Die approbierte Originalversion dieser Diplom-/Masterarbeit ist in der Hauptbibliothek der Technischen Universität Wien aufgestellt und zugänglich.
http://www.ub.tuwien.ac.at

The approved original version of this diploma or master thesis is available at the main library of the Vienna University of Technology.
http://www.ub.tuwien.ac.at/eng

FAKULTÄT
FÜR !NFORMATIK

Faculty of Informatics

Unraveled Skeletons

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Visual Computing

eingereicht von

Maximilian Langer, BSc
Matrikelnummer 01226991

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: O.Univ.Prof. Dipl.-Ing. Dr.techn. Walter G. Kropatsch

Wien, 29. April 2019

Maximilian Langer
Walter G. Kropatsch

Unraveled Skeletons

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Visual Computing

by

Maximilian Langer, BSc
Registration Number 01226991

to the Faculty of Informatics

at the TU Wien

Advisor: O.Univ.Prof. Dipl.-Ing. Dr.techn. Walter G. Kropatsch

Vienna, 29th April, 2019

Maximilian Langer

Walter G. Kropatsch

Erklärung zur Verfassung der Arbeit

Maximilian Langer, BSc
Alois Pummer-Gasse 7, 2345 Brunn am Gebirge

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 29. April 2019

Maximilian Langer

Acknowledgements

My great thanks goes to Professor Walter Kropatsch for his excellent supervision and to all people of PRIP I had inspiring talks with, namely Daniel Pucher and Ines Janusch. I also want to thank my family for supporting me throughout my studies.

Kurzfassung

Diese Diplomarbeit erarbeitet eine neue Repräsentation für zweidimensionale Objekte, genannt Unraveled Skeletons. Sie ist dafür konstruiert robust in Bezug zu Transformationen zu sein, im speziellen gegenüber artikulierten Bewegungen. Die Eigenschaften der Repräsentation wurden studiert und mögliche Applikationen analysiert.

Die Repräsentation basiert auf dem Skelett oder Medial Axis Transform des Objekts. Sie benutzt die sehr robuste Voronoi Methode, um das Skelett zu generieren und „läuft“ dann entlang des Skeletts um das Skelett herum. Bei jedem Punkt wird die minimale Distanz zum Umriss des Objekts in einen Vektor gespeichert. Das Resultat ist eine Liste (Shape signature). Diese Liste (Unraveled Skeleton Vector) kann anschließend für weitere Bearbeitung, zum Beispiel Normalisierung, herangezogen werden.

Diese Arbeit untersucht zwei mögliche Einsatzgebiete für Unraveled Skeletons: Shape Analysis und Posen-unabhängige Koordinatensysteme. Da das Unraveled Skeleton unabhängig gegenüber artikulierten Bewegungen ist, ist es geeignet für Shape Analysis oder Objekterkennung, wenn sich diese Objekte artikuliert bewegen (zum Beispiel Tiere). Verschiedene Normalisierungs- und Optimierungs-Strategien wurden erarbeitet. Außerdem kann das Unraveled Skeleton dazu benutzt werden, Teile von Objekten zu erkennen.

Das Posen-unabhängige Koordinatensystem weist jedem Punkt im Objekt und auf seinem Umriss eine eindeutige Koordinate zu, unabhängig von der Pose oder artikulierten Bewegung. Gemeinsam mit dem nächsten Punkt am Skelett und der Distanz wie sie vom Unraveled Skeleton bereitgestellt werden, können konvexe Regionen adressiert werden. Mit einer dritten Zahl kann jeder Punkt dieser Regionen eindeutig bestimmt werden, die Unraveled Skeleton Koordinaten bestehen daher aus 3 Komponenten. Diese Arbeit beschäftigt sich mit drei möglichen Anwendungen für diese Koordinaten: Shape Blending, Super-Resolution und Verbesserung von Segmentationen. Non-centered Skelette werden ebenfalls behandelt.

Alle Resultate wurden auf mehreren Datensätzen evaluiert.

Abstract

This thesis presents a new shape representation called Unraveled Skeleton. It is designed to be robust to a large number of transformations, especially articulated movement. Based on this new representation, its general properties are studied and its application domain established.

The representation is based on the skeleton or medial axis of a shape. It uses the very robust Voronoi skeletonization and "walks" around the resulting skeleton tree. At every point the minimum distance to the boundary is saved to a vector, resulting in a list of distance measures, a shape signature. This Unraveled Skeleton vector can then be used for further processing like normalization.

This thesis explores two possible directions to use the Unraveled Skeletons: Shape analysis and pose independent coordinate systems. In shape analysis the Unraveled Skeleton vector can be used for articulation independent recognition. For this purpose a number of different normalization and optimization techniques are introduced. It is also possible to use parts of the vector to match parts of objects.

The pose independent coordinate system assigns every point inside the shape and on its boundary a unique coordinate independent of pose or articulated movement. Using the closest points on the skeleton and its distance as provided by the Unraveled Skeleton one can address convex patches. Given a third component to chose a point from this patch a coordinate with three components is introduced. This paper states three applications to use the Unraveled Skeleton coordinates: Shape Blending, Super-resolution and Segmentation refinement. It also explores possibilities of non-centered skeletons in regard to Unraveled Skeleton coordinates.

All results are evaluated on multiple datasets.

Contents

Kurzfassung **ix**

Abstract **xi**

Contents **xiii**

1 Introduction **1**

2 Unraveled Skeletons **3**
2.1 Medial Axis Transform and Skeletons 4
2.2 Voronoi Skeletons 6
2.3 Distance Transform 10
2.4 Unraveled Skeletons 11
2.5 Normalization 12
2.6 Holes 14
2.7 Properties 14

3 Shape Analysis **19**
3.1 Shape Recognition 19
3.2 Shape Description 20
3.3 Graph and Vector Matching 22
3.4 Normalization and Optimization 22
3.5 Part Matching 27

4 Pose Independent Coordinates **29**
4.1 Shape Coordinates 29
4.2 Unraveled Skeleton (US) Coordinates 31
4.3 Non-centered Skeletons 35
4.4 Applications 36

5 Evaluation **39**
5.1 General 39
5.2 Shape Analysis 42
5.3 Pose-independent Coordinates 47

6 Conclusion **55**

Acronyms **57**

Bibliography **59**

CHAPTER 1

Introduction

This thesis introduces a new shape representation based on the skeleton and investigates its possible use for shape analysis and as pose-independent coordinate system. A skeleton, first introduced by Blum [3], is a reduction of a shape's dimensionality that retains information about the structure of the shape. The skeleton has been popular since its introduction for various tasks in computer vision and computer graphics: Object description, retrieval, manipulation, matching, registration, tracking, recognition, compression and part segmentation [27] to mention just a few.

The advantage of the skeleton representation of an object is the reduction in dimension and the stability under different transformations like scaling and rotation. Another reason for the widespread use of skeletons is that they do not change the topology of the shape and additionally preserve the structure and spatial relationship of object components.

The structural preservation capabilities of the skeleton make it an excellent choice to represent shape sequences, where the shape performs an articulated movement. The articulated movement is a deformation of the shape that retains multiple rigid parts connected by joints, for example a walking animal. Related to the articulated movement is the pose problem, i.e. the change of a shape extracted from different poses. In recognition, a good descriptor can relate same objects in different poses.

Representation and recognition of articulated shapes has been studied in research [22, 14], but it remains a challenging problem. The main challenge is to find a representation that can handle the pose problem and still distinguishes between different objects. The focus of the research lies in the recognition of articulated shapes and not much work has been done for the registration, i.e. uniquely addressing points on the shape across poses.

The aim of this work is to study a new shape representation (US) with its properties and applications. To enable comparison with state-of-the-art approaches, the shape representation is analyzed for recognition purposes in greater detail. Apart from the representation for recognition, various comparison methods are discussed. Additionally to recognition, the registration that is sparsely addressed in the literature (pose independent coordinates) is the second focus of this work. Pose independent coordinates unlock new ways to handle well-known problems in computer vision, like segmentation and super-resolution. In combination with registration the notion of non-centered skeletons is studied to be able to handle articulated movement even better.

The following are the contributions in this thesis:

- Proposal of a new shape representation (US) based on the skeleton and distances to the boundary.
- Description of the properties and applications of proposed shape representation.
- Normalization of proposed shape representation for different use cases.
- Matching and recognition techniques for proposed shape representation including matching parts of a shape.
- Proposal of a new pose independent coordinate system based on the US.
- Evaluation of proposed shape representation in recognition and as pose independent coordinate system.

The remainder of the thesis is structured as follows: In Chap. 2 first an introduction to shape representations with a focus on various invariances is given and skeletonization is introduced. Then the basic idea and properties of the US are presented. Chapter 3 looks at the US from the point of view of a shape analysis or shape recognition. First a look at related work is given, then matching of US representations is described. The chapter takes a closer look at normalization and optimization for a good matching and introduces part matching with the US. Pose independent coordinates are introduced in Chap. 4. The first sections give an overview of state-of-the-art techniques, then pose independent coordinates using the US are described. This chapter also takes a look at non-centered skeletons. The chapter finishes with a description of applications for US pose independent coordinates. Chapter 5 evaluates different aspects of the US, first in a general sense, then regarding shape analysis and finally regarding pose-independent coordinates. The thesis is concluded in Chap. 6.

CHAPTER 2

Unraveled Skeletons

Many computer vision tasks - such as classification or segmentation - require structural and geometric information of shapes. It is therefore an ongoing topic for research to extract features from images and shapes and represent them in a way that is detailed enough to distinguish different shapes, but general enough to group similar shapes together. For classification and clustering purposes the difference (or distance) of features should be minimized in the same class and maximized between different classes.

For in-class minimization invariances play an important role. In natural images and shapes various factors exist, which should not influence the calculated feature. For example, in most cases it should be irrelevant at which resolution an image is taken, so a feature should not rely on the absolute size of an object. In the ideal way, a feature derived with a procedure R that is invariant to a change N yields the same feature output: $R(S) = R(N(S))$, with S being the Scene [17].

According to Yang et al. the following invariances are desirable for a shape descriptor [35]:

Translation, Scale and Rotation These transformations do not change the shape at all. The resulting feature output should therefore be the same regardless of the position, size or rotation of the shape.

Affine transformation A transformation is considered affine if parallel lines are still parallel after the transformation. It is a combination of translation, scaling, rotation, flips (negative scaling) and shears.

Noise Noise on the boundary and inside the object should not change the feature output. This is important to avoid errors based on bad segmentation.

Occlusion Partly occluded shapes should give similar feature outputs for visible parts.

Additional to those invariances, pose invariance is desirable for shape descriptors that deal with shapes that may employ articulated movement, like humans or animals.

Definition 1. A shape D is constructed of multiple rigid parts R_j where points inside a rigid part do not move relative to each other from timestep t_i to timestep t_{i+1} :$\forall p_1, p_2 \in R_j : d(p_1(t_i), p_2(t_i)) = d(p_1(t_{i+1}), p_2(t_{i+1}))$, where $d(p_1, p_2)$ is the Euclidean distance between points p_1 and p_2. An articulated movement happens if two or more rigid parts share a common point c and local angles between parts change.

Rigid parts sharing a common point "rotate" around this point. One can think of articulated movement as two parts connected by a joint. For pose invariance it is not important what pose a shape is in, e.g. a horse running or jumping. In most real situations an articulated movement will introduce small deformations around the joint: compression on the inside and expansion on the outside.

In this chapter a new shape representation and descriptor is introduced that is invariant to translation, scale, rotation, noise and articulated movement.

2.1 Medial Axis Transform and Skeletons

The representation presented in this chapter relies on skeletons. In this section we review the Medial Axis Transform (MAT) and skeletons.

The Medial Axis Transform was first introduced by Blum [3]. He describes the MAT as the locus of all points where grassfire fronts meet. A grassfire considers the shape as a field of grass that is simultaneously lit from the boundary and the fire travels at constant speed. Skeleton and MAT is mostly used interchangeably in literature [30]. The definition given here is based on the one given by Bai et al. [1].

Definition 2. Given a shape D with boundary ∂D, the set $Tan(s) = \{b \in \partial D | d(b, s) \leq d(b', s), \forall b' \in \partial D\}$ contains for point $s \in D$ the closest points on the boundary. The degree $deg(s) = |Tan(s)|$ gives the number of closest points[1] for a point s. The *skeleton* $S(D) = \{s \in D | deg(s) \geq 2\}$ is a set containing all points with degree greater than one. Points in $Tan(s), s \in S$ are called the *generating* or *tangent points* of a skeleton point s.

These skeleton points also correlate with centers of maximal inscribable disks in the shape. A maximal inscribable disk is a disk confined inside the shape, that is not completely inside any other inscribable disk. The skeleton can be described - and was in fact by Blum - by the locus of the centers of all maximal inscribable disks. This furthermore gives each skeleton point a distance to the boundary based on the radius of the inscribed disk (see Sect. 2.3).

---

[1] $|X|$ denotes the cardinality of set X.

Definition 3. A point $s \in S$ can fall in exactly one of the following categories: s is an *endpoint* if it has only one skeleton point neighbor, s is a *branch* or *joint point* if $deg(s) > 2$ and s is not an endpoint and s is a *common point* if $deg(s) = 2$. The union of common points connecting two branch points, two endpoints or a branch and an endpoint is called a *skeleton edge* or branch.

The definitions given above describe the skeleton in a continuous space, but in most cases computer vision is concerned with discrete representations (e.g. images). When confined to the discrete grid, there exist different solutions for a skeleton. Therefore the term skeleton sometimes gives more liberty regarding the centeredness to improve other desired properties such as simplicity.

2.1.1 Skeletonization Approaches

Literature provides hundreds of skeletonization algorithms that solve the problem for different domains and applications. This section only gives a short overview, the interested reader is referred to the survey by Saha et al. [27] for more information.

A skeleton produced by a skeletonization algorithm should have the following properties [15]:

1. The skeleton must be thin, i.e. in images it must be one pixel wide.
2. The skeleton must approximate the medial axis.
3. Thin curves and endpoints must be preserved.
4. Connectivity of foreground and background must be preserved, i.e. the algorithm does not break the topology of the shape.

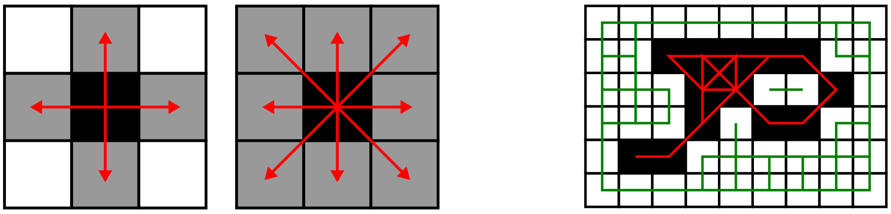
Skeletonization algorithms can be grouped into three categories [27]: Continuous geometry approaches, Continuous boundary evolution and discrete approaches using morphology or distance transform.

An approach that is widely used by different algorithms is to simulate Blums grassfire on the discrete grid with different rules when to delete pixels. As an example that uses that technique, the parallel thinning algorithm by Zhang and Suen [39] is given as it is still used an modern computer vision libraries (e.g. scikit[2]). It considers the foreground to be in an 8-connected neighborhood and the background in a 4-connected neighborhood (See Fig. 2.1). Since each pixel is considered in parallel the algorithm uses two subiterations to avoid removing a two pixel wide line at once (and thus breaking topology). In a first subiteration contour points are deleted which fulfill conditions in the

---

[2] `https://scikit-image.org/docs/dev/api/skimage.morphology.html#skeletonize` (Accessed: 13/03/19)

local 3x3 neighborhood. In a second subiteration the conditions are changed slightly. The algorithm erases pixels first from the lower right, then from the upper left. The algorithm trades centeredness for simplicity and robustness to boundary noise. Figure 2.2a shows example skeletons produced by the algorithm and illustrates a problem: The algorithm is not scale and rotation invariant. Unfortunately, this is a problem not easy to solve due to the grid based nature of the approach.



(a) 4-connected neighborhood (b) 8-connected neighborhood (c) If the foreground is 8-connected, the background must be 4-connected.

Figure 2.1: 4- and 8-connected neighborhoods.

A different problem is illustrated in Fig. 2.2b: The algorithm is not invariant to noise on the boundary. This problem can however be addressed by pruning. Pruning is a technique where branches that are generated by noise - so called *spurious branches* - are removed by using a global shape measure [27]. Pruning however is a challenging task and can lead to topology breakage if done incorrectly. A promising pruning approach was proposed by Bai et al. [1]. It first partitions the contour into multiple parts and then removes skeleton points if its generating points belong to the same part. The quality of the pruned skeleton is highly dependent on the chosen parts. Bai et al. show that Discrete Curve Evolution (DCD) yields a promising partition and that any partition preserves topology. Unfortunately, their approach needs a fixed number of parts for DCD.

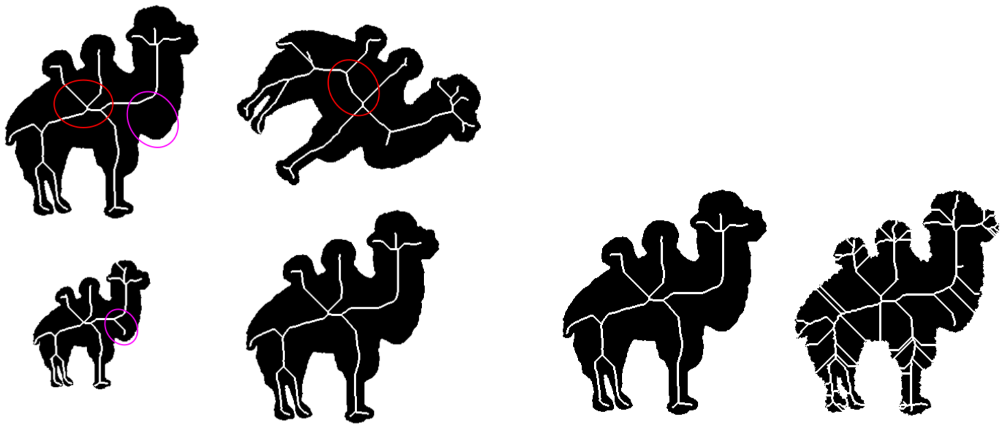
2.2 Voronoi Skeletons

A different skeletonization algorithm uses a Voronoi Diagram of the contour points [25]. It is very robust to noise and invariant to Translation, Scale and Rotation. Since these are the properties that are needed for the proposed representation, this is the skeletonization approach of choice and it will be explained in more detail.

2.2.1 Voronoi Diagram

A definition based on the one given by Costa and Cesar [7]:

Definition 4. A Voronoi Diagram (VD) or Voronoi Tessellation is a partition of $\mathbb{R}^2$ into N regions $R_i, i \in \{1, \ldots, N\}$, called *cells*, deduced from N points p_i, called *sites*. Each point $p \in R_i$ is closer to p_i then to any other point $p_j, j \neq i$, $R_i = \{p \in \mathbb{R}^2 : d(p_i, p) \leq d(p_j, p), j \in \{1, \ldots, N\}\}$. Points that are closest to two sites define the *Voronoi edges* or



(a) Influence of Translation, rotation and scale.

(b) Shape with noise

Figure 2.2: Skeletons created by algorithm of Zhang and Suen [39]. Skeleton thickness increased for illustration purposes.

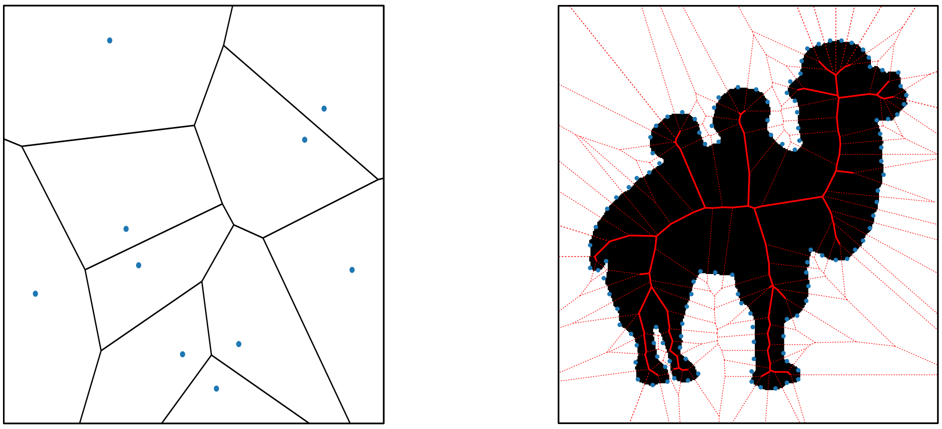
Voronoi boundaries. A Voronoi edge has two *generating sites*. *Voronoi vertices* are points connecting Voronoi edges and have three or more closest sites.

Voronoi edges contain only points that have the same distance to two or more points, which in turn is also the property of the skeleton. Figure 2.3a shows the VD of 10 random points. Figure 2.3b shows the VD of hand picked points on the boundary of a shape. Bold lines are all lines where generating sites do not lie next to each other on the contour. These bold lines already show that the skeleton is closely related to parts of the VD. By constructing the Line Voronoi Diagram (LVD), Lee [21] shows that all Voronoi edges that do not lead to concave boundary points are in fact part of the medial axis.

2.2.2 Residuals and Voronoi Pruning

Figure 2.4 shows what happens to the approach above if all contour points are considered sites. Even though the contour is smooth, discretization introduced enough noise to create many spurious branches. The pruning method introduced by Ogniewicz and Ilg [25] takes their observation into account that the shortest distance on the boundary between the generating points of a Voronoi edge is a good indicate for its importance to the skeleton. This property was also used in the last approach, where Voronoi edges with neighboring generating points were omitted.

Based on their observations they assigned each Voronoi boundary point a *Residual* value. Ogniewicz and Ilg distinguish four different Residual values that can be used to prune the skeleton (Fig. 2.5):



(a) VD for 10 random points. (b) VD for hand picked contour points.

Figure 2.3: Example of Voronoi Diagrams.

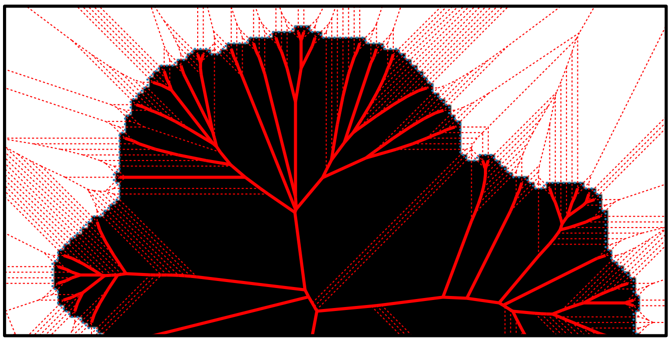


Figure 2.4: VD of camel head with all contour points as seed points.

Potential Residual The potential residual ΔR_P of a Voronoi edge e is the shortest distance on the boundary between the generating points p_A, p_B of e. This can be efficiently calculated by pre-computing a boundary potential function $W(p)$. This function assigns for every point p the distance along the boundary to a common, arbitrary origin point on the boundary. ΔR_P can than be computed:

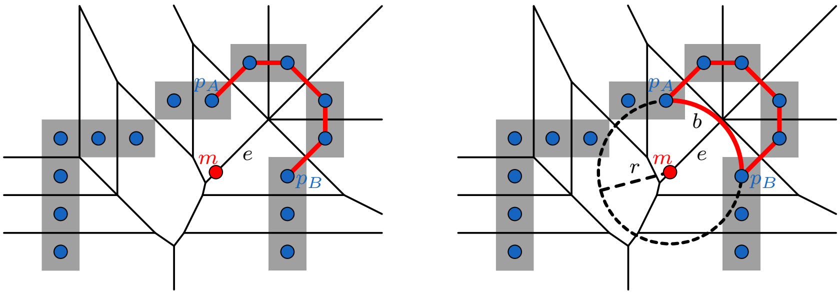
$$\Delta R_P(e) = min\{|W(p_A) - W(p_B)|, L_B - |W(p_A) - W(p_B)|\} \tag{2.1}$$

where L_B is the total length of the boundary. If p_A and p_B are on different boundaries (e.g. on a hole and on the outer boundary), $\Delta R(e)$ is infinity. Figure 2.5a shows the potential residual.

Circular Residual To show how good a skeleton point m on an edge e approximates the boundary the circular residual ΔR_C subtracts the circular arc b with center m between the generating points from the potential residual: $\Delta R_C(e, m) = R_P(e) - b$. See Fig. 2.5b for an illustration.

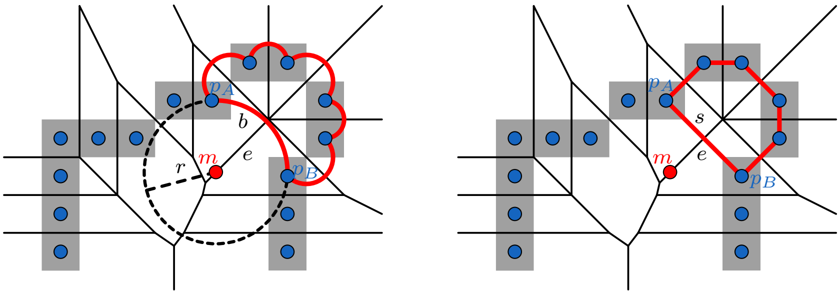
Bicircular Residual The bicircular residual ΔR_B considers boundary parts as semicircles. It is aimed to overcome shortcomings of the circular residual (See Ogniewicz and Ilg [25] for more details). ΔR_B is given as: $\Delta R_B(e,m) = \frac{2}{\pi}(\frac{\pi}{2}\Delta R_P(e) - b)$. The factor $\frac{2}{\pi}$ ensures that for edges in the center ($b << R_P(e)$) the residual functions behave similar. Figure 2.5c illustrates the bicircular residual.

Chord Residual The chord residual ΔR_H replaces the circular arc by the chord length $s = d(p_A, p_B)$: $\Delta R_H(e) = R_P(e) - s$. See Fig. 2.5d for an illustration.



(a) Potential residual

(b) Circular residual



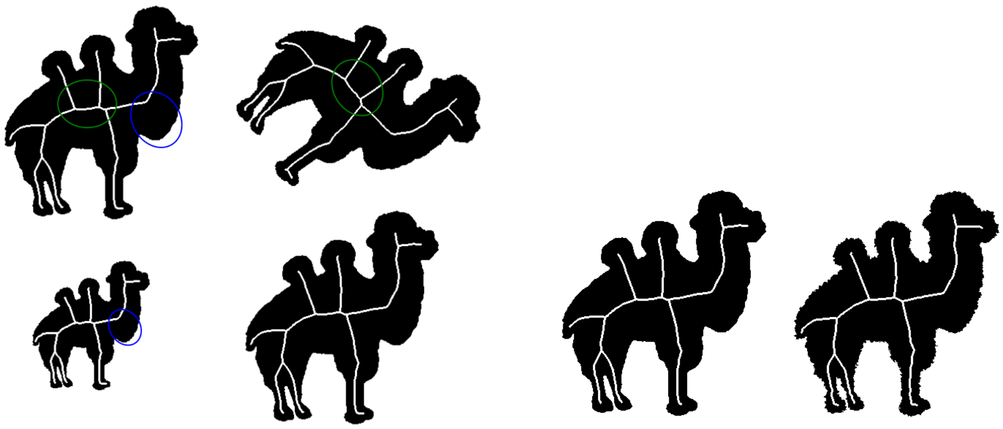
(c) Bicircular residual

(d) Chord residual

Figure 2.5: Illustration of the four residual functions (in red).

After residual computation for every point on the Voronoi boundaries the pruning is a simple matter of thresholding on the residual value. Ogniewicz and Ilg give a formula to compute threshold values for different residuals based on the size of noise on the boundary. If a is the length of a raster crack (1 or $\sqrt{2}$ for pixel grids) and n is the number of raster cracks that noise can protrude, an optimal threshold for the circular residual is $na(2-\sqrt{2})$. They also show that all their residual prunings preserve topology. See their work for proofs on thresholds and topology preservation [25].

Figure 2.6 shows Voronoi Skeletons. One can see that in difference to Fig. 2.2 the skeleton is invariant to translation, rotation and scale. Additionally the noise can be handled more elegantly by increasing the threshold.

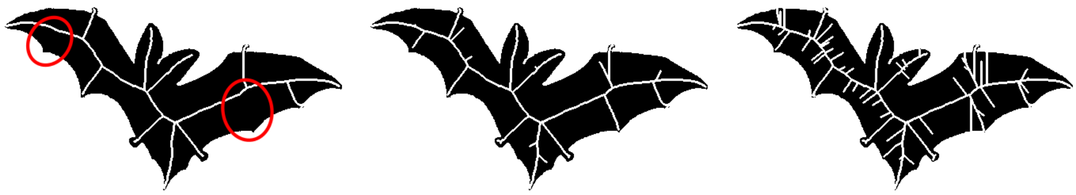


(a) Influence of Translation, rotation and scale.

(b) Shape with noise, threshold is 10

Figure 2.6: Skeletons created by Voronoi Skeletonization using circular residual. Skeleton thickness increased for illustration purposes.

Figure 2.7 shows the same shape skeletonized with different residual functions and the suggested thresholds by Ogniewicz and Ilg. It can be seen, that the circular residual gives the cleanest skeleton, but misses parts of the wings. For the rest of the work if not otherwise mentioned the circular residual is used.



(a) Circular Residual

(b) Bicircular Residual

(c) Chord Residual

Figure 2.7: Skeletons created by Voronoi Skeletonization using different residual functions. Skeleton thickness increased for illustration purposes.

2.3 Distance Transform

It is already clear from Blum's definition of the MAT that there is a relation between the distance of points in a shape to the boundary and the skeleton. The following definition is based on the one given by Klette [17].

Definition 5. Given a shape D the Distance Transform (DT) assigns every object point the closest distance to a non-object point:

$$DT(p) = \min_{q \in \mathbb{R}^2 \setminus D} \{d(p,q)\} \tag{2.2}$$

A disk centered at an object point p with radius $DT(p)$ is totally contained in the object, therefore the point with the highest DT in a shape lies on the skeleton. Figure 2.8 shows the DT on an elephant shape. The right plot shows a carpet plot of the DT. One can see that the ridges of the plot correlate with the skeleton.

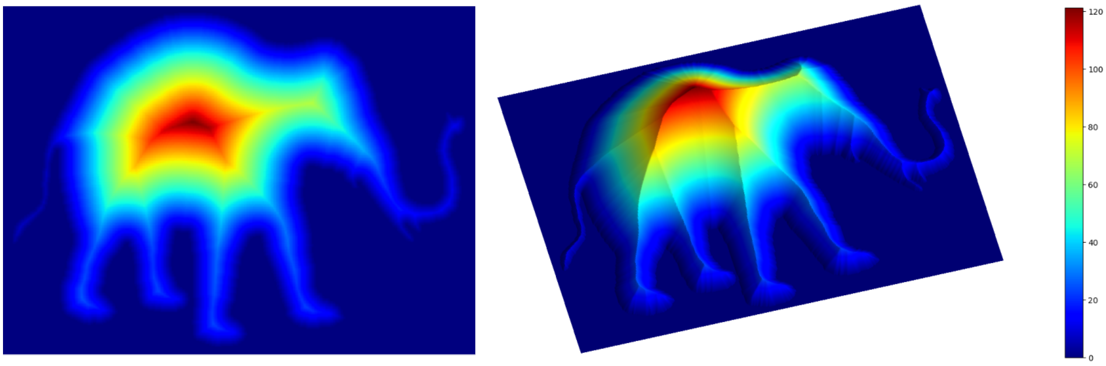


Figure 2.8: Distance transform of an elephant shape.

2.3.1 Shape Reconstruction from Skeleton

Given an unpruned skeleton S and the DT at skeleton points, the whole shape can be perfectly reconstructed by the union of circles with center $s \in S$ and radius $DT(s)$. For pruned skeletons the reconstruction looses high frequencies on the boundary. This can be used to get a more general shape of an object.

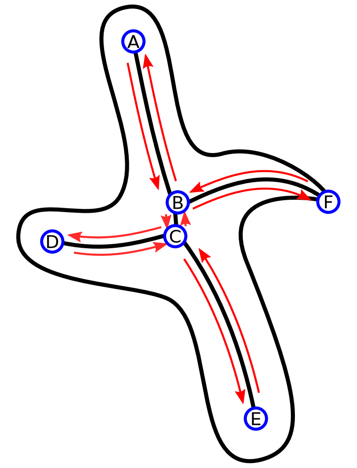
2.4 Unraveled Skeletons

The skeleton preserves topology of a shape and the DT provides geometrical information. Using a combination of these two properties by building a connectivity graph was proposed by Zaboli et al.[36]. They use graph matching to cluster shapes.

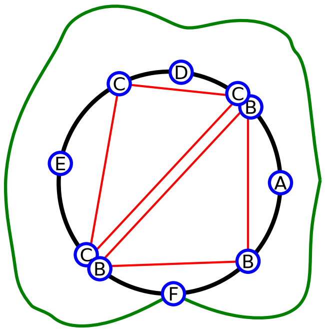
The Unraveled Skeleton (US) is a feature vector that represents the shapes thickness at various points of the skeleton. Together with the skeleton it can reconstruct the shape. The US-Vector is constructed by walking along the skeleton and writing its DT into the vector. At every branching point one turns right following a new branch and at every endpoint one turns back along the branch one came from. In the end one will come back to the starting point. This is only possible for a skeleton tree (the object has no holes).

The DT at a skeleton point $s \in S$ is also the radius function $R(s)$ at s. $R(s)$ is defined as the radius of the maximal inscribed disk at s.

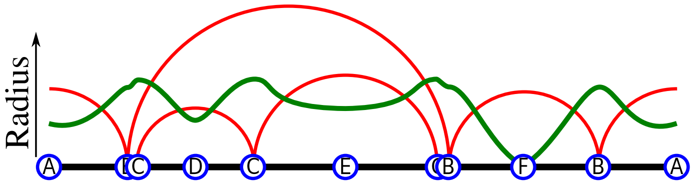
Figure 2.9 shows the unraveling process. Starting from node A the unraveling process continuous to node B, then C and so forth till it reaches node A again. The DT values are stored at the specific position. If plotting against a circle one can see that it is possible to map the shape boundary to polar coordinates. We call Figure 2.9b the Unraveled Skeleton Circle representation. The red lines and arcs in Figures 2.9b and 2.9c connect the same branching points. $US(D, S)$ gives the US vector on the shape D and skeleton S.



(a) Example shape and unraveling process



(b) Circle representation of US



(c) Unraveled Skeleton Vector

Figure 2.9: Illustration of unraveling of the skeleton and creation of feature vector.

2.4.1 Reconstruction

A reconstruction is possible if the skeleton and the US vector is given. The skeleton delivers topological and the US geometrical properties of the shape. Additionally, the skeleton provides the embedding in the plane.

2.5 Normalization

For many approaches using the US a normalized vector is desirable. Since the US is cyclic the starting point s_0 plays a miner role, but linear representation needs a robust point s_0. Here an overview is given. See Sect. 3.4 for more details on how to enhance the normalization and optimization to define distances between US vectors.

2.5.1 Eccentricity Transform

The Eccentricity Transform (ET) comes from the field of path-finding in graphs and was adopted for binary images by Kropatsch et al. [19]:

Definition 6. The eccentricity transform gives for every point $p \in D$ in a binary shape the longest minimum-distance to any other point $q \in D$:

$$ET(p) = \max_{q \in D} \{\lambda(\pi_\alpha(p, q))\} \tag{2.3}$$

where $\pi_\alpha(p, q)$ gives the shortest α-connected path in the shape between p and q and $\lambda(\pi_\alpha(p, q))$ is the length of the path $\pi_\alpha(p, q)$.

The ET is lowest in the middle of the shape and robust to deformation, as well as translation and rotation.

2.5.2 Starting Point and Orientation

A good starting point s_0 is the branching point with the lowest ET. It is the robustest point of the skeleton and does only change if there are multiple branching points with similar ET. For the shape D in (2.3) only all endpoints have to be considered with $\pi_8(p, q)$ being the distance along the skeleton S. ET can than be checked for branching points only.

After choosing s_0, a direction to start moving from the branching point must be found. Since the ET of all branching points is already known, the next node to visit is the adjacent one with the next smaller eccentricity. If there is only one branching point, the endpoint with biggest eccentricity is chosen.

Starting point and orientation with graph based centrality measures is also possible [6].

2.5.3 Length Normalization

There exist two different ways one can normalize the length of the US vector. In the first way the whole vector is simply scaled to a fixed length vector with interpolation of values. For perfect skeletons and highly articulated movement this normalization is perfect, because the skeletons inner proportion should be the same (e.g. a scaled shape). See Sect. 3.4 for advanced normalization techniques.

2.5.4 Amplitude Normalization

Amplitude normalization is used for scale independence. The feature vector is normalized between 0 and 1, where the maximal radius is mapped to 1 and 0 to 0 linearly. This ensures proper scaling of different objects. If using amplitude normalization the maximum value is needed for reconstruction.

2.6 Holes

The shape to be represented by US must not have holes, as this would introduce cycles in the skeleton graph. Depending on the application, holes play a differently important role and the need might arise to represent holes too. For completeness some solutions to the problem are suggested here.

The simplest solution to allow holes to be represented is to process them individually resulting in an US vector for the shape and every hole. The disadvantage of this approach is that the number of feature vectors for a single shape varies.

A different approach is to break the cycles in the graph in a deterministic manner. Because the ET has given robust results for determination of the starting point the following approach for breaking cycles is suggested: Calculating the ET for every branching point contained in a cycle. The branching point with the smallest eccentricity is then split into two points. This will introduce one new endpoint. The other point might downgrade to a common point. If not all cycles have been resolved proceed to the branching point with the next smallest ET in a cycle.

The splitting of a branching point can be done in multiple ways. The following rules guarantee a deterministic split:

- The branch becoming an endpoint (endpoint branch) has to be included in a cycle.
- The endpoint branch is the one where the split introduces the smallest increase in the skeletons diameter, i.e. the maximum eccentricity of any point in the skeleton. This can be achieved by taking the endpoint branch where the endpoint has the smallest eccentricity in the splitted graph.

Following these splitting rules the skeleton diameter is always the smallest possible.

2.7 Properties

Here some properties of the US representation are given.

2.7.1 Invariances of the US Representation

It is obvious that the US is highly dependent on the skeletonization as spurious branches would change the feature vector drastically. Therefore, for the representation to be invariant to translation, rotation, scale, noise and articulation, the skeleton must be invariant to those transformation too. The following presumes that the skeleton is invariant to translation, rotation, scale, flip and boundary noise. This is true for the Voronoi skeleton with an appropriate threshold.

Translation invariance The US is translation invariant, as the radius function is invariant to translation too.

Rotation invariance The US vector that is normalized in regard to starting point and orientation is rotation invariant. If considered as a cycle any US is rotation invariant. The DT using Euclidean distance is rotation invariant.

Scale invariance The length- and amplitude-normalized US vector is uniform scale invariant, non-uniform scale can lead to skeleton change and normalization change.

Flip invariance The US is not flip/mirror invariant, as the resulting US vector is also mirrored. The mirror shape is like walking left instead of right at branching points. In the case this invariance is important for a use case, two vectors have to be created.

Noise invariance The US is not invariant to noise, due to the different radius functions. The error introduced by noise on the boundary is of the same magnitude as the noise. Figure 2.10 illustrates a noisy boundary and the reconstruction using the given skeleton (black) and US (green). The US orientates itself on the noise that is closer to the skeleton, as the circles cannot penetrate to outlying boundary points. The closer the skeleton to the boundary, the better the noise is approximated.

Invariance to articulated movement If the skeleton is invariant to articulated movement, the US is too, as the radius functions of the skeleton and the skeleton do not change.

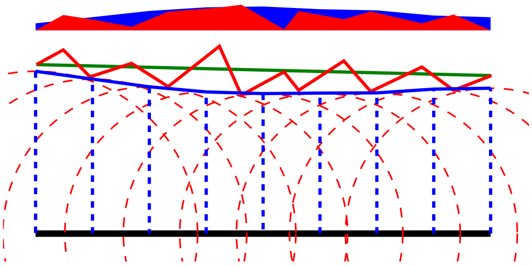


Figure 2.10: Affection of boundary noise on the US. The skeleton is black, the correct boundary green, the noise boundary red and the reconstruction by US blue. The plot on top shows the error.

2.7.2 Branching, Endpoints and Common points

Branching points P_B appear at least three times in the US vector, common points P_C twice and endpoints P_E only once. The final vector will contain about twice as many values as the skeleton has points. The exact size of the final vector is dependent on the number of branching points and endpoints: $|US(D,S)| = \sum_{p \in S(D)} deg(p)$, where $deg(p)$ is the degree of point p, as defined in Def. 3.

Theorem 2.7.1. *In an Unraveled Skeleton Circle representation lines connecting multiple occurrences of the same branching or endpoint will never cross each other. Furthermore is every circular arc of the US Circle between two occurrences of the same branching or endpoint a connected subtree of the original shape skeleton. If this subtree is removed from the original skeleton, the remaining skeleton is still a connected tree.*

Proof. The same occurrence of a skeleton point p on the US Circle (p_1, p_2) indicates that a removal of all points on one side of p will leave all points on the other, as the skeleton S is a tree structure. When we split a tree into two parts, we get two trees.

The first statement follows directly from the other two. A crossing of lines means that the two generated trees would overlap. That cannot happen unless p_1, p_2 do not belong to the same skeleton point. ☐

2.7.3 Uniqueness

The US vector is unique for every shape. Only articulated movement or rotation of a shape can produce the same US vector. This is because the symmetry in the vector indicates the length of the skeleton branches. The only exception are sections of the US vector that have the same value, as then the symmetry property cannot be used to reconstruct the skeleton branch length.

There exist only one exception where two different shapes produce the same US vector. This is a constant radius function on the skeleton over multiple, successive endpoints.

2.7.4 Symmetry

The US vector skeleton reflects mirror and rotational symmetry of a shape. Two parts of the shape are mirror symmetrical if two connected parts of the cyclic US vector are mirror symmetrical. The shape is rotation symmetrical if the same pattern repeats multiple times in the cyclic US vector. Note that symmetry in this case does allow for articulated movement and the detected symmetry might only be obvious in one pose of the shape. This allows for detection of symmetry in articulated shapes (e.g. a human).

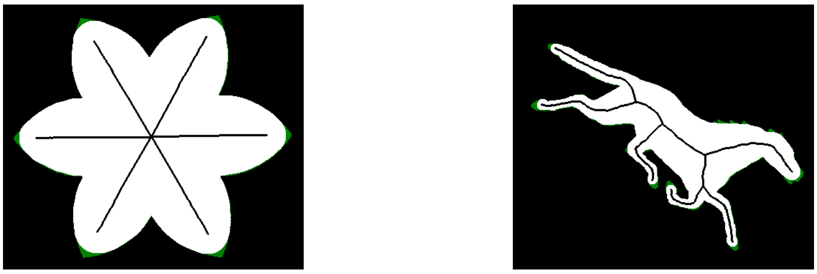
2.7.5 Structure Encoding

The local symmetry in the US vector reflects the structure of the skeleton. as soon as this symmetry around a point is broken, the symmetric section corresponds to a branch and its end to branching points. By not considering this section in the new symmetry consideration internal branches can be detected.

The correct reconstruction of the structure by the US vector is not possible in all cases. The best example is a shape that has a skeleton with same radius function everywhere. The resulting US vector has then the same value everywhere and is not distinctive.

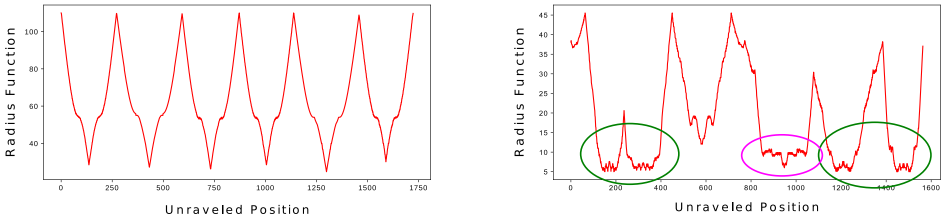
2.7.6 Examples

In Figure 2.11 two example US vectors from shapes from the MPEG7 dataset are given. Note the symmetry of the flower paddles and the similarity of the horse legs (green circles) compared to the tail (magenta circle). The figures show the skeleton and reconstruction (white) as well as the reconstruction error in green.



(a) Example 1 with skeleton and reconstruction error (green)

(b) Example 2 with skeleton and reconstruction error (green)



(c) Non-normalized US vector of Example 1

(d) Non-normalized US vector of example 2

Figure 2.11: Two example shapes with their unraveled skeleton (bottom), skeletons, reconstruction (white) and reconstruction error (green). The original shape is the combination of green and white areas.

CHAPTER 3

Shape Analysis

Shape is an important feature for human's visual perception to recognize objects [33]. Costa and Cesar [7] organize shape analysis into three steps: shape pre-processing, shape transformation and shape classification. This chapter is most concerned about the second and third step and assumes a correct pre-processing (segmentation). Based on Costa and Cesar's terminology the Unraveled Skeleton (US) is a shape descriptor. Additionally, this chapter deals with shape classification and its subcategories that need some kind of distance or similarity measure between shapes. This chapter reviews State-of-the-art shape analysis and matching based on contours and considers different distance calculation approaches for the US vector.

3.1 Shape Recognition

Shape recognition is an area of pattern recognition that concerns itself with the classification of shapes. Supervised shape classification, hence called classification, and unsupervised shape classification, hence called clustering, are similar shape recognition techniques that rely on the (dis-)similarity of features between (dis-)similar shapes. In both tasks *labels* are assigned to shapes. The difference between classification and clustering is how labels are assigned. The following definitions are adapted from Theodoridis and Koutroumbas [32]:

Definition 7. In classification or supervised learning a *classifier* is constructed by known shape-label matchings of example shapes (*ground-truth*). The classifier divides the space of all shapes into multiple regions, each region is associated with one label. In most cases the classifier is first trained on these example shapes (*training set*) and then tested against unknown shapes. This process is called *learning* or *training* a classifier.

Definition 8. Clustering or unsupervised learning assigns a shape to a class or cluster without known ground-truth. It therefore learns a grouping of similar shapes and assigns a common label.

The goal of shape retrieval is to find similar shapes to a provided one. This is closely related to clustering. For both clustering and shape retrieval a similarity measure is needed. As such a measure a metric is desirable [18]:

Definition 9. A metric is a function $d(x, y)$ with $x, y \in X$, X is some set, if the following is satisfied:

1. $d(x, y)$ is positive definite: $d(x, y) \geq 0, \forall x, y \in X$ and $d(x, y) = 0 \Leftrightarrow x = y$.
2. $d(x, y)$ is symmetric: $d(x, y) = d(y, x), \forall x, y \in X$
3. $d(x, y)$ fulfills the triangle inequality: $d(x, y) \leq d(x, z) + d(z, y), \forall x, y, z \in X$

3.2 Shape Description

Shape description is the extraction of shape properties. Simple properties of shapes are for example its area, perimeter or diameter. Zhang and Lu [38] categorize shape descriptors into two groups: region-based and contour-based features. Each of these groups is then divided into structural and global features. For example the Medial Axis Transform (MAT) is a structural region-based feature.

3.2.1 Contour-based features

Structural contour-based features break the boundary into smaller parts with properties depending on the method. An example is the approximation of the contour by a polygon or a chaincode. As a chaincode is also a good possibility to store the skeleton, a short explanation is given: Chaincodes were first proposed by Freeman [12] and encode a curve by walking along it and storing the required steps. In an 8-connected neighborhood, there are eight possible directions to go at any time. The Freeman chaincode stores the directions in a list. Giving the starting point position the contour can be reconstructed.

Global contour-based features are constructed from the boundary as a whole. Simple descriptors are for example the perimeter or the ratio between perimeter and area. An important global contour-based descriptor for this work is the shape signature, as the US falls into this category. A shape signature represents the shape by a single feature vector based on the boundary. An example is the centroid distance, that maps the boundary points on a feature vector based on the points distance to the shapes centroid, which is similar to the US.

Polar Representation

Bernier and Landry [2] introduced a shape descriptor based on the distance of contour points to the centroid. Instead of taking values along the contour, they send out rays from the center and map the boundary to the range 0 to 2π. This process produces a function that might have multiple values for non-convex shapes.

Euclidean and Geodesic Distance Profiles

Janusch et al. [16] proposed a method similar to the US vector. They select a point on the medial axis as reference point (e.g. the centroid) and assign every boundary point the distance to this point. The difference to the centroid distance is that their distance is the sum of the Euclidean distance to the nearest point on the skeleton and the geodesic distance on the skeleton from this nearest point to the reference point. The geodesic distance measures the shortest path inside a shape. In contrast to the US vector their vector is computed along the boundary instead of along the skeleton. An articulation will therefore shift the part of the vector that is behind the articulation, resulting in non-trivial comparison. Comparing only parts of objects is not effected by this behavior and the representation is therefore considered invariant to articulation.

3.2.2 Region-based features

Region-based descriptors take into account all points on the boundary as well as all points inside the shape. Structural region-based methods split the shape into parts. The skeleton and MAT is such a structural region-based feature. One can see the skeleton as a structural reduction of the region to a single curve representation with branches being parts. Global region-based features (e.g. geometric moments) will not be discussed here.

3.2.3 Shock Graphs

The skeleton represents the topology of the shape and can be converted into a graph (Skeleton Graph (SG)) where branching and endpoints are nodes and branches are edges. A different approach is used by shock graphs, where skeleton branches are converted to graph nodes that are connected if the branches share a common branching point [31]. Each node is associated with one of 4 shocks that relate to the radius function on the skeleton. A 1-shock is a monotonic in-/decrease, a 2-shock a local minimum, a 3-shock is constant and a 4-shock a local maximum.

A different shock graph was introduced by Sebastian et al. [28] and defines 1- and 3-shocks as links and 2- and 4-shocks as nodes. Additionally, they consider branching points as nodes between 1-shocks and keep the graph embedded in the plane.

3.3 Graph and Vector Matching

Compared to shock graphs, the graph associated with the US is the SG. The goal is to create a metric on two US-vectors that is small for similar shapes and big for dissimilar shapes. Many related works using skeletons employ graph matching, i.e. they calculate the difference between graphs. Most of them rely on sub-graph isomorphism [31, 8] or on the more advanced Graph Edit Distance (GED) [28] that gives the minimum required edit operations to transform one graph to another. For unrestricted graphs both are NP-hard [26]. There exist polynomial algorithms for sub-graph isomorphism on trees, that are used by shock graphs. Zaboli et al.[36] use an edit distance based approach on a tree traversal similar to the US.

The advantage of graph based matching is that structural properties are considered, at the cost of more expensive computation. Additionally, many pattern recognition methods were developed with feature vectors in mind and might not be readily adaptable for graph distances.

Other shape recognition approaches not based on skeletons extract feature vectors from the shape (boundary) [2, 9]. The feature vector enables the use of various statistical pattern recognition methods.

The US vector tries to combine the best of both methods by encoding the structural properties into the feature vector (see Sect. 2.7).

3.4 Normalization and Optimization

In practice, it is common that similar shapes' skeletons are slightly different and this can drastically alter the US vector. As the US vector is cyclic there is no real starting point, so comparison is complicated. There are three different ways to do the comparison (distance calculation): Normalization of the vector, Optimization of the comparison or a combination of both [33]. Normalization was already introduced in Sect. 2.5 and this section goes into more details important for shape analysis.

3.4.1 Normalization

Normalization is most likely the faster approach, where the US vector is modified to a unified representation. As normalization considers the underlying graph (Fig. 3.2b), we call branching and endpoints *special points* of the skeleton. The sequence of visited special points based on the unraveling process is called the *visiting sequence* (Fig. 3.2c).

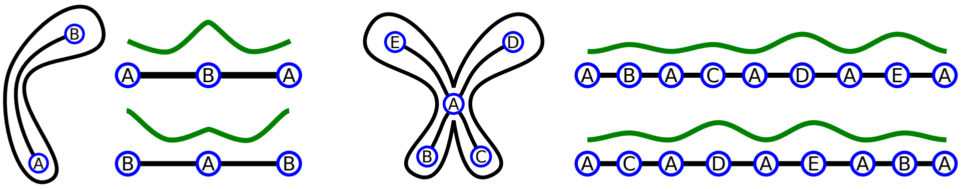
Starting Point and Orientation

A first approach is to find a unique starting point and orientation, which can be done using the eccentricity of the skeleton, as small eccentricity values indicate stable points in the center. Additionally the starting point s_0 should fall on a branching point, as branching points of small eccentricity are likely to be found across similar, scaled shapes. The orientation is towards the branching point with the second smallest eccentricity the - based on the same thinking - next most stable skeleton point. The algorithm is given in Alg. 3.1.

Algorithm 3.1: Starting Point and orientation normalization algotihm.

```
Input: $us$ US vector, $vs$ visiting sequence of special points, $ecc$ list of special
       points' eccentricities
Result: Rotated US vector to starting point and orientation
$s_0 \leftarrow$ getPointWithSmallestEccentricity$(vs, ecc)$;
$S_1 \leftarrow$ successors$(s_0)$;
$o_0 \leftarrow$ getPointWithSmallestEccentricity$(S_1, ecc(S_1))$;
return rotateVector$(us, s_0, o_0)$;
```

There exist two cases where this method will not yield good results. The first is for shapes where two special points have the same eccentricity, the most simple shape only has two endpoints (Fig. 3.1a). The second case has a valid starting point, but two possible directions. This is possible if two special points have the second smallest eccentricity and are both directly connected to the starting point. A simple case is a symmetric shape with the symmetry around a single branch point (Fig. 3.1b).



(a) Same eccentricity on both endpoints.

(b) Orientation not clear, since there are two endpoints B and C with same eccentricity to go to A.

Figure 3.1: Shapes with not well defined starting points.

A possible solution is to also take geometric information into account, and choose the point with bigger radius function if same eccentricity points exist. Unfortunately, this is not a stable approach, as the radius function is mostly very similar for points with minimal eccentricity (i.e., in the center of the shape).

Another solution considers not only one special point but more points of the visiting sequence for the possible starting points and orientations. The orientation is then chosen by the eccentricity of the next not already visited point. To stabilize the starting point and orientation picking algorithm, a threshold can be used to decide when the eccentricity is considered equal. This improved algorithm is given in Alg. 3.2. It exploits the cyclic nature of the list of eccentricity values on special points and compares all possible starting point and orientations from the start till a minimum can be determined.

Algorithm 3.2: Improved Starting Point and orientation normalization algotihm.

```
Input: $us$ US vector, $vs$ visiting sequence of special points, $ecc$ list of special
       points' eccentricities, $thres$ threshold for equality
Result: Rotated US vector to starting point and orientation
$S_0 \leftarrow$ getPointsWithSmallestEccentricity$(vs, ecc, thres)$;
$SO_{check} \leftarrow \{S_0 \times \{\text{successors}(s_0) : s_{\in} S_0\}\}$;
$ecc_{SO_{check}} \leftarrow \emptyset$;
foreach $so_0 \in SO_{check}$ do
    $ecc_{SO_{check}} \leftarrow ecc_{SO_{check}} \cup$ rotateVector$(ecc, s_0(so_0), o_0(so_0))$;
end
for $i \leftarrow 1$ to $size(vs) - 1$ do
    $ECC_{min} \leftarrow$ getArgumentOfSmallestValues$(\{ecc_{so_0}[i] : ecc_{so_0} \in$
      $ecc_{SO_{check}}\}, thres)$;
    if $|ECC_{min}| = 1$ then
        $SO_0 = SO_{check}[ECC_{min}[0]]$;
        $s_0, o_0 = s_0(SO_0), o_0(SO_0)$;
        break
    end
    $ecc_{SO_{check}} \leftarrow ecc_{SO_{check}}[ECC_{min}]$;
end
if $notset(s_0, o_0)$ then
    $SO_0 = SO_{check}[ECC_{min}[0]]$;
    $s_0, o_0 = s_0(SO_0), o_0(SO_0)$;
end
rotateVector$(us, s_0, o_0)$;
```

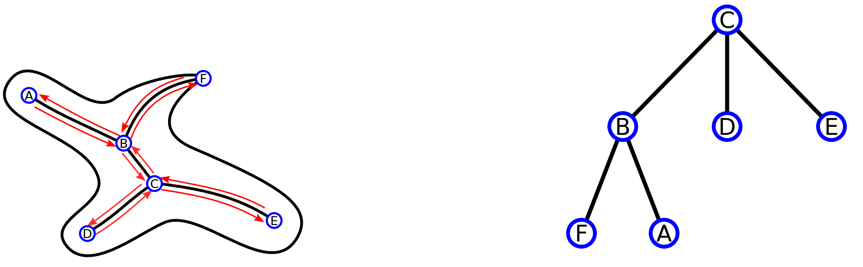
Simple Length Normalization

The simplest form of length normalization, to obtain an unified length for a US vector, is to map the whole vector to the desired length with interpolation of values. In the case of true articulated movement and simple scale this works perfectly. The problem is that in most practical cases the skeleton branches change slightly in length resulting in change of length in the US vector too. Every element of the simple normalized US vector behind a branch that changed in length will be changed, resulting in poor comparison.

Structure Aware Length Normalization

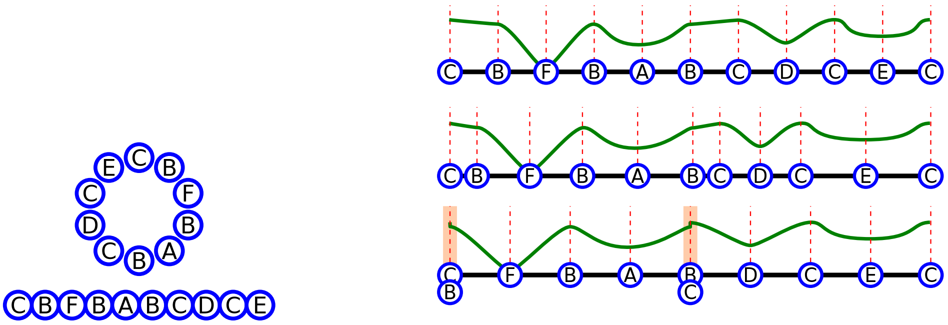
To solve this problem one has to consider the structure of the skeleton and normalize branches individually. Structure Aware Length Normalization (SALN) is the normalization where each section between two special points in the visiting sequence (US *sections*) is normalized to a fixed number n_s of samples. The final vector is then normalized with simple length normalization.

Figure 3.2 shows a shape and its simple normalization and SALN as well as the intermediate representation. The US tree (Fig. 3.2b) is constructed by starting at the starting point and adding a new node when coming across it on the visiting sequence (Fig. 3.2c). Its parent will be the predecessor of the added element in the visiting sequence. This process is similar to the one by Zaboli et al. [36]. Note that the tree reflects the structure of the skeleton/shape and its planar embedding is only depending on the starting point and orientation.



(a) Example shape with skeleton.

(b) Unique tree generated with US



(c) The visiting sequence (lower) and its cyclic nature (upper) visualized

(d) Resulting vector from simple normalization (middle), SALN (uper) and grid-based SALN (lower)

Figure 3.2: Overview of simple normalization and SALN.

SALN removes the problem of unaligned branches in the back of the US vector. On the other hand, it also has two major drawbacks: Firstly, branches that are very long/short will lose this distinctive feature, if they are forced to a unified length. Secondly, spurious branches can distort the vector very strongly.

Short Branch Removal

Another undesired behavior of skeletonization is the introduction of very short branches. With the US vector and the visiting sequence an additional pruning step can be easily achieved. All US sections shorter than a given threshold are removed. If a branch with an endpoint is removed, the branching point on the other side must be checked. It might degenerate to a common point. This can be easily detected in the visiting sequence, since the removed sections will border each other. The number of occurrences of this branching point in the visiting sequence is reduced by one (because two sequences were surrounding the endpoint and the fuse). If a branching point only has two neighbors (equal to two occurrences in the visiting sequence) it is a common point and must be removed from the special points and visiting sequence. This process merges the two US sections that are left and right of the former branching point (at both occurrences). After branch removal the starting position and orientation needs recalculation. It also changes SALN and should therefore be conducted before the SALN process.

Grid-based SALN

Another problem are lost uniqueness of elongated shape features. This can be addressed by introducing a binning feature on the lengths of branches. The general idea is the same as with regular SALN but instead of evenly spaced special points, special points are moved to closest points on a regular grid. This process can merge special points together and might transform branching to common points like short branch removal. This process depends highly on the chosen grid size. Figure 3.2d illustrates grid-based normalization. Note that at the highlighted positions branches have collapsed.

3.4.2 Optimization

Optimization in regard to distances is the process of finding the optimal (minimal) distance between two (normalized) US vectors of length n. Instead of finding starting point and orientation based on a normalization approach, multiple comparisons are conducted to find the best match. A naive algorithm would do n comparisons and taking the smallest distance as the correct one.

Special Point Optimization

As the starting point should always be on a special point an optimization can take this property to reduce the number of needed comparisons. The US vector is rotated by the distance between special points on the US vector. This can be optimized even further by only taking branching points into account, as in any case with at least one branch point, the starting point is a branch point.

Reference Point Optimization

Special point optimization only works if indeed two US vectors share the same structure and have the same special points. There are two reasons for this to be the case. If the shapes are different, their special points will be too, and an optimization should not yield a close distance anyway. The more problematic case is if the skeletonization introduces an unwanted branch (e.g. due to projection). In this case the special points might not line up anymore. If the influence of the additional branch is weak, then reference point optimization [2] can give a better solution.

In reference point optimization the local minima and maxima are considered as points for rotation. A parameter defines the number of extrema wanted and the vector is smoothed as many times as needed to obtain the desired number of extrema.

3.4.3 Distance between US Vectors

Following the observation from the last two sections the simplest distance can be calculated on two normalized US vectors by taking the n-dimensional p-distance or Minkowski distance:

$$d_p(x,y) = ||x-y||_p = (\sum_{i=1}^{n} |x_i - y_i|^p)^{1/p}, x, y \in \mathbb{R}^n, p \geq 1 \tag{3.1}$$

For $p = 2$ the Minkowski distance is the same as the Euclidean distance, $p = 1$ gives the sum of absolute differences between the points of the two vectors. If $p \to \infty$ the Minkoski distance is the Maximum distance, i.e. it gives the maximum error between two components of x, y.

For sufficiently normalized US vectors the Minkowski distance can be used, but if normalization fails, a distance based on optimization is preferred. This is basically the same distance but the minimum, maximum or average of multiple distance calculations with rotated y-US vector is used.

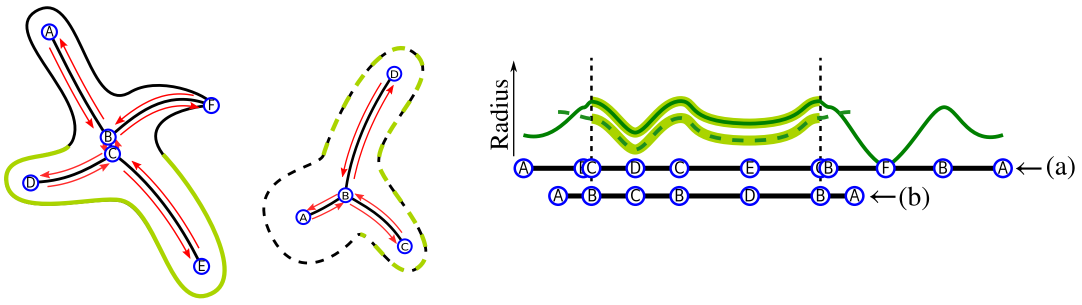
3.5 Part Matching

As stated in the beginning of the last chapter, it is important for a shape descriptor to be invariant to occlusion. The US vector does not have this property, but together with the visiting sequence it is easy to compare parts of shapes with each other. There are two tasks in part matching. The first task is to find shapes that have similar parts, the second is to mark the parts of the shapes that are similar.

3.5.1 Part Matching on the US Vector

As was already shown (Sect. 2.7) a connected part of the US vector that has the same skeleton point at both ends is a connected subtree of the original skeleton tree. It is therefore possible to compare parts of the US vector with another one and find the biggest overlapping.

The process is similar to special point optimization, but instead of a minimum distance the length of connected US sections that are identical in both US vectors is measured. This identity can be relaxed by a threshold. The matching process is illustrated in Fig. 3.3 for a successful match.



(a) First shape (b) Second shape (c) US vectors

Figure 3.3: Two shapes and their part matching with US vectors. The highlighted boundary part is matched between both shapes. Along the matched boundary part and skeleton branches the US is the same (light green highlight). As B and C are very close, they are overlapping in the US vector illustration. The seconds shape's US vector is offsetted for illustration purposes.

3.5.2 Marking Parts

When a sufficiently large overlap is found, the corresponding contour parts of the shape have to be marked. This can be achieved by combining all generating points of Voronoi edges that contribute to the matched skeleton and if necessary filling gaps of points that do not contribute to the skeleton at all. The matched part corresponds to a tree in the skeleton and can therefore be split from the main skeleton. Another approach to mark the part is therefore to find the generating points P_s of the splitted point and an arbitrary generating point p_e of an endpoint (leaf) of the matched subtree. The matched contour is then the contour between p_e and its closest neighbors along the boundary in P_s.

CHAPTER 4

Pose Independent Coordinates

The Unraveled Skeleton (US) has many invariances that are also interesting for a coordinate system that is independent of pose:

Definition 10. A Pose Independent Coordinate System (PICS) assigns every point on a shape a unique coordinate (*Shape Coordinates*), usually a tuple of real values. The point retains this coordinate independent of location, rotation and articulated movement of the shape.

Therefore, a PICS is a natural representation of the boundary and interior of a shape that allows articulated movement, translation and rotation without effecting the relation of internal points to each other. This can be useful for many applications (see Sect. 4.4) and related work has been done mainly in the field of texture mapping and shape blending in computer graphics.

4.1 Shape Coordinates

In this section related work in the field of shape-based coordinates is reviewed. The research area is small and most work has been done in the area of computer graphics in texture mapping and object deformation. Texture mapping refers to the process of projecting an image to a shape, in most cases a rectangular image (texture) to an arbitrary 3D boundary model. In most cases coordinates of the rectangular image are known for key points on the object's boundary and coordinates of the texture are interpolated in between those key points. In this case shape based coordinates improve the interpolation.

Object deformation on the other hand improves position of boundary points on deformed parts, based on coordinates of the shape (or undeformed parts).

4.1.1 Polar Coordinates

The most basic shape coordinates are polar coordinates, where every point is described by a distance and relative rotation to a starting point and orientation. In most cases the starting point is the centroid of the shape, the orientation along its greatest elongation. Polar coordinates are not articulation invariant as relative distance and orientation change on parts of the shape during movement.

4.1.2 Barycentric Coordinates

Original designed for triangles to give every point in a triangle a coordinate dependent on the triangle vertices, barycentric coordinates are extensively used in computer graphics for various tasks where interpolation is needed. Generalizations to arbitrary polygons and connected triangles (triangle mesh) have been researched for a long time [11]. In this section two recent methods are described that interpolate based on deformation of the polygon. For more barycentric extensions see Floater's survey paper [11].

Mean Value Coordinates

Mean Value Coordinates [10] were introduced to represent a vertex v_0 as a combination of its neighboring vertices $v_i, i \in \{1 \dots k\}$ in a triangle mesh. Each neighboring vertex is given a weight $\lambda_i \geq 0$ such that the following two properties are fulfilled:

$$\sum_{i=1}^{k} \lambda_i v_i = v_0 \text{ and } \sum_{i=1}^{k} \lambda_i = 1.$$

If $k = 3$ this corresponds to the classical barycentric coordinates. Since there are infinitely many solutions for weights that satisfy these properties (e.g. just using three neighbors that include v_0 and use classical barycentric coordinates), they have provided a formula to calculate the weights, such that all neighboring vertices contribute smoothly to the coordinates of v_0.

Green Coordinates and Complex Barycentric Coordinates

Green Coordinates [23] and later Complex Barycentric Coordinates [34] are designed to preserve the polygons texture after deformation. Weber et al.[34] show that Green Coordinates are a special case of Complex Barycentric Coordinates. Instead of real numbers for boundary points and weights they use complex numbers with the Cartesian x-coordinate being the real and the y-coordinate being the imaginary part. Similar to the Mean Value Coordinates two properties have to be fulfilled: The constant precision property that states that the sum of all weights must equal one (i.e. the imaginary part is zero) and the linear precision property that states that the linear combination of boundary points with coordinates for an interior point p must produce the Cartesian coordinates of p in its real and imaginary parts.

To obtain the texture of a deformed shape, they first acquire the complex barycentric coordinates for the starting shape and then use a complex function to map those coordinates to the deformed shape. Both are not rooted on articulation joints and therefore do not fulfill the definition for PICSs.

4.1.3 Skeleton-based Coordinates

There have been skeleton-based approaches to give coordinates on the boundary of 2D [29] and the surface of 3D [24] objects. These coordinates do not extend into the interior of the shape and have been considered for specific use cases.

Star Skeletons

An approach by Shapora and Rappoport uses Star-Skeletons [29] to blend two polygons into each other, given correspondences on the boundaries. Star-Skeletons are a partition of the shape into star polygons that are connected with a skeleton-like structure. A star polygon is a polygon for which there exists a point (star origin) that has line-of-sight to all points of the boundary's polygon. Every two neighboring star polygons in a shape share an edge (in the inside of the shape). The skeleton connect the midpoints of those edges and the star origins into a tree structure.

To blend two shapes the algorithm partitions the shapes into compatible star polygons. The star polygons are then represented as polar coordinates as seen from the star origins. The blending is done by interpolating the different star skeleton points (midpoints, star origins and vertices). After that the polar coordinates are recalculated to the Cartesian grid. Finding compatible star skeletons is a costly process ($O(n^4)$) and the representation is not suitable for internal points interpolation.

Skeleton Based Texture Mapping

A different use-case is covered by Madaras and Ďurikovič [24]. They use the skeleton of a boundary mesh to texture the mesh independently from its tessellation. For their method they first extract the skeleton and then associate each skeleton edge with a rectangular part in the texture image, by solving a bin-packing problem. The surface points associated with a skeleton edge are addressed with capsular parameterization. Capsular parameterization projects a capsular (i.e. a cylinder with half-spheres on the ends) to the surface. The method only considers the boundary/surface of the shape, although an extension to the interior should be possible. The greater challenge is handling intersections of the projected capsulars at skeleton edge connections.

4.2 US Coordinates

As already seen in previous chapters, the (normalized) US vector is invariant to translation, rotation and articulated movement. These properties are per definition interesting to obtain a PICS. Therefore, in the following sections the possibility of an US-based coordinate system is explored.

By considering every point of the skeleton as a point of a Voronoi diagram, each point inside the shape is associated with at least one skeleton point. Furthermore, each point can be seen as lying to the left or right in the direction of the unraveling, so they can also be associated with a position on the US. Figure 4.1 shows the closest point in the shape with respect to the directional Voronoi diagram with skeleton point sites.

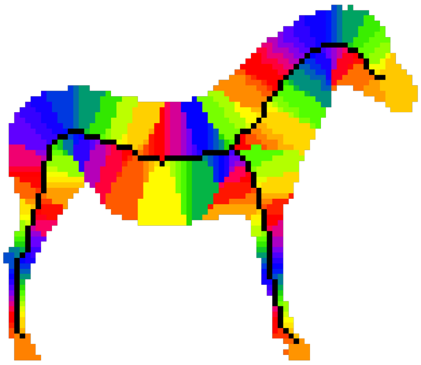
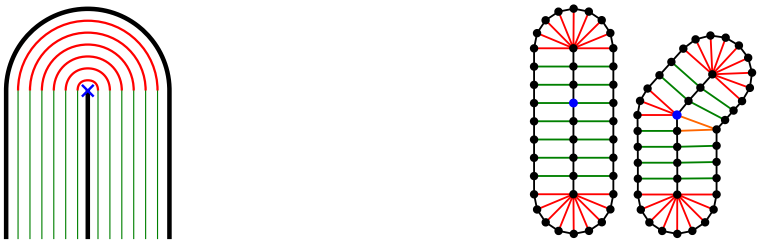


Figure 4.1: Voronoi Diagram with skeleton points as side, including consideration of side of the skeleton.

The first coordinate φ is therefore the closest position on the US. Since the US is cyclic and can be mapped to a circle (see Sect. 2.4) φ lies between 0 and 2π. It is obvious that the US must be normalized regarding its starting point and orientation (see Sect. 2.5). The second coordinate r is the distance to the skeleton. Unfortunately, with only φ and r not all points are uniquely defined as on convex parts of the skeleton or at endpoints one skeleton point has multiple, equidistant points inside the shape associated. Figure 4.2a illustrates the problem for the case of an endpoint. Additionally, articulated movement can introduce/remove problematic areas (Fig. 4.2b).

Points that share the same (φ, r)-coordinates always lie on a circular arc with the center being the skeleton point. A solution to the problem is the introduction of a third coordinate δ that can take values between zero and one and gives the position on the circular arc. This coordinate is therefore dependent on the two extrema of the circular arc, i.e. it is also dependent on the curvature at a given skeleton point. A motivating factor to use δ is that it keeps its relative position during articulated movement. The coordinate δ can be calculated by interpolation using polar coordinates centered at the skeleton point of the two arc endpoints and the point in question.



(a) Points on the red semicircles have the same distance to the endpoint (blue cross) and same φ and r coordinates.

(b) Articulation can create new ambiguous points on skeleton's (red) and shape's (orange) reflex angles.

Figure 4.2: Ambiguous coordinates on endpoints and reflex angles.

4.2.1 Coordinate Conversion

To convert the Cartesian point $p_C = (x, y)$ inside a shape D to US-coordinates $p_{US} = (\varphi, r, \delta)$, with unraveled skeleton US the following formulas can be used:

$$\varphi(p_C) = 2\pi\frac{I(u_C)}{L(US)}, r(p_C) = d(p_C, u_C) \tag{4.1}$$

with $u_C = argmin\{d_o(p_C, x) : x \in US\}$ and d_o gives the Euclidean distance under consideration of the orientation of the US (i.e. points that are not to the right of the unraveled skeleton have infinite distance). $I(u)$ gives the index of the point in the US and $L(US)$ is the total length of the US. The offset coordinate δ is more challenging, as it depends on the local geometry around u_C. Two approaches to calculate δ are given: the Voronoi-dependent approach and for calculation on the grid the skeleton-dependent approach. For both approaches the following three angles have to be calculated: begin of the circular arc α_0, the point on the circular arc β and the end of the circular arc α_1 (see Fig. 4.3). Only the relative difference is needed, so $\alpha_0 = 0$. This leads to the normalization $\delta = \frac{\beta}{\alpha_1}$.

The Voronoi-dependent approach uses all points associated with u_C with distance r and sets α_1 to be the difference between the highest and smallest angle. The angle is computed as angular part ϕ' of the polar coordinate of all points on the arc, where the polar coordinate system is centered at u_C and rotated to have $\phi' = 0$ in the inverse tangent of point u_C. The maximum and minimum ϕ' can then be used to normalize all angles between 0 and 1. The Voronoi-dependent approach can lead to lower errors when Voronoi regions change due to noise on the skeleton compared to the skeleton-dependent approach.

The skeleton-dependent approach uses the normal vector of the line segments leading to and from u_C (n_0, n_1). The end of the arc angle α_1 is based on the dot product: $\alpha_1 = \arccos n_0 \cdot n_1$. The angle β can be calculated similarly by building and normalizing the vector from u_C to p_C. Because small changes on the discretized skeleton can have a huge impact on the normal, a stabilization parameter τ is introduced. It states the distance to the next skeleton point used for normal vector calculations. An empirical comparison between skeleton- and Voronoi-dependent δ calculation is given in Sect. 5.3.2.

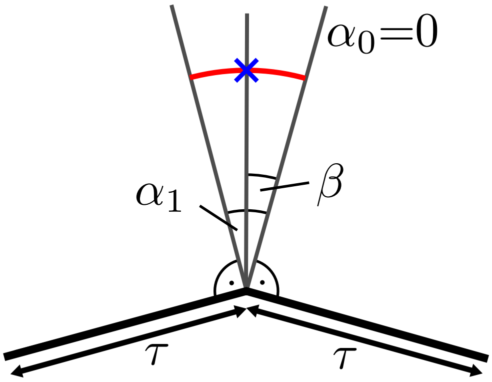


Figure 4.3: Angles needed to calculate δ of blue point.

To convert from US-coordinates to Cartesian coordinates, first the corresponding skeleton point u_C and orientation can be found using φ. The distance r and offset δ' can then be interpreted as polar coordinates. The polar coordinate interpretation of δ, δ', can be calculated using the skeleton-dependent or Voronoi-dependent approach. After α_1 is obtained with one of these approaches (relative to α_0), α_0 is set to the angle part of the polar coordinate of the arc start point and finally $\delta' = \alpha_0 + \delta * \alpha_1$.

4.2.2 Special Point Alignment

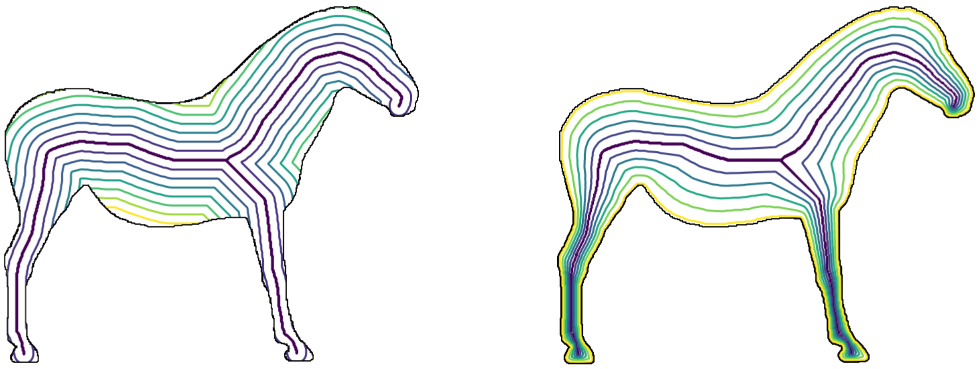
US coordinates with perfect articulation and articulation-invariant skeleton give a PICS. For real world application the articulation and skeleton will not be ideal and to avoid slight shifts in the US due to slightly changed skeletons a normalization similar to Structure Aware Length Normalization (SALN) (see Sect. 3.4.1) can be used. This only influences coordinate φ and gives every skeleton branch two equal ranges of length $\frac{1}{2n_B}$, where n_B is the number of skeleton branches. Note that it is not possible to use a grid-based SALN, as small branches cannot be removed without altering the other coordinates r and δ. It is better to prune the skeleton beforehand.

4.2.3 Deformation and Scale Independence

The US coordinates can be made deformation and scale independent by scaling the r parameter. Using scaled radius function

$$r_s(p_C) = 1 - \frac{d_H(p_C, \partial D)}{d_H(p_C, \partial D), d_H(p_C, S(D))} \tag{4.2}$$

with $d_H(p, X)$ being the Hausdorff distance from point p to set X (i.e. the minimum distance from p to any point of X), the radius coordinate is 0 along the skeleton and 1 along the shape's boundary. This radius function can easily be calculated using the Distance Transform (DT) on the skeleton's compliment and the shape. The scaled radius function has influence on the parameter δ. Figure 4.4 shows a contour plot of the coordinate r with and without scaling. Points sharing r_s and φ do not necessarily lie on a circular arc, but are connected inside the shape and in most cases are convex, so δ can be calculated similarly. Note that the scaled radius function cannot address points outside the shape.



(a) Distance coordinate r on a horse shape. Purple contour at 0, yellow at 70.

(b) Scaled coordinate r_s on a horse shape. Purple contour at 0, yellow at 1

Figure 4.4: The two approaches for non-centered skeletons [20].

4.3 Non-centered Skeletons

As already brought up before, a articulation-invariant skeleton is important for a reliable coordinate system. Furthermore, it is not relevant if the skeleton lies in the center of the object as coordinates are build independent of the centeredness-property. It is therefore necessary to have articulation points (joints) that lie on the skeleton and those points do not lie in the center in many cases (e.g. the spine of a horse is on its back). Langer et al. [20] introduced non-centered Voronoi skeletons, a way to shift the Voronoi skeleton towards the boundary of an object. They use two different ways to achieve this: elliptical distances and weighted DT. The use of non-centered skeletons for the US is possible, but the reconstruction property is lost for the general case.

4.3.1 Elliptical Line Voronoi Diagram and Weighting

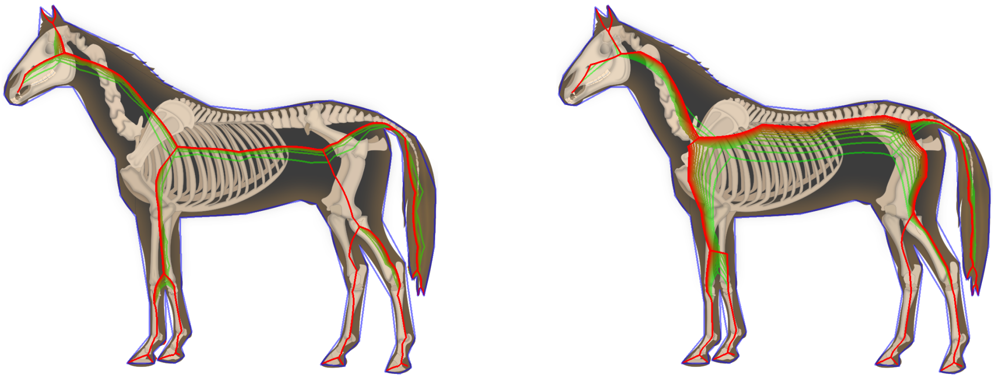
The first way of off-centering the skeleton is using the special property of the Confocal Ellipse-based Distance (CED) defined between a line segment L with endpoints f_1, f_2

and a point p:

$$ced(p, L) = d(p, f_1) + d(p, f_2) - d(f_1, f_2) \tag{4.3}$$

If considered for a Line Voronoi Diagram (LVD) the area of influence to the side for a given line segment increases with its length. This motivates the introduction of Elliptical Line Voronoi Diagram (ELVD) [13] using the CED instead of Hausdorff Distance. The ELVD of the polygonal approximation of a shape's contour can therefore modify the position of the skeleton with differently dense sampling of the boundary.

The second approach uses multiplicative weighting on the DT of line segments on the polygonal approximation of the boundary. The weight attracts the skeleton towards a line segment, i.e. it decreases the area of influence for a given line segment. Figure 4.5 shows the two approaches.



(a) Hand-picked sampling on a horse shape to move skeleton (red) towards anatomical skeleton. The green skeleton is evenly sampled. Intermediate skeletons for higher sampling.

(b) Hand-picked weights on a horse shape to move skeleton (red) towards anatomical skeleton. The green skeleton is unweighted. Intermediate skeletons for increasing weights.

Figure 4.5: The two approaches for non-centered skeletons [20].

Langer et al. [20] showed that the skeleton can be moved to any position using these techniques. To obtain a non-centered skeleton of a shape prior knowledge of the object is needed. The position of articulation points could be used for an optimization process to move the skeleton, or the weights could be learned from known examples.

4.4 Applications

This section gives some possible applications for a PICS (in particular the US coordinates) and motivates the use of PICSs. This thesis does not explore those applications, but some examples can be found in Chapt. 5.

4.4.1 Morphing and Animation

In research shape based coordinates are used most often for shape morphing and pose independent texturing. In case of morphing interesting coordinates lie on the boundary and approaches optimize intermediate frames for optically convincing transformation. The most critical part for morphing with US coordinates is the morphing of the underlying skeleton. For a simple interpolation of shapes, the skeleton must be structural identical and if not normalized must have the same length on each branch. An example use case is the calculation of an intermediate time frame of a walking animal with appropriate skeleton. Topological changing skeletons need a more sophisticated interpretation of the coordinates and have not been investigated.

Pose independent texturing is useful for animation of articulately moving shapes. Together with the US vector as a representation of the shape the US coordinates can be used to model and texture a shape based on a given skeleton. The most difficult part of texturing is handling occlusions.

4.4.2 Super-Resolution

A video sequence of a moving (articulated) shape has more information regarding the interior of the shape than a single image. This can be exploited by saving color information into a PICS-space, i.e. a representation of the object expressed in pose independent coordinates. For US coordinates this space is three dimensional (φ, r, δ). The projection of a shape to US coordinate space is called *US coordinate map*. By combining (and interpolating) multiple US coordinate maps the resulting map can be reprojected on to the Cartesian grid.

Using this method one can vary the resolution of the reprojection by changing the sampling in the US coordinate space, where more samples will give a higher resolution of the reconstructed image. For this technique to work properly it is important that the US skeleton is invariant to articulated movement. For small changes in the skeleton special point alignment can be sufficient.

4.4.3 Segmentation Improvement

Segmentation is the partition of an image into multiple semantically different regions. The default use case is the separation of foreground (object) from background. As stated in Chap. 3 the US assumes a segmentation to obtain the shape and skeleton. Most segmentation approaches are prone to errors along the boundary of regions, as in photographs foreground and background tend to blend into each other at those boundaries (e.g. motion blur). Using the US coordinates along the boundary can help to improve the segmentation in multiple ways, two examples are given.

For the first approach, it is assumed that the boundary is smooth. A smooth boundary has similar r coordinates for neighboring bounding points: $|r(p_1) - r(p_2)| < c|\varphi(p_1) -$

$\varphi(p_2)|, p_1, p_2 \in \partial D$, a small c indicates a smooth boundary. The boundary can then be optimized with regard to c and other information (e.g. color).

The benefit of the US is its invariance to articulated movement (among other invariances). It can be exploited if there already exists a (perfect) segmentation, for example from a different frame of a video or segmented by hand. One can use the coordinates along the boundary to compare the known segmentation with the actual one. In the case of multiple weak segmentations the same thinking can motivate the creation of a mean boundary representation by averaging the r coordinate for same φ and δ coordinates for all boundary points.

CHAPTER 5

Evaluation

This chapter presents the evaluation results produced with the Unraveled Skeleton (US). The chapter is divided into three parts: tests on the general representation, evaluation on shape analysis tasks and qualitative results using the Pose Independent Coordinate System (PICS).

5.1 General

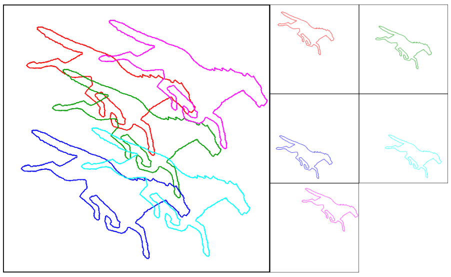
This section provides evaluation results for Chap. 2. It mainly looks at properties introduced in Sect. 2.7.

5.1.1 Invariances

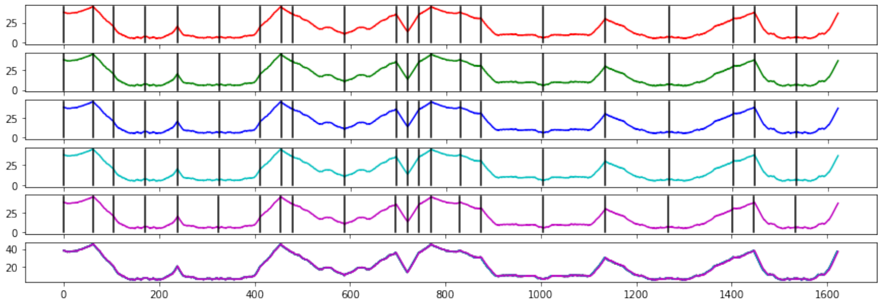
In this section the six invariances theoretically motivated in Sect. 2.7 are empirically evaluated and examples are given. The example shows a running horse with two to four transformed shapes and the US plots are shown as well as a combined plot. All single plots include vertical black line corresponding to special points (branching and endpoints). All computed US vectors use starting point and orientation normalization (Alg. 3.2). For each invariance experiment the mean error between the first and all other US vectors is given. It is relative to the maximum US value, similar to amplitude normalization.

Translation

Figure 5.1 shows the translation result on the horse shape. It can be observed that for pixel-correct translation the US vector does not change. The mean error to the first shape (red) is zero.



(a) Shape (red) and four different translations.



(b) The US vectors of the shapes above.

Figure 5.1: Example of translation invariance.

Rotation

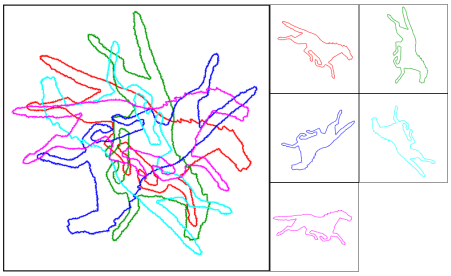
Figure 5.2 shows the rotation result on the horse shape. Minor errors occur due to rasterization. The mean error to the first shape (red) is 0.04.

Scale

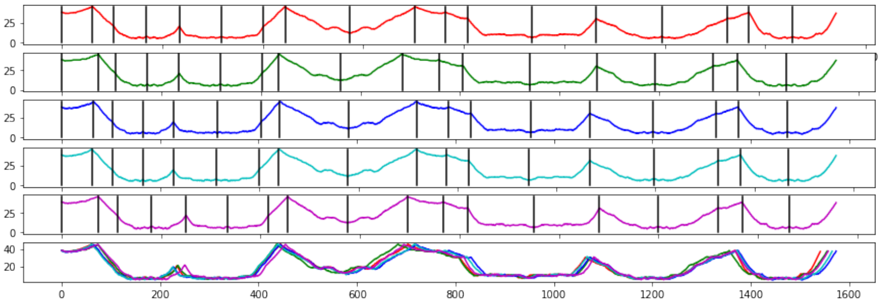
Figure 5.3 shows the scale result on the horse shape. It was simple length normalized and amplitude normalized. Note that non-uniform scaled shapes produce more different US vectors. The magenta shape has a different starting point due to change of the skeleton and its eccentricity. The mean error to the first shape (red) is 0.1, for only the uniformly scaled ones it is 0.02.

Flip

Figure 5.3 shows the result on the flipped horse shapes. The shape was flipped in the two possible ways. Note that the two US vectors of the flipped shapes are similar, because they are rotations of each other. Figure 5.4c shows the flipped US vector for the flipped shapes. This corresponds to the original one (in a cyclic fashion) and must be re-normalized in regard to the starting point and orientation.

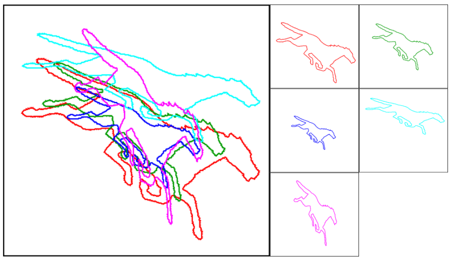


(a) Shape (red) and four different rotations.

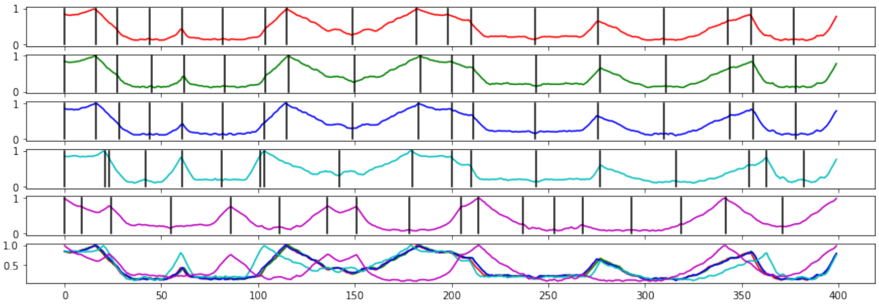


(b) The US vectors of the shapes above.

Figure 5.2: Example of rotation invariance.

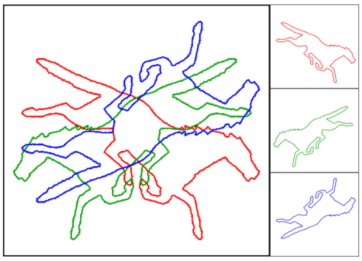


(a) Shape (red) and four different scales. Non-uniform scales in yellow and magenta.

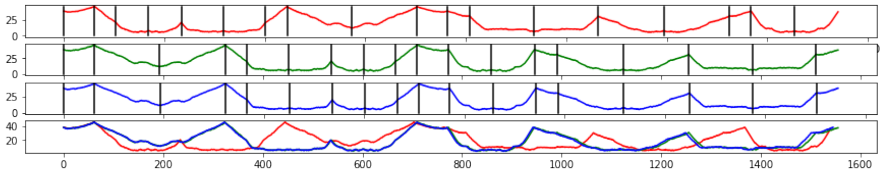


(b) The US vectors of the shapes above.

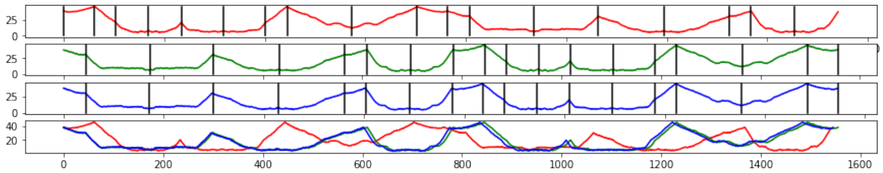
Figure 5.3: Example of scale invariance.



(a) Shape (red) the two possible flips.



(b) The US vectors of the shapes above.



(c) Flipped US vectors for flipped shapes.

Figure 5.4: Example of flip invariance.

Noise

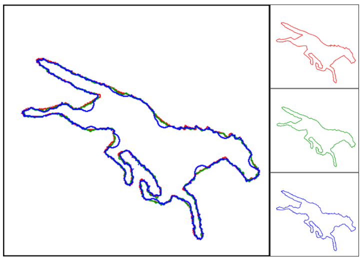
Figure 5.5 shows the noise result on the horse shape. The noise was generated by changing the boundary slightly. For this invarince the threshold for the Voronoi skeletonization was increased to avoid spurious branches due to the noise. The mean error to the first shape (red) is 0.07.

Articulated Movement

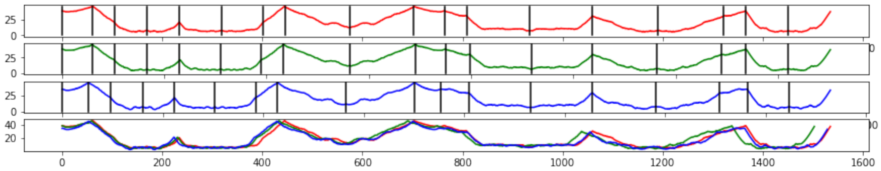
Figure 5.6 shows the result on the horse shape under articulated movement. The mean error to the first shape (red) is 0.05.

5.2 Shape Analysis

This section presents the result of shape analysis using the US vector. The main focus is on shape retrieval.

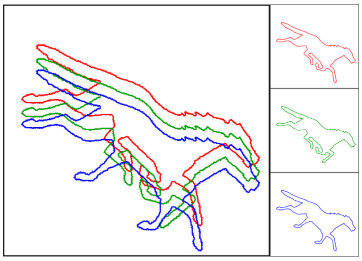


(a) Shape (red) and shape with little and a lot of noise.

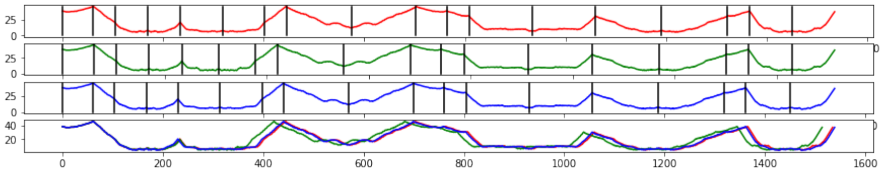


(b) The US vectors of the shapes above.

Figure 5.5: Example of noise invariance.



(a) Shape (red) and two articulated movements. Combined image translated for illustration purposes.



(b) The US vectors of the shapes above.

Figure 5.6: Example of invariance to articulated movement.

5.2.1 Datasets and Experiments

The US vector was tested on three different datasets: Leaves [1], Tools [5, 4] and MPEG7[2]. The Leaves dataset consists of 100 different classes of leaves with 16 shapes per class and has a high inter-class similarity. The Tools dataset consists of 7 classes with 5 shapes each and consists of pliers and scissors that have articulated movement. The MPEG7 dataset consists of 70 classes with 20 shapes each and has been used extensively in research. On each dataset two different experiments were conducted with different parameter settings.

Precision-Recall Plot

The first experiment uses shape retrieval to generate a precision-recall plot. This test is performed on shape retrieval and gives the precision for every possible recall value. The recall is the number of shapes found of the correct class (TP) divided by the total number of samples of this class ($TP + FP$). The precision is TP divided by the total number of found shapes ($TP + FN$) [37]. TP relates to the true positive (i.e. correctly in the result), FP to the false positive (i.e. in the result but wrongly so) and FN to the false negative (i.e. not in the result, but should be there).

To calculate the plot first all distances between all different pairs of US vectors are calculated. For every shape the minimum number of closest elements is calculated, so this set includes x TP-shapes. This is repeated for every x smaller than the number of shapes in a class. In the end the mean value of the precision for each x over all samples is taken. The *Accuracy* is calculated by the ratio of in-class results to all results for the closest $n - 1$ elements, where n is the number of shapes per class.

Classification

The second experiment uses a k-Nearest-Neighbor (kNN) classifier to obtain a confusion matrix. The kNN predicts a label for a given feature by looking at the closest k samples in the training set (*neighbors*) and assigns a label by taking a majority vote on the labels of those k neighbors. For every class c the confusion matrix assigns a value to every class depending on the number of predictions for a test sample from c. The results are than presented in a square matrix with a row/column for every class. In the perfect case only the diagonal of the matrix is different from zero. The training set includes 2/3 of each class.

Parameters

The following parameters were altered: the normalization function $norm$, the length of the US vector n, the length of a section in Structure Aware Length Normalization (SALN) n_s, the grid resolution for grid based SALN n_g and the distance function $dist$.

---

[1] `https://archive.ics.uci.edu/ml/datasets/One-hundred+plant+species+leaves+data+set`

[2] `http://www.dabi.temple.edu/~shape/MPEG7/dataset.html`

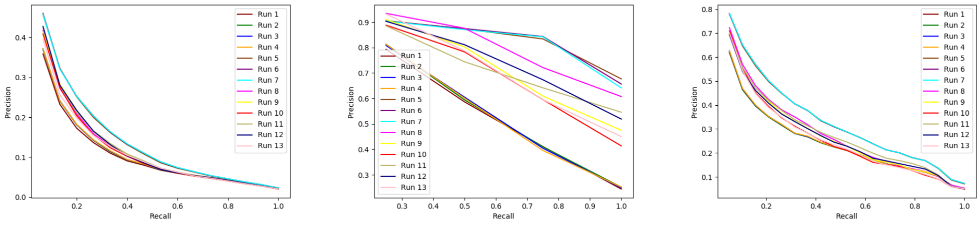
Run	*norm*	n	n_s	n_g
1	SALN	100	20	-
2	SALN	100	50	-
3	SALN	500	50	-
4	SALN	1000	100	-
5	simple	100	-	-
6	simple	500	-	-
7	simple	1000	-	-
8	grid based SALN	100	-	5
9	grid based SALN	500	-	10
10	grid based SALN	500	-	20
11	grid based SALN	1000	-	10
12	grid based SALN	1000	-	20
13	grid based SALN	1000	-	50

Table 5.1: The different runs for experiments in shape analysis evaluation.

Table 5.1 shows the different parameters used for all experiments. Every run in the table uses the following distances: Manhattan, Euclidean, Chebyshev. The minimum length of branches ϵ was set to 5 for all runs. For all tests a Voronoi skeletonization threshold of 20 was used.

5.2.2 Results

Table 5.2 shows the accuracy for different runs and datasets. The accuracies have all been achieved with the Manhattan distance, because it produced the highest accuracies of tested simple distances. Figure 5.7 shows the precision-recall plot of all runs with the Manhattan distance.



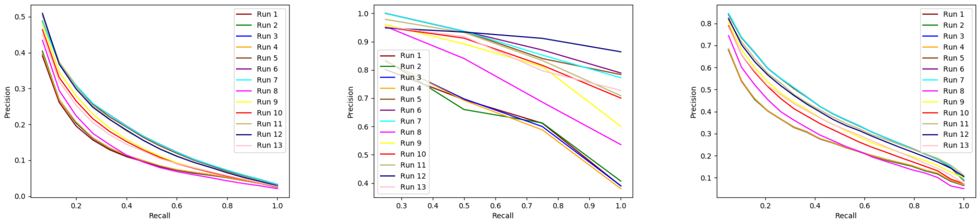
(a) Leaves dataset. (b) Tools dataset. (c) MPEG7 dataset.

Figure 5.7: Precision-Recall plots for the three datasets and the Manhattan distance.

The right columns of Table 5.2 use the optimization distance from Sect. 3.4.2. Figure 5.8 shows the precision-recall plot of all runs with the optimization distance.

	Manhattan Distance			Optimization		
Run	Leaves	Tools	MPEG7	Leaves	Tools	MPEG7
1	0.111	0.407	0.237	0.125	0.514	0.266
2	0.114	0.400	0.236	0.128	0.535	0.265
3	0.114	0.414	0.238	0.129	0.529	0.267
4	0.115	0.414	0.239	0.129	0.529	0.268
5	0.150	**0.730**	0.323	0.185	0.800	0.381
6	**0.151**	0.721	**0.326**	**0.186**	0.829	**0.384**
7	**0.151**	0.714	0.325	0.185	0.807	**0.384**
8	0.130	0.671	0.271	0.139	0.636	0.285
9	0.131	0.586	0.262	0.167	0.743	0.346
10	0.127	0.550	0.254	0.161	0.779	0.320
11	0.135	0.571	0.277	0.184	0.743	0.375
12	0.132	0.593	0.265	0.181	**0.850**	0.364
13	0.126	0.550	0.255	0.155	0.736	0.337

Table 5.2: Accuracy on different runs and datasets. Best run per dataset is bold. Comparison of Manhattan distance and optimization.



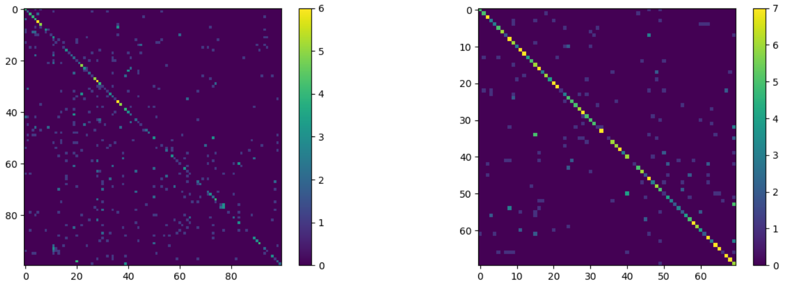
(a) Leaves dataset. (b) Tools dataset. (c) MPEG7 dataset.

Figure 5.8: Precision-Recall plots for the three datasets and the optimization distance.

The kNN did classify all test data correctly on the tools dataset for run 6, 7 and 12. With an accuracy of 32.8% run 12 gave the best result on leaves classification (see Fig. 5.9a) and run 6 (with 65.7%) on MPEG7 classification (see Fig. 5.9b). All results were achieved with the optimization distance.

5.2.3 Discussion

It can be observed that the results are highly dependent on the chosen normalization and that the length of branches is an important feature, since the SALN has low scores for every dataset. Additionally, it can be seen that the length of the final vector plays a major role, especially for grid-based SALN. It shows, that the US vector gives a good approximation of the shape for only a few values.



(a) Leaves dataset.

(b) MPEG7 dataset.

Figure 5.9: Confusion matrix of classifying the test set.

Based on the superiority of the optimization distance one can conclude, that the starting point normalization methods used can still be improved. Additionally the results on the optimization distance show that length normalization (grid based SALN) can improve the overall results, e.g. by removing small branches.

5.3 Pose-independent Coordinates

The US coordinates were qualitatively tested. In this section we compare different methods and parameters for conversion of Cartesian to US coordinates.

5.3.1 US Coordinates on the Grid

The biggest challenge is, although continuously well defined, the representation on the grid. The main problem is that color information is only available for discrete points and by changing the skeleton, the new coordinates required are not well aligned with the original ones.

There exist different ways of implementing the US coordinates, here two are compared: grid interpolation and retrieval.

Grid Interpolation

The idea of grid interpolation is to create a 3D grid that covers the whole range of coordinates, i.e. for φ between 0 and 2π and for r and δ between 0 and 1. The resolution of the grid has to be of a size such that all information of the shapes interior can be saved. A good start is two times the length of the skeleton in φ-direction ($\Delta\varphi$) , because every discrete position of the skeleton is covered that way. The maximum value of the radius function r_{max} on the skeleton (maximum US vector value) is a sensible value for the resolution in r-direction (Δr), since this is the number of possible points between skeleton and shape boundary. To cover a half-circle in δ-direction ($\Delta\delta$) a resolution of $r_{max}\pi$ is needed.

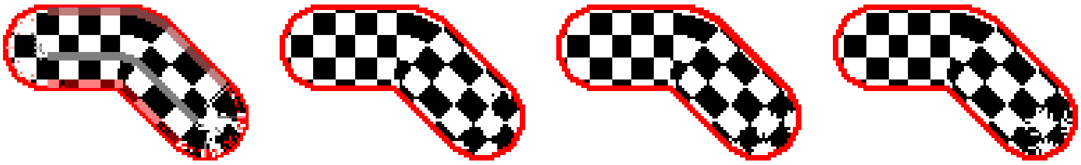
This grid will be sparse after filling it with known values from an image. The unknown values can then be interpolated from known values in a two step procedure. At first the δ values are interpolated along the same (φ, r) coordinates. Afterwards for every δ value each (φ, r) image is interpolated. This yields a 3D lookup image.

Retrieval

The grid interpolation approach takes up a lot of resources ($O(\Delta\varphi \times \Delta r \times \Delta\delta)$ in memory) and is useful, if multiple poses and re-calculations to Cartesian grid are needed, since color lookups are in linear time. For fewer re-calculations one can use retrieval methods, where the closest coordinate in a 3D-space is looked up for every US coordinate of a different pose. This process can be made more robust by taking the closest k coordinates and taking a weighted average based on the distance. For this a normalization is critical, so that distances are shortest along the δ axis. This approach has a time complexity for color lookups of $O(n)$, where n is the number of points in the 3D-space.

δ Binning

An additional parameter δ_B is introduced with δ binning. When sampling distances on the grid, circular arcs of equal distance may get disconnected. This results in coordinates getting the same δ values although they lie on a discrete circular arc. The binning introduces a rounding of r values to a fixed number of possible values. Experiments have shown that the a good value for δ_B lies close to the biggest value of the radius function. Figure 5.10 shows different δ binning values for a capsule. Note the under representation (gray values) for $\delta_B = 5$, the gray and light red area results form multiple pixels having the same r component resulting in mean values. For $\delta_B = 100$ noise is generated on the capsule's end (lower right side). Delta binning is useful for the interpolation and the retrieval approach.



(a) $\delta_B = 5$ (b) $\delta_B = 15$ (c) $\delta_B = 30$ (d) $\delta_B = 100$

Figure 5.10: Bend capsular with maximum radius 30 and different δ_B.

5.3.2 Results

Figure 5.11 gives a qualitative comparison of interpolation and retrieval. The original object is converted to US coordinates and then mapped to the same and transformed shape. One can see that for small $\Delta\delta$ the capsule's ends are not represented correctly, even for the simple reconstruction (note the blurred lines on the capsule's ends). Another observation is that for the retrieval method the bend down end of the capsule is represented better. For the interpolation method at this region there are artifacts from interpolation of the 3D image. Artifacts of the retrieval method are smoothing for too high values of k, noticeable in the most bottom-right image.

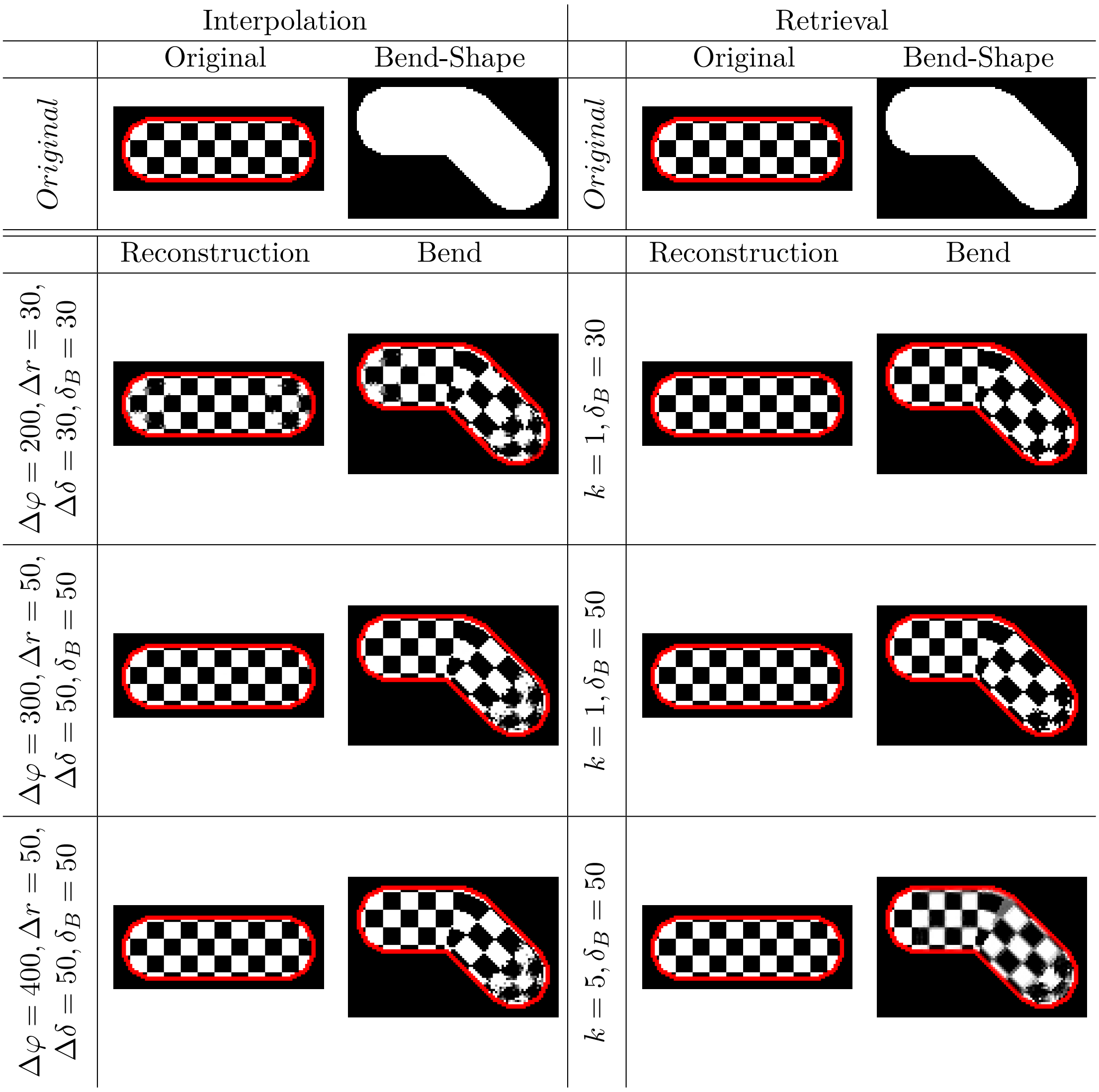
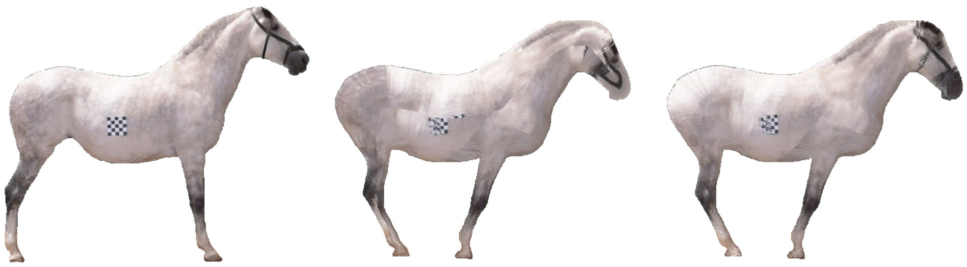

Interpolation
Retrieval
Original
Bend-Shape
Original
Bend-Shape
Original
Original
Reconstruction
Bend
Reconstruction
Bend
$\Delta\varphi = 200, \Delta r = 30, \Delta\delta = 30, \delta_B = 30$
$k = 1, \delta_B = 30$
$\Delta\varphi = 300, \Delta r = 50, \Delta\delta = 50, \delta_B = 50$
$k = 1, \delta_B = 50$
$\Delta\varphi = 400, \Delta r = 50, \Delta\delta = 50, \delta_B = 50$
$k = 5, \delta_B = 50$

Figure 5.11: Comparison of interpolation and retrieval with different resolutions.

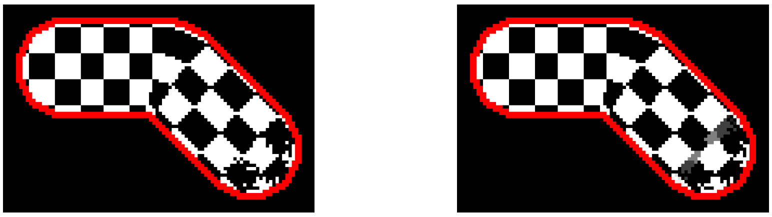
Length normalization

When shapes' skeletons have more than one branch it is good to normalize φ, so all branches have the same resolution. Each branch between special points in the US therefore gets $\frac{2\pi}{n_B}$ where n_B is the total number of branches. Figure 5.12 shows the difference between a reconstruction on a changed skeleton with and without length normalization.

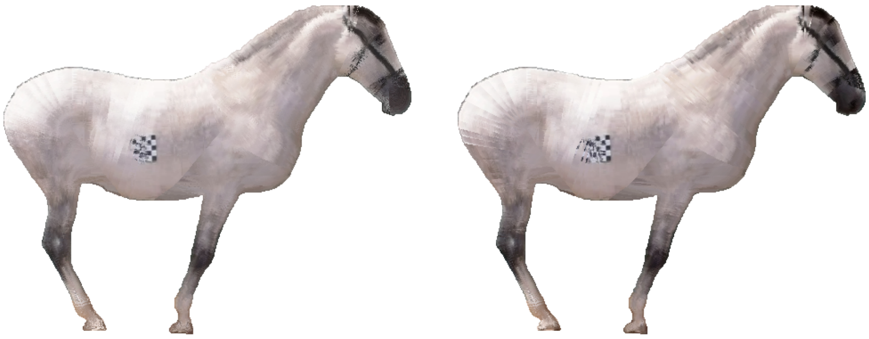


(a) Original image (b) No length normalization (c) Length normalization

Figure 5.12: Normalization and no normalization of δ using retrieval with $k = 3$ and $\delta_B = 75$. Note the incorrect representation on the head. One can see the misalignment of the branches along the centerlines of the shape.



(a) Voronoi based on capsule (b) Skeleton based on capsule



(c) Voronoi based on horse (d) Skeleton based on horse

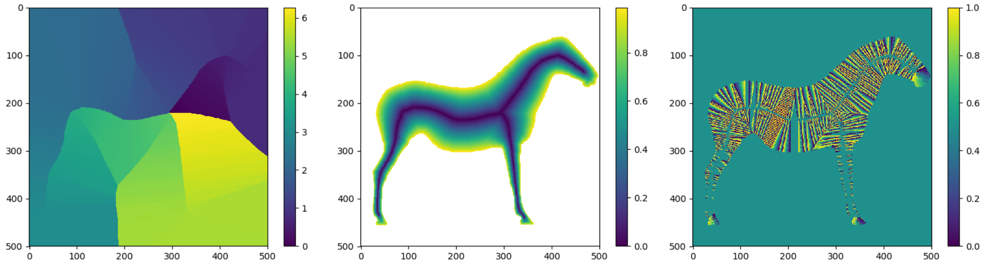
Figure 5.13: Voronoi and skeleton based calculation of δ using retrieval with $k = 1$ and $\delta_B = 50$ on capsule and $k = 3$ and $\delta_B = 75$ on horse. The original images are in Fig. 5.11 and Fig. 5.12.

δ coordinate calculation

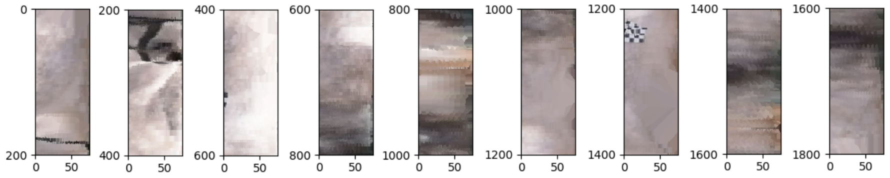
Two methods were used for calculation of δ as described in Sect. 4.2.1: Voronoi and skeleton based. Figure 5.13 shows results using the two methods. Note that the skeleton based method looks smoother but introduces gray values that are not present in the original image. This is due to multiple pixels of the original image getting packed into the same coordinate.

5.3.3 Coordinates and US coordinate space images

Figure 5.14 shows the different coordinates on the example of the image of a horse. Figure 5.14d shows a slice of the interpolated 3D image at coordinate $\delta = 0.5$. As can be seen, the normalized r coordinate is only valid inside the shape. The δ coordinate is unstable on a slightly different rasterization of the skeleton.



(a) φ coordinate (b) r coordinate (c) δ coordinate



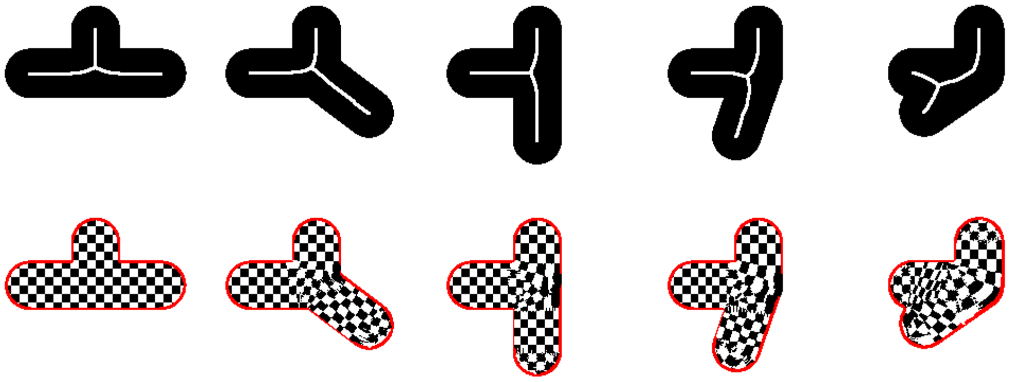
(d) Interpolated image at $\delta = 0.5$. Vertical axis is φ, horizontal axis is r. Reformatted for display purposes.

Figure 5.14: Voronoi and skeleton based calculation of δ using retrieval with $k = 1$ and $\delta_B = 50$ on capsule and $k = 3$ and $\delta_B = 75$ on horse.

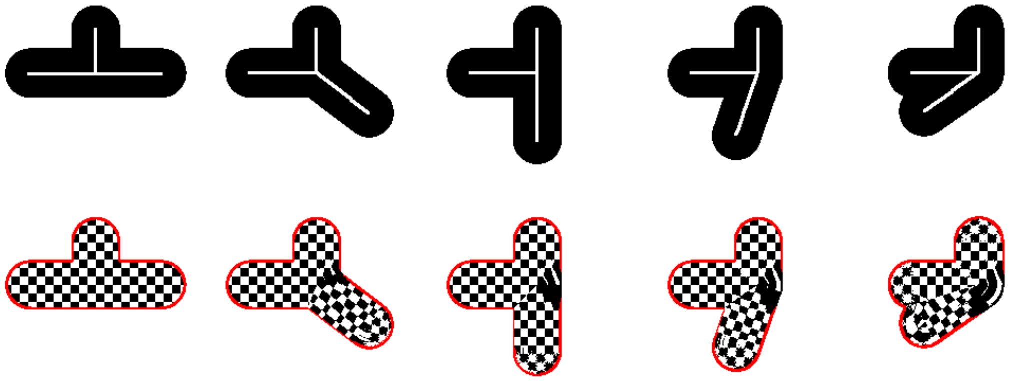
5.3.4 Non-Centered Skeletons

Figure 4.5 gives a comparison between the non-centered and normal skeleton for US coordinates. Note that the non-centered skeleton approach does not change anything that has not moved (left and top branch), but does not interpolate smoothly, because all pixels related to the bend have the same φ (as opposed to the normal skeleton, where φ values are smoothed due to the inaccurate skeleton at the joint). One can also see that the rightmost shape (the most strongly bent one) introduces new r values for many points using the scale invariant normalization of r based on the boundary of the object.

Figure 5.16 shows an example application transferring color information from one shape to another using US coordinates and non-centered skeletons.



(a) Voronoi Skeletons. First row shows shapes and skeletons, second row transformed image with the retrieval method. The leftmost column is the known color image. Skeleton thickness increased for illustration purposes.

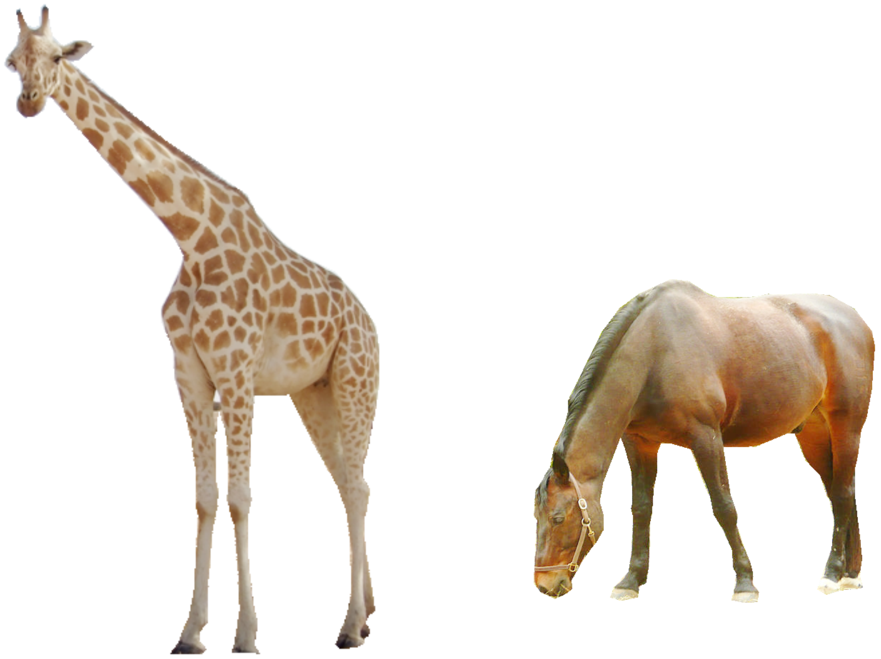


(b) Non-centered skeletons. First row shows shapes and skeletons, second row transformed image with the retrieval method. The leftmost column is the known color image. Skeleton thickness increased for illustration purposes.

Figure 5.15: Comparison of articulation using normal Voronoi skeletons and Non-Centered Voronoi skeletons.

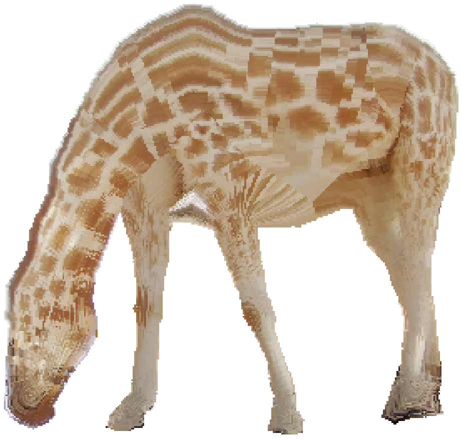
5.3.5 Discussion

The biggest - and still unsolved - challenge of the US coordinates is the rasterization of the skeleton in particular in regard to the δ coordinate. The simple checkerboard examples show that for correct articulation the result is acceptable. Slight changes in the skeleton give recognizable results, but of worse quality then the original.



(a) Giraffe Image (input)

(b) Horse Image (input)



(c) Giraffe on Horse

Figure 5.16: Transfer of color information from a giraffe to a horse shape[1].

---

[1] Giraffe and Horse image under Creative Commons Attribution 3.0 with authors Clémence Delmas and Sheila respectively. Shared via wikimedia.org.

CHAPTER 6

Conclusion

This thesis introduced a new shape representation based on the unraveled skeleton (US). The properties and applications of the representation were covered and a quantitative and a qualitative evaluation was given.

Its invariance to articulated movement is a major advantage of the representation (as shown in Sect. 2.7.1 and Sect. 5.3.4) and is based on the skeleton making the skeletonization the greatest challenge for an adequate representation. In this regard Voronoi skeletons were reviewed and its strengths analyzed, but even the fairly robust Voronoi skeleton can be problematic with articulated shapes. For this problem the non-centered skeletons have been studied.

The US was studied in regard to shape analysis and special normalization and optimization strategies were introduced, also in regard to matching only parts of shapes. In addition, the US was used to build a coordinate system based on the skeleton (US coordinates) and therefore independent to articulated movement.

The properties of the US look very promising, although more research must be conducted to make the process more general in shape analysis. The normalization methods presented must be refined in regard to scale and local changes, as these are the greatest challenges for the use of the US in shape analysis. The evaluation also shows that normalization can still be improved as optimization can achieve better results.

Regarding US coordinates, the biggest challenge are points with ambiguous distance, that has been solved by introducing a new polar coordinate. This coordinate is hard to relate to intuitively (unlike the other two), and prone to errors. For this an elegant solution for the problem must be found, e.g. based on the curvature of distance iso-lines.

It can be extremely useful to combine shape analysis and pose-independent coordinates to match objects with there shape and color information. The color information can be accessed with the pose-independent coordinates taking advantage of all its properties.

In general the US has many desirable properties and this thesis gives a first introduction to the benefits and limitations, and how to overcome some of them.

Acronyms

CED Confocal Ellipse-based Distance. 35, 36

DCD Discrete Curve Evolution. 6

DT Distance Transform. 11, 12, 15, 35, 36

ELVD Elliptical Line Voronoi Diagram. 36

ET Eccentricity Transform. 13, 14

GED Graph Edit Distance. 22

kNN k-Nearest-Neighbor. 44, 46

LVD Line Voronoi Diagram. 7, 36

MAT Medial Axis Transform. 4, 10, 20, 21

PICS Pose Independent Coordinate System. 29, 31, 32, 34, 36, 37, 39

SALN Structure Aware Length Normalization. 24–26, 34, 44–47

SG Skeleton Graph. 21, 22

US Unraveled Skeleton. xiii, 1, 2, 11–17, 19–29, 31–44, 46–49, 51, 52, 55, 56

VD Voronoi Diagram. 6–8

Bibliography

[1] Xiang Bai, Longin Jan Latecki, and Wen-Yu Liu. Skeleton pruning by contour partitioning with discrete curve evolution. *IEEE transactions on pattern analysis and machine intelligence*, 29(3):449–462, 2007.

[2] Thomas Bernier and Jacques-Andre Landry. A new method for representing and matching shapes of natural objects. *Pattern Recognition*, 36(8):1711–1723, 2003.

[3] Harry Blum et al. A transformation for extracting new descriptors of shape. *Models for the perception of speech and visual form*, 19(5):362–380, 1967.

[4] Alexander M Bronstein, Michael M Bronstein, Alfred M Bruckstein, and Ron Kimmel. Analysis of two-dimensional non-rigid shapes. *International Journal of Computer Vision*, 78(1):67–88, 2008.

[5] Alexander M Bronstein, Michael M Bronstein, and Ron Kimmel. *Numerical geometry of non-rigid shapes.* Springer Science & Business Media, 2008.

[6] Fred Buckley and Frank Harary. *Distance in graphs.* Addison-Wesley, 1990.

[7] Luciano da Fontoura Da Costa and Roberto Marcondes Cesar Jr. *Shape analysis and classification: theory and practice.* CRC Press, Inc., 2000.

[8] Cecilia Di Ruberto. Recognition of shapes by attributed skeletal graphs. *Pattern Recognition*, 37(1):21–31, 2004.

[9] Michael Donoser, Hayko Riemenschneider, and Horst Bischof. Efficient partial shape matching of outer contours. In *Asian Conference on Computer Vision*, pages 281–292. Springer, 2009.

[10] Michael S Floater. Mean value coordinates. *Computer aided geometric design*, 20(1):19–27, 2003.

[11] Michael S Floater. Generalized barycentric coordinates and applications. *Acta Numerica*, 24:161–214, 2015.

[12] Herbert Freeman. On the encoding of arbitrary geometric configurations. *IRE Transactions on Electronic Computers*, 10(2):260–268, 1961.

[13] Aysylu Gabdulkhakova, Maximilian Langer, Bernhard W Langer, and Walter G Kropatsch. Line voronoi diagrams using elliptical distances. In *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, 2018.

[14] Raghuraman Gopalan, Pavan Turaga, and Rama Chellappa. Articulation-invariant representation of non-planar shapes. In *European Conference on Computer Vision*, pages 286–299. Springer, 2010.

[15] Richard W Hall. Parallel connectivity-preserving thinning algorithms. In *Machine Intelligence and Pattern Recognition*, volume 19, pages 145–179. Elsevier, 1996.

[16] Ines Janusch, Nicole M Artner, and Walter G Kropatsch. Euclidean and geodesic distance profiles. In *International Conference on Discrete Geometry for Computer Imagery*, pages 307–318. Springer, 2017.

[17] Reinhard Klette. *Concise computer vision.* Springer, 2014.

[18] Reinhard Klette and Azriel Rosenfeld. *Digital geometry: Geometric methods for digital picture analysis.* Elsevier, 2004.

[19] Walter G Kropatsch, Adrian Ion, Yll Haxhimusa, and Thomas Flanitzer. The eccentricity transform (of a digital shape). In *International Conference on Discrete Geometry for Computer Imagery*, volume 4245, pages 437–448. Springer, 2006.

[20] Maximilian Langer, Aysylu Gabdulkhakova, and Walter G Kropatsch. Non-centered voronoi skeletons. In *International Conference on Discrete Geometry for Computer Imagery*, volume 11414, pages 355–366. Springer, 2019.

[21] Der-Tsai Lee. Medial axis transformation of a planar shape. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 4(4):363–369, 1982.

[22] Haibin Ling and David W Jacobs. Shape classification using the inner-distance. *IEEE transactions on pattern analysis and machine intelligence*, 29(2):286–299, 2007.

[23] Yaron Lipman, David Levin, and Daniel Cohen-Or. Green coordinates. *ACM Transactions on Graphics (TOG)*, 27(3):78, 2008.

[24] Martin Madaras and Roman Ďurikovič. Skeleton texture mapping. In *Proceedings of the 28th Spring Conference on computer Graphics*, pages 121–127. ACM, 2013.

[25] Robert Ogniewicz and Markus Ilg. Voronoi skeletons: Theory and applications. In *Computer Vision and Pattern Recognition, 1992. Proceedings CVPR'92., 1992 IEEE Computer Society Conference on*, pages 63–69. IEEE, 1992.

[26] Kaspar Riesen. *Structural Pattern Recognition with Graph Edit Distance: approximation algorithms and applications.* Advances in computer vision and pattern recognition. Springer, 2016.

[27] Punam K Saha, Gunilla Borgefors, and Gabriella Sanniti di Baja. A survey on skeletonization algorithms and their applications. *Pattern Recognition Letters*, 76:3–12, 2016.

[28] Thomas B Sebastian, Philip N Klein, and Benjamin B Kimia. Recognition of shapes by editing their shock graphs. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 26(5):550–571, 2004.

[29] Michal Shapira and Ari Rappoport. Shape blending using the star-skeleton representation. *IEEE Computer Graphics and Applications*, 15(2):44–50, 1995.

[30] Frank Y Shih. *Image processing and pattern recognition: fundamentals and techniques.* John Wiley & Sons, 2010.

[31] Kaleem Siddiqi, Ali Shokoufandeh, Sven J Dickinson, and Steven W Zucker. Shock graphs and shape matching. *International Journal of Computer Vision*, 35(1):13–32, 1999.

[32] Sergios Theodoridis and Konstantinos Koutroumbas. *Pattern recognition.* Acad. Pr., Amsterdam [u.a.], 4. ed., 3. [print.] edition, 2010.

[33] Petrus Johannes Van Otterloo. *A contour-oriented approach to digital shape analysis.* PhD thesis, Delft University of Technology, 1988.

[34] Ofir Weber, Mirela Ben-Chen, Craig Gotsman, et al. Complex barycentric coordinates with applications to planar shape deformation. In *Computer Graphics Forum*, volume 28, page 587, 2009.

[35] Mingqiang Yang, Kidiyo Kpalma, and Joseph Ronsin. *A survey of shape feature extraction techniques*, pages 43–90. In-Tech, 2008.

[36] Hamidreza Zaboli, Mohammad Rahmati, and Abdolreza Mirzaei. Shape recognition by clustering and matching of skeletons. *Journal of computers*, 3(5):24–33, 2008.

[37] Dengsheng Zhang and Guojun Lu. A comparative study of curvature scale space and fourier descriptors for shape-based image retrieval. *Journal of Visual Communication and Image Representation*, 14(1):39–57, 2003.

[38] Dengsheng Zhang and Guojun Lu. Review of shape representation and description techniques. *Pattern recognition*, 37(1):1–19, 2004.

[39] TY Zhang and Ching Y Suen. A fast parallel algorithm for thinning digital patterns. *Communications of the ACM*, 27(3):236–239, 1984.