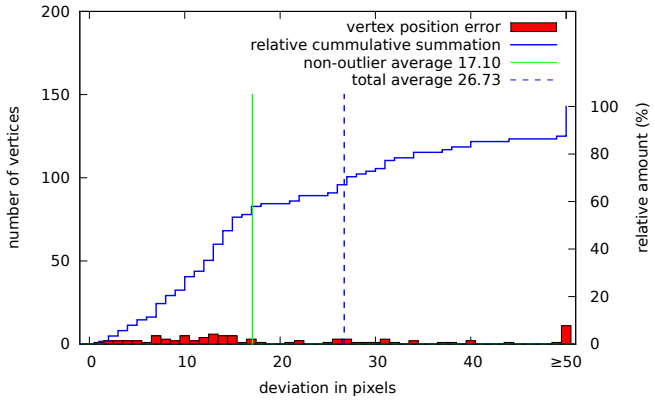


Figure 7.8: Gradient descent method based on gray value and shape information (SML focus measure, $T = 7$, $N = 1, s = 3$)

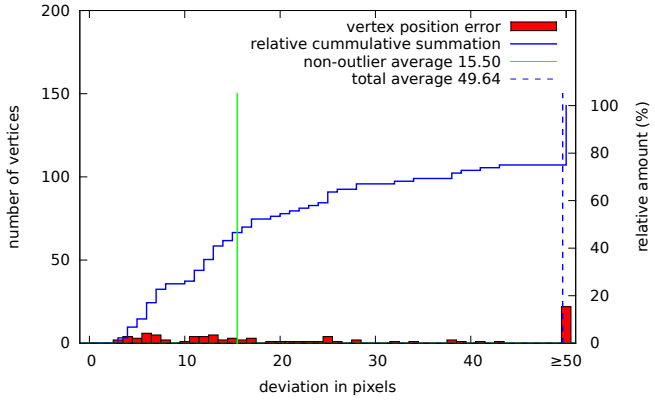


Figure 7.9: Gradient descent method based on gray value and edge information

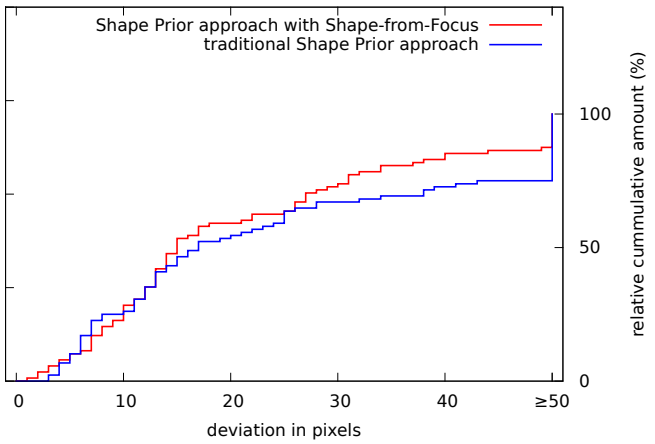


Figure 7.10: Comparison of both methods

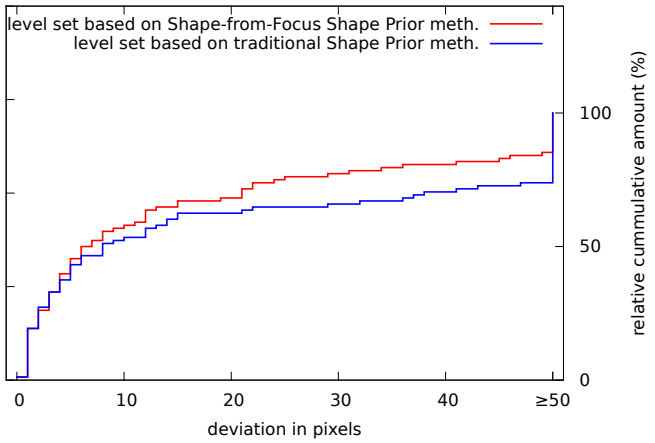


Figure 7.11: Level set: Impact of initialization

feature vectors, instead of only one feature. Hence, we (again) altered the feature vector of the statistical approach by adding depth information. The best results are achieved with the following feature vector:

$$f(v) = (I(v), ||\nabla I(v)||, depth_{focus}(v)) \tag{7.12}$$

7.3.4 Analysis on indentation images

Now we compare the statistical level set approach including shape information with the traditional region based level set method (section 3.2) and the statistical level set approach (section 3.3) without shape information. The methods are initialized with the results achieved with the Shape Prior approach including the depth information. The results are shown in Fig. 7.12. In contrast to the approximative Shape Prior approach, the results of the precise level set segmentation approach are more similar. The approach including the depth information (red line) seems to be slightly more robust than the region based approach (regarding deviations of e.g. ≤ 40 pixels). However, the region based approach (green line) tends to be more accurate (regarding deviations of e.g. ≤ 5 pixels). The statistical approach without the depth information (based on gray value and edge information) tends to be in the middle of the other mentioned approaches.

To understand this behavior, you should watch the different depth images provided by the shape-from-focus method (figures 7.5, 7.6). Images with low noise and high contrast, which can be segmented well without any depth information often have an unreliable depth information (noisy and large regions without depth information (marked red)). Consequently this images suffer from the additional information. In opposite the depth information of highly noisy images usually is quite accurate, so the segmentation performance can be increased.

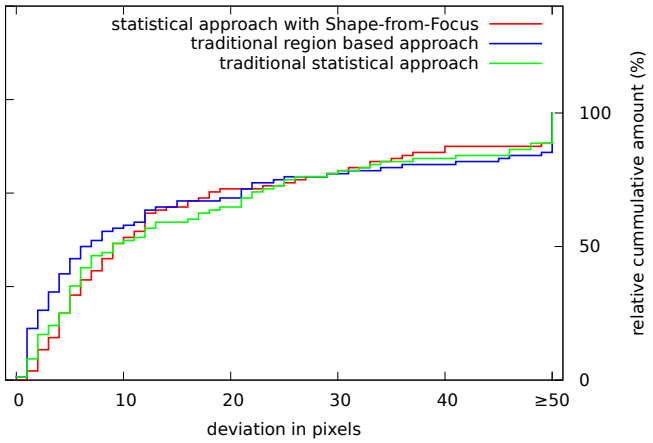


Figure 7.12: Comparison of level set methods

Consequently, we only recommend to use the shape-from-focus approach if the image is hard to segment by the traditional algorithms (e.g. our proposed traditional multi-resolution algorithm). To automatically determine the quality of images, a measure must be introduced.

7.3.5 Shape from focus only in special cases

Unfortunately, the shape from focus approach is computationally expensive and for high quality images (with low noise) even disadvantageous. However, for low quality images, the segmentation performance can be increased. Consequently, we considered to apply the shape from focus approach (both stages) only on specific "bad" images, whereas the "good" images are segmented traditionally (as proposed in chapter 6).

We investigated a quite simple distinction method. If the sum of the SML focus measures of all points Σ_{SML} of the image I is higher than a defined threshold T_{bad}, the shape from focus approach is applied. We observed that "good" images usually have a low Σ_{SML} whereas "bad" ones have a high Σ_{SML}.

$$\Sigma_{SML} = \sum_{(x,y)\in I} F_{SML}(x,y) \tag{7.13}$$

In figure 7.13 the results with different thresholds T_{bad} are shown. Threshold 0 means that every image is segmented with the shape from focus method whereas threshold infinity means that every image is segmented with the traditional approach. Moreover two more sensible thresholds in the middle has been chosen. Actually, the high overall segmentation performance of the shape from focus approach cannot be increased. The higher the threshold

is, the less robust is the segmentation (as far as outliers are concerned). The slightly less precise segmentation performance of the shape from focus approach cannot be improved. However, this definitely decreases the computational costs as only some "bad" images are segmented with the expensive shape from focus method. In our case, with a threshold of 1,000,000 only the half of the images is segmented with the expensive method. In the case of images of a higher quality, an even much smaller ratio would be segmented with the shape from focus approach.

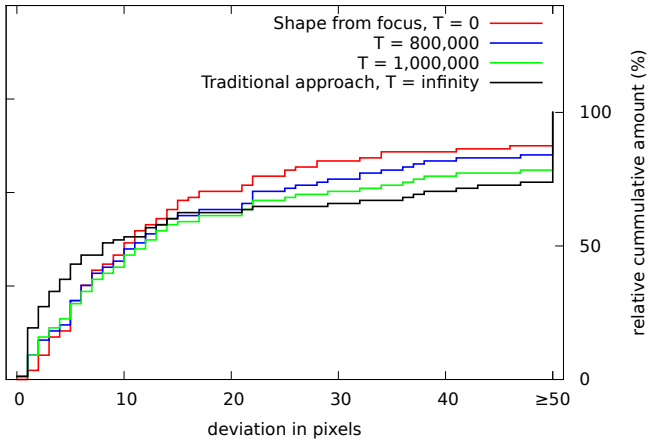


Figure 7.13: Comparison of different thresholds

7.3.6 Depth from logarithmic image processing

In this section we investigate a model which provides new arithmetic operations for image processing. The standard linear operators for addition $(+)$, subtraction $(-)$ and multiplication $(\times)$ which are normally used, do not correspond with the human visual system. In order to eliminate this inconveniences, in [22] the logarithmic image processing (LIP) framework has been proposed.

The operations are based on gray tone values ($f \in S$) (0 corresponds with white, M (255) with black), i.e. the pixel values (I) must be inverted before the operations are applied. Instead of the standard $+$ (plus), $-$ (minus), $\times$ (times) and $|\cdot|$ (absolute) operators the LIP operators $\oplus$, $\ominus$, $\otimes$, $||\cdot||_{LIP}$ are defined by the following equations:

$$f = 256 - I \tag{7.14}$$

$$\forall f, g \in S : f \oplus g = f + g - \frac{f \cdot g}{M} \tag{7.15}$$

$$\forall f,g \in S : f \ominus g = M \cdot \frac{f-g}{M-g} \tag{7.16}$$

$$\forall f \in S, \forall \lambda \in \mathbb{R} : \lambda \otimes f = M - M \cdot (1 - (\frac{f}{M}))^{\lambda} \tag{7.17}$$

$$\forall f,g \in S : f \otimes g = \phi(f) \cdot \phi(g) \tag{7.18}$$

$$\forall f \in S : ||f||_L = |\phi(f)| \tag{7.19}$$

$$\forall f \in S : \phi(f) = -M \cdot \ln(1 - \frac{f}{M}) \tag{7.20}$$

M is the highest pixel value (in our case $M = 255$).

In [6] focus measures based on the logarithmic image processing model were investigated. Moreover the authors concentrated on shape from focus applications.

We get down to the two common focus measures Tenengrad (section 7.1.2) and sum-modified-Laplacian (SML) (section 7.1.1). We have already investigated the traditional operators. Now we concentrate on the LIP versions. The LIP-Tenengrad measure is computed in the following way:

$$TEN_{LIP}(x,y) = ||\frac{\delta f}{dx}||_L^2 + ||\frac{\delta f}{dy}||_L^2 \tag{7.21}$$

$$\begin{aligned}\frac{\delta f}{dx} = \frac{1}{4} \otimes ((f_{(x-1,y-1)} \oplus (2 \otimes f_{(x-1,y)}) \oplus f_{(x-1,y+1)}) \ominus \\ ((f_{(x+1,y-1)} \oplus (2 \otimes f_{(x+1,y)}) \oplus f_{(x+1,y+1)}))\end{aligned} \tag{7.22}$$

$$\begin{aligned}\frac{\delta f}{dy} = \frac{1}{4} \otimes ((f_{(x-1,y-1)} \oplus (2 \otimes f_{(x,y-1)}) \oplus f_{(x+1,y-1)}) \ominus \\ ((f_{(x-1,y+1)} \oplus (2 \otimes f_{(x,y+1)}) \oplus f_{(x+1,y+1)}))\end{aligned} \tag{7.23}$$

The LIP-SML measure is computed in the following way:

$$SML_{LIP}(i,j) = \sum_{x=i-N}^{i+N} \sum_{y=j-N}^{j+N} ML(x,y) \quad if\ ML(x,y) \geq T \tag{7.24}$$

$$\begin{aligned}ML_{LIP}(x,y) = ||(2 \otimes f_{(x,y)}) \ominus f_{(x-1,y)} \ominus f_{(x+1,y)}||_L + \\ ||(2 \otimes f_{(x,y)}) \ominus f_{(x,y-1)} \ominus f_{(x,y+1)}||_L\end{aligned} \tag{7.25}$$

7.3.7 Analysis on indentation images

Although in some cases, the use of the LIP operators within our multi-resolution algorithm leads to better results, this cannot be generally proven. For both stages, significant performance improvements cannot be achieved, as far as the whole database is concerned. So we propose to use the traditional SML measure.

Chapter 8

Segmentation of unfocused images

In order to acquire focused images, the Vickers hardness testing utilities rely on auto-focus systems. Such systems apply a focus measure to images of the same object with different focus setups to decide, which image is in focus. In [16] different measures are investigated with reference to Vickers hardness images.

Whereas most segmentation algorithms only regard focused images, we have investigated an approach (chapter 7: shape from focus) that uses the information from unfocused images as well, to improve the segmentation results. The segmentation results can be improved, but the execution costs (runtimes) are very high.

The next step is, to find out if it is possible to compute approximative results from unfocused images. This might be a good idea, as the auto-focus algorithm consumes a lot of time, which could be used by an approximative segmentation algorithm based on an unfocused image.

We investigate the effect of wrongly focused images on the segmentation algorithms. We have sorted the images according to their focus level (fl). The exact step size between two consecutive focus levels cannot be generally specified, as it depends on the optical zoom of the camera for the individual image:

zoom	step size
10 x	10,000 nm
20 x	5,000 nm
40 x	1,000 nm

For example, if an image is 10x enlarged, a step size of 10,000 nm is

chosen (i.e. while the camera moves, every 10,000 nm a picture is taken). The higher the zoom, the smaller the step size must be. For example $fl = -5$ means that the focused plane is five steps deeper than the best focus level. In figure 8.1 the effect of different setups is shown. The same database as in chapter 7 is used with 25 indentation image series (i.e. 100 corners). The quality of the images is considerably lower than the quality of the standard database.

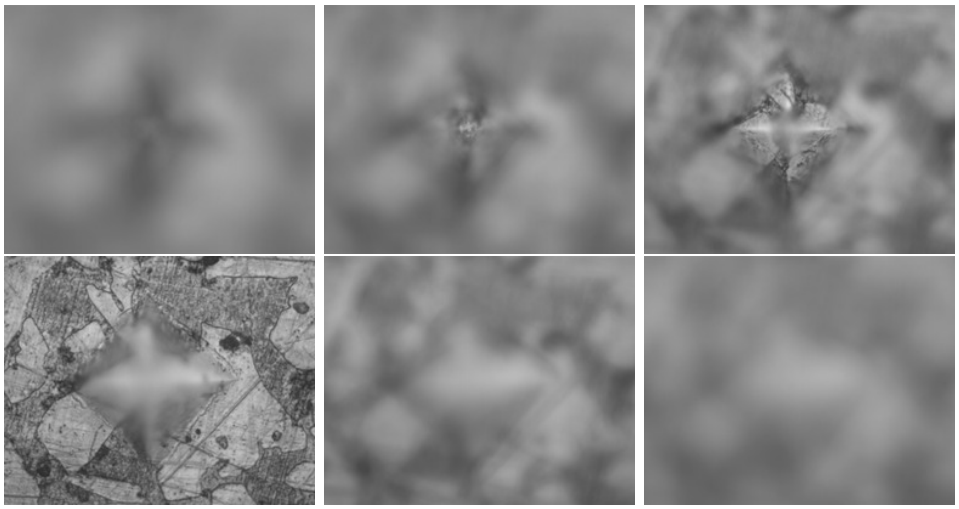


Figure 8.1: Different focus settings, reaching from $fl = -15$ (top left) to $fl = +10$ (bottom right)

8.1 Effect on the gradient descent method

First we would like to know the effect of unfocused images on our proposed gradient descent algorithm (section 4.1), as this method serves as the first stage of our multi-resolution algorithm to achieve approximative results. We have tested the algorithm with different focus levels. We expected that the accuracy slightly suffers but the robustness stays about the same (tests with blurred images showed that the number of outliers stays about the same), which is more important for us in the first segmentation stage.

Figure 8.2 shows results with settings which focus points that are farther away from the camera compared with the best-focus strategy (red line). The robustness (i.e. few outliers) of the segmentation did not only stay unchanged as expected, but can actually be increased if the unfocused images are used. However, the segmentation accuracy (e.g. the ratio of corner points with a deviation of maximal 20 pixels) slightly decreases. As the curves are crossing, we cannot identify a best configuration just by looking at the results.

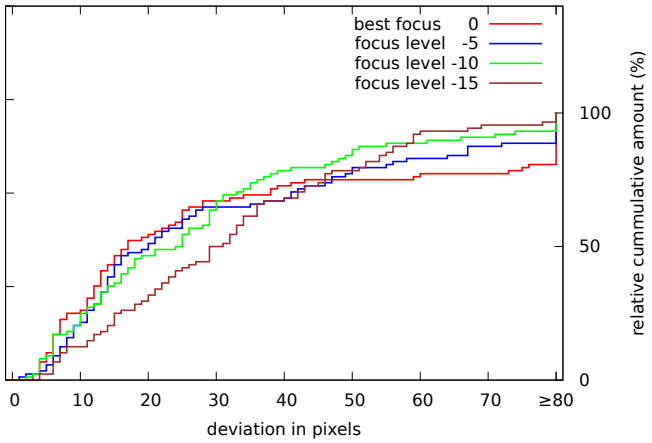


Figure 8.2: Different focus setups (focal point far away)

The results in figure 8.3 are achieved with images where points are in focus that are nearer than the image points. The segmentation performance with these images definitely decreases. Consequently, we specialized on focus levels shown in figure 8.2.

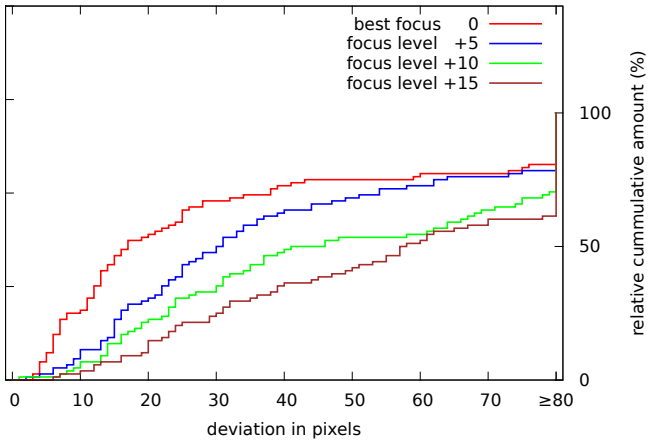


Figure 8.3: Different focus setups (focal point near)

Now we know that the segmentation process even benefits from (the right) unfocused images as far as outliers are concerned. However, the accuracy decreases. As the results of the gradient descent algorithm "only" serve as initializations for the level set algorithm, in the next section we investigate the impact of the different initializations on the level set algorithm.

8.2 Effect of different initializations on the overall performance

We initialized the level set method with the results, gathered from the gradient descent with different focus levels to get a knowledge of the impact on the overall performance of the multi-resolution algorithm. However, the level set algorithm still operates on the focused images. The question is, how accurate the first stage results have to be in order to achieve precise overall results of the second stage. Figure 8.4 shows that the differences

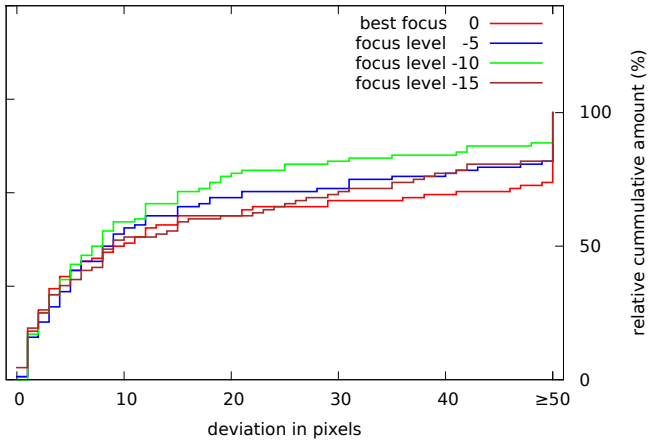


Figure 8.4: The level set method with different initializations

of the initializations definitely influences the overall segmentation output of the multi-resolution algorithm. The multi-resolution algorithm does not generate the best results if the best focused images are provided to the first stage algorithm. Whereas when regarding the gradient descent algorithm (figure 8.2) we cannot identify a winning focus-configuration, as the curves are crossing, now we can. The focus level $fl = -10$ is superior to the others for nearly each maximal deviation. Especially the number of outliers declines considerably. So we come to the conclusion that the segmentation of unfocused images with our proposed first stage gradient descent algorithm is even superior to the segmentation of perfectly focused images, as far as an appropriate focus level is chosen. The precomputed results with the $fl = -10$ gradient descent strategy are just slightly less accurate (if small deviations are regarded) than the best focus strategy, but the number of outliers is minor, which is beneficial. Although the $fl = -15$ strategy has even less outliers, the overall performance decreases, as the accuracy suffers too much.

8.3 Effect of focus on the level set method

So far we have investigated the impact of unfocused images on the first approximative stage of our multi-resolution algorithm. As the segmentation performance even increases, next we investigate the impact on the proposed stage 2 (level set) algorithm (section 3). We initialized the level set method with the results achieved with the focus level $fl = -10$, as it turned out to be the best choice for our database. The level set segmentation method is evaluated with different focus levels. In figure 8.5 you can see that the

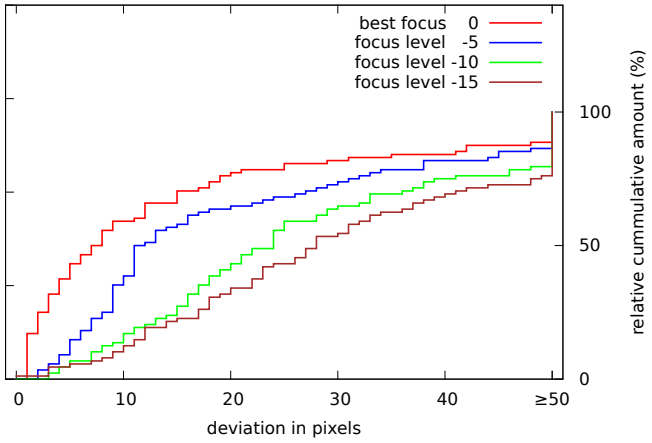


Figure 8.5: The level set method with different focus setups

segmentation output definitely suffers, if the images for the second stage algorithm are not focused.

To put it in a nutshell, the overall segmentation performance decreases if the images for the second stage algorithm are not focused. However, the performance even increases, if the approximative first stage algorithm segments the (right) unfocused images.

8.4 Effect on the corner template matching approach

We have already compared our proposed multi-resolution approach with the template matching method introduced in [7] (section 6.3.3). Now we investigate the impact of unfocused images on the template matching method. In opposite to the newly introduced multi-resolution algorithm now we investigate the influence on the whole algorithm, not just on one stage (approximative segmentation stage or precise segmentation stage). First of all we point out that the referenced template matching method suffers from the lower

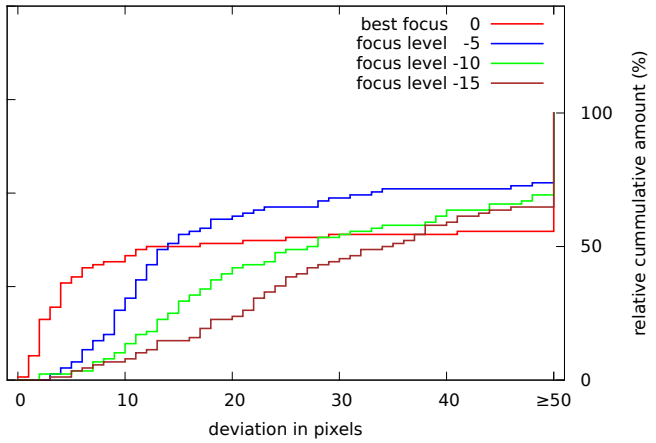


Figure 8.6: The corner template matching approach with different focus setups

quality of the images in the utilized database. As shown in figure 8.6 the outliers ratio is huge compared with the outliers ratio of the multi-resolution algorithm. The template matching of unfocused images results in a slightly lower outliers ratio on the one hand. On the other hand, the accuracy highly decreases. If an accurate segmentation is required we do not recommend to segment unfocused images. Moreover if the image quality is as low as in our database, we recommend to use the introduced multi-resolution algorithm instead of the template matching approach.

8.5 Effect on the approximative template matching approach

The corner template matching approach introduced in [7] is an algorithm to precisely detect the corners of the Vickers indentations. As shown in figure 8.5, a precise segmentation algorithm suffers from unfocused images. Now we investigate the impact of unfocused images on the approximative template matching approach introduced in [17]. Whereas in [7] corner templates are used to precisely detect the object's corners, in [17] different square templates are matched with the images. As the number of templates is limited, a high accuracy cannot be achieved. However, the aim is to achieve a low outliers ratio.

Figure 8.7 shows that the effect of unfocused images on the approximative template matching is similar to the effect on the approximative gradient descent algorithm (shown in figure 8.2). The outliers ratio can be improved with unfocused images (especially with $fl = -10$).

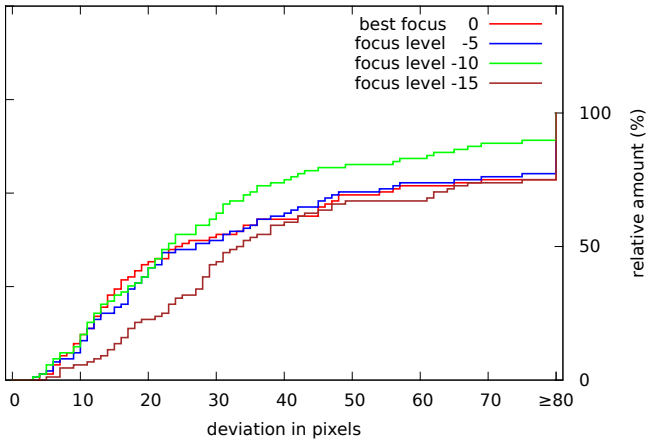


Figure 8.7: The approximative template matching approach with different focus setups

8.6 Effect on the template matching refinement

In [8] a refinement approach has been introduced which is based on the results of the approximative template matching method. As with the level set approach, we also investigate this approach with reference to different initializations. The different initializations gathered with different focus levels are discussed in section 8.5. The refinement method is always based on the focused images. In figure 8.8 the results are shown. As with the level set approach, the best results are not achieved with the best focused images in the first stage. Especially with images of focus level -10, the outliers can significantly be reduced.

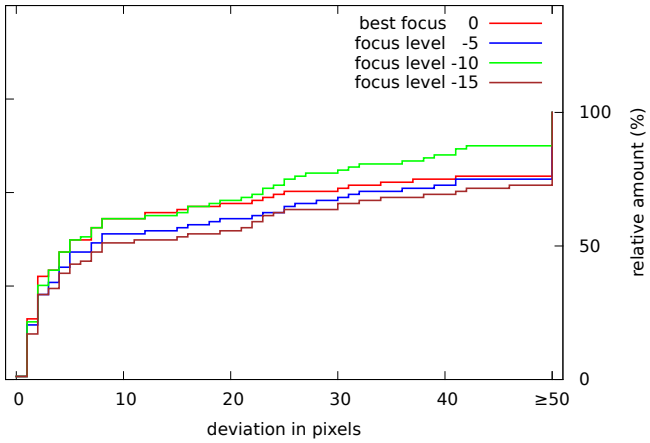


Figure 8.8: Refinement of corner template matching approach

8.7 Compare multi-resolution with template matching approach

In this section, we compare the best results of our proposed multi-resolution active contours algorithm with the best results achieved with the 3-stage method proposed in [8] (approximative template matching followed by refinement), as both methods seems to be quite robust and exact (compared with the template matching approach introduced in [7]). The first approximative stages of both methods are based on the images of focus level -10, whereas the precise stages are based on the focused images, as with this configuration, the best results are achieved. In figure 8.9, the approaches

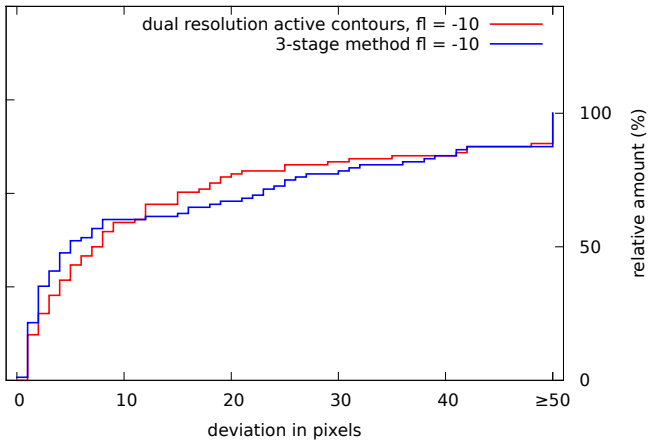


Figure 8.9: Refinement of corner template matching approach

are compared. Actually, we cannot determine a winner, as the curves are crossing. Whereas the 3-stage method proposed in [8] is slightly more precise (deviations < 10 pixels), our proposed active contours multi-resolution approach is more robust (deviations $15 - 30$ pixels). In section 6.3.3, the segmentation performance of both methods has been compared with reference to focused images and the database with images of a considerably higher quality. Interestingly enough, on this images, the active contours based approach is more precise (small deviations).

8.8 Compare with shape from focus

Finally, we compare the improved results which are achieved with unfocused images ($fl = -10$) and the proposed shape prior gradient descent algorithm (chapter 7), with the shape from focus results. Figure 8.10 shows that the performances are quite similar. The shape from focus method is still slightly more accurate. However, the number of outliers is about the same.

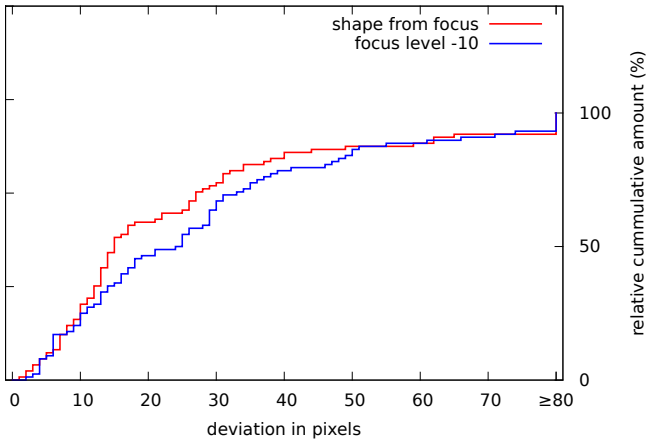


Figure 8.10: Gradient descent approach: shape from focus vs. traditional approach based on unfocused images

Although the differences are quite small, the shape from focus strategy not only introduces robustness allied with the approximative gradient descent algorithm, but also allied with the (precise) level set segmentation algorithm. The segmentation of unfocused images only improves the results of approximative algorithms, as detailed information is missing in such images. A comparison of the following three strategies is shown in figure 8.11:

- Strategy 1: Shape-from-focus in both stages
- Strategy 2: Shape-from-focus in first stage and region based approach in second stage
- Strategy 3: Unfocused approach in first stage and region based approach in second stage

The first strategy is more robust, but less precise than the second strategy. Actually, the overall performance of the third strategy yields the best overall performance (robustness and accuracy)! Interestingly enough, the unfocused strategy 3 is more competitive than strategy 2 although the first stage's results are rather the opposite (figure 8.10) and the second stage algorithm is the same.

8.9 Conclusion

Approximative segmentation methods actually benefit from (the right) unfocused images. However, the precise segmentation performance declines when the input images are unfocused. Figure 8.12 shows the focused and

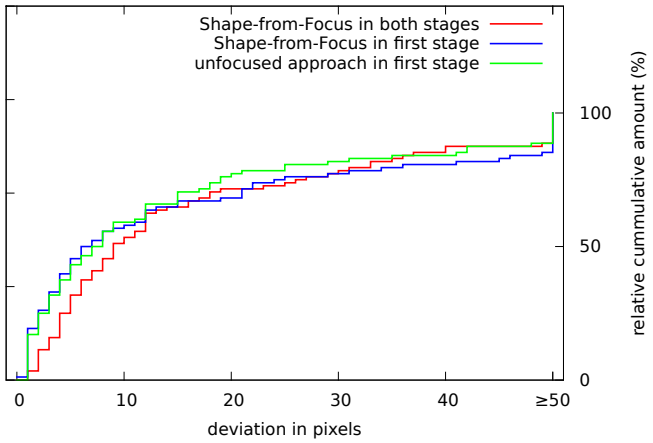


Figure 8.11: Level set method

the corresponding unfocused images. Especially the unfocused images on the left side are obviously easier to locate. The effect on the background is similar to the effect of a Gaussian blurring filter (shown in figure 8.13). But the most important fact is that the indentation is reinforced. The contrast between the indentation and the background is higher with the unfocused images. This effect definitely cannot be imitated with a blurring filter.

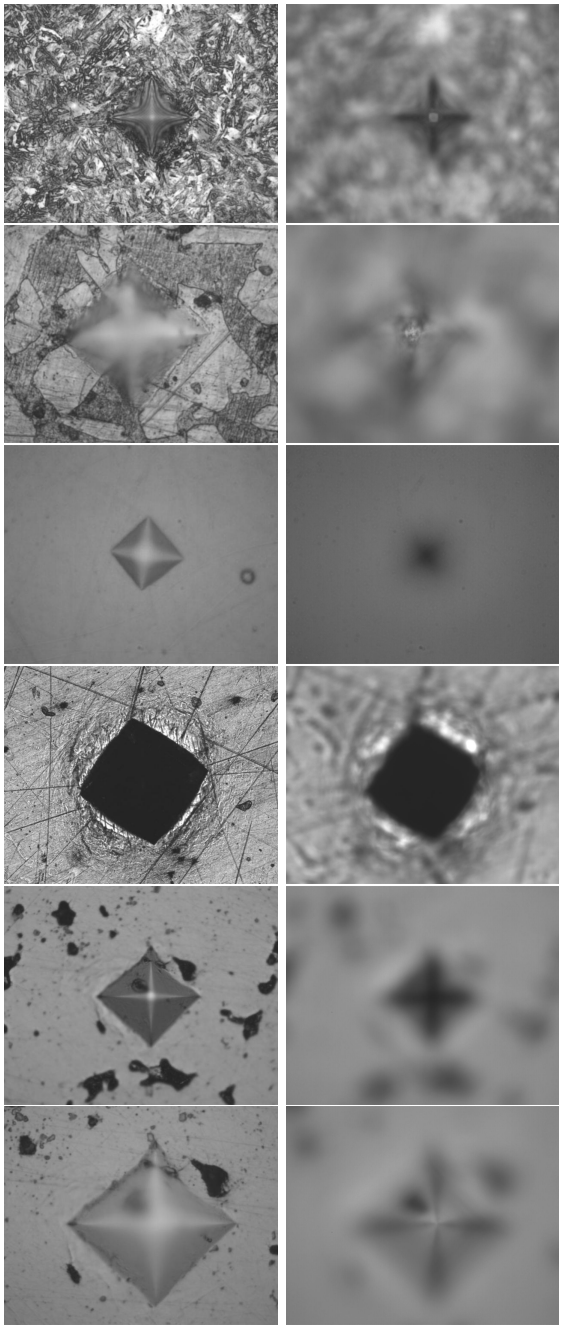


Figure 8.12: Focused ($fl = 0$) an unfocused ($fl = -10$) images

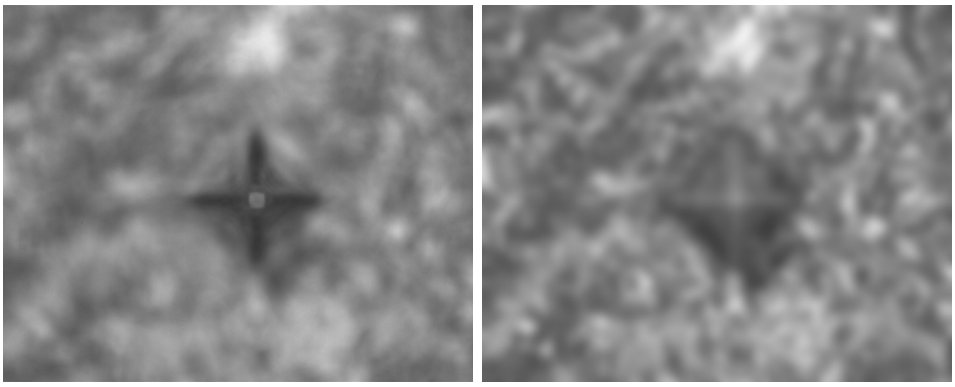


Figure 8.13: Unfocused image ($fl = -10$, left) vs. Gaussian blurred focused image (right)

Chapter 9

Reducing runtime: gradual enhance

In the last chapter we investigated the influence of unfocused images on the segmentation performance of different algorithms. Actually, the segmentation performance can even be improved by using the right unfocused images. This knowledge definitely is beneficial. However, our intention was to reduce the runtime of the overall hardness testing method.

The auto-focus system takes pictures (and computes the focus metric) and moves the camera for one step until the peak of the focus metric is reached. We aim in utilizing the free cpu cycles (when moving of the camera) for an approximate localization of the indentation. So far we know which focus levels are beneficial for segmentation. But to identify the focus level, first we have to know the best focus configuration. Consequently, an approximative segmentation cannot start before the focused image is computed.

Now we investigate the following strategy:

1. The focus starting setup is chosen that points are in focus which are farther away than the surface of the material and even farther away than any part of the imprint.
2. Start the proposed first stage gradient descent segmentation algorithm on the unfocused image which is taken with the mentioned focus setup. Approximative results are achieved.
3. Until the end-criterion is reached:
    - Alter the focus setup by one step (to the direction of the best focus) and get the image.
    - Initialize the gradient descent algorithm with the current approximative results and the new image.

- Increase the initialization variable "radius" by 2 pixels.
- Run the algorithm with only 5 iterations to enhance the approximative results.
- New approximative results are achieved.

The first image to segment is highly unfocused. Consequently, an exact segmentation surely cannot be achieved. However, the blurred image can be segmented robustly. Whereas the first image is segmented as proposed in section 4, the enhanced images are not. These images are initialized according to the current approximative results. As the balloon force is applied, the radius of the shape has to be increased by some pixels (e.g. 2), to allow a moderate growth. Otherwise the evaluated shape could only shrink, as the balloon forces it to shrinks.

The proposed policy allows to start the segmentation even before the focused image is taken. As the auto-focus process takes a significant amount of time, the available processor cycles can be used for an approximative segmentation.

We identified the following two major issues:

1. Which is an appropriate end-criterion (for instruction 3).
2. Whereas all enhancement segmentations are fast as only 5 iterations are necessary, the first initial step is computationally expensive (Runtimes are discussed in chapter 10).

These issues are discussed in the following sections.

9.1 Appropriate end-criterion

The end criterion should reliably determine, when the segmentation results cannot be refined any longer. One straightforward approach is, to try to refine the results, until the focused image is reached. This can be robustly determined using a focus metric. Unfortunately, the results do not necessarily improve until the best focused image is reached. The reasons for this paradox behavior are discussed in section 8, where a similar behavior is observed.

The results with different focus levels as stopping conditions are shown in figure 9.1. Although the behavior is similar to the behavior in section 8, the effect is smaller. The outliers ratio generally is lower than with the single image approach (shown in figure 8.2). The black line graph in figure 9.1 represents the best results, achieved with one single image (focus level

is -10).

As with the single image approach (chapter 8), segmenting until the focused image is achieved, is not advantageous as especially the number of outliers potentially increases. With our database, the lowest outliers ratio (no outliers) can be achieved when stopping at focus level -10 as shown in figure 9.1. However, we would have to compute until the focused image is achieved (i.e. the peak in the focus measure is reached), as not before this image is detected, the focus levels can be determined. This is because the focus levels are defined relatively to be focused image.

Even though the results seem to be more similar compared with the single image approach, the impact of the different results, which are only the initializations for the stage 2 algorithm, on the level set algorithm is considerable, as shown in figure 9.2. In comparison with the single image approach (black line graph), the gradual enhancement method's outliers ratio (focus level -10) is slightly lower, whereas the single image approach (focus level -10) is slightly more accurate. However, the differences are very small.

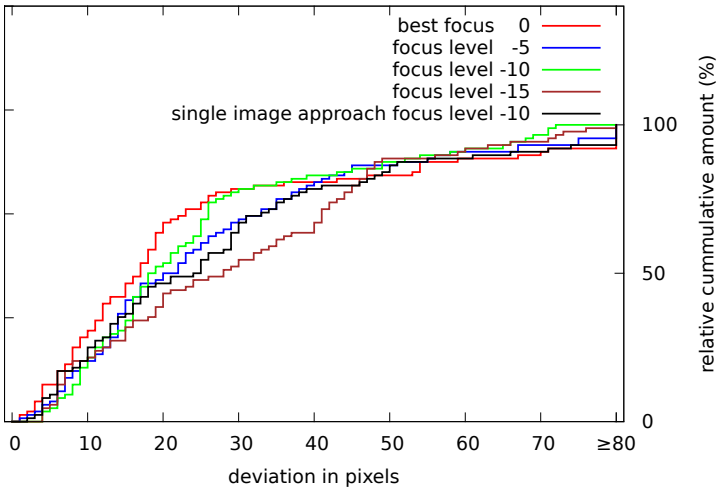


Figure 9.1: The effect of different stopping focus levels on the segmentation performance of the first stage algorithm

9.1.1 Determine from the focus measure

Moreover, we investigated if it would be possible to determine the end criterion by regarding the focus measure (Sum-modified-Laplacian). On the one hand we tried to determine a stop criterion from the sum of the focus measures of one image. The intention is, that highly noisy images, containing more high frequency information (i.e. higher focus measure) are stopped

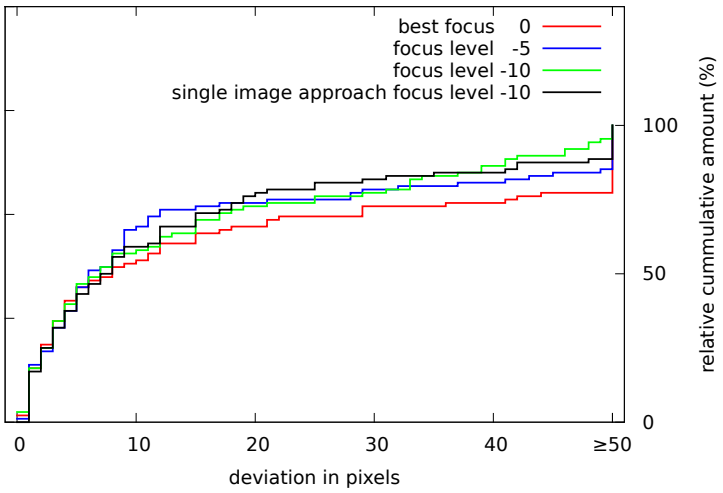


Figure 9.2: The effect of the different initializations on the level set algorithm

earlier because the focus threshold is earlier reached than high quality images. On the other hand we tried to determine a stop criterion based on the maximal focus metric of any pixel of the image. Whereas the first method is at least slightly more competitive than the best focus strategy, the second method does not work at all. We also investigated if a stop condition could be determined by regarding the first or second order derivation of the total focus measure of an image. However, the achieved results are not worth mentioning.

9.1.2 Determine from the extremums of gray value

We observed that especially the maximum gray value (brightest pixel) of the image, is changing over the focusing process in an interesting way. First of all, if the focused level is very far away (focus level is low, $fl << -10$), the maximum value is low. When increasing the focus level, the maximum value is increasing about until fl is between 10 and 15. After that, the value is decreasing. We investigated if the extremum (maximum) of the maximal pixel value corresponds with the ideal stopping condition of the enhancement algorithm. Unfortunately, the results are less competitive than when stopping at focus level -10.

9.1.3 Determine from a Gaussian fitting function

In this approach we try to fit (least squares) a Gaussian curve g to the points consisting of a focus level (x-axis) and a focus measure (y-axis).

$$g(A, \mu, \sigma) = A \cdot e^{-\frac{1}{2} \cdot (\frac{x-\mu}{\sigma})^2} \tag{9.1}$$

A Gaussian curve has been chosen, because it is similar to the focus-measure curve. The advantage of this approach would be that the segmentation could stop, if a defined distance from the maximum is reached. Our aim is, to determine the average value μ of the fitting Gaussian distribution, which should correspond with the maximum of the focus measure.

First, we investigate the easiest case, where all points "left to" the average value are already available. In the figures 9.3a and 9.3c results of example images are shown. For each focus level (x-axis), the corresponding focus measure (y-axis) is shown (red crosses). The blue lines represent the Gaussian fitting functions. The real best focus level for the shown images is 20. If all points (focus level 0 to 19) are available, an approximative determination of the best focus level often is possible (as depicted). However, for our application it would be necessary to determine the best focus level some levels before the best focus level is reached. In the figures 9.3b and 9.3d, an approximation with fewer available points (2 points are missing) is executed. Actually this highly affects the fitting process. The determined μ very often varies more than 5 pixels from the actual highest focus metric. In practice with even more missing points (e.g. 10 points) an approximative determination would totally fail. The following problems were identified:

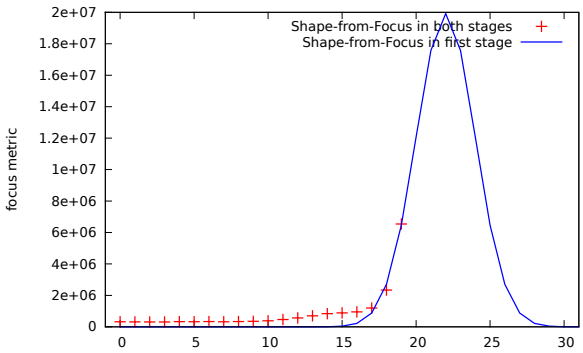
- The focus-level curve cannot exactly be described by a Gaussian curve.
- We only have points on the left side of the average value of the Gaussian distribution available. In practice, the last (e.g. 10) points as well are not given.

As we are not able to determine the averages of the fitting function robustly even for only 2 missing points, we have to emphasize that this approach is not appropriate for our (practical) usage.

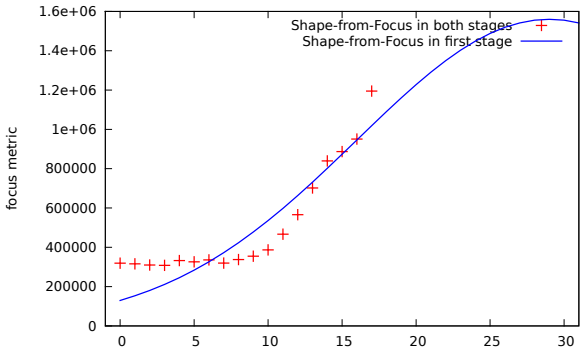
9.2 Speeding up the initial segmentation

Whereas it is fast to enhance the approximative results (only some iterations are necessary), the initial segmentation takes quite a long time. As the initial contour starts at the boundary of the image (initial radius is about 50 pixels as the images are downscaled by factor 10), has to shrink until it collapses and shrinks one pixel per iteration, about 50 iterations are necessary. If the image for the initial segmentation could be even more downscaled, this step could be accelerated, as downscaling does not only reduce the number of iterations, but even reduces the costs per iteration.

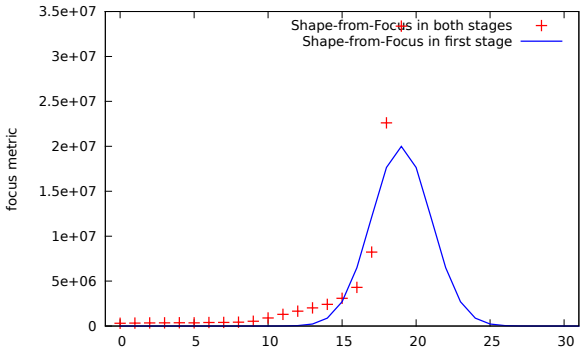
Unfortunately, the statistical gradient descent approach suffers if the number of pixels decreases. The major part is the computation of the probability density function for the two regions (inside and outside). In our case



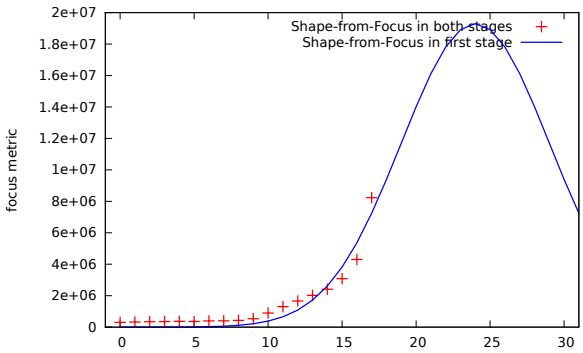
(a) Image 1, all points



(b) Image 1, 2 points missing



(c) Image 2, all points



(d) Image 2, 2 points missing

Figure 9.3: Gaussian fitting of the focus curve

the density function is 2-dimensional (gray value, gradient information). The discrete probability density function is represented by a 256x256 (256 gray values and 256 different gradient values) matrix (i.e. the number of values is $256 \cdot 256 = 65536$, independent of the image size). In the case of very small images, the number of pixels is much smaller (e.g. 50x60 image: 3000 pixels) than the number of matrix elements. As the empirical density function would be very rough and unsteady, a high degree of blurring would have to be applied, which is computationally complex. Moreover, the robustness of the segmentation would suffer.

We decided to choose an alternative strategy. Whereas the reduction of the image size affects the method, increasing the step size of the contour does not, as far as robustness is concerned. Instead of modifying the evolving shape parameters by one per iteration, we propose to increase the step size (i.e. in one iteration, each parameter is adjusted by the positive or negative step size or stays the same). Increasing the step size to 4, we achieved less accurate results after the initial segmentation step, but after the enhancement steps, the results were exactly the same. The runtime of the first step can be reduced by factor 4.

Chapter 10

Runtime analysis

Our framework is mostly implemented in Java, like all stage 1 algorithms, the evaluation of the results and the shape from focus computation. The level set methods are implemented in C++. We use the Ofeli level set library, which implements the traditional Chan-Vese model. In order to investigate alternative methods and improvements, the software has been expanded. The code definitely is not optimized for execution speed but to allow easy modification and experimentation.
The tests where executed on a notebook with an Intel Core 2 Duo T5500 1.66 GHz processor.

10.1 Stage 1 algorithms

Stage 1 algorithm	average	std. dev.
1. Proposed statistical 2 dimensional gradient descent algorithm (figure 4.12)	2.28 s	0.20 s
2. Directed edge based gradient descent algorithm (figure 4.8)	0.14 s	0.07 s
3. Statistical 1 dimensional gradient descent algorithm (figure 4.9)	0.30 s	0.10 s
4. Statistical 3 dimensional gradient descent algorithm (figure 4.13)	13.36 s	0.80 s
5. Shape-Prior 3 dimensional gradient descent algorithm*	13.45 s	0.82 s
6. Gradient descent with gradual enhancement (chapter 9)		
Initial segmentation	1.0 s	
One single refinement step	0.14 s	

10.2 Stage 2 algorithms

Stage 2 algorithm	average	std. dev.
1. Proposed Chan-Vese algorithm with evaluation of shape parameter	1.34 s	0.15 s
2. Statistical 2 dimensional approach without Parcen window on probability density function	1.51 s	0.13 s
3. Shape-Prior 3 dimensional level set approach* with shape-from-depth information without Parcen window	4.24 s	0.88 s

* If a Shape-Prior approach should be used, it is necessary to compute the shape-from-focus depth information before. This takes about 8 seconds with our Java implementation. The execution runtime do not depend on the quality or focus level of the image.

The costs of the proposed local Hough transform are about 0.7 seconds.

10.3 Total costs

Multi-resolution algorithm	average
1. Proposed multi-resolution algorithm (stage 1 algorithm 1, stage 2 algorithm 1, Hough transform)	4.3 s
2. Stage 1 directed edge strategy (stage 1 algorithm 2, stage 2 algorithm 1, Hough transform)	2.2 s
3. Approaches relying on shape-from-focus information	> 10 s

In the case of high quality images and if a runtime of about 4 seconds per image is acceptable, the first multi-resolution algorithm in section 10.3 should be chosen in order to achieve the best results.

The second algorithm is a more effizient approach as far as runtime is concerned. However, the segmentation performance is lower.

The third strategy which involves shape-from-focus knowledge, is computationally expensive. As the approach with unfocused images (chapters 8 and 9) is competitive as far as segmentation performance is concerned and much more effizient (as effizient as strategy 1) than the shape from focus strategy, for lower quality images we recommend this approach.

The strategy 6 in stage 1 especially is interesting, if the approximative segmentation on unfocused images could start even before the focused image is achieved.

Chapter 11

Conclusion

First of all we have investigated different active contours and level set approaches. We perceived, that these methods are not able to segment the Vickers indentations appropriately, without a suitable initialization. As one general initialization for all images definitely is not suitable, we developed a localization method, based on downscaled images and a strict shape. As this method is highly robust but not accurate (not the intension), we initialize the precise methods (level set approach) with the approximative results and achieve very robust and highly accurate overall segmentation results especially for high quality images but also for low quality images (compared with approaches in literature).

Moreover, we have investigated the shape from focus approach which determines the depth of the image points by regarding differently focused images. As the determination of the shape information is not robust enough, we allied our proposed approach with the shape information and achieved even more competitive segmentation results for low-quality images.

Finally we explored the impact of unfocused images on the segmentation process. Actually the overall segmentation performance even increases, if the first approximative stage is based on the right unfocused images.

Bibliography

[1] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. *International journal of computer vision*, 22(1):61–79, 1997.

[2] T. Chan and A. Vese. Active contours without edges. *Image processing, IEEE transactions on*, 10(2):266–277, February 2001.

[3] T. Chan and W. Zhu. Level set based shape prior segmentation. *Computer Vision and Pattern Recognition CVPR IEEE Computer Society Conference on*, pages 1164–1170, 2005.

[4] L. Cohen. On active contour models and balloons. *CVGIP: Graphical models and image processing*, 53(2):211–218, March 1991.

[5] D. Cremers, M. Rousson, and R. Deriche. A review of statistical approaches to level set segmentation: integrating color, texture, motion and shape. *International journal of computer vision*, 72(2):195–215, 2006.

[6] M. Fernandes, Y. Gavet, and J.-C. Pinoli. Improving focus measurements using logarithmic image processing. *Journal of Microscopy*, doi: 10.1111/j.1365-2818.2010.03461.x, 2011.

[7] M. Gadermayr, A. Maier, and A. Uhl. Algorithms for microindentation measurement in automated Vickers hardness testing. In J.-C. Pinoli, J. Debayle, Y. Gavet, F. Cruy, and C. Lambert, editors, *Tenth International Conference on Quality Control for Artificial Vision (QCAV'11)*, number 8000 in Proceedings of SPIE, pages 80000M–1 – 80000M–10, St. Etienne, France, June 2011. SPIE.

[8] M. Gadermayr, A. Maier, and A. Uhl. A robust algorithm for automated microindentation measurement in vickers hardness testing. *Journal of Electronic Imaging*, 2012. accepted.

[9] D. Hetzner. Microindentation hardness testing of materials using ASTM E384. *Microscopy and Microanalysis*, 9:708–709, 2003.

[10] P. Hough. Method and means for recognizing complex patterns. *International Conference on High Energy Accelerators and Instrumentation, U.S. Patent 3,069,654*, 1962.

[11] Y. Ji and A. Xu. A new method for automatically measurement of vickers hardness using thick line hough transform and least square method. In *Proceedings of the 2nd International Congress on Image and Signal Processing (CISP'09)*, pages 1–4, 2009.

[12] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International journal of computer vision*, 1(4):321–331, 1988.

[13] C. Kiser, C. Musial, and P. Sen. Accelerating active contour algorithms with the gradient diffusion field. *Pattern Recognition ICPR 2008 19th International Conference on*, pages 1–4, 2008.

[14] W. Liming, Z. Qu, D. Yaohua, and Z. Miaoxian. Automatically analyzing the impress image of vickers hardness test using wavelet. *China mechanics engineering*, 15(6), March 2006.

[15] M. Macedo, V. Mendes, A. Conci, and F. Leta. Using hough transform as an auxiliary technique for vickers hardness measurement. In *Proceedings of the 13th International Conference on Systems, Signals and Image Processin (IWSSIP'06)*, pages 287–290, 2006.

[16] A. Maier, G. Niederbrucker, and A. Uhl. Measuring image sharpness for a computer vision-based Vickers hardness measurement system. In J.-C. Pinoli, J. Debayle, Y. Gavet, F. Cruy, and C. Lambert, editors, *Tenth International Conference on Quality Control for Artificial Vision (QCAV'11)*, number 8000 in Proceedings of SPIE, pages 80000N–1 – 80000N–10, St. Etienne, France, June 2011. SPIE.

[17] A. Maier and A. Uhl. Robust automatic indentation localisation and size approximation for vickers microindentation hardness indentations. In *Proceedings of the 7th International Symposium on Image and Signal Processing (ISPA 2011)*, pages 295–300, Dubrovnik, Croatia, Sept. 2011.

[18] V. Mendes and F. Leta. Automatic measurement of Brinell and Vickers hardness using computer vision techniques. In *Proceedings of the XVII IMEKO World Congress*, pages 992–995, Dubrovnik, Croatia, June 2003.

[19] S. K. Nayar. Shape from focus system. In *Computer vision and pattern recognition proceedings CVPR '92*, pages 302–308, 1992.

[20] S. Osher and J. A. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulations. *Journal of computational physics*, 79(1):12–49, November 1988.

[21] P. Panagiotidis, A. Antonatos, and G. Tsananas. Case depth determination by using Vickers micro-hardness test method. In *Proceedings of the 4th International Conference on Non-Destructive Testing, ICNDT '07*, Chania, Greece, Oct. 2007.

[22] J. C. Pinoli. The logarithmic image processing model: connections with human brightness perception and contrast estimators. *J. Math. Imaging Vis.*, 4(7):341–358.

[23] Z. Qu, Y. Guozheng, and Z. Yi. A new method for quickly and automatically analysis of the image of vickers hardness using wavelet theory. *Acta metrologica sinica*, 26(3):245–248, July 2005.

[24] T. Sugimoto and T. Kawaguchi. Development of an automatic Vickers hardness testing system using image processing technology. *IEEE Transactions on Industrial Electronics*, 44(5):696–702, Oct. 1997.

[25] C. Xu and J. L. Prince. Gradient vector flow: A new external force for snakes. In *IEEE Proc. Conf. on Computer Vision and Pattern Recognition*, pages 66–71, 1997.

[26] L. Yao and C.-H. Fang. A hardness measuring method based on hough fuzzy vertex detection algorithm. *IEEE Trans. on Industrial Electronics*, 53(3):963–973, 2006.

[27] G. P. Zhu, Q. S. Zeng, and C. H. Wang. Robust shape based active contour for circle detection. *The Imaging Science Journal*, 56(3):175–178, 2008.