

Technical Report

Pattern Recognition and Image Processing Group
Institute of Computer Aided Automation
Vienna University of Technology
Favoritenstr. 9/183-2
A-1040 Vienna AUSTRIA
Phone: +43 (1) 58801-18351
Fax: +43 (1) 58801-18392
E-mail: {pripbachelor}@prip.tuwien.ac.at
URL: <http://www.prip.tuwien.ac.at/>

PRIP-TR-118

November 27, 2008

Bachelor Thesis at PRIP
Collection of Summer Term 2008

Yll Haxhimusa, Walter G. Kropatsch, Martin Kampel (Ed.)

Preface

This is the first collection in the series of Bachelor Thesis at PRIP that will be published twice a year, namely the summer term collection will be published at the end of October, and the winter term collection at the end of March of each year.

In this collection three bachelor thesis are selected. These works are done during the summer semester 2008. The first work deals with the new image distance transformation called eccentricity transform. The author of the work, Martin Reiterer, presents a novel approximation that uses thinning of a given image region to reduce the amount of data for accelerating the eccentricity transform. In the second thesis, Hermann Czedik-Eysenberg shows a system for identifying unknown coins by matching their shape with a database of known coin shapes. In the third work, Andreas Zweng in his thesis deals with the problem of automatic number plate recognition.

We would like to thank all the authors and their supervisors for their support.

Yll Haxhimusa (Ed.)
Walter G. Kropatsch
Martin Kampel
Vienna University of Technology
Faculty of Informatics
Institute of Computer Aided Automation
Pattern Recognition and Image Processing Group

Vienna, October 2008

Contents

1	Thinning Based Eccentricity on Volumetric Data	2
2	Implementierung von Methoden zum Formvergleich handgepraegter Muenzen	31
3	Automatic Number Plate Recognition in Image Sequences	62

Thinning Based Eccentricity on Volumetric Data

Martin Reiterer
Supervisor: Walter G. Kropatsch

Thinning Based Eccentricity on Volumetric Data

Martin Reiterer

October 8, 2008

Abstract

Eccentricity has its roots in graph theory where it describes the length of the shortest path from one node to the node furthest away. This unit of measurement was picked up for describing shapes in digital images. It is called the eccentricity transform and defined in any dimension. Furthermore it is robust against salt & pepper noise and quasi invariant to articulated motion. On the other hand, computation takes a long time for large images, especially for typical three or higher dimensional ones. So in practice, one could consider to perform transformation only on an approximated and consequently reduced version of the initial volumetric dataset. In this paper a novel approximation is proposed that uses thinning of a given image region to reduce the amount of data for accelerating the eccentricity transform. Based on this concept, detailed studies are performed for verification of its descriptive performance, invariance to articulation and robustness against salt & pepper noise.

1 Introduction

In digital image processing, image transformations are a commonly used tool for feature enhancement. One rather new transform is called the eccentricity transform [8, 5, 6, 7, 10], which has its origins in graph theory. There, the term eccentricity specifies the longest of the shortest distances from node to all other nodes in a connected graph [2].

Based on digital images, the eccentricity transform computes for every pixel/voxel of a binary shape its shortest distance to the pixel/voxel furthest away. Benefits are its robustness against salt & pepper noise

and its quasi invariance to articulated motion. These are positive properties for description of natural three dimensional images, such as medical data. Problems on medical data rely on the following cases:

1. Caused by the method of image acquisition, they are more or less affected by noise and
2. parts of patients (like a hand) are often scanned in different poses.

For this purpose, eccentricity will be an efficient feature for image characterization, because it is invariant to both problems. On the other hand, computation takes a long time, especially on three or higher dimensional images. So, in practice it could be more convenient to perform the eccentricity transform based on only an approximation of the initial image. This yields to a reduced size of data and therefore to an acceleration in computation of the eccentricity transform.

In this paper, a new approximation, called Thinning Based Eccentricity is presented, which utilizes thinning of an image region primarily to reduce amount of pixels/voxels. Expected properties are:

- improved invariance to articulated motion and
- topology preserving data reduction

The real scale of them is studied in Section 3.3.

In the following section, a short recall of three dimensional (volumetric) images (Section 2.1), the eccentricity transform (Section 2.2) and the concept of thinning (Section 2.3) is presented. After this recall, the novel approach of Thinning Based

Eccentricity on Volumetric Data is described in detail in Section 3. Within this Section a detailed empirical study of descriptive performance, invariance to articulation and robustness against salt & pepper noise is provided. In Section 4, concepts for possible computation on unsegmented images is illustrated. The last part of this paper contains a short conclusion.

2 Recall

In this section volumetric images, shape matching with using the eccentricity transform and thinning is presented.

2.1 Volumetric images

In contrast to two dimensional images three dimensional ones are described by one additional spatial dimension. This leads to an increased size of these images in contrast to the two dimensional case and therefore also to an increased effort in computation on them.

Volumetric representations are frequently used in Computer Aided Design (CAD), Computer Aided Manufacturing (CAM), medicine, simulation, molecular biology, virtual reality or games.

Volume is a geometric property that specifies spatial distribution of three dimensional objects. Similar to a two dimensional digital image, which is composed of a set of pixels, a volumetric one is composed of a set of voxels. The term *voxel* stands for volume element and specifies a three dimensional discrete sampling point. Based on miscellaneous fields of application, such points can be aligned on different types of grids. The most used one is the Cartesian grid, where all voxels are aligned at the same distance to each other, parallel with the axes of the Cartesian coordinate system.

Figure 1 presents an object, described by voxels which are aligned on \mathbb{Z}^3 . Each voxel in this grid characterizes the image at one discrete position.

Volumetric images can be considered as a func-

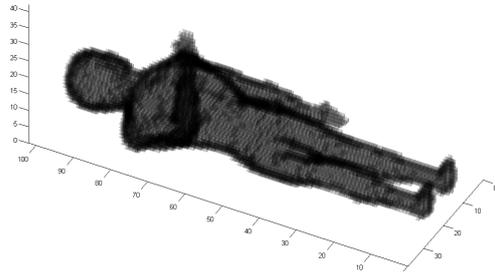


Figure 1: Example of a voxel represented 3D object.

tion that maps a vector of three elements (spatial position) into the domain of sampling values. In Function (1) a representation for binary volumetric images is presented, where background voxels (transparent cubes) are characterized by zeros and object regions (black cubes) by values of one.

$$f : \mathbb{Z}^3 \mapsto [0, 1] \quad (1)$$

In the three dimensional discrete domain, connectivity information can be considered with respect to 6 (N_6), 18 (N_{18}), or 26 (N_{26}) neighbors of one voxel. Definition of these neighborhoods is based on voxel adjacency. A voxel p is *n-adjacent* to a second one q , if Equation (2) is fulfilled.

$$N_n(p) \cap q \neq \{\} \quad (2)$$

$N_n(p)$ is called the n-neighborhood of p and contains all voxels that are n-connected to p . The definitions of different neighborhoods (types of connectivity) are explained in the remaining part of this section.

6 neighborhood ($N_6(p)$): Two voxels $V_1 = (x_1, x_2, x_3)$ and $V_2 = (y_1, y_2, y_3)$ are 6-connected, if they share one common voxel face, i.e.:

$$|x_1 - y_1| + |x_2 - y_2| + |x_3 - y_3| = 1 \quad (3)$$

$N_6(p)$ connectivity is also called face-adjacency. Figure 2 (a) illustrates the basic idea of sharing voxel faces. Figure 2 (b) pictures the 6-connected (face-adjacent) neighborhood of one voxel, which is marked by a dark gray value.

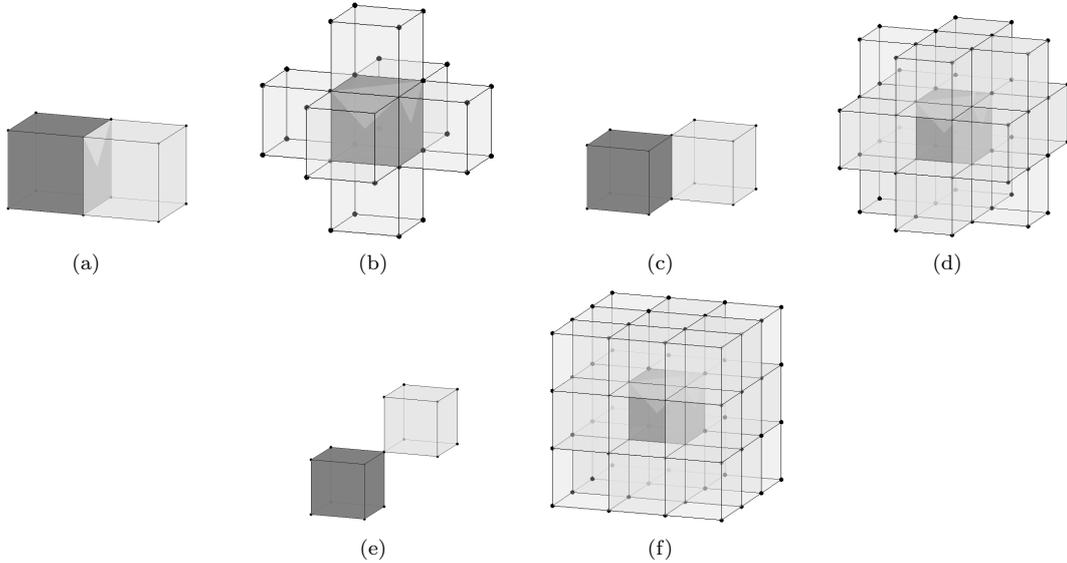


Figure 2: Illustration of 6-, 18- and 26-connectivity. Images (a) and (b) illustrate a reference voxel's 6-connected neighborhood. Concept of 18-connected neighborhood is pictured in images (c) and (d) and concept of 26-connected one in images (e) and (f).

18 neighborhood ($N_{18}(p)$): Two voxels are 18-adjacent if they share at least one common edge (edge-adjacency). Edge-adjacency is satisfied for the voxels V_1 and V_2 if they fulfill:

$$\begin{aligned} \max \{ |x_1 - x_2|, |y_1 - y_2|, |z_1 - z_2| \} = 1 \\ \wedge \\ |x_1 - x_2| + |y_1 - y_2| + |z_1 - z_2| \leq 2 \end{aligned} \quad (4)$$

In Figure 2 (b) the basic idea of sharing edges is illustrated. Figure 2 (c) pictures a voxel's 18-connected neighborhood.

26-connectivity ($N_{26}(p)$): Additional to face and edge adjacent voxels also voxels connected by at least one common vertex are considered as neighbors too (vertex-adjacency). Two voxels V_1 and V_2 are 26-connected if they fulfill Equation 5:

$$\max \{ |x_1 - y_1|, |x_2 - y_2|, |x_3 - y_3| \} = 1 \quad (5)$$

Figures 2 (e) and (f) illustrate the concept of $N_{26}(p)$ connectivity. In image (e), basic idea of vertex neigh-

borhood is illustrated. Additionally, 26-adjacent neighborhood of one reference voxel (dark one) is pictured in image (f).

2.2 Eccentricity

In Computer vision and pattern recognition usually relevant information is extracted out of an image region. One such approach is called the distance transform (DT). The DT computes for every point of an image the radius of the largest inscribable circle. Advantages are its robustness against scaling, orientation and positioning. The concept of extracting region skeletons is closely connected to the DT [21]. On the other hand, DT works only well if the input image is not affected by noise. The problem is that a noisy pixel or voxel is interpreted as a little hole within the image region. This causes that the DT differs in great extent between an image and its noisy version.

Another transform is the eccentricity transform

[10], which computes for every point of an image the geodesic distance to the point furthest away. Geodesic distance between two image elements (voxels), is defined as the length of the shortest path which connects them within the image region. This concept is robust against salt & pepper noise and quasi invariant to articulated motion. In the following subsections basic definition, concept of computation and the matching process are explained.

2.2.1 Basic definition

The eccentricity transform is defined in any dimension. In this paper, it is considered in the domain of three dimensional volumetric datasets. An image is specified by a binary set of voxels, divided into background and object points. A shape S is defined as a closed subset of voxels in \mathbb{Z}^3 . Between every pair of voxels (p_1, p_2) of a connected image region, one or more paths π can be built. Such paths are defined as a sequence of distinct object voxels with one start p_1 and one end point p_2 . Additionally each path is parameterized by a variable t , which is defined on the interval of $[0, 1]$, s.t. $\pi(0) = p_1$ and $\pi(1) = p_2$ and all voxels in between can be determined by a linear interpolation from start p_1 and end point p_2 along the path. The length of a path $\pi(t)$ is denoted by $\lambda(\pi)$. Let $\Pi(p_1, p_2)$ be the set of possible paths between two voxels p_1 and p_2 , where the minimum distance between them can be formally described by:

$$d(p_1, p_2) = \min \{ \lambda(\pi(p_1, p_2)) \mid \pi \in \Pi(p_1, p_2) \} \quad (6)$$

Based on this definition of minimum (geodesic) distance between two arbitrary voxels, the concept of eccentricity is described by Equation 7. There it is specified as the geodesic distance from one voxel p to the voxel furthest away q . These furthest away voxels are a subset of S and called eccentric points. Properties of them are explained in detail in [6].

$$ECC(S, p) = \max \{ d(p, q) \mid q \in S \}, \forall p \in S \quad (7)$$

We consider that $S \subseteq \mathbb{Z}^3$ is 6-connected and distances are Euclidean with the condition that paths have to be included in the volumes of the voxels of S .

2.2.2 Computation

Efficient computation of geodesic distances with respect to one concrete seed point can be performed by using Fast Marching algorithm (FM) [19]. This method uses Partial Differential Equations and reduces the effort of computation of geodesic distances from one voxel to all other voxels of an image region to $O(n \log(n))$, where n denotes the number of object voxels.

Based on FM that estimates for all object voxels $q \in S$ their geodesic distances to one concrete seed point p , the eccentricity transform can be computed by using Algorithm 1. In line one, the output

Algorithm 1 $ECC(S)$

Require: S

- 1: $ECC \leftarrow \text{zeros}(\text{sizeof}(S))$
 - 2: $P \leftarrow S$
 - 3: **for all** $q \in P$ **do**
 - 4: $D^S(q) \leftarrow FM(q, S)$
 - 5: **for all** $x \in S$ **do**
 - 6: $ECC(x) \leftarrow \max \{ ECC(x), D^S(x) \}$
 - 7: **end for**
 - 8: **end for**
 - 9: **return** ECC
-

image ECC is initialized by a three dimensional array of the same size as input image S . At the same time, all array values are set by default to value zero. In the following line, the set of possible eccentric points P is set to S . If P can be a-priori set to the eccentric points of S , the result would still be correct but with lower effort in computation. Following, the geodesic distance fields of all region voxels are estimated in Line four. These distance fields are compared element-wise to each other and only the maximums are considered as eccentricity values for the output image ECC . The effort of this procedural method is limited to $O(p n \log(n))$, where p denotes the number of voxels in P and n the number of voxels in S . Based on Line three, where P is set to S , computational effort can also be denoted by $O(n^2 \log(n))$.

2.2.3 Matching

The Eccentricity transform can be considered as a preprocessing step for feature extraction performed on binary images. Two regions cannot be directly compared on their transformed representation. For this reason a descriptor needs to be derived out of the transformed images to represent their features in a comparable way. Therefore histogram matching is used which compares the region’s eccentricity value distributions.

A histogram is a statistical tool to estimate distributions of e.g. numbers or colors. It is computed by counting the number of occurrences of each characteristic value. In the continuous case like for the eccentricity transform, such discrete values do not exist. For this reason intervals are built. Instead of counting the number of discrete characteristic value occurrences, the number of values within each interval is estimated. In both cases resulting histograms are k -dimensional vectors, where k denotes the number of distinct discrete values or interval classes. Each element of this vector represents the number of occurrences for one value or interval class. The eccentricity histogram $h(i, S)$ of a transformed shape S is defined by the following function:

$$h(i, S) = \frac{1}{|S|} \sum_{p \in S} I(i, ECC(p)), \forall i = 1 \dots k \quad (8)$$

$$I(i, ecc) = \begin{cases} 1 & | \frac{i-1}{k} \leq \frac{ecc-m}{M-m} < \frac{i}{k} \\ 0 & \text{otherwise} \end{cases}$$

where $ECC(p)$ denotes the eccentricity transform of S , m the smallest distance value and M the largest one. The number of distance classes k needs to be specified w.r.t. the size of the transformed image regions. In [7, 8], eccentricity values are usually categorized into $k = 200$ classes for images of the size of 128^3 voxels. For thin objects like lines, k should be set to the length of the image region’s diameter to avoid undersampling or oversampling of the eccentricity transformed shape.

Similarity of two image regions is determined by computing the distance between their eccentricity

histograms h_1 and h_2 . Possible options could be the ℓ_2 -norm or more sophisticated ones like χ^2 -statistic or diffusion distance [11]. Based on numerical experiments [7] it is concluded that the ℓ_2 -norm, which computes the mean-square-deviation between two histograms, performs well for matching eccentricity histograms. It is mentioned that the ℓ_2 -norm, which is described by Equation 9, gives comparable results to χ^2 and diffusion distance.

$$\delta(h_1, h_2) := \sqrt{\sum_{i=1}^k (h_1(i) - h_2(i))^2} \quad (9)$$

Function δ represents the ℓ_2 -norm. The histograms h_1 and h_2 are usually normalized to achieve invariance w.r.t. scaling. Furthermore, variable k denotes the number of different distance classes and therefore also the number of bins in the histograms.

With the help of histogram matching a value is gained, which gives us a value for the degree of dissimilarity between different objects. On the other hand this technique is also limited in its descriptive performance, because important information like topology are not considered. More detailed information to the topic of eccentricity histogram matching can be found in [7].

2.3 Thinning

Thinning is the process of peeling outer layers of an image region in a topology preserving way, till an approximation of a region skeleton is reached. On the other hand, detailed shape information gets lost, that is necessary to distinguish very similar objects of the same class. Nevertheless, in applications like image classification, where objects need to be assigned to specific classes, this disadvantage can be considered as a favorable feature.

In the ideal case, a thinning algorithm needs to fulfill a set of requirements:

1. *topology preservation*: Thinning needs to preserve the structure of the original object.

2. *centeredness*: Extracted medial lines or surfaces need to be centered as exact as possible w.r.t. the region's boundary.
3. *as thin as possible*: Extracted skeletons have to be as thin as possible.

In the domain of three dimensional images, approaches of thinning can be grouped with respect to their extracted skeletons. Medial line based as well as medial surface based thinning approaches can be distinguished. The former is used to reduce an object to a path-like minimum representation. Medial surface thinning shrinks an image region with respect to its principle surfaces. A comparison between them is illustrated in Figure 3. Figure 3 (a) pictures the medial line of a simple object. Referring to the same object, the approach of medial surface extraction is illustrated in Figure 3 (b).

2.3.1 Computation

In this subsection, computation of medial line as well as medial surface thinning is presented. Based on the architecture of voxel elimination, the whole pool of thinning algorithms can be categorized into:

- *Parallel approaches* erode a set of voxels at the same time. The thinning algorithms presented in this section divide one erosion step into several parallel subiterations per layer, to preserve the object's topology. Every subiteration is usually responsible for one kind of boundary voxels. Independent to each other, all points of one subiteration that satisfy at least one deletion condition are removed.
- *Sequential approaches* check their deletion conditions point after point, dependent of previously eliminated ones. This leads in contrast to parallel methods to a simplified set of deletion rules, because parallel algorithms have to check additionally if thinning requirements are fulfilled independent of points that are eliminated within the same subiteration. Otherwise, the result of thinning directly depends on the order of voxel elimination. This fact may lead

to different skeletons for the same object within different orientations.

- *Hybrid approaches* are a compromise between parallel and sequential ones. They combine advantages such as runtime efficiency of parallel thinning with simplified deletion rules of sequential methods. For this purpose, deletable points are marked in parallel and re-checking is performed within a final sequential step.

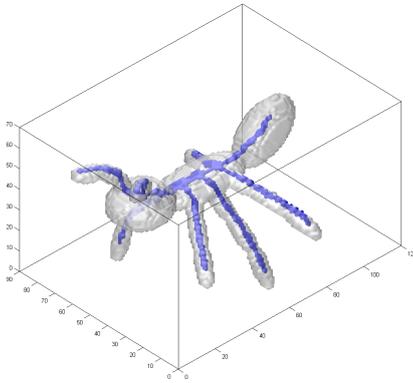
All existing methods are based on the same basic idea of layer-to-layer boundary erosion where within each deletion step, all boundary voxels that satisfy a set of deletion conditions are eliminated.

Medial Line Thinning

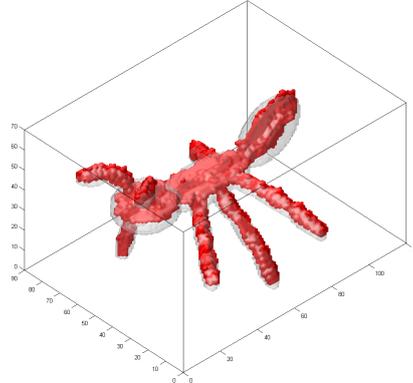
A thinning method for extracting approximations of medial lines is presented in [16]. This hybrid approach erodes an object B of a three dimensional finite and segmented image \mathbb{Z}^3 by checking the following set of conditions of simple point [18] and conditions of end point, for each border-voxel p in parallel:

1. $N_{26}(p) \cap (B \setminus \{p\})$ is not empty (end point)
2. $N_{26}(p) \cap (B \setminus \{p\})$ is 26-connected
3. $(\mathbb{Z}^3 \setminus B) \cap N_6(p)$ is not empty
4. $(\mathbb{Z}^3 \setminus B) \cap N_6(p)$ is 6-connected in the set $(\mathbb{Z}^3 \setminus B) \cap N_{18}(p)$

In [16] line thinning is implemented by Algorithm 2. Deletable voxels are eroded separately for the object directions of up (U), down (D), north (N), south (S), east (E) and west (W) boundary points. The main part of computation is performed within function SUBITER by checking conditions of simple point and conditions of end point in two steps. First, all border points of one direction (U, D, N, S, E, W) are collected in parallel to a linked list, for which conditions of simple point and conditions of end point are fulfilled. These collected border points are re-checked within a final sequential step. If conditions of simple point or conditions of end point are also fulfilled in the sequential process



(a) medial line extraction



(b) medial surface extraction

Figure 3: Comparison between medial line and medial surface based thinning algorithms. The left image shows an illustration of the medial line [16] of a volumetric modeled ant. On the right the same ant is reduced to its medial surface [14].

Algorithm 2 MedialLine (X)

Require: X

```

1: Y = X
2: repeat
3:   modified = 0
4:   modified = modified + SUBITER(Y, U)
5:   modified = modified + SUBITER(Y, D)
6:   modified = modified + SUBITER(Y, N)
7:   modified = modified + SUBITER(Y, S)
8:   modified = modified + SUBITER(Y, E)
9:   modified = modified + SUBITER(Y, W)
10: until modified > 0
11: return Y

```

(after deletion of points for which all conditions were satisfied), these points are deleted. For a detailed specification of this algorithm see [16].

This method was primarily developed for estimation of medial lines within segmented medical images of elongated objects like blood vessels or colons. Therefore it was designed in a way that it is robust against irregularities of the object's boundary. Based on its hybrid concept, spurious branches are even suppressed during the thinning process.

The medial lines produced by Algorithm 2 are only effectively computed for elongated binary objects. Computation on other binary object classes results in smooth lines, which approximate the object's structure based on centerlines for every elongated object part (see Figure 3 (a)).

The concept of branch generation can be illustrated in a simplified way with taking reference to flat objects. A new branch is built for every elongated part the length of which is bigger than half of the length of its diameter. Otherwise, the object part is only considered as an irregularity of the object's boundary. An illustration of this concept

is pictured in Figure 4. In contrast to elongated sub-object B, sub-object A is longer than the half of the length of its diameter. This leads to only a new branch on position C. Sub-object B is considered as an irregularity of the boundary, which results in an adapted medial line (D) of the elongated main object part. It should be considered that the centerlines produced in this way are not optimal approximations for the real skeleton, which contains much more branches.

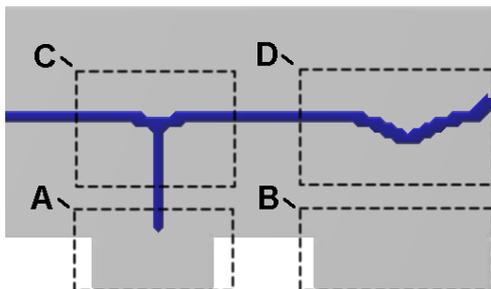


Figure 4: Illustration of branch generation by utilizing Algorithm 2. In contrast to elongated sub-object B, A leads to the creation of a new branch C. B is only considered as an irregularity of the objects boundary and leads to an adaption (D) of the medial line.

Furthermore, Algorithm 2 is not invariant to orientation, because generated lines depend on:

- The order of directions for which function SUBITER is called and
- the order, how voxels are eliminated within each subiteration (SUBITER)

We decided to use this method for verification of the performance of Thinning Based Eccentricity (personal communication with the author ¹), because it is comparatively fast and proved on natural objects too (successfully applied in medical image analysis).

Medial Surface Thinning

To illustrate the performance of Thinning Based

¹Kálmán Palágyi

Eccentricity in combination with medial surface thinning, tests are performed based on the thinning method proposed in [14]. This method computes thinning in a parallel way, where each boundary layer is eliminated within three subiterations. As presented in Algorithm 3, boundary voxels are reduced at the same time for each pair of opposite object directions (U & D, N & S, W & E). Voxels

Algorithm 3 Medial_Surface (X)

Require: X

- 1: $Y = X$
 - 2: **repeat**
 - 3: $Y = \text{DEL}(Y, \text{UD})$
 - 4: $Y = \text{DEL}(Y, \text{NS})$
 - 5: $Y = \text{DEL}(Y, \text{EW})$
 - 6: **until** on points are deleted
 - 7: **return** Y
-

of each subiteration (DEL) are eliminated in parallel if at least one of twenty-four matching masks is fulfilled. These masks have the size of twenty-eight voxels and are constructed in a way that they represent a set of conditions of simple point [18] and conditions of surface end point. The following surface end point condition ensures the generation of surfaces:

A black point p is a surface end point in a picture if the set $N_6(p)$ contains at least one opposite pair of white points.

Otherwise, distances within the masks are no good approximations of the Euclidean distance. Therefore, thinning based on Algorithm 3 is not invariant to orientation and scaling. Furthermore, geometrical correctness is guaranteed. Correctness w.r.t. topology can only be guaranteed by additional consideration of skeleton extraction based on distance maps like DT. For a more detailed description of this algorithm see [14].

3 Thinning Based Eccentricity

The Eccentricity Transform on volumetric images is associated with a big effort in computation.

Therefore, in practical applications it could be more convenient to perform it on only an approximation of the original image. Based on the motivation to find such a shape approximation, which accelerates speed of computation and preserves descriptive performance of the eccentricity transform at the same time, we developed the concept of Thinning Based Eccentricity (TBE). In this way an object is eroded on its outer layers by performing piecewise thinning operations (see Section 2.3). Based on thinning, objects are reduced in an efficient way, because they cover more voxels on their outer layers than on inner ones.

This leads to a smaller set of voxels, for which geodesic distances need to be computed. So, if an object can be reduced by performing thinning, TBE can help to describe it in an efficient way. On the other hand, TBE will produce a bigger computational effort than the eccentricity transform on the initial object, if no voxels can be eliminated (all skeleton properties are already satisfied). The reason therefore is based on the fact that additionally to the eccentricity transform on the whole object, effort of checking thinning conditions for the first erosion layer need to be considered.

In the ideal case, thinning also leads to a decreased number of possible eccentric points. This could be a favorable property for algorithms that perform FM on a predefined set of such points for faster computation of the eccentricity transform (Section 2.2.1). Nevertheless, there exist object classes where thinning does not lead to the elimination eccentric points. (see Figure 5).

Furthermore, thinning extracts the object’s structure which is characteristic for all objects of the same class. This is responsible for an increasing degree of similarity of objects of the same class the more they are thinned.

3.1 Computation

The procedural method of comparing images by TBE can be structured into three sequential steps. First,

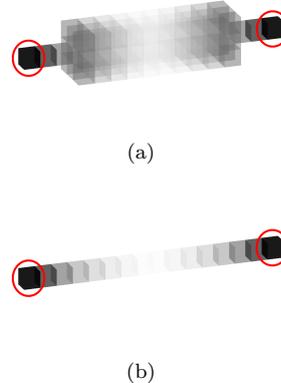


Figure 5: Illustration of a simple object where thinning does not reduce the number of eccentric points (encircled voxels). Image (a) shows the initial object and (b) its thinned version. High eccentricity values are represented by dark gray values and low eccentricity values by bright ones.

thinning is applied onto the initial image region and thereon the eccentricity transform is computed. This is still problematic, because the presented thinning methods are defined on 26-connected image and 6-connected background regions, denoted by $(26, 6)$ connectivity. On the other hand, our computation of the eccentricity transform is based on paths that are only able to travel across voxel faces, denoted by $(6, 26)$ connectivity. To solve this problem, $(26, 6)$ connected results of thinning need to be converted into $(6, 26)$ connected ones for computation of the eccentricity transform. For this purpose, all pairs of voxels (p, q) that are only connected by one single edge ($q \in N_{18}(p) \setminus N_6(p)$) or one single vertex ($q \in N_{26}(p) \setminus N_{18}(p)$) need to be connected additionally by a 6-adjacent path. This is realized by a morphological closing operation with two different structure elements. In the first step of dilation, every voxel is substituted by its 26-connected neighborhood, to gain 6-connected paths between p and q . Following, an erosion operation is performed w.r.t. the voxels 6-connected neighbors. In this way, 6-connectivity between two edge-adjacent voxels p

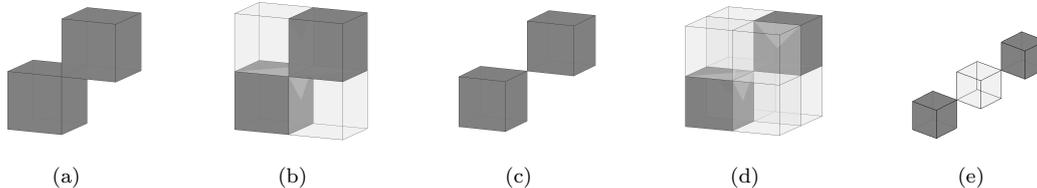


Figure 6: Illustration of converting 26-connected voxels into 6-connected ones by performing morphological closing. Image (a) pictures an edge-connected voxel pair and (b) the same pair after the morphological closing operation. In images (c) and (d) the same process is illustrated for a vertex-connected voxel pair. In situation (e), morphological closing does not produce 6-connectivity. The reference voxels (p and q) are always marked by dark gray values. Voxels that are produced by morphological closing are marked by light gray ones.

and q is established by selecting all paths of the length of one voxel between them (see Figure 6 (a) and (b)). If p and q are vertex-adjacent voxels, they get connected by all possible paths of the length of two voxels (see Figure 6 (c) and (d)).

It also need to be considered that artefacts are produced in the following cases:

- Holes with a diameter of less than two voxels are eliminated.
- Clefs with a thickness of less than two voxels are closed.
- If Condition 10 is satisfied for an arbitrary background voxel $r \in \mathbb{Z}^3 \setminus S$, it can occur that r becomes an isolated 6-connected image region. In this case, two region voxels (p and q) that are unconnected in the initial image S get connected by a not 6-adjacent path (see Figure 6 (e)).

One approach to avoid artefacts could be to adapt the sample rate (image resolution) in a way that there do no occur problematic voxel constellations.

$$r \in N_{26}(p) \setminus N_6(p) \wedge r \in N_{26}(q) \setminus N_6(q) \quad (10)$$

$$\forall p, q \in S$$

Based on the different approaches of thinning,

TBE can be computed w.r.t the object's medial lines (Algorithm 4) as well to its medial surfaces (Algorithm 5).

Algorithm 4 TBELine (X)

Require: X

- 1: $Y = \text{Medial_Line}(X)$
- 2: $Y = \text{dilate}(Y, N_{26}(p))$
- 3: $Y = \text{erode}(Y, N_6(p))$
- 4: $Y = \text{ECC}(Y)$
- 5: **return** Y

Algorithm 5 TBESurface (X)

Require: X

- 1: $Y = \text{Medial_Surface}(X)$
- 2: $Y = \text{dilate}(Y, N_{26}(p))$
- 3: $Y = \text{erode}(Y, N_6(p))$
- 4: $Y = \text{ECC}(Y)$
- 5: **return** Y

In a final step, the TBE transformed images are compared by performing eccentricity histogram matching (described in Section 2.2.3). An illustration of this concept is pictured in Figure 7. The procedural method of comparing two images by TBE is presented in Algorithm 6, where eccentricity histograms are estimated with the help of function *hist* and histogram matching is performed in function δ

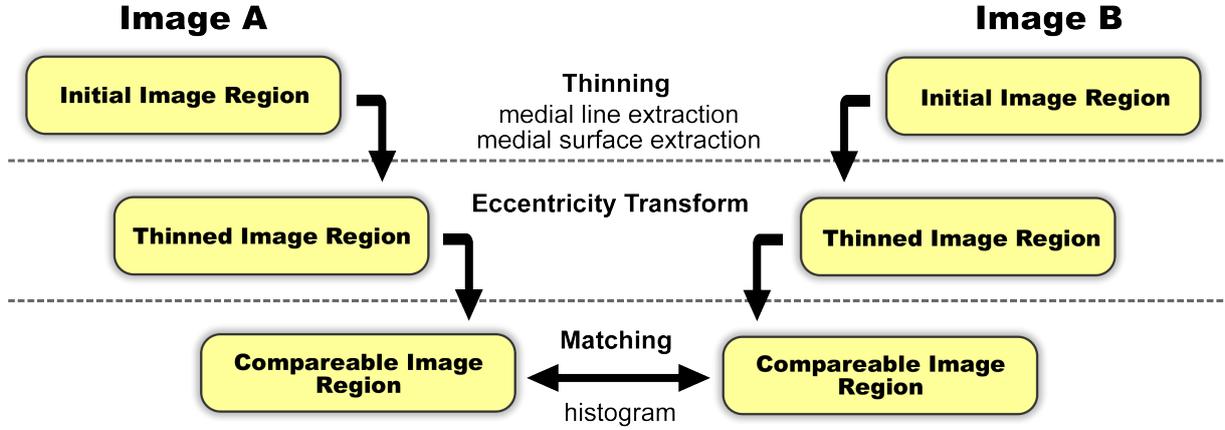


Figure 7: Concept of matching objects by TBE. Flow diagram of the three basic steps in computation.

(see Section 2.2.3).

Algorithm 6 $TBEDistance(X, Y, method)$

Require: $X, Y, method$

- 1: **if** $method = LINE$ **then**
 - 2: $tbeX = TBE_Line(X)$
 - 3: $tbeY = TBE_Line(Y)$
 - 4: **else**
 - 5: $tbeX = TBE_Surface(X)$
 - 6: $tbeY = TBE_Surface(Y)$
 - 7: **end if**
 - 8: $hX = hist(X)$
 - 9: $hY = hist(Y)$
 - 10: $dist = \delta(hX, hY)$
 - 11: **return** $dist$
-

3.2 Properties

Based on TBE’s sequential processing pipeline (presented in Section 3.1), descriptiveness directly depends on:

1. expressiveness of the extracted region skeleton and
2. appropriateness of the eccentricity histogram to describe this skeleton.

The first step of thinning is the most crucial one, because if a region cannot be uniquely identified by its skeleton, it never can be identified uniquely by the skeleton’s eccentricity histogram. Thus, influence of thinning onto the results of the eccentricity transform is studied in Section 3.3.

Besides the previously mentioned property of data reduction, TBE also offers increased invariance against articulated motion (discussed in Section 3.3.1) and extraction of information about structural composition of principle object parts. Based on this information, an object can be divided into its principle components to compare decided object parts.

On the other hand TBE is not robust against salt & pepper noise, because results of thinning differ in great extent between an object and its noisy version. For more detailed information see Section 3.3.1.

3.3 Experimental verification

Within this section Thinning Based Eccentricity is studied w.r.t. descriptive performance, data reduction, invariance to articulation and robustness against salt & pepper noise. For this purpose,

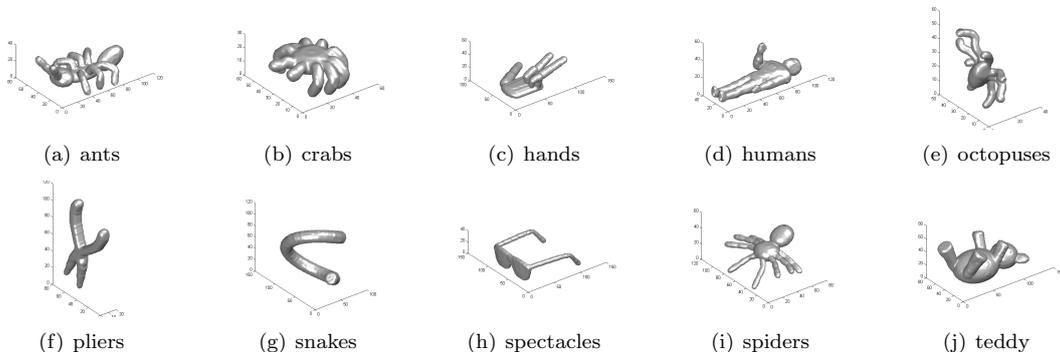


Figure 8: Overview of object categories of McGill Benchmark

	smallest object	largest object
volume (voxels)	2254	172779
eccentric points	76	2060
diameter	111	143
thickness	6	46

Table 1: Comparison between the average class properties of the smallest object (spectacle) and the largest one (teddy).

empirical tests are performed on McGill Benchmark database [24] that contains 455 objects, grouped into nineteen classes. Ten of them contain objects with different part articulations. For verification only this subset of 255 images is used to illustrate TBE’s invariance to articulated motion. Figure 8 pictures one representative of each of these ten object classes (ants, crabs, hands, humans, octopuses, pliers, snakes, spectacles, spiders and teddies).

Each image has the size of 128x128x128 voxels, where object voxels are characterized by ones and background voxels by values of zero. The smallest object is a spectacle and the largest one a teddy. A comparison between their volume, diameter and thickness is presented in Table 1.

Benchmarks are performed w.r.t. medial line as well as medial surface based thinning approaches, presented in 2.3.1. In this way, thinned object

versions are determined on every ten percent of thinning, where zero percent characterizes the initial object and one-hundred percent its minimum skeleton. Based on these thinned object versions, the eccentricity transform is computed and eccentricity histogram matching is performed. For this purpose each reduced object is matched separately against the whole set of thinned McGill database entities of the same thinning ratio. Objects that are used for querying are not considered at retrieval process. This restriction is necessary, because queried entities would always be most similar to itself and therefore not representative to evaluate retrieval performance. In the case of medial line based TBE, the number of histogram bins is always adapted to the queried object’s maximum eccentricity value. This avoids histogram artifacts caused by undersampling or oversampling of the medial line’s eccentricity values.

Results are evaluated with the help of the ratio values of *precision* and *recall*, which are commonly used for verification of static document sets. In practice, these magnitudes are imposed on dynamic systems such as web search engines [4], too. Let D denote a fixed set of database entities, Q the queried object, R the resulting object set and I the set of relevant entities. Relevant entities I are the whole subset of D whose entities are of the same class as Q . Based on these terms, precision is specified as the ratio of relevant retrieved objects I

to the whole number of retrieved objects R :

$$precision = \frac{I}{R} \quad (11)$$

Recall is defined as the ratio between relevant objects I within result set R and the total number of relevant objects within the set of database entities D :

$$recall = \frac{I}{D} \quad (12)$$

Referring to these magnitudes, we are able to evaluate the results of matching by means of precision-recall curves. Precision-recall curves oppose recall ratio on increasing precision steps. Thus, values nearby a recall value of one mark best results.

In Figures 16 and 17 the precision-recall rates of Algorithm 4 are pictured for each object class at every ten percent of thinning. Additionally, for each class average object elongatedness is illustrated, which is defined as the ratio between longitudinal and latitudinal object expansion. High thinning ratios produce bigger jumps in elongatedness than lower thinning values, because an objects contains more voxels on its outer layers than on inner ones. So, on inner layers more erosion steps have to be performed to achieve the same reduction than at one erosion step on an outer layer.

Same benchmarks are evaluated w.r.t. Algorithm 5. Resulting precision-recall curves are shown in Figures 18 and 19.

Based on the fact that measured images are usually affected by noise, we also studied its effect onto the results of TBE. For this reason previously described configuration of the test environment is adapted in a way that the queried object Q is affected by salt & pepper noise. Resulting precision-recall curves for Algorithm 4 are presented in Figures 20 and 21. Curves of medial surface based approach are pictured in Figures 22 and 23.

3.3.1 Discussion

Reduced computational effort, gained by TBE, directly depends on the amount of eliminated ob-

ject voxels. Based on the number of object voxels n , effort of the eccentricity transform is limited to $O(n^2 \log(n))$ (see Section 2.2.2). Effort of thinning is described by $O(n)$.

At TBE, comparatively low computation time of thinning is used to accelerate computation of the eccentricity transform. Based on the number of skeleton voxels m , effort of TBE is limited to:

$$O((n - m) + m^2 \log(m)) \quad (13)$$

So, the time needed for object description is reduced, the more voxels are eliminated.

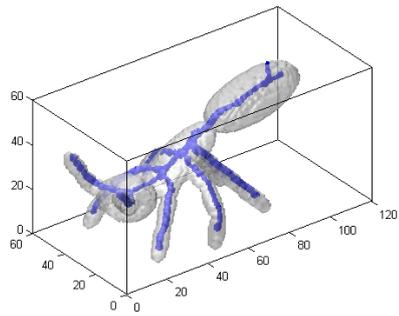
As already mentioned in Section 3.2, thinning improves eccentricity's quasi invariance against articulated motion. If the eccentricity transform is performed on an articulated object, geodesic paths are attracted by the concavity of the articulation point. This yields to different results between an object and its articulated version. Figure 9 (a) illustrates a simple 3-D object and (b) its eccentricity histogram. The same object is pictured in Figure 9 (c) within a different pose. As shown in Figure 9 (d), eccentricity histograms of these objects are a little bit different, based on different paths around the point of articulation.

At TBELine, this point is even reduced to one single voxel. Therefore all geodesic paths of the eccentricity transform which move through the point of articulation are forced to move through this single voxel. So, histograms of articulated objects that are transformed by TBELine are not disturbed by articulation.

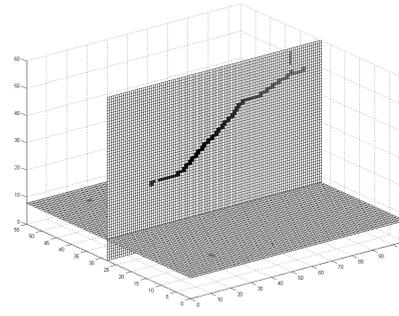
At TBESurface, the point of articulation is described by surfaces. Results differ less between articulated objects, the more their articulation point is thinned into the normal-direction of its articulation axis. Similar to TBELine, best invariance can be achieved if the thickness of the articulation point is reduced to only one voxel w.r.t. to the normal-direction of its articulation axis.

Descriptiveness:

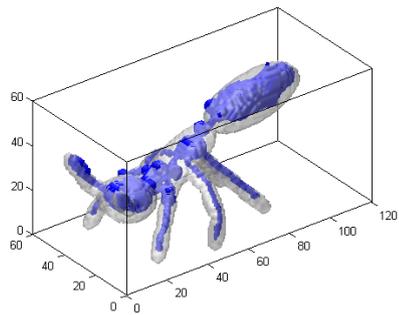
Our experiments on the McGill Benchmark database have shown that descriptiveness of TBE depends on the properties of the method that is used for



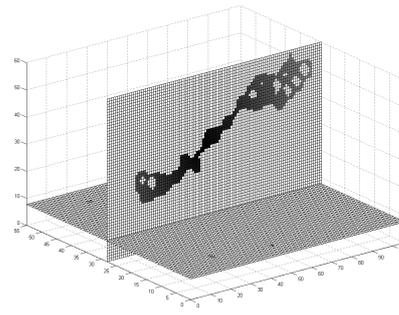
(a) unnoisy skeleton



(b) slice of unnoisy skeleton



(c) noisy skeleton



(d) slice of noisy skeleton

Figure 13: Comparison between skeletons of noisy and unnoisy images, which are generated by medial line extraction. Image (a) illustrates the medial lines of an unnoisy image region and (b) its slice representation. In picture (c) medial lines of the same but noisy image region are presented. A slice view of this noise caused skeleton is shown in (d).

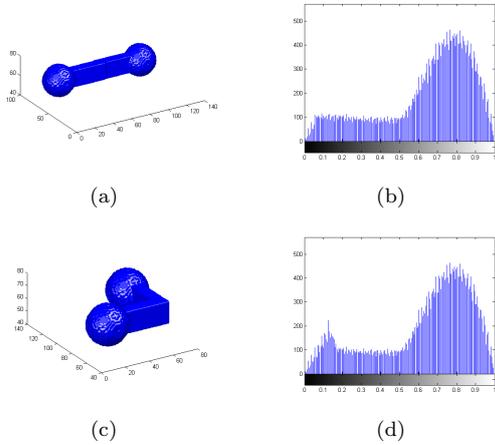
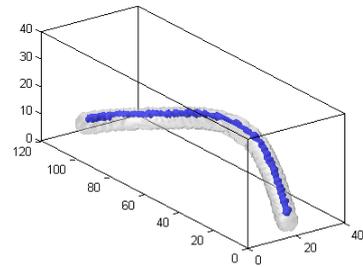


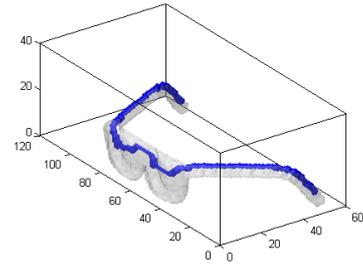
Figure 9: Comparison of the eccentricity histograms of articulated objects. Image (a) presents the initial object and (b) its eccentricity histogram. In (c) the same object is illustrated in an articulated position. Image (d) shows the eccentricity histogram of the articulated object.

thinning. In contrast to the eccentricity transform, similar or better precision-recalls (see Figures 16 and 17) are gained by TBELine only below **seventy percent** of thinning. At higher thinning ratios, it occurs that objects of different classes get more and more similar to each other. For example spectacles with very small glasses converge to a similar medial line like snakes (see Figure 10). The reason therefore is based in losing important information about the objects latitudinal distribution and form properties. Algorithm 2 is designed to produce centerlines without much (spurious) branches. This yields to the behavior that small glasses are interpreted as irregularities of the spectacles boundary (see Section 2.3.1).

At the same time, eccentricity histograms based on TBELine are converging to a noisy appearance for thinning ratios around one-hundred percent (see Figure 11). These errors are based on 6-connected object discretization (see Section 2.1). The eccentricity transform of a paraxial line results in a flat



(a) snake



(b) spectacle

Figure 10: Comparison between objects, which are reduced to similar medial lines.

histogram (Figure 12 (a) and (c)). If the same line is rotated into a non-paraxial orientation, it is discretized to a sequence of connected paraxial line-segments (see Figure 12 (b)). This yields to spikes within its eccentricity histogram (see Figure 12 (d)). The spikes are caused by additional voxels, which are needed for representation of a rotated 6-connected line of uniform thickness. In Figure 12 (b), the voxels are marked that are responsible for the spikes within the line's eccentricity histogram. This leads to an increased sensitivity of eccentricity histogram matching to line orientation. Based on this fact, we observed bad discriminative performance for objects of McGill database that are fully thinned by Algorithm 2.

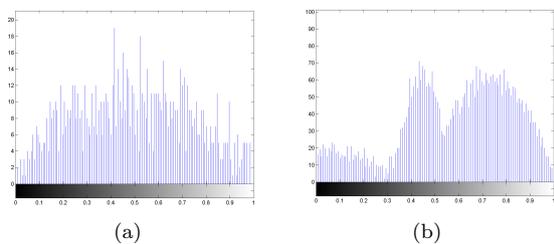


Figure 11: Comparison between the eccentricity histograms of a ninety (a) and one-hundred percent (b) TBELine thinned ant.

At TBESurface, skeletons of different object classes are much less similar at high thinning ratios. The reason is based on preservation of information about the object's spatial distribution. At previously described experiments, one hundred percent of reduction can be performed without any restrictions in discriminative performance for the classes of ants, hands, octopuses and spectacles.

Otherwise, power of TBESurface still depends on descriptiveness of eccentricity histogram matching. One limitation is that histograms do not consider information about the object's structure. Based on this fact, the eccentricity histograms of crabs and teddies are very similar, because a crab has more but thin legs and a teddy less but thicker ones.

Furthermore, histograms of objects of different

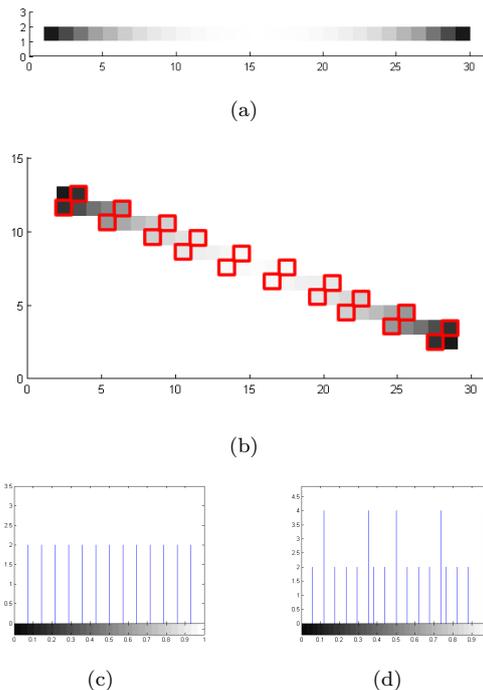


Figure 12: Comparison between the eccentricity histograms of a paraxial and a non-paraxial line. Image (a) illustrates a paraxial line and (c) its eccentricity histogram. The same line, rotated at twenty degree, is pictured in (b). The eccentricity histogram of the rotated line is shown in (d). Dark voxels denote high eccentricity values and bright ones low eccentricity values.

classes may be more similar than histograms of different objects of the same class. This well known problem is already discussed in [8].

Robustness against noise:

At empirical studies with noisy images, descriptive performance of both TBESurface and TBELine is decreased at high ratios of object thinning (see Figures 20, 21, 22 and 23).

At line thinning, an increasing degree of dissimilarity between thinned noisy and thinned unnoisy objects is observed for increasing thinning ratios. An example for them is presented in Figure 13. In Images (a) and (b) the skeleton of an ant and its sliced representation is illustrated. The sliced representation visualizes two dimensional projections of cuts through the volume. In (c) and (d) the same object is affected by salt & pepper noise. Based on sensitivity of TBELine to noise, the skeleton of the noisy image does not represent the object’s structure any more. The reason is based on preservation of topology during thinning operation [12], because holes within the initial object must not merge with the background or another hole. So, at the fully thinned object, every noise caused hole keeps isolated by object voxels.

On the other hand, surface thinning is more robust against noise. In contrast to sequential thinning approaches, parallel ones, such as TBESurface, usually consider a wider area of neighborhood voxels for checking conditions of simple point and conditions of end point. Based on this bigger neighborhood, noise caused holes of the size of one voxel can be determined much easier.

But if the objects are reduced by high thinning ratios, also descriptive performance of TBESurface is decreased by noise caused errors (see Figures 22 and 23).

Computational Effort:

Based on the previously discussed problems, best precision recalls for TBELine as well as TBESurface are gained only below seventy percent of possible thinning. At this thinning ratio, computational

	smallest object	largest object
region voxels	1114 (1132)	55481 (137292)
eccentric points	32 (33)	1996 (1929)
reduced effort	77% (77%)	90% (38%)

Table 2: Comparison between the properties of the smallest object and the largest one of McGill database. Both objects are thinned at seventy percent by TBELine. Values for noisy images are denoted in parenthesis.

	smallest object	largest object
region voxels	2047(2066)	80348 (103881)
eccentric points	58 (58)	2074 (2142)
reduced effort	18% (16%)	80% (65%)

Table 3: Comparison between the properties of the smallest object and the largest one of McGill database. Both objects are thinned at seventy percent by TBESurface. Values for noisy images are denoted in parenthesis.

effort is computed by using Formula 13. In the Tables 2 and 3, object properties and reduced effort w.r.t. the eccentricity transform are presented for the smallest object and the largest one (referring to noisy/unnoisy line-/surface-thinned objects). On the whole database, computation time is decreased with line-thinning by 89% percent for unnoisy images. For noisy ones, computational effort can only be reduced by 46% percent, because less voxels are eliminated (see Figure 13). If the database is thinned by TBESurface, most reduction (69%) is achieved for noisy images, because layer erosions are also performed w.r.t. noise caused holes within the object. For unnoisy images, TBESurface achieved an acceleration of 59%.

4 Future tasks

In the broader sense, results of TBE directly depend on quality of image segmentation, too. So, if image region extraction is already affected by errors of impreciseness, descriptive performance of the extracted skeletons and the thereon performed

eccentricity transform is reduced in succession. A promising method for avoiding such errors is based on skeletonization algorithms that are directly performed on unsegmented gray scale images.

One such approach [23] is studied in the field of medial line localization for blood vessels which are scanned by MRI. Based on a clearly predefined start P_{START} and end point P_{END} within a blood vessel, a geodesic path is computed, where distances between two pixels are weighted by their gray values. This process generates a centerline that is attracted by every concavity of the gray scale region. For this reason, image points before and behind the extracted line are weighted by their gray values. Based on them, the image is distorted in a way that the curve gets more straight. This process is repeated till the line cannot be straighten any more. Within a final step, inverse transformations need to be performed to obtain the initial image appearance.

Based on these multiple steps of image distortion and back-transformation, method [23] is very time-consuming. A less complex alternative could be to perform centerline adaption by moving the line into the direction of the normals of every line point. This normals can be estimated by utilizing e.g. DT. In two or higher dimensions, a line segment is characterized by normals in more than one direction. In the ideal case, the line is modified in a way that the length of all opposite normals, that are starting at the centerline and ending at the region boundary, are equal. An Illustration of this concept is shown in Figure 14.

Furthermore, also performance of eccentricity histogram matching (see Section 2.2.3) can be improved w.r.t. robustness and flexibility by performing computation directly on the unsegmented gray scale image. In this way, the eccentricity transform is computed for a primary region, which is specified by a predefined threshold T_1 . Additionally a second threshold T_2 is defined which during matching allows that geodesic paths are able to move also through the surrounding area of the primary region. If segmentation is not affected by errors,

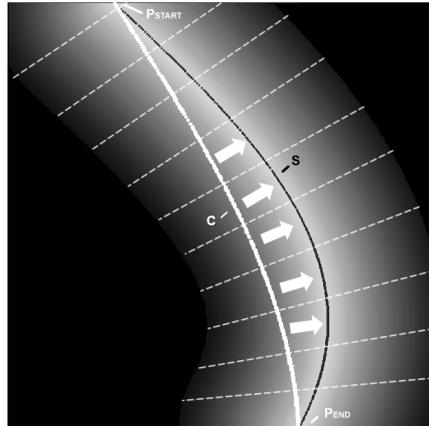


Figure 14: Illustration of the concept for centerline adaption. The initial determined line C is shifted into the direction of its longest normals. This process is repeated till the length of all opposite normals is equal. The resulting centerline is denoted by S .

eccentricity computation is only performed on the primary image region. So, errors of segmentation can be eliminated by adapting the image region's eccentricity histogram to them of the queried one. Basic concept of this approach is illustrated in Figure 15.

5 Conclusion

This paper presents a novel approach for efficient matching of three dimensional objects by utilizing the eccentricity transform. Based on the motivation of decreasing computational effort by reducing the object's volume, thinning is performed. On the thinned object, the eccentricity transform is performed faster the more voxels can be eliminated, because geodesic distances only need to be computed for a reduced set of voxels. In a final step, the thinned and eccentricity transformed objects are compared by matching their eccentricity histograms. For empirical verification of Thinning Based Eccentricity's descriptive performance, invariance to articulation and robustness against noise, studies are performed on a database of articulated 3D shapes. The eccentricity transform

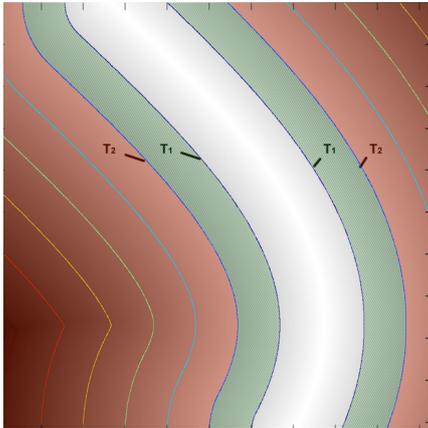


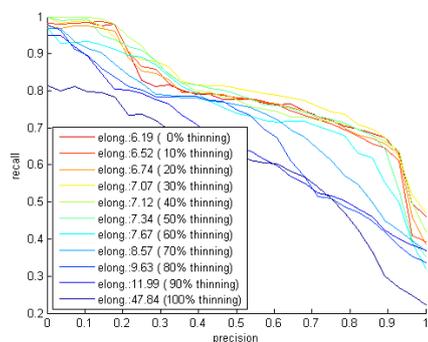
Figure 15: Concept for computation of the eccentricity transform directly on a unsegmented gray scale image. First, primary eccentricity histogram is determined based on the image region specified by threshold T_1 . At eccentricity histogram matching, the region specified by threshold T_2 is used for histogram adaptation.

is computed on medial line as well medial surface based thinning approaches. At both variants, best results are gained around seventy percent of possible voxel elimination. For this thinning ratio, computational effort is reduced at ninety percent for line thinning and sixty percent for surface based approach. Evaluated output demonstrated better results at descriptive power and more robustness against noise for surface based process. Furthermore, descriptiveness of Thinning Based Eccentricity depends on restrictions of histogram matching too. Based on additional consideration of structural high level information descriptive performance may be improved.

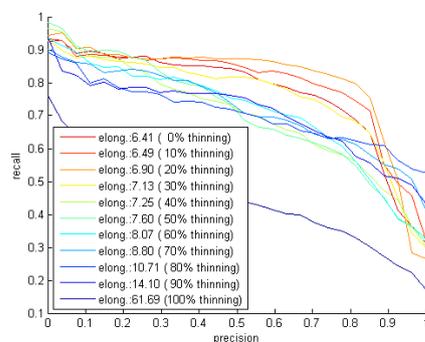
References

- [1] Gunilla Borgefors. Digital distance transforms in 3d images using information from neighbourhoods up to $5 \times 5 \times 5$. *Computer Vision and Image Understanding*, 88/1:24–53, 2002.
- [2] Gary Chartrand and Linda Lesniak. *Graphs & digraphs*. CRC Press, 2005.
- [3] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing, 3rd ed.* Prentice Hall, Upper Saddle River, NJ., 2008.
- [4] David Hawking, Nick Craswell, Paul Thistlewaite, and Donna Harman. Results and challenges in web search evaluation. *Proceeding of the eighth international conference on World Wide Web*, pages 1321–1330, 1999. Toronto, Canada.
- [5] A. Ion, S. Peltier, Y. Haxhimusa, and W. G. Kropatsch. Decomposition for efficient eccentricity transform of convex shapes. *CAIP 2007, 2007*. Vienna, Austria, Springer, Heidelberg (2007).
- [6] A. Ion, S. Peltier, and W. G. Kropatsch. Computing the eccentricity transform of a polygonal shape. *CIARP 2007, 2007*. Vienna, Austria, Springer, Heidelberg (2007).
- [7] A. Ion, G. Peyrè, Y. Haxhimusa, S. Peltier, W. G. Kropatsch, and L. Cohen. Shape matching using the geodesic eccentricity transform - a study. *The 31st annual workshop of the Austrian Association for Pattern Recognition (OAGM/AAPR)*, pages 97–104, May 2007. Schloss Krumbach, Austria. OCG.
- [8] Adrian Ion, Nicole M. Artner, Gabriel Peyré, Salvador B. López Mármol, Walter G. Kropatsch, and Laurent Cohen. 3d shape matching by geodesic eccentricity. *Workshop on Search in 3D (in conjunction with CVPR 2008)*, 2008.
- [9] Adrian Ion, Walter G. Kropatsch, and Eric Andres. Euclidean eccentricity transform by discrete arc paving. *DGCI 2008, LNCS 4992*, pages 213–224, 2008.
- [10] W. G. Kropatsch, A. Ion, Y. Haxhimusa, and T. Flanitzer. The eccentricity transform (of a digital shape). In *13th DGCI*, pages 437–448, October 2006. Springer.

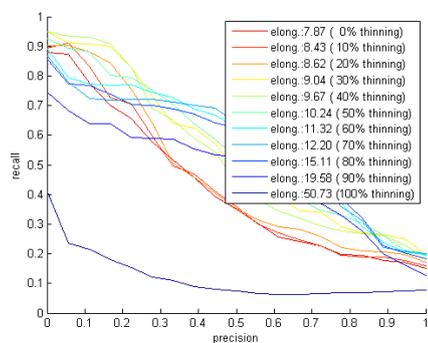
- [11] H. Ling and K. Okada. Diffusion distance for histogram comparison. *In Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2006.
- [12] Cherng-Min Ma. Topology preservation on 3d images. volume 2060, pages 201–207. SPIE, 1993.
- [13] Cherng-Min Ma, Shu-Yen Wan, and Her-Kun Chang. Extracting medial curves on 3d images. *Pattern Recognition Letters*, 23:895–904, 2002.
- [14] Kálmán Palágyi. A 3-subiteration 3d thinning algorithm for extracting medial surfaces. *Pattern Recognition Letters*, 23:663–675, 2002.
- [15] Kálmán Palágyi and Attila Kuba. A 3d 6-subiteration thinning algorithm for extracting medial lines. *Pattern Recognition Letters*, 19:613–627, 1998.
- [16] Kálmán Palágyi, Erich Sorantin, Emese Balogh, Attila Kuba, Csongor Halmai, Balázs Erdoehelyi, and Klaus Hasegger. A sequential 3d thinning algorithm and its medical applications. *IPMI 2001*, pages 409–415, 2001.
- [17] D. Reniers, J. J. van Wijk, and A. Telea. Computing multiscale curve and surface skeletons of genus 0 shapes using a global importance measure. *IEEE Transactions on Visualization and Computer Graphics*, 14/2:355–368, 2008.
- [18] P. K. Saha and B. B. Chaudhuri. Detection of 3-d simple points for topology preserving transformations with application of thinning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16:1028–1032, 1994.
- [19] J. Sethian. Level sets methods and fast marching methods. *Cambridge Univ. Press*, 2nd, 1999.
- [20] H. Sundar, D. Silver, N. Gagvani, and S. Dickinson. Skeleton based shape matching and retrieval. *Proceedings of the Shape Modeling International 2003 (SMI03)*, 2003.
- [21] Son Tran and Liwen Shih. Efficient 3d binary image skeletonization. *Proceedings of the 2005 IEEE Computational Systems Bioinformatics Conference Workshop (CSBW'05)*, 2005.
- [22] M. van Eede, Diego Macrini, Alexandru Telea, Cristian Sminchisescu, and Sven J. Dickinson. Canonical skeletons for shape matching. *ICPR*, 2:64–69, 2006.
- [23] R. Joop van Heekeren, Frank G. A. Faas, and Lucas J. van Vliet. Finding the minimum-cost path without cutting corners. *In SCIA*, pages 263–272, 2007.
- [24] J. Zhang, K. Siddiqi, D. Macrini, A. Shokoufandeh, and S. Dickinson. Retrieving articulated 3-d models using medial surfaces and their graph spectra. *International Workshop On Energy Minimization Methods in Computer Vision and Pattern Recognition*, 2005. Springer.



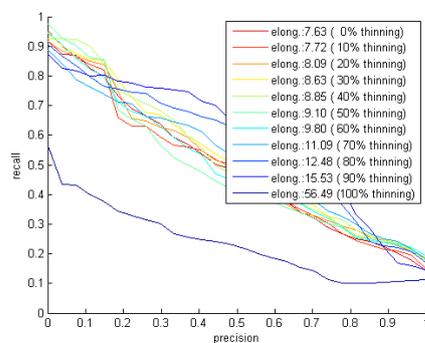
(a) ants



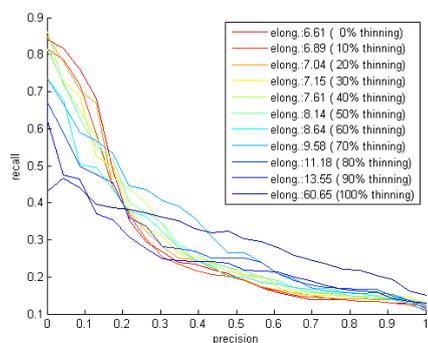
(b) crabs



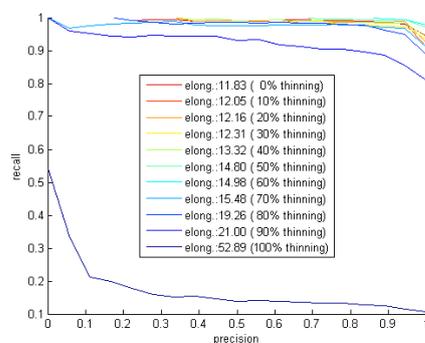
(c) hands



(d) humans

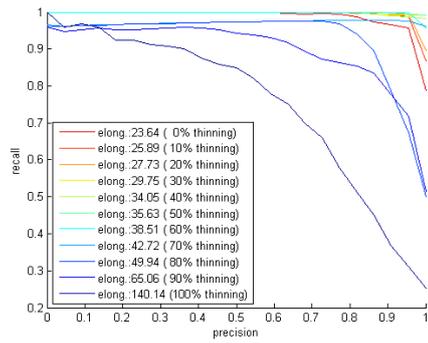


(e) octopuses

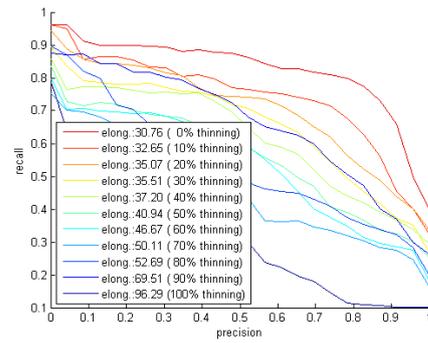


(f) pliers

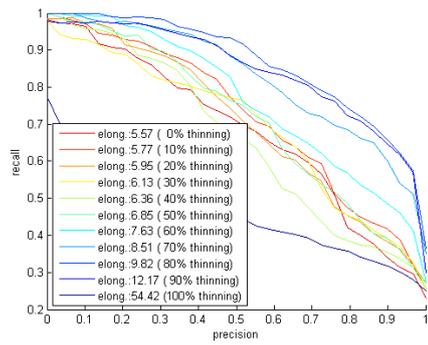
Figure 16: Precision recall rates of medial line based TBE for the classes ants, crabs, hands, humans, octopuses and pliers of McGill Benchmark database. Curves are pictured for thinning ratios from zero to one hundred percent of possible reduction. Additionally, average object elongatedness values are quoted to every object class and its steps of thinning.



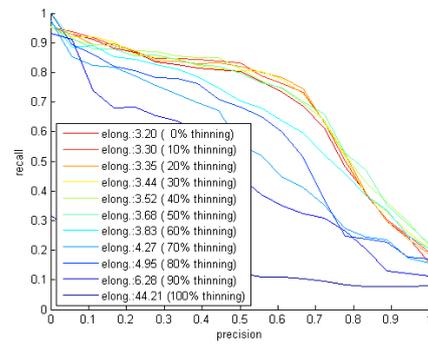
(a) snakes



(b) spectacles

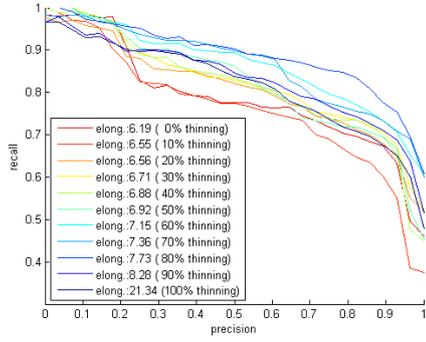


(c) spiders

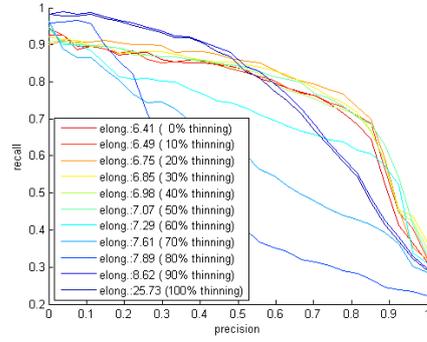


(d) teddies

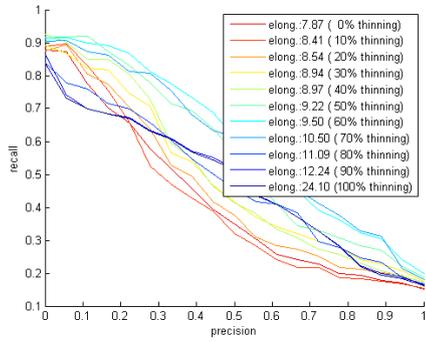
Figure 17: Precision recall rates of medial line based TBE for the classes snakes, spectacles, spiders and teddies of McGill Benchmark database. Curves are pictured for thinning ratios from zero to one hundred percent of possible reduction. Additionally, average object elongatedness values are quoted to every object class and its steps of thinning.



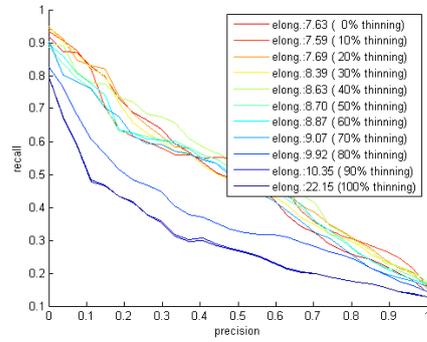
(a) ants



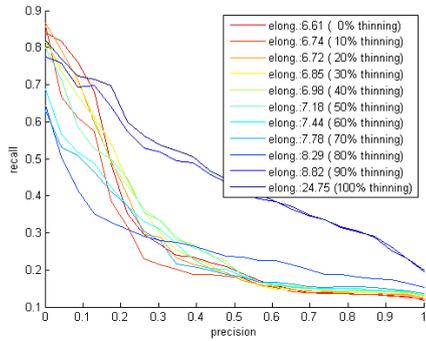
(b) crabs



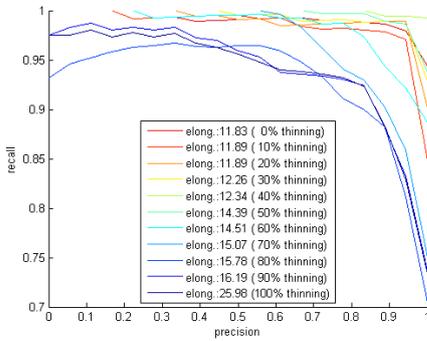
(c) hands



(d) humans

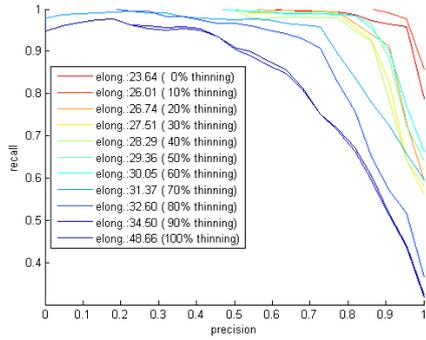


(e) octopuses

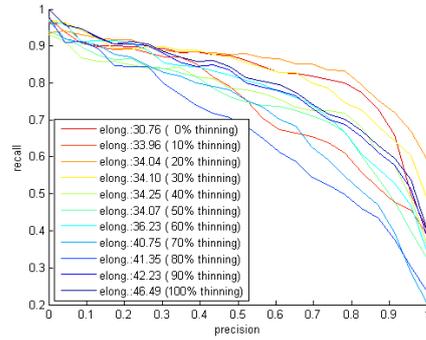


(f) pliers

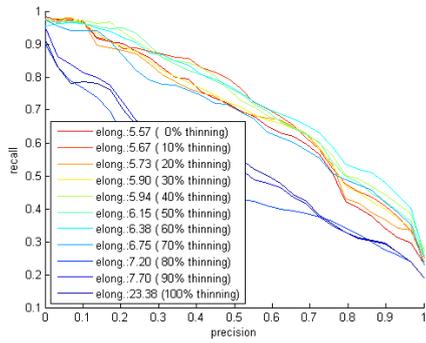
Figure 18: Precision recall rates of medial surface based TBE for the classes ants, crabs, hands, humans, octopuses and pliers of McGill Benchmark database. Curves are pictured for thinning ratios from zero to one hundred percent of possible reduction. Additionally, average object elongatedness values are quoted to every object class and its steps of thinning.



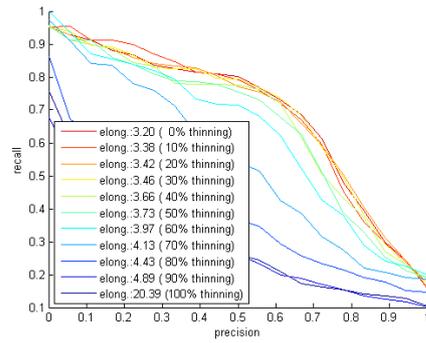
(a) snakes



(b) spectacles

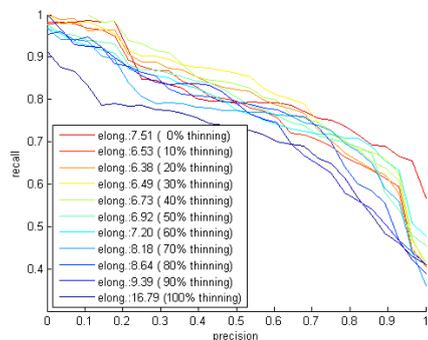


(c) spiders

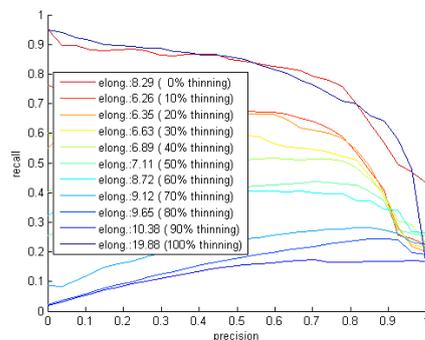


(d) teddies

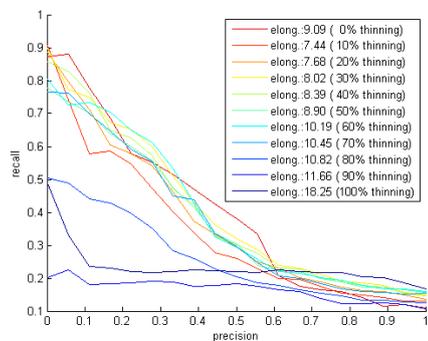
Figure 19: Precision recall rates of medial surface based TBE for the classes snakes, spectacles, spiders and teddies of McGill Benchmark database. Curves are pictured for thinning ratios from zero to one hundred percent of possible reduction. Additionally, average object elongatedness values are quoted to every object class and its steps of thinning.



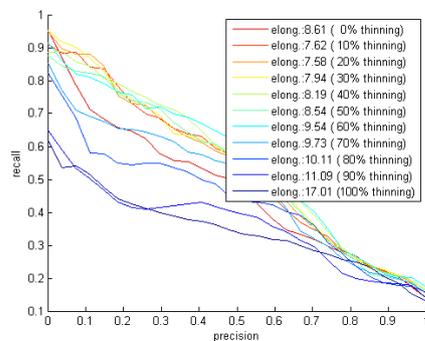
(a) ants



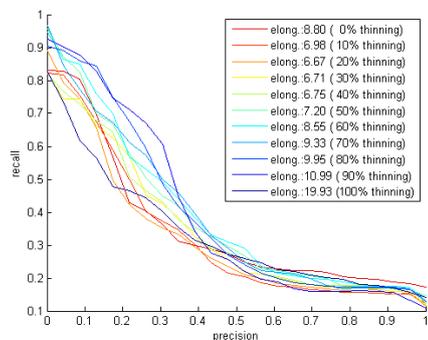
(b) crabs



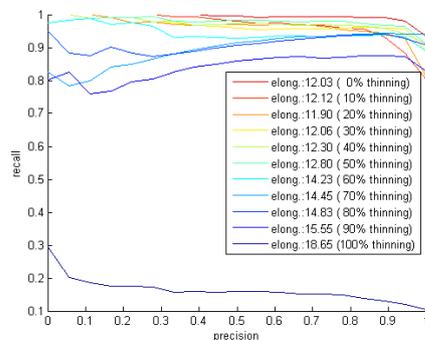
(c) hands



(d) humans

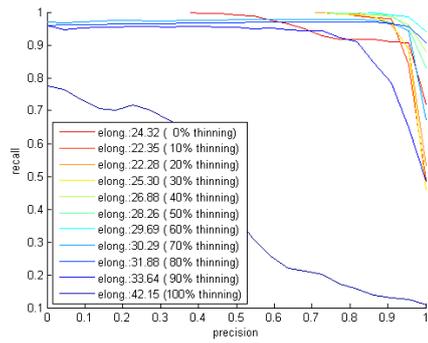


(e) octopuses

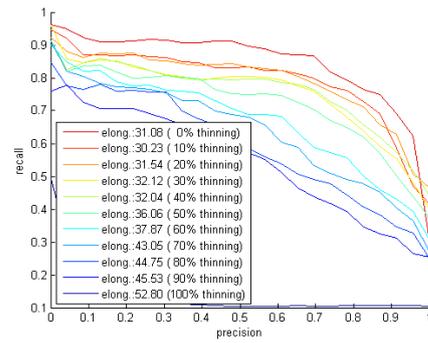


(f) pliers

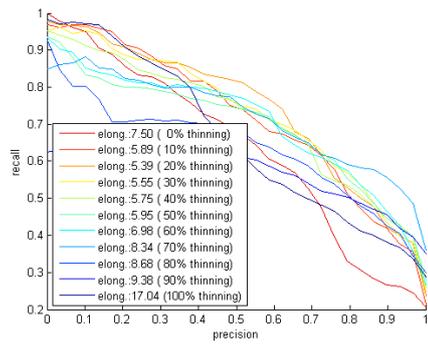
Figure 20: Precision recall rates of medial line based TBE for the classes ants, crabs, hands, humans, octopuses and pliers of McGill Benchmark database. Curves are pictured for noisy images at thinning ratios from zero to one hundred percent of possible reduction. Additionally, average object elongatedness values are quoted to every object class and its steps of thinning.



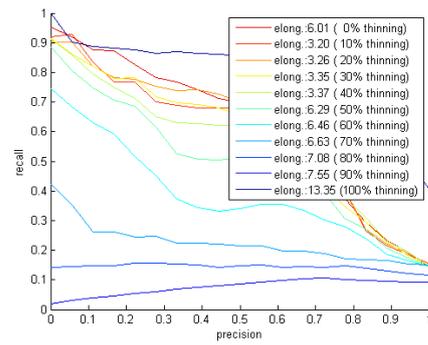
(a) snakes



(b) spectacles

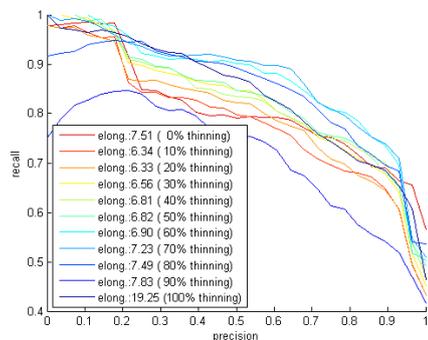


(c) spiders

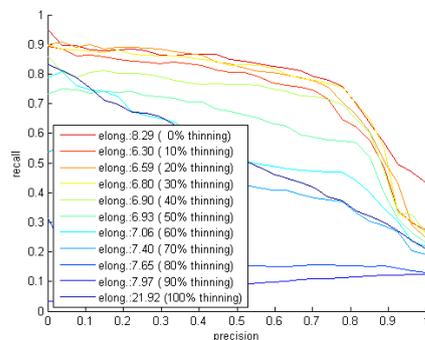


(d) teddies

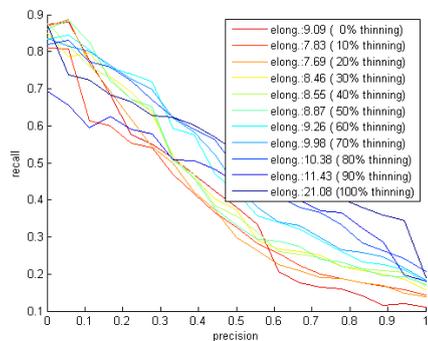
Figure 21: Precision recall rates of medial line based TBE for the classes snakes, spectacles, spiders and teddies of McGill Benchmark database. Curves are pictured for noisy images at thinning ratios from zero to one hundred percent of possible reduction. Additionally, average object elongatedness values are quoted to every object class and its steps of thinning.



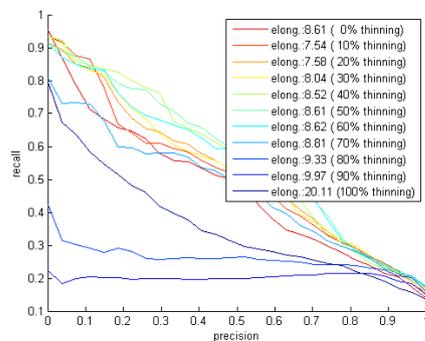
(a) ants



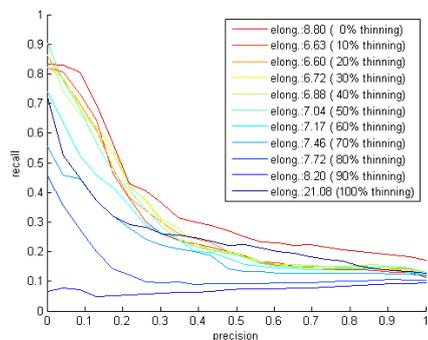
(b) crabs



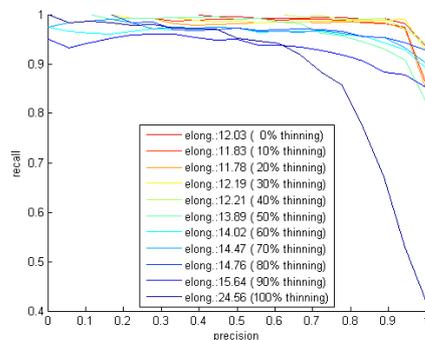
(c) hands



(d) humans

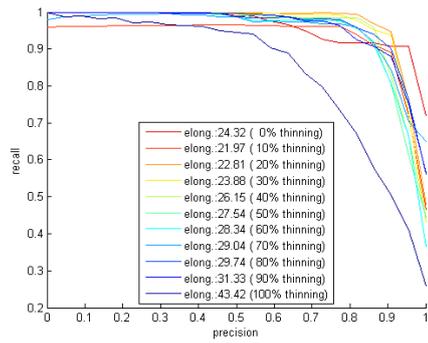


(e) octopuses

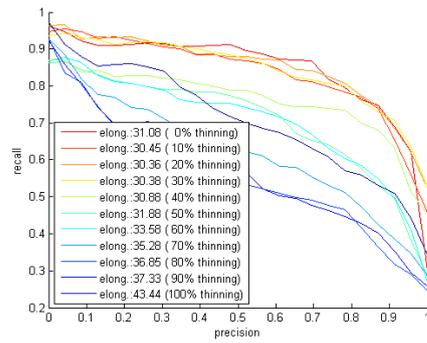


(f) pliers

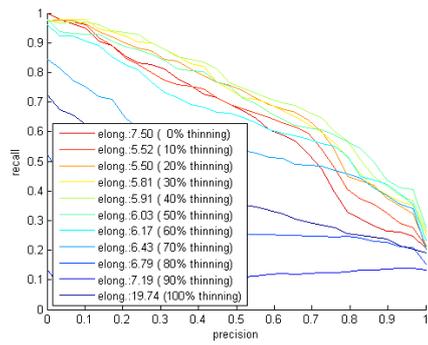
Figure 22: Precision recall rates of medial surface based TBE for the classes ants, crabs, hands, humans, octopuses and pliers of McGill Benchmark database. Curves are pictured for thinning ratios from zero to one hundred percent of possible reduction. Additionally average object elongatedness values are quoted to every object class and its steps of thinning.



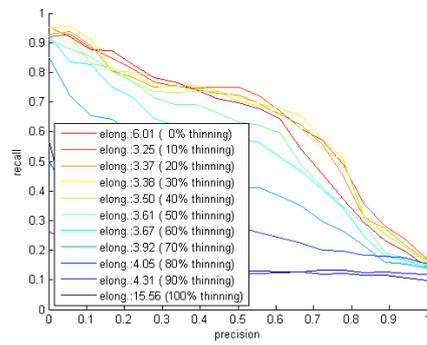
(a) snakes



(b) spectacles



(c) spiders



(d) teddies

Figure 23: Precision recall rates of medial surface based TBE for the classes snakes, spectacles, spiders and teddies of McGill Benchmark database. Curves are pictured for thinning ratios from zero to one hundred percent of possible reduction. Additionally average object elongatedness values are quoted to every object class and its steps of thinning.

**Implementierung von Methoden zum
Formvergleich handgepraegter Muenzen**

Hermann Czedik-Eysenberg
Supervisor: Martin Kampel, and Reinhold Huber-Moerk

Implementierung von Methoden zum Formvergleich handgeprägter Münzen

Hermann Czedik-Eysenberg
Bachelorarbeit für das Studium
Software & Information Engineering
Technische Universität Wien
e0526426@student.tuwien.ac.at

3. September 2008

Betreuung:
Dr. Martin Kampel¹
Dr. Reinhold Huber-Mörk²

ABSTRACT

Reliable methods for the identification of ancient coins on images are required to combat the illicit trade of stolen coins in the World Wide Web. Because of the manual minting, use-wear signs and the impact of centuries, ancient coins feature unique outlines that can be used to distinguish and identify them. It is shown how a system for identifying unknown coins by matching their shape with a database of known coin shapes can be implemented. The dissimilarity of the outline of coins on two images is calculated by a shape matching method that is invariant to scale, translation, rotation and mirroring. First the images have to be segmented to separate the coins from the background. Then the coin boundaries are extracted and converted to a format that is suitable for comparison. The boundary information can be extracted and stored preliminarily for a database of known coin images to accelerate the identification of unknown coins. Experiments on a large dataset of ancient coins show that the proposed identification method is competitive.

¹Technische Universität Wien, 183-2, Seminar 183.125

²Austrian Research Centers GmbH - ARC

1 Einleitung

Handgeprägte historische Münzen unterscheiden sich von maschinell gefertigten modernen Münzen in mehrerer Hinsicht. Durch Verwendung von unterschiedlichen (auch manuell angefertigten) Münz-Stempeln und die händische Prägung ist jede Münze ein Unikat. Historische Münzen des gleichen Typs (z.B. ein römischer Denar) unterscheiden sich untereinander in Größe und Textur wesentlich stärker als moderne Münzen. Diese Einzigartigkeit wird durch Abnutzung und Witterungseinflüsse über die Jahrhunderte noch unterstützt. Sie weisen insbesondere eine individuell eindeutige Form des Münzrandes auf. Diese Eigenschaft lässt sich zur automatisierten Identifikation solcher Münzen anhand von digitalen Aufnahmen heranziehen.

1.1 COINS-Projekt

Diese Arbeit entsteht im Rahmen des COINS-Projekts³ (Combat On-line Illegal Numismatic Sales) der Europäischen Union. An diesem Projekt sind mehrere Forschungsinstitute, Universitäten und kulturelle Einrichtungen aus Italien, Österreich, Großbritannien, Belgien und Rumänien beteiligt. Das Ziel ist Technologien zu entwickeln, die die Identifikation und Verfolgbarkeit historischer Münzen ermöglichen, um den illegalen Handel von gestohlenen Münzen im Internet zu verhindern. Es wurden drei Kernaktivitäten definiert:

1. Standardisierung von numismatischen Beschreibungen und Datenstrukturen
2. Entwicklung einer Web-Suchmaschine
3. Forschung an bildbasierten Erkennungsmethoden für historische Münzen

Die erste Aktivität umfasst die Entwicklung von standardisierten Organisations-Tools für Münzdatenbanken und die Erstellung eines mehrsprachigen Thesaurus, um eine übernationale Zusammenarbeit beim Kampf gegen illegalen Münzhandel zu ermöglichen. Zudem werden durch Museen Referenz-Kollektionen von Münzbildern mit zugehörigen standardisierten Beschreibungen bereitgestellt.

Die Web-Suchmaschine soll dazu in der Lage sein automatisiert im Internet illegale Verkäufe von historischen Münzen auffinden. Geplant ist die Installation einer Server-Infrastruktur, die das World Wide Web unüberwacht indiziert und durch die Kombination von Textsuche mit Bildanalysen die Aufdeckung von eventuell illegalen Münzverkäufen möglich macht.

Die Web-Suche ist eng verbunden mit der dritten Aktivität, in die der Inhalt dieser Arbeit fällt. Neue Methoden zur zuverlässigen Klassifikation und Identifikation von historischen Münzen sollen entwickelt, implementiert und evaluiert werden. Zwei beispielhafte Anwendungsszenarien der zu entwickelnden Methoden wurden identifiziert:

1. Eine unbekannte verdächtige Münze (die z.B. bei einem Polizei-Einsatz sichergestellt wurde) soll identifiziert werden. Dazu wird diese mit einer Datenbank von bekannten gestohlenen Münzen verglichen.
2. Eine bekannte gestohlene Münze wird auf dem Markt gesucht. Beschreibungen und Aufnahmen der Münze sind verfügbar und können mit den Informationen zu anderen Münzen, die beispielsweise aus dem Index einer zuvor durchgeführten Web-Suche stammen, verglichen werden.

Beide Anwendungsfälle benötigen Methoden zur zuverlässigen Identifikation von historischen Münzen. In Abschnitt 6 wird untersucht, ob und wie das in dieser Arbeit vorgestellte Verfahren die nötigen Voraussetzungen für diese Aufgaben erfüllt.

³<http://www.coins-project.eu/>

1.2 Problemdefinition

Klassifikation bezeichnet die Einteilung von einzelnen Objekten in unterschiedliche Kategorien. Münzen könnten beispielsweise nach ihrem Ursprungsland oder ihrem Wert klassifiziert werden. Meistens ist die Klassifikation nach dem Münztyp von Interesse. Unter einem Münztyp werden alle Münzen einer Währung mit gleicher Denomination und Design zusammengefasst (z.B. 1 Euro-Cent Münzen aus Österreich).

Identifikation ist ein Spezialfall der Klassifikation bei dem jedes Objekt eine eigene Kategorie darstellt. Die Identifikation einer Münze ist genau dann positiv wenn sie mit demselben physischen Objekt (sich selbst) verglichen wird.

In dieser Arbeit wird ein Verfahren, durch das antike Münzen mit Hilfe von Aufnahmen (Fotos oder Scans) identifiziert werden können, vorgestellt und evaluiert. Zur Identifikation wird eine einzelne Aufnahme einer unbekanntes Münze mit einer Datenbank von bereits identifizierten Münzaufnahmen verglichen. Um die unbekanntes Münze zu identifizieren, müssen in der Datenbank Aufnahmen derselben Münze gefunden werden. Es ist nötig, einzigartige Eigenschaften der individuellen Münzen zu bestimmen und miteinander zu vergleichen. Hier wird dafür die Form der äußeren Kontur, die bei historischen Münzen in der Regel nicht einem perfekten Kreis entspricht, verwendet. Ziel ist es, eine schnelle Methode zum Vergleich der Konturen zweier Münzbilder zu finden, diese auf Zuverlässigkeit zu überprüfen und mit anderen Ansätzen zur Identifikation zu vergleichen. Der Vergleich und das Berechnen der Ähnlichkeit zwischen zwei Formen wird allgemein als Shape Matching bezeichnet [5].

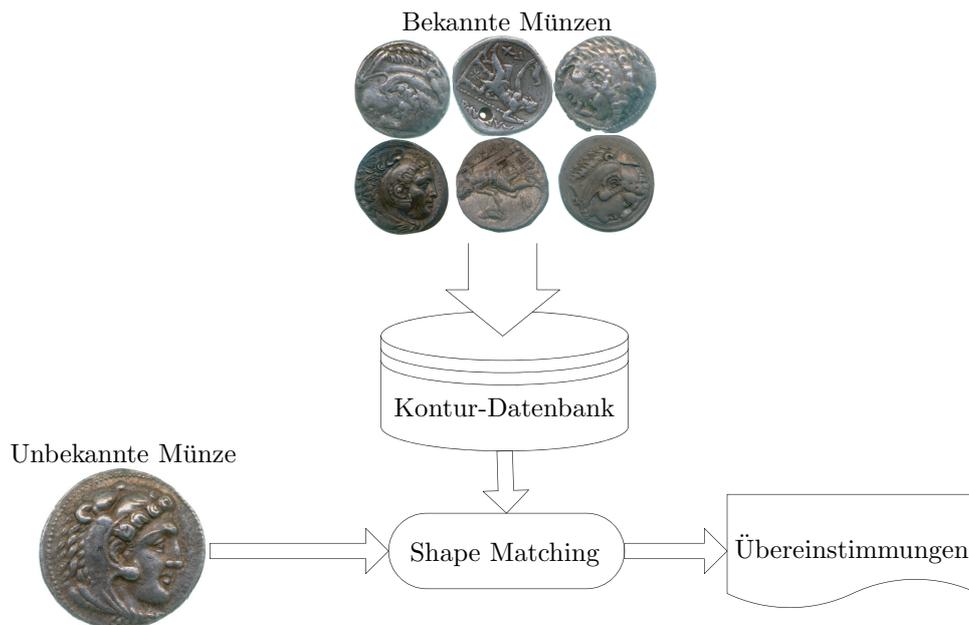


Abbildung 1: Ablauf der Identifikation einer unbekanntes Münze

Abbildung 1 zeigt den allgemeinen Ablauf der Identifikation einer Münze. Die Form der unbekanntes Münze wird mit der Form jeder bekannten Münze aus der Datenbank verglichen, um Übereinstimmungen zu finden.

1.3 Bisherige Forschung und Motivation

Bei der zeitgleichen Umstellung von 12 europäischen Währungen auf den Euro im Jahr 2002 entstanden große Mengen von Münzen, deren Klassifikation nur automatisiert erfolgen konnte. Es gab daher vermehrte Forschung und Publikationen zur automatisierten Verarbeitung von Münzbildern. Mehrere Methoden, deren Zuverlässigkeit in Experimenten gezeigt wird, zur Klassifikation von modernen Münzen wurden entwickelt [6, 16, 18, 22]. Diese Ansätze eignen sich für die Klassifikation von historischen Münzen allerdings nur bedingt [23]. Die Identifikation individueller moderner Münzen ist kein Forschungsthema und aufgrund der Homogenität zwischen den einzelnen Elementen eines Münztyps nur schwer möglich. Die Methoden, die zur Klassifikation von modernen Münzen entwickelt wurden, sind für historische Münzen ungeeignet, da grundlegende Annahmen bei diesen nicht erfüllt sind: Die Kontur von handgeprägten Münzen ist nicht regelmäßig und weist keine Rotationssymmetrie auf. Um eine zuverlässige Klassifikation und Identifikation von historischen Münzen zu ermöglichen, müssen daher neue Methoden entwickelt werden, die die Besonderheiten dieser Münzen beachten.

In [25] werden zwei Ansätze vorgestellt, die die eindeutige Form des Umrisses verwenden, um antike Münzen zu identifizieren: Shape-Contexts-Matching ist eine Methode des Shape Matching, die gute Ergebnisse beim Vergleich allgemeiner Formen liefert [3]. Der zweite Ansatz wird als Robust Shape Matching bezeichnet und ist eine Shape-Matching-Methode, die speziell auf die Eigenschaften von Münzrändern (Abweichung von einem Kreis) optimiert ist. Beide Methoden liefern vielversprechende Ergebnisse, wurden allerdings bisher nur mit kleinen Datensätzen (12 Bilder von Münzen) evaluiert. In dieser Arbeit wird die Idee des zweiten Ansatzes, nämlich die spezielle Darstellung der Münzkontur als Signal, welches die Abweichung von einem Kreis angibt, aufgegriffen und weiter optimiert. Die Methode wird hier als DCSM (engl.: *deviation from circular shape matching*) bezeichnet. Beide Methoden werden mit Hilfe einer größeren Münzbilder-Datenbank getestet und miteinander verglichen.

Eine andere Möglichkeit, um eine Münze eindeutig zu beschreiben, sind lokale Deskriptoren des Münzstempels. Dabei werden ausgezeichneten Punkte (engl.: *interest points*) auf der Münzoberfläche, die durch besondere Eigenschaften hervortreten, extrahiert. In [25] wird dieser Ansatz zur Klassifikation und Identifikation von historischen Münzen angewendet. Identifikation über lokale Deskriptoren wird in dieser Arbeit mit der hier vorgeschlagenen Methode experimentell verglichen und diskutiert.

1.4 Bildrepräsentation

Um in der Folge Methoden zur Identifikation von Münzen unter Verwendung von digitalen Aufnahmen beschreiben zu können, ist es nötig von einer Konvention für die digitale Repräsentation eines Bildes auszugehen. Die Möglichkeit der Einbeziehung von Farbinformation soll in dieser Arbeit nicht untersucht werden, daher werden nur Grauwert-Bilder verarbeitet. Ein digitales Grauwert-Bild soll durch eine $n \times m$ -Matrix I dargestellt werden, deren Einträge $I(i, j)$ jeweils die Intensität eines quadratischen Bildpunktes (Pixel) mit den Koordinaten (i, j) als einen von endlich vielen Werten beschreiben. Bildpunkte werden mit 8 Bit abgespeichert, d.h. es sind 256 unterschiedliche Grauwerte von 0 (Schwarz) bis 255 (Weiß) möglich. Die Zeilen- bzw. Spaltenanzahl n und m der Matrix werden in der Folge auch als Höhe bzw. Breite des Bildes bezeichnet.

Liegt eine digitale Farbaufnahme einer Münze vor, weil diese beispielsweise aus der vorhandenen Datenbank eines Museums stammt, so muss diese zuerst in ein Grauwert-Bild konvertiert werden. Üblicherweise wird es sich dabei um ein RGB-Bild handeln, welches aus drei $n \times m$ -Matrizen R , G und B besteht, deren Einträge $R(i, j)$, $G(i, j)$ und $B(i, j)$ die Intensitäten der drei Grundfarben Rot, Grün und Blau, aus denen sich

ein Bildpunkt mit den Koordinaten (i, j) zusammensetzt, repräsentieren. Eine Konvertierung eines RGB-Bildes zu einem Grauwert-Bild I ergibt sich durch

$$I(i, j) = 0.3 \cdot R(i, j) + 0.59 \cdot G(i, j) + 0.11 \cdot B(i, j) \quad \forall i = 1, \dots, n; j = 1, \dots, m. \quad (1)$$

Die Gewichtungsfaktoren der einzelnen Farben in Gleichung 1 resultieren aus einer unterschiedlichen Wahrnehmung der Farben durch das menschliche Auge, und werden so beispielsweise auch im NTSC-Standard verwendet [8]. Für Konvertierungen von anderen Bildrepräsentationen zu dem gewünschten Grauwert-Format sei auf Lehrbücher zu digitaler Bildverarbeitung verwiesen (z.B. [5, 10]).

1.5 Aufbau der Arbeit

Im folgenden Abschnitt 2 werden die Methoden, die einen Formvergleich von historischen Münzen ermöglichen, beschrieben. Abschnitt 3 stellt eine erstellte Implementierung als Software-Bibliothek kurz vor. Diese wird in in Abschnitt 4 angewandt um mit Hilfe von unterschiedlichen Testdaten die Methoden experimentell zu optimieren und zu evaluieren. Die Ergebnisse der Experimente auf einer großen Datenbank von historischen Münzen werden in Abschnitt 5 diskutiert und mit der Auswertung von anderen Verfahren verglichen. In Abschnitt 6 werden die Resultate zusammengefasst und Schlussfolgerungen, insbesondere für die praktische Anwendung und weitere Forschung, gezogen.

2 Methoden

2.1 Segmentierung

Segmentierung bezeichnet die Trennung des Objekts (in diesem Fall der Münze) vom Hintergrund und stellt allgemein einen wichtigen Vorverarbeitungsschritt in der Bildverarbeitung dar [10]. Für jeden einzelnen Bildpunkt muss entschieden werden, ob er zum Objekt gehört oder nicht. Dabei entsteht ein Binärbild, auf dem jedes Pixel des Objekts den Wert 1 und alle anderen den Wert 0 haben. Eine zuverlässige und exakte Segmentierung ist für die hier beschriebene Anwendung besonders relevant, da der äußere Rand der Münze, der später mit dem einer anderen Münze verglichen werden soll, über das Binärbild ermittelt wird. Fehler bei der Segmentierung wirken sich also direkt auf das Ergebnis des Formvergleichs aus.

Elementare Segmentierungsverfahren können allgemein nach verschiedenen Ansätzen in drei Kategorien eingeteilt werden [10]: (1) Pixelbasierte Methoden, die nur die Grauwerte der einzelnen Pixel verwenden; (2) regionenorientierte Verfahren, die die Grauwerte in zusammenhängenden Bereichen untersuchen; und (3) kantenbasierte Methoden, die Kanten erkennen und versuchen ihnen zu folgen.

Unterscheidet sich das Objekt nicht klar vom Hintergrund, ist die Segmentierung keine einfache Aufgabe [10]. Mit Hilfe verschiedener Optimierungsstrategien kann versucht werden eine minimale Abweichung zwischen den Bilddaten und einem gegebenem Modell, das mit Hilfe von Hintergrundwissen über das zu segmentierende Objekt erzeugt wird, zu erreichen. Bei Aufnahmen von modernen Münzen kann man beispielsweise im Regelfall davon ausgehen, dass diese die Form eines Kreises beschreiben. Dieses Wissen wird in [18] erfolgreich zur Segmentierung von modernen Münzen unter Verwendung der Hough-Transformation benützt. Die äußere Form von historischen Münzen bildet allerdings keinen perfekten Kreis und bei der Segmentierung dieser kann daher nicht uneingeschränkt davon ausgegangen werden. Durch eine Reduktion einer historischen Münze auf einen Kreis würde außerdem die wesentliche Eigenschaft der einzigartigen Form, die hier zur Identifikation verwendet werden soll, verloren gehen.

2.1.1 Bildaufnahme

Die Zuverlässigkeit der Segmentierung ist stark von der Qualität der Münzaufnahmen abhängig. Besonders die Vermeidung von Schattenwürfen und die Verwendung eines einheitlichen Hintergrunds mit hohem Kontrast zur Münze haben sich als vorteilhaft erwiesen [26]. Kampel und Zambanini [11] beschreiben wie Münzaufnahmen durchgeführt werden müssen, um diese positiven Eigenschaften zu erreichen. So sollte die Münze beispielsweise auf einer abgehobenen Glasplatte platziert werden, um die Auswirkungen des Schattens zu minimieren. Auch auf sinnvolle Kameraparameter, die Beleuchtung und geeignete Bildformate zur Speicherung wird eingegangen.

2.1.2 Globales Thresholding

Ein einfaches pixelbasiertes Segmentierungsverfahren ist globales Thresholding. Diese Methode kann angewendet werden, wenn sich die Grauwerte der Münze stark von denen des Hintergrundes abheben, d.h. ein starker Kontrast besteht. Ein Schwellwert (engl.: *threshold*) $t \in [1; 255]$ wird gewählt und zur Erzeugung des Binärbildes werden alle Pixel mit einem Grauwert größer-gleich t als zum Objekt und alle anderen zum Hintergrund gehörig gezählt, oder umgekehrt. Das Binärbild B ergibt sich also durch

$$B(i, j) = \begin{cases} 1 & \text{wenn } I(i, j) \geq t \\ 0 & \text{wenn } I(i, j) < t \end{cases} \quad \forall i = 1, \dots, n; j = 1, \dots, m, \quad (2)$$

wobei I für das ursprüngliche Grauwert-Bild mit der Höhe n und Breite m steht. Diese Funktion hat nur das gewünschte Ergebnis (d.h. die Pixel des Objekts haben den Wert 1, alle anderen den Wert 0), wenn die Münze heller als der Hintergrund ist, andernfalls muss das resultierende Binärbild invertiert werden.

Kritisch ist die Wahl des Schwellwerts t . [19] beinhaltet eine Übersicht unterschiedlicher Methoden des Thresholding und der Wahl eines optimalen Schwellwerts. Experimente haben gezeigt, dass aufgrund von möglichen ungleichmäßigen Beleuchtungsverhältnissen, inhomogenem Hintergrund und geringem Kontrast das Ergebnis von globalem Thresholding aber, selbst bei optimaler Wahl von t , im Allgemeinen für Münzaufnahmen nicht zufriedenstellend ist.

2.1.3 Adaptives Thresholding

Statt einem globalen Schwellwert für das gesamte Bild, wird beim adaptiven Thresholding der Schwellwert für jedes Pixel oder Bildbereiche dynamisch verändert. In [25] wird eine von Yanowitz und Bruckstein [24] vorgeschlagene Methode für die Segmentierung von antiken Münzen angewandt, der die Idee zugrunde liegt, dass durch Variation des Schwellwerts unregelmäßige Hintergründe ausgeglichen werden können. Es wird eine sogenannte Schwellwertfläche P (engl.: *threshold surface*), d.h. eine Matrix die für jedes Pixel $I(i, j)$ des Bildes einen eigenen Schwellwert $P(i, j)$ beinhaltet, berechnet. Die Schwellwertfläche wird zuerst mit den Bildpunkten gefüllt, die durch Kantendetektion über ein Gradient-Bild gesucht wurden. Alle weiteren Pixel der Threshold-Oberfläche werden zwischen den vorhandenen durch iteratives Lösen der Laplace Gleichung (3) mit Hilfe eines Algorithmus von Southwell [21] interpoliert.

$$\frac{\partial^2 P(i, j)}{\partial i^2} + \frac{\partial^2 P(i, j)}{\partial j^2} = 0 \quad (3)$$

Das segmentierte Bild B ergibt sich durch Anwendung jedes einzelnen der Schwellwerte $P(i, j)$ auf die zugehörigen Bildpunkte $I(i, j)$ des Grauwert-Bildes I mit der Höhe n und

Breite m , entsprechend Gleichung 4.

$$B(i, j) = \begin{cases} 1 & \text{wenn } I(i, j) \geq P(i, j) \\ 0 & \text{wenn } I(i, j) < P(i, j) \end{cases} \quad \forall i = 1, \dots, n; j = 1, \dots, m \quad (4)$$

2.1.4 Segmentierung basierend auf der lokalen Grauwert-Spannweite

Zambanini und Kampel [26] präsentieren ein neues Segmentierungsverfahren speziell für antike Münzen, basierend auf lokaler Entropie und der lokalen Grauwert-Spannweite (engl.: *gray value range*). Durch Evaluierung mit einem Test-Set, das sehr unterschiedliche Aufnahmen von Münzen enthält, wird gezeigt, dass diese Methode eine wesentlich höhere Zuverlässigkeit als adaptives Thresholding aufweist.

Daher wird in dieser Arbeit eine ähnliche Methode angewandt. Die Grundidee und Voraussetzung ist, dass der Hintergrund homogen ist und daher auf der Münzoberfläche und insbesondere am Rand der Münze die Grauwert-Spannweite höher ist. Für den vorliegenden Datensatz an Münzbildern (siehe Abschnitt 4.2) konnte experimentell gezeigt werden, dass der Einfluss von lokaler Entropie nicht signifikant für das Endergebnis ist.

Die Grauwert-Spannweite ist definiert als die Differenz zwischen dem maximalen und minimalem Grauwert in einer Umgebung um ein Pixel. Hier wird als Umgebung ein zirkuläres Strukturelement K mit einem Radius von 3 Pixeln verwendet. Die Berechnung der Grauwert-Spannweite kann effizient als Differenz zwischen der morphologischen Grauwert-Dilatation und Grauwert-Erosion des ursprünglichen Bildes I jeweils mit dem Strukturelement K implementiert werden [20].

$$R = (I \oplus K) - (I \ominus K) \quad (5)$$

Gleichung 5, in der \oplus für Grauwert-Dilatation und \ominus für Grauwert-Erosion eines Bildes mit einem Strukturelement steht, definiert diese Operation, die auch als morphologischer Gradient bezeichnet wird. Ein Pixel $R(i, j)$ des resultierenden Bildes R entspricht der Grauwert-Spannweite in einer Umgebung K eines Pixels $I(i, j)$ des Grauwert-Bildes I .

Das resultierende Bild R wird, entsprechend Gleichung 2, mit einem globalen Schwellwert in zwei Klassen unterteilt um ein Binärbild zu erzeugen. Die Wahl des Schwellwerts t ist wiederum kritisch für die Qualität der Segmentierung. Experimente haben gezeigt, dass die Ergebnisse mit einem manuell gewählten globalen Schwellwerts nicht ausreichend sind. In [26] werden daher mehrere unterschiedliche globale Schwellwerte angewendet und ein Maß für die Qualität der Segmentierung etabliert. Auch wenn die Konturen von historischen Münzen keinen perfekten Kreis beschreiben, kann man davon ausgehen, dass sie nahe an einem Kreis liegen. Ein Kreis ist die geometrische Figur in der Ebene, die für einen gegebenen Flächeninhalt den geringsten Umfang aufweist. Dieses Wissen soll dazu verwendet werden um den optimalen Schwellwert t_{opt} zu finden.

Um die Form des segmentierten Objekts zu untersuchen, muss dieses zuerst im Binärbild gefunden werden. Homogene Bereiche auf der Münze und inhomogene Bereiche im Hintergrund können zu Löchern, d.h. falsch markierten Bereichen, führen, die im nächsten Schritt möglichst entfernt werden sollen. Die Löcher auf der Münze werden mit einer Floodfill-Operation, die effizient rekursiv oder durch Verwendung eines Stacks implementiert werden kann [10], geschlossen: Alle Pixel, die von einem Pixel in der Ecke des Bildes, das somit sicher nicht zur Münze gehört, erreichbar sind (d.h. über eine Nachbarschaftsrelation verbunden sind und die gleiche Farbe haben), gehören zum Hintergrund. Alle anderen sind somit Teil der Münze (in der nun keine Löcher mehr möglich sind) oder Teil von Löchern im Hintergrund. Der größte zusammenhängende Bereich, der nicht zum Hintergrund gehört, stellt am wahrscheinlichsten die Münze dar. Dieser Bereich kann durch eine Suche nach Zusammenhangskomponenten [20] und Ermittlung des Flächeninhalts für jede Komponente selektiert werden und wird in einem Binärbild

ohne Löcher abgespeichert. Zusätzlich werden auch der Flächeninhalt und Umfang der extrahierten Münze gespeichert, da diese im nächsten Schritt zur Ermittlung des optimalen Schwellwerts benötigt werden.

Der Vorgang wird für verschiedene Schwellwerte durchgeführt und schlussendlich wird die Segmentierung akzeptiert, für die die extrahierte Münzform eine möglichst große *Kompaktheit* [17] aufweist. Die Kompaktheit c einer Form ist definiert als

$$c = \frac{4\pi A}{U^2}, \quad (6)$$

wobei A für den Flächeninhalt und U den Umfang der Form steht. Für einen Kreis nimmt c den Wert 1 an, für alle anderen Formen gilt $c < 1$. Je größer die Kompaktheit der extrahierten Münzform ist, desto ähnlicher ist sie einem Kreis und umso wahrscheinlicher ist der Schwellwert für die Segmentierung gut gewählt.

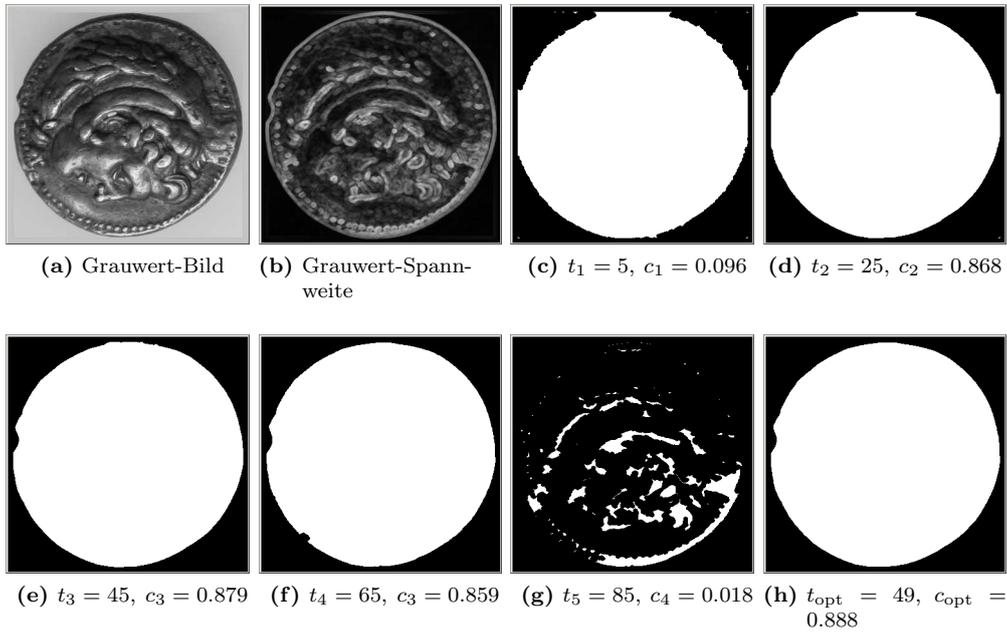


Abbildung 2: Segmentierung durch Thresholding des Grauwert-Spannweiten-Bildes mit unterschiedlichen Schwellwerten

In [26] wird diese Methode mit sieben fixen empirisch ermittelten Schwellwerten, mit konstanter Differenz zwischen je zwei Schwellwerten, durchgeführt. Tests, die mit der hier verwendeten Münzdatenbank (siehe Abschnitt 4.2) vorgenommen wurden, haben allerdings gezeigt, dass das Ergebnis der Segmentierung noch verbessert werden kann, wenn nach der Bestimmung eines groben Schwellwerts t für den c ein Maximum annimmt, nicht abgebrochen wird, sondern in einer lokalen Umgebung von t weitergesucht wird, um den optimalen Schwellwert t_{opt} zu finden, für den eine maximale Kompaktheit erreicht wird. Abbildung 2 zeigt das ursprüngliche Grauwert-Bild einer Münze (a), das resultierende Bild nach Berechnung der Grauwert-Spannweite (b) und die Ergebnisse der Segmentierung durch fünf grobe Schwellwerte mit einer Differenz von 20 sowie die jeweils dazugehörige Kompaktheit der Form (c)-(g). Die Kompaktheit für $t_3 = 45$ ist am höchsten und daher wurde in einer Umgebung von t_3 nach dem optimalen Threshold ($t_{\text{opt}} = 49$) mit maximaler Kompaktheit ($c_{\text{opt}} = 0.888$) gesucht und so das beste Segmentierungsergebnis (h) dieser Methode für diese Münze erreicht. Es ist ersichtlich, dass

die Kompaktheit bei Ausdehnung der Münzform und Zerklüftung der Kontur, durch zu niedrige oder zu hohe Schwellwerte, niedriger wird.

2.1.5 Experimente & Diskussion

Die auf der lokalen Grauwert-Spannweite basierende Segmentierung wurde auf Bilder des verwendeten Münz-Datensets (siehe Abschnitt 4.2) angewendet und das Ergebnis manuell kontrolliert. Dabei wurden keine groben Segmentierungsfehler gefunden. Helle Reflektionen am Rand der Münze und Verschmutzungen im Hintergrund, die nahe bei der Münze liegen, können allerdings zu kleinen Einbuchtungen bzw. Ausbuchtungen der entstehenden Münzform führen. Diese betreffen aber jeweils nur einen geringen Anteil der Gesamtkontur einer Münze. Ein typisches Beispiel für eine fehlerhafte Einbuchtung in dem segmentierten Objekt zeigt Abbildung 2f. Selbst bei optimaler Wahl des Schwellwerts kann nicht bei jeder Münze eine so erfolgreiche Segmentierung wie in Abbildung 2h erreicht werden.

Die Segmentierung ist für die Zuverlässigkeit der Gesamtmethode ausschlaggebend, da alle weiteren Schritte davon abhängig sind. Die Bilder der verwendeten Münz-Datensets wurden unter günstigen Bedingungen aufgenommen (siehe Abschnitt 2.1.1) und enthalten keine anderen Gegenstände. Die Segmentierung von Aufnahmen, die heterogenen Hintergrund, geringen Kontrast, Schattenwürfe oder zusätzliche Objekte beinhalten, ist wesentlich schwieriger und wird in dieser Arbeit nicht weiter untersucht.

2.2 Boundary Extraction

Ausgehend vom Binärbild soll die Kontur der Münze extrahiert (engl.: *boundary extraction*) und in eine zum Vergleich geeignete Darstellung konvertiert werden. Voraussetzung ist, dass das Binärbild nur eine zusammenhängende Münzform enthält. Die in Abschnitt 2.1.4 beschriebene Segmentierung stellt das sicher.

Durch Abfahren der Pixel (engl.: *boundary tracing*) im Binärbild, die einen Übergang zwischen 0 und 1 darstellen, also den Randpixeln des Objekts, erhält man eine Folge $\mathbf{P} = \langle P_1, P_2, \dots, P_n \rangle$ der Randpunkte der Münze, die durch zweidimensionale kartesische Koordinaten beschrieben werden. Zu beachten ist, dass die Folge im oder gegen den Uhrzeigersinn sortiert ist und n von der Auflösung des Bildes abhängig ist.

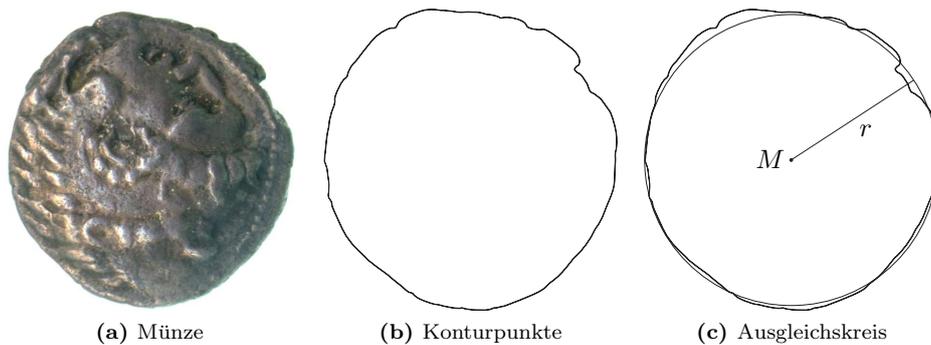


Abbildung 3: Schritte der Boundary Extraction

Für das segmentierte Objekt kann nun ein sogenannter Ausgleichskreis bestimmt werden, dessen Mittelpunkt M der Schwerpunkt der Form ist (Gleichung 7). Der Radius r berechnet sich als mittlerer Abstand zwischen dem Mittelpunkt M und allen Randpunkten aus \mathbf{P} (Gleichung 8). In Gleichung 8 und auch in folgenden Gleichungen

steht der Operator $\|\cdot\|$ für die L_2 -Norm.

$$M = \frac{1}{n} \sum_{i=1}^n P_i \quad (7)$$

$$r = \frac{1}{n} \sum_{i=1}^n \|P_i - M\| \quad (8)$$

Abbildung 3 zeigt eine Münze (a), die geometrische Darstellung der Konturpunkte (b) und den für die Konturpunkte bestimmten Ausgleichskreis (c) mit Mittelpunkt M und Radius r .

Die Form soll als eindimensionales diskretes Kontursignal $S = \langle s_1, s_2, \dots, s_n \rangle$ repräsentiert werden, dessen Werte sich durch

$$s_i = \frac{\|P_i - M\| - r}{r} \quad \forall i = 1, \dots, n \quad (9)$$

berechnen. Der Wert s_i gibt also die minimale relative Distanz zwischen dem Randpunkt P_i und dem Bogen des Ausgleichskreises an und ist positiv oder negativ, je nachdem ob der Punkt außerhalb oder innerhalb des Ausgleichskreises liegt. Um Skalierungsinvarianz zu erreichen werden die Werte durch Division durch den Radius normalisiert. Es sollte beachtet werden, dass die Länge n des Kontursignals der Anzahl der Randpunkte entspricht und damit auch von der Auflösung des Bildes abhängt.

2.2.1 Sampling

Damit zwei Kontursignale, die die Konturen von zwei Münzen repräsentieren, skalierungsinvariant und effizient miteinander verglichen werden können, sollen diese auf eine fixe Länge l gekürzt werden. Die Kürzung eines Kontursignals S der Länge n heißt Sampling und wird durch Entnahme von $l < n$ repräsentativen Stichproben aus S erreicht.

Das ursprüngliche Signal wird zuerst mit einem Tiefpassfilters bandbeschränkt um Aliasing-Effekte zu vermeiden. Dazu wird das Signal durch Faltung mit einem Mittelwertfilter der Länge $2\frac{n}{l}$, die sich aus dem Nyquist-Shannon Sampling-Theorem [8] ergibt, geglättet. Die Implementierung kann effizient durch Multiplikation im Frequenzraum erfolgen [5].

Äquidistantes Sampling

Bei äquidistantem Sampling werden die Stichproben unter Verwendung eines konstanten Sampling-Intervalls $d = \frac{n}{l}$ aus dem Kontursignal entnommen.

Um das auch für ein nicht ganzzahliges d zu ermöglichen wird das Kontursignal in eine kontinuierliche Funktion mit einem Definitionsbereich $D = [1, n]$ umgewandelt. Der Funktionswert $f(x)$ ergibt sich durch Anwendung der Formel für lineare Interpolation (Gleichung 10), in der s_i für den i . Wert des Kontursignals S der Länge n steht, das zuvor geglättet wurden.

$$f(x) = s_{\lfloor x \rfloor} + (s_{\lceil x \rceil} - s_{\lfloor x \rfloor})(x - \lfloor x \rfloor) \quad \forall x \in \mathbb{R}, 1 \leq x \leq n \quad (10)$$

Ein Wert s_i des gekürzten Kontursignals $S_{ED} = \langle s_1, s_2, \dots, s_l \rangle$ ergibt sich durch

$$s_i = f(1 + (i - 1) \cdot d) \quad \forall i = 1, \dots, l, \quad (11)$$

wenn f die für das Kontursignal S durch lineare Interpolation erzeugte kontinuierliche Funktion ist und $d = \frac{n}{l}$ gilt.

Äquiangulares Sampling

Anders als beim äquidistanten Sampling werden bei äquiangularem Sampling die l Stichproben unter Verwendung eines konstanten Winkels $\varphi = \frac{360^\circ}{l}$ bestimmt. Ausgehend vom Mittelpunkt M des Ausgleichskreises werden unter wiederholter Drehung um den Winkel φ die l Strahlen aufgestellt (d.h. die Folge $\varphi_i = (i - 1) \cdot \varphi \quad \forall i = 1, \dots, l$ von Winkeln wird linksdrehend oder rechtsdrehend abgeschlagen), deren Schnittpunkte mit der Münzkontur die Kontursignalwerte bestimmen, die als Stichproben für das gekürzte Signal herangezogen werden.

Für die Implementierung muss erneut die Folge $\mathbf{P} = \langle P_1, P_2, \dots, P_n \rangle$ der Randpunkte der Münzform betrachtet werden. Für jeden Randpunkt wird eine Darstellung als Polarkoordinaten $P_i = (r_i, \theta_i)$ mit dem Mittelpunkt M des Ausgleichskreises als Ursprung berechnet. Die Radialkoordinate r_i , die dem euklidischen Abstand zwischen dem Punkt P_i und M entspricht (d.h. $r_i = \|P_i - M\|$), wird hier nicht benötigt. Die Winkelkoordinate θ_i wird durch $\theta_i = \text{atan2}(y_i, x_i)$ berechnet, wobei (x_i, y_i) für die kartesischen Koordinaten des Vektors $\overrightarrow{MP_i}$ und $\text{atan2}(\cdot)$ für eine bivariate Arkustangens-Funktion (Gleichung 12), die Werte aus dem Intervall $]0^\circ, 360^\circ[$ liefert, stehen.

$$\text{atan2}(y, x) := \begin{cases} \arctan \frac{y}{x} & \text{wenn } x > 0 \\ \arctan \frac{y}{x} + 180^\circ & \text{wenn } x < 0, y \geq 0 \\ \arctan \frac{y}{x} - 180^\circ & \text{wenn } x < 0, y < 0 \\ +90^\circ & \text{wenn } x = 0, y > 0 \\ -90^\circ & \text{wenn } x = 0, y < 0 \end{cases} \quad (12)$$

Für jeden Winkel φ_i der Winkelreihe, der ausgehend vom Mittelpunkt des Ausgleichskreises abgeschlagen werden soll, wird nun der Münzkontur-Punkt P_k gesucht, dessen Winkelkoordinate θ_k dem Winkel φ_i entspricht. Der zugehörige Wert s_k des ursprünglichen Kontursignals wird als Stichprobe \bar{s}_i für das gekürzte Kontursignal verwendet. Liegt der gewünschte Winkel φ_i , der abgeschlagen werden soll, zwischen den Winkelkoordinaten von zwei aufeinanderfolgenden Konturpunkten P_k und P_{k+1} (mit $\theta_k < \varphi_i < \theta_{k+1}$) kann wiederum der Wert \bar{s}_i für das gekürzte Kontursignal $S_{EA} = \langle \bar{s}_1, \bar{s}_2, \dots, \bar{s}_l \rangle$ durch lineare Interpolation entsprechend Gleichung 13 berechnet werden, in der s_k für den k . Wert des ursprünglichen Kontursignals S steht.

$$\bar{s}_i = s_k + (s_{k+1} - s_k) \frac{\varphi_i - \theta_k}{\theta_{k+1} - \theta_k} \quad \text{mit } \theta_k < \varphi_i < \theta_{k+1} \quad \forall i = 1, \dots, l \quad (13)$$

Es muss beachtet werden, dass der Übergang zwischen dem Randpunkt mit kleinster Winkelkoordinate und dem Randpunkt mit größter Winkelkoordinate einen Spezialfall darstellt und gesondert behandelt werden muss. Problematisch ist, wenn durch Einbuchtungen in der Kontur die Folge der Winkelkoordinaten der Randpunkte nicht streng monoton wachsend (d.h. $\theta_{i+1} > \theta_i \forall i = 1, \dots, n$) ist. Zu einem Winkel φ_i kann es somit mehrere Stellen der Kontur geben, deren zugehörige Kontursignalwerte als Stichprobe \bar{s}_i des gekürzten Signals verwendet werden könnten. Eine Strategie für die Wahl muss definiert werden. Hier wird das Verfahren durch die Zusicherung ergänzt, dass im Zweifelsfall der Kontursignalwert s_k als Stichprobe verwendet wird, dessen zugehöriger Konturpunkt P_k den kleinsten Index k hat. Es soll angemerkt werden, dass die Information über solche Einbuchtungen im Kontursignal durch äquiangulares Sampling auf jeden Fall verloren geht. Einbuchtungen dieser Art kommen in der verwendeten Testdatenbank von historischen Münzen allerdings nicht vor und es wird daher davon ausgegangen, dass diese Einschränkung für die Anwendung weniger relevant ist.

Abbildung 4 stellt den Unterschied der beiden beschriebenen Sampling-Verfahren geometrisch dar. Jeweils die ersten vier Konturpunkte, deren zugehörige Kontursignalwerte für das gekürzte Signal verwendet werden, sind eingezeichnet und durch Strecken

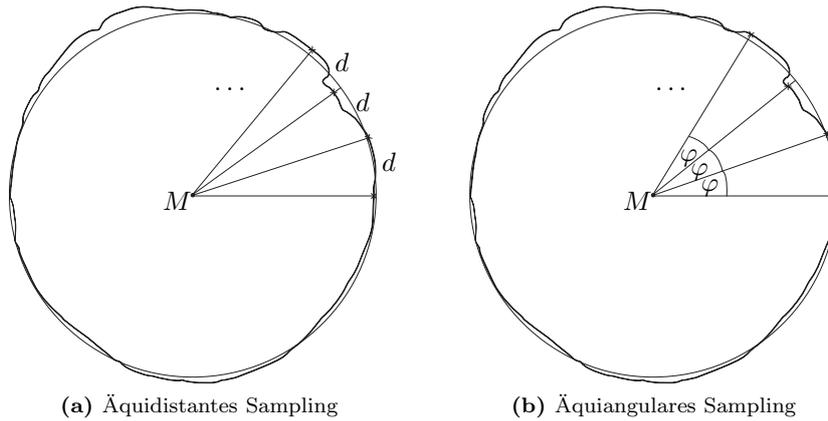


Abbildung 4: Geometrische Darstellung der Sampling-Verfahren

mit dem Mittelpunkt M des Ausgleichskreises verbunden. Bei äquidistantem Sampling (a) entspricht das konstante Sampling-Intervall d den Bogenlängen zwischen je zwei gewählten Konturpunkten auf der Münzkontur, die Winkel zwischen den Strecken sind unterschiedlich. Im Gegensatz dazu sind bei äquiangularem Sampling (b) die Winkel φ zwischen je zwei Verbindungsstrecken vom Mittelpunkt zu benachbarten ausgewählten Konturpunkten konstant und die Bogenlängen variieren.

Die Vermutung, dass äquiangulares Sampling für diese Anwendung besser geeignet ist, wird in Abschnitt 4 experimentell bestätigt und diskutiert.

2.3 Shape Matching

Zwischen zwei (auf die gleiche Länge l gekürzten) Kontursignalen S und T , die zwei aus Bildern extrahierte Münzenkonturen beschreiben, soll nun ein Distanz-Maß $d(S, T)$ etabliert werden, über das dann eine Identifikationsmethode implementiert werden kann.

Das Maß $d(S, T)$ ist ein Skalar, das die Unähnlichkeit (engl.: *dissimilarity*) von zwei Münzenkonturen invariant unter Translation, Skalierung, Rotation und Spiegelung (Flips) angibt. Die ersten beiden dieser Invarianzen werden bereits über die zuvor definierte Repräsentation einer Münzkontur als diskretes Kontursignal erreicht.

Translationsinvarianz

Translation bezeichnet eine Verschiebung der Münze auf dem Bild gegenüber der Münze, mit der verglichen werden soll. Die Kontursignale sind invariant unter Translationen, da die Münzform durch die Segmentierung bestimmt wird und von dem ermittelten Schwerpunkt aus gemessen wird.

Skalierungsinvarianz

Skalierung ist ein Größenunterschied zwischen den Münzen, der durch unterschiedliche Bildauflösungen entstehen kann. Durch die zuvor beschriebene Normalisierung der einzelnen Werte und das Sampling auf eine fixe Länge l sind die Kontursignale invariant unter Skalierungen.

Rotationsinvarianz

Bei einer Rotation wurden die Münzen unterschiedlich gedreht aufgenommen. Dadurch ergibt sich auch eine zirkuläre Verschiebung (engl.: *circular shift*) der beiden Kontursignale, was beim Vergleich beachtet werden muss.

Flipinvarianz

Bilder von Münzen können entweder die Vorder- oder die Rückseite der Münze zeigen. Ein Umdrehen der Münze wird als Flip bezeichnet und bewirkt eine Spiegelung des zur Münze gehörigen Kontursignals. Auch das muss beim Vergleich beachtet werden, um eine Identifikation einer Münze zu ermöglichen, unabhängig davon welche Seite das verwendete Vergleichsbild zeigt.

2.3.1 Vergleich im Ortsbereich

Ein Maß für die Unähnlichkeit eines Signals $S = \langle s(1), s(2), \dots, s(l) \rangle$ mit dem Vergleichssignal $T = \langle t(1), t(2), \dots, t(l) \rangle$ im Ortsbereich (engl.: *space domain*) kann über die mittlere quadratische Differenz (MQD) (engl.: *mean squared error, MSE*) berechnet werden. Um Invarianz unter Rotation zu erreichen, wird die minimale MQD (MMQD) des Signals mit allen möglichen zirkulären Verschiebungen des Vergleichssignals berechnet (Gleichung 14).

$$\text{MMQD}(S, T) = \min_{u=0}^{l-1} \frac{1}{l} \sum_{i=1}^l [s(i) - t((i+u-1 \bmod l) + 1)]^2 \quad (14)$$

Flipinvarianz kann erreicht werden, wenn die MMQD zusätzlich für das Signal S mit dem gespiegelten Vergleichssignal $T' = \langle t'(1) = t(l), t'(2) = t(l-1), \dots, t'(l) = t(1) \rangle$ berechnet wird.

$$d_{\text{MQD}}(S, T) = \min(\text{MMQD}(S, T), \text{MMQD}(S, T')) \quad (15)$$

Die unter Rotation und Flips invariante Unähnlichkeit d_{MQD} der Signale S und T ergibt sich als Minimum beider Berechnungen (Gleichung 15).

2.3.2 Vergleich im Frequenzbereich

Eine weitere Möglichkeit stellt der Vergleich von Signalen im Frequenzbereich (engl.: *frequency domain*) dar. Dazu wird das Konturspektrum betrachtet, das durch Fourier-Deskriptoren, die sich aus der diskreten Fourier-Transformation (DFT) der Signale ergeben, dargestellt wird. Die komplexwertigen Koeffizienten der diskreten Fourier-Transformierten $\mathcal{F}(S) = \langle \hat{s}(1), \hat{s}(2), \dots, \hat{s}(l) \rangle$ eines Signals $S = \langle s(1), s(2), \dots, s(l) \rangle$ werden über Gleichung 16 berechnet.

$$\hat{s}(k) = \sum_{j=1}^l \exp\left(-2\pi i \cdot \frac{(j-1)k}{l+1}\right) \cdot s(j) \quad \forall k = 1, \dots, l \quad (16)$$

Hinweis: In der Praxis kann die DFT effizient mit dem FFT-Algorithmus (engl.: *fast fourier transform*) berechnet werden [4].

Die Koeffizienten der Fourier-Transformierten repräsentieren die Phase und Amplitude der einzelnen Sinuskurven aus denen sich das diskrete Signal zusammensetzt, absteigend sortiert nach Frequenz. Durch Bildung des Absolutbetrags jedes komplexen Koeffizienten entsteht eine Folge $D(S) = \langle |\hat{s}(1)|, |\hat{s}(2)|, \dots, |\hat{s}(l)| \rangle$ von Fourier-Deskriptoren des Signals S , die nur die Amplituden der Sinuskurven beschreibt und daher unabhängig von Rotation und Flips ist.

$$d_{\text{FD}}(S, T) = \frac{1}{l-v-w} \sum_{i=1+v}^{l-w} |d_S(i) - d_T(i)| \quad (17)$$

Gleichung 17 definiert ein Maß $d_{\text{FD}}(S, T)$ für die Unähnlichkeit von zwei Signalen S und T , das effizient im Frequenzbereich durch Bildung der mittleren absoluten Differenz

(MAD) zwischen den zugehörigen Fourier-Deskriptoren $D(S) = \langle d_S(1), d_S(2), \dots, d_S(l) \rangle$ und $D(T) = \langle d_T(1), d_T(2), \dots, d_T(l) \rangle$ berechnet wird. Die Konstanten $v \geq 0$ und $w \geq 0$ sind kleine Zahlen, die verwendet werden um sehr niederfrequente und hochfrequente Fourier-Deskriptoren beim Vergleich nicht zu berücksichtigen. Dadurch wird der Einfluss von Fehlern beim Bestimmen des Ausgleichskreises und Quantisierungsrauschen (d.h. Störungen, die sich durch die digitale Darstellung und Rundungsfehler ergeben) vermieden. Sinnvolle Werte für v und w werden in Abschnitt 4 experimentell ermittelt.

2.3.3 Kombiniertes Vergleich im Ortsbereich und Frequenzbereich

Die zuvor definierten Unähnlichkeits-Maße $d_{\text{MQD}}(S, T)$ und $d_{\text{FD}}(S, T)$ zwischen zwei Kontursignalen S und T sind im Allgemeinen nicht äquivalent und bieten unterschiedliche Vorteile und Nachteile. Beim Vergleich im Frequenzraum ist es, wie zuvor beschrieben, auf einfache Weise möglich den Einfluss von Fehlern zu limitieren. Andererseits ist der Vergleich der Fourier-Deskriptoren allein eine relativ eingeschränkte Sicht auf die Kontur. Das heißt Konturen unterschiedlicher Münzen, die insgesamt (nach menschlichem Empfinden) sehr unterschiedlich sind, können trotzdem einen relativ niedrigen Unähnlichkeitswert liefern, da die Amplituden der einzelnen Sinuskurven ähnlich sein könnten, z.B. wenn sich die Münzkonturen abschnittsweise ähneln. Der Vergleich im Ortsbereich ist eine ganzheitliche Sicht und bietet außerdem die Möglichkeit, wenn zwei Signaturprofile derselben Münze verglichen werden, die relative Rotation der Münze zwischen den Bildern herauszufinden, da Vergleiche in allen zirkulären Verschiebungen durchgeführt werden (d.h. über u aus Gleichung 14). Dadurch ergibt sich zwar kein direkter Vorteil für die Identifikation, die Zusatzinformation kann aber nützlich sein (siehe Abschnitt 6).

Es wird daher eine lineare Kombination der beiden Maße berechnet, um ein neues endgültiges Maß $d(S, T)$ für die Unähnlichkeit der Kontursignale S und T zu erhalten, das insgesamt eine zuverlässigere Identifikation ermöglichen soll:

$$d(S, T) = \lambda \cdot d_{\text{MQD}}(S, T) + (1 - \lambda) \cdot d_{\text{FD}}(S, T) \quad (18)$$

Ein optimaler Gewichtungsfaktor $\lambda \in [0, 1]$ wird in Abschnitt 4 empirisch bestimmt.

2.4 Identifikation

Unter Verwendung des zuvor beschriebenen Unähnlichkeitsmaßes ist es möglich ein Identifikationsverfahren zu implementieren.

Es wird davon ausgegangen, dass eine unbekannte Münzaufnahme mit Hilfe von mehreren bekannten Referenz-Münzaufnahmen (in der Folge als "Datenbank" bezeichnet) identifiziert werden soll. Die unbekannte Münzaufnahme wird dazu mit jeder Aufnahme aus der Datenbank durch Berechnung eines Unähnlichkeitswertes verglichen. Zur Bewertung kann das *nearest neighbor*-Verfahren verwendet werden. Das heißt die Münze der Datenbank, die die geringste Unähnlichkeit aufweist, bestimmt die Identität der unbekanntes Münze. Voraussetzung für eine korrekte Identifikation ist natürlich, dass die Datenbank zumindest eine Aufnahme der zu identifizierenden Münze enthält. Bei den Experimenten in Abschnitt 4 wird nur dieses Identifikationsverfahren verwendet. Beinhaltet eine Datenbank mehrere Aufnahmen jeder Münze, so könnte auch das *k-nearest neighbor*-Verfahren mit Mehrheitswahl (engl.: *majority voting*) verwendet werden. Das heißt die ersten k Einträge der Liste werden betrachtet und die darin enthaltene Mehrheit bestimmt die Identität einer Münze. Ist keine eindeutige Mehrheit enthalten, so muss ein anderes Verhalten definiert werden. Es könnte dann beispielsweise auf das ursprüngliche *nearest neighbor*-Verfahren zurückgegriffen werden.

2.5 Verifikation

Für eine korrekte Identifikation einer unbekanntes Münze auf einem Bild wird vorausgesetzt, dass sich in der Datenbank, die zum Vergleich verwendet wird, zumindest eine Aufnahme derselben Münze befindet. Ist die Münze in der Datenbank nicht vorhanden, so wäre es wünschenswert, wenn sie zurückgewiesen, d.h. als *unbekannt* klassifiziert wird. Durch die zuvor definierte Identifikation mit dem *k-nearest neighbor*-Verfahren würde eine Münze, die sich nicht in der Datenbank befindet, fälschlicherweise als die Münze der Datenbank mit der ähnlichsten Kontur identifiziert werden. Ziel der Verifikation ist diesen Fall zu erkennen und die Münze als nicht identifizierbar zurückzuweisen. Zusätzlich kann durch die Verifikation die Falschidentifikationsrate (d.h. die Wahrscheinlichkeit, dass eine Münze falsch identifiziert wird, obwohl sie sich in der Datenbank befindet) verringert werden. Es sollte allerdings beachtet werden, dass die Korrektidentifikationsrate (d.h. die Wahrscheinlichkeit, dass eine Münze, die in der Datenbank vorhanden ist, korrekt identifiziert wird) nicht direkt erhöht, sondern sogar reduziert werden kann, da eine falsche Identifikation durch die Verifikation hier zwar (möglicherweise) erkannt, allerdings niemals ausgebessert, sondern stattdessen zurückgewiesen wird.

Eine Möglichkeit zur Realisierung der Verifikation wäre die Zurückweisung aller Münzen, deren Unähnlichkeit zu ihrem gefundenen *nearest neighbor* einen festgesetzten Schwellwert übersteigt. Experimente haben jedoch gezeigt, dass das zuvor definierte Unähnlichkeitsmaß für diese Anwendung ungeeignet ist. Der lokale Vergleich der Unähnlichkeiten zwischen einem Münzbild und allen Münzbildern aus der Datenbank ist möglich und zur Identifikation über den *nearest neighbor* geeignet. Die Anwendung eines globalen Schwellwerts ist aber nicht sinnvoll, da es durchaus möglich ist, dass die Unähnlichkeit zwischen zwei Bildern, die dieselbe Münze zeigen, aufgrund von Quantisierungsfehlern der digitalen Darstellung und Segmentierungsfehlern, höher ist, als die zwischen einer anderen Münze und einer dritten.

Eine bessere Realisierung der Verifikation ist möglich, wenn Aufnahmen von beiden Seiten der zu identifizierenden Münze zur Verfügung stehen. Es kann dann für beide Bilder jeweils der *nearest neighbor* in der Datenbank gesucht werden. Wenn die Identifikation für beide gleich ist, dann wird diese akzeptiert, andernfalls zurückgewiesen. Das Verifikationsverfahren basiert auf der Annahme, dass für eine Münze, die nicht in der Datenbank vorhanden ist, die Wahrscheinlichkeit gering ist, dass sie zweimal auf die gleiche Weise falsch identifiziert wird. Es muss allerdings beachtet werden, dass diese Wahrscheinlichkeit immer von der Größe der Datenbank abhängig ist.

3 Implementierung

In Abschnitt 2 wurden Methoden beschrieben, um aus einem Bild die Münzkontur zu extrahieren und in ein Kontursignal umzuwandeln. Außerdem wurde gezeigt, wie die Unähnlichkeit zwischen zwei Kontursignalen berechnet werden kann und damit die Identifikation unbekannter Münzaufnahmen möglich ist. In diesem Abschnitt soll eine konkrete Implementierung als Software-Bibliothek, die zur Evaluierung und Demonstration der Methoden erstellt wurde, vorgestellt werden.

3.1 Software-Bibliothek

Alle in Abschnitt 2 beschriebenen Methoden wurden in einer plattformunabhängigen Software-Bibliothek in der Programmiersprache C++ implementiert. Dabei wurde auf die quelloffene Programmbibliothek OpenCV⁴ von Intel⁵ aufgebaut.

⁴<http://opencvlibrary.sourceforge.net/>

⁵Intel ist eine Marke der Intel Corporation in den USA und anderen Ländern.

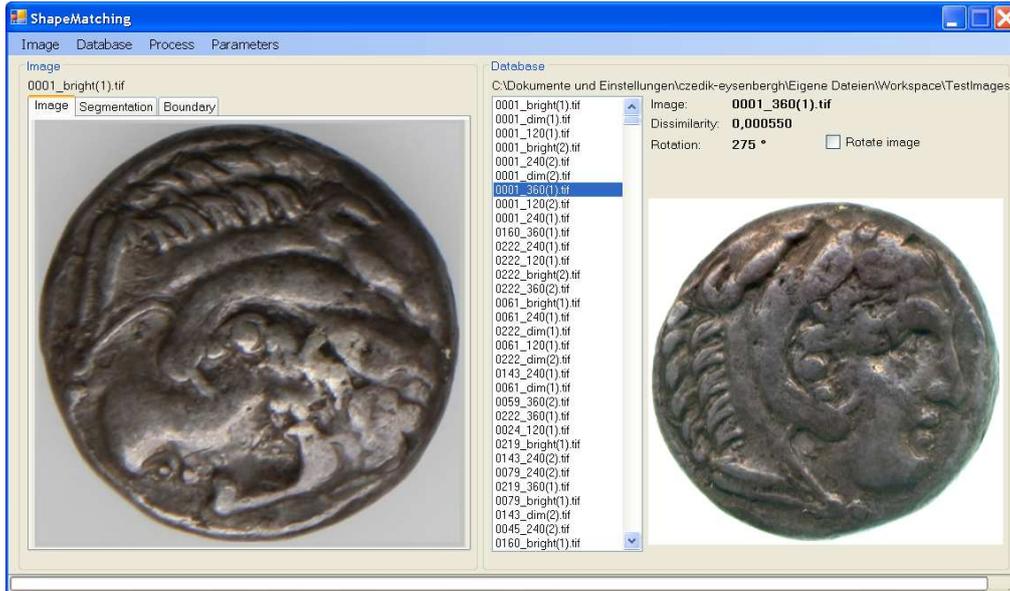


Abbildung 5: Grafische Benutzeroberfläche

Die erzeugte Software-Bibliothek stellt insbesondere einen Aufruf zur Verfügung, um eine Abbildung einer unbekanntes Münze unter Verwendung einer Datenbank von Aufnahmen bekannter Münzen zu identifizieren. Dazu wird die Münzkontur aus der unbekanntes Abbildung extrahiert, mit allen Münzkonturen aus der Datenbank verglichen und eine nach dem Unähnlichkeitswert aufsteigend sortierte Liste zurückgegeben. Diese kann unterschiedlich interpretiert werden, beispielsweise durch das zuvor beschriebene *nearest neighbor*-Verfahren. Damit der Vergleich mit der gesamten Datenbank möglichst schnell abläuft, kann durch einen weiteren Aufruf für alle Münzaufnahmen aus der Datenbank die Kontur bereits extrahiert und als Kontursignal abgespeichert werden. Die erste Methode kann dann auf diese Daten zugreifen. Sollen mehrere unbekanntes Aufnahmen mit Hilfe der gleichen Münz-Datenbank identifiziert werden, bringt das eine große Zeitersparnis, da die Segmentierung und Konturextraktion für jede Münze der Münz-Datenbank nur ein Mal durchgeführt werden muss.

3.2 Grafische Benutzeroberfläche

Zusätzlich wurde eine einfache grafische Benutzeroberfläche in der Sprache C# programmiert, um die durch die Bibliothek bereitgestellten Aufrufe einfach zu testen und zu demonstrieren. Abbildung 5 zeigt die Oberfläche, in der eine Aufnahme geladen wurde und in einer Liste die ähnlichsten Münzen aus der Datenbank angezeigt werden.

4 Experimente

Durch Experimente mit unterschiedlichen Datensätzen soll die Korrektheit der verwendeten Implementierung evaluiert werden. Außerdem soll die Zuverlässigkeit der Identifikationsmethode, unter Verwendung unterschiedlicher Werte für die in Abschnitt 2 beschriebenen Parameter, bestimmt werden.

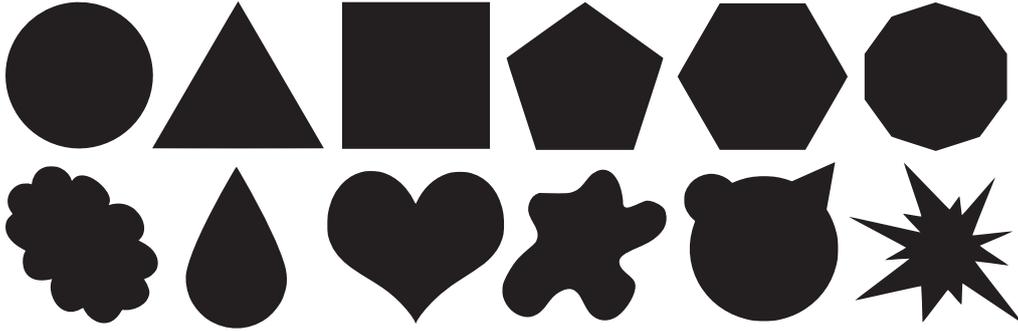


Abbildung 6: Beispiele aus den synthetischen Testdaten

4.1 Synthetische Daten

Zur Simulation und Überprüfung der Korrektheit der Implementierung wurden synthetische Daten erstellt. Diese bestehen aus einem Datensatz von 36 Vielecken und anderen einfachen geometrischen Formen (Beispiele in Abbildung 6). Von jeder Form wurden insgesamt 6 Binärbilder, auf denen die Formen unterschiedlich rotiert, skaliert, verschoben und gespiegelt (um Flips zu simulieren) dargestellt werden, erstellt. Insgesamt stehen also 216 Bilder zur Verfügung und werden in einer Menge \mathcal{V} zusammengefasst.

Es soll nun gezeigt werden, dass die Identifikationsmethode alle Formen (in der Folge als Klassen bezeichnet) unabhängig von der Darstellung unterscheiden kann. Dafür wurde ein Unähnlichkeitswert $d(V, W)$ zwischen je zwei Bildern (d.h. $\forall V, W \in \mathcal{V}$) berechnet.

4.1.1 Resultate

Sowohl für Ermittlung des Unähnlichkeitswerts durch den Vergleich im Ortsbereich als auch durch Vergleich im Frequenzbereich (und damit auch für alle durch Gleichung 18 gebildeten Linearkombinationen mit $\lambda \in [0, 1]$) konnte gezeigt werden, dass für die Unähnlichkeitswerte $d(V, W)$ zwischen Bildern V und W Gleichung 19 gilt, in der $c(X)$ für die Klasse eines Bildes $X \in \mathcal{V}$ steht.

$$d(A, B) < d(A, C) \quad \forall A, B, C \in \mathcal{V} : c(A) = c(B) \wedge c(A) \neq c(C) \quad (19)$$

Das heißt, wird ein Bild A mit einem Bild B verglichen, das die gleiche Form wie A zeigt, so ist der Unähnlichkeitswert immer kleiner, als wenn das Bild A mit einem Bild C , das eine andere Form als A zeigt, verglichen wird.

Aufgrund der Gültigkeit von Gleichung 19 ist es möglich alle verwendeten Formen korrekt mit der *nearest neighbor*-Methode zu identifizieren, sofern die zu identifizierende Form zumindest einmal - in einer möglicherweise unterschiedlichen Darstellung - in der Vergleichsdatenbank vorhanden ist. Das Ziel des Tests mit synthetischen Daten wurde somit erreicht.

4.2 Historische Münzdatenbank

Zum Testen der Methoden an echten Daten wird eine Münz-Datenbank verwendet, die in der Folge als COINS-Datenbank bezeichnet wird. Die COINS-Datenbank wurde durch das Fitzwilliam Museum in Cambridge (UK) zur Verfügung gestellt und beinhaltet 2400 Aufnahmen von 240 verschiedenen historischen Münzen. Beide Seiten jeder Münze wurden 5 mal unter Verwendung unterschiedlicher Verfahren aufgenommen. Je drei Scans in unterschiedlichen Rotationen (mit einem Versatz von 120°) und zwei Fotos, die unter variierenden Lichtbedingungen aufgenommen wurden, stehen zur Verfügung. Die Scans



Abbildung 7: Beispiele aus der COINS-Datenbank

haben eine bessere Auflösung (ungefähr 600×600 Pixel) und wurden bereits händisch - durch Ersetzen des Hintergrundes mit weißer Farbe - vorsegmentiert. Die Fotos haben eine geringere Qualität (ungefähr 300×300 Pixel) und einige davon sind ziemlich unscharf. Der Hintergrundbereich wurde noch nicht entfernt, aber bei der Aufnahme wurde auf die Vermeidung von starken Schattenwürfen geachtet. Wie in Abschnitt 2 beschrieben, werden mit dem auf der lokalen Grauwert-Spannweite basierenden Segmentierungsverfahren sehr gute Ergebnisse erzielt. Jedes Bild ist durch eine Identifikationszahl markiert, die die Identität der darauf abgebildeten Münze eindeutig beschreibt. Diese Datenbank ist zum Testen von Identifikationsverfahren besonders geeignet, da sie nur sehr wenige unterschiedliche Münztypen beinhaltet. Alle Bilder der COINS-Datenbank zeigen Münzen die in der Herrschaft Alexander des Großen (336 bis 323 v.Chr.) oder später nach dem Vorbild alexandrinischer Münzen geprägt wurden. Alle folgen dem gleichen Schema: Die Vorderseite zeigt den Kopf von Herkules in einer Löwenhaut. Einige Numismatiker gehen davon aus, dass Alexander als Herkules repräsentiert wird. Die Rückseite bildet den Göttervater Zeus, sitzend auf einem Thron, ab. Es gibt eine Reihe von kleinen Variationen, die Numismatiker dazu einsetzen um den Stempel und die Zeit der Prägung zu klassifizieren. Zwei unterschiedliche Münzen sollen hier voneinander unterschieden werden können, selbst wenn sie dem gleichen Münztyp angehören. Das ist auch für Experten nicht immer eine triviale Aufgabe.

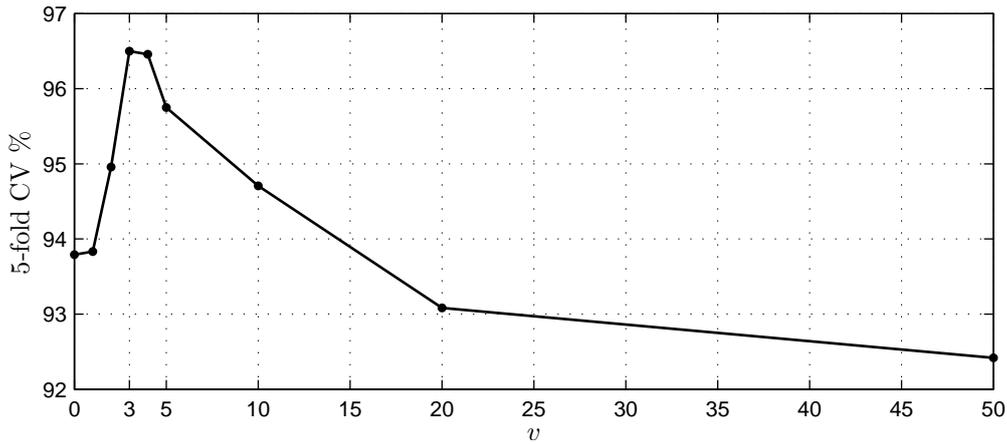


Abbildung 8: 5-fold CV mit unterschiedlichen Werten für Parameter v

v	0	1	2	3	4	5	10	20	50
%	93.79	93.83	94.96	96.5	96.46	95.75	94.71	93.08	92.42

Tabelle 1: 5-fold CV mit unterschiedlichen Werten für Parameter v

4.2.1 Cross-Validation

Die in Abschnitt 2 beschriebenen Methoden sind durch mehrere Parameter, die einen Einfluss auf die Zuverlässigkeit des realisierten Identifikationsverfahrens nehmen, konfigurierbar. Um optimale Werte für die Parameter zu bestimmen, ist es notwendig die Zuverlässigkeit der Identifikation mit unterschiedlichen Parametern zu bewerten. Das Testverfahren *k-fold Cross-Validation* weist eine geringe Varianz auf und ist daher gut geeignet, um aus unterschiedlichen Klassifizierern den besten auszuwählen [12]. Da Identifikation ein Spezialfall der Klassifikation ist, kann dieses Verfahren auch hier angewendet werden.

Man geht von einem Datensatz D aus, der m Klassen und l Elemente pro Klasse, insgesamt also $n = ml$ Elemente beinhaltet. In diesem Fall werden alle Bilder, die dieselbe Münze zeigen, in einer Klasse zusammengefasst. Die COINS-Datenbank beinhaltet daher $m = 240$ Klassen mit $l = 10$ Elementen pro Klasse und insgesamt $n = 2400$ Elemente. Der Datensatz wird vollständig in k gleich große disjunkte Teilmengen (engl.: *folds*) D_1, D_2, \dots, D_k partitioniert. Der Klassifizierer wird k mal getestet. Für jeden Test $t \in \{1, 2, \dots, k\}$ wird $D \setminus D_t$ als Trainings- und D_t als Testset verwendet. Jedes Element aus dem Testset wird als Anfrage verwendet und soll durch Vergleich mit jedem Element aus dem Trainingsset klassifiziert werden. Die Bewertung des Klassifizierers ergibt sich durch die insgesamt Anzahl der korrekten Klassifizierungen dividiert durch die Anzahl der Elemente im Datensatz (d.h. $n \cdot m = 2400$).

Das Ergebnis ist sowohl von der Anzahl k der Teilmengen als auch von der Aufteilung der Elemente in die Teilmengen abhängig. Bei *kompletter Cross-Validation* wird der Durchschnitt aller $\binom{n}{n/k}$ Möglichkeiten, wie $\frac{n}{k}$ Elemente aus insgesamt n ausgewählt werden können, berechnet. *Komplette Cross-Validation* ist üblicherweise rechnerisch zu aufwändig und *k-fold Cross-Validation* bietet eine gute Abschätzung, die durch Wiederholung mit unterschiedlichen Partitionierungen weiter verbessert werden kann [12]. Es sollte dafür gesorgt werden, dass die Verteilung der Klassen in den Partitionen möglichst der Verteilung im gesamten Datensatz entspricht, und man spricht dann von *stratifizier-*

ter *Cross-Validation*.

Zur Bestimmung der optimalen Parameter der in Abschnitt 2 beschriebenen Methoden wird *5-fold Cross-Validation* (5-fold CV) verwendet. Die Bilder der COINS-Datenbank werden entsprechend den unterschiedlichen Aufnahmebedingungen in Partitionen eingeteilt. Das heißt beispielsweise alle Scans mit der gleichen Rotation, unabhängig ob die Vorder- oder Rückseite der Münze gezeigt wird, werden in die gleiche Partition eingeordnet. Es gibt also 5 Partitionen, die je 480 Bilder, nämlich 2 Aufnahmen jeder Münze, beinhalten. Das Ergebnis der 5-fold CV wird hier mit 100 multipliziert um einen Prozentwert zu erhalten.

Tests mit den beiden vorgestellten Sampling-Verfahren haben gezeigt, dass äquiangulares Sampling, unabhängig von den anderen Parametern, allgemein zu um rund 5% besseren Ergebnissen führt, womit die zuvor geäußerte Vermutung, dass äquiangulares Sampling für diese Anwendung besser geeignet ist, bestätigt wurde. Eine Begründung dafür wird in Abschnitt 5 diskutiert.

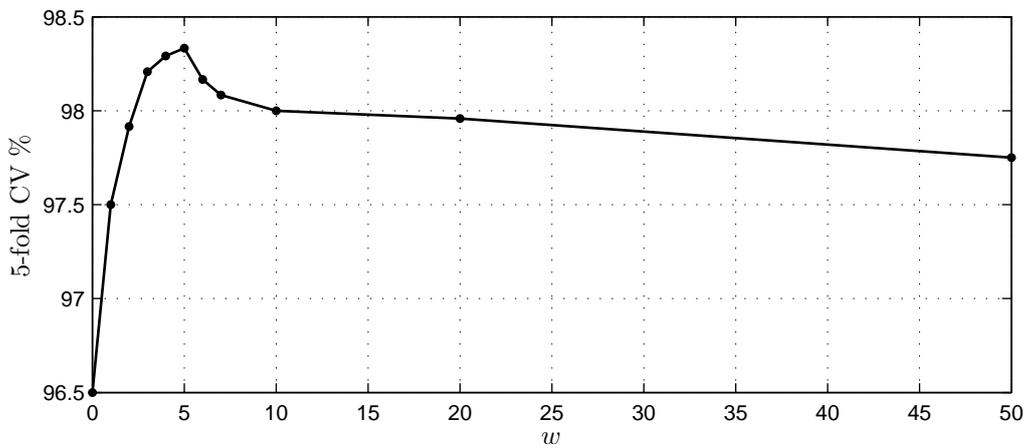


Abbildung 9: 5-fold CV mit unterschiedlichen Werten für Parameter w

w	0	1	2	3	4	5	6	10	20	50
%	96.5	97.5	97.92	98.21	98.29	98.33	98.17	98	97.96	97.75

Tabelle 2: 5-fold CV mit unterschiedlichen Werten für Parameter w

Außerdem wurde der Einfluss der Anzahl l der Stichproben beim Sampling des Kontursignals auf die Qualität der Identifikation untersucht. Es konnte gezeigt werden, dass sich wiederum unabhängig von den anderen Parametern ab einer Anzahl von 200 Stichproben das Ergebnis der 5-fold CV mit mehr Stichproben nur mehr geringfügig verbessert (kleiner 1%). In den folgenden Versuchen wird daher ein Wert $l = 256$ verwendet, der als Kompromiss zwischen rechnerischem Aufwand und Qualität gewählt wurde.

Durch eine weitere Testserie wurde der optimale Wert für den Parameter v , nämlich die Anzahl von niederfrequenten Fourier-Deskriptoren, die beim Vergleich nicht berücksichtigt werden sollen, aus Gleichung 17 bestimmt. Die anderen Parameter wurden auf den Werten $w = 0$ (Gleichung 17) und $\lambda = 0$ (Gleichung 18) festgesetzt. Durch $\lambda = 0$ ergibt sich eine volle Gewichtung auf dem Vergleich im Frequenzbereich, da v nur für diesen relevant ist. Tabelle 1 und Abbildung 8 zeigen die getesteten Werte für v und die zugehörigen Ergebnisse der 5-fold CV. Es ist ersichtlich, dass bei $v = 3$ ein Maximum der Bewertung besteht und somit wurde der optimale Wert für v gefunden.

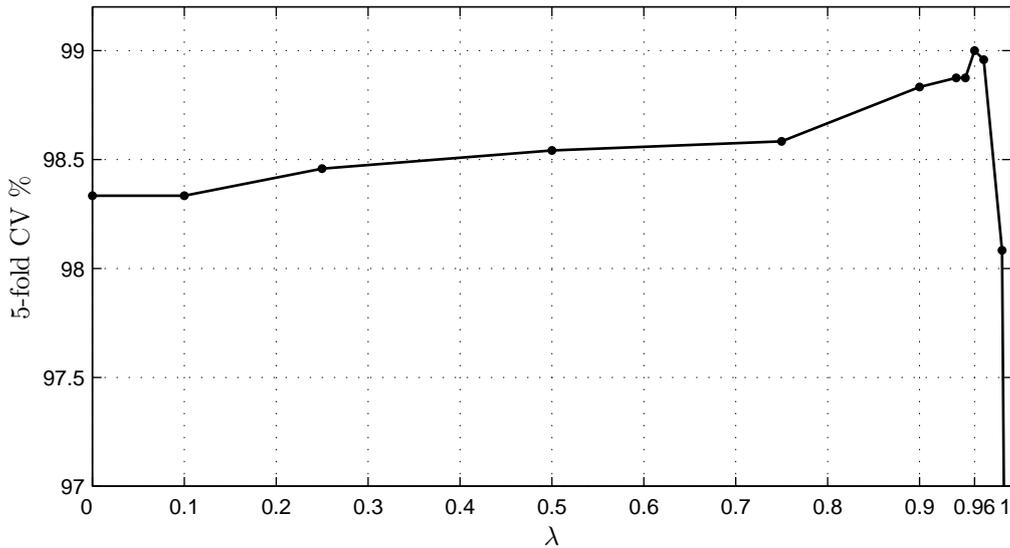


Abbildung 10: 5-fold CV mit unterschiedlichen Werten für Parameter λ

λ	0	0.1	0.5	0.75	0.9	0.95	0.96	0.97	0.99	1
%	98.33	98.33	98.54	98.58	98.83	98.88	99	98.96	98.08	92.17

Tabelle 3: 5-fold CV mit unterschiedlichen Werten für Parameter λ

Tabelle 2 und Abbildung 9 beinhalten die Ergebnisse der 5-fold CV mit unterschiedlichen Werten für den Parameter w aus Gleichung 17, der die Anzahl von zu ignorierenden hochfrequenten Fourier-Deskriptoren bestimmt. Der ermittelte optimale Wert $v = 3$ wurde in dieser Testserie bereits verwendet. Für den Gewichtungsfaktor λ wurde wiederum ein Wert von 0 fixiert. Die Unterschiede zwischen den berechneten Bewertungen bei dieser Testserie sind zwar nur gering, zwischen dem Verzicht auf zu ignorierende hochfrequente Fourier-Deskriptoren (d.h. $w = 0$) und dem optimalen Wert $w = 5$ ergibt sich allerdings eine Verbesserung von fast 2%, die bei der Implementierung nicht außer Acht gelassen werden sollte.

Zuletzt wurden für den Gewichtungsfaktor λ aus Gleichung 18 unterschiedliche Werte evaluiert. Tabelle 3 und Abbildung 10 stellen die Ergebnisse der 5-fold CV bei dieser Testserie. Für v und w (Gleichung 17) wurde immer der zuvor ermittelte optimale Wert 3 bzw. 5 verwendet. Es zeigt sich, dass für $\lambda = 0.96$ die Bewertung ein Maximum von 99% erreicht. Diese Bewertung stellt das beste Ergebnis für die 5-fold CV dar, das mit dem in dieser Arbeit beschriebenen Verfahren für die COINS-Datenbank erreicht wurde. Eine weitere Optimierung durch noch genauere Bestimmung von λ ist nicht sinnvoll, da die Ergebnisse für $\lambda = 0.95$ und $\lambda = 0.97$ bereits sehr dicht am erreichten Maximum liegen. Voraussetzung ist, dass die Bewertungsfunktion stetig ist und nur ein Maximum aufweist. Der Kurvenverlauf in Abbildung 10 deutet stark darauf hin und ein weiteres besseres Maximum ist unwahrscheinlich.

4.2.2 Bull's Eye Test

Der sogenannte *Bull's Eye Test* (BET) ist eine für die Bewertung von Retrieval-Verfahren übliche Erprobung [1]. Retrieval bezeichnet allgemein das Auffinden eines Objekts in

einem großen Datenbestand. Das hier beschriebene Identifikationsverfahren kann auch als Retrieval-Verfahren angesehen werden. Ausgegangen wird wiederum von einem Datensatz mit m Klassen und l Elementen (Bildern) pro Klasse. Jedes Bild des Datensatzes wird als Anfrage verwendet und mit allen Bildern verglichen (inklusive sich selbst). Für jede Anfrage wird unter den ersten $2l$ ähnlichsten Bildern (d.h. die Bilder die den geringsten Unähnlichkeitswert aufweisen) die Anzahl der Bilder, die dieselbe Klasse haben wie das Anfrage-Bild, gezählt und aufsummiert. Die Gesamtzahl wird durch die maximale erreichbare Zahl l^2m bei optimaler Klassifizierung dividiert und mit 100 multipliziert, um einen Prozentwert zu erhalten, der ein Maß für die Zuverlässigkeit des Klassifikationsverfahrens ist.

Der BET wurde mit der COINS-Datenbank und den zuvor ermittelten optimalen Parametern durchgeführt und liefert eine Bewertung von 94.3%. Dieser Wert wird in Abschnitt 5 mit den Ergebnissen anderer Shape-Matching-Verfahren verglichen.

4.2.3 Double Training Test

Cross-Validation und der Bull's Eye Test dienen dazu, die Zuverlässigkeit des Identifikationsverfahrens zu bewerten, und dadurch eine Optimierung und den Vergleich mit anderen Verfahren zu ermöglichen. Sie haben allerdings bei dieser Anwendung keinen direkten Bezug zur Praxis. Ein Wert für die Wahrscheinlichkeit, dass eine unbekannte Münze in einer realistischen Datenbank von gestohlenen Münzen gefunden und damit identifiziert werden kann, soll ermittelt werden. Bei Cross-Validation umfasste das Trainingsset aus der COINS-Datenbank jeweils 8 Bilder derselben Münze. Um eine korrekte Identifikation einer unbekanntes Münze mit dem *nearest neighbor*-Verfahren zu erhalten, musste nur mindestens eines der 8 Münzbilder, die dieselbe Münze zeigen, eine geringere Unähnlichkeit aufweisen, als alle Münzbilder, die eine andere Münze zeigen. In einer realen Datenbank von gestohlenen Münzen werden von jeder Münze nicht so viele Bilder vorhanden sein. Typischerweise steht von jeder Münze jeweils nur eine Aufnahme der Vorderseite und eine der Rückseite zur Verfügung. Dieses Setup soll mit Hilfe der COINS-Datenbank simuliert werden und wird hier als Double Training Test (DTT) bezeichnet. Das Testset beinhaltet nur genau 8 Bilder, die alle dieselbe Münze zeigen, das Trainingsset besteht aus den übrigen 2 Bildern derselben Münze, von denen eines die Vorder- und eines die Rückseite zeigt, und allen Bildern von den anderen Münzen, d.h. insgesamt 2392 Bilder. Wie zuvor soll jedes Bild des Testsets durch Vergleich mit jedem Bild aus dem Trainingsset und Betrachtung des *nearest neighbor* identifiziert werden. Dieser Test wird mit jeder Münze für alle 5 Möglichkeiten das Trainingsset aufzubauen wiederholt, wenn jeweils zwei Bilder, die unter gleichen Aufnahmebedingungen (also z.B. je ein Scan der Vorder- und der Rückseite mit 0° Rotation) akquiriert wurden, enthalten sein sollen. Insgesamt müssen also $5 \cdot 240 = 1200$ Tests durchgeführt werden, von denen jeder maximal 8 korrekte Identifikationen aufweisen kann (da das Testset aus 8 Münzen besteht). Insgesamt ist die maximal erreichbare Anzahl an korrekten Identifikationen daher $1200 \cdot 8 = 9600$. Die tatsächlich erreichte Anzahl wird durch diese Zahl dividiert und mit 100 multipliziert um einen Prozentwert zu erhalten. Die Tests müssen allerdings in der Praxis nicht zwingend getrennt voneinander durchgeführt werden. Das Ergebnis kann auch durch geschickte Auswertung eines einzelnen Tests, bei dem jedes Münzbild mit jedem anderen verglichen wurde, ermittelt werden.

Das hier beschriebene Identifikationsverfahren erreicht beim DTT mit den zuvor bestimmten optimalen Parametern eine Bewertung von 93.19%. Diese Zahl entspricht der Korrektidentifikationsrate, d.h. der Wahrscheinlichkeit, dass eine unbekannte Aufnahme korrekt identifiziert wird. Die Falschidentifikationsrate ist daher 6.81%.

4.2.4 Verifikation

In Abschnitt 2.5 wurde eine Idee zur Verifikation der Identifikation beschrieben, die nun experimentell untersucht werden soll. Die Verifikation ist möglich, wenn Aufnahmen von beiden Seiten einer zu identifizierenden Münze verfügbar sind. Die Identifikation wird nur akzeptiert, wenn beide Aufnahmen gleich klassifiziert werden. Folglich wurde der DTT wiederholt und jede Klassifikation einer Münzaufnahme wurde zusätzlich durch die Klassifikation der zugehörigen (unter gleichen Aufnahmebedingungen akquirierten) Münzaufnahme der anderen Seite verifiziert. Die Falschidentifikationsrate konnte dadurch auf 2.28% gesenkt werden. Allerdings wurde auch die Korrektidentifikationsrate auf 89.23% verschlechtert, da durch die Verifikation auch korrekte Identifikationen zurückgewiesen werden.

Die Zurückweisung von Identifikationen für Münzen, die in der Datenbank nicht vorhanden sind, d.h. die Klassifikation als *unbekannt*, ist ein wünschenswertes Ziel der Verifikation. Dies wird durch einen weiteren Test, der hier *No Training Test* (NTT) genannt wird, evaluiert. Als Testset werden jeweils alle Aufnahmen einer Münze verwendet und als Trainingsset alle anderen Münzaufnahmen. Eine korrekte Identifikation ist daher nicht möglich und alle Aufnahmen des Testsets sollten als *unbekannt* klassifiziert werden.

	Bekannte Münzen (DTT)	Unbekannte Münzen (NTT)
Korrekt klassifiziert	89.23%	65.67%
Falsch klassifiziert	10.77%	34.33%

Tabelle 4: Ergebnisse der Verifikations-Experimente

Tabelle 4 stellt die Wahrscheinlichkeiten einer korrekten Klassifikation der Münzen beim DTT und NTT nach Verifikation dar. Eine korrekte Klassifikation bedeutet, dass eine bekannte Münze (d.h. eine Münze, die im Trainingsset vorhanden ist) korrekt identifiziert wurde oder dass eine unbekannte Münze, die nicht im Trainingsset vorhanden ist, richtigerweise als *unbekannt* zurückgewiesen wird. Umgekehrt liegt eine falsche Klassifikation vor, einerseits wenn eine bekannte Münze falsch identifiziert oder als *unbekannt* klassifiziert wird, andererseits wenn eine unbekannte Münze fälschlicherweise identifiziert wird, statt als *unbekannt* zurückgewiesen zu werden.

5 Diskussion

In Abschnitt 4 wurden optimale Parameter für das hier präsentierte Identifikationsverfahren bestimmt. Es wurde gezeigt, dass äquiangulares Sampling bei dieser Anwendung äquidistantem Sampling vorzuziehen ist. Da bei einigen Münzen Segmentierungsfehler auftreten, die kleine Einbuchtungen und Ausbuchtungen auf der Münzkontur bewirken, war dieses Ergebnis zu erwarten. Durch diese Fehler ergibt sich eine Verlängerung der Kontur, die eine Veränderung aller Kontursignalwerte ab der betroffenen Stelle bewirkt, wenn äquidistantes Sampling verwendet wird. Bei äquiangularem Sampling sind nur wenige Kontursignalwerte, die direkt bei Einbuchtungen oder Ausbuchtungen gewählt werden, verfälscht und es ist daher robuster im Falle von typischen Segmentierungsfehlern, bei denen nur ein geringer Anteil der Kontur betroffen ist - vorausgesetzt die Fehler sind nicht so stark, dass sich eine deutliche Verschiebung des Ausgleichskreises ergibt.

Es wurde gezeigt, dass die Verwendung der Konstanten $v = 3$ und $w = 5$ zur Bestimmung der Fourier-Deskriptoren, die beim Vergleich eines diskreten Signals der Länge 256 im Frequenzbereich ignoriert werden sollen, zu einer Verbesserung des Verfahrens führen. Die Idee dahinter ist, dass durch Segmentierungsfehler kleine Unterschiede beim Bestim-

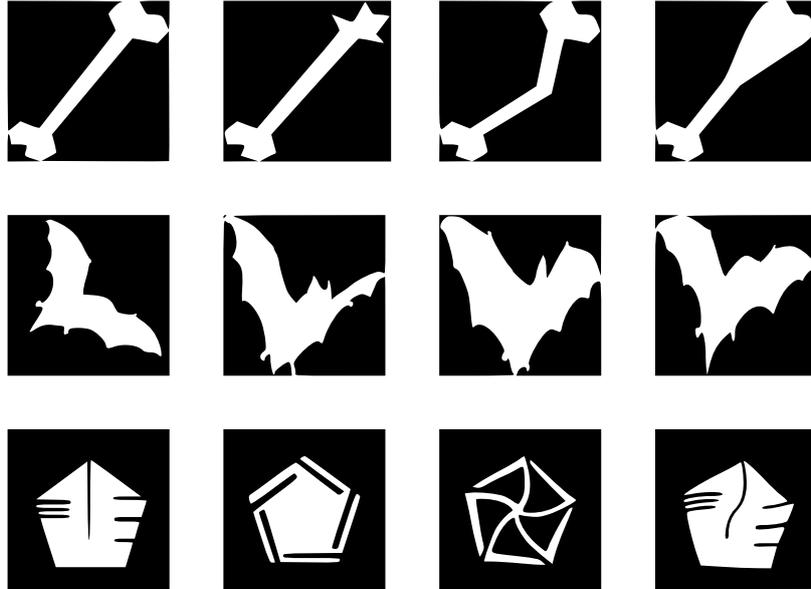


Abbildung 11: Beispiele aus dem MPEG7-CE-Shape-1-Datensatz

men des Ausgleichskreises entstehen, die zu inkorrekten Werten bei den niederfrequenten Fourier-Deskriptoren führen, und es daher sinnvoll ist diese beim Vergleich nicht zu berücksichtigen. Ähnlich verhält es sich bei hochfrequenten Fourier-Deskriptoren. Diese werden durch Quantisierungsfehler und das Sampling beeinflusst und tragen daher nicht sinnvoll zu der Berechnung des Unähnlichkeitsmaßes bei. Es sollte beachtet werden, dass diese Werte auf die Anzahl der verwendeten Samples, in diesem Fall 256, abgestimmt werden müssen.

Der optimale Gewichtungsfaktor $\lambda = 0.96$ ist unabhängig von der Anzahl der Samples, da sowohl beim Vergleich im Frequenzbereich als auch im Ortsbereich über die Länge des gesampteten Kontursignals gemittelt wird. Der Wert von λ , der nahe bei 1 liegt, könnte zu der Schlussfolgerung führen, dass der Vergleich im Ortsbereich für die Zuverlässigkeit des Identifikationsverfahrens wichtiger ist. Das ist allerdings nicht korrekt, da das Unähnlichkeitsmaß im Ortsbereich aufgrund der Art der Berechnung immer wesentlich kleiner ist. Die Gewichtung erfolgt praktisch eher zugunsten einer Verstärkung der Bedeutung des Vergleichs im Frequenzbereich, da dieser durch die Möglichkeit, selektiv niedrige und hohe Frequenzen des Kontursignals zu ignorieren, zu einem fehlerresistenteren Ergebnis führt. Die Einbeziehung des Vergleichs im Ortsbereich ist trotzdem sinnvoll, da sich dadurch eine weitere Zuverlässigkeitssteigerung des Identifikationsverfahrens ergibt, wie die Ergebnisse der Experimente zeigen.

Durch die Optimierung der Parameter in Abschnitt 4 konnte die Zuverlässigkeit des Identifikationsverfahrens verbessert werden. Das Verfahren soll nun mit anderen Möglichkeiten der automatisierten Identifikation von historischen Münzen auf Bildern verglichen werden.

5.1 Shape-Contexts-Matching

Zur Evaluierung von unterschiedlichen Shape-Matching-Verfahren wurden diese in mehreren aktuellen Publikationen entsprechend der Spezifikation des MPEG-7 Core Experiments [9] zur Klassifikation von allgemeinen Formen auf dem MPEG-7-CE-Shape-1-Datensatz angewendet. So werden beispielsweise von Latecki et al. [14] sechs unterschied-

liche Shape-Matching-Verfahren mit Hilfe des BET auf diesem Datensatz verglichen. In einer späteren Publikation von Ling und Okada [15] wird gezeigt, dass bei den Ergebnissen des BET eine Methode, die auf Shape-Contexts-Matching (SCM) [3] basiert, alle bisher vorgeschlagenen Verfahren übertrifft. Der MPEG-7-CE-Shape-1-Datensatz, der in Folge einfach nur mehr als MPEG7-Datensatz bezeichnet wird, beinhaltet 70 Klassen mit je 20 Elementen pro Klasse, insgesamt also 1400 Bilder. Die Formen wurden semantisch nach menschlichem Verständnis beschriftet und so in Klassen unterteilt. Abbildung 11 zeigt einige Beispiele aus diesem Datensatz, wobei alle Bilder in der gleichen Zeile jeweils zur gleichen Klasse gehören.

	SCM	DCSM
MPEG7	76.79%	71.75%
COINS	50.64%	94.30%

Tabelle 5: BET-Ergebnisse für den MPEG7-Datensatz und die COINS-Datenbank mit SCM und DCSM

Hier werden die Ergebnisse einer Implementierung von SCM in MATLAB aus [7] verwendet, bei der sich eine BET-Bewertung für den MPEG7-Datensatz von 76.79 % ergibt und damit ein sehr guter Wert im Vergleich mit anderen Verfahren erreicht wird. Der Shape-Context eines Konturpunkts wird als zweidimensionales Histogramm, das die Distanzen zu allen anderen Konturpunkten und die Differenzen der Tangentorientierungen beinhaltet, dargestellt. Das Shape-Matching erfolgt durch Lösen eines Assignment-Problems mit dem *Hungarian Algorithm* [13].

Bei der Anwendung der in dieser Arbeit vorgeschlagenen DCSM-Methode auf dem MPEG7-Datensatz wurde eine erwartungsgemäß schlechtere Bewertung des BET von 71.75 % erzielt. DCSM ist für die allgemeinen Formen dieses Datensatzes weniger geeignet, da die Methode von zu starken Annahmen über die Objekteigenschaften ausgeht, die nur bei Münzen und ähnlichen Formen erfüllt sind. Die Unterschiede zwischen den Elementen einer Klasse im MPEG7-Datensatz sind so groß, dass insbesondere die Variationen beim Bestimmen des Ausgleichskreises, und die daraus resultierenden unterschiedlichen Kontursignale, die korrekte Klassifikation bei vielen Formen verhindern. Obwohl die Methode nicht für diese Daten entworfen wurde, erreicht sie trotzdem ein besseres Ergebnis als vier der sechs in [14] veröffentlichten Shape-Matching-Verfahren.

Umgekehrt wurde SCM für den Vergleich der Konturen von historischen Münzformen getestet. Ein BET auf der COINS-Datenbank erzielte eine Bewertung von nur 50.64% [7]. Dieses Ergebnis wird von dem durch das hier beschriebene DCSM-Verfahren erreichten Ergebnis von 93.30% deutlich übertroffen. Somit konnte gezeigt werden, dass der Einsatz eines speziellen Shape-Matching-Verfahrens für den Formvergleich historischer Münzen gerechtfertigt ist. Global gesehen unterscheiden sich die Formen der meisten Münzen nicht erheblich, denn die Konturen weichen alle nur geringfügig von einem Kreis ab. Deshalb müssen diese geringen Unterschiede durch eine geeignete Darstellung hervorgehoben werden. Wie die Testresultate zeigen, wird diese Anforderung durch das hier berechnete Kontursignal in hohem Maße erfüllt.

Überdies konnte das hier vorgestellte DCSM-Verfahren bei den Experimenten SCM auch in der Geschwindigkeit deutlich übertreffen. Ein Vergleich eines einzelnen Münzbildes mit allen 2400 Bildern der COINS-Datenbank, auf denen Münzkonturen bereits extrahiert und in geeigneter Weise abgespeichert sind, dauert auf einem üblichen Computer mit 3 GHz CPU durchschnittlich 5 Sekunden bei DCSM und über 8 Minuten bei SCM. Der Unterschied um einen Faktor von rund 100 ergibt sich nicht nur aus der effizienten Implementierung von DCSM in C++, die für diese Arbeit erstellt wurde, sondern auch durch den allgemein höheren rechnerischen Aufwand von SCM, da bei jedem SCM-Vergleich ein komplexes Optimierungsproblem gelöst werden muss [3].

5.2 Lokale Deskriptoren

Ein anderer Ansatz zur Identifikation von antiken Münzen ist der Vergleich von lokalen Deskriptoren der Münzoberfläche. Die Idee ist vielversprechend, da analog zu den Konturen auch der Stempel jeder historischen Münze durch die manuelle Prägung, Abnutzung und Verwitterung einzigartig ist.

Die Zuverlässigkeit der Identifikation von Münzen aus der COINS-Datenbank wurde für [7] mit einer Methode, die auf dem Vergleich von SURF-Features (engl.: *speeded up robust features*) [2] basiert, evaluiert. Die ausgezeichneten Punkte werden als lokale Extrema der Gauß-Funktion angewendet im *scale space* ermittelt. Jeder SURF-Deskriptor beschreibt Haar-Wavelets in einer Nachbarschaft eines ausgezeichneten Punktes. Die Testauswertungen mit SURF-Features werden nun mit dem hier vorgeschlagenen Shape-Matching-Verfahren verglichen.

Als Test wurde 5-fold CV mit den gleichen Training- und Testsets wie in Abschnitt 4.2.1 durchgeführt. Die Identifikation mit Hilfe von SURF-Features erreicht eine Bewertung von 60.64%. Das ist ein deutlich schlechteres Ergebnis als die Bewertung von 99%, die mit dem hier beschriebenen DCSM-Verfahren erzielt wurde. Einer der Nachteile des Vergleichs der Münzstempel ist die fehlende Invarianz gegenüber Flips. Es muss beachtet werden, dass die Methode damit auf jeden Fall bei diesem Test benachteiligt ist, da im gesamten Trainingsset von jeder Münze nur vier statt acht Bildern vorhanden sind, durch die eine korrekte Identifikation dieser Münze auf einer Testset-Aufnahme möglich ist. Dies ist aber keine ausreichende Begründung für die erheblich schlechtere Bewertung.

Die Identifikation der Münzen aus der COINS-Datenbank stellt lokale Deskriptoren vor eine große Herausforderung, denn es ist notwendig die Einzigartigkeit jeder Münze ausreichend hervorzuheben. Grob betrachtet sind fast alle Münzen in der COINS-Datenbank sehr ähnlich, denn gehören alle dem gleichen Münztyp an und unterscheiden sich nur durch minimale Variationen der Münzstempel. Die Auswahl und Beschreibung der ausgezeichneten Punkte ist außerdem nicht völlig unabhängig von den Beleuchtungsverhältnissen und der Qualität der Aufnahmen, die in der COINS-Datenbank absichtlich unterschiedlich gewählt sind. Andererseits ist die Zuverlässigkeit des hier vorgeschlagenen Formvergleichs abhängig von der Qualität der Segmentierung und damit indirekt auch von den Aufnahmebedingungen beeinflusst. Lokale Deskriptoren sind besser zur Klassifikation nach dem Münztyp als zur Identifikation geeignet, wie auch Experimente in [25] zeigen.

6 Schlussfolgerungen

In der vorliegenden Arbeit wird ein Identifikationsverfahren für handgeprägter Münzen implementiert, das auf der einzigartigen Münzform basiert. Eine unbekannte Münzaufnahme wird durch Vergleich mit einer Datenbank von bekannten Aufnahmen identifiziert. Dazu wird sie mit jeder Münzaufnahme durch ein neues skalierungs-, translations-, rotations- und spiegelungsinvariantes Shape-Matching-Verfahren, DCSM, verglichen und so das ähnlichste Münzbild gefunden, das wahrscheinlich die gleiche Münze zeigt.

Die Auswertung des praxisrelevanten DTT ergab, dass durch das Identifikationsverfahren 93.19% aller Münzen der COINS-Datenbank korrekt identifiziert werden können, wenn jeweils nur zwei Aufnahmen einer zu identifizierenden Münze im Trainingsset vorhanden sind. Man kann davon ausgehen, dass dieses Ergebnis auch durch andere Methoden, die ausschließlich die äußere Kontur der Münzen vergleichen, nicht wesentlich übertroffen werden kann. Theoretisch ist zwar jede Münzform einzigartig, viele weisen aber eine sehr hohe Ähnlichkeit auf und aufgrund der digitalen Darstellung und unvermeidbarer Segmentierungsfehler ist eine Identifikationsrate von 100% nicht erreichbar.

In der Praxis stellt die Abhängigkeit von der erfolgreichen Segmentierung den größten Nachteil dieses Identifikationsverfahrens dar. Vielversprechende Segmentierungsverfahren, auch speziell für antike Münzen [26], wurden entwickelt und in dieser Arbeit angewandt, können aber keine fehlerfreie Segmentierung garantieren. Die Aufnahmebedingungen der für die Experimente verwendeten COINS-Datenbank sind bekannt und das Segmentierungsverfahren konnte darauf optimiert werden. Liegen Aufnahmen vor, die heterogenen Hintergrund, geringen Kontrast, Schattenwürfe oder zusätzliche Objekte beinhalten, so steigt auch die Zahl der Segmentierungsfehler und damit sinkt die Zuverlässigkeit des Identifikationsverfahrens.

Es wurde gezeigt, dass eine Verifikation des Identifikationsverfahrens bis zu einem gewissen Grad möglich ist. Die Wahrscheinlichkeit einer falschen Identifikation kann dadurch verringert werden, allerdings nur auf Kosten der insgesamt möglichen Identifikationen. Je nach Anwendung kann diese Eigenschaft trotzdem wünschenswert sein. Die Möglichkeit Münzen, die nicht identifizierbar sind, weil sie in der Vergleichsdatenbank nicht vorhanden sind, als *unbekannt* zu klassifizieren, ist durch den Verifikationsschritt auch vorhanden. Das erreichte Ergebnis von 65.67% korrekten Klassifikationen als *unbekannt* ist allerdings für eine praktische Anwendung nicht ausreichend. Das Problem hierbei ist die Annahme, dass die Wahrscheinlichkeit hoch ist, dass zwei Bilder der gleichen Münze, wenn diese nicht in der Datenbank vorhanden ist, unterschiedlich – natürlich in beiden Fällen falsch – identifiziert werden. Tatsächlich ist es wahrscheinlich, dass die beiden Aufnahmen gleich identifiziert werden, weil die Münzkonturen auf beiden Bildern – abgesehen von Quantisierungsrauschen und Segmentierungsfehlern – gleich sind und somit beide die größte Ähnlichkeit zur gleichen Münze aus der Datenbank aufweisen. So kommt es dazu, dass eine falsche Identifikation einer eigentlich nicht identifizierbaren Münze durch die Verifikation bestätigt wird.

Die Verifikation der Identifikationen durch ein anderes Verfahren, das nicht auf dem Formvergleich basiert, wäre sinnvoller. Die Kombination dieses Identifikationsverfahrens mit anderen Ansätzen wird erforscht und führt zu einer höheren Zuverlässigkeit. In [7] wird das hier beschriebene Shape-Matching-Verfahren mit dem Lokalen-Deskriptoren-Ansatz kombiniert. Die Bewertung des dort verwendeten *leave-N-out*-Tests mit der COINS-Datenbank kann von 90.23%, die durch reinen Formvergleich erreicht werden, auf 95.16% durch die Kombination mit SURF-Features zur Beschreibung des Münzstempels erhöht werden. Weitere Forschung sollte auch die Möglichkeiten der Verifikation dieses Identifikationsverfahrens durch den Lokalen-Deskriptoren-Ansatz beinhalten, um auch eine zuverlässige Methode zur Klassifikation als *unbekannt* zu erreichen.

Die in Abschnitt 1.1 definierten Anwendungsfälle für das COINS-Projekt können durch das hier beschriebene Identifikationsverfahren teilweise erfüllt werden. Der 1. Anwendungsfall (die Identifikation einer einzelnen verdächtigen Münze mit Hilfe einer Datenbank von gestohlenen Münzen) kann mit hoher Zuverlässigkeit vollzogen werden, insbesondere da eine menschliche Verifikation der Identifikation möglich ist. Es gibt viele ähnliche Anwendungsfälle die auch abgedeckt werden - beispielsweise ein Museum, das eine Münze aus seinem Besitz mit Hilfe einer internen Datenbank des Münzbestandes identifizieren möchte. Die Realisierung des 2. Anwendungsfalles aus Abschnitt 1.1 (die automatisierte Suche im Web nach einer gestohlenen Münze) ist durch die mangelhafte Option der automatisierten Verifikation eines Identifikationsergebnisses nur eingeschränkt möglich. In jedem Fall ist vor der Identifikation ein weiterer Schritt zur automatisierten Filterung von Bildern, die keine Münzen enthalten, notwendig, da die Anwendung des Verfahrens auf beliebige Bilder zu nicht verwertbaren Resultaten führen könnte. Die Wahrscheinlichkeit von Segmentierungsfehlern steigt außerdem für Bilder, die aus einer Websuche stammen, da diese unter möglicherweise schlechteren Aufnahmebedingungen akquiriert wurden, d.h. Schattenwürfe, geringen Kontrast, inhomogenen Hintergrund und andere Objekte enthalten können. Weitere Forschung an Segmentie-

rungsmethoden für antike Münzen und Evaluierung mit Datensätzen von “problematischen” Bildern sollte daher durchgeführt werden.

In Abschnitt 2.3.3 wurde beschrieben wie als Nebenprodukt des Formvergleichs im Ortsbereich die Rotation zwischen zwei Abbildungen derselben Münze ermittelt werden kann. Diese Information kann in der Praxis dazu verwendet werden, beide Bilder gleich zu rotieren, um eine menschliche Verifikation oder den Vergleich mit einem anderen Ansatz zu vereinfachen bzw. effizienter zu machen.

Es kann konkludiert werden, dass der Formvergleich von historischen Münzen auf Bildern möglich ist und sinnvoll zur Identifikation eingesetzt werden kann. Nach Extraktion aller Münzformen aus einer Vergleichsdatenbank, kann die Identifikation einer unbekanntes Münze sehr schnell erfolgen. Es wurde durch Experimente mit einem großen Datensatz von historischen Münzen gezeigt, dass das hier beschriebene Verfahren besser zum Formvergleich von Münzen geeignet und deutlich schneller ist, als andere Shape-Matching-Verfahren, die vielversprechende Ergebnisse für allgemeine Formen liefern. Das Verfahren weist zusätzlich eine höhere Zuverlässigkeit auf, als Ansätze die nur auf dem Vergleich des Münzstempels basieren. Die Identifikation durch Formvergleich ist, im Gegensatz zu anderen Ansätzen, außerdem unabhängig davon, welche Seite einer Münze durch die Vergleichsaufnahme gezeigt wird (Flipinvarianz). Die Kombination mit einer Methode zum Vergleich des Münzstempels steigert die Zuverlässigkeit dennoch weiter und sollte daher in der Praxis eingesetzt werden.

7 Danksagungen

Die Entstehung dieser Arbeit wurde von Dr. Reinhold Huber-Mörk (Austrian Research Centers GmbH - ARC) und Dr. Martin Kampel (Technische Universität Wien, E183-2) betreut. Die beschriebenen Methoden wurden gemeinsam mit den Betreuern erarbeitet und durch diese geprüft. Der Autor möchte sich für die ausgezeichnete Betreuung und die Möglichkeit, an dem Wissen und der Praxis der Betreuer auf dem Gebiet der digitalen Bildverarbeitung teilzuhaben, bedanken. Die Testauswertungen des Lokalen-Deskriptoren-Ansatzes mit der COINS-Datenbank wurden von Mag. Maia Zaharieva (Technische Universität Wien, E183-2) erstellt und freundlicherweise für die Arbeit zur Verfügung gestellt.

Weiterer Dank geht an Dr. Mark Blackburn und Jonathan Jarret, sowie deren Team vom Fitzwilliam Museum, Cambridge, UK, für das Bereitstellen der verwendeten Testbilder und numismatischer Expertise.

Dr. Georg Czedik-Eysenberg, Isabella Czedik-Eysenberg und der gesamten Familie des Autors sei gedankt für das Korrekturlesen dieser Arbeit und allgemeine Unterstützung während des Entstehungsprozesses.

Diese Arbeit wurde als Teil des COINS-Projekts (FP6-SSP5-044450) von der Europäischen Union unterstützt. Der Inhalt spiegelt allerdings nur die persönliche Sicht des Autors wider und die Europäische Union kann für keine missbräuchliche Verwendung der in dieser Arbeit enthaltenen Informationen oder deren Auswirkungen verantwortlich gemacht werden.

Literatur

- [1] Tomasz Adamek and Noel O'Connor. Efficient contour-based shape representation and matching. In *MIR '03: Proceedings of the 5th ACM SIGMM international workshop on Multimedia information retrieval*, pages 138–143, New York, NY, USA, 2003. ACM.
- [2] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *Computer Vision – ECCV 2006*, volume 3951/2006 of *LNCS*, pages 404–417. Springer, 2006.
- [3] Serge Belongie, Jitendra Malik, and Jan Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(4):509–522, 2002.
- [4] James W. Cooley and John W. Tukey. An algorithm for the machine calculation of complex fourier series. *Mathematics of Computation*, 19:297–301, 1965.
- [5] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing (3rd Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2006.
- [6] Reinhold Huber, Herbert Ramoser, Konrad Mayer, Harald Penz, and Michael Rubik. Classification of coins using an eigenspace approach. *Pattern Recognition Letters*, 26(1):61–75, 2005.
- [7] Reinhold Huber-Mörk, Maia Zaharieva, and Hermann Czedik-Eysenberg. Object identification using shape and local descriptors. 2008. (submitted).
- [8] Anil K. Jain. *Fundamentals of digital image processing*, chapter 3.9: Color coordinate systems, 4: Image Sampling and Quantization. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1989.
- [9] Sylvie Jeannin and Miroslaw Bober. Description of core experiments for mpeg-7 motion/shape. Technical Report ISO/IEC JTC 1/SC 29/WG 11 MPEG99/N2690, 1999.
- [10] Bernd Jähne. *Digitale Bildverarbeitung*. Springer, Berlin, 6 edition, 2005.
- [11] Martin Kämpel and Sebastian Zambanini. Coin data acquisition for image recognition. In *36th Conference on Computer Applications and Quantitative Methods in Archaeology (CAA'08)*, Budapest, Hungary, 2008. (in press).
- [12] Ron Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *IJCAI (International Joint Conference on Artificial Intelligence)*, pages 1137–1145, 1995.
- [13] Harold W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2:83–87, 1955.
- [14] Longin J. Latecki, Rolf Lakämper, and Ulrich Eckhardt. Shape descriptors for non-rigid shapes with a single closed contour. In *Proc. Conf. on Computer Vision and Pattern Recognition*, pages 424–429, 2000.
- [15] Haibin Ling and Kazunori Okada. An efficient earth mover's distance algorithm for robust histogram comparison. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 29(5):840–853, 2007.

- [16] Michael Nölle, Harald Penz, Michael Rubik, Konrad J. Mayer, Igor Holländer, and Reinhard Granec. Dagobert – a new coin recognition and sorting system. In *Proc. of Intl. Conf. on Digital Image Computing – Techniques and Applications*, pages 329–338, 2003.
- [17] Markus Peura and Jukka Iivarinen. Efficiency of simple shape descriptors, 1997.
- [18] Marco Reisert, Olaf Ronneberger, and Hans Burkhardt. An efficient gradient based registration technique for coin recognition. In *Proc. of Muscle CIS Coin Competition Workshop*, pages 19–31, 2006.
- [19] Mehmet Sezgin and Bülent Sankur. Survey over image thresholding techniques and quantitative performance evaluation. *J. Electron. Imaging*, 13(1):146–165, 2004.
- [20] Milan Sonka, Vaclav Hlavac, and Roger Boyle. *Image Processing, Analysis, and Machine Vision*. PWS - an Imprint of Brooks and Cole Publishing, 2nd edition, 1998.
- [21] Richard V. Southwell. *Relaxation Methods in Theoretical Physics*. Oxford University Press, Oxford, 1946.
- [22] Laurens J. Van Der Maaten and Paul Boon. COIN-O-MATIC: A fast system for reliable coin classification. In *Proc. of the Muscle CIS Coin Competition Workshop*, pages 7–18, Berlin, Germany, 2006.
- [23] Laurens J. Van Der Maaten and E.O. Postma. Towards automatic coin classification. In *Proc. of Conf. on Electronic Imaging and the Visual Arts*, pages 19–26, Vienna, Austria, 2006.
- [24] Shimon D. Yanowitz and Alfred M. Bruckstein. A new method for image segmentation. *Computer Vision, Graphics and Image Processing*, 46(1):82–95, 1989.
- [25] Maia Zaharieva, Reinhold Huber-Mörk, Michael Nölle, and Martin Kampel. On ancient coin classification. In *Intl. Symp. on Virtual Reality, Archaeology and Cultural Heritage*, pages 55–62, 2007.
- [26] Sebastian Zambanini and Martin Kampel. Segmentation of ancient coins based on local entropy and gray value range. In *Proc. of the 13th Computer Vision Winter Workshop*, pages 9–16, 2008.

Automatic Number Plate Recognition in Image Sequences

Andreas Zweng
Supervisor: Martin Kampel

Automatic Number Plate Recognition in Image Sequences

Bakkalaureatsarbeit

von

Andreas ZWENG

Matr.Nr.: 0525927



Juli 2008

Betreuer:

DI Dr. Martin Kempel

1 Einleitung

Diese Bakkalaureatsarbeit beschäftigt sich mit *Automatic Number Plate Recognition*, zu Deutsch *Automatische Kennzeichentafelerkennung*, und geht speziell auf verwendete Algorithmen und deren Probleme ein. Es werden verschiedene Algorithmen vorgestellt und miteinander verglichen, wobei Vorteile und Nachteile in einer Gegenüberstellung diskutiert werden.

Bevor die Algorithmen im Detail erklärt werden, werden die Probleme der Sensortechnik besprochen und verschiedene Arten von Kameras vorgestellt, welche auf das Ergebnis der Algorithmen Einfluss haben.

Das Ziel dieser Arbeit ist die Detektion einer sich im Bild befindlichen Kennzeichentafel und die Klassifikation der Zeichen mithilfe von zusätzlicher Information bereits vergangener Frames im Videostream. Dadurch soll die Klassifikation mit fortschreitender Zeit stabiler werden und in ein Resultat konvergieren.

Diese Arbeit wurde in Kooperation und mit Förderung der Firma CogVis GesmbH durchgeführt.

2 State of the Art

Die korrekte Kennzeichentafelerkennung hängt von den umgebenden Bedingungen ab. Deswegen gibt es in der Forschung zu bestimmten Problemstellungen mehrere Ansätze zur Lösung des Problems. Die Kennzeichentafelerkennung kann allgemein in drei große Abschnitte eingeteilt werden.

- Die Kennzeichen Lokalisierung
- Die Zeichen Segmentierung
- Die Zeichen Klassifikation

Jeder dieser Punkte könnte je nach angewandter Methode nochmals in mehrere Unterpunkte geteilt werden. Während es für die Zeichensegmentation und für die Zeichenklassifikation schon effiziente Ansätze gibt, ist die Lokalisierung noch immer das größte Problem der gesamten Klassifikation. In den folgenden Abschnitten werden Vorteile und Nachteile sowie mögliche Einsatzgebiete, für die drei genannten Punkte, von bestehenden Methoden verglichen.

2.1 Kennzeichenlokalisierung

- 1.) In [7], sowie in dieser Arbeit wird das Kennzeichen anhand der Kantenprojektion gefunden. Dazu wird zuerst ein vertikaler Kantendetektor (Sobel-Kernel) auf das Bild angewandt und durch die gefundenen maxima im Histogramm der vertikalen Kantenprojektion die obere und untere Kante des Kennzeichens gefunden. In diesem eingeschränkten „Band“ wird nun der horizontale Sobel-Kantendetektor angewandt und auf ähnliche Art die linke und rechte Begrenzung des Kennzeichens gefunden (siehe Abschnitt 4). Der Vorteil dieser Methodik ist die geringe Komplexität des Algorithmus, wodurch eine schnelle Detektion durchgeführt werden kann. Der Nachteil liegt darin, dass in Bildern mit komplexem Hintergrund (viele Kanten) das Kennzeichen unter Umständen nicht die maximale Kantenprojektion liefert, wodurch es zu Fehllokalisierungen kommen kann. Ein Einsatzgebiet für diese Methode wäre eine Szene mit homogenem Hintergrund.

- 2.) In [11] soll das Kennzeichen anhand der horizontalen Kanten am Auto Schrittweise eingeschränkt werden. Dabei werden im ersten Schritt horizontale Kanten gefunden und daraus ein Histogramm gebildet (Kantenprojektion). Die horizontalen Kanten am Auto ermöglichen eine Einschränkung auf den Bereich der Autobreite. Danach wird eine vertikale Kantenprojektion wie in Punkt 1.) durchgeführt und das Kennzeichen an den oberen und unteren Kanten abgeschnitten. Im letzten Schritt werden horizontale Kanten gefunden und das Kennzeichen wie in Punkt 1.) links und rechts limitiert. Der Vorteil dieser Methode liegt an der Vorverarbeitung des Bildes zur Einschränkung des Suchbereiches für die Kantenprojektion. Ein Nachteil könnte sein, dass das Kennzeichen durch die horizontalen Kanten des Autos nicht eingeschränkt werden kann, wenn es im Hintergrund ähnlich starke horizontale Kanten gibt. Ein Einsatzgebiet dieser Methode wäre eine Szene mit einem Hintergrund, welcher keine starken horizontalen Kanten aufweist.
- 3.) In [3] wird der Harris-Corner-Detector angewandt um das Kennzeichen zu lokalisieren. Zunächst wird eine grobe Lokalisierung durchgeführt, danach werden die gefundenen Regionen verifiziert. Und letztlich werden die übrig gebliebenen Regionen expandiert und die Kennzeichenregion wird genauer lokalisiert.

2.2 Zeichensegmentation

- 1.) In [7], wie auch in dieser Arbeit werden die Zeichen mithilfe einer Schwellwertoperation vom Hintergrund des Kennzeichens segmentiert. Dabei wird ein globaler oder in einer präziseren Berechnung mehrere lokale Schwellwerte mittels iterativem Prozess gefunden und auf die Region angewandt. Die Region ist nach diesem Schritt als binäres Bild repräsentiert, wobei der Hintergrund schwarz und der Vordergrund weiß ist. Anschließend werden die Zusammenhangskomponenten gesucht und als unabhängige Objekte betrachtet.
- 2.) In [6] werden die Zeichen wie folgt segmentiert. Zunächst werden die Zeichen mittels „contrast-stretching“ verstärkt. Anschließend folgt eine Kantendetektion mithilfe des Laplace-Operators. Als nächsten Schritt folgt die Suche nach den Zusammenhangsregionen mittels „region-growing“. Danach werden mittels bekannter Information der Kennzeichenattribute (Abstand der Zeichen zueinander usw.) die Zeichen verworfen, die den gegebenen Anforderungen nicht entsprechen. Letztlich wird die Region binärisiert.

2.3 Zeichenklassifikation

- 1.) In [1] wurde ein 2-Layer-Probabilistisches Neuronales Netzwerk verwendet. Die verwendete Topologie ist 108-180-36, was bedeutet, dass der input layer 108 Knoten beinhaltet, der middle layer 180 Knoten (36 Zeichen mal 5 Musterbeispiele) und der output layer besteht logischerweise aus 36 Knoten, welche die Zeichen „A-Z“ und „0-9“ repräsentieren.
- 2.) In [10] werden die Zeichen mithilfe von drei Überprüfungen klassifiziert. Zum einen wird eine normalisierte Skelett-Repräsentation des Zeichens berechnet und zusätzlich auch eine normalisierte Hintergrundmaske des Zeichens. Bei der Klassifikation wird dann beide Templates aller Zeichen mit dem aktuellen Zeichen verglichen. Das Zeichen mit dem geringsten Unterschied wird als klassifiziertes Zeichen gewählt.

3 Sensortechnik

Diese Arbeit beschäftigt sich mit der Kennzeichentafelerkennung in Bildfolgen, daher ist auch die Videoaufnahme ein wesentlicher Bestandteil der Bakkalaureatsarbeit.

Es werden verschiedene Technologien und Parameter von Kameras mit ihren Vor- und Nachteilen besprochen.

3.1 Charge Coupled Device (CCD) und CMOS Technologie

Bei der Wahl der Kamera spielt die Aufnahmetechnologie eine wesentliche Rolle, da sowohl die CCD-Technologie, als auch CMOS ihre Vor- und Nachteile haben. Das Prinzip dieser Aufnahmeverfahren basiert auf einem fotoelektrischen Effekt, wobei eine negative Platte eines lichtempfindlichen Materials mit Photonen beschossen wird und dieses Elektronen emittiert. Diese Elektronen werden so als Signal interpretiert, dass die Lichthelligkeit proportional mit der Anzahl der Elektronen steigt.

Ein Sensor empfängt auf der entsprechenden Fläche nicht immer gleich viele Photonen, vielmehr schwankt der Photonenbeschuss um ein Mittel, z.B.: etwa 50 Photonen pro Sekunde. Hierbei wäre es möglich, dass in einer Zeiteinheit 30 und in einer anderen Zeiteinheit 65 oder 70 Photonen auf den Sensor treffen.

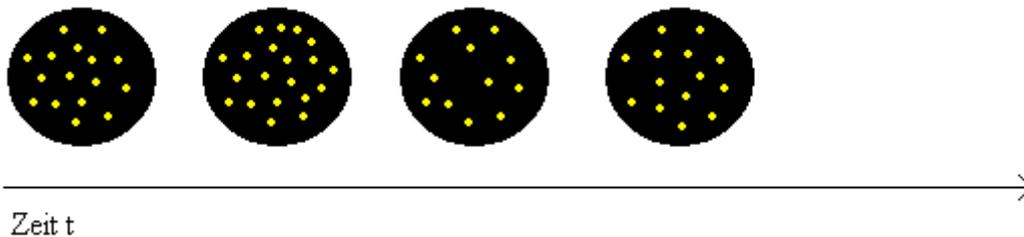


Abbildung 1: Photonenbeschuss eines Sensors zu verschiedenen Zeitpunkten gemessen.

Die Diskretisierung in ein digitales Signal wird durch einen Digital/Analog Wandler verursacht, welcher die Geometrie des Senders nutzt um einzelne Regionen zu Pixel zusammenzufassen. Der Sensor könnte zum Beispiel eine regelmäßige Rechteckige Anordnung von einzelnen Elementen besitzen, welche später als Pixel zusammengefasst werden.

3.1.1 Charge Coupled Device (CCD)

CCD-Sensoren bestehen meist aus einer Anordnung von Lichtempfindlichen Zellen, welche in Matrixform angeordnet sind. Diese Zellen repräsentieren die Pixel (picture elements), welche üblicherweise ein Ausmaß von $5\mu\text{m}$ bis $20\mu\text{m}$ Kantenlänge haben. Die Größe einer Zelle bestimmt die Lichtempfindlichkeit des CCD-Sensors. Je größer die Zelle nun ist, desto Lichtempfindlicher ist der Sensor, gleichzeitig wird die Auflösung aber kleiner.

Es gibt nun mehrere Technologien, den Auslesevorgang der Zellen zu veranlassen.

- *Full Frame-CCD*
- *Frame Transfer-CCD*
- *Interline Transfer-CCD*
- *Frame Interline Transfer-CCD*

Bei Full Frame-CCD wird der ganze Sensor, also das gesamte Bild in einem Schritt ausgelesen und übertragen. Bei Frame Transfer-CCD hingegen werden die Ladungen nach der Belichtung in einen abgedunkelten Bereich des Chips verschoben um das Bild während der Belichtungszeit auslesen zu können. Interline Transfer-CCD lesen jeweils eine ganze Zeile aus der Matrix aus und speichern sie vor dem Verschieben in den Ausleseverstärker in einer Zwischenspeicherzelle ab. Dadurch spart man hier den mechanischen Verschluss und kann die Belichtungszeit elektronisch steuern. Dadurch sind beim Interline Transfer-CCD sehr kurze Belichtungszeiten möglich.

3.2 Verschlusszeiten und Sensorrauschen

Für die automatische Kennzeichentafelerkennung in Bildfolgen ist es wichtig, dass die Qualität der aufgenommenen Bilder für eine Klassifikation noch ausreichend gut ist. Ein gutes Bild hat kein verwischtes Kennzeichen und wenig Rauschen. Das Rauschen ist durch den bildgebenden Sensor gegeben und kann mit einfachen Mitteln wie einem Median-Filter unterdrückt werden. Ein verschwommenes Bild, welches durch zu lange Verschlusszeiten der Kamera bei bewegten Objekten entsteht, kann ab einem gewissen Grad die Klassifikation unmöglich machen. Verschlusszeiten, welche bei einem sich bewegenden Auto noch Bilder liefern, die sich für eine Erkennung eignen sollen, liegen etwa im 1/100 Sekunden Bereich wenn sich das Auto mit einer Geschwindigkeit um etwa 20 km/h bewegt.

4 Kennzeichentafel Lokalisierung

Als ersten Schritt in der automatischen Kennzeichentafelerkennung muss zunächst der Bereich gefunden werden, welcher das Kennzeichen eingrenzt um in diesem Bereich anschließend weitere Analysen machen zu können. Für den Schritt der „Character-Klassifikation“ spielt die vollständige und korrekte Segmentierung des Kennzeichens eine entscheidende Rolle. Kann das Kennzeichen nicht korrekt segmentiert werden, so können die Zeichen ebenfalls nicht erkannt werden und die Klassifikation schlägt fehl. Dadurch, dass die gesamte Klassifikation in diesem Schritt oft scheitert, werden mehrere Kennzeichen-Kandidaten für die Klassifikation im Bild gesucht und diese in einem weiteren Schritt untersucht. Der Kandidat, welcher mit der höchsten Wahrscheinlichkeit ein Kennzeichen repräsentiert, wird für die weitere Klassifikation verwendet.



Abbildung 2: Eine korrekte Segmentierung des Kennzeichens (grün unterlegt)

4.1 Preprocessing

Die beschriebenen Algorithmen basieren auf der Tatsache, dass ein Kennzeichen meist eine dunkle Schrift auf einem hellen Hintergrund haben. Dadurch sind in diesem Bereich härtere Kanten als in der natürlichen Umgebung, wobei harte Kanten bedeutet, dass der Farbunterschied sehr groß ist. In Abb. 2 etwa ist der Hintergrund der Kennzeichentafel weiß und die Schrift darauf schwarz. Im besten Fall wäre das ein Grauwertunterschied (nach der Konvertierung des Farbbildes in ein Grauwertbild) von 255, wobei 0 den kleinsten Wert (Schwarz) und 255 den größten Wert (Weiß) darstellen.

Aus diesem Grund wird die Farbinformation des Bildes nicht benötigt und das Bild kann in ein Grauwertbild konvertiert werden, was zum einen die Komplexität und zum anderen die Berechnungszeit senkt. Das Konvertieren eines Farbwertes in den zugehörigen Grauwert kann wie folgt berechnet werden:

$$G = 0.3 \cdot R + 0.59 \cdot G + 0.11 \cdot B$$

Formel 1: Konversion eines RGB-Wertes in einen Grauwert

Aufgrund der Tatsache, dass es in dieser Arbeit um die Detektion von Kennzeichen in Bildfolgen geht, muss eine weitere Tatsache beachtet werden, die für Videos gilt. Ein Bild aus einer Bilddatei (.bmp, .jpeg etc.) wird prinzipiell so geladen, dass die linke obere Ecke des Bildes die Koordinaten $x=0$ und $y=0$ hat. Bei einem Video besitzt die linke untere Ecke des Bildes die Koordinaten $(0,0)$. Somit muss das Bild vor der Grauwertkonversion noch gedreht werden.



Abbildung 3: Konversion und Spiegelung eines Videoframes in ein Grauwertbild

4.2 Mittelwertfilterung

Sogenannte „rank-filter“ werden verwendet um Bereiche mit starken Kanten zu finden. Es handelt sich hierbei um nichts anderes als Mittelwertfilter, wobei der horizontale Filter breiter als hoch ist und der vertikale Filter höher als breit ist.

$$m_h = [1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1] \cdot \frac{1}{9}$$

$$m_v = [1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1]^T \cdot \frac{1}{9}$$



Abbildung 4: (a) Horizontaler und (b) vertikaler "Rank-filter" angewandt auf das aktuelle Input-Frame

4.3 Kantendetektion und Kantenprojektion

Ein Merkmal einer Kennzeichentafel sind starke Kanten innerhalb der Tafel aufgrund der schwarz-weiß Übergänge zwischen den schwarzen Buchstaben und dem weißen Hintergrund. Aus diesem Grund erfolgt die Lokalisierung des Kennzeichens mithilfe einer Kantendetektion. Zuerst werden die Kantenfilter gewählt:

$$G_X = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$$

$$G_Y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Die beiden Kantenfilter werden sind der vertikale und der horizontale Sobel-Kantenfilter. Alternativ kann auch der Prewitt Kantendetektor verwendet werden.

$$G_X = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$$

$$G_Y = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

Bei diesem Kantenfilter werden die Grauwerte in der aktuellen Gradientenrichtung nicht so wie beim Sobel-Operator extra gewichtet.

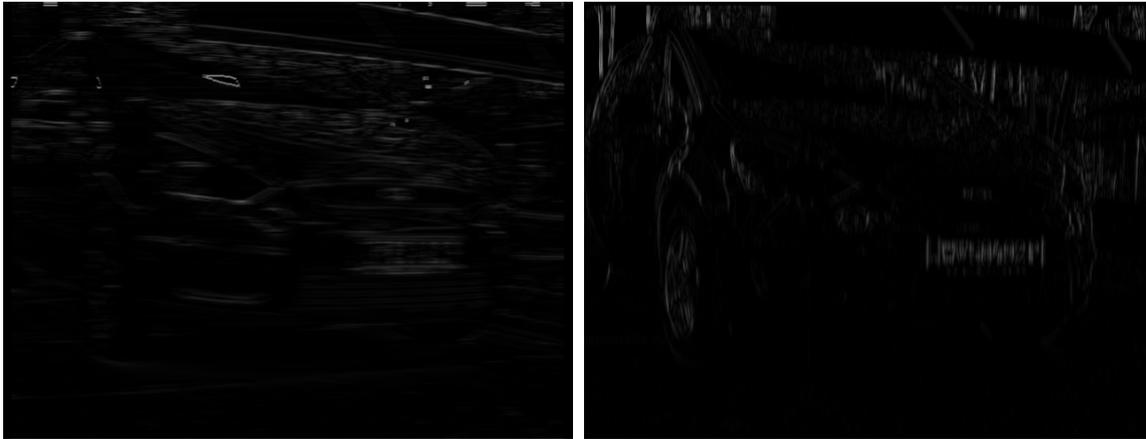


Abbildung 5: (a) Horizontale und (b) vertikale Kantendetektion mit dem Sobel Kantenoperator

Nach der Kantendetektion werden die Kanten auf die y-Achse projiziert (vertikale Kantenprojektion). Die vertikale Kantenprojektion eines Bildes kann wie folgt beschrieben werden:

$$P_y = \sum_{i=0}^{h-1} I(x, i)$$

Wobei „h“ die Höhe des Bildes ist und „I(x,i)“ der Grauwert des Bildes in der Zeile i und der Spalte x ist. Äquivalent dazu kann die horizontale Kantenprojektion folgendermaßen beschrieben werden:

$$P_x = \sum_{i=0}^{w-1} I(i, y)$$

Wobei „w“ die Breite des Bildes ist und „I(i,y)“ der Grauwert des Bildes in der Zeile i und der Spalte x ist.

Ist das Kennzeichen sehr verschmutzt, kann der Kontrast zwischen den Buchstaben und dem Hintergrund nicht ausreichend sein, um mit diesem Ansatz die Kennzeichentafel korrekt zu lokalisieren. Mit diesem Ansatz kann erkannt werden, ob ein Kennzeichen im Bild vorhanden ist oder nicht. In Videostreams ist es oft der Fall, dass in bestimmten Frames kein Auto sichtbar ist und dementsprechend auch kein Kennzeichen zu sehen ist. In diesem Fall muss keine weitere Detektion stattfinden. Um dies zu bewerkstelligen, wird der Durchschnitt des höchsten Kantenprojektionswertes berechnet und die Detektion fortgeführt, falls der aktuelle höchste Kantenprojektionswert ja nach Rauschen 10% bis 20% höher ist, als der berechnete Durchschnitt. Die Probleme dieses Ansatzes sind:

- 1.) Der Hintergrund muss eine geringere Kantenprojektion erzeugen, als die Szene mit einem Auto mit Kennzeichen
- 2.) Der Zeitraum, in welchem der Durchschnittswert berechnet wird, muss so gewählt werden, dass sich dieser nicht zu schnell verändert, wenn ein Kennzeichen für längere Zeit im Bild zu sehen ist und gleichzeitig soll sich der Durchschnittswert an Dämmerungsbedingte Änderungen anpassen können.

In den meisten Situation reicht eine Durchschnittsberechnung bzw. Erneuerung über 1-5 minuten.

4.4 Band clipping und Plate clipping

Mithilfe der Kantenprojektionen kann das Kennzeichen in zwei Phasen lokalisiert werden [7]. Im ersten Schritt wird die obere und untere Schranke des Kennzeichens mithilfe einer Schwellwertoperation eingegrenzt. Dieser Schwellwert wird aus dem Histogramm der vertikalen Kantenprojektion berechnet. Zunächst muss das Histogramm mit einem Mittelwertfilter geglättet werden um lokale Minima zu vermeiden. Der Mittelwertfilter sollte an die Anzahl der Histogramm-Pins angepasst sein und hat das gleiche Aussehen wie die Filter in Abschnitt 3.2. Anschließend wird der höchste Funktionswert des Histogramm's bestimmt:

$$Hy_{max} = \arg \max \{ P_y \}$$

Hy_{max} bezeichnet die Zeile im digitalen Bild mit der höchsten Kantenprojektion, woraus angenommen wird, dass sich an dieser Stelle ein Kennzeichen befindet. Die obere und die untere Schranke des Kennzeichens wird ausgehend von Hy_{max} folgendermaßen berechnet:

$$Hy_0 = \max \{ P_y \leq c_y \cdot P_y(Hy_{max}) \}$$

$$Hy_1 = \min \{ P_y \leq c_y \cdot P_y(Hy_{max}) \}$$

c_y ist eine Konstante mit der der Fuß des Maxima gefunden wird. Der Wert ist in praktischen Anwendungen auf 0.42 festgelegt. Je kleiner der Wert gewählt wird, desto breiter ist die gefundene Region. Dieser Ansatz wird „Band clipping“ genannt und ist ähnlich wie das „Plate clipping“, welches die horizontale Kantenprojektion verwendet um die linke und rechte Begrenzung des Kennzeichens im Bild zu finden. Dabei verwendet das „Plate clipping“ nur die Kanten im ausgewählten Bereich des „Band clipping“.

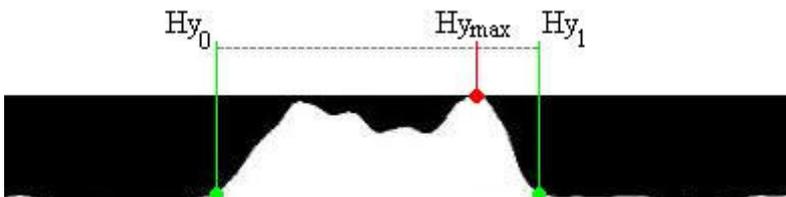


Abbildung 6: Bestimmen der Grenzen des Kennzeichens mithilfe des Histogramms aus der Kantenprojektion

Unter guten Bedingungen ist die Position des Maximums im Histogramm auch die Position des Kennzeichens im Bild. Eine gute Bedingung wäre etwa eine Szene mit homogenem Hintergrund und wenigen Kanten im Bild. In der Praxis existiert dieser Fall nicht, deswegen kommt es vor, dass der höchste Wert im Histogramm auf ein anderes Objekt im Bild verweist. Aus diesem Grund müssen mehrere Kandidaten eines Kennzeichens gefunden werden.

4.5 Kandidaten für Kennzeichen

In einem Bild können keine Kennzeichen, ein oder mehrere Kennzeichen vorhanden sein. Um den wahrscheinlichsten Kandidaten zu finden werden zwei Kriterien eines Kennzeichens verwendet. Das erste Kriterium ist das Verhältnis von Höhe zu Breite, welches in den meisten Ländern gleich ist und in etwa 1:5 ist. Das zweite Kriterium ist die Anzahl der Zeichen auf einem Kennzeichen. Eine Region mit zwei oder drei Zeichen bzw. Ziffern ist mit großer Wahrscheinlichkeit kein Kennzeichen. Die Analyse dieser Zusammenhangskomponenten wird in Abschnitt 4.3 beschrieben.

Der aktuelle Kandidat wird verworfen, wenn die Abweichung eines der Kriterien größer als ein vorgegebener Schwellwert ist. Die Auswahl der Kandidaten erfolgt wie in folgendem Pseudocode beschrieben [7].

```
FOR i = 1 to 3
  FOR j = 1 to 3
    find top and bottom limitations using vertical projection
    find left and right limitations using horizontal projection
    IF region is a number plate
      BREAK
    ELSE
      delete regions in histograms
      continue
    ENDIF
  ENDFOR
ENDFOR
```

Der beste Kandidat wird also gefunden indem sowohl in der horizontalen als auch in der vertikalen Projektion die drei wahrscheinlichsten Regionen ausgewählt (Insgesamt neun) werden und anschließend analysiert werden. Alternativ zu den drei mal drei Regionen können auch mehr Regionen gesucht werden, allerdings ist es in der Praxis nicht der Fall, dass mehr Aussagekräftige Spitzen in den Projektionen vorkommen als drei. In Abbildung 7 ist eine Situation dargestellt, in welcher in der vertikalen Projektion das Maximum auf der Höhe der Scheinwerfer liegt. Anschließend wurden die drei besten Regionen mit den höchsten horizontalen Kantenprojektionen gefunden und alle nach der Analyse verworfen, da das Verhältnis von Höhe zu Breite nicht richtig ist und es keine Zeichen oder ähnliche Zusammenhangskomponenten in diesen Regionen gibt. Der zweite Kandidat der vertikalen Kantenprojektion ist auf der Höhe des Kennzeichens und die horizontale Kantenprojektion ist hier auch so eindeutig, dass das Kennzeichen gefunden werden konnte. Diese Region wurde nicht verworfen, da genügend Zusammenhangskomponenten gefunden werden konnten und das Verhältnis von Höhe zu Breite auch im akzeptierten Bereich liegt.

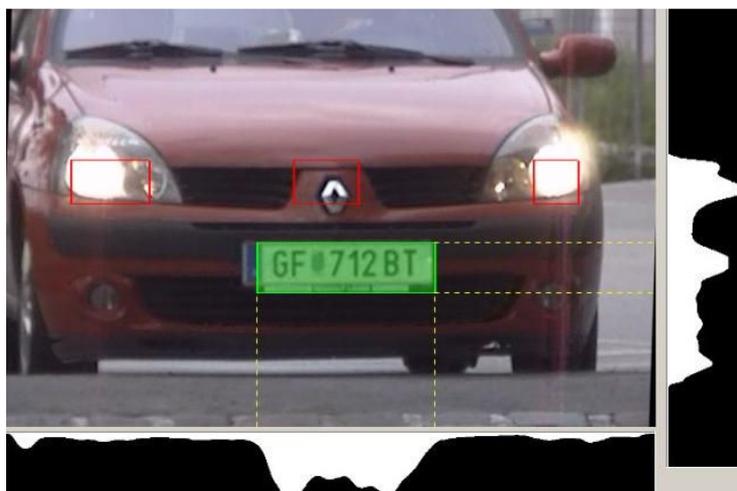


Abbildung 7: Verwurf von Kandidaten

5 Character Segmentation

Nachdem das Kennzeichen lokalisiert wurde, können die einzelnen Zeichen des Kennzeichens extrahiert und anschließend klassifiziert werden. Zunächst werden die Zeichen mithilfe einer Schwellwertberechnung vom Hintergrund getrennt und anschließend analysiert.

5.1 Normalisation der Pixelintensitäten

Die Pixelintensitäten in der Kennzeichenregion können bei einem 8-Bit Grauwertbild genau 256 Werte annehmen. Um zu bestimmen, welcher Pixel zum Vordergrund gehört und welcher Pixel zum Hintergrund, wird ein Schwellwert bestimmt, welcher die Zeichen (Vordergrund) vom Hintergrund trennen kann. Es besteht die Möglichkeit einen globalen Schwellwert für die ganze Region zu bestimmen, oder es kann auch die Region in kleinere Regionen geteilt werden und für jede Region ein Schwellwert bestimmt werden. Zwischen den Regionen wird der Schwellwert interpoliert. Der Methode des globalen Schwellwertes hat den Nachteil, dass Lichtvariationen auf dem Kennzeichen zu Fehlsegmentationen führen kann. Der Algorithmus der globalen Schwellwertbestimmung kann wie folgt beschrieben werden [7]:

- 1.) Bestimme einen Schwellwert mit dem begonnen werden soll (z.B.: $t=128$ bei 256 möglichen Werten)
- 2.) Der Schwellwert wird auf die Pixel der Region angewandt und in zwei Klassen unterteilt. Die Klasse der Pixel, deren Wert über dem Schwellwert liegt $S_a = \{[x,y] | f(x,y) > t\}$ und die Klasse der Pixel, deren Wert unter dem Schwellwert liegt $S_b = \{[x,y] | f(x,y) < t\}$
- 3.) Berechne die Mittelwerte μ_a und μ_b der beiden Klassen wie folgt:
$$\mu_a = \frac{1}{|S_a|} * \sum_{[x,y] \in S_a} f(x,y), \mu_b = \frac{1}{|S_b|} * \sum_{[x,y] \in S_b} f(x,y)$$
- 4.) Berechne einen neuen Schwellwert $t = \frac{1}{2}(\mu_a + \mu_b)$
- 5.) Wiederhole die Schritte 2,3,4 bis die Veränderung des Schwellwertes zwischen zwei Durchläufen unter einen bestimmten Schwellwert fällt.

Der gleiche Algorithmus kann für Unterregionen in der Kennzeichenregion angewendet werden. Hierbei kann der *Ansatz von Chow und Kaneko* verwendet werden, welcher die lokalen Schwellwerte der Unterregionen bestimmt und für jedes Pixel die Nachbarregionen zur gewichteten Interpolation heranzieht.

Das Resultat der beschriebenen Algorithmen zur Segmentation der Zeichen vom Hintergrund kann wie in Abbildung 8 aussehen.

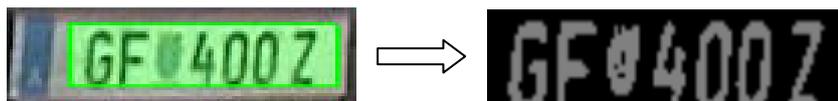


Abbildung 8: Schwellwertbildung zur Segmentation von Vordergrund zu Hintergrund

5.2 Segmentation mittels horizontaler Projektion

Aus dem berechneten Binärbild der Kennzeichenregion müssen die Zeichen einzeln behandelt werden um sie später klassifizieren zu können. Die Segmentation mittels horizontaler Projektion nutzt die Tatsache, dass zwischen den Zeichen ein Platz ist, worin kein Vordergrund vorkommt. Aus diesem Grund werden die Pixelintensitäten der Region für jede Spalte aufsummiert und auf die Horizontale Ebene projiziert. Die Platzhalter entsprechen den maxima in Abbildung 9 [7].

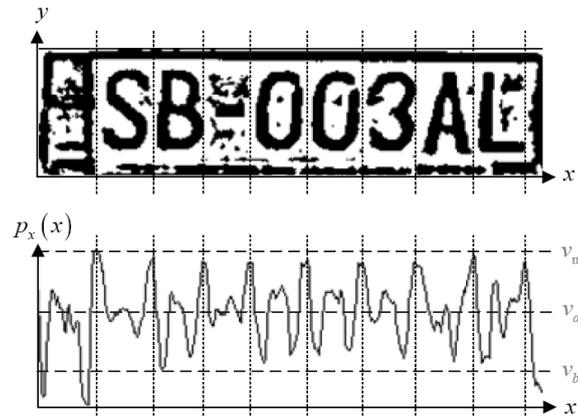


Abbildung 9: Projektion der Pixelintensitäten zur Segmentation der Zeichen [7]

Die Zeichen können demnach mithilfe der maxima getrennt und für die weitere Klassifikation als eigene Regionen verwendet werden.

5.3 Segmentation mittels Blobanalyse

Aufgrund der Tatsache, dass die Kennzeichenregion nun ein Binärbild ist, kann die Segmentation der Zeichen auch mithilfe einer Blobanalyse erfolgen. Dazu werden im ersten Schritt alle Zusammenhangskomponenten (blobs) als eigenes Objekt betrachtet. Die Detektion, welcher blob ein Zeichen ist wird mithilfe der Eigenschaften der Blobs bestimmt. Um bestimmte blobs auszuschließen, wird der Median der Größe und der Median der räumlichen Lage aller blobs berechnet und blobs verworfen, welche eine zu große Distanz zu diesen Werten haben. Abbildung 10 demonstriert der Verwurf eines Zeichens, welcher nicht den berechneten Anforderungen entspricht.

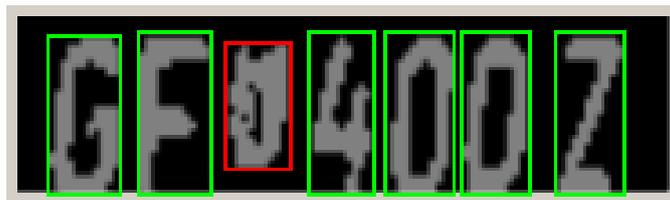


Abbildung 10: Verwurf eines blobs

Der Vorteil dieser Methode ist, dass die Blobanalyse auch bei der Kandidatenentscheidung eingesetzt werden kann indem Kandidaten verworfen werden, welche zu wenige oder zu viele Zeichen beinhalten.

6 Merkmalsextraktion und Character Klassifikation

Zur Merkmalsextraktion, für die Klassifikation der Zeichen, sollten genügend Merkmale berechnet werden um gute Separationen in den Pfaden des Entscheidungsbaumes zu erreichen [9]. Je mehr Merkmale berechnet werden, desto mehr Möglichkeiten bestehen um Zeichen zu klassifizieren. Zuvor findet ein Training der Zeichenmerkmale für das komplette Trainingsset, welches die extrahierten Zeichen der Kennzeichen selber sind, statt um geeignete Merkmale für die Separation zu finden. Es werden nicht nur absolute Werte wie die Euler-Number berechnet, sondern auch Merkmal-zu-Merkmal Relationen. Hierfür wird sogenanntes „zoning“ verwendet um lokale Merkmale aus den Zeichen zu berechnen und in Relation zu stellen. Um genügend Zonen bilden zu können, muss das Zeichen eine entsprechende Auflösung besitzen. Das Zeichen könnte etwa in zwei mal zwei Zonen unterteilt werden und in diesen Zonen würden anschließend Merkmale berechnet. Abbildung 11 demonstriert die Idee des „zoning“. Entscheidungen im Entscheidungsbaum, welcher ein binärer Baum ist, könnten als Pseudocode folgendermaßen aussehen:

Wenn Merkmal_1 aus Zone_1 > Merkmal_1 aus Zone_2 dann...

Der entstehende Entscheidungsbaum könnte die Form des Baumes in Abbildung 12 haben.

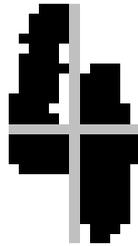


Abbildung 11: Die Zonen eines Zeichens zur lokalen Merkmalsextraktion

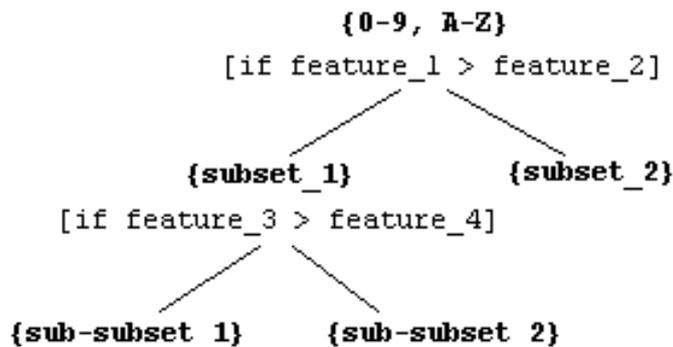


Abbildung 12: Binärer Entscheidungsbaum

Der Grund für die Entscheidungen in dieser Form von Relationen ist, dass die absoluten Werte der Zeichen sehr stark durch Rotation, Skalierung und Translation beeinflusst werden.

6.1 Merkmalsextraktion

Zur Merkmalsextraktion wurden unter anderem folgende Merkmale berechnet [8]:

- center of mass
- mean radius
- compactness
- width to height relation
- euler number
- signature

Es gilt, je mehr Merkmale berechnet werden, desto besser können die Zeichen im allgemeinen Fall separiert werden. Die Anzahl der Merkmale bestimmt die Zeit des Trainings dahingehend, dass das Training länger dauert, je mehr Merkmale verwendet werden. Eine Merkmalsselektion sollte deswegen vor dem Training durchgeführt werden um Merkmale zu finden, welche keine Separation von Zeichen ermöglichen kann. Ein Merkmal ist dann überflüssig bzw. es kann dann Zeichen nicht separieren, wenn der Wert des Merkmals für alle verwendeten Zeichen sehr ähnlich ist.

Es wurde weiters eine Signatur der Zeichen berechnet welche den Abstand des Randes zum ersten Randpixel des Zeichens in Pixel misst. Die Zeichen werden vorher in eine Einheitsgröße normalisiert (z.B.: in 8x8 Pixel) um die gleiche Länge der Signatur für alle Zeichen zu erreichen. Diese Signatur kann ebenfalls als Merkmal verwendet werden oder es können auch nur einzelne Punkte der Signatur verwendet werden.

6.2 Entscheidungsbaum als Heuristik

Der Entscheidungsbaum teilt das Zeichenset in kleinere Zeichensets dementsprechend auf, wie sie durch Merkmale am besten zu trennen sind. Wie in Abbildung 12 zu erkennen ist, steht im obersten Knoten das komplette Zeichenset, welches in den darauffolgenden Ebenen kleiner wird. Eine Entscheidung ist dann gefällt, wenn der Entscheidungsbaum in ein Blatt mündet, welches aus einem Zeichenset von genau einem Zeichen besteht. Durch das Training der Merkmale kann man erfahren, welche Relationen von Merkmalen das Zeichenset in welcher Weise separiert. Das Ergebnis einer Entscheidung könnte etwa sein, dass fünf Zeichen zu ungefähr 99% der Entscheidung entsprechen, sieben Zeichen zu ungefähr 99% der Entscheidung nicht entsprechen und fünf Zeichen der Entscheidung zu 40%, 50% oder 60% entsprechen. In diesem Fall würden die ersten fünf Zeichen in den rechten Zweig münden, die zweiten sieben Zeichen in den linken Zweig des Entscheidungsbaumes kommen und die restlichen fünf Zeichen in beide Zweige kommen. Dies ist eine Gegebenheit, die durch die Trennbarkeit der Zeichen entsteht. Je größer das Zeichenset ist, desto unwahrscheinlicher wird es, dass eine Merkmal-zu-Merkmal Relation das komplette Zeichenset so trennen kann, sodass alle Zeichen zu 95% oder mehr in einen Zweig kommen. Falls bei einer Entscheidung 95% eines Zeichens der Entscheidung entspricht und es nur in den rechten Zweig aufgenommen wird, dann entsteht bei dieser Entscheidung ein Verlust von 5% Entscheidungsgenauigkeit. Diese 5% Verlust sind in diesem Fall auf das Trainingsset gerechnet, im Falle eines großen Trainingsset ist im Realfall der Verlust aber etwa gleich groß. Es ist also der Verlust pro Entscheidung zu minimieren wodurch es oft dazu kommt, dass Zeichen in beide Zweige aufgeteilt werden. Der Binärbaum wird dadurch größer.

7 Verwendung der temporären Information

Die Klassifikation der Zeichen erfolgt in mehreren Schritten. Im Gegensatz zur einmaligen Klassifikation in Bildern, können in Bildfolgen Analysen über die Zeit gestellt werden, welche eine bessere Klassifikation garantieren. Für diesen Ansatz ist die Methode der einzelnen Klassifikation des Kennzeichens nicht von Bedeutung, wodurch es möglich ist, dass jeder Ansatz in Kombination mit dieser Analyse zu verwenden ist.

Die Klassifikation besteht aus folgenden Schritten:

- 1.) Aus einem Bild (Frame) des Videostreams wird das Kennzeichen erkannt und jedes darin befindliche klassifizierte Zeichen gespeichert. Diese Klassifikation wird die „single classification“ genannt.
- 2.) Im gleichen Frame wird eine Analyse der bereits vorhandenen Klassifikation durchgeführt. Für jede „single classification“ wird die Länge des Kennzeichens gespeichert und anschließend mit den Längen der anderen Kennzeichen des gleichen Autos verglichen. Es wird nun der Median der Längen aller vorhandenen Klassifikationen gebildet und diejenigen Kennzeichen berücksichtigt, welche eben diese Länge haben. Dadurch haben alle zu analysierenden Kennzeichen die gleiche Länge. Im nächsten Schritt wird für jede Stelle des Kennzeichens der Median der klassifizierten Zeichen berechnet. Das Ergebnis bestimmt die Endgültige Klassifikation des Zeichens an dieser Stelle. Das Resultat dieser Analyse wird die „Temporal Classification“ genannt.
- 3.) Im letzten Schritt werden die temporären Klassifikationen als ganzes analysiert und eine endgültige Entscheidung getroffen. Dafür werden alle temporären Klassifikation herangezogen, welche im Zeitfenster der Erkennung, welches in Abschnitt 3.3 erklärt ist, klassifiziert wurden und das Kennzeichen als finale Entscheidung gewählt, welches das größte Gewicht in der Liste der Klassifikationen erreicht. Die Gewichtung erfolgt entsprechend der Anzahl der „single classifications“. Diese finale Entscheidung wird pro Kennzeichen nur ein mal durchgeführt und zwar genau am Ende dieses Zeitfensters.

Ein Beispiel dieses Vorganges wird in Tabelle 1 demonstriert.

Single Classification	Median der Kennzeichenlänge	Median aller Zeichen	Temporal Classification
128A	$L(128A) = 4$ 4	<u>128A</u> 128A	128A
128A	$L(128A) = 4$ $L(128A) = 4$ 4	<u>128A</u> 128A 128A	128A
28A	$L(128A) = 4$ $L(128A) = 4$ $L(28A) = 3$ 4	<u>128A</u> 128A 128A	128A
12BA	$L(128A) = 4$ $L(128A) = 4$ $L(28A) = 3$ $L(12BA) = 4$ 4	<u>128A</u> 128A 128H 128A	128A
128H	$L(128A) = 4$ $L(128A) = 4$ $L(28A) = 3$ $L(12BA) = 4$	128A 128A 128H 128H	128A

	L(128H) = 4 4	128A	
12B	L(128A) = 4 L(128A) = 4 L(28A) = 3 L(12BA) = 4 L(128H) = 4 L(12B) = 3 4	128A 128A 128H 128H 128A	128A
12B	L(128A) = 4 L(128A) = 4 L(28A) = 3 L(12BA) = 4 L(128H) = 4 L(12B) = 3 L(12B) = 3 4	128A 128A 128H 128H 128A	128A
12B	L(128A) = 4 L(128A) = 4 L(28A) = 3 L(12BA) = 4 L(128H) = 4 L(12B) = 3 L(12B) = 3 L(12B) = 3 4	128A 128A 128H 128H 128A	128A
128	L(128A) = 4 L(128A) = 4 L(28A) = 3 L(12BA) = 4 L(128H) = 4 L(12B) = 3 L(12B) = 3 L(12B) = 3 L(128) = 3 3	28A 12B 12B 12B 128 12B	12B
128A	L(128A) = 4 L(128A) = 4 L(28A) = 3 L(12BA) = 4 L(128H) = 4 L(12B) = 3 L(12B) = 3 L(12B) = 3 L(128) = 3 L(128A) = 4 4	128A 128A 128H 128H 128A 128A	128A

Tabelle 1: Beispiel einer Kennzeichenerkennung mittels Bildfolgenanalyse

Tabelle 1 zeigt die Robustheit des Ansatzes an einem Beispiel mit schlechter „single classification“. Im Realfall sind die einzelnen Classification noch besser und die Analyse wird noch eindeutiger entschieden.

Die finale Entscheidung wird anhand dieses Beispiels wie folgt berechnet:

- Kandidaten aus den temporären Klassifikationen: 128A und 12B
- Gewichte anhand der Länge der Statistik (Anzahl der „single classifications“)
 - 128A = 1+2+3+4+5+6+7+8+10 = 46
 - 12B = 9

Somit ist die finale Entscheidung in diesem Fall „128A“ mit einem Gewicht von 46. Der zweitbeste Kandidat hat ein Gewicht von 9, was einer Entscheidungseffizienz von ~83.63% entspricht, wobei nur 30% aller Frames eine korrekte „single classification“ erhalten.

8 Tests und Ergebnisse

Der Algorithmus wurde auf eine zwölf Stunden lange Videosequenz angewendet und die einzelnen Klassifikationen wurden manuell bewertet. Als Detektionsgenauigkeit wurden folgende Maße verwendet:

- Detektionsrate der vorhandenen Kennzeichen (ist ein Kennzeichen im Bild, oder nicht)
- Detektionsrate der einzelnen Zeichen
- Detektionsrate der finalen Klassifikation

Die Detektionsrate der Lokalisierung wurde nicht gemessen, da sie durch die Bildfolgen schwer zu messen ist. In vielen Fällen stimmt die Lokalisierung des Kennzeichens nur in einzelnen Frames nicht, wodurch die Klassifikation nur leicht beeinflusst wird. Die Ergebnisse der Detektionsraten sind in Tabelle 2 aufgelistet.

Detektionsrate der vorhandenen Kennzeichen	Detektionsrate der einzelnen Zeichen	Detektionsrate der finalen Klassifikation
100% (200 von 200)	~97.5%	95% (190 von 200)

Tabelle 2: Verschiedene Detektionsraten

Die Detektionsrate der einzelnen Zeichen wurde auf etwa 300.000 Zeichen gemessen, welche aus der zwölf Stunden langen Videosequenz extrahiert und manuell annotiert wurde. Das Training der Zeichen wurde auf einem Intel Quad Core 2.6 GHz durchgeführt und dauerte etwa 2 Wochen um den gesamten Entscheidungsbaum zu berechnen. Die Detektionsrate der einzelnen Zeichen beträgt in etwa 97.5%, wobei die meisten Fehler bei Zeichen wie „0“ und „O“ bzw. „8“ und „B“ passieren. „0“ und „O“ sind aus verschiedenen Winkeln fast identisch und können daher nicht in jedem Frame korrekt erkannt werden. Die Detektionsrate hier liegt bei etwa 85% für beide Zeichen. Ähnliches gilt für „8“ und „B“, wobei hier das Hauptproblem die Segmentierung ist. Ist ein „B“ verschmutzt, ist der segmentierte Blob meist runder als wenn es ohne Verschmutzung extrahiert wäre. Dadurch ähnelt dieses Zeichen der „8“ auch sehr stark. Die Klassifikationsraten liegen hierbei bei etwa 90% bei beiden. Die restlichen Zeichen haben eine Detektionsrate von durchschnittlich 99%. Die Detektion ob ein Kennzeichen im aktuellen Frame sichtbar ist, war in allen getesteten Fällen erfolgreich. Die Detektionsrate der finalen Klassifikation liegt bei 95% (190 von 200 Autos).

Die Performance des Algorithmus liegt bei ~300 fps (frames per second) wenn kein Kennzeichen sichtbar wird und etwa 100-250 fps wenn eines sichtbar ist. Die Performance bei Erkennen eines Zeichens variiert aufgrund der Anzahl der gefundenen Kandidaten, welche analysiert werden. Die Performance wurde auf einem Intel Pentium Centrino 1.83 GHz mit 2 GByte RAM getestet, wobei maximal 100 MByte gebraucht wurden.



Abbildung 13: Falsche Klassifikation durch verdrehtes Kennzeichen

In Abbildung 13 ist eine temporäre Klassifikation zu sehen, welche durch ein verdrehtes Kennzeichen falsch klassifiziert wurde. Das Problem liegt hier an der Segmentation der Zeichen und nicht selbst an der Zeichenklassifikation. Abbildung 14 zeigt eine korrekte Klassifikation trotz schwerer Lichtverhältnisse. Die „single classification“ könnte in solchen Fällen von der richtigen Zeichenfolge abweichen, durch die statistische Analyse ist die temporäre Klassifikation jedoch korrekt.



Abbildung 14: Korrekte Klassifikation trotz schwerer Lichtverhältnisse

In Abbildung 15 ist eine Klassifikation zu sehen, welche innerhalb von fünf bis zehn frames zum richtigen Ergebniss konvergiert. Die falsche Klassifikation ist hierbei auf die Auflösung der vergangenen Frames zurückzuführen, in welchen die Zeichen der Kennzeichentafel zu klein sind um eine robuste Klassifikation zu garantieren. In diesem Fall wird ein Parameter eingesetzt, welcher die minimale Größe der Zeichen beschränkt um die Klassifikation zu verbessern.



Abbildung 15: Konvergierende Klassifikation innerhalb weniger Frames

Abbildung 17 zeigt eine korrekte Klassifikation trotz der Verdeckung der Person, welcher zuvor das Kennzeichen komplett verdeckt hatte. Die Statistik der Klassifikationen in den einzelnen Frames wurde hier nicht gelöscht, aufgrund der Tatsache, dass die Kantenprojektion über dem Durchschnittswert bleibt (siehe Abschnitt 3.3).



Abbildung 16: Korrekte temporäre Klassifikation



Abbildung 17: Korrekte temporäre Klassifikation



Abbildung 18: Korrekte temporäre Klassifikation trotz falscher Lokalisierung

In Abbildung 18 wird ein Kennzeichen nicht richtig lokalisiert, da es einen anderen Kandidaten gibt, welche Informationen von Zeichenähnlichen blobs enthält. Die Klassifikation findet statt, aber sie fällt nicht schwer ins Gewicht, da zu diesem Zeitpunkt schon einige „single classifications“ vorhanden sind. Abbildung 19 zeigt die Konvergenz eines Kennzeichens, welches Anfangs Fehlklassifiziert wurde aufgrund des „O“ anstatt der „0“.

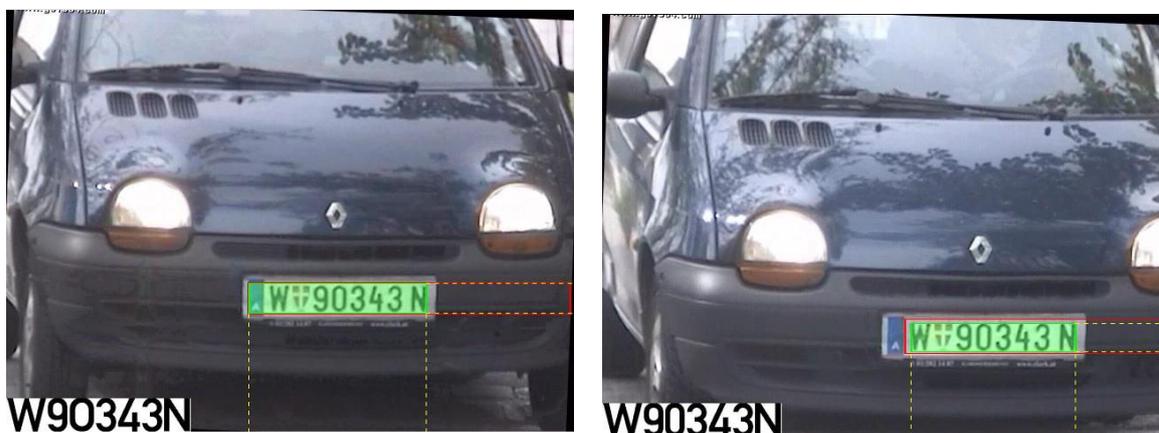


Abbildung 19: Konvergierende Klassifikation eines Kennzeichens

9 Zusammenfassung und zukünftige Verbesserungen

In dieser Arbeit ist ein neuer Ansatz für Kennzeichentafelerkennung in Video Streams vorgestellt worden. Die Erkennung findet nicht so wie zuvor in nur einem Frame statt, sondern es werden kontinuierlich so viele Frames klassifiziert, wie der Algorithmus verarbeiten kann. Die einzelne Klassifikation kann auch mit einem anderen Ansatz verwendet werden, allerdings ist der Algorithmus besser, je mehr Frames für ein Kennzeichen verarbeitet werden können, da damit die Anzahl der „single classifications“ steigt und die temporäre Klassifikation gegen das richtige Kennzeichen konvergiert.

Zukünftige Verbesserungen sollen in die Richtung von Länderunabhängigen Klassifikationen gehen. Die Idee ist, dass für jedes Land Kennzeichen trainiert werden und ein eigener Entscheidungsbaum pro Land erstellt wird. Zusätzlich wird ein gemeinsamer Entscheidungsbaum für alle Zeichen aller Länder erstellt. Im ersten Schritt wird für das Kennzeichen das zugehörige Land bestimmt, aus welchem das Kennzeichen stammt. Dies geschieht anhand der klassifizierten Zeichen aus dem allgemeinen Entscheidungsbaum, wobei jenes Land zugewiesen werden soll, welches die meisten Vorkommen im Kennzeichen besitzt. Anschließend wird der Entscheidungsbaum des Landes eingesetzt, welches vorher bestimmt wurde.

In einem Diagramm zusammenfassend kann die Arbeit abstrakt wie in Abbildung 20 beschrieben werden.

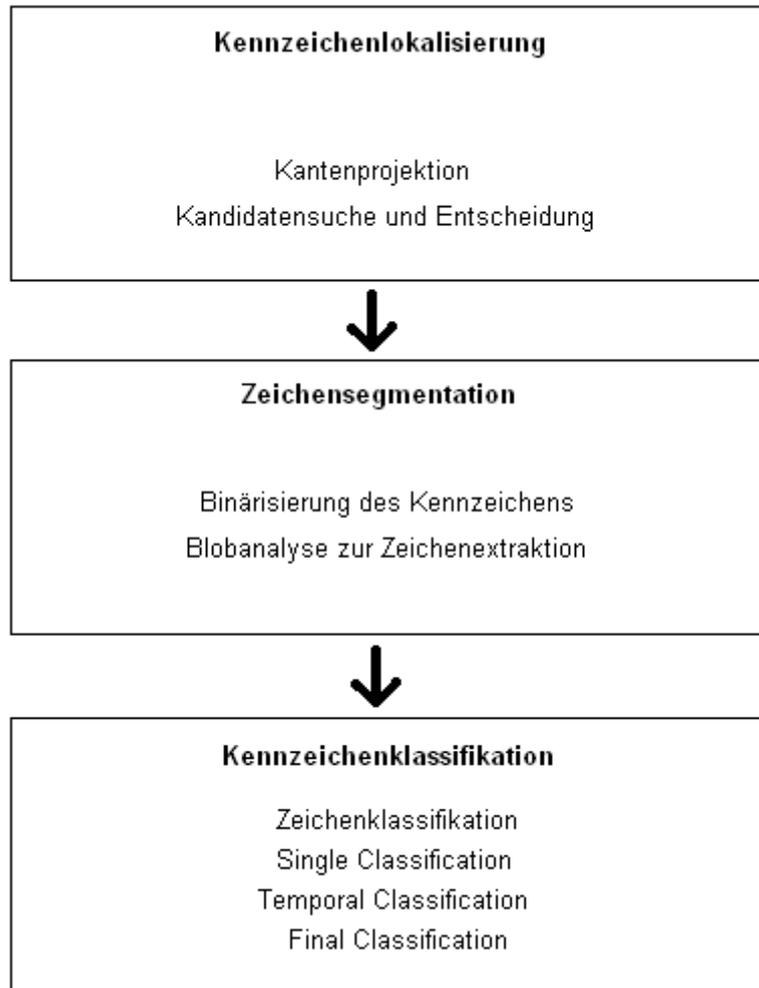


Abbildung 20: Zusammenfassung der Algorithmen

10 Referenzen

- [1] C.N.E. Anagnostopoulos, I.E. Anagnostopoulos, V. Loumos, and E. Kayafas. **A license plate-recognition algorithm for intelligent transportation system applications.** *Intelligent Transport Systems*, 7(3): pages 377-392, September 2006.
- [2] Gonzalez R., Woods R.: **Digital Image Processing**, Prentice Hall, Upper Saddle River, New Jersey, 2007
- [3] Jianmin Xu Hui Fu Zhong Qin, Shengli Shi. **Method of license plate location based on corner feature.** In *World Congress on Intelligent Control and Automation*, pages 8645-8649. IEEE, 2006.
- [4] Kim H., Kim J., **Region-based shape descriptor invariant to rotation, scale and translation**, *Signal Process. Image Commun.* 16 (2000) pages 87–93.
- [5] Maury Mitchell, Matthew Hudnall, David Brown, David Cordes, Randy K. Smith, and Allen S. Parrish. **A host architecture for automobile license plate recognition.** In *Intelligence and Security Informatics*, pages 87-94. IEEE, 2007.

- [6] Mei Xie Feng Yang, Zheng Ma. **A novel approach for license plate character segmentation.** In *International Conference on Industrial Electronics and Applications*, pages 1-6. IEEE, 2006.
- [7] Ondrej Martinsky, **Algorithmic and Mathematical Principles of Automatic Number Plate Recognition Systems**, B.SC Thesis, Brno 2007
- [8] Peura M., Iivarinen J., **Efficiency of simple shape descriptors**, in: *Proceedings of the Third International Workshop on Visual Form*, Capri, Italy, May, 1997, pages. 443–451.
- [9] Siti Norul Huda Sheikh Abdullah, Marzuki Khalid, Rubiyah Yusof, and Khairuddin Omar. **Comparison of feature extractors in license plate recognition.** In *AMS '07: Proceedings of the First Asia International Conference on Modelling & Simulation*, pages 502-506, Washington, DC, USA, 2007. IEEE Computer Society.
- [10] Vladimir Shapiro, Georgi Gluhchev, and Dimo Dimov. **Towards a multinational car license plate recognition system.** *Mach. Vision Appl.*, 17(3): pages 173-183, 2006.
- [11] Xiao-Feng Chen, Bao-Chang Pan, Sheng-Lin Zheng, **A License Plate Localization Method based on Region Narrowing**, In *Proceedings of the Seventh International Conference on Machine Learning and Cybernetics, Kunming, 12-15 July 2008*, pages 2700-2705, 2008
- [12] Zhang Y., Zhang C.: **New Algorithm for Character Segmentation of License Plate**, *Intelligent Vehicles Symposium*, IEEE, 2003

Abbildungsverzeichnis

Abbildung 1: Photonenbeschuss eines Sensors zu verschiedenen Zeitpunkten gemessen.....	5
Abbildung 2: Eine korrekte Segmentation des Kennzeichens (grün unterlegt)	6
Abbildung 3: Konversion und Spiegelung eines Videoframes in ein Grauwertbild	7
Abbildung 4: (a) Horizontaler und (b) vertikaler "Rank-filter" angewandt auf das aktuelle Input- Frame	8
Abbildung 5: (a) Horizontale und (b) vertikale Kantendetektion mit dem Sobel Kantenoperator.....	9
Abbildung 6: Bestimmen der Grenzen des Kennzeichens mithilfe des Histogramms aus der Kantenprojektion	10
Abbildung 7: Verwurf von Kandidaten	11
Abbildung 8: Schwellwertbildung zur Segmentation von Vordergrund zu Hintergrund.....	12
Abbildung 9: Projektion der Pixelintensitäten zur Segmentation der Zeichen [7]	13
Abbildung 10: Verwurf eines blobs	13
Abbildung 11: Die Zonen eines Zeichens zur lokalen Merkmalsextraktion	14
Abbildung 12: Binärer Entscheidungsbaum	14
Abbildung 13: Falsche Klassifikation durch verdrehtes Kennzeichen.....	19
Abbildung 14: Korrekte Klassifikation trotz schwerer Lichtverhältnisse.....	19
Abbildung 15: Konvergierende Klassifikation innerhalb weniger Frames	20
Abbildung 16: Korrekte temporäre Klassifikation.....	20
Abbildung 17: Korrekte temporäre Klassifikation	21
Abbildung 18: Korrekte temporäre Klassifikation trotz falscher Lokalisierung.....	21
Abbildung 19: Konvergierende Klassifikation eines Kennzeichens	22
Abbildung 20: Zusammenfassung der Algorithmen.....	23