Die approbierte Originalversion dieser Dissertation ist an der Hauptbibliothek der Technischen Universität Wien aufgestellt (http://www.ub.tuwien.ac.at).

The approved original version of this thesis is available at the main library of the Vienna University of Technology (http://www.ub.tuwien.ac.at/englweb/).

DISSERTATION

Registration on the Basis of Surface Geometry and Texture

ausgeführt zum Zwecke der Erlangung des akademischen
Grades eines Doktors der technischen Wissenschaften
unter der Leitung von

o.Univ.Prof. Dr. Helmut Pottmann
Institutsnummer E104
Institut für Diskrete Mathematik und Geometrie

eingereicht an der Technischen Universität Wien
Fakultät für Mathematik und Geoinformation

von

Dipl.-Ing. Stephan H. Kölpl
Matrikelnummer 8832372
Sandgasse 9, 8010 Graz

Graz, im Dezember 2005

Dedicated to my grandfather

Thomas Kölpl

1910 - 1985

Abstract

The registration of two or more three-dimensional shape representations plays an important role in areas as different as robotics, engineering and medicine, among many others. Although there by now exist a variety of powerful methods and algorithms, there still are unresolved problems such as the registration of shapes departing from general initial positions and of special geometries such as kinematic surfaces. This dissertation aims to find and evaluate methods to solve those open issues.

The first chapter introduces the topic, touches the standard algorithm ICP (Iterative Closest Point), discusses the problems that are covered in this work, and briefly discusses existing literature on registration with acquired surface texture and on global registration.

The second chapter describes registration using acquired surface texture. For smoothly distributed textures a method based on the gradient of texture is presented. This method and its applicability to kinematic shapes is evaluated with examples in 2D and 3D.

In the third chapter integral invariant descriptors in 2D and 3D are presented, their implementation is covered and various examples are shown. Whereas the planar integral invariants described are already covered in the literature, three new spatial integral invariant descriptors are presented, all based on the intersection between the kernel sphere and the shape to be described.

These integral invariants are used in chapter four to support registration departing from general initial positions (global registration). Demonstrated is their advantage over differential invariants, i.e. the lower sensitivity to distortions (noise) and their scalability. Apart from the implementation, examples in 2D and 3D are shown.

Kurzfassung (German)

Die Registrierung von zwei oder mehreren Repräsentationen dreidimensionaler Formen spielt eine große Rolle in so verschiedenen Anwendungen wie Robotik, Maschinenbau und Medizin neben vielen anderen. Wenn auch inzwischen vielfältige leistungsfähige Methoden und Algorithmen existieren, so gibt es doch noch offene Probleme wie die Registrierung von Formen in allgemeinen Startlagen und von speziellen Geometrien wie kinematische Flächen. Diese Dissertation hat zum Ziel Methoden zur Lösung dieser offenen Probleme zu finden und zu evaluieren.

Das erste Kapitel führt in das Thema ein, geht auf den Standardalgorithmus ICP (Iterative Closest Point) ein, weist auf die in dieser Arbeit behandelten Probleme hin und behandelt in Kürze bisherige Arbeiten zu Registrierung mithilfe von Oberflächentextur und zu globaler Registrierung.

Das zweite Kapitel behandelt die Registrierung mithilfe von aufgenommener Oberflächentextur. Für gleichmäßig, annähernd stetig verlaufende Texturen wird eine Methode auf der Basis von Texturgradienten vorgestellt. Diese Methode und ihre Anwendbarkeit für kinematische Formen wird anhand von Beispielen in 2D und in 3D evaluiert.

Das dritte Kapitel stellt integrale invariante Deskriptoren in 2D und in 3D vor, behandelt ihre Implementierung und zeigt vielfältige Beispiele. Während die behandelten zweidimensionalen Invarianten schon in der Literatur beschrieben wurden, werden drei neue dreidimensionale integrale Invarianten basierend auf der Schnittkurve zwischen Kernel und zu beschreibender Fläche vorgestellt.

Im vierten Kapitel werden diese integralen Invarianten verwendet um Registrierung ausgehend von allgemeinen Startlagen zu erreichen (Globale Registrierung). Ersichtlich wird der Vorteil gegenüber differentiellen Invarianten, d.h. die Immunität gegen Störungen (Rauschen) und die Skalierbarkeit. Neben der Implementierung werden Beispiele in 2D und 3D gezeigt.

Résumé (French)

La mise en correspondance de représentations de formes 3D joue un rôle important dans des domaines aussi différents que la robotique, l'ingénierie et la médecine. Même s'il existe de nos jours de nombreux algorithmes et méthodes performants, il demeure des problèmes non résolus comme la mise en correspondance de formes ayant une position initiale quelconque et des géométries spécifiques comme par exemple les surfaces cinématiques. Cette thèse cherche à développer et à évaluer des méthodes permettant de résoudre ces problèmes.

Le premier chapitre présente le sujet, introduit l'algorithme standard ICP (Iterative Closest Point), discute des problèmes couverts par cette thèse et énumère brièvement les précédents travaux portant sur les mises en correspondance globale et à l'aide de textures de surface.

Le second chapitre décrit la mise en correspondance utilisant des textures de surface optiques. Une méthode basée sur le gradient de texture et fonctionnant pour des textures également réparties est présentée. Cette méthode et son application à des formes cinématiques sont évaluées sur des exemples en 2D et 3D.

Dans le troisième chapitre, des descripteurs invariants intégraux en 2D et 3D sont introduits, leur implémentation est décrite et de nombreux exemples sont présentés. Même si les invariants intégraux en 2D décrits ont déjà été couverts par de précédents travaux, trois nouveaux invariants intégraux 3D sont présentés. Ceux-ci sont basés sur l'intersection entre le noyau et la forme à décrire.

Ces invariants intégraux sont utilisés dans le quatrième chapitre pour permettre une mise en correspondance partant de positions initiales quelconques (mise en correspondance globale). Leur immunité aux distorsions (bruit) et leur adaptabilité aux changements de taille démontre clairement leur avantage sur les invariants différentiels. Suivant leur implémentation, des exemples en 2D et 3D sont montrés.

Аннотация (Russian)

Регистрация двух и более изображений трёхмерных форм находит широкое применение в роботике, машиностроении и медицине. Если существуют разнообразные продуктивные методы и алгоритмы, то существуют также открытые проблемы, как, например, регистрация формы в общем начальном положении и регистрация специальной геометрии, как, например, кинематические поверхности. Эта диссертация ставит целью найти и оценить методы решения этих проблем.

Первая глава – вводная, знакомит нас с темой и стандартными алгоритмами ’Iterative Closest Point’ (ICP), излагает проблемы, решение которых является целью этой работы, кратко описывает уже проведённые работы по регистрации с помощью поверхностных текстур и работы по глобальной регистрации.

Вторая глава описывает регистрацию с помощью оптических поверхностных текстур. Для равномерных, систематически движущихся и плавно изменяющихся текстур предлагается метод на основе градиента текстур (перепада текстур). Этот метод и возможность его использования для кинематических форм показывается на примерах 2D и 3D.

Третья глава представляет интегральные неизменные дескрипторы в 2D и 3D - пространствах, описывает их реализацию и показывает разнообразные примеры. Тем временем, как рассмотренные двухмерные инварианты уже были описаны другими авторами, в моей работе представлены три новых трёхмерных интегральных инварианта. Они базируются на кривой пересечения между ядром оператора и описанной поверхностью.

В червёртой главе эти интегральные инварианты уже используются для достижения регистрации исходя из общих начальных положений (глобальной регистрации). При этом хорошо видно преимущество перед дифференциальными инвариантами – это устойчивость к шумам и масштабируемость. Наряду с реализацией также показаны примеры в 2D и 3D - пространствах.

Acknowledgments

I would like to sincerely thank my professor Helmut Pottmann for his supervision of my thesis, for his support and patience that I received although I come from a different field of studies as an engineer of mechatronics.

Of all the people of the Institute of Discrete Mathematics and Geometry, especially Dr. Stefan Leopoldseder, my supervising assistant professor, deserves my thankfulness, furthermore Dr. Martin Peternell.

Finally my friends and colleagues who supported me during my thesis shall be mentioned here, most of all Felizitas Berner, Ingénieur Informatique Jean-Philippe Andreu and Julia Rass.

The underlying CAD model for the shape first time depicted in fig. 2.2 is courtesy of Böhler Schmiedetechnik GmbH & Co KG.

This work has been financially supported by the Austrian Science Fund (FWF) under grant P16002-N05.

Graz, December 2005 Stephan H. Kölpl

Typesetting by the author using LaTeX 2ε.
Copyright ©2005 by Stephan H. Kölpl

Contents

Abstract **III**

Kurzfassung **IV**

Résumé **V**

Аннотация **VI**

Acknowledgments **VII**

List of Figures **XI**

List of Tables **XII**

1 Introduction **1**

- 1.1 Registration 1
- 1.2 The ICP Algorithm 2
- 1.3 Shortcomings of ICP 3
- 1.4 Selected Prior Work and own Contribution 5

2 Registration with Acquired Texture **7**

- 2.1 General Remarks 7
- 2.2 Registration with Acquired Texture in 2D 10
  - 2.2.1 Theory of Registration with Acquired Texture in 2D . 10
  - 2.2.2 Results of Registration with Acquired Texture in 2D . 15

2.3 Registration with Acquired Texture in 3D 17
2.3.1 Theory of Registration with Acquired Texture in 3D . 17
2.3.2 Results of Registration with Acquired Texture in 3D . 18

3 Geometry - Inherent Texture 22
3.1 Geometry - Inherent Texture in 2D 22
3.1.1 Integral Invariant Descriptors in 2D 22
3.1.2 The Integral Distance Invariant 23
3.1.3 The Integral Area Invariant 24
3.2 Geometry - Inherent Texture in 3D 27
3.2.1 Integral Invariant Descriptors in 3D 27
3.2.2 The Intersection Descriptor 27
3.2.3 The Intersection-Barycenter Descriptor 32
3.2.4 The Intersection-Eigenvalue Descriptor 38

4 Registr. using Geom.-Inherent Texture 40
4.1 Global Registration 40
4.2 Global Registration in 2D using Geometry Inherent Texture . 42
4.2.1 Implementation in 2D 42
4.2.2 Results in 2D 46
4.3 Global Registration in 3D using Geometry Inherent Texture . 51
4.3.1 Implementation in 3D 51
4.3.2 Results in 3D 54

Bibliography 63

Lebenslauf (cv in German) A

List of Figures

1.1 Convergence of ICP compared to tangent based and curvature based method 4

2.1 Tangential movement on rotational symmetric surface 8
2.2 Example for a shape with an extruded section 9
2.3 Scanned view of a beverage can 9
2.4 Minimization of greyscale value differences 11
2.5 Acquisition of greyscale values from image 'peppers512x512' . 16
2.6 Registration of planar spline curve with texture 16
2.7 Registration of circle segment with texture 17
2.8 Beverage can as kinematic shape example 19
2.9 Registration of kinematic shape with texture 20

3.1 Smooth, open and edgy, closed planar curve 24
3.2 Integral distance invariant for a smooth planar curve w/o and with noise 24
3.3 Integral distance invariant for an edgy planar curve w/o and with noise 25
3.4 Normalized integral distance invariant curves for a smooth and an edgy planar curve for a range of kernel radii 25
3.5 Approx. integral area invar. for a smooth planar curve w/o and with noise 26
3.6 Normalized approx. integral area invariant curves for a smooth planar curve for a range of kernel radii 26

3.7 Detail of implementation of the integral intersection descriptor 32
3.8 Comparison of three integral invariants using the face data set 33
3.9 Comparison of three integral invariants using the press data set 34
3.10 Implementation of the integral intersection-barycenter descriptor 37
3.11 Consecutive views of Stanford bunny w. int. barycenter inv. . . 37

4.1 Data sets used for planar registration using integral invariants 44
4.2 Initial positions for planar data sets 45
4.3 Comparison of nearest neighbors in 2D vs. in (2 + 1)D 46
4.4 Comparison of nearest neighbors in 2D vs, in (2 + 1)D for noisy data 47
4.5 The registration process for the planar curve 4 with noise added 49
4.6 Ground truth error metric versus iterations for the registration of curve 4 (see fig. 4.5) with noise added 49
4.7 The registration process for the planar curve 4 with noise added using multiple integral invariant descriptors 50
4.8 Ground truth error metric versus iterations for the registration of curve 4 with noise added using multiple integral invariant descriptors 50
4.9 Areas without descriptor values as function of kernel radius . . 53
4.10 Data set Face and initial position 55
4.11 Ground truth error metric versus number of iterations for the registration of the Face data set 55
4.12 Data set Drop Forge A and initial position 57
4.13 Ground truth error metric versus number of iterations for the registration of the Drop Forge A data set 57
4.14 Data set Drop Forge A and B 59
4.15 Ground truth error metric versus iterations for the registration of the Drop Forge A and B data sets 59

List of Tables

3.1 Integral intersection invariant vs. kernel radii for various cases 30

3.2 Integral intersection-barycenter invariant vs. kernel radii for various cases 36

4.1 Success of planar registration with/without integral invariants 51

Chapter 1

Introduction

1.1 Registration

The term registration describes the process of establishing the spatial transformation needed to best-possibly align two or more representations of a shape. The definition of the best alignment hereby relates to a defined error metric, most often based on Euclidean distances between overlapping regions. Based upon the established registering transformation a range of subsequent processes and methods can be applied depending on the goal in mind. For an early publication on the topic see Faugeras and Hebert [FH86].

Registration is of ever increasing importance for a range of quite different applications. One example is reverse engineering, where a closed digital representation of a general physical object such as a mechanical machine part is wanted [EFF98]. From this digital model, information about the basic underlying structure would then be extracted in order to arrive, in the ideal case, at a CAD model allowing the reproduction of the part in question, see [PR98][PWL01] and [PLWP02].

Similar to this there exist archaeological applications which in contrast to reverse engineering typically would stop with a closed digital model and a suitable visualization of it. This field of application can be classified in that it is not limited to special geometry and that it suffices to employ rigid transformations.

Another growing field is medical applications, for example in the context of computer assisted surgery. For a comprehensive treatment of this topic with respect to the mid-90s state of science see the doctoral thesis of D.A. Simon [Sim96]. Due to the properties of the object to be modeled (i.e., the human body) and its possibly changing shape during the scanning process (consider breathing), non-rigid registration is often necessary here.

Still other applications might be restricted to special shapes with their own advantages and inconveniences, such as the registration of multiple acoustic range views for underwater scene reconstruction, described in [CFM02]. Also covering special geometries is the example of an existing application: the modeling of tunnels in the process of their building/drilling using hierarchical feature vector matching on the surface texture in order to establish correspondences between overlapping surface scans.

Having stated that the registration problem concerns two or more shapes, one special case in quality control shall be emphasized here. Namely the case, when a shape represented by a 3D data point cloud is to be best possibly registered on its ideal, that is the underlying CAD model. This case is special in that it benefits from large overlap up to identity of the boundaries of the two shapes and in that here a point cloud has to be registered onto an analytical representation. Multiview registration on the other hand can be challenging in terms of the amounts of data treated. Pulli proposed a method that 'aligns scans pairwise with each other and use the pairwise alignments as constraints that the multiview step enforces while evenly diffusing the pairwise registration errors.' [Pul99].

With our application in mind, there is a slight focus in this work on rigid registration of general objects with an aim toward remote engineering and quality control. Especially the interest in the problem field of kinematic surfaces covered in chapter 2 was fanned by practical experience in quality control of drop forged machine parts.

1.2 The ICP Algorithm

A classical standard algorithm to perform registration is the ICP algorithm (short for Iterative Closest Point). It has been presented by Besl and McKay [BM92]. About the same time Chen and Medioni proposed a similar algorithm [CM92]. It differs from ICP in that it employs Gauss-Newton iteration, and because of this performs distinctively better than ICP for fine registration.

ICP is an iterative algorithm consisting of two steps. We consider two sets of points representing overlapping regions of a shape. The moving point set is to be rigidly transformed (moved) toward the fixed data set. In the first step for each point in the moving set the closest (in terms of Euclidean distance) point in the fixed set is searched. This results in a point set $Y = (\mathbf{y}_1, \mathbf{y}_2, ...)$ of closest points in the fixed data set to the point sequence of the moving point set $X = (\mathbf{x}_1, \mathbf{x}_2, ...)$ where points with equal indices correspond to each other.

This step of each iteration is the most time consuming, calling for efficient implementation, possibly even using appropriate data structures. In this work such nearest-neighbor search is always done using the ANN (Approximate Nearest Neighbor) implementation of D.M.Mount [AMN+98].

The second step consists of computing the rigid motion m that moves the mobile point set such that distances between moved points $m(\mathbf{x}_i)$ and corresponding points $\mathbf{y}_i$ in the fixed point set are minimized;

$$F = \sum_{i=1}^{N} \|m(\mathbf{x}_i) - \mathbf{y}_i\|^2. \tag{1.1}$$

Besl and McKay as well as Horn [Hor87] showed how to solve this least squares problem explicitely. The translational part of the resulting transformation moves the barycenter of the moving point set onto the barycenter of the fixed point set whereas the rotational part results from the eigenvector belonging to the maximum eigenvalue of a symmetric 4 x 4 matrix involving the covariance.

ICP did of course not remain unchanged. Numerous adaptions and enhancements have been published, such as the fast algorithms by Jost and Hügli [JH02] and the 'grid closest point' method by Farag et al. [YAHF98] to name just two. For a good summary on 'efficient variants of ICP' see Levoy and Rusinkiewicz who besides propose 'Iterative Corresponding Point' to be a more appropriate expansion of 'ICP' due to the real nature of its working principle [RL01].

1.3 Shortcomings of ICP

Nevertheless the ICP algorithm itself has its weaknesses. One issue concerns the rate of convergence. This however has already been addressed by various publications, among them [PHYH04], [MGPG04], [PH03] (see also [LPZ03]), [PLH04], and [PLH02].

One approach described is not only to use the simple point correspondences found in step one of ICP as the basis of registration, but to consider the neighborhood of the point and replacing the surface there by the tangent plane or even a second order approximation. Fig. 1.1 shows the relative advantage of improved methods over ICP in terms of speed of convergence. The data sets to be registered are visible in their initial position in the top left picture. The three diagrams show mean squared ground truth distances vs. number of iterations for the ICP method (top right), a method minimizing distances to the tangent plane (bottom left), and a method minimizing distances to a local second order approximation (SDM, bottom right) respectively. For the latter two methods the curves are only drawn to the point where the change of the subjective error metric of the algorithm underwent a certain predefined level or again got worse.

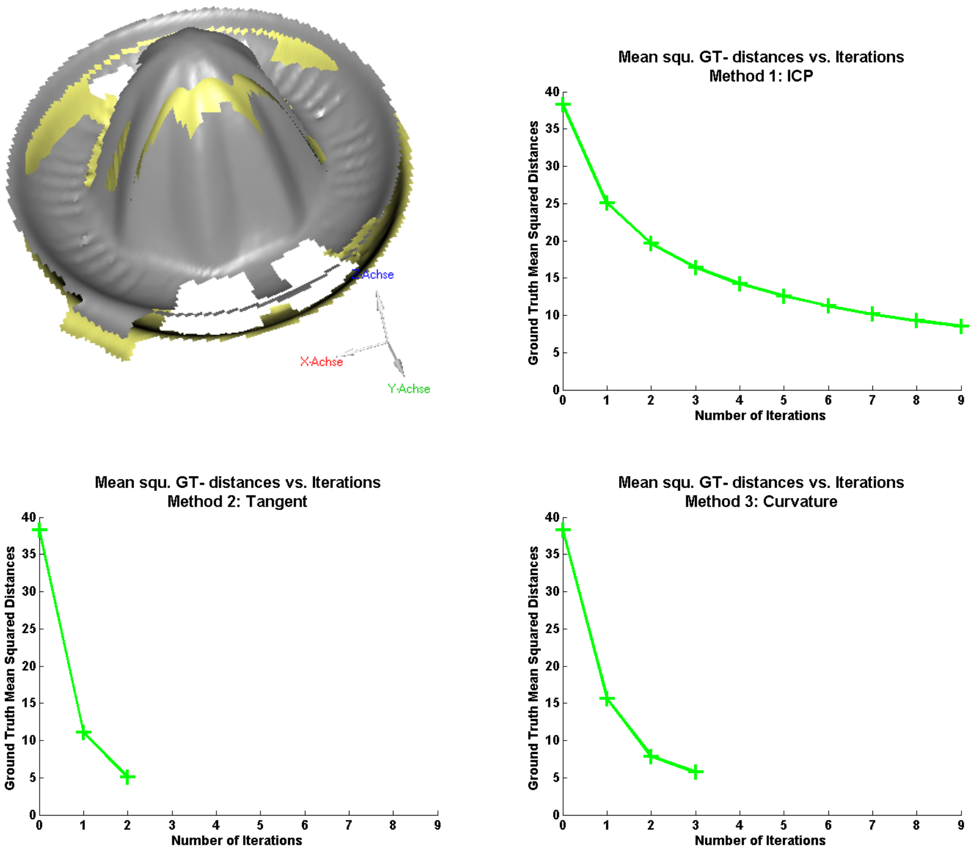

Figure 1.1: Convergence of ICP compared to tangent based and curvature based method

Another issue concerns kinematic shapes. Surface patches on kinematic shapes such as rotationally symmetric or extruded objects have one ore more degrees of freedom, in that they can be moved on the kinematic shape in a way that does not increase or decrease the subjective error metric of purely geometric registration methods. This weakness inherent to all purely geometric methods but can hardly be ignored given the wide presence of such geometries in applications.

One final issue mentioned here is the necessity to provide well pre-aligned initial positions to assure that ICP finds the global minimum contrary to getting stuck in a local one. On the one hand so called global registration starting from general initial positions only employing the ICP method will work only for a narrowly limited range of shapes and applications. On the other hand there is growing need for methods that achieving global registration for more general shapes in a fully automatic manner, i.e., without needing human interaction.

1.4 Selected Prior Work and own Contribution

Registration of kinematic shapes and with acquired texture

Among the literature concerning the use of surface texture for registration purposes we find the approach of Johnson and Kang [JK97]. They propose a version of ICP they call 'color ICP', which projects the texture of data before registered by means of ICP onto a 'final consensus surface' where the overlapping textures then are blended. The authors claim a decrease in registration error by an order of magnitude.

Mainly for an archaeological application Sablatnig and Kampel describe a method for the registration of the front- and the backviews of rotationally symmetric objects using the axis of rotation of the 'fragments' [JK97].

We propose a simple method that works for the registration of smoothly textured shapes. It works independently of geometric registration and uses texture alone but can be combined with geometrical registration methods. This method can be used for the registration of all kinds of kinematic shapes and is not restricted to rotational symmetry.

Integral invariant descriptors and global registration

Affine and Euclidean invariants are frequently used in image processing: for an example in image retrieval see [TG99]. Manay et al. proposed a special group of Euclidean invariants in the plane, gained by integral methods, and used these 'integral invariants' for shape recognition, demonstrating their advantages over differential invariants [MHYS04]. Further they demonstrated how to match even heavily deformed shapes via smooth re-parametrization of their integral invariant signatures [MCYS05].

Telea et al. use similar invariants, based on moment analysis, for the classification of surfaces [TRC03] and one more step closer to our application Sharp et al. use a weighted linear combination of positional and feature distances for ICP registration [SLW02].

We propose to employ Euclidean invariant features to support spatial rigid registration. Therefore we present some new integral invariant descriptors which we use to gain an additional description of shapes. We then use these descriptors for the establishment of point correspondences in the first step of geometrical registration algorithms.

Using one or several integral invariant descriptors in this way certainly will not be able to ensure global registration for general shapes. Nevertheless it considerably widens the range of applications and shapes where global registration is possible. In other terms, it enlarges the 'funnel of attraction' into global minima when compared to the standard method.

Basing on integral invariants Huang and Pottmann describe a method for automatic and robust multi-view registration [HP]. A series of integral invariants are used there to extract a few important clusters from each scan. In a next step surfaces are matched pairwise based on cluster-cluster correspondences. Finally recursive application of pairwise surface matching is used to achieve global multiple view registration.

Chapter 2

Registration with Acquired Texture

2.1 General Remarks

While the first chapter showed that for most cases registration starting from close initial positions is a field covered by potent methods, there are special cases that pose difficulties.

Imagine for instance two or more overlapping views of a rotational symmetric object. Further imagine that no additional information is available, not even in such a form as the presence of local deviations from the ideal shape. In this context a shape deviation would assist registration and a systematic analysis of the influence of shape deviations of various kinds on the registration process would certainly be interesting for the case where an acquired shape is registered onto and compared to its ideal, that is, the CAD model. But this lies beyond the scope of this work and it suffices to state that for general registration a method should also work in the case of vanishing deviations from the ideal shape, which nevertheless in most instances is the optimal and wanted case.

Purely based on geometry, the above mentioned views possesses a degree of freedom in their movement with respect to each other. This degree of freedom in mutual movement is such that a movement of rotation around the common axis of symmetry will in general not increase (or decrease) the subjective error metric of the algorithm. Pure classical ICP implementation is known to tend to suppress tangential movement and therefore will not 'use' this degree of freedom in movement. But indeed ongoing relative movement of point sets some times can be encoutered during registration in commercially available software which (presumably) use more complex methods, at least when multiple point sets are involved (see fig. 2.1 for an example of registration of multiple views using an established software package).

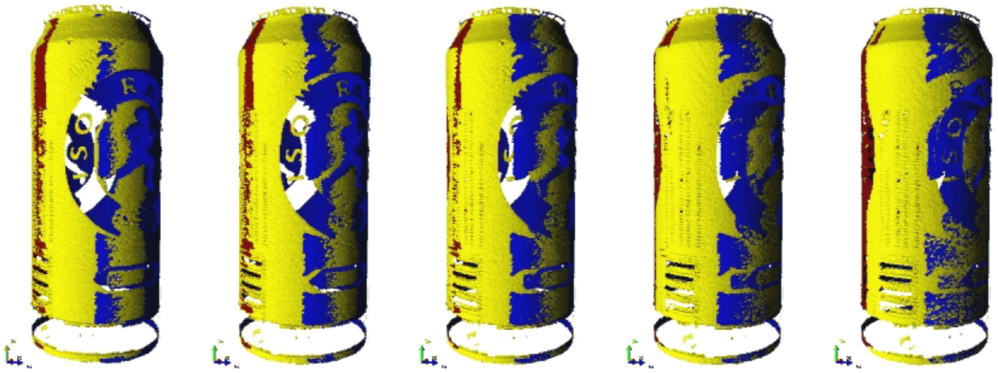

Figure 2.1: Tangential movement on rotational symmetric surface (Positions shown for iteration 1, 9, 17, 25, and 33 from left to right. The yellow point set can be seen to slide on the others, note the barcode field.)

The figure shows a sequence of intermediate positions in registering multiple views of a beverage can. The initial position (leftmost picture) is the result of the acquisition using a numerically controlled turntable and therefore already close to the real global minimum. The other pictures show from left to right the positions after the 9th, 17th, 25th, and finally 33rd iteration. The point set drawn in yellow can clearly be seen to be gliding on the other point sets (look at the barcode field in the lower part of the can). Even more, the speed of movement does not decrease and the movement goes on after the 33rd iteration shown.

Rotational symmetry is of course not the only registration specific problem case within the group of kinematic surfaces (for an overview see for example [Pot04]). The mechanical part depicted in Fig. 2.2 is a quite typical example for the other field of calamities. The central section contains a shape that can be described by the movement of its cross section as generatrix along a straight line. Also due to its length this section posed difficulties when registering using commercially available software and indeed overlapping views did tend to glide on each other parallel to the length axis.

The third 'dangerous' case of kinematic surfaces are helical surfaces. These are shapes that are traced out by a generating curve undergoing the combination of a uniform rotation about an axis and a uniform translation along the same axis. But since it rarely if ever occurs in practical applications (put

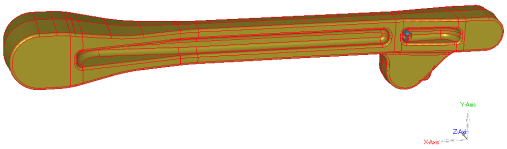

Figure 2.2: Example for a shape with an extruded section in the center

aside spiral drills and the occasional application in architecture) it does not need special attention here.

To overcome this difficulty that cannot be overcome purely on a geometrical basis, additional information is welcome. And indeed, one source of additional information, potentially helpful for the registration process, often is easily accessed. Due to the working principle of many 3D scanning devices, more precisely due to their incorporation of CCDs or similar sensors, a classical central projected digital photographic image of the view often is available as byproduct of the scanning process. Consequently a greyscale or even color value is easily available for each valid spatial vertex of a scan (see fig. 2.3).

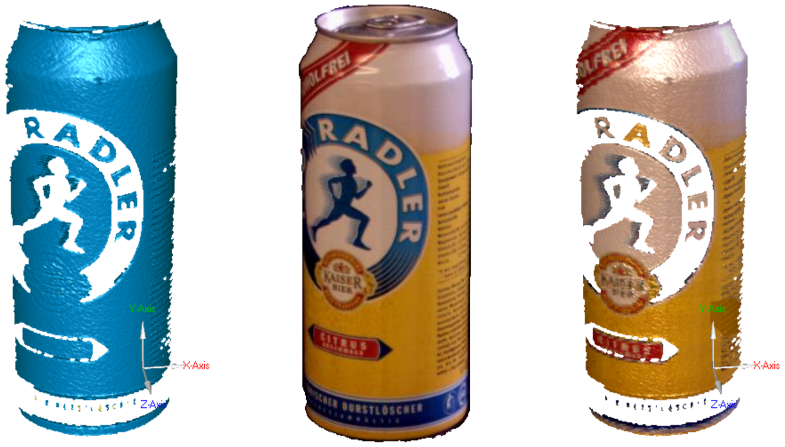

Figure 2.3: Scanned view of a beverage can
(left: only vertices / center: digital photography taken while scanning / right: vertices with associated color information)

Although this additional information per vertex therefore is often easily available, it has to be handled with caution, or more precisely it needs to be treated using the proper methods. And although it often is used as a straightforward enhancement with an additional geometric dimension in image processing, even with some degree of empirical success, this additional dimension is in its principle incommensurable with the geometric dimensions. Therefore, classical Euclidean geometry in principle does not apply [KvD02].

Nevertheless the additional dimension of independent information can be handled separately and set in relation between multiple views. This is shown in principle using a simple approach for the planar and the more interesting spatial case.

2.2 Registration with Acquired Texture in 2D

2.2.1 Theory of Registr. w. Acquired Texture in 2D

Although of less importance and interest than the spatial case, for demonstrative purposes first the planar case is covered.

Every vertex $\mathbf{p}$ of a curve in addition to its Cartesian coordinates $\mathbf{x} = (x, y)$ also possesses one or three scalar values f or f_R, f_G, f_B (for the red-, green- and blue-values), respectively. In the following only one scalar value (i.e., the greyscale) per point shall be considered, in the case of RGB images it has to be handled differently. In a simplistic approach the texture part of the optimization could just be tripled for the three independent components and each weighted by one third.

Changes in lighting and viewpoint will inevitably have complex influence on the greyscale value acquired for identical points on a surface. This has to be taken into account when, in addition or alternatively to geometrical distances, one tries to minimize the 'greyscale distance' of corresponding points. With these complex relations at hand, we will restrain to a simple, linear model in estimating changes of the greyscale values.

Let us now consider two sequences of points in a plane, representing pieces of a planar curve. We will denote the two point sequences as the moving system (points $\mathbf{x}_i$) and the fixed system (points $\mathbf{y}_i$), respectively. Each of the points — in addition to its coordinate values in the xy plane — also has a scalar value, say a greyscale value as an example for an 'acquired texture'. 'Acquired texture' here means a surface color texture (e.g., taken from the surface with some scanning device), in contrast to 'geometry inherent texture' as for example an integral invariant descriptor, see chapter 3.

For each pair of points $\mathbf{x}_i, \mathbf{y}_i$ found to be corresponding (by geometric distances) we expect also their greyscale values $f(\mathbf{x}_i)$ and $f(\mathbf{y}_i)$ to be similar. But ss mentioned before, even if the two points represent the same location on a physical object, due to different viewpoints and probably also different lighting conditions, the greyscale values can, and generally will, differ. Consequently along with the geometrical transformation of the moving curve we also will apply a certain re-coloration of its texture.

In the following we will describe a registration algorithm for the two point sequences $\mathbf{x}_i, \mathbf{y}_i$ where the only texture differences are minimized, not the geometric distances. However, the method described below can be easily combined with geometric methods as for instance squared distance minimization (SDM) (see [PHYH04]).

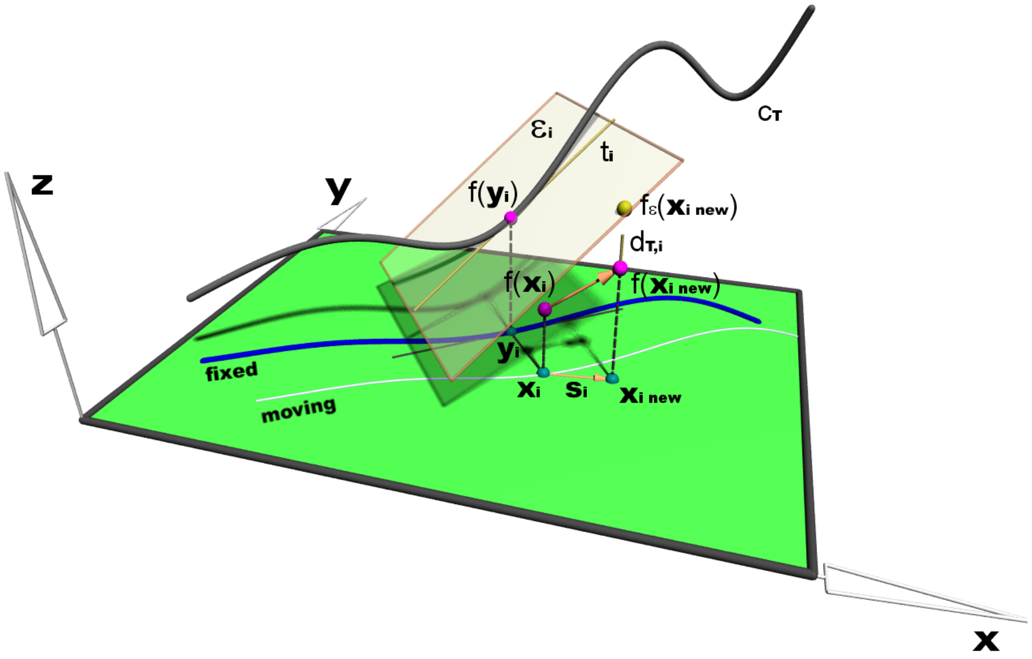

Figure 2.4: Minimization of greyscale value differences

In fig. 2.4 the light curve (in white) represents the moving curve segment which contains the points $\mathbf{x}_i$. The dark curve (in blue) represents the fixed curve segment containing the points $\mathbf{y}_i$. Both curves lie in the xy-plane. The z axis represents the greyscale values, therefore the spatial curve c_T visualizes the fixed curve $\mathbf{y}_i$ together with its greyscale values $f(\mathbf{y}_i)$.

We now consider a single point $\mathbf{x}_i$ of the moving curve. Let $\mathbf{y}_i$ denote its footpoint on the fixed curve, i.e., $\mathbf{y}_i$ is the geometrically closest point to $\mathbf{x}_i$ on the fixed curve. After one iteration of our registration algorithm, the point $\mathbf{x}_i$ will be transformed to a new position $\mathbf{x}_{inew}$. Let the difference vector be denoted by $\mathbf{s}_i = \mathbf{x}_{inew} - \mathbf{x}_i$. Note that also the color value $f(\mathbf{x}_i)$ changes with the transformation to $f(\mathbf{x}_{inew})$, due to re-colorization. We will apply a linear model for this re-colorization, which will be later given in equation (2.8).

Now we want to determine the new position $\mathbf{x}_{inew}$ of the moving point $\mathbf{x}_i$ such that its grey value corresponds to the grey value of the fixed point $\mathbf{y}_i$ in the following way: Let t_i denote the tangent line in the point $(\mathbf{y}_i, f(\mathbf{y}_i))$ of the corresponding spatial curve c_T. Note that here we assume that the fixed curve and its texture are smooth and differentiable. Let ϵ_i denote a plane that passes through the tangent t_i and is parallel to the vector $\mathbf{y}_i - \mathbf{x}_i$, see fig. 2.4. Now we consider the transformed grey value $f(\mathbf{x}_{inew})$ to be in good correspondence to the respective grey value $f(\mathbf{y}_i)$, if the distance $d_{T,i}$ (T for texture) of $f(\mathbf{x}_{inew})$ to the plane ϵ_i is small. Note that we will measure $d_{T,i}$ in vertical direction, i.e., in z-direction,

With our considerations for a single point $\mathbf{x}_i$, which is depicted in fig. 2.4, we have motivated our objective function of our least squares optimization step:

$$F_T = \sum_{i=1}^{N} (d_{T,i})^2. \tag{2.1}$$

Let us first look at the geometrical transformation that maps $\mathbf{x}_i = (x_{1,i}, x_{2,i})$ to $\mathbf{x}_{inew}$. This is a rigid transformation

$$\mathbf{x}_{inew} = \begin{pmatrix} a \\ b \end{pmatrix} + \begin{pmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{pmatrix} \mathbf{x}_i, \tag{2.2}$$

with unknowns a, b, ϕ.

Since we have small rotation angles ϕ we linearize the above motion and obtain an affine motion

$$\mathbf{x}_{inew} = \begin{pmatrix} a \\ b \end{pmatrix} + \begin{pmatrix} 1 & -\phi \\ \phi & 1 \end{pmatrix} \mathbf{x}_i. \tag{2.3}$$

The difference vector $\mathbf{s}_i = \mathbf{x}_{inew} - \mathbf{x}_i$ therefore is given by

$$\mathbf{s}_i = \begin{pmatrix} a - x_{2,i}\phi \\ b + x_{1,i}\phi \end{pmatrix}. \tag{2.4}$$

Of course we will use the linearized motion (2.3) only for determining the unknowns a, b, ϕ, afterward the points $\mathbf{x}_i$ are displaced with equation (2.2).

Now we intersect the vertical line through $\mathbf{x}_{inew}$ with the plane ϵ_i and we obtain a z-value $f_\epsilon(\mathbf{x}_{inew})$ which we have to compare with the re-colored grey value $f(\mathbf{x}_{inew})$, i.e.,

$$d_{T,i} = f_\epsilon(\mathbf{x}_{inew}) - f(\mathbf{x}_{inew}) \tag{2.5}$$

.

Let us look at the two grey values in detail. First we have

$$f_\epsilon(\mathbf{x}_{inew}) = f(\mathbf{y}_i) + \alpha_i f'(\mathbf{y}_i) \tag{2.6}$$

where $f'(\mathbf{y}_i)$ denotes the first derivative of $f(\mathbf{y})$ with respect to an arc length parameter and α_i denotes the length of the difference vector $\mathbf{s}_i$ in direction of the unit tangent vector $\mathbf{t}_i$ in $\mathbf{y}_i$ to the fixed curve $\mathbf{y}$, i.e.,

$$\alpha_i = \mathbf{s}_i \cdot \mathbf{t}_i. \tag{2.7}$$

For the recolored grey value of $\mathbf{x}_{inew}$ we assume

$$f(\mathbf{x}_{inew}) = f(\mathbf{x}_i) + c + \mathbf{d} \cdot \mathbf{x}_i, \tag{2.8}$$

i.e., an affine mapping where the unknowns are the scalar c and the two-dimensional vector $\mathbf{d} = (d_1, d_2)$.

Substitution of equation (2.4), (2.5), (2.6), (2.7) and (2.8) in equation (2.1) leads to

$$F_T = \sum_{i=1}^{N} (f(\mathbf{y}_i) + f'(\mathbf{y}_i)(t_{1i}(a - x_{2i}\phi) + t_{2i}(b + x_{1i}\phi)) - f(\mathbf{x}_i) - c - d_1 x_{1i} - d_2 x_{2i})^2. \tag{2.9}$$

This is a quadratic function in the unknowns a, b, ϕ, c, d_1, d_2 and its minimization leads to the solution of a linear system of equations. The system of equations consists of the following components:

The solution vector $\mathbf{x}$:

$$\mathbf{x} = \begin{pmatrix} a \\ b \\ \phi \\ c \\ d_1 \\ d_2 \end{pmatrix}, \tag{2.10}$$

the vector **b**:

$$\mathbf{b} = \begin{pmatrix} \sum f'(\mathbf{y}_i)t_{1i}(f(\mathbf{y}_i) - f(\mathbf{x}_i)) \\ \sum f'(\mathbf{y}_i)t_{2i}(f(\mathbf{y}_i) - f(\mathbf{x}_i)) \\ \sum f'(\mathbf{y}_i)(f(\mathbf{y}_i) - f(\mathbf{x}_i))(t_{2i}x_{1i} - t_{1i}x_{2i}) \\ \sum f(\mathbf{x}_i) - f(\mathbf{y}_i) \\ \sum x_{1i}(f(\mathbf{x}_i) - f(\mathbf{y}_i)) \\ \sum x_{2i}(f(\mathbf{x}_i) - f(\mathbf{y}_i)) \end{pmatrix}, \tag{2.11}$$

and the six columns of the coefficient matrix **A**:

$$\mathbf{a}_1 = \begin{pmatrix} \sum f'(\mathbf{y}_i)^2 t_{1i}^2 \\ \sum f'(\mathbf{y}_i)^2 t_{1i}t_{2i} \\ \sum f'(\mathbf{y}_i)^2(t_{1i}t_{2i}x_{1i} - t_{1i}^2 x_{2i}) \\ \sum -f'(\mathbf{y}_i)t_{1i} \\ \sum -f'(\mathbf{y}_i)t_{1i}x_{1i} \\ \sum -f'(\mathbf{y}_i)t_{1i}x_{2i} \end{pmatrix} \tag{2.12}$$

$$\mathbf{a}_2 = \begin{pmatrix} \sum f'(\mathbf{y}_i)^2 t_{1i}t_{2i} \\ \sum f'(\mathbf{y}_i)^2 t_{2i}^2 \\ \sum f'(\mathbf{y}_i)^2(t_{2i}^2 x_{1i} - t_{1i}t_{2i}x_{2i}) \\ \sum -f'(\mathbf{y}_i)t_{2i} \\ \sum -f'(\mathbf{y}_i)t_{2i}x_{1i} \\ \sum -f'(\mathbf{y}_i)t_{2i}x_{2i} \end{pmatrix} \tag{2.13}$$

$$\mathbf{a}_3 = \begin{pmatrix} \sum f'(\mathbf{y}_i)^2(t_{1i}t_{2i}x_{1i} - t_{1i}^2 x_{2i}) \\ \sum f'(\mathbf{y}_i)^2(t_{2i}^2 x_{1i} - t_{1i}t_{2i}x_{2i}) \\ \sum f'(\mathbf{y}_i)^2(t_{1i}x_{2i} - t_{2i}x_{1i})^2 \\ \sum f'(\mathbf{y}_i)(t_{1i}x_{2i} - t_{2i}x_{1i}) \\ \sum f'(\mathbf{y}_i)(t_{1i}x_{1i}x_{2i} - t_{2i}x_{1i}^2) \\ \sum f'(\mathbf{y}_i)(t_{1i}x_{2i}^2 - t_{2i}x_{1i}x_{2i}) \end{pmatrix} \tag{2.14}$$

$$\mathbf{a}_4 = \begin{pmatrix} \sum -f'(\mathbf{y}_i)t_{1i} \\ \sum -f'(\mathbf{y}_i)t_{2i} \\ \sum f'(\mathbf{y}_i)(t_{1i}x_{2i} - t_{2i}x_{1i}) \\ N \\ \sum x_{1i} \\ \sum x_{2i} \end{pmatrix} \tag{2.15}$$

$$\mathbf{a}_5 = \begin{pmatrix} \sum -f'(\mathbf{y}_i)t_{1i}x_{1i} \\ \sum -f'(\mathbf{y}_i)t_{2i}x_{1i} \\ \sum f'(\mathbf{y}_i)(t_{1i}x_{1i}x_{2i} - t_{2i}x_{1i}^2) \\ \sum x_{1i} \\ \sum x_{1i}^2 \\ \sum x_{1i}x_{2i} \end{pmatrix} \tag{2.16}$$

$$\mathbf{a}_6 = \begin{pmatrix} \sum -f'(\mathbf{y}_i)t_{1i}x_{2i} \\ \sum -f'(\mathbf{y}_i)t_{2i}x_{2i} \\ \sum f'(\mathbf{y}_i)(t_{1i}x_{2i}^2 - t_{2i}x_{1i}x_{2i}) \\ \sum x_{2i} \\ \sum x_{1i}x_{2i} \\ \sum x_{2i}^2 \end{pmatrix} \tag{2.17}$$

Two remarks have to be made about this method. First, without additional measures employing such a differentiating algorithm of course can only be applied successfully to suitable, that is smooth textures. Second, when analyzing the working principle of this approach, it becomes clear it only can achieve tangential movement. The reason is that it only minimizes texture differences. For illustration: in the hypothetical case of a straight line owning a significant, suitable texture with an identical copy translated by some distance in a direction perpendicular to the line, the algorithm would not decrease the distance between the lines, since it would find the corresponding texture values (that is the texture values of a point and its footpoint) to be perfectly matching without a transformation needed. It can however be combined with a geometry based method in order to achieve the movement normal to the tangent plane.

2.2.2 Results of Registr. w. Acquired Texture in 2D

Test Cases

As an example of a smooth texture the well known picture 'peppers512x512' has been used (fig. 2.5). A synthetic planar curve has been laid on it giving each point of the curve's point set the underlying greyscale value as its texture (fig. 2.5 shows this for the noisy subset of the smooth example curve used).

General Case

This case employs a smooth planar curve constructed as interpolating cubic spline curve with control points (0,0), (1.5,1), (2,-1), (4,1), (3,2), and (5,5). The fixed data set comprises approximately 500 points, while the mobile data set approximately 100 points with noise added. The mobile data set therefore is not a subset of the bigger, fixed one. The transformation applied to the mobile data set to attain the initial position was a rotation by $\pi/10$ about the origin (0,0) and a translation by $x = 1.5$ and $y = -1.0$.

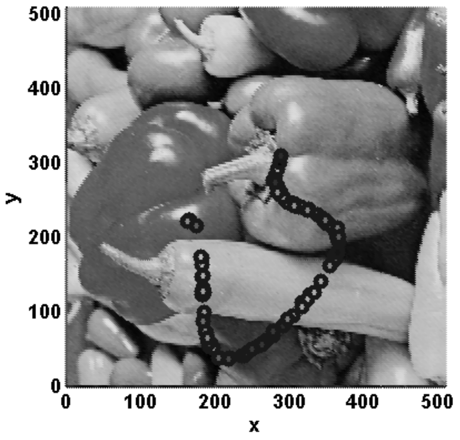

Figure 2.5: Acquisition of greyscale values from image 'peppers512x512' (Drawn onto the picture are the points of the mobile spline data set used in the 'general' case below)

For comparison the registration also was conducted using straightforward ICP and curvature based method. Not surprisingly the curvature based method converges fastest, but although being considerably slower, the purely texture based method achieves convergence within ten iterations (fig.2.6).

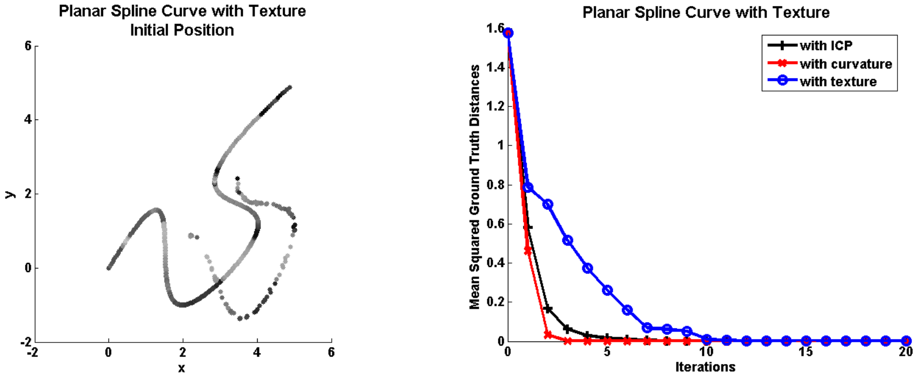

Figure 2.6: Registration of planar spline curve with texture

'Kinematic' Case

Texture based registration is able to show its potential with the following data set. The fixed one is a circle segment comprising approximately 200 points and the mobile one approximately 100 points with noise added. Again the mobile set is not a subset of the fixed one. The transformation applied is the same as for the general case above.

Here, as could be foreseen, the ICP based method clings to the closest geometrical minimum and does not reverse the applied rotation. The curvature method (presumably by chance or possibly due to the deviations of the shape from a perfect circle) gets it better but also then stands still without achieving the full rotation needed. Different so the texture based method which is slowest but achieves the best convergence (fig.2.7).

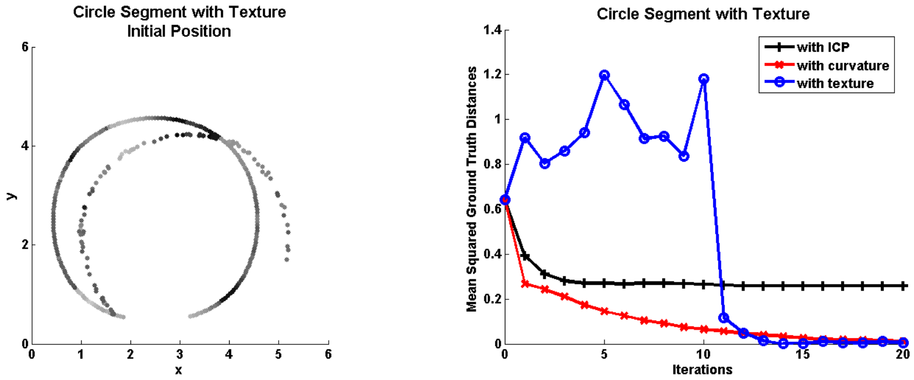

Figure 2.7: Registration of circle segment with texture

2.3 Registration with Acquired Texture in 3D

2.3.1 Theory of Registr. w. Acquired Texture in 3D

The application of the 2D concept in section 2.2.1 to the 3D case will be briefly discussed here. The moving point cloud $\mathbf{x}_i$ is surface like, as well as the fixed point cloud $\mathbf{y}_i$. $\mathbf{x}_i$ and $\mathbf{y}_i$ with the same index i are again corresponding points.

First of all the linearization of a 3D motion can be described by

$$\mathbf{x}_{inew} = \bar{\mathbf{c}} + \mathbf{c} \times \mathbf{x}_i, \tag{2.18}$$

where the new position of $\mathbf{x}_i$ depends linearly on the unknown 3D vectors $\bar{\mathbf{c}}$ and $\mathbf{c}$, see [PW01].

The difference vector $\mathbf{s}_i = \mathbf{x}_{inew} - \mathbf{x}_i$ also depends linearly on $\bar{\mathbf{c}}$ and $\mathbf{c}$.

The re-coloration $f(\mathbf{x}_{inew})$ is analogous to the 2D case

$$f(\mathbf{x}_{inew}) = f(\mathbf{x}_i) + c + \mathbf{d} \cdot \mathbf{x}_i, \tag{2.19}$$

with unknowns $c, \mathbf{d} = (d_1, d_2, d_3)$.

For the equivalent of equation (2.6) we proceed as follows. Let $\mathbf{t_1}$ and $\mathbf{t_2}$ define two orthogonal, normalized vectors in the tangent space of the fixed surface shape in the point $\mathbf{y}_i$. With respect to these two vectors we can write the greyscale gradient

$$\nabla f(\mathbf{y}_i) = f_1 \cdot \mathbf{t_1} + f_2 \cdot \mathbf{t_2}. \tag{2.20}$$

With this notation we have

$$f_\epsilon(\mathbf{x}_{inew}) = f(\mathbf{y}_i) + \alpha_1 \mathbf{t_1} + \alpha_2 \mathbf{t_2}, \tag{2.21}$$

with

$$\alpha_1 = \mathbf{s}_i \cdot \mathbf{t}_1, \quad \alpha_2 = \mathbf{s}_i \cdot \mathbf{t}_2. \tag{2.22}$$

Since the objective function is completely analogous to the planar case

$$F_T = \sum_{i=1}^{N} (d_{T,i})^2 = \sum_{i=1}^{N} (f(\mathbf{x}_{i,\epsilon}) - f(\mathbf{x}_{inew}))^2, \tag{2.23}$$

we again have a quadratic function in the unknowns $\overline{\mathbf{c}}, \mathbf{c}, c, \mathbf{d}$ to minimize.

2.3.2 Results of Registr. w. Acquired Texture in 3D

The verification and demonstration in 3D is given for a kinematic shape. The (0.5l) beverage can depicted in fig. 2.8 has been scanned using the scanner and a turntable. Two overlapping views, differing from each other by a rotation of 30 ° have been chosen.

The ground truth (GT) reference is not defined by the initial position delivered by scanner and turntable but is the result of manual n-by-n point registration using distinctive features on both views followed by an automatic registration as fine-tuning. Then the initial position of the experiment is achieved by turning one point set away from the other by 10 ° approximately around the common axis of rotation (simulating an exaggerated angle error of the turntable) and then moving it 5mm radially away from this axis.

This object originally has been chosen due to its clear and distinguished surface pattern. But when viewed in conjunction with the gradient based

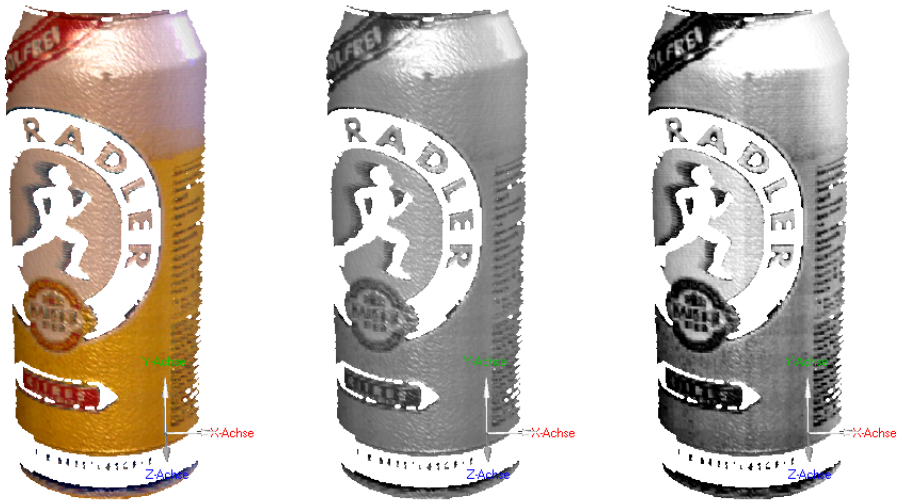

Figure 2.8: Beverage can as kinematic shape example
(left: original scan, center: greyscale view, right: contrast enhanced)

registration method it has to be noted that texture transitions on it in most cases are not smooth. And when converted to greyscale the situation becomes even worse since the main differently colored areas are reduced to similar values of grey (see middle picture of fig. 2.8). Still the algorithm performs distinctively better than the purely geometry based methods (fig. 2.9).

Due to the difficult nature of the registration problem 30 instead of the otherwise 20 iterations are shown. Pure minimization of footpoint distances (i.e., ICP) for this shape shows very clearly the imminent problem (top row of fig. 2.9). The free point set very fast clings to the surface and then slides on it not stopping at the correct position since it is not a geometric minimum, neither global nor local. Consequently the ground truth graph descends to a minimum and afterward ascends again with roughly the same speed. Presumably the movement will stop (in a setup with only two point sets) when they fully overlap.

Minimization of distances to tangent planes and to second order surface approximations roughly do the same alas with a smaller movement (second and third row of fig. 2.9). The texture based method on the other hand not only converges fastest it also stops at the minimum (bottom row of fig. 2.9). The left column shows the associated final positions after 30 iterations in a projected view. For the texture based method it shows a distinctive difference, a gap between the surfaces. Nevertheless this fits well to the inability to translation normal to a surface. A remedy could be a weighted

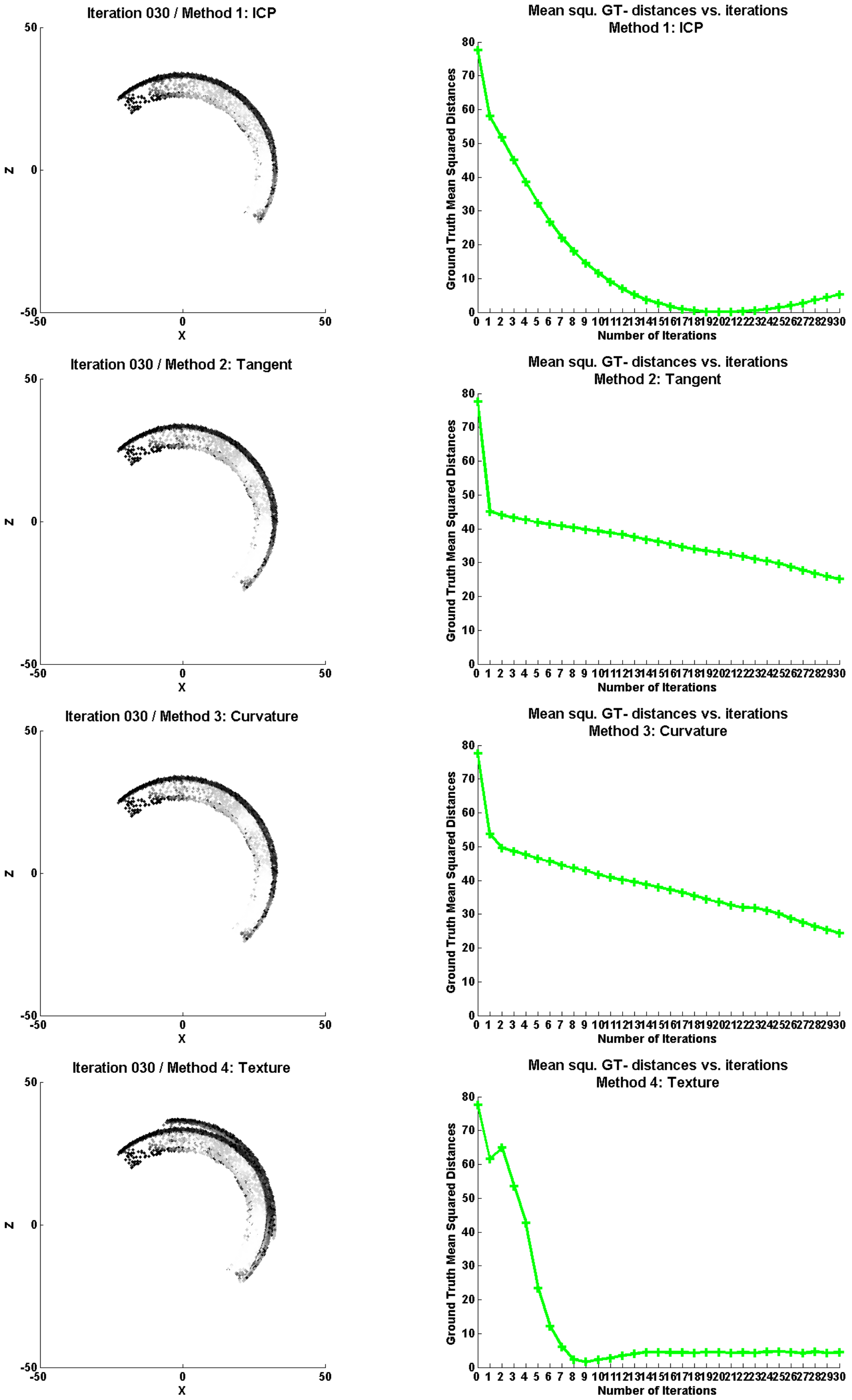

Figure 2.9: Registration of kinematic shape with texture

combination with a purely geometric method with a low weight for the latter.

That leaves the question about the (computational) costs of the various methods. The experiments have been carried out using a non-optimized implementation in Matlab. In relation to ICP the computation time of 30 iterations, with the results as presented, where:

- ICP based 100%
- Tangent based 302%
- Curvature based 793%
- Texture based 639%

Chapter 3

Geometry - Inherent Texture

3.1 Geometry - Inherent Texture in 2D

3.1.1 Integral Invariant Descriptors in 2D

In [MHYS04] Manay et al. present for closed planar contours two representatives of a class of functionals which are invariant with respect to Euclidean and similarity transformations. Those functionals are obtained using integral operations and the authors propose their use as a basis to define various notions of distances between shapes. They share some useful properties with differential invariants, but what especially distinguishes them from the latter is also an important motivation for their use: integral invariants are far less sensitive to noise in the data compared to their differential counterparts.

Another of their useful features is the intrinsic scale space that is easily established by varying the kernel measure used. This allows an analysis at multiple levels of resolution and accompanied with it, under different levels of measurement noise. In the above mentioned paper the authors show for one class of integral invariant functionals their uniqueness of representation in the limit of vanishing kernel measure. This limit corresponds to curvature and therefore relates this integral invariant to differential invariants, a fact that enables the use of existing literature on differential invariants for theoretical results.

The two representatives have been named ’integral distance invariant’ and ’integral area invariant’ by the authors. For the ’integral area invariant’ they give an approximation concerning the relation with curvature and therefore with local differential invariants. But really this approximation describes a circle segment connected to the so called 'circular arc invariant', see [PHYK05].

While in chapter 4 applicability of said descriptors for the support of rigid registration is examined, this chapter covers the definition and certain characteristics of a selected range of integral invariants for planar curves and (in the following section) for spatial surfaces.

3.1.2 The Integral Distance Invariant

The following is based on closed planar contours, but is applicable to open contours as well. In this case however some considerations have to be made, considerations about uniquely defining the two sides of the contour (since 'inside' and 'outside' do not apply) and about the handling of the two open ends. Whereas in the 2D section no special treatment of these boundary points and adjacent curve segments has been applied, later in the more application oriented 3D section, an edge region defined by the kernel measure is left without descriptors.

Nevertheless here we consider a closed planar contour that shall be defined by a function $\gamma : \mathbb{S}^1 \longrightarrow \mathbb{R}^2$ with arc length ds. Then the integral distance invariant for a point $p \in \gamma$ is defined as

$$I_\gamma(p) \doteq \int_\gamma d(p,x)ds(x). \tag{3.1}$$

Here $d(x,y) \doteq ||y-x||$ is the Euclidean distance measure in $\mathbb{R}^2$. Therefore $I_\gamma(p)$ delivers for the point p as value the average distance to every other point on the contour. This clearly is invariant under Euclidean transformations. The invariant can be localized by using a defined neighborhood or 'kernel' in the form of a circle with radius r which is centered in the point p. The integrand would then be weighted by this kernel $q(p,x)$ leading to

$$I_\gamma(p) \doteq \int_\gamma q(p,x)d(p,x)ds(x). \tag{3.2}$$

In the following this kernel represents the indicator function with value 1 inside the kernel and 0 elsewhere, such that (3.2) equals the average distance to every other point inside the kernel as integral invariant value. Nevertheless the kernel can be defined tailored to ones needs, just one thinkable example would be a Gaussian kernel.

Fig. 3.2 shows the integral distance invariant values for the planar curve depicted in fig. 3.1, left. The right picture shows the results for a noisy data set with normally distributed noise with $\sigma = 1$ and an amplitude of $+/-0.005$. It shows clearly that the overall shape of the curve of integral invariant values stays constant in the presence of noise and does not amplify the disturbances as differential invariants would.

Similarly and with similar results fig. 3.3 depicts the results for clean and noise-added data set for the closed edgy contour on the right side of fig. 3.1. Fig. 3.4 finally shows for the two contours the results for a range of kernel radii, normalized by kernel radius $(I_\gamma(p)/r)$.

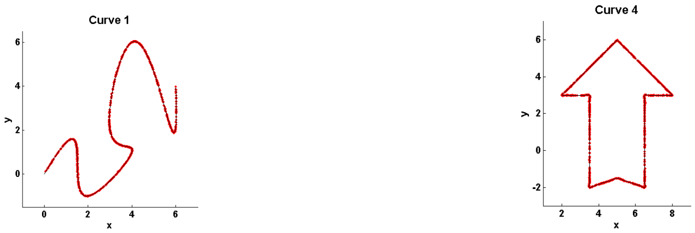

Figure 3.1: Smooth, open (left) and edgy, closed (right) planar curve

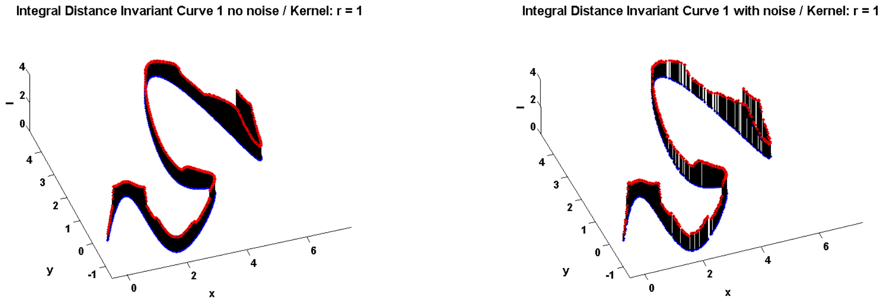

Figure 3.2: Integral distance invariant for a smooth planar curve without (left) and with (right) noise

3.1.3 The Integral Area Invariant

The 'integral area invariant' simply is the intersection area of the circular kernel and the planar domain enclosed by the closed curve γ (respectively the intersection of the kernel with one uniquely defined side of the plane in the case of an open curve). Therefore it is defined as

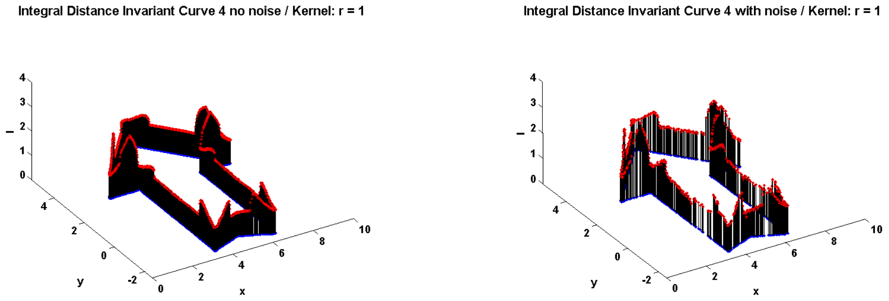

Figure 3.3: Integral distance invariant for an edgy planar curve without (left) and with (right) noise

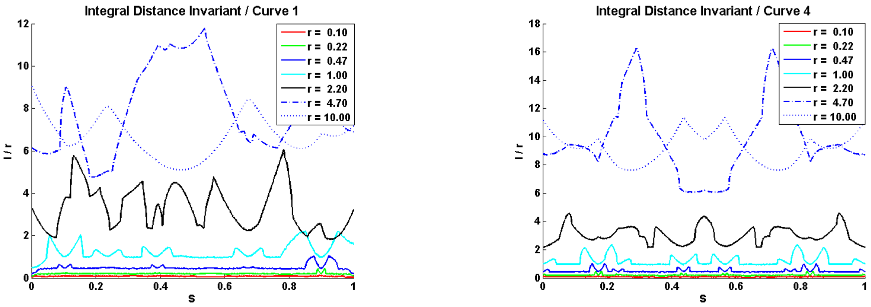

Figure 3.4: Normalized integral distance invariant curves for a smooth (left) and an edgy (right) planar curve for a range of kernel radii

$$I_\gamma^r(p) \doteq \int_{B_r(p)\cap\bar{\gamma}} dx \tag{3.3}$$

whith $\bar{\gamma}$ as the interior of the curve, dx the corresponding measure of this two-dimensional object and a kernel $h(p,x) = \chi(B_r(p) \cap \bar{\gamma})(x)$. As briefly mentioned before, the first order approximation given by Manay et al.

$$I_\gamma^r(p) \simeq r^2 \arccos\left(\frac{1}{2} r\kappa(p)\right) \tag{3.4}$$

(for small kernel radius r and with curvature κ in the point p) really is the circular sector spanned by the circular arc on the boundary of the kernel disc spanning between the two intersection points between kernel circle and contour γ. The length of this circular arc is the 'integral circular arc invariant' and it is the derivative of the area invariant with respect to the kernel radius. Interestingly the underlying angle that spans said arc has been used before by Conolly for a quite different purpose, which is the classification of protein surface shapes [Con86]. The correct version of the relation of integral area invariant with curvature reads

$$I_\gamma^r(p) = \frac{\pi}{2}r^2 - \frac{\kappa}{3}r^3 + O(r^4). \tag{3.5}$$

Keeping this in mind due to its advantages in terms of computational costs we nevertheless use the circular sector as approximation for an integral area invariant in the following figures. Fig. 3.5 again illustrates for the smooth, open contour the insensibility of this class of functionals with respect to noise. Fig. 3.6 shows the normalized values for a range of kernel radii.

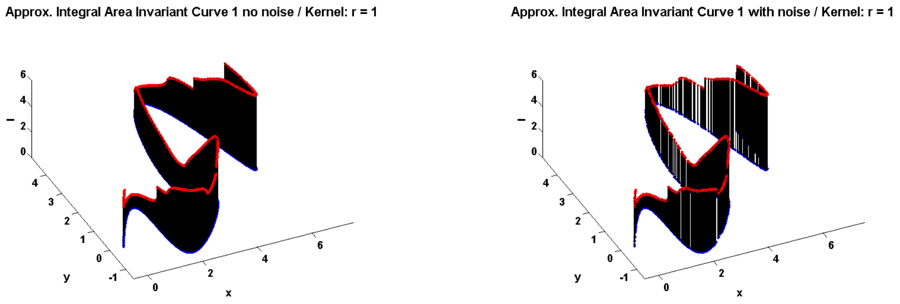

Figure 3.5: Approximated integral area invariant for a smooth planar curve without (left) and with (right) noise

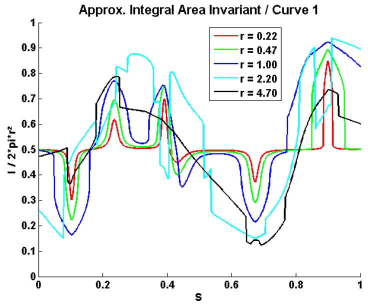

Figure 3.6: Normalized approximated integral area invariant curves for a smooth planar curve for a range of kernel radii

Normalization is done by division through the area of the kernel disk $2\pi r^2$ (respectively half of it as did Manay et al.). Fig. 3.6 shows this for the open contour illustrating that for small kernel radii this value tends toward 1/2, which is the result for the intersection with the tangent in point p.

3.2 Geometry - Inherent Texture in 3D

3.2.1 Integral Invariant Descriptors in 3D

The extension of the integral invariants from planar curves to surfaces in $\mathbb{R}^3$ is quite natural. If we use a ball where we used a circular disk in $\mathbb{R}^2$ respectively a sphere instead of a circle we get the appropriate kernels for the spatial domain. Consequently the integral area invariant becomes an integral volume invariant.

In addition to the more general variants, three more integral invariant descriptors shall be presented here, which all share one common feature: they all are based on the intersection curve of the kernel sphere and the surface to be described. They incorporate the length of said intersection, the distance of the intersection's barycenter to the point p, and the two bigger eigenvalues of a principal component analysis (PCA) of the intersection curve respectively.

3.2.2 The Intersection Descriptor

We start with the following situation: in a point of interest p on a three-dimensional surface we place the center of a kernel ball $B_r(p)$. Differing from the integral volume invariant here we are only interested in the intersection curve between the kernel ball and the given surface or more precisely in its length.

Theory of the Intersection Desriptor

For a theoretical analysis we replace the surface in question at the point p of consideration by a second order Taylor approximation. The coordinate frame shall be the principal frame at p. Therefore the paraboloid P, approximating the given surface up to second order is expressed by

$$z_p = \frac{1}{2}\left(\kappa_1 x^2 + \kappa_2 y^2\right), \tag{3.6}$$

where κ_1 and κ_2 are the principal curvatures, while the kernel ball has the equation

$$z_s^2 = r^2 - x^2 - y^2. \tag{3.7}$$

We want to intersect paraboloid and sphere, but have to bear in mind, that it is not yet sufficient to reduce the kernel to a half-ball as the principal curvatures can assume opposite signatures. In order not to neglect this case with a surface of saddle-shape, $F(x, y)$ for the intersection curve's projection onto the xy-plane has to be

$$\left(\frac{\kappa_1 x^2}{2} + \frac{\kappa_2 y^2}{2}\right)^2 = r^2 - x^2 - y^2. \tag{3.8}$$

The intersection curve itself clearly lies on a sphere, namely the kernel ball, consequently it is advisable to use spherical coordinates. An alternative approach using cylindrical coordinates produces the same results, see [PHYK05]. With

$$x = r\cos\varphi\cos\lambda \tag{3.9}$$

$$y = r\cos\varphi\sin\lambda \tag{3.10}$$

$$z = r\sin\varphi, \tag{3.11}$$

equation (3.8) becomes

$$\begin{aligned}\left(1/2\kappa_1 r^2\cos^2(\varphi)\cos^2(\lambda) + 1/2\kappa_2 r^2\cos^2(\varphi)\sin^2(\lambda)\right)^2 &= \\ r^2 - r^2\cos^2(\varphi)\cos^2(\lambda) - r^2\cos^2(\varphi)\sin^2(\lambda)&\end{aligned} \tag{3.12}$$

and with simplification of the trigonometric expressions and dividing by r^2 the equation reads

$$r^2\left(1/2\kappa_1\cos^2(\varphi)\cos^2(\lambda) + 1/2\kappa_2\cos^2(\varphi)\sin^2(\lambda)\right)^2 = \sin^2(\varphi). \tag{3.13}$$

This equation naturally has four solutions $\varphi_{1,2,3,4} = f(\lambda)$, two of which are real, complementing each other to the amount of π and therefore being mirror images of each other and allowing us to randomly pick one of them. Inserted into the equations (3.9) to (3.11) this delivers a parametrization of the intersection curve only by one parameter; λ. To calculate the arc-length of the intersection curve, its line integral has to be solved, in our case over the range of λ:

$$I^r = \int \sqrt{\dot{x}^2(\lambda) + \dot{y}^2(\lambda) + \dot{z}^2(\lambda)}d\lambda. \tag{3.14}$$

An precise closed solution of this expression which, after parametrization by λ and building the first derivations, becomes rather complex, is not necessary. It is enough to use approximates of sufficiently high order for the expressions of x, y, z as functions in λ. We are mainly interested in the second and third order terms of the approximation for the intersection curve's length, therefore we use Taylor approximates for $r = 0$ up to fifth order to be sure. After first derivation with respect to λ in a second step we do the same again with the integrand

$$i_r = \sqrt{\dot{x}^2 + \dot{y}^2 + \dot{z}^2} \tag{3.15}$$

and replace it by a Taylor approximation for $r = 0$ up to O(5). This leaves us with an expression of sufficiently low complexity for symbolic integration

$$I^r = 4 \cdot \int\limits_0^{\pi/2} (i_{r-approx})\, d\lambda \tag{3.16}$$

(symmetry permits to perform integration for one quadrant from $\lambda = 0$ to $\lambda = \pi/2$ only and multiply the resulting length by four) resulting in

$$I^r = 2\pi r + \frac{1}{32}\pi \left(\kappa_1^2 - 10\kappa_1\kappa_2 + \kappa_2^2\right) r^3 + O\left(r^5\right). \tag{3.17}$$

As easily foreseeable, the first order term represents the circular intersection between the kernel ball and a plane going through its center, i.e., the case of vanishing curvature κ_1 and κ_2. The third oder term also shows symmetry with respect to κ_1 and κ_2. It can not only assume negative but also positive values, the latter in the case of principal curvatures with opposite signatures since it incorporates Gaussian curvature $\kappa_1\kappa_2$. This invariant is less attractive in terms of stability in the presence of noise than the integral barycenter invariant covered next. The reason is that small perturbations add up to the length of the intersection curve while (when evenly distributed) the barycenter largely stays unchanged. The extrema consequently are:

- $I^r = 2\pi r$ for $\kappa_1 = \kappa_2 = 0$
- $I^r = 0$ for $\kappa_1 = \kappa_2 = \infty$ and $\kappa_1 = \kappa_2 = -\infty$
- $I^r = 4\pi r$ for $\kappa_1 = -\kappa_2 = \infty$ and $\kappa_1 = -\kappa_2 = -\infty$.

Omitting fifth order terms in (3.17) is feasible only for small values of the kernel radii. Table 3.1 shows the dependency of the integral intersection invariant on the kernel size for various cases. In this table, I^r_{sym} is the solution obtained by symbolic integration, if possible for the given values of kernel radius and curvatures, I^r_{num} is the result of a numerical solution implemented in Matlab using an increment size of $\Delta\lambda = \pi/20.000$ and I^r_{app} is the value of the approximation using equation (3.17), omitting fifth order terms. Q finally stands for the ratio I^r_{app}/I^r_{num} (for small values of r, Q tends to 1).

Kernel - Radius:			**1e1**	**1.00**	**1e-1**	**1e-2**	**1e-3**	**1e-4**	**1e-5**
κ_1	κ_2								
Case a): Close to intersection with a plane									
8e-3	8e-3	I^r_{sym}	6.28e1	6.28e0	-	6.08e-2	-	-	-
		I^r_{num}	6.28e1	6.28e0	6.28e-1	6.28e-2	6.28e-3	6.28e-4	6.28e-5
		I^r_{app}	6.28e1	6.28e0	6.28e-1	6.28e-2	6.28e-3	6.28e-4	6.28e-5
		Q	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Case b): Near-infinite curvatures									
1e10	1e10	I^r_{sym}	2.81e-4	8.89e-5	2.81e-5	8.89e-6	2.81e-6	8.89e-7	2.81e-7
		I^r_{num}	2.81e-4	8.89e-5	2.81e-5	8.89e-6	2.81e-6	8.89e-7	2.81e-7
		I^r_{app}	-e23	-e20	-e17	-e14	-e11	-e8	-e5
		Q	-e27	-e24	-e22	-e19	-e17	-e14	-e12
Case c): Near-infinite curvatures with opposite signatures									
1e10	-1e10	I^r_{sym}	-	-	-	-	-	-	-
		I^r_{num}	1.13e2	1.13e1	1.13e0	1.13e-1	1.13e-2	1.14e-3	1.15e-4
		I^r_{app}	e23	e20	e17	e14	e11	e8	e5
		Q	e21	e19	e17	e15	e13	e11	e9
Case d): Rotational paraboloid									
15	15	I^r_{sym}	7.23e0	2.22e0	5.31e-1	6.27e-2	6.28e-3	-	6.21e-5
		I^r_{num}	7.23e0	2.22e0	5.31e-1	6.27e-2	6.28e-3	6.28e-4	6.28e-5
		I^r_{app}	-e5	-e2	4.52e-1	6.27e-2	6.28e-3	6.28e-4	6.28e-5
		Q	-e5	-e2	0.85	1.00	1.00	1.00	1.00
Case e): General case									
5	0.8	I^r_{sym}	22.3e1	-	6.27e-1	6.28e-2	6.28e-3	6.34e-4	-
		I^r_{num}	22.3e1	5.30e1	6.27e-1	6.28e-2	6.28e-3	6.28e-4	6.28e-5
		I^r_{app}	-e3	4.87e1	6.27e-1	6.28e-2	6.28e-3	6.28e-4	6.28e-5
		Q	-e2	0.92	1.00	1.00	1.00	1.00	1.00
Case f): General case with bigger difference in principal curvatures									
10	0.1	I^r_{sym}	4.64e1	6.23e0	6.34e-1	6.28e-2	-	-	-
		I^r_{num}	4.64e1	6.23e0	6.34e-1	6.28e-2	6.28e-3	6.28e-4	6.28e-5
		I^r_{app}	e3	1.51e1	6.37e-1	6.28e-2	6.28e-3	6.28e-4	6.28e-5
		Q	e2	2.43	1.01	1.00	1.00	1.00	1.00
Case g): Saddle / opposite signatures of principal curvatures									
10	-0.5	I^r_{sym}	-	-	-	-	-	-	-
		I^r_{num}	1.02e2	7.28e0	6.39e-1	6.28e-2	6.28e-3	6.28e-4	6.28e-5
		I^r_{app}	e4	2.10e1	6.43e-1	6.28e-2	6.28e-3	6.28e-4	6.28e-5
		Q	e2	2.89	1.01	1.00	1.00	1.00	1.00

Table 3.1: Integral intersection invariant vs. kernel radii for various cases

Table 3.1 clearly shows that for big values of kernel the radius the approximation fails for all cases except for the trivial one, the intersection with a plane or for a situation close to it as demonstrated in case a). Since curvature depends on the scale and all cases resemble this trivial one for small kernel radii, it is as expected that there the approximation improves in accuracy. The results of the numerical solution confirm what has been mentioned before about the limiting values. For the cases a) and d) to g) the values tend toward $2\pi r$, for case b) toward zero whereas for case c) the tendency is toward $4\pi r$.

Implementation for triangulated point cloud data

The practical implementation that shall be considered here is oriented at the application of the method described on point cloud surfaces. Let us assume that we have acquired a triangulated point cloud, say, by means of a 3D laser scanner. To calculate the integral intersection descriptor value of a point of interest, as a first step a distance field for this point is calculated. It contains the squared distances to each other point in the data set. Based on this an iso-line is established using the kernel radius (or more precise the square of it) as criterion. This means finding the points on triangulation edges, whose distance value, interpolated between the two adjacent grid points, is equal to the kernel radius specified. The result is an ordered iso-poly-line, whose segments are summed up to deliver the integral intersection descriptor for the point in question, if, and only if, the poly-line is closed.

Figure 3.7 shows the intersection of the kernel ball for a point on a data set representing a statue of a cat. The kernel radius here is 20. In the detail on the right side the point of interest is drawn as red circle and the iso-line points on the triangulation edges are drawn as blue stars.

Results

Inherent to the method is that the boundary region is left without descriptor values, see figs 3.8 and 3.9. The reason for this is the necessity to restrict to closed iso-poly-lines in order to enable global comparability. The intersection curve of kernel and surface must not be cut by the edge of the shape since this would distort the integral invariant. Its width is at least the kernel radius. Holes in the data set become equally enlarged in this way. Visible in figures 3.8 and 3.9 are the descriptor values for the data set representing a human face and one of a fruit press for two different kernel radii for the integral intersection invariant as well as the two other invariants covered in the following.

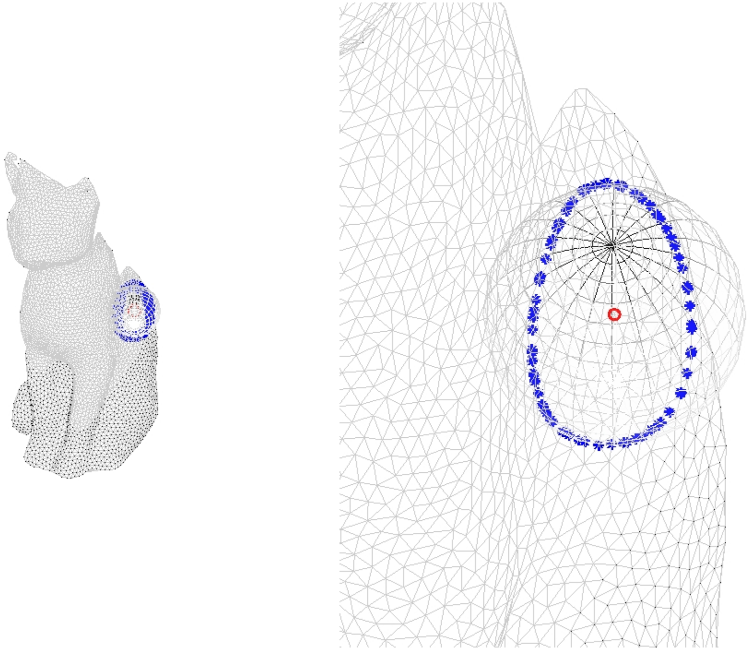

Figure 3.7: Detail of implementation of the integral intersection descriptor The red ring is the point of interest and blue stars are the intersection points between kernel sphere and triangle mesh.

3.2.3 The Intersection-Barycenter Descriptor

What shall be called intersection-barycenter descriptor here is based on the intersection descriptor. Looking for a more robust variant of the intersection descriptor we take the signed distance between the barycenter of the intersection curve and the point of interest p.

Theory of the Intersection-Barycenter Desriptor

We want to know the location of the barycenter of the intersection curve between the surface-approximating paraboloid and the kernel ball in the principal coordinate system centered in the point of interest p. When replacing the surface with a second order Taylor approximation it is obvious that the barycenter lies on the z-axis of this principal coordinate system since the four quadrants of the intersection curve are mirrored equivalents of each other. Therefore when, as in the section before, we use spherical coordinates and integrate over λ, it makes no difference whether we integrate, say, from $-\pi$ to π or from 0 to $\pi/2$. The result is the same. The equation to be solved

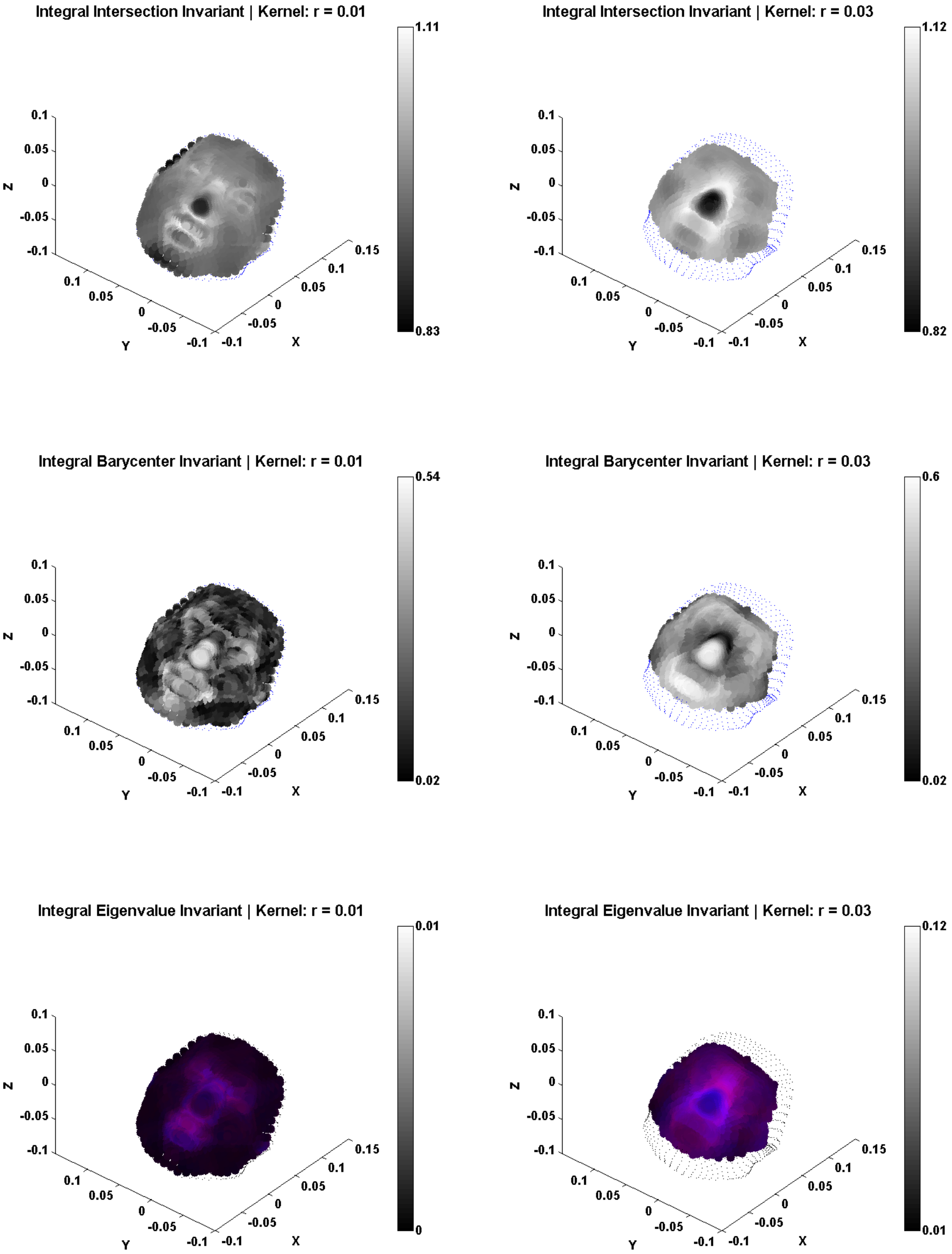

Figure 3.8: Comparison of three integral invariants using the face data set with kernel radii 0.01 and 0.03: top row - intersection invariant / middle row - barycenter invariant / bottom row - eigenvalue invariant

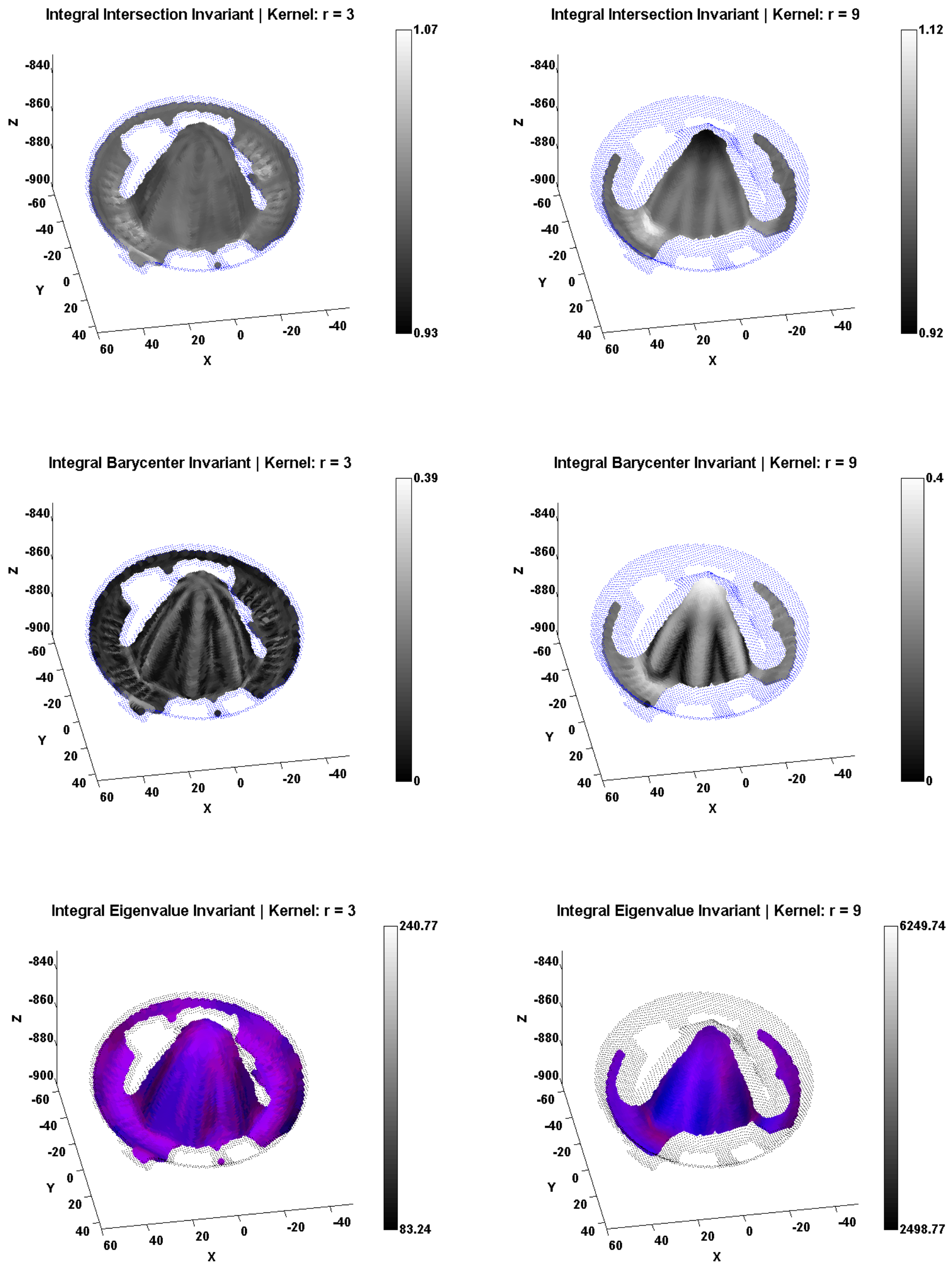
Integral Intersection Invariant | Kernel: r = 3
1.07
0.93
Integral Intersection Invariant | Kernel: r = 9
1.12
0.92
Integral Barycenter Invariant | Kernel: r = 3
0.39
0
Integral Barycenter Invariant | Kernel: r = 9
0.4
0
Integral Eigenvalue Invariant | Kernel: r = 3
240.77
83.24
Integral Eigenvalue Invariant | Kernel: r = 9
6249.74
2498.77
Z
-840
-860
-880
-900
Y
-60
-40
-20
0
20
40
X
60
40
20
0
-20
-40

Figure 3.9: Comparison of three integral invariants using the press data set with kernel radii 3 and 9: top row - intersection invariant / middle row - barycenter invariant / bottom row - eigenvalue invariant

consequently is

$$z_s = \frac{\int \left(z\left(\lambda\right) \cdot \sqrt{\dot{x}^2 + \dot{y}^2 + \dot{z}^2} \right) d\lambda}{\int \left(\sqrt{\dot{x}^2 + \dot{y}^2 + \dot{z}^2} \right) d\lambda} \tag{3.18}$$

The complexity of the expression here is higher than for the pure intersection descriptor and an approximation of Taylor type for $r \longmapsto 0$ up to fifth order is employed one additional time at the final stage, giving for the approximation of the intersection-barycenter

$$z_s = \frac{r^2}{2} \cdot \left(\frac{\kappa_1 + \kappa_2}{2} \right) - \frac{r^4}{64} \cdot \left(3\kappa_1^3 + \kappa_1^2\kappa_2 + \kappa_1\kappa_2^2 + 3\kappa_2^3 \right) + O\left(r^5\right), \tag{3.19}$$

or with the mean curvature H the distance between barycenter and p is

$$d_c^r = z_s = \frac{H}{2} r^2 + O(r^3). \tag{3.20}$$

This is an important point when comparing this invariant to the pure intersection invariant. The latter incorporates Gaussian curvature and therefore can be assumed to show less stable behavior, i.e., a bigger sensitivity to noise.

A numerical and, where possible, symbolical verification in the same way as in the section before proves the validity of this approximation for small values of kernel radii as can be seen in table 3.2.

Implementation for triangulated point cloud data

The implementation is based handily on the implementation of the intersection descriptor. But here just the barycenter for the set of points acquired is calculated. This lowers the computational cost for this invariant. Figure 3.10 shows a detail on the cat data set. There the blue line is the the connection of the point of interest p with the barycenter of the intersection curve (blue stars), its length is the invariant wanted.

Results

This descriptor is in some way complementary to the intersection descriptor described in section 3.2.2. For regions with high principal curvature of

Kernel - Radius:			**1e1**	**1.00**	**1e-1**	**1e-2**	**1e-3**	**1e-4**	**1e-5**
κ_1	κ_2								
Case a): Close to intersection with a plane									
8e-3	8e-3	I^r_{sym}	3.99e-1	5.59e-3	-	-	-	-	-
		I^r_{num}	3.99e-1	4.00e-3	4.00e-5	4.00e-7	4.00e-9	4.00e-11	4.00e-13
		I^r_{app}	3.99e-1	4.00e-3	4.00e-5	4.00e-7	4.00e-9	4.00e-11	4.00e-13
		Q	1.00	1.00	1.00	1.00	1.00	1.00	1.00
		$Conv$	0,94	0	0	0	0	0	0
Case b): Near-infinite curvatures									
1e10	1e10	I^r_{sym}	1e1	1e0	1e-1	1e-2	1e-3	1e-4	1e-5
		I^r_{num}	1e1	1e0	1e-1	1e-2	1e-3	1e-4	1e-5
		I^r_{app}	-e35	-e31	-e27	-e23	-e19	-e15	-e11
		Q	-e34	-e31	-e28	-e25	-e22	-e19	-e16
		$Conv$	e-22	e-20	e-18	e-16	e-14	e-12	e-10
Case c): Near-infinite curvatures with opposite signatures									
1e10	-1e10	I^r_{sym}	-	-	-	-	-	-	-
		I^r_{num}	e-9	e-10	e-12	e-15	e-16	e-20	e-21
		I^r_{app}	0	0	0	0	0	0	0
		Q	-	-	-	-	-	-	-
		$Conv$	-	-	-	-	-	-	-
Case d): Rotational paraboloid									
15	15	I^r_{sym}	9.93e0	9.36e-1	5.35e-2	7.46e-4	7.30e-6	-	-
		I^r_{num}	9.93e0	9.36e-1	5.35e-2	7.46e-4	7.50e-6	7.50e-8	7.50e-10
		I^r_{app}	-e6	-e2	3.28e-2	7.46e-4	7.50e-6	7.50e-8	7-50e-10
		Q	-e5	-e2	0.61	1.00	1.00	1.00	1.00
		$Conv$	e-4	e-2	0.51	0.99	0.95	0	0
Case e): General case									
5	0.8	I^r_{sym}	-	6.21e-1	1.39e-2	1.45e-4	-	-	-
		I^r_{num}	9.52e0	6.53e-1	1.40e-2	1.45e-4	1.45e-6	1.45e-8	1.45e-10
		I^r_{app}	-e4	-4.80e0	1.39e-2	1.45e-4	1.45e-6	1.45e-8	1.45e-10
		Q	-e3	-e1	0.99	1.00	1.00	1.00	1.00
		$Conv$	e-3	0.13	0.8	0.8	0	0	0
Case f): General case with bigger difference in principal curvatures									
10	0.1	I^r_{sym}	-	-	-	-	-	-	-
		I^r_{num}	9.22e0	6.82e-1	2.24e-2	2.52e-4	2.53e-6	2.53e-8	2.53e-10
		I^r_{app}	-e5	-4.45e1	2.05e-2	2.52e-4	2.52e-6	2.52e-8	2.52e-10
		Q	-e4	-e2	0.92	1.00	1.00	1.00	1.00
		$Conv$	e-4	e-2	0.61	0.83	0	0	0
Case g): Saddle / opposite signatures of principal curvatures									
10	-0.5	I^r_{sym}	-	-	-	-	-	-	-
		I^r_{num}	7.16e0	6.09e-1	2.10e-2	2.37e-4	2.38e-6	2.38e-8	2.38e-10
		I^r_{app}	-e5	-4.38e1	1.91e-2	2.37e-4	2.37e-6	2.37e-8	2.37e-10
		Q	-e5	-e2	0.91	1.00	1.00	1.00	1.00
		$Conv$	e-4	e-2	0.60	0.82	0	0	0

Table 3.2: Integral intersection-barycenter invariant vs. kernel radii for various cases

equal signature, for example an isolated peak, the values of the intersection-barycenter descriptor will be high, whereas for the (pure) intersection descriptor they are low (see the tip of the nose in fig. 3.8. On the other side: for vanishing Gaussian curvature, i.e., the intersection with a plane, the value of the intersection-barycenter descriptor is zero, but for the pure intersection descriptor it is $2\pi r$.

Fig. 3.11 finally shows three consecutive views of the Stanford bunny with integral barycenter invariant for a kernel radius $r = 10$. It is a representative example of the situation encountered when employing integral invariant descriptors for the registration of multiple views: the geometry is well described with the kernel radius chosen, but the described surface area is significantly decreased, possibly requiring a bigger amount of overlap which has to be considered in the scanning process.

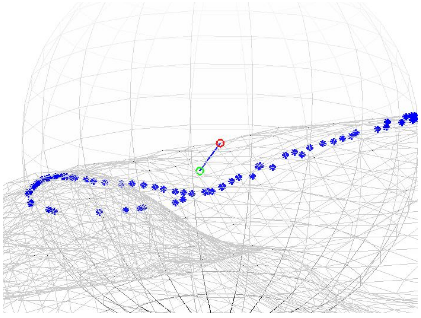

Figure 3.10: Implementation of the integral barycenter descriptor
red circle - point of interest / green circle - barycenter / blue stars - intersection of kernel sphere with triangle mesh

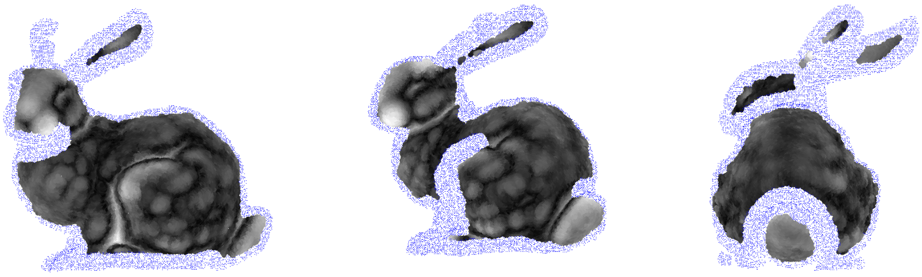

Figure 3.11: Consecutive views of Stanford bunny with integral barycenter invariant and kernel radius r = 10

3.2.4 The Intersection-Eigenvalue Descriptor

Finally one more integral invariant descriptor is presented based on the intersection curve of kernel sphere and surface. It continues where the barycenter invariant stopped in that it employs a principal component analysis on said intersection curve.

Theory of the Intersection-Eigenvalue Desriptor

Theoretical considerations already being covered in detail in section 5.3 of [PHYK05], here only a brief description and interpretation of the results will be given. For the intersection curve c^r we set up the inertia or covariance matrix $J(c^r)$. Its first two eigenvalues $\lambda_{1,2}$ as a two-component descriptor we use as invariants to describe the surface. Considering instead of the surface its local Taylor approximation of second order with main curvatures $\kappa_{1,2}$, we have to compute the moments $\lambda_i = M_i^r$,

$$M_{c1}^r = \int y^2 ds, M_{c2}^r = \int x^2 ds \tag{3.21}$$

ending up with

$$M_{c1}^r = \pi r^3 + \frac{\pi}{64}(\kappa_1^2 - 14\kappa_1\kappa_2 - 11\kappa_2^2)r^5 + O(r^6) \tag{3.22}$$

and

$$M_{c2}^r = \pi r^3 + \frac{\pi}{64}(\kappa_2^2 - 14\kappa_1\kappa_2 - 11\kappa_1^2)r^5 + O(r^6). \tag{3.23}$$

The third eigenvalue is of order r^5 in contrast to r^3 for the two others and therefore is the smallest (for small kernel radii), underlining its connection to the normal vector of the tangent plane (in the limit). While the coefficients of r^5 in equation (3.22) and (3.23) are not directly the curvature κ_1 and κ_2, they are useful replacements [PHYK05].

Implementation for triangulated point cloud data

Implementation continues where the barycenter invariant stopped, from there on being straightforward in that it is not specific to triangulated point cloud data.

Results

In the figs. 3.8 and 3.9 the bottom row illustrates the eigenvalue descriptor for the two kernel radii $r = 3$ and $r = 9$. Since it consists of two components it has been coded into the diagram by assigning each valid vertex a color value where the biggest eigenvalue rules the intensity of the red component of the vertex's color in the RGB color space and the second biggest eigenvalue the intensity of the blue component leaving the green component at zero. Consequently for example a bright distinctively red point in the diagram signifies a vertex with a dominating biggest eigenvalue while a bright purple belongs to a vertex with little difference between biggest and second biggest eigenvalue, both high in relation to the whole range in the diagram concerned.

Chapter 4

Registration using Geometry-Inherent Texture

4.1 Global Registration

As mentioned in the first chapter, ICP-based and similar methods for the registration of shapes, i.e., methods that operate only in the planar respectively the spatial domain, deliver satisfying results for close initial positions. Close initial position in this context means such positions of the shapes that in terms of distances of the actually overlapping regions are close to the desired global minimum. Consequently the overlapping regions are close both in translational distance and orientation.

When acquiring digital models of existing shapes this condition often can be satisfied by using a suitable acquisition setup. A representative example is the scanning an object using a 3D scanning device and a numerically controlled turntable whose axis is fixed with respect to the scanner. After determining the position of said axis in relation to the scanner, sufficiently close initial positions between multiple views of the object are easily obtained, given the accuracy of existing rotary positioning- and scanning technology. Nevertheless in this example top- and bottom views as well as areas with occlusions might make acquisition of extra views and manual postprocessing necessary. Besides, existing hardware well limits the measurable objects in size and weight.

The concept above can in simple terms be described as 'knowing while scanning' with satisfying accuracy the variable position and orientation of a scanning device with respect to a world coordinate system containing the shape to be acquired. For the example with the turntable mentioned above and for the spectator presumably resting in the world coordinate system it was of course the shape on the turntable which moved in its one rotational degree of freedom while the sensor rested.

The principle can also be transformed to systems with higher degrees of freedom. An existing example is the use of an industrial robot arm with all six degrees of freedom, holding a specially tailored 3D scanning device as described in [KNB02], [NKB02], [SNKL04], and [NSKL04]. Similar to the simpler application with the turntable, one of the main challenges here is to efficiently establish a sufficiently precise relation between the coordinate systems of the shape or object wanted and the scanning tool. The position and orientation of the robot's last link (referred to as the 'hand') is known with well defined accuracy. Less so the position and orientation of the sensor's coordinate system in relation to its mount, they often cannot be defined with sufficient accuracy and need to be established experimentally. This is referred to as the sensor-to-hand calibration.

Besides these problems, for many applications the implementation of this 'knowing while scanning' concept will not be possible at all. One example is the use of hand held scanning devices, another one the necessary treatment of special situations such as views with occlusions or top/bottom views mentioned above. Here other means have to be employed to allow or support correct registration without or at least with as little as possible human interaction. Apart from registration there also exist other fields of application which also pose similar challenges and therefore can profit from the solution of this problem. One example is the retrieval of geometric objects from databases.

This situation calls for methods enabling globally stable registration. 'Global' shall be understood here as opposed to starting from close initial positions that allow the successful application of ICP-based methods working only within the planar respectively the spatial domain. Therefore global stability in this context is interpreted as the ability of a method to converge to a global minimum even when starting from general positions, i.e., positions that are widely separated concerning all degrees of freedom. This of course assumes suitable geometry of the data to be registered: the difficulties with kinematic surfaces remain for example.

It suffices to employ such a globally stable method to the point, where a close initial position as mentioned earlier is achieved. In terms of computational efficiency and accuracy it makes sense to switch to an efficient descendant of the classical ICP method from there on.

There exists a great variety of geometric features that are invariant under Euclidean transformation. Some of them have already been used to assist rigid registration. Sharp et al. for example study the employment of curvature, a differential invariant, as well as moment invariants and spherical harmonics invariants for the registration of geometries [SLW02]. Here in this work the focus is to employ integral invariants as described in chapter 3, which promise advantages such as locality of computation and scalability while not being as sensitive to noise as differential invariants.

The principle is to calculate these descriptors in order to gain additional information about the shapes to be registered. This information of course is to be gained in a preprocessing step, before registration. The additional information shall span additional dimensions of the domain in which registration is to take place. In this work the augmented domain is used only to establish nearest neighbor correspondences, which in turn serve as the basis for error minimization just as in step two of ICP. How many additional dimensions are used, which type of integral invariant descriptors, and with which kernel radii is a matter of the application, i.e. the characteristics of the shapes to be registered.

Furthermore it is advantageous to amplify the values of the integral invariant descriptors with respect to the planar respectively spatial geometrical data in order to increase their weight. This amplification factor is called the 'gain' in the following and again the right choice depends on the application and the type of invariant. Nevertheless the following sections show in principle the advantage of the employment of integral invariants.

4.2 Global Registration in 2D using Geometry Inherent Texture

4.2.1 Implementation in 2D

The purely planar, not integral invariant – assisted method used for comparison is based simply on minimization of foot point distances in the plane to get the parameters for the planar transformation (that is in detail one two component vector of translation and one angle of rotation). The integral invariant–assisted methods do the same, the difference lies in how the nearest neighbor correspondences are established. The conventional method used for comparison looks for nearest points in the planar domain and then minimizes the distances between them, whereas using the integral invariants the nearest points in the $2+1$ or more generally $2+n$ dimensional, enhanced domain are determined. Nevertheless, after these point pairs are found, only the distances in the plane are minimized. Hereby a symmetric approach in establishing nearest neighbor correspondences is used, i.e., between two point clouds correspondences in both directions are acquired.

There is one consequence of the relative difference of ranges of the incommensurable coordinates x and y (respectively x, y and z for the spatial case) versus the invariant and the amplification of the invariant domain that has to be kept in mind for the implementation: maximum distance criteria for the search for nearest neighbors have to be handled differently, due to their possibly bigger range in the amplified additional coordinate(s). This also must not be forgotten in the dynamic handling of such criteria over the course of registration, when in later stages of the registration process corresponding regions are closer to each other and the search radius for correspondences is decreased.

Data used

Four different point data sets are used shown in fig. 4.1. All of them are of synthetic origin, created using spline methods. The curves were created to represent different characteristics, open and closed as well es smooth and edgy. In the following figures the curve itself is always drawn in black, one set of sampled, noise added data points (set 'A') is drawn as red dots the other (set 'B') as blue dots. The total number of sampled data points in all four curves is 1000. In order to avoid unpredictable additional influences on the results of the registration process a rather even distribution of points has been chosen. Based on the underlying curve, two different noisy sampled data sets are generated, therefore in the noisy case, two nonidentical noisy point sets have to be registered. Of those only the one remaining in its original position is drawn in red in fig. 4.1. The noise added to the two data sets 'A' and 'B' is normally distributed with $\sigma = 1$ and an amplitude of +/- 0.005. Spline control points are marked by small crosses.

Initial positions

In order to be sufficiently 'general', also meaning sufficiently representing a worst case scenario, for an examination of globally stable planar registration the initial position shall be defined as follows: first the closest points of the two shapes to be registered shall be separated from each other by about the

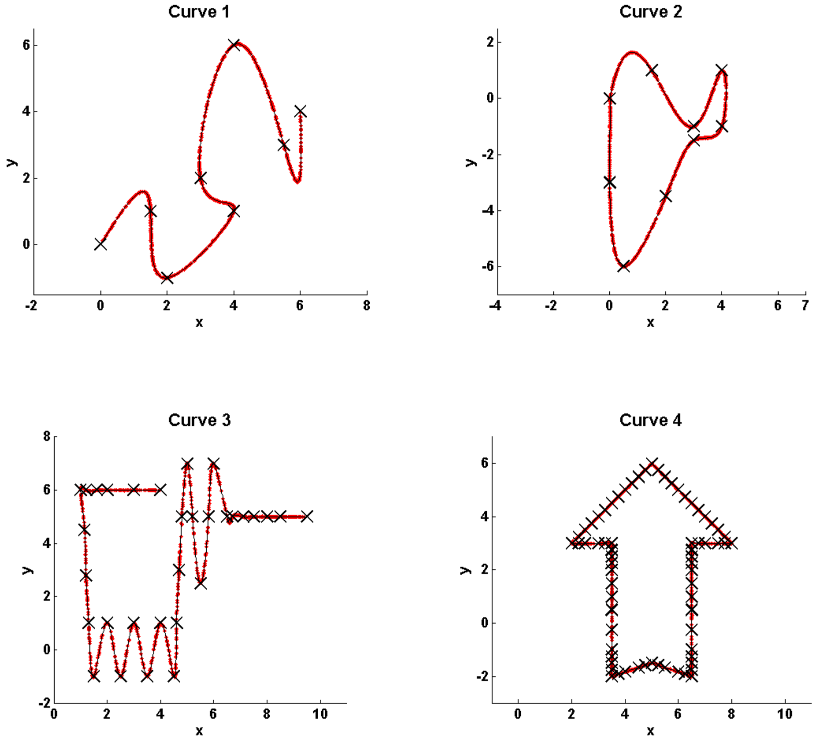

Figure 4.1: Data sets used for planar registration using integral invariants (The curve is drawn in black while red dots mark one of the two sampled noise added data sets and crosses mark the spline control points used for construction.)

maximum distance between two points inside one of the two point clouds. Second a significant rotation shall be applied in two steps, up to roughly, but not exactly 180 degrees. The purpose of this is to set up traps in the form of local minima that have to be overcome by the registration method. Especially symmetrical shapes as presented in curve 4 in fig. 4.1 combined with said rotation will likely pose an obstacle for conventional algorithms as ICP working only in the planar domain.

Fig. 4.2 shows for each of the four data sets two different initial positions with translation and with rotations of 89 and 166 degrees applied. Those values have been chosen to get roughly but not exactly a rotation by 90 and 180 degrees respectively.

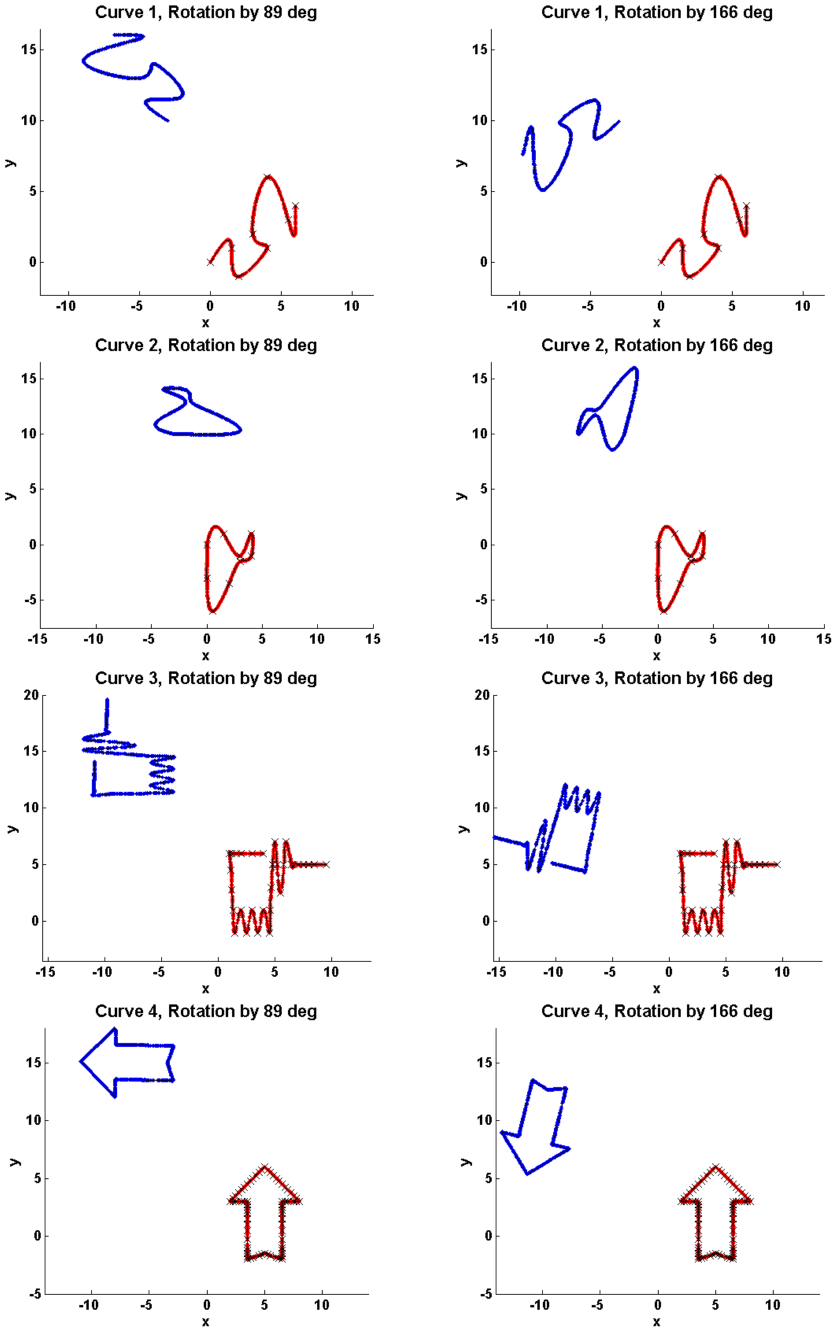

Figure 4.2: Initial positions for planar data sets
(The fixed point set is drawn in red whereas the moving one is drawn in blue. The left column contains the initial position with a rotation of about $\pi/2$ and the right column those with rotation of about π.)

4.2.2 Results in 2D

Single integral invariant descriptor used

The obvious difference between the establishment of nearest neighbor correspondences only in the planar domain and in the domain enhanced with one integral invariant descriptor respectively is shown for non–noisy data in fig. 4.3. For this example the integral distance invariant has been used. Although for the experiments conducted a symmetric approach in establishing nearest neighbor has been used, only one direction is illustrated in order to avoid confusion.

For six randomly chosen points on the red curve (marked as black crosses) the corresponding nearest neighbors on the blue curve are displayed. Nearest neighbors in the planar domain are shown as green circles connected by green dashed lines whereas those in the integral invariant enhanced domain are drawn as black circles connected by black dotted lines. In this figure it can clearly be seen that for the chosen kernel size and amplification of the integral invariant values and for this shape the use of integral invariants yields better correspondences. Looking at the distribution of integral invariant values for this curve (fig. 3.2) shows why.

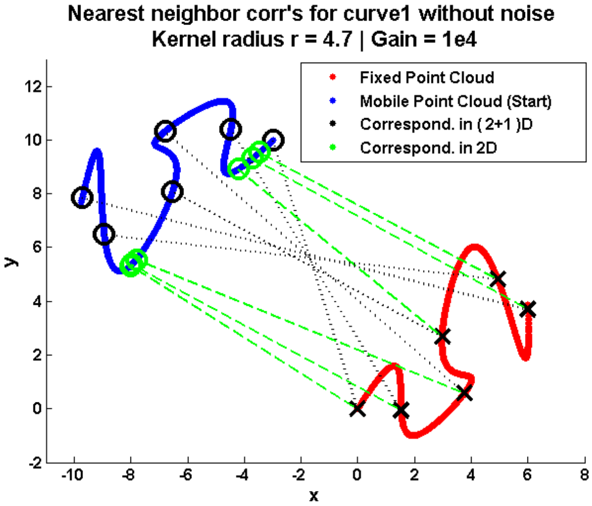

Figure 4.3: Comparison of nearest neighbors in 2D and in $(2+1)$D
(Only one direction shown. Black crosses mark random points on the fixed data set. Black respectively green circles mark the corresponding nearest neighbors in the (2+1)D respectively the 2D domain.)

Using data with added noise decreases the advantage of the integral invariant enhanced method to some degree as can be observed in fig. 4.4. Nevertheless also with noisy data there is a distinctive difference between the purely planar determination of nearest neighbors and additionally using an integral invariant. The purely planar method exhibits globally seen poor choice of nearest neighbors.

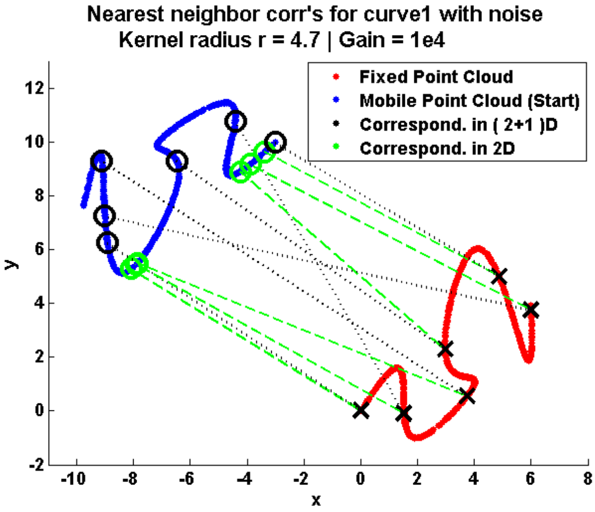

Figure 4.4: Comparison of nearest neighbors in 2D and in (2 + 1)D for noisy data
(Only one direction shown. Black crosses mark random points on the fixed data set. Black respectively green circles mark the corresponding nearest neighbors in the (2+1)D respectively the 2D domain.)

Investigating the results for a range of gain factors from $G = 1.0e0$, $G = 1.0e1$ to $G = 1.0e6$ and for kernel radii $r = 1.0$, $r = 2.2$, $r = 4.7$ and $r = 10.0$ have lead to the following conclusion: the smaller the amplification factor and the kernel radius (in relation to curvature and overall size of the data set), the closer nearest neighbors found in the $(2 + 1)$D domain are to those in the 2D domain. Amplification of the invariant domain with a gain factor G bigger than $1.0e6$ did not further improve results.

When looking at the registration process for the curve 4 with noise added and a rotation of 166 degrees we see the following (fig. 4.5): due to reasons of symmetry in this shape standard registration does not achieve the necessary rotation. It locks in a local minimum failing to find the global one. All intermediate states of the 20 iterations are drawn in green for the classical method and in black for the integral invariant assisted method. Fig. 4.6 shows the registration process in terms of mean squared ground truth distances versus iteration count.

The four different curves, each with two different starting positions (rotation by 89 degrees and 166 degrees) and each with and without noise added give 16 different cases. Those cases where used to compare performance of to the classical method versus registration assisted by

$$\text{integral distance invariant for kernel radii}$$
$$r = \{0.10, 0.22, 0.47, 1.0, 2.2, 4.7, 10.0\}$$

$$\text{and the integral area invariant for kernel radii}$$
$$r = \{0.47, 1.0, 2.2, 4.7\}.$$

The results are shown in table 4.1.

Without using integral invariants only one case, the smooth open curve 1 rotated by 89 degrees could be successfully registered (both without and with noise), resulting in a 12.5% success rate. Using the integral distance invariant on the other hand results in a success rate of 100%, meaning that in all cases at least one of the kernel values listed was successful. Furthermore in 62.5% of the cases all of the kernel values were successful. For the integral area invariant the success rate of all cases was again 100% respectively 87.5% (remember that for the second figure a smaller range of kernel values was used).

Multiple integral invariant descriptors used

It is of further interest, whether the convergence gets even better when multiple integral invariant descriptors are used simultaneously. This can mean different types of descriptors, different kernel radii as well as a combination of both. Fig. 4.8 shows the result in terms of mean squared ground truth distances vs. iteration count for establishing nearest neighbor correspondences in $2D$, in $(2+1)D$ and in $(2+4)D$. For $(2+4)D$ the distance invariants with $r = 4.7$ and with $r = 10.0$ and area invariants with $r = 1.0$ and $r = 2.2$ are used, but for $(2+1)D$ only the distance invariant with $r = 4.7$ (this is the fastest converging integral invariant of all if only one integral invariant is used). The diagram shows that in combination the convergence indeed is faster than only with the single distance invariant. In fig. 4.7 the intermediate positions of the mobile point cloud for the first six iterations can be seen for the use of only one integral invariant (black) and of four integral invariants (magenta) showing the latter slightly ahead.

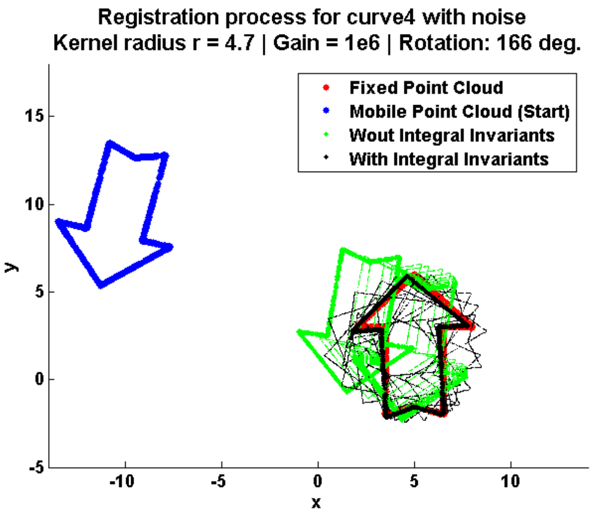

Figure 4.5: The registration process for the planar curve 4 with noise added (Intermediate positions of the point sets after each of the 20 iterations are drawn in black respectively green for the integral invariant assisted respectively the non assisted method.)

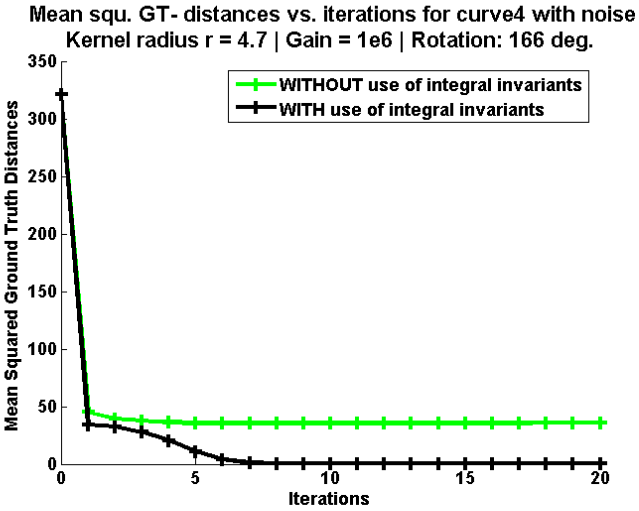

Figure 4.6: Ground truth error metric versus iterations for the registration of curve 4 (see fig. 4.5) with noise added

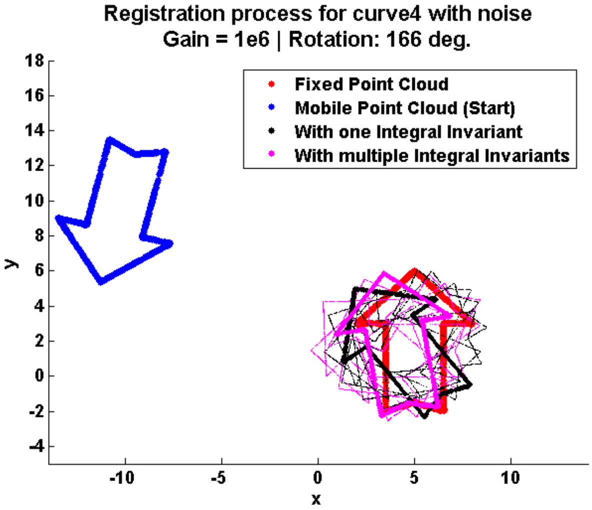

Figure 4.7: The registration process for the planar curve 4 with noise added using multiple integral invariant descriptors
(Intermediate positions of the point sets after each of the first 6 iterations are drawn in black respectively magenta for the method using one integral invariant respectively four integral invariants.)

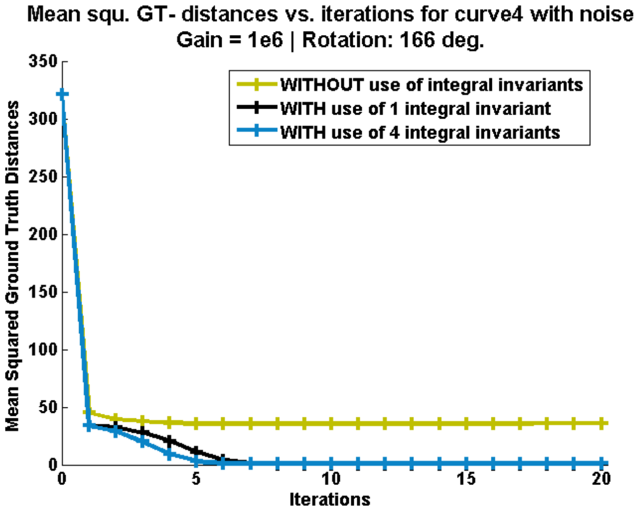

Figure 4.8: Ground truth error metric versus iterations for the registration of curve 4 with noise added using multiple integral invariant descriptors
(The integral invariant descriptors used are the distance invariants with $r = 4.7$ and with $r = 10.0$ and area invariants with $r = 1.0$ and $r = 2.2$. The black curve for comparison only employs the distance invariant with $r = 4.7$.)

	w/o int. invar.	**int. dist. invar.**	**int. area invar.**
DATA SETS WITHOUT NOISE:			
Curve 1 / 89°	√	7/7	4/4
- / 166°	F	7/7	4/4
Curve 2 / 89°	F	7/7	4/4
- / 166°	F	7/7	4/4
Curve 3 / 89°	F	7/7	4/4
- / 166°	F	7/7	4/4
Curve 4 / 89°	F	7/7	4/4
- / 166°	F	7/7	4/4
DATA SETS WITH NOISE:			
Curve 1 / 89°	√	7/7	4/4
- / 166°	F	▷4/7◁	4/4
Curve 2 / 89°	F	▷4/7◁	4/4
- / 166°	F	▷4/7◁	4/4
Curve 3 / 89°	F	7/7	▷3/4◁
- / 166°	F	▷4/7◁	▷3/4◁
Curve 4 / 89°	F	▷5/7◁	4/4
- / 166°	F	▷2/7◁	4/4

Table 4.1: Success of planar registration without integral invariants, with distance and with area invariant.
(√ marks where registration without integral invariants worked and F where it failed. 4/7 signifies that of in total seven kernel values examined, four enabled successful registration. Triangles mark the cases where not all of the used kernel radii lead to success.)

4.3 Global Registration in 3D using Geometry Inherent Texture

4.3.1 Implementation in 3D

The spatial case is a straightforward extension of the planar one, i.e., when registering while employing one integral invariant descriptor, we operate in a $(3+1)D$ domain. Nevertheless due to the increased degrees of freedom for Euclidean transformations here, a global minimum generally is harder to achieve than in the planar case.

Differing from the planar analysis, we do not restrict ourselves to identical or fully overlapping data sets now. When acquiring digital models of existing objects, considerations of efficiency call for a minimum of scans to be acquired. Nevertheless overlapping regions are necessary for the registration process, the actual amount depending on the application, i.e., the shape as well as on the quality of initial positions.

While integral invariant descriptors do support registration, this does not come completely for free. There is a clear trade-off when choosing kernel radii. Big kernel radii often result in more 'descriptive' integral invariant descriptor values with bigger variation over a shape. This is clearly visible for the biggest kernel radius $r = 20$ where the edges of the part show more distinctively compared to the rather uniform distribution for the smallest $r = 5$ in fig. 4.9. Small (in relation to the geometry of the shape) kernel radii capture less variation of the shape's features and of course in the limit case tend toward $2r\pi$.

On the other hand, due to their generation by intersecting with a kernel ball, with increasing kernel radius increasing regions bordering edges and holes of the data set remain without valid descriptor values (again see the increase of area without descriptor values from top to bottom, i.e.; for increasing kernel radii in fig. 4.9). This loss of usable surface of course takes place 'on both sides', i.e., on both shapes to be registered onto each other amplifying the difficulty further. When registering identical, fully overlapping data sets this problem is of less importance since the integral invariant described area still is identical on both data sets.

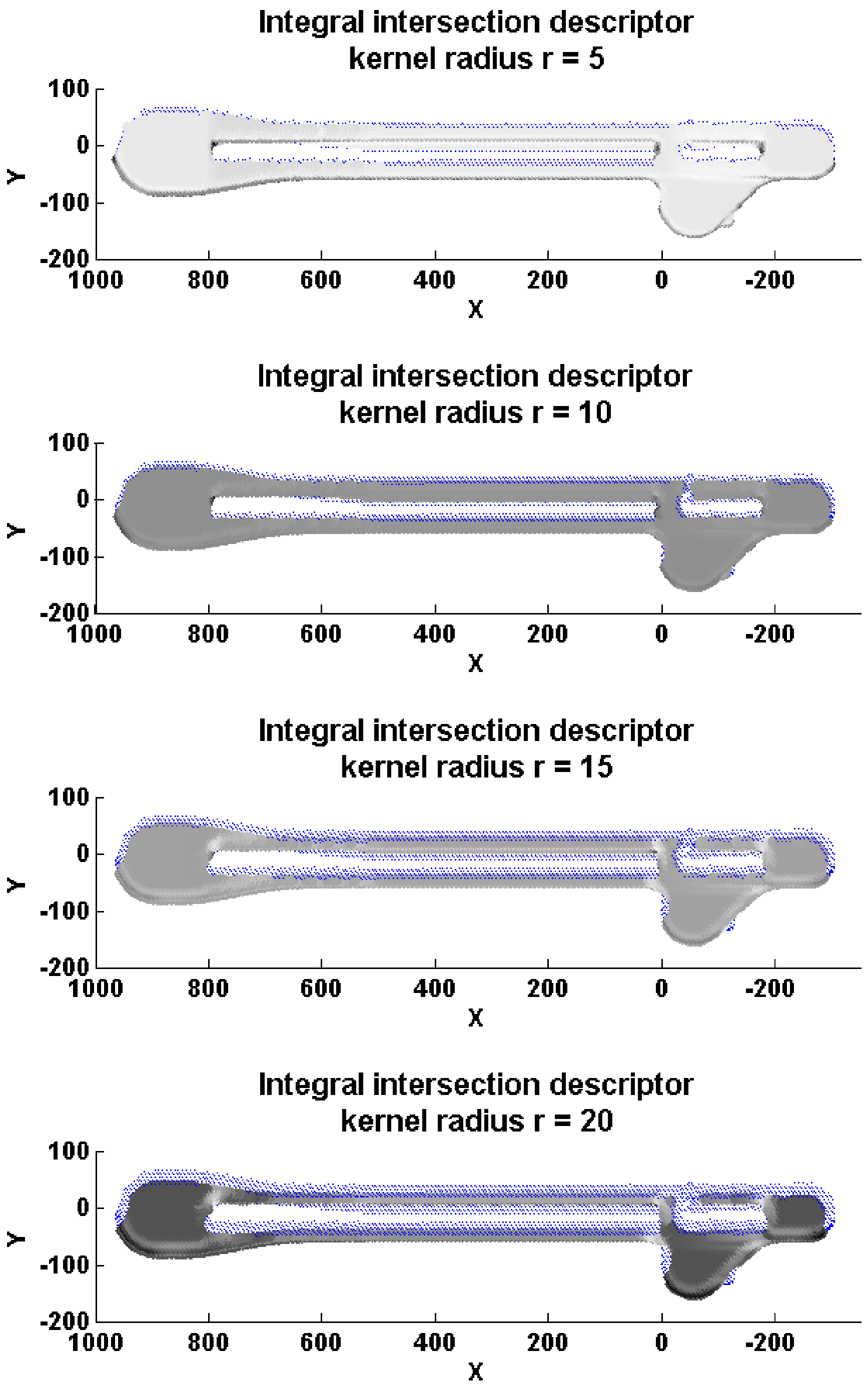

Figure 4.9: Areas without descriptor values as function of kernel radius (Points w/o descriptor values drawn blue)

4.3.2 Results in 3D

Identical shapes: data set 'Face'

The first data set used for this experiment is the model of a human face already presented in chapter 3 (fig. 4.10 left). It is rotated by a yaw angle of y = 10 °, a pitch of p = 20 ° and a roll of r = 150 ° and finally translated by the vector (0.05, 0.05, 0.25) for the initial position to be registered onto itself (fig. 4.10 right).

All three integral invariant descriptors where calculated for kernel radii

$$r = \{0.010, 0.015, 0.020, 0.025, 0.030\}$$

with the first respectively the last value being the limits between insignificant and uniform descriptor values and too little an area with values, and registration has been carried out for amplification factors

$$G = 1.0e0, 1.0e1, ..., 1.0e7.$$

Not surprisingly without the help of integral invariants the registration got locked in a local minimum. Five kernel radii multiplied by eight amplification factors result in 40 experiments for each type of integral invariant. The overall success rates, i.e. the percentage of experiments where the global minimum was reached within 20 iterations are:

Intersection descriptor: $26/40 = 65\%$
Barycenter descriptor: $28/40 = 70\%$
Eigenvalue descriptor: $28/40 = 70\%$

Fig. 4.11 shows a typical example. In most cases the integral barycenter invariant converged faster than the integral intersection invariant. This was expected in general since the barycenter descriptor is more stable than the intersection descriptor.

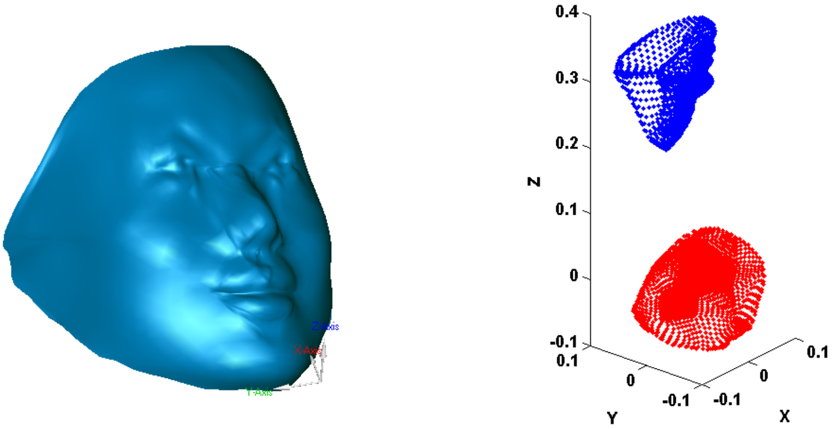

Figure 4.10: Data set Face (left) and initial position (right)

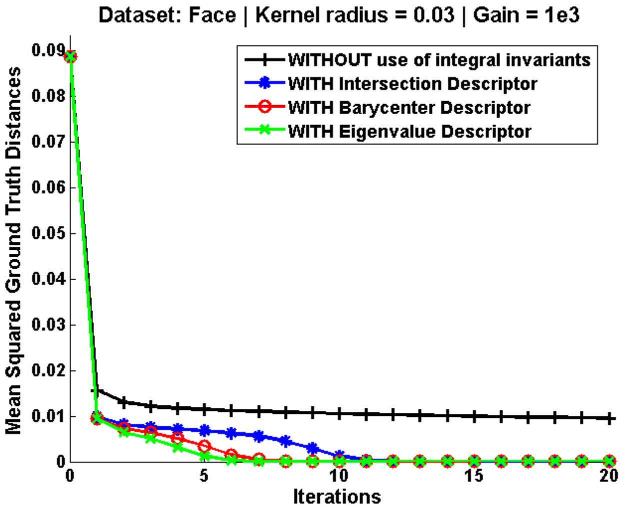

Figure 4.11: Ground truth error metric versus number of iterations for the registration of the Face data set

Identical shapes: data set 'Drop Forge A'

The second data set is the shape for a drop forged mechanical part (fig. 4.12 left) already presented in fig. 4.9. The applied transformation consists of rotation by yaw / pitch / roll of 80 ° / 0 ° / 60 ° and translation vector (200, 700, 0) resulting in the initial position shown in fig. 4.12 right.

Only descriptor values for a

$$\text{kernel radius of } r = 10$$

were used. The result repeats those of the Face data set. Without the help of integral invariants help a local minimum is reached where the wrong ends of the lengthy part are matched together resulting in a stable local minimum and high error (fig. 4.13, black curve). On the other hand the integral invariants: the amplification factors used here were

$$G = 1.0e0, 1.0e1, ..., 1.0e9.$$

With the integral eigenvalue invariant the global minimum was reached with factors $G = 1.0e2$ and higher, while with their in relation smaller values integral intersection and integral barycenter invariant clung to the global minimum with factors $G = 1.0e6$ and higher.

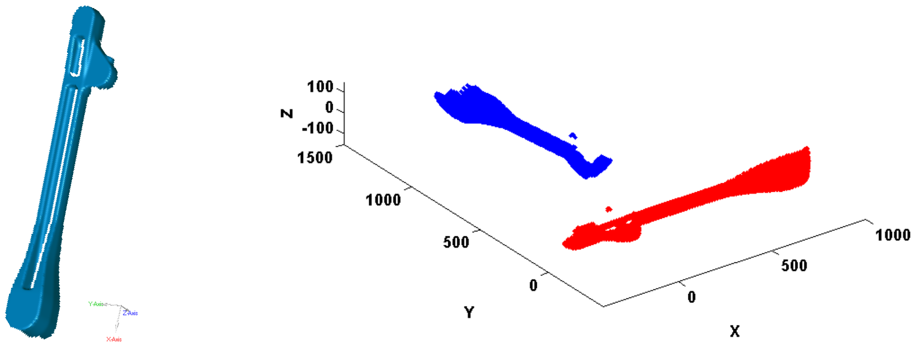

Figure 4.12: Data Drop Forge A (left) and initial position (right)

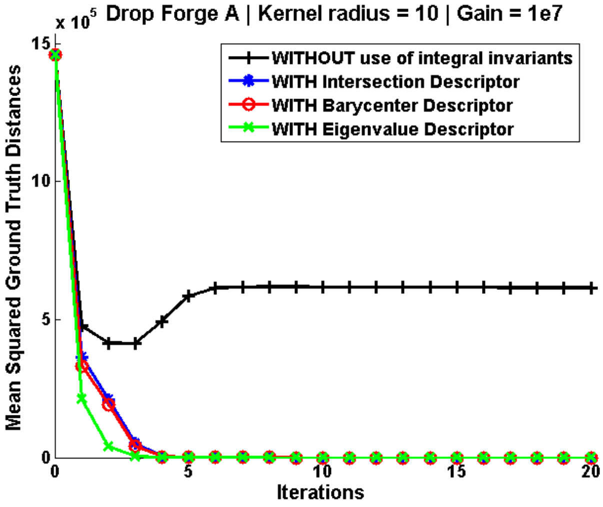

Figure 4.13: Ground truth error metric versus number of iterations for the registration of the Drop Forge A data set

Overlapping shapes: data sets 'Drop Forge A' and 'Drop Forge B'

This certainly is the application with the biggest practical significance. The lengthy part itself is proving rather difficult with its symmetry and its rather uniform appearance of equal features all over the shape, such as the constant radius of the edges. A pre-alignment could be done by principal component analysis, but this would likely produce equal chances for matching false ends together as for matching together the right.

We take two different but overlapping views of the drop forging part, see fig. 4.14. The transformation applied to data set 'B' to get the initial position is the same as for the analysis above with the identical data set 'Drop Forge A', that is a rotation by yaw / pitch / roll of 80 ° / 0 ° / 60 ° and translation vector (200, 700, 0). Also

$$\text{kernel radius } r = 10$$

and amplification factors

$$G = 1.0e0, 1.0e1, ..., 1.0e9$$

are chosen the same. Again, with overlapping data sets the performance of the various methods is similar to the previous cases. Without integral invariants registration takes the wrong turn and ends up matching wrong ends of the shape together. Using the integral eigenvalue invariant registration is able to match right ends from amplification factors $G \geq 1.0e3$, with the intersection invariant from $G \geq 1.0e7$, and the barycenter descriptor works with $G \geq 1.0e8$. The ground truth error metric of the latter situation is shown in fig. 4.15.

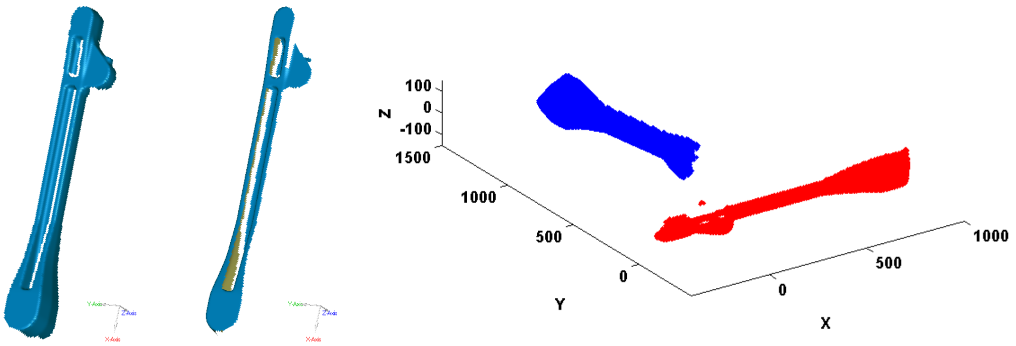

Figure 4.14: Data set Drop Forge A (left) and B (middle) and initial position (right)

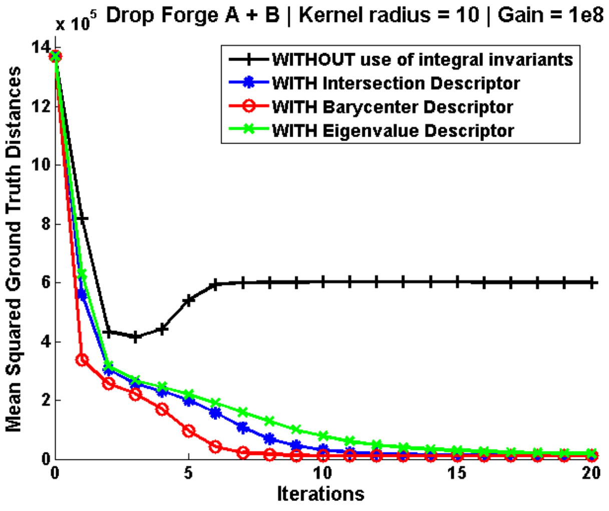

Figure 4.15: Ground truth error metric versus number of iterations for the registration of the Drop Forge A and B data sets

Bibliography

[AMN$^+$98] S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Y. Wu. An optimal algorithm for approximate nearest neighbor searching fixed dimensions. *Journal of the ACM*, 45(6):891–923, 1998.

[BM92] P. J. Besl and N. D. McKay. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–258, 1992.

[CFM02] U. Castellani, A. Fusiello, and V. Murino. Registration of multiple acoustic range views for underwater scene reconstruction. *Computer Vision and Image Understanding: CVIU*, 87(1–3):78–89, 2002.

[CM92] Y. Chen and G. Medioni. Object modelling by registration of multiple range images. *International Journal of Image and Vision Computing*, 10(3):145–155, 1992.

[Con86] M. L. Connolly. Measurement of protein surface shape by solid angles. *J. Mol. Graph.*, 4(1):3–6, 1986.

[EFF98] D. W. Eggert, A. W. Fitzgibbon, and R. B. Fisher. Simultaneous registration of multiple range views for use in reverse engineering of cad models. *Comput. Vis. Image Underst.*, 69(3):253–272, 1998.

[FH86] O. D. Faugeras and M. Hebert. The representation, recognition, and locating of 3-d objects. *Int. J. Rob. Res.*, 5(3):27–52, 1986.

[Hor87] B. K. P. Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America*, 4(4), 1987.

[HP] Q.-X. Huang and H. Pottmann. Automatic and robust multiview registration. Technical report, Geometry Preprint Series, Vienna Univ. of Technology.

[JH02] T. Jost and H. Hügli. Fast ICP algorithms for shape registration. In *Proceedings of the 24th DAGM Symposium on Pattern Recognition*, pages 91–99, London, UK, 2002. Springer-Verlag.

[JK97] A. E. Johnson and S. B. Kang. Registration and integration of textured 3-d data. In *NRC '97: Proceedings of the International Conference on Recent Advances in 3-D Digital Imaging and Modeling*, page 234, Washington, DC, USA, 1997. IEEE Computer Society.

[KNB02] S.H. Kölpl, A. Niel, and M. Burgstaller. Robot-based 3-dimensional measurement of complex parts using coded light. In *Proc. of the 3th International symposium on Robotics and Automation, ISRA 2002 Toluca, Mexico*, September 2002.

[KvD02] J. J. Koenderink and A. J. van Doorn. Image processing done right. In *ECCV '02: Proceedings of the 7th European Conference on Computer Vision-Part I*, pages 158–172, London, UK, 2002. Springer-Verlag.

[LPZ03] S. Leopoldseder, H. Pottmann, and H. Zhao. The d^2-tree: A hierarchical representation of the squared distance function. Technical Report 101, Geometry Preprint Series, Vienna Univ. of Technology, March 2003.

[MCYS05] S. Manay, D. Cremers, A. J. Yezzi, and S. Soatto. One-shot integral invariant shape priors for variational segmentation. In *EMMCVPR*, pages 414–426, 2005.

[MGPG04] N. J. Mitra, N. Gelfand, H. Pottmann, and L. Guibas. Registration of point cloud data from a geometric optimization perspective. In R. Scopigno and D. Zorin, editors, *Eurographics Symposium on Geometry Processing*, pages 23–32, 2004.

[MHYS04] S. Manay, B. W. Hong, A. J. Yezzi, and S. Soatto. Integral invariant signatures. *Computer Vision - ECCV 2004*, pages 87–99, 2004.

[NKB02] A. Niel, S. H. Kölpl, and M. Burgstaller. Robotic three-dimensional measurement system for complex metal parts using structured light. In *Proc. ISPRS Symposium on Photogrammetric Computer Vision*, page 190 ff., September 2002.

[NSKL04] A. Niel, P. Sommer, S. H. Kölpl, and Y. Lypetskyy. Measuring with robots: A flexible system for automated reconstruction of free formed parts. In *Proc. of 3 D Modelling 2004; Paris, France*, 2004.

[PH03] H. Pottmann and M. Hofer. Geometry of the squared distance function to curves and surfaces. In H.-C. Hege and K. Polthier, editors, *Visualization and Mathematics III*, pages 223–244. Springer, 2003.

[PHYH04] H. Pottmann, Q.-X. Huang, Y.-L. Yang, and S.-M. Hu. Geometry and convergence analysis of algorithms for registration of 3D shapes. Technical Report 117, Geometry Preprint Series, Vienna Univ. of Technology, June 2004.

[PHYK05] H. Pottmann, Q.-X. Huang, Y.-L. Yang, and S. Kölpl. Integral invariants for robust geometry processing. Technical Report 146, Geometry Preprint Series, Vienna Univ. of Technology, November 2005.

[PLH02] H. Pottmann, S. Leopoldseder, and M. Hofer. Simultaneous registration of multiple views of a 3D object. *ISPRSA*, 34(3A):265–270, 2002.

[PLH04] H. Pottmann, S. Leopoldseder, and M. Hofer. Registration without ICP. *Computer Vision and Image Understanding*, 95(1):54–71, 2004.

[PLWP02] H. Pottmann, S. Leopoldseder, J. Wallner, and M. Peternell. Recognition and reconstruction of special surfaces from point clouds. *ISPRSA*, 34(3A):271–276, 2002.

[Pot04] H. Pottmann. Industrial geometry. Lecture Notes, Vienna Univ. of Technology, 2004.

[PR98] H. Pottmann and T. Randrup. Rotational and helical surface approximation for reverse engineering. *Computing*, 60:307–322, 1998.

[Pul99] K. Pulli. Multiview registration for large data sets. In *3DIM*, pages 160–168, 1999.

[PW01] H. Pottmann and J. Wallner. *Computational Line Geometry.* Mathematics + Visualization. Springer, Heidelberg, 2001.

[PWL01] H. Pottmann, J. Wallner, and S. Leopoldseder. Kinematical methods for the classification, reconstruction and inspection of surfaces. In *SMAI 2001: Congrès national de mathématiques appliquées et industrielles*, Publications de l'Équipe de Mathématiques Appliquées (Numéro special), pages 51–60. Université de Technologie de Compiègne, 2001.

[RL01] S. Rusinkiewicz and M. Levoy. Efficient variants of the ICP algorithm. In *Proceedings of the Third Intl. Conf. on 3D Digital Imaging and Modeling*, pages 145–152, 2001.

[Sim96] D. Simon. *Fast and Accurate Shape-based Registration.* PhD thesis, Carnegie Mellon University, Pittsburgh, PA., 1996.

[SLW02] G. C. Sharp, S. W. Lee, and D. K. Wehe. ICP registration using invariant features. *IEEE Trans. Pattern Anal. Mach. Intell*, 24(1):90–102, 2002.

[SNKL04] P. Sommer, A. Niel, S. H. Kölpl, and Y. Lypetskyy. High precision measurement of free formed parts using industrial robots. In *Proc. of ROSE 2004, Graz, Austria*, pages 79–84, 2004.

[TG99] T. Tuytelaars and L. J. Van Gool. Content-based image retrieval based on local affinely invariant regions. In *VISUAL '99: Proceedings of the Third International Conference on Visual Information and Information Systems*, pages 493–500, London, UK, 1999. Springer-Verlag.

[TRC03] A. Telea, M. Rumpf, and U. Clarenz. Robust feature detection and local classification for surfaces based on moment analysis. *IEEE Trans. Vis. Comput. Graph.*, 10(5):516–524, 2003.

[YAHF98] S. M. Yamany, M. T. Ahmed, E. E. Hemayed, and A. A. Farag. Novel surface registration using the grid closest point (GCP) transform. In *ICIP (3)*, pages 809–813, 1998.

Lebenslauf (German)

Am 17. Juni 1968 wurde ich als Sohn des Diplom-Ingenieurs Rudolf Kölpl und seiner Frau Elsbeth Kölpl, geb. Hodel, in Brugg / Kanton Aargau / CH geboren. Von 1975 bis 1979 besuchte ich die Volksschule in Kapfenberg / Stmk. und in Graz, von 1979 bis 1983 das Bundesgymnasium Dreihackengasse in Graz und von 1983 bis 1988 die Abteilung für Nachrichtentechnik und Elektronik der HTBLuVA Graz-Gösting, wo ich am 13. Juni 1989 mit ausgezeichnetem Erfolg maturierte.

Den Grundwehrdienst leistete ich von Oktober 1988 bis März 1989 beim LWSR 54 in Graz ab. Anschließend begann ich meine Berufslaufbahn beim Institut für elektron. Systementwicklung der Joanneum Research Forschungs-GmbH in Graz und arbeitete ab 1991 bei der Firma TBK Automation ebenfalls in Graz.

Im Herbst 1994 kündigte ich meine Anstellung und inskribierte an der TU Graz das Diplomstudium Wirtschaftsingenieurwesen - Maschinenbau, Studienzweig Mechatronik im Maschinenbau. Während dieses Studiums absolvierte ich im Studienjahr 96/97 zwei Semester im Rahmen des Programms ERASMUS an der Loughborough University of Technology / Leistershire / UK.

Meine Diplomarbeit verfasste ich am Institut Français de la Mécanique Avancée (IFMA) in Clermont-Ferrand / F. Das Diplomstudium schloss ich am 16. Jänner 2001 ab und am 26. Jänner 2001 wurde mir der akademische Grad eines Diplom-Ingenieurs verliehen.

Anschließend arbeitete ich bis August 2003 am Institut für Digitale Bildverarbeitung von Joanneum Research in Graz.

Graz, Dezember 2005 Stephan H. Kölpl