APPLICATIONS OF COMPUTATIONAL GEOMETRY TO COMPUTER AIDED DESIGN AND MANUFACTURING

Applications of Computational Geometry to Computer Aided Design and Manufacturing

Student: Carlos A. Vanegas
Advisor: Prof. Oscar E. Ruiz

DEPARTAMENTO DE CIENCIAS BÁSICAS
UNIVERSIDAD EAFIT
MEDELLÍN, COLOMBIA

Submitted in partial fulfillment of the requirements for the degree of Bachelor of Science in Mathematical Engineering in the Department of Basic Sciences at EAFIT University

April, 2007

Email: cvanega3@eafit.edu.co

Acknowledgements

To my parents and my sister, for all their love and encouragement.

My sincere gratitude to my advisor, Prof. Oscar Ruiz, for his reliable support and invaluable supervision during these years of work. Special thanks to Prof. Xoan Leiceaga at Universidade de Vigo, Spain, for sharing with me his interesting perspective of life and inviting me twice as visiting researcher of his R+D+I group. Many thanks to Prof. Gloria Sierra, for her opportune guidance and interest, and to Prof. Juan D. Restrepo for his decisive early support and friendly humor. Thanks to Prof. Carlos A. Cadavid for his help with the technical revision of parts of this work.

Many thanks to the faculty at EAFIT University who supported and participated in the project of creating the new undergraduate program in Mathematical Engineering (Applied Mathematics): Profs. Gerardo Arango, Marta C. Gómez, Maria E. Puerta, Daniel I. Velásquez, Luciano Ángel, Luis A. Quintero, Carlos M. Vélez, Ermilson Velásquez, Gabriel I. Loaiza, Raúl A. Gomez, and Manuel Sierra, among others, and to Mrs. Nora Ramírez.

Thanks to the faculty at the Depts. of Mechanical Engineering (Profs. Jorge L. Ochoa and Manuel J. García) and Computer Science (Profs. Juan G. Pulido, Helmuth Trefftz, Juan F. Cardona, Francisco J. Correa, and Alberto Restrepo) who have given me valuable academic and administrative support.

My grateful acknowledgements to EAFIT University and the Andres Bello Agreement for financially supporting my undergraduate studies and the research projects in which I participated throughout these years. Some of the people who made possible such a support are: Dr. Juan L. Mejía (President), Dr. Julio Acosta (Vice-President for Academic Affairs), Dr. Felix Londoño, Mrs. Amalia Urrea, Dean Alberto Rodriguez, Mr. Jorge G. Tabares, Mrs. Maria E. Hoyos, Mrs. Ángela Echeverri, Mrs. Ángela Sánchez, all of them from EAFIT University.

Deep thanks to my friends and colleagues at the CAD/CAM/CAE Laboratory, EAFIT University (Eliana Vásquez, Sebastián Peña, Miguel Granados, Juan F. Duque, Leidy Suárez) and at the Group of Engineering and Design, University of Vigo (Manuel Rodríguez, José Prieto, Xavier y Alexandre Alonso, Jorge y Jacobo Cabaleiro, Eva Soto, Pablo Gamallo, Marta Abruñedo, Norma Martin).

Many thanks to my friends and classmates in Mathematical Engineering at EAFIT: Andrés M. Villegas, Juan D. Jaramillo and John J. Ramírez, among others.

Preface

1. Research Engagement of Student Carlos Vanegas

It is my pleasure to provide a context to international readers of this document, on what a graduation project is in EAFIT University, and on the Particular trajectory of Carlos Vanegas' work.

Carlos Vanegas is a student of the undergraduate program in Mathematical Engineering (Applied Mathematics) at EAFIT University, Medellín, Colombia, and has been a research assistant under my supervision in the CAD/CAM/CAE Laboratory at EAFIT since January 2004. During his years of work in my Laboratory, Carlos has participated in three research projects, has been coauthor of four research papers that have been or will be published in international conferences or journals, and has been twice awarded research internships at universities abroad.

Carlos began his training in the CAD/CAD/CAM Laboratory during the first semester of 2004, while he was a laboratory teaching assistant for my course on CAD/CAM Systems. During the second semesters of 2004 and 2005, Carlos improved and implemented for large data sets our algorithm for curve and surface reconstruction from planar point sets with stochastic noise. Such work was part of projects 1) Computation of 1- and 2-Manifolds PL-1 for CAGD, and 2) Surface reconstruction for CAD/CAM/CG.

Because of his distinguished performance in my laboratory, Carlos was awarded a research internship at the University of Vigo, Vigo, Spain, from January to July, 2005. He worked for the Design and Engineering Group (GED—Group de Enxeñeria e Deseño) under direct supervision of Prof. Dr. Eng. Xoan A. Leiceaga. His contribution was defined to be the research and development of algorithms and software for application of computational geometry to computer aided design, analysis and manufacturing of ship hulls.

As a result of his especially good performance during his internship, Carlos Vanegas was once again awarded an invitation to work at GED, from February to July 2006, under supervision of Prof. Leiceaga. His contribution was defined to be the design and implementation of the dynamic model and visualization system to be used in a mobile crane simulator. During this internship, as well as during his entire participation in my Laboratory, Carlos Vanegas showed excellent working ethics, very good technical skills, capability to work unsupervised and a smooth adaptation to the societal environment. His supervisors in Vigo have let me know in both occasions of their satisfaction with Carlos' performance and personal characteristics.

During the second semester of 2006, Carlos developed under my supervision a project on detection of wear regions in cylindrical surfaces for the polymer processing industry, in which he designed and implemented a stochastic geometry application (CylWear). This software is currently under copyright registration process by EAFIT University, and obtained outstanding evaluation by the Institute for Training and Research on Plastic and Rubber, (Instituto de Capacitación e Investigación del Plástico y el Caucho—ICIPC, Medellín, COLOMBIA).

2. Publications related to this Graduation Project

The results of the works in which Carlos has participated, have been included in the following papers:

- Ruiz O, Vanegas C. Piecewise linear curve reconstruction from point clouds. In I. Horvath, J. Duhovnik (Eds.), Proceedings of the TMCE 2006 Ljubljana, Slovenia: Sixth International Symposium on Tools and Methods of Competitive Engineering pp. 285–298.
- X. Leiceaga, O. Ruiz, C. Vanegas, M. Rodríguez, J. Prieto, E. Soto. Bi-Curve And Multi-Patch Smoothing with Application to the Shipyard Industry. Accepted for presentation at the ADM-INGEGRAF conference, to be held in Perugia, Italy, in June 2007.
- Ruiz O, Vanegas C, Statistical Assessment of Global and Local Cylinder Wear. Accepted for presentation at the IEEE 5th International Conference on Industrial Informatics, to be held in Vienna, Austria, in July 2007.
- Ruiz O, Vanegas C, Cadavid C. Principal Component and Voronoi Skeleton alternatives for curves reconstruction from noisy point sets. Accepted for publication in the special issue on shape search, reconstruction and optimization of the Journal of Engineering Design.

3. Assistance to Conferences related to this Graduation Project

As a result of these achievements, Carlos has been awarded two travel grants by EAFIT University: the first one to present our paper on Curve Reconstruction in Ljubljana, Slovenia, in April 2006, and the second one to present our paper on Curve and Surface Smoothing for Naval applications in Perugia, Italy, in June 2007.

4. Hourly Intensity for a Graduation Project

The total time that an undergraduate student of Mathematical Engineering (Applied Mathematics) is required to designate to the development and writing of his undergraduate is 150 hours. During his work in my laboratory, and in the laboratory of Prof. Xoan Leiceaga, Carlos has worked an average of 25 hours per week, 42 weeks per year, during 3 years (over 3100 hours in total). Based on this fact, I estimate that the total number of hours that Carlos has carried out research activities directly related to the material included in this work is roughly 1700 hours.

5. Grade Point Average

Carlos holds an overall undergraduate grade point average (GPA) of 4.7 in a 5.0 point scale. It must be pointed out that, unlike relative grading systems, EAFIT University uses an absolute grading scale. Under such a circumstance, a full score (5.0) can only be achieved by a student with full score in every homework, exam, project and other grade in a course. Therefore, a 5.0 grade is extremely uncommon.

6. Graduate Study Perspectives

As usual in the CAD CAM CAE Laboratory at EAFIT, Carlos Vanegas wishes to begin graduate studies immediately after obtaining his B.Sc. degree. In August 2006, Carlos began his application process to the Doctoral Program in Computer Science at Purdue University, Indiana, USA. In February 2007, Carlos was offered a Graduate Assistantship to begin his studies at Purdue in Fall 2007 (a top and competitive research University in USA). According to the Dept. of Computer Science at Purdue, the Graduate Program in Computer Science received over 640 applications for 20 slots (96.8% rejection probability), for Fall 2006.

I must point out that Carlos had open doors at several world-class universities and research institutions for his Doctoral track. I am glad about his choice for Purdue University. I know that such a choice will be good for both parties involved.

Sincerely Yours,

Prof. Oscar E. Ruiz
oruiz@eafit.edu.co
Coordinator CAD CAM CAE Laboratory
EAFIT University, Medellin, COLOMBIA

Introduction

An increasing number of industrial CAD CAM CAE applications deal at a given stage with geometric problems, for which solutions must be devised using elements of computational geometry. Computational geometry is the study of algorithms and data structures for the solution of geometric problems and the manipulation of geometric entities. Stochastic geometry is the mathematical discipline which studies mathematical models for random geometric structures. This work is a compilation of computational geometry methods that have been devised and implemented in the CAD CAM CAE Laboratory at EAFIT University (Medellin, COLOMBIA) and the Group of Engineering and Design at University of Vigo (Vigo, SPAIN). Such methods are solutions proposed to three different geometric problems, all of them originating in real CAD/CAM/CAE industrial applications: curve reconstruction, assessment of cylindricity, and curve and surface smoothing.

Subfields of computational geometry include combinatorial geometry, stochastic geometry, and numerical geometry. Specific examples of each of these subfields can be found within the methods presented in this work. For instance, the point set partitioning problem, which is a step in the processing of the point set in both the curve reconstruction and cylindricity assessment problems, belongs to the subfield of combinatorial geometry. Another problem of this type is the Delaunay triangulation of a point set, which is a step in the deterministic method proposed to the curve reconstruction problem.

Two of the problems attacked in this work (curve reconstruction and assessment of cylindricity) deal with input data sets exhibiting a stochastic nature. Consequently, statistical methods (including Principal Component Analysis -PCA), were used in the proposed solutions for the problems. PCA aims to reduce the dimension of data sets in order to simplify further data processing. The dimension is reduced by eliminating those variables that contribute the least to the variance of the data set. In the problem of curve reconstruction, PCA is used by us to reduce the dimension of a local region of the point cloud from 3 to 1, by identifying the direction in space in which the point set presents the greatest variance. In the case of cylindricity assessment, PCA is used to determine the longitudinal axis of the cylinder from a set of sampled points. The axis is used to calculate a transformation that unwraps the point set in order to bring the point set to a space in which analysis of cylinder wear is significantly easier.

The curve- and surface-smoothing problem presented here requires the manipulation of B-spline curves and surfaces so that given continuity conditions can be achieved throughout the union of several surfaces. For this purpose, the control points of such curves and surfaces are modified following geometric criteria of continuity. The method proposed to solve this problem represents an application of numerical geometry, since geometric modeling by means of spline curves and surfaces is considered to lie in this field. The problem of calculating new positions for control points so that tangent plane continuity is achieved in a point common to four patches, belongs to the field of combinatorial geometry. In this case, an heuristic approach is proposed. The convergence from the dynamic-system point of view, of such heuristic algorithm is intuitively discussed in this document, but no mathematical proof of convergence is provided, as it somehow falls outside of our intended scope.

The reconstruction of a curve from a noisy, unorganized point sample of it, is one of the most important problems in the reverse engineering of geometric models. To be solved in this work, this problem requires the application of combinatorial geometry and spatial statistics.

The assessment of cylindricity of nominally cylindrical objects from a sample taken on the surface of the object is a relevant problem in metrology, since a large fraction of mechanical parts are cylinders. In particular, in the injection of plastics and polymers, the wear of the extrusion cylinder represents an important portion of power losses. Therefore, its diagnostic is economically attractive for the parties involved. Applications of spatial statistics and combinatorial geometry build up the solution proposed.

The smoothing of B-spline curves and surfaces is a frequent problem in the modeling of ship hulls, since B-splines have been used extensively in the past to define ship hull geometry for design purposes. Due to the complex shape of some ship hulls, it is preferred to use a collection of B-splines patches to model the hull rather than a single patch. Automatically achieving tangent plane continuity throughout the collection of patches is then desirable for design purposes, since the manual repositioning of control points defining the patches is tedious and unfeasible in most cases. Methods of numerical geometry and (heuristic) combinatorial geometry are used in the implementation of the automatic process proposed here.

Contents

List of Figures

CHAPTER 1

Principal Component and Voronoi Skeleton Alternatives for Curve Reconstruction from noisy Point Sets

CONTEXT: The CAD CAM CAE Laboratory at EAFIT University, under my coordination, started the research of the application of a statistical method, Principal Component Analysis (PCA), to geometrical scenarios in 2000. Through the years, several papers have been written on this subject, progressing in the level of application and formalization of PCA to Stochastic Geometry. This work has been founded by EAFIT University and the Colombian Council of Research and Technology (COLCIENCIAS) in several research projects. In addition, the German Service of Academic Exchange (Deutscher Akademischer Austauschdienst - DAAD) has also founded my visiting research at the Max-Planck-Institut fur Informatik at Universität des Saarlandes in 2004, where further research on the topic was carried out.

Carlos Vanegas, research assistant under my direction in the CAD CAM CAE Laboratory, was able to program the application of the devised methods to large sets of data. For such a purpose, theoretical contributions were needed, which appear in two papers:

- O. E. Ruiz, and C. A. Vanegas, "Piecewise linear curve reconstruction from point clouds," in Proc. 6th International Symposium series on Tools and Methods of Competitive Engineering, Ljubljana, Slovenia, April 18-22, 2006, pp 285-298.
- Ruiz O, Vanegas C, Cadavid C, "Principal Component and Voronoi Skeleton alternatives for curves reconstruction from noisy point sets," To be published in the special issue on shape search, reconstruction and optimization, of the Journal of Engineering Design.

As co-authors of such publications, we give our permission for this material to appear in this document. We are ready to provide any additional information on the subject, as needed.

Prof. Oscar E. Ruiz
oruiz@eafit.edu.co
Coordinator CAD CAM CAE Laboratory
EAFIT University, Medellin, COLOMBIA

ABSTRACT. Surface reconstruction from noisy point samples must take into consideration the stochastic nature of the sample. In other words, geometric algorithms reconstructing the surface or curve should not insist in following in a literal way each sampled point. Instead, they must interpret the sample as a "point cloud" and try to build the surface as passing through the best possible (in the statistical sense) geometric locus that represents the sample. This work presents two new methods to find a Piecewise Linear approximation from a Nyquist-compliant stochastic sampling of a quasi-planar C^1 curve $C(u) : R \rightarrow R^3$, whose velocity vector never vanishes. One of the methods articulates in an entirely new way Principal Component Analysis (statistical) and Voronoi-Delaunay (deterministic) approaches. It uses these two methods to calculate the best possible tape-shaped polygon covering the planarised point set, and then approximates the manifold by the medial axis of such a polygon. The other method applies Principal Component Analysis to find a direct Piecewise Linear approximation of $C(u)$. A complexity comparison of these two methods is presented along with a qualitative comparison with previously developed ones. It turns out that the method solely based on Principal Component Analysis is simpler and more robust for non self-intersecting curves. For self-intersecting curves the Voronoi-Delaunay based Medial Axis approach is more robust, at the price of higher computational complexity. An application is presented in Integration of meshes originated in range images of an art piece. Such an application reaches the point of complete reconstruction of a unified mesh.

1. Introduction

Reconstructing a curve or a surface from a point set is one of the most important problems in the reverse engineering of geometric models. In some cases curve reconstruction plays an important role in the surface reconstruction problem [21]. It is the goal of this paper to present two methods involving statistical (Principal Component Analysis -PCA) and deterministic techniques (Voronoi-Delaunay) for reconstructing a set of curves from noisy unorganised point sets. An application for surface reconstruction is presented, using data sets resulting from objects captured by range images. The references examined indicate that such a combination of methods has not been tried before for curve and surface reconstruction, or for range image mesh integration.

Even though this work will concentrate on quasi-planar curves, the statistical methods involved directly extend to arbitrary curves in $3D$. Two types of noisy unorganised point sets have been considered. One of them results from sampling and adding statistical noise to a set of mutually disjoint *regular* parametric curves (i.e. whose first derivative vector is continuous and never vanishes) $C_i(u)$ in R^3. The other point sample is originated in a cluster of individual meshes from range images. The point samples are assumed to comply with the Shannon or Nyquist criteria for digital sampling.

Problem Statement. Given a sample $S = \{p_0, \ldots, p_N\}$ from an (unknown) set of mutually disjoint regular (open or closed) quasi-planar parametric curves $C_i(u)$ in R^3 and which may self-intersect, a PL (Piecewise Linear) estimation of each $C_i(u)$ is to be found. As seen later, without loss of generality we may assume that $C \subset R^2$.

The statistical methods which estimate the tangent to a curve $C_i(u)$ are not capable of determining by themselves the correct sense of the $\pm v$ tangent vector. For this reason we require that the curve has certain continuity in the derivative and that in the neighbourhood of each of its points it is well approximated by a straight line. That is, $C_i(u)$ must be C^1-continuous and its velocity vector must never vanish (i.e. the curve must be *regular*).

In this paper the stated problem is solved and an application of its solution is presented, for integration of range image meshes. To integrate a set of meshes of individual range images, the set of meshes is sliced by parallel planes. Each slice S_k turns out to be a coplanar set of points $S_k = \{P_{0_k}, P_{1_k}, \ldots, P_{N_k}\}$ with a strong statistical component stemming from the optical sampling error. The proposed algorithm finds a PL estimation of the curve $C_k(u)$ that adequately fits the points in the noisy unorganised point set S_k. The Literature Review section illustrates that such an integration of individual range meshes is still an open problem in several aspects. Section 5.1 discusses the application of PL curve reconstruction in detail.

Another application of the proposed algorithms in integration of individual range meshes arises when a particular slice k is missing or incomplete In the case of range imaging, this occurs when a portion of the object is not captured by any of the images. In such a case, point samples from levels $k-1$ and $k+1$ are projected onto the insufficiently sampled plane k. The resulting cross section on plane k must then be recovered from a possibly noisy point set. This point set should be treated with statistical tools, and the cross sections recovered should be the best fit to the planar point cloud contained in plane k.

A variant of the first type of noisy point sets (used to illustrate the Voronoi-Delaunay method) consists of a noisy sample of a self-intersecting planar parametric curve. Figure 1.1 shows a situation in which the local geometry of a planar slice (for example a Computer Axial Tomography - CAT) added to the presence of stochastic noise renders a set of points that look like the one in Figure 10(a). Clearly, less extreme situations may render an "8"-like section in the presence of a high level of stochastic noise. In the case of a sample of an "8"-like section two legal resulting PL approximations are equally likely: (a) *two* separate circular polygons, and (b) *one* polygon with a thin wasp waist. It is clear that near the self-intersecting point any algorithm may be confused. A survey of reverse engineering methods is presented in Varady, Martin and Cox [31], being evident the use of curve reconstruction from point samples for generation of revolution or extrusion 2-manifolds. One of such applications is presented by Lee [21]. This application is particularly important in reverse engineering when the designer interactively tests the fitting of such surfaces to specific portions of the point set.

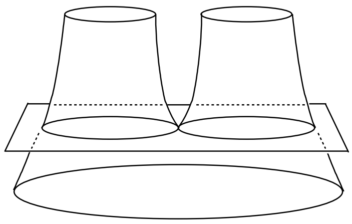


FIGURE 1.1. 2-manifold sample which renders a non-manifold curve.

2. Literature Review

Several solutions are available for curve reconstruction from point sets without noise. A survey on techniques for the case of closed, smooth, and uniformly sampled curves can be found in Edelsbrunner [11]. Methods for non-uniformly sampled smooth curves, and for uniformly sampled non-smooth curves are cited by Althaus *et al.* [2]. Some TSP (Travelling Salesman Problem) and tour improvement heuristics were used by Althaus and Mehlhorn [3], and good experimental results were reported. In Amenta, Bern and Eppstein [4] the PL approximation of a C^2 curve sampled in a dense pattern proportional to its local feature size (a modification of the Nyquist criterion) is discussed. Two graphs, the *crust* and β-*skeleton* are discussed, whose edge set exhaust the point sample. It should be noted that the curve reconstructed by these algorithms passes through each of the sampled points, and this type of solution is not adequate for the noisy point sets considered in the present paper.

The methods proposed for the case of non self-intersecting unorganised noisy point sets include spring energy minimization [12], implicit simplicial curves [29], α-shape polygonal boundaries and medial axes [10], and moving least squares [21]. A review of these methods along with their limitations can be found in Lee [21]. Verbeek *et al.* [32] approximate an open curve by k segments that are least squares approximations of point subsets contained in Voronoi regions for sets of segment. By increasing k, better approximations to the curve $C(u)$ are found until a fitting criterion is met. However, the segments still need to be joined in a Hamiltonian graph, significantly adding to the complexity of the algorithm. The segments of the Hamiltonian graph may be larger than the segments found in fitting the point set. This has the effect of producing a PL approximation that may be deformed when compared with the $C(u)$ curve.

[7] attack the problem of noisy point samples by computing a new point set having less noise as than the initial point set. The actual PL approximation to $C(u)$ is computed using a crust algorithm (in this case the NN Crust by Dey and Kumar [9]). The new point set is calculated as follows: for each sample point p a thin rectangle is built with its main axis normal to the curve tangent and covering a certain number of point samples. The centre of such rectangle replaces p for the remaining of the algorithm run. The rectangle centres are closer to the $C(u)$ curve than the original sampled points. From all these rectangle centres one keeps the most external ones. In this way, the point set is pruned while a supporting width for crust algorithms is provided. At the end, a crust algorithm is called. In the method discussed in our paper, a ball $B(p, r)$ replaces the rectangle, and the centre of mass of the points inside the ball is assumed to be on $C(u)$. Also, the ball contains a point set whose main trend is tangent to $C(u)$ instead of normal to it. In our approach, no additional crust algorithm is needed, since the PL approximation to $C(u)$ is directly built using the centre of mass of those points in the sample which are contained in the ball.

Wang, Pottmann and Liu [33] fit B-splines to a set of noisy point sets using curvature - based squared distance minimization. For this reason, the minimization requires the form of the equation (spline), and makes no attempt to attack noisy point sets with self - intersecting conditions. On the other hand, no discussion of the complexity of the algorithm is provided in time or in computational space. We feel that keeping the objective as a PL curve avoids the literal formulation of B-splines in the algorithm. Also, our research has as a goal the representation of non-manifold curve samples as PL non self-intersecting curves (i.e., manifold topologies), which allow for the subsequent usage of the PL curves in geometrical or topological constructs.

Kegl [19] and Kegl and Krzyzak [20] explore the recovery of a Principal Graph underlying a 2D point sample (e.g. a character meant to by pen strokes). The authors set up a numerical optimization algorithm that weights two competing criteria in the graph: (i) should as closely as possible follow the many pixels in the stroke, and (ii) should not have high curvature portions. An important feature for the application of this algorithm is that, since a character is sought, the final P.L. approximation does not have to be a manifold. Therefore, self-intersections are permitted (like in the “H” or “8” characters). In our case, the final result of the reconstruction should be a set of disjoint non self-intersecting curves, and therefore one must take care of higher requirements than the ones [20] and [19] met.

Range Images and Point Set Surfaces Because the algorithms proposed in this paper are to be applied to the integration of range images, the authors consider that a review on range images is worth as a motivation for the reader. Range imaging offers a manner of digitizing the shape of three-dimensional objects. Because all opaque objects self occlude, no single range image suffices to describe the entire object, making necessary the combination of a collection of range images from diverse viewpoints into a single polygonal mesh that completely describes the object. Turk and Levoy [30] create individual meshes for the different range images and clip them against each other for integration. Unfortunately, their integration method shows instabilities documented in [8]. [8] integrate range images by creating a scalar field containing the minimal signed distance $f(x, y, z)$ from the point (x, y, z) to the object's surface. Afterwards, a Marching Cubes algorithm creates the B-Rep of the iso-surface $f(x, y, z) = 0$. A shortcoming of this method is the fact that the signed distance is calculated as a directional (instead of a scalar) property, and therefore there is no guarantee that the scalar field correctly registers the signed distance from a point to the surface. In Soucy and Laurendeau [28] the very high computational cost of combining range image meshes after registration and surface meshing is discussed. In this reference overlapping components of the meshes corresponding to different range images must be identified, with a large computational cost, of the order $O(2^N - 1)$ where N is the number of range images. This reference unrealistically assumes the accuracy of the range data, as precision of the range data deteriorates in the periphery of each range image. In Zhou, Liu and Li [34] a heuristic method for merging overlapping triangular meshes from range images is discussed. This article does not prove the correctness of the method exposed, which is based on the distance between triangles that are considered as overlapping. The less likely mesh is projected against the more likely one, based on a purely geometric projection, giving rise to topological inconsistencies that are not dealt with rigorously.

For the direct treatment of the integrated point cloud from individual range images Hoppe et al [18] use the k nearest point neighbours of a particular point p in the cloud to estimate the best local tangent plane. The plane is then used to construct the signed distance function $f(q) : R^3 \to R$ from point q to such plane. A Marching Cubes algorithm is then used to construct an approximation for the manifold $f(q) = 0$. This reference does not discuss the reconstruction of manifolds with border, nor the behaviour of the algorithm in incorrectly smoothing sharp edges of the piece. Indeed, their examples show a strong trend to filter out high frequencies. For these reasons, directly fitting surfaces to point sets has been an open research field since 1992. As a consequence, there has been a steady stream of publications in this direction. Ohtake et al [25] use spherical influence regions to calculate most likely points on the surface and local normal vectors. For these authors and others, however, a difficulty with direct reconstruction of the manifold from the integrated point cloud remains in the fact that stitching together the local planes (triangles) gives rise to non-manifold topologies. Adamson and Alexa [1] propose the computing of ellipsoidal weighting functions per sample to represent an implicit surface using supporting regions around each sample (Point Set Surfaces). It must be noted that such an approach does not explicitly compute the Boundary Representation of the model. Instead, it lends itself for visualization with ray casting.

The authors of the present article have found that the issues arising in curve reconstruction and in a possible application of it to range mesh integration are still an open problem in applied computational geometry. As seen from the literature review, curve reconstruction of self-intersecting curves is also unsolved. In range images, a reliable algorithm for mesh integration has not been proposed. Even in commercial systems [26] such an integration requires the user interaction for correcting self-intersecting portions, holes, etc., that are left after the triangulation merges. Such facts have encouraged the authors to publish the present paper.

Section 3 examines the adaptation of statistical methods to be used in the present problem. Section 4 discusses the concepts necessary to implement the algorithms and their articulation in reaching the solution. Results for several types of point sets including non-smooth, self-intersecting, and non-uniform sets obtained with both methods are presented in Section 5. Section 5.1 describes an interesting integration of one of the methods to surface reconstruction from range images, and presents the results obtained for a real object. Section 6 discusses the computational complexity of the implemented methods. Finally, Section 7 draws the relevant conclusions, and proposes bases for future work.

3. Statistical Approach

The statistical approach for curve reconstruction from point samples has precursors in Hastie and Stuetzle [17]. In this reference, the authors define Principal Curves as smooth ones, which pass through the middle of, and are self-consistent with, a sampled cloud of n-dimensional data with dispersion (relative to the unknown curve) following a normal distribution (μ, σ).

3.1. Principal Component Analysis (PCA). Although the following discussion deals with noisy point sets in R^2 and R^3, it may be useful to know that the stochastic analysis presented is applicable to samples in n dimensions (in fact, the Principal Component Analysis method was developed for the treatment of samples in n-dimensional space, with $n >> 3$).

Let $S = \{p_i \in R^n : 1 \leqslant i \leqslant m\}$ be a set of m sample points in R^n . Without loss of generality one may assume that

$$\mu_1 = \mu_2 = ... = \mu_n = 0 \tag{1}$$

meaning that the expected value of the n-dimensional distribution or the p_i's is the origin of R^n. Let Σ be the covariance matrix of the sample, where $\Sigma_{i,j}$ is the covariance of the i^{th} against the j^{th} component of the p_i points.

One is interested in rotating S with an orthogonal transformation R such that the new set $S' = \{q_i \in R^n : 1 \leqslant i \leqslant m\}$ of transformed sample points $q_i = R * p_i$ presents maximal dispersion in the direction of the first axis of R^n, the second maximal dispersion in the direction of the second axis, and so on. For a $3D$ point set that has a stochastic *linear* trend, establishing the direction of maximal dispersion is equivalent to identifying the direction vector of the line from which the sample was taken. For a $3D$ point set with an stochastic *planar* trend, establishing the direction of minimal dispersion identifies the normal vector of the plane from which the sample was taken.

Let X_p, Y_p, Z_p be the unit vectors pointing in the directions in which S has the largest (σ_x), second largest (σ_y) and smallest variance (σ_z), respectively. It may be shown that

(1) The pairs $(\pm X_p, \sigma_x)$, $(\pm Y_p, \sigma_y)$, and $(\pm Z_p, \sigma_z)$ are eigenvector - eigenvalue pairs of the matrix Σ:

$$\begin{aligned} \Sigma * (\pm X_p) &= \sigma_x * (\pm X_p) \\ \Sigma * (\pm Y_p) &= \sigma_y * (\pm Y_p) \\ \Sigma * (\pm Z_p) &= \sigma_z * (\pm Z_p) \end{aligned} \tag{2}$$

(2) $\pm X_p$, $\pm Y_p$, $\pm Z_p$ are mutually orthogonal:

$$X_p \bullet Y_p = X_p \bullet Z_p = Z_p \bullet Y_p = 0 \tag{3}$$

(3) $R * [X_p, Y_p, Z_p, O_p] = [X_w, Y_w, Z_w, O_w]$ and therefore:

$$R = \begin{bmatrix} X_p & Y_p & Z_p & O_p \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} X_p^T & 0 \\ Y_p^T & 0 \\ Z_p^T & 0 \\ O_p^T & 1 \end{bmatrix} \tag{4}$$

where $[X_w, Y_w, Z_w, O_w]$ is the World Coordinate System or a fixed reference frame. Without loss of generality, one may assume that $X_w = [1, 0, 0]^T, Y_w = [0, 1, 0]^T, Z_w = [0, 0, 1]^T, O_w = [0, 0, 0]^T$ and therefore the right hand side of item (iii) above is a clipped 4×4 identity matrix. Because an eigenvector can always be normalized, it can also be assumed that $|X_p| = |Y_p| = |Z_p| = 1$. Equation (4) results from the completion of the identity matrix in item (iii) and its (trivial) inversion.

As a result, $[X_p, Y_p, Z_p, O_p]$ is easily found and constitutes a right handed coordinate system. In particular, $[X_p, Y_p, Z_p]$ is an orthogonal matrix. As desired, a parametric line $p(\eta) = O_p + \eta * X_p$ which passes through the centre of gravity of the point cloud S is found by sorting and naming the eigenvector-eigenvalue pairs of Σ so that $\sigma_x \geq \sigma_y \geq \sigma_z$.

From Equations (2) and (4) it is clear that for quasi-planar data set, the eigenvector Z_p associated to σ_z is the estimation of the direction normal to the fitting plane, since σ_z is by definition the direction of minimal dispersion of the (quasi-planar) set of points. Conversely, for line data, the estimation of the direction vector of the line is the eigenvector X_p, since it is associated to the eigenvalue σ_x representing the maximal dispersion.

3.2. Least Squares Fitting. Section 3.1 explained how the coordinate system $[X_p, Y_p, Z_p, O_p]$ is calculated using PCA, by computing the eigenvector-eigenvalue pairs $(\pm X_p, \sigma_x)$, $(\pm Y_p, \sigma_y)$, and $(\pm Z_p, \sigma_z)$, of matrix Σ . Because geometric kernels do not usually have routines for calculation of n-dimensional eigenpairs, a method was devised for the 3-dimensional case at hand. The method takes advantage of the fact that point samples from Coordinate Measurement Machines, Machine Tool stylos, CAT scans, etc., are planar or quasi-planar. As a consequence, a very close estimation of the lowest dispersion direction (the vector Z_p normal to the plane) can be easily achieved. The point cloud projected on this plane loses one dimension and therefore the problem becomes 2-dimensional. Therefore, a solution of the eigenpair problem in Equation (2) can be achieved as an extension of a Least Squares (LS) fitting. The LS method cannot be directly applied since it is based on the *implicit* equation $y = mx + b$, which has no solution if m is the tangent of $\pm 90^o$. A random rotation around z, followed by LS fitting and the corresponding counter rotation of the point data set, avoids this problem and allows to express the 3D trend of the point cloud in terms of a *parametric* equation $p(\eta) = O_p + \eta * X_p$.

In two dimensions, the LS method detects the trend m of a linear phenomenon. Since the 3D problem at hand is projected into 2D space, finding m in 2D reduces to calculating the projection of the 3D direction vector X_p of $p(\eta)$ onto the best fitting plane for the point set. Notice that the point set is not exactly planar because of the machine tool sampling errors. Since Least Squares is just a PCA in two dimensions, in what follows, "PCA" and "Least Squares" should be read as synonyms.

3.3. Point Sample Partition. Regardless of the method employed to estimate a PL approximation for the curves, it is capital to recognize the fact that the data set must be partitioned into the data sets originated from the individual curves $C_i(u)$. In order to perform such a partition let us define an equivalence relation on the point set S, as follows. If the sampling conditions are anisotropic and constant over R^3, a point $p \in S$ is said to "be the extended neighbour" of a point $q \in S$, if and only if there exists a sequence of points of the sample S starting at p and ending at q such that no two consecutive points of the sequence are farther apart by more than a fixed distance ϵ from each other. Let $r(p, q)$ be read as "p is an extended neighbour of q". Formally, two points p, q are Extended Neighbours of each other, whenever there exists a sequence $[q_1, ..., q_w]$ such that each $q_i \in S$, $q_1 = p$, $q_w = q$ and $|q_i - q_{i+1}| \leq \epsilon$. The $r(\ )$ relation defined above is an equivalence relation since it satisfies:

(1) $r(P_i, P_i)$ (reflexive: a point P_i is extended neighbour of itself).
(2) $r(P_i, P_j) \wedge r(P_j, P_k) \rightarrow r(P_i, P_k)$ (transitive: if P_i and P_j, and P_j and P_k are extended neighbours, P_i and P_k are so).
(3) $r(P_i, P_j) \rightarrow r(P_j, P_i)$ (symmetric: if P_i is extended neighbour of P_j then P_j is extended neighbour of P_i).

This equivalence relation $r(\ )$ splits S into subsets $S_1, S_2, ...$ with the property that $r(P_i, P_j)$ holds (are extended neighbours) if and only if P_i and P_j belong to the same S_k. Properties (i), (ii) and (iii) of the relation $r(\ )$ imply that $\cup_i S_i = S$ and $S_i \cap S_j = \phi, i \neq j$. Each S_i of the partition happens to be the set of points sampled from the curve $C_i(u)$. The partition of the set S by the equivalence relation $r(\ )$ is realized by using a standard algorithm of *transitive closure* which will not be discussed here.

4. Algorithms

Two algorithms for determining a PL approximation for quasi-planar 1-manifolds in R^3 are presented in this section, along with two figures that show partial results obtained at the main steps of each one of them.

4.1. Data Pre-Processing. The point data must be pre-processed in the following sequence: (i) Scaling: to guarantee that a standard bounding box of the set S is available (PCA requires such a box). (ii) Partition: to divide S into subsets, each one containing the points of S corresponding to an individual $C_i(u)$ curve. (iii) Identification of Best Plane: to find a statistical plane Π fitting the quasi-planar point set S. (iv) Correction to Planar Set: to project S onto Π in order to have a perfectly planar point set. (iv) Transformation to XY Plane: to use the algorithmic results in literature which deal with point sets in the XY plane. Step (ii) is required since several $C_i(u)$ curves may have been sampled and the point set would represent several unrelated curves. In what follows, the notation $C_i(u)$ will be changed to $C(u)$ since the analysis is per curve. A post-processing step consisting in reversing the transformations performed in the pre-process, is necessary in order to bring the found solution back to the original space.

4.2. Curve Reconstruction with Least Squares. After the data pre-processing steps mentioned in Section 4.1 take place, the Least-Squares-based algorithm takes as input a quasi-planar set S, and returns a polyline that fits these points by performing the steps discussed below and displayed in Figure 1.2.

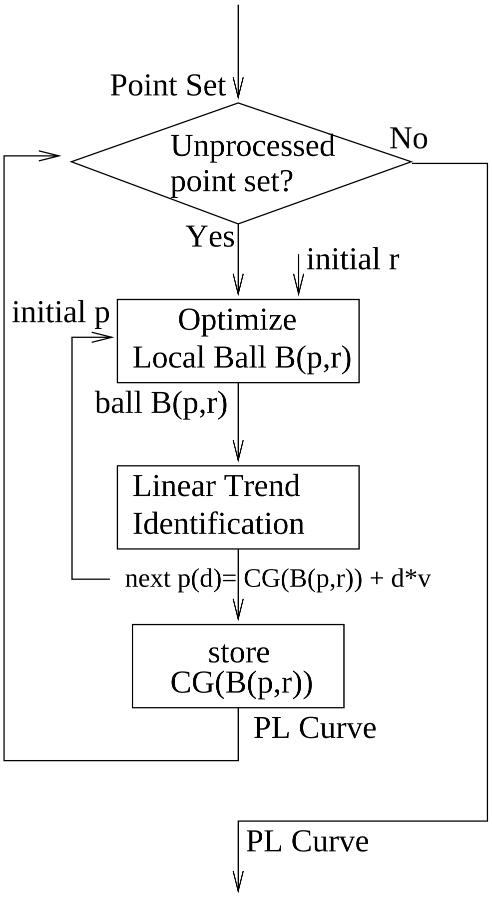


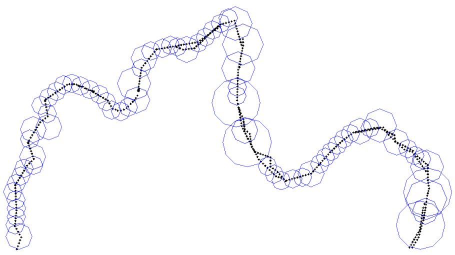
FIGURE 1.2. Curve Reconstruction with Principal Component.

4.2.1. *Optimal Local Point Set Estimation.* Given a noisy unorganised point set, resulting from a stochastic sample with variance $[\sigma_x, \sigma_y, \sigma_z]$ of a planar 1-manifold $C(u)$ (possibly open) lying on plane Π in R^3, one is interested in estimating the tangent line $dC(u)/du|_{u=u^*}$, at point $C(u^*)$ of the curve $C(u)$. PCA and Least Squares are applied to points of the sample which are contained inside a ball $B(P_s, R)$, centred at a seed point P_s and having radius R. Two competing aspects must be compromised: (i) the ball should be small enough so that data set S can be considered to fit a linear estimation of the local tangent; (ii) the ball should be large enough so that the goodness of the linear estimation is kept. To achieve (i) and (ii) an iterative search is conducted for a combination of P_s and R, optimal for the linear fitting of local neighbourhoods of S. The iterative search starts with a ball $B(p(0), r(0))$ enclosing a set $S(0)$ of points. Let $\epsilon\,(p, r)$ be a function that associates a least-square regression fitting error to the points inside a ball with centre p and radius r. It is desired to find the values of p and r that minimize ϵ. Applying the PCA to the point set, a measurement of the fitting error is found. In the $k-th$ iteration a new value ot $r(k)$ is proposed ($r(k+1)$), which changes the size of the ball $B(r(k+1), p(k))$. This ball, in turn will enclose a different set of points $S(k+1)$, with new centre of gravity $p(k+1)$. The fitting of a new straight line to the set $S(k+1)$ will render a new fitting error. The iterations stop when such an error has a local minimum. This *ad hoc* process was found to have good convergence.

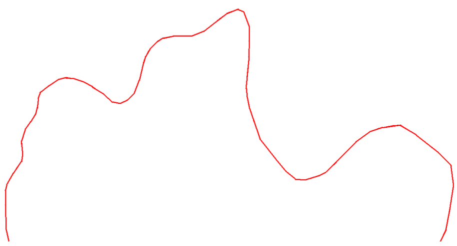
4.2.2. *Piecewise Linear Reconstruction of* $C(u)$. In the following discussion the term $B(p, r)$ will mean both the ball with radius r centred at p, and the subset of the point sample contained in such a ball. The context will define which meaning is intended. The algorithm in Figure 1.2 performs an estimation of the linear trend of the points in the optimized ball $B(p, r)$. Such an estimation produces a parametric equation for a straight line in space $p(\eta) = O_p + \eta * v$ with $|v| = 1$, where O_p is the centre of gravity of the points inside $B(p, r)$, v is the linear trend of the line (also called X_p in section 3.1). O_p is denoted as $CG(B(p, r))$ in Figure 1.2. Such a point is stored directly in the linear estimation of the $C(u)$ curve. The ball for the next iteration is initially centred at $O_p + d * v$, where d is the progression step of the algorithm and $|v| = 1$. Since $B(p_i, r_i)$ and $B(p_{i+1}, r_{i+1})$ intersect, it is clear that each sampled point may be used in several balls, and therefore in the estimation of successive tangents. Notice that the index i corresponds to already optimised balls in evolving localities of the curve $C(u)$ such that $|p_{i+1} - p_i| \approx d$. In Figure 1.2 the need for determining whether $+v$ or $-v$ is the correct trend is omitted (recall that PCA returns $\pm v$). This is easily done by ensuring that $v_i \bullet v_{i+1} > 0$. The later requirement is reasonable since the curve $C(u)$ is assumed to be regular.

The algorithm will continue as long as there are enough available points of the set S (see section 3.3) which fall inside a ball. Each point can be used in several balls, being their number set by the user. In Figure 1.2 the marking of the multiply used points of S is omitted for the sake of clarity. When this algorithm terminates, the curve $C(u)$ has been piecewise linearly estimated.

A noisy point set generated from a range image Multi-Mesh sample is shown in figure 3(a), together with the balls used by the reconstruction algorithm. Figure 3(b) shows the resulting reconstructed curve.



(a) Noisy Point Set from Range Image Multi-Mesh sample and Balls used in the reconstruction process.



(b) Reconstructed Curve.

FIGURE 1.3. PCA-based Reconstruction.

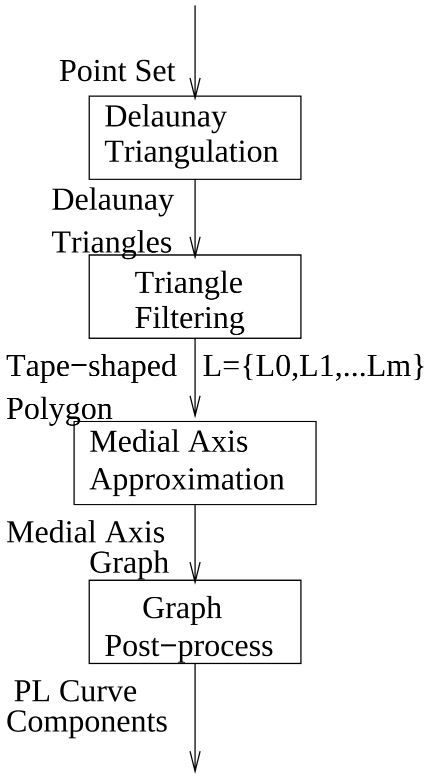


FIGURE 1.4. Line Reconstruction through Delaunay-Voronoi Techniques.

4.3. Principal Curve via Delaunay Triangulation. The following discussion will be illustrated using a planar 1-manifold with border (open $C(u)$). Later on, the concepts explained will be applied on self-intersecting (i.e. non-manifold) planar curves.

For planar self-intersecting curves, PCA alone is not robust enough. Additional processing is required since the points in the neighbourhood of the self-intersection are exhausted for purposes of PCA estimation as the PL approximation crosses the first time over the intersection neighbourhood. As the PL curve revisits the intersection neighbourhoods no points are available for identifying the trend of the curve, and the algorithm tends to look for another point (i.e. curve) neighbourhood where to work, without really having reproduced the intersection. The result is an incomplete curve stage, therefore missing the self-intersection detail.

To deal with self-intersecting curves, it was decided to determine the tape-shaped polygon T_σ covering S (definition below). Figure 1.4 displays the algorithm discussed next.

Definition. Tape-shaped Polygon T_σ. Let $C : R \to R^3$ be a planar regular parametric curve, which may self-intersect. Without loss of generality let us assume that $C \subset R^2$.

Given $\sigma > 0$ a real number, define $T_\sigma = \{p \in R^2 : d(p, C) \leq \sigma\}$. There exists $\sigma_0 > 0$ such that if $0 < \sigma \leq \sigma_0$ then for every $p \in T_\sigma$ (i) the set of points $\{q_{p,1}, ..., q_{p,r_p}\} \subset C$ formed by those points whose distance to p equals $d(p, C)$ is finite. The $q_{p,i}$ points in C are the ones which realise the distance from p to C; (ii) the distance between any two points in the set $\{q_{p,1}, ..., q_{p,r_p}\}$ is less than 2σ. Observe that σ is dictated by the precision of the measurement device which samples C. We assume that the measurement device allows a precision of $\sigma \leq \sigma_0$ and therefore that $S \subset T_\sigma$. Note that the $q_{p,i}$ could be regarded as the points to be sampled in the curve C in absence of sampling noise and that if C is non self-intersecting then q is unique for each $p \in S$.

For small enough values of σ_0 (Nyquist samples) T_σ resembles a tape region covering the curve C. Let us define $Q_S = \{q \in C : d(p,q) = d(p,C), \text{for some } p \in S\}$. Note that if S has no statistical noise, $Q_S = S$ and Q_S would be a noise-free Nyquist sample of C.

4.4. Approximation of T_σ. Under the condition of S being a Nyquist-compliant sample, this article proposes an algorithm to approximate the tape-shaped polygon T_σ. The algorithm follows three steps:

(1) calculates the Delaunay Triangulation of S, $DT(S)$;
(2) then selects from $DT(S)$ small triangles;
(3) and finally, makes T_σ the boundary of the union of the triangles selected in (ii).

In order to apply such a method, an estimation of what a "small triangle" is, should be made precise. For this purpose the typical area and edge length of Delaunay triangles belonging to T_σ need to be estimated. To do that, PCA is iteratively run on neighbourhoods of the data set, thus determining the line $p(\eta) = P_0 + \eta * v$ that best approximates the tangent to the $C(u)$ curve in that neighbourhood. The points of S that produce such a fit are contained inside a ball $f_D * B(P_0, R_0)$ approximately centred on a local neighbourhood of $C(u)$. Delaunay triangles contained within a scaled version of this ball, namely $f_D * B(P_0, R_0)$ (with $f_D = 1.3$ being an empirically chosen enlarging factor) might be considered as "typical" of the ones forming T_σ, rendering "typical area" $\bar{A}$ and "typical edge length" $\bar{l}$ values.

One considers that a triangle is small if either of the following criteria ([14] and [16]) holds:

(1) **Enclosure**: Accept a Delaunay triangle DT_i if it is contained within the local PCA ball, that is, if $DT_i \subseteq B(P_0, R_0)$ where $B(P_0, R_0)$ is the best local PCA ball (see Figure 5(c)).
(2) **Area and Edge Length**: Accept a Delaunay triangle DT_i if its Area or maximal Edge Length are small. That is, if $Area(DT_i) \leq f_A * \bar{A}$ or if $E_{max} \leq f_l * \bar{l}$, respectively, for fixed constants f_A and f_l.

We give an informal discussion for the correctness of the procedure to obtain an approximation of T_σ. The tests run gave a good performance in the filtering of Delaunay triangles. An advantage of the implemented algorithm is that the application of PCA to the local neighbourhoods of the point cloud allows the estimation of the sizes of the triangles to be deleted and to be kept.

Let us suppose that, contrary to the assumption, a large triangle $DT_i = [v_j, v_k, v_l]$ belongs to T_σ. Since it is a Delaunay triangle, its circumcircle contains no points of S. But since DT_i is a large part of T_σ, a large portion of T_σ contains no sample points, contradicting the fact that S is a Nyquist sample. On the other hand, suppose that a small triangle $DT_i = [v_j, v_k, v_l]$ is not entirely contained in T_σ. If DT_i is completely outside T_σ, then it creates a contradiction since $S \subset T_\sigma$. If v_j, v_k, v_l are in T_σ but the triangle joins two approaching branches of C, the sample S is characteristic of a non-manifold situation and therefore DT_i is part of T_σ.

For the sake of simplicity T_σ will be denoted simply by T. An *approximation* of the medial axis of T, called here the *skeleton of T*, is the sought PL approximation of the $C(u)$ curve. Since the skeleton is a graph, it needs to be post-processed in order to extract from it the PL approximation of $C(u)$.

Figure 5(a) shows a data set from a planar non self-intersecting curve sampled stochastically. This figure presents a data set which has been already resized, its best plane estimated, and their points projected onto this plane, which produces a planar set. The Delaunay Triangulation of this point set is displayed in Figure 5(b).

4.4.1. *Polygon Synthesis based on Filtered Delaunay Triangulation.* The polygon L_0 obtained after application of criteria (i) and (ii) is shown in Figure 5(d). Observe that L_0 has no holes for this example. In that figure light triangles are the accepted ones based on the PCA criterion and dark triangles are the ones accepted based on area or edge length criteria. The following relations hold among accepted Delaunay triangles and their edges [22] :

(1) Each edge of an accepted Delaunay triangle DT_i has one or two accepted triangles incident to it.
(2) Edges $e_{i,j}$ in which Delaunay triangles DT_i and DT_j are incident are internal to the tape-shaped region T.
(3) Edges e_i in which only one Delaunay triangle DT_i is incident form the boundary ∂T. They may be either in the outermost or in an internal loop.

4.5. Medial Axis VS. Principal Curve. Figure 5(d) presents the minimal polygon T that covers the point set S. Its border ∂T, built by filtering the original Delaunay Triangulation, is coloured black in Figure 5(e). A very fine resample of the border ∂T) is then performed, and a Delaunay triangulation for this new point set is calculated. This new Delaunay triangulation also appears in Figure 5(e).

An approximation to the *medial axis* $MA(T)$ of T is a skeleton $SK(T)$, which is built in the following manner ([15], [5], [24]):

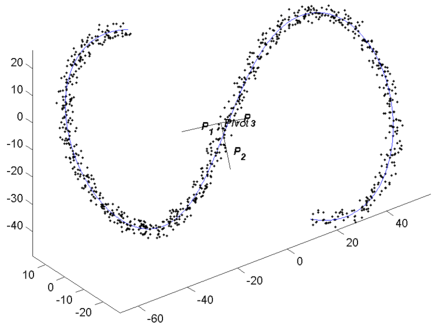
(1) Construct the Voronoi Diagram $VD(T)$ and Delaunay Triangulation $DT(T)$ of the vertices of T (see Figure 5(e)).
(2) Keep from $DT(T)$ only those Delaunay triangles contained in T. Call this set $\overline{DT(T)}$.
(3) Keep from $VD(T)$ only those Voronoi edges which are finite and are dual to the edges in $\overline{DT(T)}$. Call this set $\overline{VD(T)}$.
(4) If $\overline{VD(T)} \nsubseteq T$ then re-sample ∂T with a smaller interval and go to step (i) above. Otherwise, $\overline{VD(T)}$ is the sought skeleton of T, $SK(T)$.

As it is evident from Figure 5(f), the skeleton $SK(T)$ of the polygon T is a PL approximation of the 1-manifold $C(u)$.

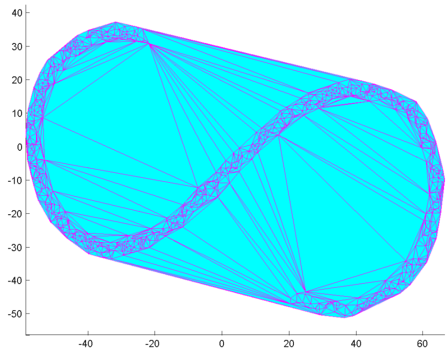
Notice that several resamples of ∂T may be needed in order to converge to $SK(T)$. Figure 5(e) shows one such resample. The boundary ∂T of the S-shaped polygon T in Figure 5(f) is sampled with a small enough interval. This tight sampling guarantees that the portion of the Voronoi Diagram confined to T, $SK(T)$, is acceptable as an approximation of $MA(T)$, the medial axis of T.

5. Results

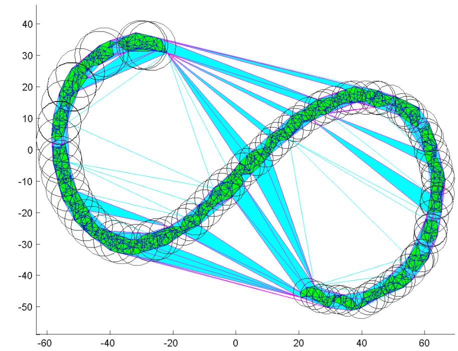
Section 5.1 illustrates three PCA curve reconstructions obtained for diverse point sets. It also discusses the application of PCA-based curve reconstruction to surface reconstruction from range images. Section 5.2 illustrates the results obtained using the Delaunay Triangulations methodology in dealing with the PL Approximation of planar 1-manifolds without Border (closed $C(u)$).



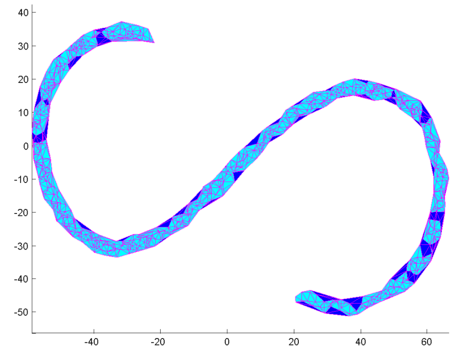
(a) Point Sample of Planar S-shaped $C(u)$ Manifold.



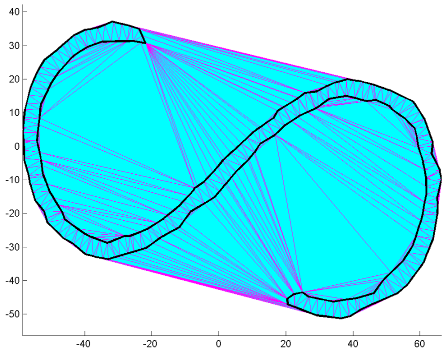
(b) Delaunay Triangulation of S-shaped Planar Point Sample.



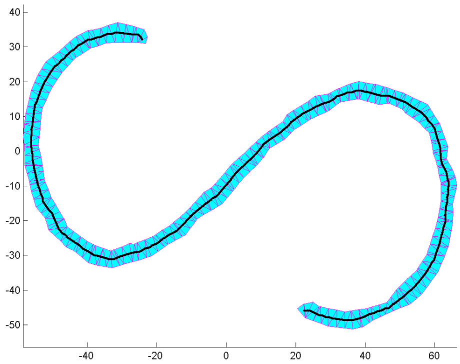
(c) Filtering of Delaunay Triangulation with PCA Balls.



(d) Triangles Selected by Area and Length Criteria.

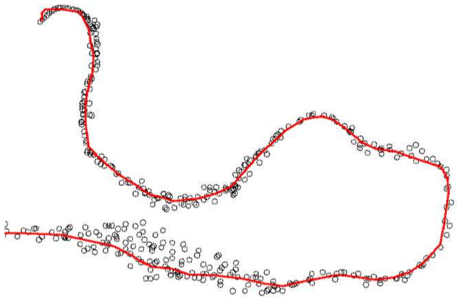


(e) Tape Polygon and its Delaunay Triangulation.

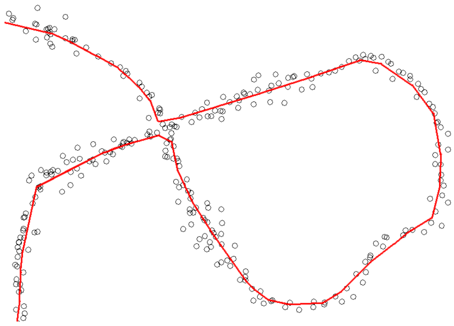


(f) Filtered DT and Skeleton.

FIGURE 1.5. Piecewise Linear Approximation of S-shaped $C(u)$ by Combined PCA and Voronoi-Delaunay Methods.



(a) Near Self-Intersecting, Non-Uniform Point Cloud.



(b) Self-Intersecting Non-Uniform Point Cloud.

FIGURE 1.6. Curve reconstructions obtained for different point sets by Least-Squares-based process.

5.1. Least Squares Fitting Results. The PCA-based algorithm was tested on several noisy unorganised point sets, which include non-uniform, non-smooth, near self-intersecting, and self-intersecting ones. Figures 6(a) and 6(b) present the results obtained for two sets, each one having some of these features. Near self-intersecting, non-uniform point clouds, as the one shown in Figure 6(a), can be adequately reconstructed by varying the length of the segments of the reconstructed polyline, considering the dispersion of points contained in each ball. The radius optimization process, described in section 4.2.1, turns out to be useful for this purpose.

In Figure 6(b) a point set sampling a self-intersecting curve $C(u)$ is displayed. As mentioned in Section 4.3, a PCA algorithm alone is not robust enough for reconstructing self-intersecting point clouds. However, due to the randomness of the starting point of the reconstruction mentioned in Section 4.2.1, certain runs can result in adequately reconstructing the PL approximation of $C(u)$, while other runs will not. Because of this, the skeleton method for curve reconstruction was considered.

Notice that criteria for identifying the ends of *open* noisy point sets are needed in order to *correctly* reconstruct open curves. These criteria include the fact that when the PCA algorithm finds an end of the curve $C(u)$, the evolution to a next centre of the fitting ball $B(p, r)$ is possible only in one direction. This condition allows to discriminate samples of open vs. closed curves. In the example discussed, (Aphrodite data set), however, all the sampled curves are closed.

5.1.1. *Application to Surface Reconstruction from Range Images.* Range imaging is a technique for digitizing three-dimensional objects, given a set of range images. A range image is a function $I \times J \rightarrow R^3, \langle i, j \rangle \mapsto P_{ij}$, where $I \times J$ is the grid of pixels in the range image, and $P_{ij} = \langle x_{ij}, y_{ij}, z_{ij} \rangle$ is the point in the surface of the optically sampled object, captured by the pixel in position $\langle i, j \rangle$ of the grid of pixels.

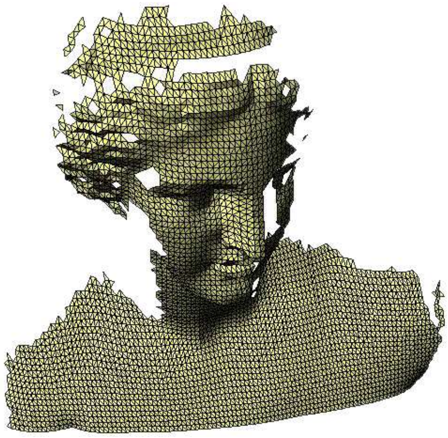
As no single range image suffices to describe the entire object, it is necessary to combine a collection of range images (see Figs. 7(a) and 7(b)) into a single triangular mesh that completely describes the object. The steps listed below were followed in order to generate such mesh from the individual pictures (considered already registered with respect to each other): (i) Construction of the individual mesh M_i for each individual range image R_i (Figs. 7(a) and 7(b)) ; (ii) slicing of the complete set of meshes $M_i, i = 1, 2, ...$ with a set of parallel, equi-spaced planes, thus building planar samples of points; (iii) reconstruction of a set of curves (contours) from the sampled points by using the algorithm discussed in Section 4.2 (see contours in Figure 1.8); and (iv) use of an algorithm for surface reconstruction from planar slices. In this case, the algorithm discussed in [27] was used. The reconstruction of Aphrodite's head is presented in order to illustrate the mesh integration process. The range images used were a courtesy of Fraunhofer Inst. for Computer Graphics, Darmstadt, Germany.

In step (ii), a set of parallel planes are defined, and the intersection between each plane and the collection of shells recovered from the range images is calculated. A set of planar samples of points $S_1, S_2, \ldots, S_k, \ldots$ is generated by sampling the polylines resulting from each intersection. Figure 3(a) shows one such coplanar sample $S_k = \{P_{0_k}, \ldots, P_{N_k}\}$ for Aphrodite's head model.

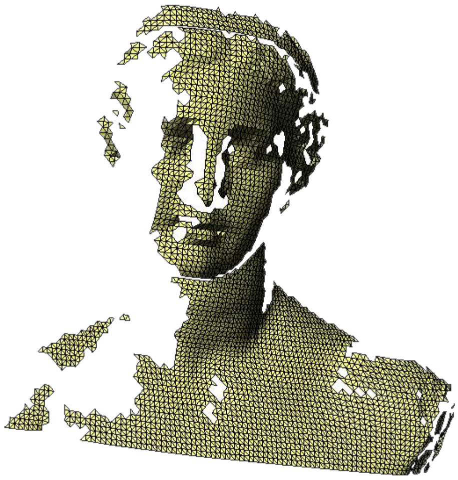
More than 100 levels (the number and separation dictated by the Nyquist criterion applied in the axial direction) of slicing were obtained from sampling the collection of meshes corresponding to Aphrodite's sculpture head and neck, and the same number of polylines were reconstructed from these sets (Figure 1.8). In spite of the large number of range images available for Aphrodite's sculpture, some of its regions were not covered by any of these, and therefore several sets of points needed to be manually completed. Once the sets were completed, none of the reconstructed polylines were edited. The surface reconstructed from the integrated, stochastically recovered contours is shown in Figures 9(a) to 9(c). Figures 9(a) and 9(b) correspond to resampling planes which are not orthogonal, and to an unfinished reconstruction (there is still a border). Figure 9(c) represents the integrated result for slicing planes parallel to plane XY. The final Aphrodite's surface reconstruction is shown in figure 9(d).

5.2. Medial-Axis, Delaunay Triangulation Results. Application of Medial Axis or Delaunay Triangulation methods is justified when the sampled curve $C(u)$ is self-intersecting. For this reason, these methods were not tested with the Aphrodite data set, but with planar self-intersecting Bezier curves sampled with stochastic noise. The discussion of such tests follows.

5.2.1. *Pre-processing to Transform into XY Plane.* As before, the point sample of $C(u)$ renders a quasi-planar point set. According to the discussion, an isotropic scaling was applied to the point set, because PCA is sensitive to dimensional issues. PCA was then applied to estimate the best plane Π fit to the point set, and a modified Householder transformation was used to project all points onto Π. In addition, a rigid transformation is used to bring the (now perfectly) planar point set to the XY plane, following the process described in section 4.1. Figure 10(a) shows the initial point set, along with a coordinate frame attached to the plane Π.



(a) $i - th$ Mesh from Front Range Image i Aphrodite.



(b) $k - th$ Mesh from Front Range Image k Aphrodite.

FIGURE 1.7. Range Image Data Set. Courtesy from Fraunhofer Inst. Computer Graphics, Darmstadt, Germany.

5.2.2. *Delaunay-based Medial Axis Processing.* The Delaunay Triangulation of the point set projected onto Π and then transformed to XY is illustrated in Figure 10(b). In the construction of the tape shaped polygon T, Delaunay Triangles included in PCA balls are accepted (Figure 10(c)). The triangles not entirely included in PCA balls may still be accepted based on the Edge Length or Area criteria (see Figure 10(d)). Notice that T is a connected 2-dimensional region with boundary $\partial T = L_0 \cup L_1 \cup ... \cup L_m$ in Figure 1.4. After the region T has been synthesized by consolidating Delaunay triangles chosen according to the above criteria the border ∂T must be determined. This step is a standard procedure in Boundary Representation construction and is conducted according to the rules in section 4.4.1. The next goal is to identify the Medial Axis (MA) of T. An exact calculation is out of question because MA produces curved portions. However, if a resample RT of T

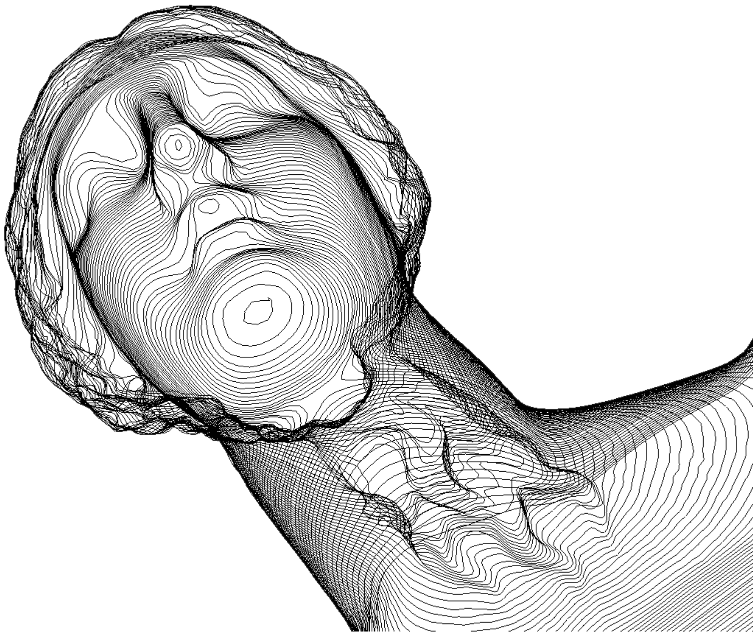


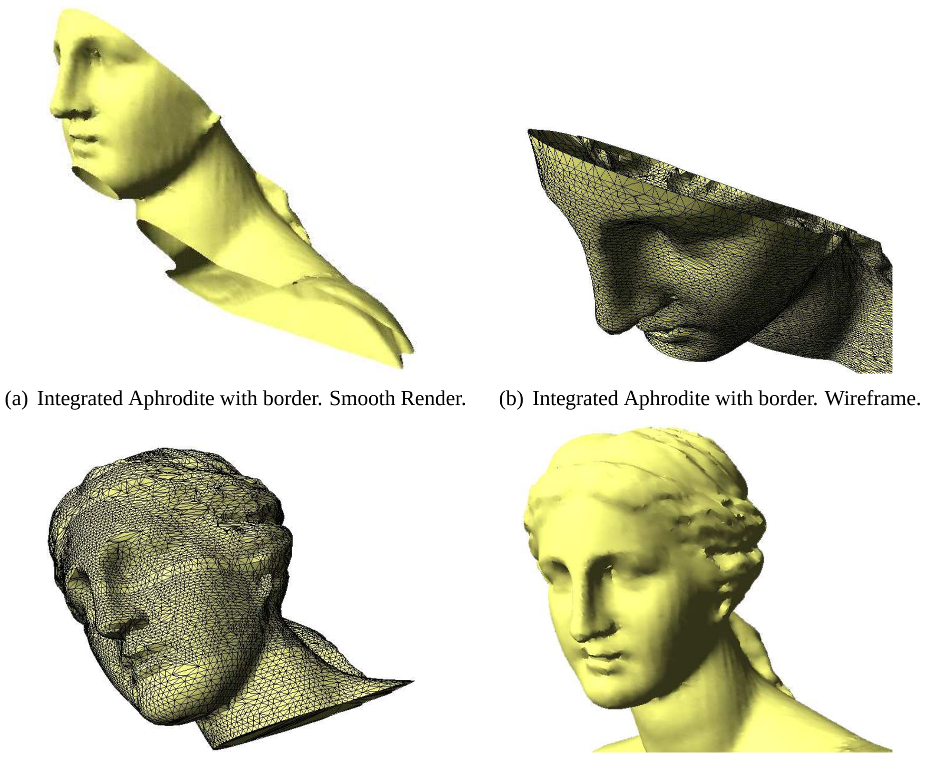
FIGURE 1.8. Aphrodite's head contours recovered from planar samples of points. Test data courtesy from Fraunhofer Inst. for Computer Graphics, Darmstadt, Germany.

is fine enough, its medial axis may be approximated as the sequence of Voronoi Edges of RT completely included in T. The *border* ∂T is resampled (see Figure 10(e)) and a new Delaunay Triangulation is calculated. The Delaunay Triangulation $DT(RT)$ of RT is purged to keep only those Delaunay Triangles internal to T. In this form, again, T is re-triangulated, but this time with triangles whose circumscribed centre lie inside T. The loci of such centres is $SK(T)$, the skeleton approximation for the medial axis $MA(T)$ of T (see Figure 10(f)). As can be seen in Figure 10(f), it is possible that the re-triangulation of T breaks this region into separate ones. This result is expected, since it indicates the presence of self-intersections in the original set. The algorithm corrects them by splitting the tape polygon T into annular sub-parts T_i. Care must still be exercised, as $SK(T)$ may be outside of a T_i region, as shown in Figure 10(f). This situation, however, is not harmful since the skeletons SK_i do not intersect each other, and therefore serve as PL approximations of the original $C_i(u)$ curves.

Figure 1.11 shows comparative results for a self-intersecting curve $C(u)$ (double "8") obtained using PCA (Figure 11(a)) and Voronoi-Delaunay (Figure 11(b) and Figure 11(c)) methods. Figure 11(a) shows that PCA alone processes the total point set but is not able to solve the self-intersection issue. The Voronoi-Delaunay result in Figure11(b) solves the self-intersection by generating several tangent closed curves. The Voronoi-Delaunay result in Figure 11(c) generates a PL approximation with wasp waist.

6. Complexity Analysis

For this complexity analysis, worst-case scenarios will be considered. In the case of the Delaunay Triangulation of N points in R^2 a complexity of $O(N^2)$ is counted, instead



(a) Integrated Aphrodite with border. Smooth Render. (b) Integrated Aphrodite with border. Wireframe.

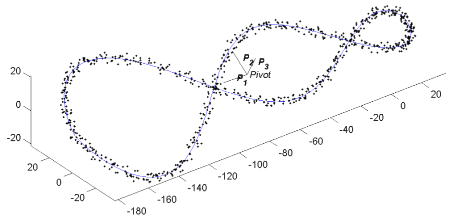
(c) Integrated Aphrodite without Border. Wireframe. (d) Integrated Aphrodite without Border. Smooth Render.

FIGURE 1.9. Results of Range Image Integration. Test data courtesy from Fraunhofer Inst. for Computer Graphics, Darmstadt, Germany.

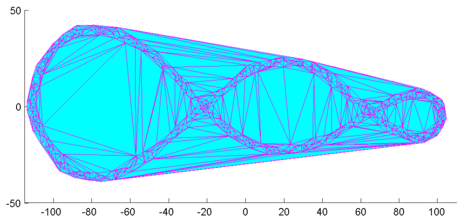
of $O(N)$ reported in [6], due to the fact that no special data structure is assumed. An sketch of the complexity analysis performed is presented in the following subsections. Since only well known facts on the complexity of the Delaunay Triangulations and Graph Theory are used, the reader is invited to consult the most basic literature on such topics.

Pre-processing. Point Sample Partition. Since in both cases (self-intersecting and non self-intersecting curves) the closure operation needs to be performed, such a part is omitted in the discussion. Instead, it is assumed that a pre-process to separate all possible curve samples in the initial set is performed. Therefore, the following discussion is per curve.

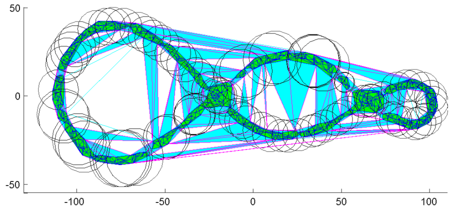
6.1. Alternative 1. Non Self-intersecting curve. PCA Analysis. The algorithm has a worst-case complexity of $O(N^2)$ in classifying N points in at most N balls. For each ball, the cost of PCA in a constant dimensional space (2D or 3D) is $O(N)$. Therefore, a worst-case cost of $O(N^3)$ is calculated. Figure 1.12 shows the execution times for the point set Aphrodite in a computer Pentium IV, Processor Clock at 3.2GHz with 2GB RAM. The curve presents an average complexity of $O(N^{1.55})$, which confirms that the expected value of complexity is much better than the worst case scenario discussed above.



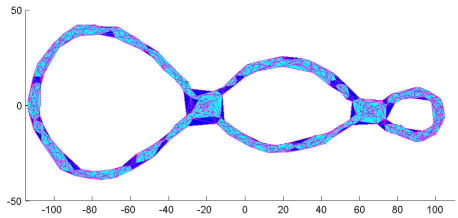
(a) Point Sample of Planar Double-8 $C(u)$ Manifold.



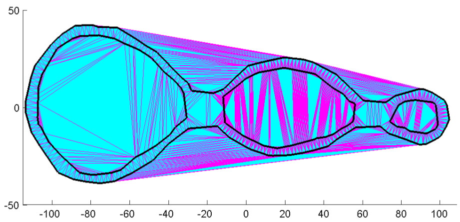
(b) Delaunay Triangulation of Planar Double-8 Point Sample.



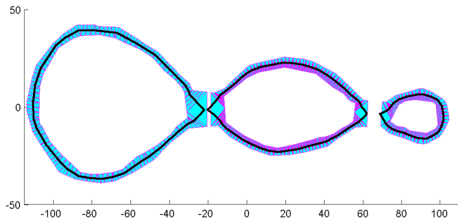
(c) Filtering of Delaunay Triangulation with PCA Balls.



(d) Selected Triangles by Area and Length Criteria.



(e) Tape Polygon and its Delaunay Triangulation.

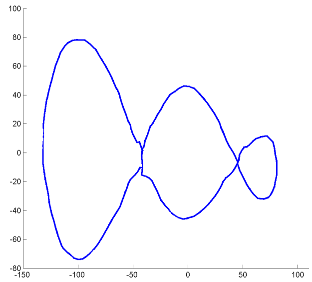


(f) Filtered DT and Skeleton.

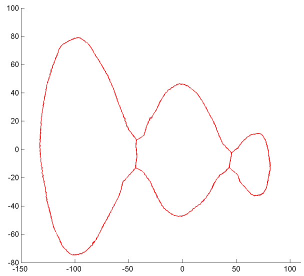
FIGURE 1.10. Process of P.L. Approximation of Double-8 self-intersecting $C(u)$ by Combined PCA and Voronoi-Delaunay Methods.

6.2. Alternative 2. Self-intersecting curve. Delaunay Triangulation and Medial Axis. The complexity analysis for the approximation of the skeleton of the tape polygon T follows.

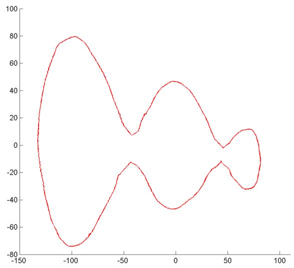
(1) Initial Delaunay Triangulation. First box in Figure 1.4. The number of triangles is $O(N)$. Cost: $O(N^2)$.
(2) First Purge Process (using only edge length and area criteria) in a set of N triangles. Part of second box in Figure 1.4. Cost: $O(N)$.
(3) Determination of $\partial T = L_0 \cup L_1 \cup .....$ from a set of N triangles. Part of second box in Figure 1.4. Cost: $O(N^2)$.
(4) Resampling of each edge of ∂T in k points. Part of third box in Figure 1.4. Cost: $O(k.N)$.
(5) Second Delaunay Triangulation for a set of $k.N$ points, giving $O(k.N)$ as the number of triangles. Part of the third box in Figure 1.4. Cost: $O(k^2.N^2)$.



(a) PCA-based Algorithm. Result.



(b) VD-based Algorithm. Multi-Polygon Result.



(c) VD-based Algorithm. Single-Polygon Result.

FIGURE 1.11. Final Results. PL Approximations of Double-8 self-intersecting $C(u)$ by PCA and Voronoi-Delaunay Methods.

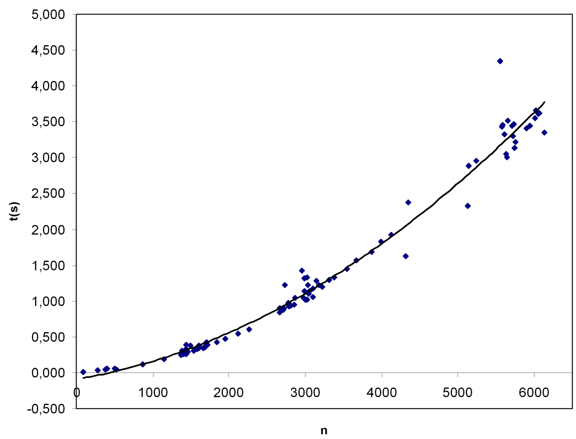


FIGURE 1.12. Execution Time of Principal Component Analysis (time vs. number of points).

(6) Second Purge Process, to see which ones of $k.N$ triangles fall inside T (T is already known from step (iii)). In the worst case, one has $O(k.N)$ as the number of vertices of the skeleton. Part of third box in Figure 1.4. Cost: $O(N^2)$.

(7) Construction of the Skeleton Graph with $O(k.N)$ vertices. The initial point sample for the self-intersection curve respects the Nyquist criterion (the level of stochastic noise is smaller than half of the minimal geometric detail to be sampled). Fourth box in Figure 1.4. Cost: $O(k^3.N^3)$.

In conclusion, the whole process costs $O(k^3.N^3)$ if the initial curve is self-intersecting, with the construction of the final graph being the most expensive part.

7. Conclusions and Future Work

Two methods have been presented for obtaining the PL approximation of a collection of planar regular curves $C(u)$ stochastically sampled. The Principal Component Analysis -PCA- method is useful for cases when the point set corresponds to a sample of non self-intersecting curves. This method returned correctly reconstructed PL 1-manifolds for non-trivial point sets (open, unorganised, noisy, non-uniform, non-smooth, near self-intersecting).

A new application of the PCA method for surface reconstruction from Range Imaging is also discussed, and results for a real model are presented. The integration method correctly merged together a set of meshes obtained from several individual range images, into a single mesh. This approach of merging individual meshes from range pictures overcomes some of the limitations present in common usage methods based on the direct meshing from the integrated point cloud from the range pictures. The direct methods do not render a manifold topology even when the model sampled is a manifold. Our method always renders a manifold provided that it works on a Nyquist sample.

The second method (Delaunay-based Medial Axis) can be used when self-intersecting curves have been sampled, and therefore when the PCA algorithm is not applicable. This new method synthesizes the $SK(T)$ skeleton of the tape-shaped 2D region covering the point set S. This skeleton is a 1-manifold for Nyquist samples of the curve. The existing literature has not considered the reconstruction from samples of self-intersecting (or non-manifold) curves.

Future Work. When the point sample of a self-intersecting curve has low quality, building a graph, which is the PL approximation of the curve, out of the medial axis of the tape polygon T covering the curve needs improvement. In this case the graph representing the principal shape presents “hair”, (i.e. high frequency artifacts), that need to be eliminated.

Bibliography

[1] Adamson A., and Alexa M., Anisotropic Point Set Surfaces. In *Afrigaph '06: Proceedings of the 4th international conference on Computer graphics, virtual reality, visualisation and interaction in Africa,* 2006, p. 7–13.

[2] Althaus, E., Mehlhorn, K., Näher, S. and Schirra, S., Experiments on Curve Reconstruction. In *ALENEX 2000*, p. 103–114, 2000.

[3] Althaus, E., and Mehlhorn, K., Polynomial time TSP-based curve reconstruction. In *Symposium on Discrete Algorithms (SODA)*, p. 686–695, 2000.

[4] Amenta, N., Bern, M. and Eppstein, D., The Crust and the Beta-Skeleton: Combinatorial Curve Reconstruction. *Graphical Models and Image Processing: GMIP,* 1998, **60**(2), 125–153.

[5] Boissonnat, J.D., Shape reconstruction from planar cross-sections. *Computer Vision, Graphics and Image Processing,* 1988, **44**, 1–29.

[6] Boissonnat, J.D., and Yvinec, M., *Algorithmic Geometry*, 1998 (UK: Cambridge University Press).

[7] Cheng, S.W., Funke, S., Golin, M., Kumar, P., Poon, S.H. and Ramos, E., Curve reconstruction from noisy samples. *Computational Geometry*, 2005, **31**, 63–100.

[8] Curless, B., and Levoy, M., A Volumetric Method for Building Complex Models from Range Images. *Computer Graphics,* 1996, **30**, 303–312.

[9] Dey, T.K., and Kumar, P., A simple provable algorithm for curve reconstruction. In *10th. Annual ACM-SIAM Symposium on Discrete Algorithms*, 1999.

[10] Edelsbrunner, H., and Mücke, E.P., Three-dimensional alpha shapes. *ACM Transaction on Graphics,* 1994, **13**, 43–72.

[11] Edelsbrunner, H., Shape reconstruction with the Delaunay complex. In *LATIN'98: Theoretical Informatics, volume 1380 of Lecture Notes in Computer Science,* p. 119–132, 1998.

[12] Fang, L., and Gossard, D.C., Fitting 3D curves to unorganized data points using deformable curves. In *Visual Computing, Proceedings of CG International*, p. 535–543, Berlin, 1992.

[13] Fomenko, A., and Kunii, T., *Topological Modeling for Visualization*, 1997 (Tokio: Springer Verlag).

[14] Fortune S., Voronoi Diagrams and Delaunay Triangulations. In *Du DZ, Hwang F, editors. Computing in Euclidean Geometry, Lecture Notes Series on Computing,* World Scientific, 1992.

[15] Geiger, B., Three dimensional modeling of human organs and its application to diagnosis and surgical planning. 1993, Research Report 2105, INRIA, Sophia-Antipolis, Valbonne, France.

[16] Guibas, L., and Stolfi, J., Primitives for the manipulation of general subdivisions and the computation of Voronoi diagrams. *ACM Transactions on Graphics*, 1985, **2**(4), 74–123.

[17] Hastie, T., and Stuetzle, W., Principal curves. *Journal of the American Statistical Association*, 1989, **84**, 502–516.

[18] Hoppe, H., DeRose, T., Duchamp, T., McDonald J. and Stuetzle, W., Surface Reconstruction from Unorganized points. In *ACM SIGGRAPH. 19th Annual Conference on Computer Graphics and Interactive Techniques*, p. 71–78, Chicago, 1992.

[19] Kegl, B., Principal Curves: Learning, Design, and Applications. PhD thesis, Concordia University, Montreal, Canada, 1999.

[20] Kegl, B., and Krzyzak, A., Piecewise Linear Skeletonization Using Principal Curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 2002, **24**(1), 59–74.

[21] Lee, I.K., Curve reconstruction from unorganized points. *Computer Aided Geometric Design,* 2000, **17**(2), 161–177.

[22] Mantyla, M., *An Introduction to Solid Modeling*, 1988 (Maryland: Computer Science Press).

[23] Morse, M., *The calculus of variations in the large*, 1934 (New York: American Mathematical Society).

[24] Ogniewicz, R., Skeleton-Space: a Multiscale Shape Description Combining Region and Boundary Information. In *IEEE Conference on Computer Vision and Pattern Recognition*, p. 746–751, Seattle, 1994.

[25] Ohtake, Y., Belyaev, A. and Seidel, H., Skeleton-Space: n Integrating Approach to Meshing Scattered Point Data. In *SPM '05: Proceedings of the 2005 ACM Symposium on Solid and Physical modeling,* p. 61–69, Cambridge, 2005.

[26] Raindrop Geomagic®Inc. *Studio 8*. http://www.geomagic.com, ©2006.

[27] Ruiz, O.E., Cadavid, C.A., Granados, M.A., Peña, S. and Vásquez, E. 2D shape similarity as a complement for Voronoi–Delone methods in shape reconstruction. *Elsevier J. on Computers and Graphics,* 2005, **29**(1), 81–94.

[28] Soucy, M., and Laurendeau, D., A General Surface Approach to the Integration of a Set of Range Views. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 1995, **17**(4), 344–358.

[29] Taubin, G., and Ronfard, R., Implicit simplicial models for adaptive curve reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 1996, **18**(3), 321–325.

[30] Turk, G., and Levoy, M., Zippered Polygon Meshes from Range Images. In *SIGGRAPH'94: Computer Graphics Proceedings, Annual Conference Series*, p. 311–318, Orlando, Florida, 24–29 July, 1994.

[31] Varady, T., Martin, R. and Cox, J., Reverse Engineering of Geometric Models. An introduction. *Computer Aided Design*, 1997, **29**(4), 255–268.

[32] Verbeek, J., Vlassis, N. and Kröse, B., A Soft k-Segments Algorithm for Principal Curves. In *Dorffner G, Bischof H, Hornik K, editors. ICANN 2001, LNCS 2130,* 2001, p. 450–456.

[33] Wang, W., Pottmann, H. and Liu, Y., Fitting B-spline curves to point clouds by curvature-based squared distance minimization. *ACM Transactions on Graphics,* 2006, **25**(2), 214–238.

[34] Zhou, H., Liu, Y. and Li, L., Incremental Mesh-based Integration of Registered Range Images: Robust to Registration Error and Scanning Noise. In *Seventh Asian Conference on Computer Vision, Hyderabad, India*, 2006, p. 958–968.

CHAPTER 2

Statistical Assessment of Global and Local Cylinder Wear

CONTEXT: A project to devise a method for the evaluation of wear regions in cylindrical surfaces was developed at the CAD CAM CAE Laboratory at EAFIT University. The results of this method sought to provide sufficient statistical information about the deformation of cylinders used in the polymer processing industry, in order to help production managers to make accurate decisions on the replacement of worn cylinders. The project was financed by EAFIT University, and evaluated by the Institute for Training and Research on Plastic and Rubber, (Instituto de Capacitación e Investigación del Plástico y el Caucho - ICIPC, Medellín, COLOMBIA).

Carlos Vanegas, research assistant under my direction in the CAD CAM CAE Laboratory, and myself devised and implemented this method in the period July-December 2006. The software produced (CylWear) is currently under copyright registration process. Theoretical and implementation contributions of this work appear in the paper:

- Ruiz O, Vanegas C, "Statistical Assessment of Global and Local Cylinder Wear", Accepted for presentation at the IEEE 5th International Conference on Industrial Informatics (INDIN07), to be held in Vienna, Austria, July 23-27, 2007.

As co-author of such a publication, I give my permission for this material to appear in this document. I am ready to provide any additional information on the subject, as needed.

Prof. Oscar E. Ruiz
oruiz@eafit.edu.co
Coordinator CAD CAM CAE Laboratory
EAFIT University, Medellin, COLOMBIA

ABSTRACT. Assessment of cylindricity has been traditionally performed on the basis of cylindrical crowns containing a set of points that are supposed to belong to a controlled cylinder. As such, all sampled points must lie within a crown. In contrast, the present paper analyzes the cylindricity for wear applications, in which a statistical trend is assessed, rather than to assure that all points fall within a given tolerance. Principal Component Analysis is used to identify the central axis of the sampled cylinder, allowing to find the actual (expected value of the) radius and axis of the cylinder. Application of k-cluster and transitive closure algorithms allow to identify particular areas of the cylinder which are specially deformed. For both, the local areas and the global cylinder, a quantile analysis allows to numerically grade the degree of deformation of the cylinder. The algorithms implemented are part of the CYLWEAR© system and used to assess local and global wear cylinders.

1. Introduction

Regarding extrusion or injection cylinders there is an economic interest in quantifying the degree of deformation away from a mathematical cylinder. The software processing a point sample of the interior of a cylinder is expected to fulfill the following criteria: (i) independence of the coordinate frame of the measurement, (ii) identification of the axis of the cylinder, (iii) identification and quantification of local, high wear areas, (iv) automated quantification of global wear.

The present article discusses a software that takes as input a point cloud evenly sampled on the interior wall of a cylinder and that is contained between two planes, approximately perpendicular to the cylinder axis. The point sample is assumed to be evenly spread in such an area, in such a manner that no part is over-sampled or under-sampled. No order is assumed in the point cloud.

2. Literature Review

An important application in metrology is the evaluation of cylindricity, since a large fraction of mechanical parts are cylinders. The evaluation of cylindricity is not simple, because it requires a number of circularity traces to be taken at different horizontal sections of the cylinder and must be combined with the straightness of the generators of the cylinder [1].

In the evaluation of cylindricity the zone cylinder has become a standard for the quality control community. The zone cylinder is the cylindrical crown contained between two co-axial cylinders with minimum radial separation (width) and containing all the data points. Determining the zone cylinder involves the calculation of the direction axis, and internal and external radius.

Sampling nominally cylindrical objects usually involves an apparatus consisting of a turntable, a probe, and the support of the probe. This measurement system involves three different axes: the axis of rotation of the table, the axis of the cylindrical object and the axis of the probe support [2]. In practice, these axes are not parallel, and accurate information of the orientation of the cylinder is not available. Therefore, the direction axis must be calculated.

A comparison of different methods for cylindricity evaluation is presented in [1]. An approach using normal least squares was introduced, which minimizes the squares of the perpendicular distances from the measured points to the axis of the cylinder. The author also presents a method based on the development of the surface of the cylinder, in which the surface is "flattened" using as reference the axis of the probe support. The flatness of the surface is then obtained from the mean plane equation.

Reference [2] presents a linear programming-based approach to estimate the minimum zone cylinder enclosing a set of points. The linear programming problem is iteratively solved in a 6-dimensional space generated by 6 parameters that define a hyperboloid associated to the cylinder. The above-mentioned approach is therefore relevant to the evaluation of overall cylinder deformation, but does not aim to solve the local wearing detection problem. The efficiency and accuracy of this method was improved through a procedure in which points that cannot provably define the solution are culled from the input point set [3].

The problem of finding the minimum width cylinder containing a set of points is an extension to three dimensions of the problem of finding the annulus of smallest width containing a set of points in the plane. Several works have addressed the minimum width annulus problem. Reference [4] proposes a fast algorithm that exploits the properties of convex-hull and Voronoi diagrams. Reference [5] proposes a generalized method for the minimum width annulus in a d-dimensional space. Reference [6] addresses this problem in 2-dimensions (disks) and 3-dimensions (balls). Their method for testing disk roundness (mentioned below) is extended to the evaluation of balls by partitioning them into several slices, each of which is evaluated as a disk.

Reference [7] also studies the problem of determining whether a manufactured cylinder is sufficiently round. They first introduce a procedure for testing roundness in disks, in which set of probes are iteratively taken at uniform intervals directed at the origin, using the finger probing model of [8]. The procedure stops when a decision is made on whether the sample points can be covered by some “thin” annulus. The roundness testing procedure is extended to cylinders by projecting the points on the surface of the cylinder onto the XY plane, and applying the “thin” annulus criterion to the projected points. Notice that this method assumes that the sampled cylinder is resting on the XY plane and that its orientation is known. As noted above, such an assumption is not always valid since the axes of the measurement system (e.g. the axis of rotation of the cylinder) are unknown in practice.

The problem of cylinder fitting is also addressed by [9, 10], as a part of their method for detecting bore holes for Industrial Automation. They propose a sequential cylinder parameter fitting in which the orientation of the axis is first calculated, followed by the calculation of the radius and the position of the axis. A previous step in this bore detection method consists in estimating the normal vector to the surface at each sampled point.

The first sub-quadratic solution to the minimum width cylindrical shell problem, based on a linearization of such problem, is presented in [11]. Again, the problem addressed is that of estimating the global deviation of a point set from a cylindrical shape, and does not cover our aim of statistical assessment.

General comments to the reviewed literature are: (i) a dimensional quality control problem is attacked, which poses the question of whether a workpiece must be rejected or not, (ii) the determination of minimal enclosing and maximal enclosed cylinders, minimum zone cylinder, etc. are sought, usually in cylinders which are placed in a particular position of the space, (iii) in the item (i), data are interpreted in literal way, i.e. ignoring the trends or statistical indicators of their quality. The approach undertaken in our work is to produce a statistical diagnose of the cylindricity (see section 3.4), and therefore each data is taken as inherently biased by several sources of noise. One can do so, since our work is aimed to help the production manager to numerically evaluate the need to replace the cylinder when (from his/her point of view) the wear or distortion in the cylinder reaches unacceptable values (see section 3.7). As a result, we do not use the typical deterministic geometry algorithms of the literature. Instead, we apply stochastic geometry to diagnose the data.

3. Methodology and Results

The diagnosis of cylinder wear is basically a treatment of an unordered point set, collected on the internal wall of the sampled cylinder. The point set is measured in the particular unknown (local) coordinate system of the reading instrument, different from the World Coordinate System -WCS- of the shop floor. Although the points are collected on a definite geometrical shape (a cylinder), the numerical values output by the measuring device contain several stochastic components.

The processing of the point cloud (see Figure 2.1) is as follows:

(1) Assuming for the cylinder a Length/Diameter ratio larger than 5.0, a Principal Component Analysis -PCA- is run. The PCA allows to identify the direction of largest dispersion in the data, which is the direction of the cylinder axis. As a by-product, the center and mean radius of the cylinder are also identified.
(2) A rigid geometric transformation is applied to the point cloud to align the cylinder with the Z axis of the WCS.
(3) A Quantile Analysis is performed, which renders the histogram of frequencies of radius deviations for the global point cloud.
(4) The cylindrical data is developed (unwrapped or flattened) onto $\mathbb{R}^2$ (XY plane) to perform a local analysis.
(5) A low pass filter is applied to the data, which eliminates the high frequencies of the point cloud.
(6) A surface is reconstructed for the point data, using a lift of a Delaunay Triangulation, to facilitate the visual identification of the high wear areas. At this time, the data resembles a rectangular mountain region, whose heights correspond to the areas (regions of (θ, h) values) with larger cylinder wear.
(7) Two alternative algorithms are applied to automatically identify such high wear areas: k-cluster and Extended Neighborhood Analyses.
(8) Quantile and Mean-Median Analyses are performed on the local wear regions.
(9) All the results are given in the form of text files (for documentation and analysis) and via graphic output (for the easy identification by the user).

3.1. Measured Data. Three sources of deviation of point data away of a perfect cylinder are assumed: (i) a general wear, (ii) localized wear spots, and (iii) measurement noise introduced by the scanner. The point set has an arbitrary orientation and position, and it is necessary to determine the coordinate system in which it was collected by the measuring devise. The nominal radius and length of the cylinder are assumed to be known.

3.2. Transformation of Measured Data to the World Coordinate System. The purpose of this section is to rigidly transform measured data so that the calculated axis of the cylinder is coincident with the Z axis of the WCS and its center is coincident with the origin O of the WCS. However, we know neither the axis of the cylinder, nor its effective radius and length. To determine such values is the purpose of the following section.

3.2.1. *Principal Component Analysis.* Let $P' = \left\{(x', y', z') \in \mathbb{R}^3\right\}$ be the set of points sampled on the surface of a cylinder $C\,(R, H, A, O)$, where R, H, A, O are the nominal radius, nominal length, axis and center of gravity of the cylinder, respectively. It must be noticed that only R and H are known. The actual values of radius, height, axis,

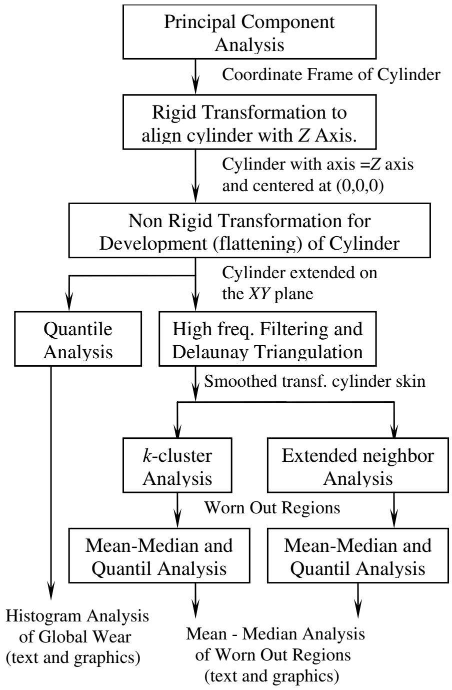


FIGURE 2.1. Cylindricity Diagnose with Point Cloud

and center must be determined from P'. By applying a Principal Component Analysis -PCA- the trends in the collected data will be identified (see [12], [13]).

Let Σ be the (3×3) covariance matrix of the process

$$P' = \{(x_1, y_1, z_1), \ldots, (x_n, y_n, z_n)\},$$

with c_{ij} being the cross covariance of components i and j of the point set. Σ is semi-positive definite, since it is symmetric with non-negative main diagonal. The eigenvalues of Σ are non-negative real numbers λ_i. Then, Σ satisfies the equation $\Sigma.V = V.\Lambda$ with V a matrix whose columns are the (orthogonal) eigenvectors of Σ, and Λ is a diagonal matrix containing the eigenvalues of Σ. Without sacrificing generality one may sort the eigenvalues in decreasing order, say $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq 0$, and write the eigenpairs of the covariance matrix as:

$$\Sigma.V = \Sigma.\begin{bmatrix} v_1 & v_2 & v_3 \end{bmatrix} = \begin{bmatrix} v_1 & v_2 & v_3 \end{bmatrix}.\begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix}$$

with λ_i being the variance of the data in the direction v_i. It follows that v_1 is the direction of the data P' in which maximal variance appears, v_2 is the direction in which the next decreasing variance appears, and v_3 is the direction with lowest data variance in P'. For a Length/Diameter ratio larger than 5.0, it can be seen that $v_1 \equiv A$, i.e. the axis of the cylinder A is the eigenvector associated with the largest eigenvalue or variance, λ_1 (the direction with highest variance of the data P'). Therefore $\Sigma.A = \lambda_1 A$. The triad v_1, v_2, v_3 is orthogonal, and we may enforce the condition $v_1 \times v_2 = v_3$, forming a right handed canonical coordinate system. Notice that given the cylindrical symmetry of the data, the second and third variances are almost the same. Except for numeric stochastic errors: $\lambda_2 \approx \lambda_3$.

3.2.2. *Transformation to a standardized coordinate system.* Once we know the axis $A = (A_x, A_y, A_z)$ and the center of mass $O = (O_x, O_y, O_z)$ of the measured cylinder, we must find out a 4×4 rigid transformation

$$M = \begin{bmatrix} & R^*_{3\times 3} & & T^*_{3\times 1} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

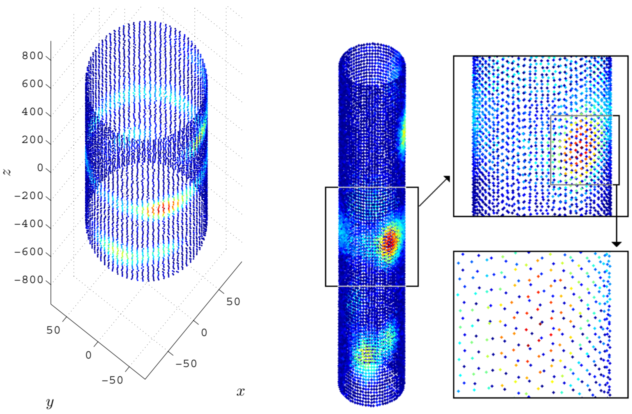
to move the point data in such a way that the axis of the cylinder is coincident with the Z axis of the WCS, and its center of mass is coincident with the origin of the WCS.

$$\begin{bmatrix} & R^* & & T^* \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} v_2 & v_3 & A & O \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \tag{5}$$

Once R^* and T^* have been determined from (5), each point sampled can be transformed with (6), so the data set looks like in Figures 2(a) and 2(b).

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} & R* & & T* \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} \tag{6}$$

3.3. Mapping of Normalized Cylinder 3D data onto 2D. After a normalization has been performed on the measured data, the axis of the cylinder coincides with the Z axis, and its center of mass with the origin O. The next step is to "unwrap" the point cylinder, and to extend the point set on the XY plane. The function used to do so is not an isometry, since the cylinder data is shrunk in order to fit into a rectangular basis of size 1.0×1.0.



(a) Transformation of general measured point set to bring cylinder axis to Z axis

(b) Detail of the point set with local damage on cylinder surface

FIGURE 2.2. Sampling of a cylinder surface with local damage

The point set $P = \{(x, y, z)\}$ (which is the cylinder point sample with its axis aligned with the Z axis of the World Coordinate System) is transformed into a new set

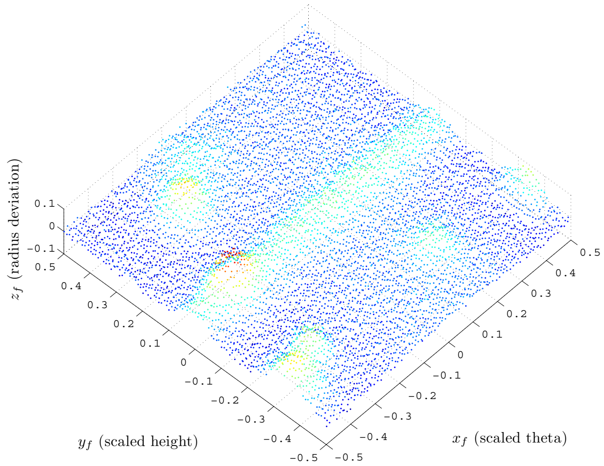


FIGURE 2.3. Point set with noise, placed in general position in space. Measured data in an experiment.

$Q = \{(x_f, y_f, z_f)\}$, with the following characteristics: (i) the z_f coordinate of each point in Q is the deviation, for the corresponding point in P, away from the calculated radius of the cylinder, (ii) the point set Q is organized as a function $g : \mathbb{R}^2 \to \mathbb{R}$, with $z_f = g(x_f, y_f)$, (iii) the (x, y) pairs are included in a rectangular domain in $\mathbb{R}^2$. This means, the cylinder has been unwrapped and extended on the XY plane (Figure 2.3). The unwrapping transformation for set P into Q is described in equation (7). It maps each point (x, y, z) sampled on the surface of a cylinder into (x_f, y_f, z_f) with $(x_f, y_f) \in [-0.5, 0.5] \times [-0.5, 0.5]$.

(7)
$$\begin{aligned} x_f &= \frac{\arctan(x, y)}{2\pi} - \frac{1}{2} \\ y_f &= \frac{1}{2}\left(\frac{z}{\max(|\max_i z_i|, |\min_i z_i|)}\right) \\ z_f &= \|(x, y)\| - R \end{aligned}$$

The reader may notice that in Figure 2.3 the intensity is not uniform. This is due to the fact that a color coding is given to the z_f coordinate. Consequently, regions with larger deviation from the nominal radius (regions with higher wear) look lighter in the image.

3.4. Statistical Analysis. The points in Figure 2.3 have a z_f coordinate that represents the deviation with respect to the nominal cylinder radius. This deviation is due to three causes: (i) a general wear of the cylinder, (ii) localized wear in specific regions of the cylinder, and (iii) a stochastic noise resulting from the measurement process. The purpose of this step is to measure the deviation of the data that is explained by each factor, i.e. how much in the collected data are these components present. Figure 2.4 shows the histogram of frequencies with respect to z_f. The horizontal axis is divided into intervals of the z_f variable. The vertical values correspond to the number of points whose radial deviation z_f falls within each interval. In this histogram we can see in the range $[-0.02, 0.02]$ an approximately normal distribution with mean $\mu = 0$. This distribution corresponds to the sampling error of the instrument (factor (iii) above). Above a deviation of 0.02 away from the nominal radius we find the effects (i) and (ii) mentioned before. Thus, in the interval

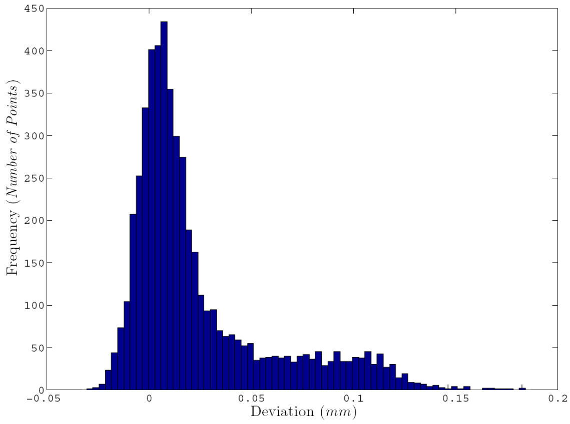


FIGURE 2.4. Histogram of the cylinder radius deviation.

$[0.02, 0.07]$ one will find the cutting deviation to classify localized wear or damage in the cylinder.

By using the frequency histogram of Figure 2.4, one is able to separate the set of points Q into points showing only overall wear vs. points showing overall and localized wear. In the histogram, the cutting value is $\varepsilon = 0.02$. This means, points whose radial deviation is below 0.02mm are considered to have overall wear. Points with radial deviation above 0.02mm are considered to present overall and localized wear. These points constitute the set Q_ε.

3.5. Cluster Analysis. The purpose of the cluster analysis applied to a set of n points in $\mathbb{R}^m$ is to identify k groups (k being set by the user) in the n points, such that in each group the points are close to each other, and at the same time far away from the points in other groups. In this manner, in the initial population, k clusters of neighboring points are identified. Let the population be formed in this application by n points in the set Q_ε. The space of the points is $\mathbb{R}^m = \mathbb{R}^3$. Let each point q_i in Q_ε be noted as:

$$(x_{fi}, y_{fi}, z_{fi}) = (X_{i,1}, X_{i,2}, X_{i,3}) \in Q_\varepsilon$$

The mean of the j-th variable $(j = 1, 2, 3)$ in the l-th group is noted by $\bar{X}_{(l),j}$, for $l = 1, \ldots, k$. The distance of the point q_i to the l-th cluster is:

$$D_{(i,l)} = \left[\sum_{j=1}^{3} (X_{i,j} - \bar{X}_{(l)j})^2 \right]^{1/2}$$

The error of the partition is given by the summation of the distance of each point to the cluster under which it is classified. The error of a partition $P(n, k)$ of the n points in k clusters is noted by:

$$\varepsilon(P(n,k)) = \sum_{i=1}^{n} [D_{(i,l(i))}]^2$$

where $l(i)$ is the set under which the i-th point is classified, which is the one for which the distance $D(i, l)$ is a minimum. It must be noticed that for each partition of the set Q_ε there will exist a value $\varepsilon(P(n, k))$. The partition that makes $\varepsilon(P(n, k))$ a minimum is our k-mean partition.

The method of the k-means is summarized as follows:

(1) Propose k initial points $\bar{X}_{(l)}$.
(2) For each point q_i find out its corresponding cluster $l(i)$ (for which the summation of the $D(i,l)$ is a minimum).
(3) Recalculate $\bar{X}_{(l)}$ as the centroid of the q_i points belonging to the cluster $l(i)$.
(4) Repeat the steps 2 and 3 until $l(i)$ remains constant for every i between successive iterations. At this point, $\varepsilon(P(n,k))$ reaches a minimum.

In this manner the points migrate from one cluster to another, until the reduction of $\varepsilon(P(n,k))$ is zero. After the $l(i)$ are found with the previous algorithm, and as a visualization aid, the convex hull of each $l(i)$ may be found and drawn. In the particular case of the wear of the cylinders, such a visual post-processing helps in displaying the zones of the cylinder whose wear is higher. The main inconvenience of the k-means method is the need of pre-establishing k, the number of clusters. For this reason an alternative method is introduced next.

3.6. Partition Analysis. The set Q_ε in (8) represents all the point data whose distance to the axis of the cylinder is higher than the threshold. Notice that the proposed algorithm seeks to eliminate the user interaction and to identify and bound the different deformation regions. Therefore, Q_ε must be partitioned into the local zones that present a particular wear of the cylinder. For such a purpose we define an equivalence relation R on Q_ε and then we calculate a partition Π of Q_ε by R. Let R be the equivalence relation defined as:

$$R(a,b) \Leftrightarrow \exists q_1, q_2, ..., q_w((q_i \in Q_\varepsilon, i = 1, ..., w)\wedge \\ (a = q_1) \wedge (b = q_w) \wedge (\|q_i - q_{i+1}\| < \delta)) \tag{8}$$

This equivalence relation basically states that points a and b belonging to Q_ε are equivalent if and only if there exists a path of points starting at a and ending at b such that two points q_i and q_{i+1} of the path are not separated from each other by more than a distance δ. In order to partition Q_ε in a partition of all points that are equivalent to each other, we apply algorithm 1.

Figure 2.5 shows the results of the partition algorithm applied on $Q = Q_{0.02}$. The three resulting data sets are automatically classified by the algorithm, projected on the XY plane, and the convex hull of the projection calculated and displayed on such a plane in Figure 2.5.

Figures 2.6 and 2.7 present the different noise factors in the flattened data set. Figure 2.6 shows the unfiltered data set in the scaled dimensions of the cylinder, while figure 2.7 shows the filtered data set mapped back to the physical dimensions of the cylinder. The localized damage in this data set has the shape of a mountain ridge (typical of a case in which a foreign object slides inside the cylinder) accompanied by isolated peaks. The highest deformation is present in a region centered in point $h = 500$mm and $\theta = 60^{\circ}$. Also, the wear located at $[0^{\circ}, 100]$ is the same as the one located at $[360^{\circ}, 100]$, since $0^{\circ} = 360^{\circ}$ because the cylinder wraps itself.

3.7. Diagnose Output. Three different outputs are produced from the process previously discussed: (i) graphical; (ii) histograms of frequency of radial deformation; and (iii) output file. They are discussed next.

Algorithm 1 Partitioning Algorithm to calculate neighborhoods of cylinder deformation

$\Pi = []$
while Q_ε **do**
  $p = \text{first}(Q)$
  queue_to_expand $= \{p\}$
  $Q_\varepsilon = Q_\varepsilon - \{p\}$
  partition $= \{\}$
  while *queue_to_expand* **do**
    element_to_expand $=$ first(*queue_to_expand*)
    partition $=$ *partition* $\cup$ $\{$*element_to_expand*$\}$
    queue_to_expand $=$ *queue_to_expand* $-$ $\{$*element_to_expand*$\}$
    for a such that $R($*element_to_expand*$, a)$ **do**
      $Q_\varepsilon = Q_\varepsilon - \{a\}$
      queue_to_expand $=$ *queue_to_expand* $\cup$ $\{a\}$
    end for
  end while
  $\Pi = [\Pi,$ *partition*$]$
end while

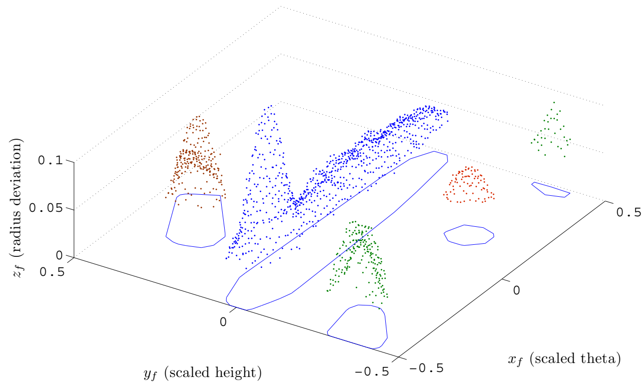


FIGURE 2.5. Result of localized deformations found with partition analysis.

3.7.1. *Graphical Output.* The radial deformation is converted to a function $f : \Theta \times H \to \Delta R$ (the deviation of the radius form its nominal value, see Figure 2.6). Delaunay triangulations and filtering are applied to display such a surface, as well as the regions of $f : \Theta \times H$ which represent a higher ΔR. Colors green and blue mean lower deformation, while colors yellow and red indicate higher deformation.

3.7.2. *Histogram of Frequencies of Radial Deformation.* A histogram results from plotting the number of samples n_i measured which fall into each range of radius deviation ($\Delta\,(R_i)$) (Figure 2.4). Two clearly differentiated regions appear: (i) A normal distribution of measurement errors, centered in 0, containing negative values of ΔR. Values of ΔR between $[-\sigma, +\sigma]$ correspond to the measurement errors; and (ii) Values of ΔR above σ representing the deterministic trend of the data, which corresponds to the wear.

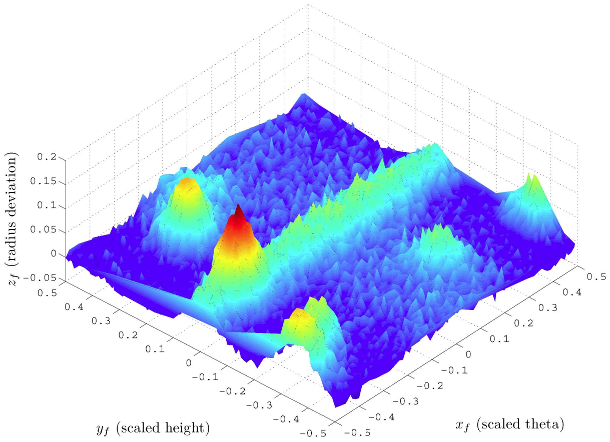


FIGURE 2.6. Surface of Radius deformation with physical dimensions (workpiece coordinate system)

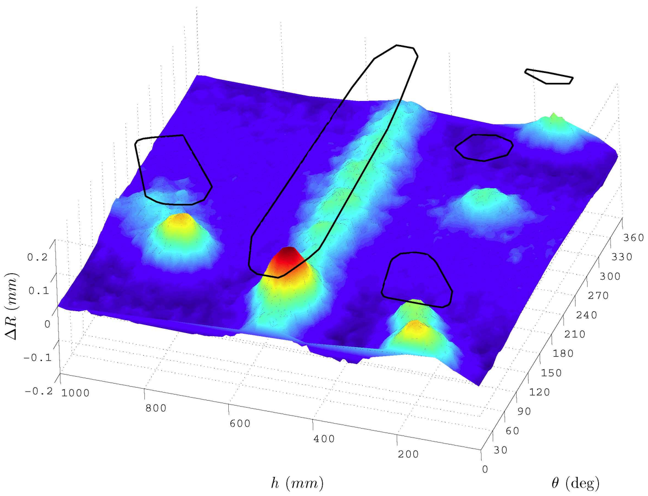


FIGURE 2.7. Filtered, automatically-detected localized Wear Regions (using Partition Analysis)

3.7.3. *Output File.* The output file contains two basic components: (i) the quantile information for the *global deformation* of the cylinder radius (ΔR deviation from the nominal radius); and (ii) the statistical information for each one of the *local areas of higher wear*. Global information corresponds to a text version of the histogram information discussed above. Local deformation includes for each area of large deformation the mean, median, standard deviation, maximal deviation and position of the wear area (θ, h).

4. Conclusions and Further Work

This article has presented a software tool to diagnose the general and local wear of a cylinder. No assumption is made on the orientation or position of the cylinder in the space, or on the coordinate frame of the measuring devise. The software implemented is successful in identifying the position in space of the cylinder (in this case, five degrees of freedom). These algorithms filter out high frequencies in the data, fit a surface to the resulting point cloud, and identify by two alternative methods the regions of largest local wear. Several statistical reports (quantile and frequency histogram) are produced, which also diagnose the cylinder in local spots as well as globally.

Future efforts include:

(1) Bringing the devised tools to the domain of dimensional quality control.
(2) Approaching the problem as a non linear minimization or optimization one.
(3) Using the findings in the previous item to diagnose other geometries different from the cylindrical one (torus, spheres, partial cylinders, cones, etc.).

Bibliography

[1] T. S. R. Murthy, “Comparison of different algorithms for cylindricity evaluation,” Int. J. Mach. Tool Design & Res., vol 22, no. 4, pp. 283-292, 1982.

[2] O. Devillers, and F. P. Preparata, “Evaluating the cylindricity of a nominally cylindrical point set,” in Proc. 11th Annual ACM-SIAM Symposium on Discrete Algorithms, Philadelphia, PA, 2000, pp. 518-527.

[3] O. Devillers, and F. P. Preparata, “Culling a set of points for roundness or cylindricity evaluations,” Int. J. Comput. Geom. Appl, vol. 13, no. 3, pp. 231-240, 2003.

[4] U. Roy, and X. Zhang, “Establishment of a pair of concentric circles with the minimum radial separation for assessing rounding error”, Computer Aided Design, vol. 24, no. 3, pp. 161-168, 1992.

[5] J. Garcia, and P. A. Ramos, “Fitting a set of points by a circle,” in Proc. ACM Conference on Computational Geometry, 1997, pp. 139-146.

[6] P. Bose, and P. Morin, “Testing the quality of manufactured disks and balls,” Algorithmica, vol. 38, no. 1, pp. 161-177, Oct. 2003.

[7] P. Bose, and P. Morin, “Testing the quality of manufactured disks and cylinders,” in Proc. 9th international Symposium on Algorithms and Computation, Lecture Notes in Computer Science, vol. 1533, K. Chwa and O. H. Ibarra, Eds. London: Springer-Verlag, 1998, pp. 129-138.

[8] R. Cole, and C. K. Yap, “Shape from probing,” Journal of Algorithms, vol. 8, no. 1, pp. 19-38, Mar. 1987.

[9] G. Biegelbauer, and M. Vincze, “Fast and robust bore detection in range image data for industrial automation,” in Proc. 2nd international Symposium on 3D Data Processing, Visualization, and Transmission, Washington, DC, 2004, pp. 526-533.

[10] G. Biegelbauer, and M. Vincze, “3D vision-guided bore inspection system,” in Proc. 4th IEEE international Conference on Computer Vision Systems, Washington, DC, 2006.

[11] S. Har-Peled, and K. R. Varadarajan, “Approximate shape fitting via linearization,” in Proc. 42nd Annual IEEE Symposium Found. Comp. Sci., 2001, pp. 66-73.

[12] O. E. Ruiz, C. A. Cadavid, M. J. Garcia, and R. Martinod, “Principal component analysis -PCA- and Delone triangulations for PL approximation C1-continuous 1-manifolds in Rn,” in Proc. Computer Graphics and Imaging (CGIM), Kauai, HI, August 17-19, 2004.

[13] O. E. Ruiz, and C. A. Vanegas, “Piecewise linear curve reconstruction from point clouds,” in Proc. 6th International Symposium series on Tools and Methods of Competitive Engineering, Ljubljana, Slovenia, April 18-22, 2006, pp 285-298.

CHAPTER 3

Bi-curve and Multi-patch Smoothing with Application to the Shipyard Industry

CONTEXT: The CAD CAM CAE Laboratory at EAFIT University has kept throughout the years cooperation agreements with research universities and institutions in Europe including: Max-Planck-Institut fur Informatik at Universitt des Saarlandes, Saarbrcken, Germany, Fraunhofer Inst. Graphische Datenverarbeitung, Darmstadt, Germany, and the University of Vigo, Vigo, Spain. As a part of such agreements, students hold visiting research assistant positions at the hosting institution for periods ranging from 6 to 12 months.

Carlos Vanegas has been twice invited by Prof. Xoan A. Leiceaga Baltar, director of the Group of Graphic Engineering and Design (Grupo de Enxeera Grfica e Deseo) at the University of Vigo, Vigo, Spain, to join his group as visiting research assistant. During his first internship (January-July 2005) Carlos Vanegas participated in the development of a CAD system for the design and manufacturing of ship hulls for the shipyard industry of Galicia (Spain) and North of Portugal. The project was financed with INTERREG III-A funding of the European Union. During his second internship, Carlos Vanegas participated in the development of a crane simulator, and was responsible for the implementation of the dynamic model, part of the spherical visualization system, and part of the interface between the dynamic model and the user control module.

The method presented in this chapter was devised and implemented by Carlos Vanegas with valuable suggestions from Engineers Manuel Rodriguez and Jose Prieto and constitutes one of the tools included in the system for the naval industry. Contributions of this work appear in the paper:

- X. Leiceaga, O. Ruiz, C. Vanegas, M. Rodríguez, J. Prieto, E. Soto, "Bi-Curve And Multi-Patch Smoothing with Application to the Shipyard Industry", accepted for presentation in the ADM-INGEGRAF conference, to be held in Perugia, Italy, June 6-9, 2007.

The results of the crane simulator project, developed during his second internship, are subject to confidentiality, and no publication on such results is possible at this time.

As co-authors of such a publication, we give our permission for this material to appear in this document. We are ready to provide any additional information on the subject, as needed.

Prof. Dr. Catedratico Xoan Leiceaga Baltar
leiceaga@uvigo.es, xleiceaga@aztecaingenieria.es
Head of the Graphic Engineering and Design Group
University of Vigo, Galicia, SPAIN

.

Prof. Oscar E. Ruiz
oruiz@eafit.edu.co
Coordinator CAD CAM CAE Laboratory
EAFIT University, Medellin, COLOMBIA

ABSTRACT. Algorithms are proposed and implemented in a commercial system which allow for the C^1-continuity matching between adjacent B-spline curves and B-spline patches. These algorithms only manipulate the positions of the control points, therefore respecting the constraint imposed by the sizes of the available commercial steel plates. The application of the algorithms respect the initial hull partition made by the designers and therefore the number and overall shape and position of the constitutive patches remains unchanged. Algorithms were designed and tested for smoothing the union of (a) two B-spline curves sharing a common vertex, (b) two B-spline surfaces sharing a common border, and (c) four B-spline surfaces sharing a common vertex. For this last case, an iterative heuristic degree-of-freedom elimination algorithm was implemented. Very satisfactory results were obtained with the application of the presented algorithms in shipyards in Spain.

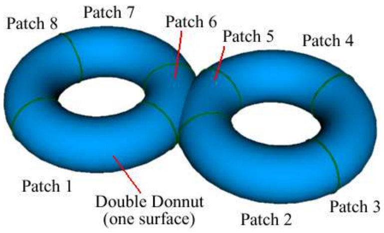
1. Introduction and Literature Review

B-spline curves and surfaces have been used extensively in the past to define ship-hull geometry for design purposes [11, 6]. The popularity of B-spline for free-form surface design lies in their useful characteristics, such as local support, the convex hull, and variation-diminishing properties [2]. A discussion of B-spline curves and surfaces, and their suitability for ship hull surface definition can be found in Rogers [10].

Applications of e.g. Computational Fluid Dynamics use single patch representations, which solve the issue of smoothness by itself [8], but do not reflect that the manufacture and assembly are performed with smaller standard plates, as produced in the steel mills. Also, fitting the complex surface of a ship hull with a single B-spline patch may lead to either an inaccurate representation, or a designer-unfriendly representation i.e. a single patch with a high number of control points. On the other hand, since a single B-spline patch can only represent surfaces of simple topological type, a surface of arbitrary topological type (see Figures 1(a) and 1(b)) must be defined as a set of B-spline patches [5]. The set of patches must constitute a partition of the ship hull surface and must also maintain tangent plane continuity (C^1 continuity) across neighboring patches. Enforcing C^1 continuity between adjacent patches while at the same time fitting the patch network to the points (of the ship hull surface in this case) is a challenging problem [5].

Loop [7] presents an algorithm for creating a smooth set of rectangular and triangular spline surfaces, starting with an irregular mesh of polygonal flat faces. The algorithm takes into consideration curvature parameters to decide the tiling or merging of patches. The final result may have spline patches of sizes and shapes dictated by the curvature criteria. Because of this characteristic, the algorithm is not suitable to be applied in the problem at hand, in which one must respect the constraint posed by the predefined plates with which the hull is to be constructed.

Ball [1] and Peters [9] derive continuity conditions for the subdivision of surfaces. Ball uses Fourier transform-based techniques to do so. Peters presents a method for verifying



(a) Partition of a 2-genus 2-Manifold (double donnut)

(b) Partition of Ship Bow

FIGURE 3.1. Non-rectangular Partition of 2-manifolds with Rectangular Patches

smoothness of subdivided B-spline surfaces generated using Doo-Sabin [4] and Catmull-Clark [3] subdivision algorithms. In our case, subdivision is not only unnecessary but also not allowed, since the steel plates to manufacture the hull are pre-defined. Our goal is to respect the collection of B-spline patches, and to slightly modify their control points to achieve C^1 continuity among them.

Bardis [2] presents an algorithm for C^1 continuity between adjacent patches which requires the merging of all the knot vectors of the B-spline patches, the unification of the order and of the number of vertices of the control polygons, and the use of arbitrarily selected scalar functions called *bias*. Hence, it was not compliant with our goal of smoothing B-splines by modifying only their control points.

For the making of software for the shipbuilding industry no explicit algorithms for B-spline curve and surface smoothing were found in the reviewed literature. It thus became necessary to design and implement own algorithms for this task. It is the purpose of this paper to present the designed algorithms for B-spline curve and surface smoothing, together with the results obtained to smooth real ship B-spline surface patches. The paper is structured as follows: Section 2 presents a brief description of the ship hull surface modeling process using B-spline curves and surfaces. Section 3 presents an algorithm for B-spline curves smoothing. Section 4 presents two algorithms for B-spline surfaces smoothing: one for two adjacent surfaces sharing a common border, and one for four surfaces incident to a common vertex. Conclusions are presented in section 7.

2. Hull surface modeling using a set of B-spline surfaces

The computer modeling of a ship hull is performed, in our case, from the ship hull lines. These lines are planar curves in R^3 resulting from the intersection of the ship hull surface against cross sections perpendicular to the axes of the ship coordinate system. The modeling process is roughly as follows: (i) A set of B-spline curves is manually fitted to ship hull lines. Several rectangular regions on the ship hull surface result from this process, as shown in figure 3.2. (ii) Rectangular B-spline patches are generated from the four B-spline curves that enclose each of these regions. An initial model of the ship hull surface, constituted by a network of C^0-continuous rectangular B-spline patches is thus obtained. (iii) Each pair of adjacent patches is smoothed using the implementation of the algorithm described in section 4.2.1. Every set of four patches sharing a common vertex is also smoothed using the implementation of the algorithm described in section 4.2.2. The final result of the process is a set of rectangular B-spline patches whose union is C^1-continuous, and constitutes the final model of the ship hull surface (see Figure 3.10).

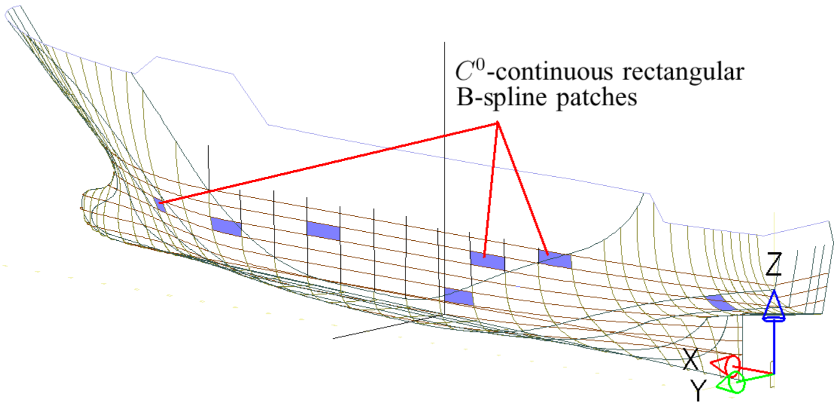


FIGURE 3.2. Set of B-spline curves interpolating the ship lines and local C^0 B-Spline patches

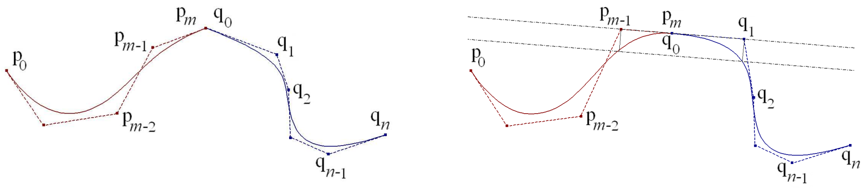
3. Methodology. Smoothing of B-spline curves in shared vertices

3.1. Condition for C^1 continuity between B-Spline curves. Let P and Q be two B-Spline curves in R^3. Let $S_P = \{\mathbf{p}_0, \mathbf{p}_1, \ldots, \mathbf{p}_m\}$ and $S_Q = \{\mathbf{q}_0, \mathbf{q}_1, \ldots, \mathbf{q}_n\}$, $\mathbf{p}_i, \mathbf{q}_i \in R^3$, be the sequences of control points of P and Q, respectively. If $\mathbf{p}_m = \mathbf{q}_0$, i.e. P and Q are C^0-continuous at $\mathbf{p}_m$, then P and Q are also C^1-continuous at $\mathbf{p}_m$ if $\mathbf{p}_{m-1}$, $\mathbf{p}_m$, and $\mathbf{q}_1$ are collinear, and $\mathbf{p}_m$ lies between $\mathbf{p}_{m-1}$ and $\mathbf{q}_1$, i.e. if there exists $\lambda \in (0,1) \subset R$ such that

$$\mathbf{p}_m = \mathbf{q}_0 = (1-\lambda)\,\mathbf{p}_{m-1} + \lambda \mathbf{q}_1 \tag{9}$$

3.2. Algorithm for C^1 continuity between curves. Given two separate B-Spline curves P and Q in R^3 connected at a common endpoint $\mathbf{p}_m = \mathbf{q}_0$ (see figure 3(a)), the **goal** of a curve smoothing process is to determine new positions for the control points of P and Q so that the two curves become C^1-continuous at $\mathbf{p}_m$, i.e. the normalized direction vectors of P and Q at $\mathbf{p}_m$ are equal. If the union of the curves P and Q is required to be smoothed at point $\mathbf{p}_m$, and $\mathbf{p}_{m-1}$, $\mathbf{p}_m$ and $\mathbf{q}_1$ are not collinear, at least one of these three points must be moved in order to do so. Although infinite solutions to this problem exist (there are infinite ways of arranging three points to lie in a same line), some of them are more suitable for design and construction purposes. For instance, sometimes the shared control point is desired to remain fixed (see figure 3(b)).

Suppose that we want to force $\mathbf{p}_{m-1}$, $\mathbf{p}_m$, and $\mathbf{q}_1$ to lie in the same line, by moving $\mathbf{p}_{m-1}$ and $\mathbf{q}_1$ to new positions $\mathbf{p}^*_{m-1}$ and $\mathbf{q}^*_1$, and leaving $\mathbf{p}_m$ fixed. A way to calculate $\mathbf{p}^*_{m-1}$ and $\mathbf{q}^*_1$ is as follows: Let L be the line passing through $\mathbf{p}_{m-1}$ and $\mathbf{q}_1$, and L^* be the line passing through $\mathbf{p}_m$ and parallel to L. Let $\Pi_{\mathbf{p}_{m-1}}$ and $\Pi_{\mathbf{q}_1}$ be the planes with normal vector $\hat{\mathbf{n}}$ and respective pivot points $\mathbf{p}_{m-1}$ and $\mathbf{q}_1$, where $\hat{\mathbf{n}} = (\mathbf{q}_1 - \mathbf{p}_{m-1})/(\|\mathbf{q}_1 - \mathbf{p}_{m-1}\|)$. It can be seen that possible values for $\mathbf{p}^*_{m-1}$ and $\mathbf{q}^*_1$ that satisfy equation 9 are given by $\mathbf{p}^*_{m-1} = \Pi_{\mathbf{p}_{m-1}} \cap L^*$ and $\mathbf{q}^*_1 = \Pi_{\mathbf{q}_1} \cap L^*$.



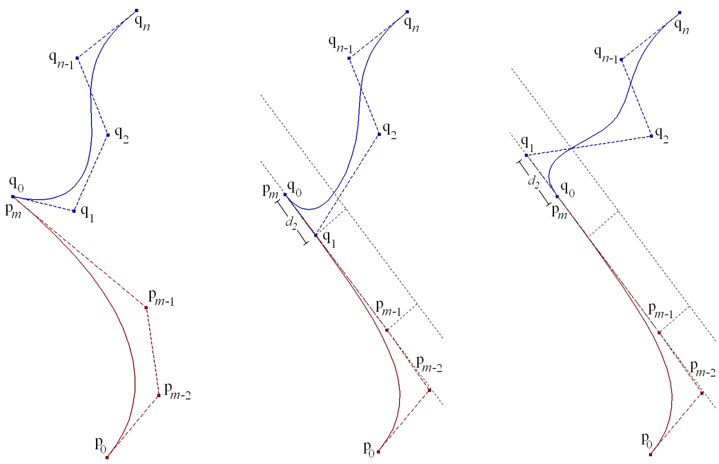
(a) B-Spline curves with C^0 continuity at $p_m = q_0$

(b) B-Spline curves with C^1 continuity at $p_m = q_0$

FIGURE 3.3. C^1 Continuity between adjacent B-Spline curves by adjusting p_{m-1} and q_1

3.2.1. *Exception Treatment.* Let λ^* be the value of λ at which $\mathbf{p}^*_{m-1}$ and $\mathbf{q}^*_1$ satisfy equation 9. Because the procedure described above does not ensure that $\lambda^* \in (0,1)$, an additional step becomes necessary. If $\lambda^* \notin (0,1)$, then $\mathbf{p}_m$ does not lie between $\mathbf{p}^*_{m-1}$ and $\mathbf{q}^*_1$. It is necessary to force $\mathbf{p}_m$ to lie between $\mathbf{p}^*_{m-1}$ and $\mathbf{q}^*_1$. Since $\mathbf{p}_m$ is required to remain fixed, $\mathbf{p}^*_{m-1}$ or $\mathbf{q}^*_1$ should be moved again. To avoid an excessive change in the geometry of the curves, the point to be moved will be the one that lies the closest to $\mathbf{p}_m$.

Let $d_1 = \left\|\mathbf{p}_m - \mathbf{p}^*_{m-1}\right\|$ and $d_2 = \|\mathbf{p}_m - \mathbf{q}^*_1\|$. If $d_1 \le d_2$, $\mathbf{p}^*_{m-1}$ will be moved to a final position $\mathbf{p}^{**}_{m-1} = \mathbf{p}_m + \left(\mathbf{p}_m - \mathbf{p}^*_{m-1}\right)$. If $d_1 > d_2$, $\mathbf{q}^*_1$ will be moved to a final position $\mathbf{q}^{**}_1 = \mathbf{p}_m + (\mathbf{p}_m - \mathbf{q}^*_1)$ (see Figures 4(a)- 4(c)).



(a) B-Spline curves with C^0 continuity at $p_m = q_0$

(b) B-Spline curves after applying algorithm without correction

(c) B-Spline curves become C^1-continuous at $p_m = q_0$ after correction

FIGURE 3.4. Exception Treatment. Continuity between adjacent B-Spline Curves

4. Methodology. Smoothing of B-Spline Surfaces in shared borders

4.1. Condition for C^1 continuity between B-Spline surfaces. Let A be a B-Spline surfaces and P^A the array of control points of A,

$$P^A = \begin{bmatrix} \mathbf{p}^A_{11} & \mathbf{p}^A_{12} & \cdots & \mathbf{p}^A_{1n} \\ \mathbf{p}^A_{21} & \mathbf{p}^A_{22} & \cdots & \mathbf{p}^A_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{p}^A_{m1} & \mathbf{p}^A_{m2} & \cdots & \mathbf{p}^A_{mn} \end{bmatrix}$$

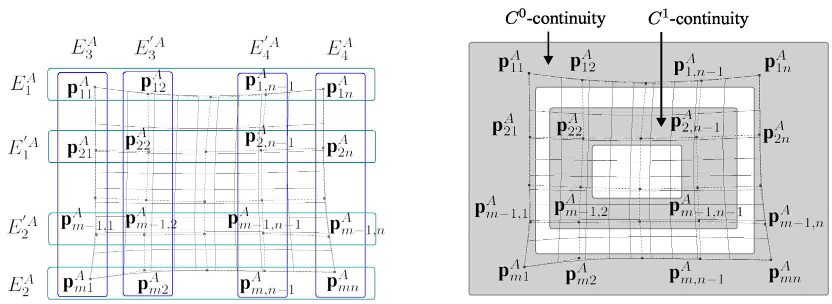
where $\mathbf{p}^A_{ij} \in R^3$.

Definition. *Alignment* of PL curves.

Let $E_1 = [\mathbf{p}_{11}, \mathbf{p}_{12}, \ldots, \mathbf{p}_{1n}]$, $E_2 = [\mathbf{p}_{21}, \mathbf{p}_{22}, \ldots, \mathbf{p}_{2n}]$ and $E_3 = [\mathbf{p}_{31}, \mathbf{p}_{32}, \ldots, \mathbf{p}_{3n}]$ be three sequences of control points, where $\mathbf{p}_{ij} \in R^3$. We say that E_1, E_2 and E_3 are *aligned* if for all $j = 1, 2, \ldots, n$, the points $\mathbf{p}_{1j}$, $\mathbf{p}_{2j}$ and $\mathbf{p}_{3j}$ are collinear exactly in that order, i.e. satisfy equation $\mathbf{p}_{2j} = (1 - \lambda)\,\mathbf{p}_{1j} + \lambda \mathbf{p}_{3j}$ with $\lambda \in (0, 1)$.

The boundary control point sequences for A are $E_1^A = \left[\mathbf{p}^A_{11}, \mathbf{p}^A_{12}, \ldots, \mathbf{p}^A_{1n}\right]$, $E_2^A = \left[\mathbf{p}^A_{m1}, \mathbf{p}^A_{m2}, \ldots, \mathbf{p}^A_{mn}\right]$, $E_3^A = \left[\mathbf{p}^A_{11}, \mathbf{p}^A_{21}, \ldots, \mathbf{p}^A_{m1}\right]$ and $E_4^A = \left[\mathbf{p}^A_{1n}, \mathbf{p}^A_{2n}, \ldots, \mathbf{p}^A_{mn}\right]$. Let B be another B-Spline surface. We say that the control points of the i-th border of A are equal to the control points of the j-th border of B, if there exist $i, j \in \{1, 2, 3, 4\}$, such that $E_i^A = E_j^B$ or $E_i^A = E_j^{*B}$, where E_j^{*B} is the reverse-order vesion of E_j^B. A necessary but not sufficient condition for A to be C^0-continuous with B at the i-th border of A and the j-th border of B is that the control points of these two borders be equal.

Let us also define a sequence of control points $E_i^{'A}$ associated to each border E_i^A, for $i = 1, 2, 3, 4$, as per figure 5(a). $E_1^{'A} = \left[\mathbf{p}^A_{21}, \mathbf{p}^A_{22}, \ldots, \mathbf{p}^A_{2n}\right]$,
$E_2^{'A} = \left[\mathbf{p}^A_{m-1,1}, \mathbf{p}^A_{m-1,2}, \ldots, \mathbf{p}^A_{m-1,n}\right]$, $E_3^{'A} = \left[\mathbf{p}^A_{12}, \mathbf{p}^A_{22}, \ldots, \mathbf{p}^A_{m2}\right]$ and
$E_4^{'A} = \left[\mathbf{p}^A_{1,n-1}, \mathbf{p}^A_{2,n-1}, \ldots, \mathbf{p}^A_{m,n-1}\right]$.



(a) Sequences of control points $E_i^{'A}$ associated to each E_i^A, for $i = 1, 2, 3, 4$

(b) Control points governing C^0 and C^1 continuity

FIGURE 3.5. Sequences of control points in A

Let A be C^0-continuous with B, at the i-th border of A and the j-th border of B. This implies that $E_i^A = E_j^B$ or $E_i^A = E_j^{*B}$. Unless otherwise stated, two surfaces ”being C^0-continuous” means that they meet at border i (in A) and j (in B). Also we assume WLOG that $E_i^A = E_j^B$ (the vertices are enumerated in identical order). The same observation holds for C^1 continuity. We say that A is C^1-continuous with B, if $E_i^{'A}$, E_i^A, and $E_j^{'B}$ are aligned exactly in that order.

4.2. Algorithms for surface C^1 continuity. Two different smoothing processes are identified here. The basic surface-smoothing process consists in achieving C^1 continuity between two surfaces at their common border, i.e. the border at which the surfaces are C^0-continuous. A second process consists in achieving C^1 continuity between four pairwise-C^0-continuous surfaces sharing a vertex, at their common borders.

4.2.1. *C^1 continuity between two surfaces at a common border*. **Given** two separate B-Spline surfaces A and B in R^3, connected at a common *border*, $E_i^A = E_j^B$, the **goal** of a surface-smoothing process is to determine new positions for the control points of A and B so that the two surfaces become C^1-continuous at their common border. The procedure is to make collinear the $E_{ik}^{\prime A}$, E_{ik}^A, $E_{jk}^{\prime B}$ points for $k = 1 \ldots m$, that is, to pairwise align the control points at the seam between the two patches (m is the number of control points of such borders).

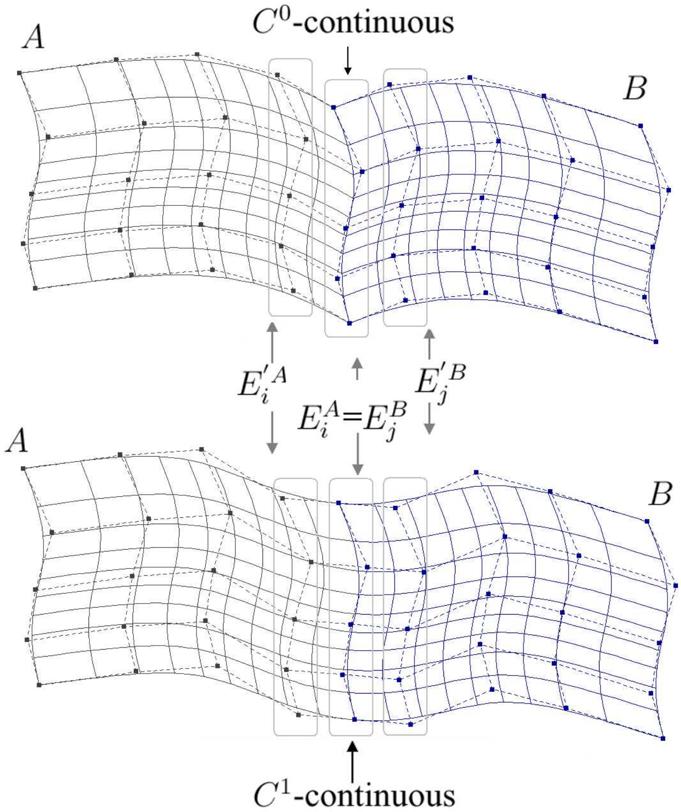


FIGURE 3.6. $C^{0,1}$-continuity between A and B at i-th border of A, and j-th border of B

4.2.2. *C^1 continuity between four surfaces at common vertex.* Let A, B, C, and D be adjacent B-Spline surfaces, meeting at one vertex. The meeting borders among them are: $E_i^A = E_k^B$, $E_l^B = E_m^C$, $E_n^C = E_o^D$, $E_p^D = E_j^A$. The common vertex is $P_{i^*j^*}^A = P_{k^*l^*}^B = P_{m^*n^*}^C = P_{o^*p^*}^D$ Subscripts take values between 1 and 4.

The arrangement of surfaces A, B, C, D, shown in figure 3.7 satisfies the previous conditions, since the four surfaces are pairwise-C^0-continuous and have a common control point that belongs to all the borders at which the surfaces are C^0-continuous.

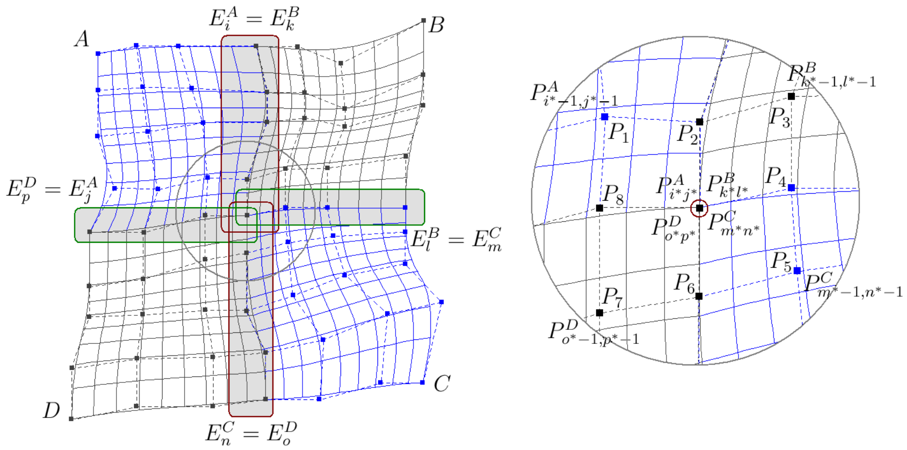


FIGURE 3.7. C^0-continuous set of four adjacent B-Spline surfaces meeting at a common point

Given four B-Spline surfaces A, B, C, and D in R^3, satisfying conditions mentioned above, the **goal** of a surface-smoothing process is to determine new positions for the control points of A, B, C and D, so that the union of the four surfaces becomes C^1-continuous.

Separately achieving pairwise-C^1 continuity between the four B-Spline surfaces includes calculating correct modified positions of the controls points of A, B, C and D. However, such a process does not correctly calculate the positions for the common point (P_0) and its surrounding 8 vertices ($P_1, \ldots, P_8$ in Figure 3.7)

Algorithm 2 calculates the modified positions of $P_0, P_1, \ldots, P_8$ such that C^1 Continuity among the union of A, B, C and D is achieved. This algorithm is based on the fact that if $P_0, P_1, \ldots, P_8$ lie on the same plane, and the elements in each of the following sequences $s_1 = [P_1, P_2, P_3]$, $s_2 = [P_3, P_4, P_5]$, $s_3 = [P_5, P_6, P_7]$, $s_4 = [P_7, P_8, P_1]$, are collinear exactly in that order, then C^1 Continuity is achieved at points $P_0, P_1, \ldots, P_8$. For the sake of compactness in the article we omit the proof of convergence for algorithm 2.

Algorithm 2 C^1 continuity between four surfaces

```
Identify values of i, j, k, l, m, n, o, p
Pairwise-smooth surfaces A with B, B with C, C with D, D with A
Calculate best-fit plane Π* for points P0, P1, ..., P8
Project points P0, P1, ..., P8 into Π*
while P1, P3, P5, P7 do not converge do
    Move P1 to make P1, P2, P3 collinear (algorithm in section 3.2)
    Move P3 to make P3, P4, P5 collinear
    Move P5 to make P5, P6, P7 collinear
    Move P7 to make P7, P8, P1 collinear
end while
```

The Figure 3.8 shows the dynamic non-linear system simulation of the state variables P_1, P_3, P_5 and P_7. It illustrates that this algorithm iteratively modifies the positions of P_1, P_3, P_5 and P_7 so that upon convergence the quadrilateral $[P_1, P_3, P_5, P_7]$ contains the fixed points P_2 in $\overline{P_1, P_3}$, P_4 in $\overline{P_3, P_5}$ P_6 in $\overline{P_5, P_7}$ and P_8 in $\overline{P_7, P_1}$. It can be seen that convergence is extremely fast (about 3 iterations) to the final positions.

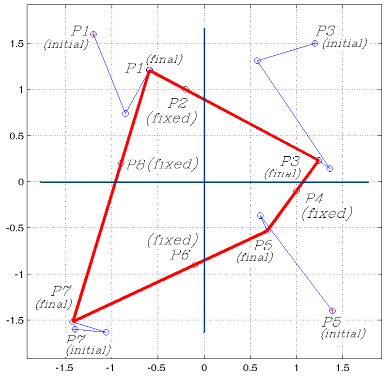


FIGURE 3.8. State Space Non-Linear Dynamic System Simulation of the 4-patch Smooting Algorithm

5. Results

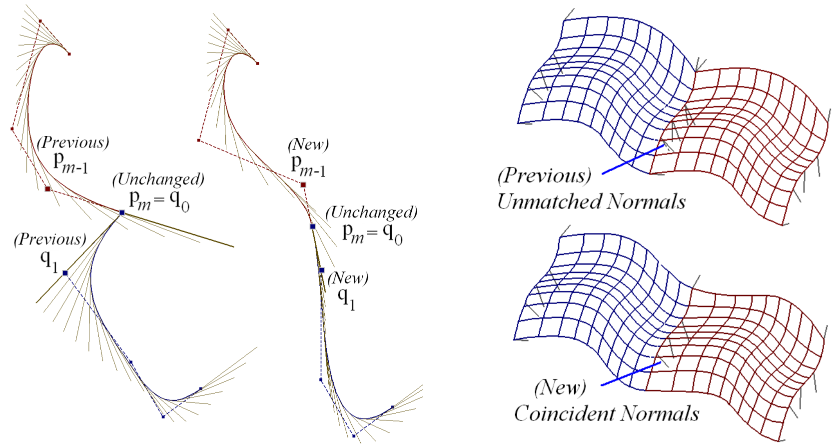
A large number of adjacent B-spline curves were smoothed using the industrial implementation of the algorithm described in section 3. After the algorithm was applied, the upper bound of the angular deviation between tangent vectors at the boundary of matched curves was 2.9×10^{-5} degrees (figure 9(a)).

Likewise, a large number of adjacent B-spline surfaces were smoothed using the algorithm described in section 4.2.1. The relative error between the normal vectors of both surfaces along their common border remained below 10^{-5} degrees (figure 9(b)). Figure 3.10 shows the final result of the 4-patch smoothing algorithm.

6. Conclusions

Industrially implemented algorithms for B-spline curve and surface smoothing were discussed in this paper. The algorithms achieve C^1 continuity between adjacent curves and surfaces by modifying only the positions of their control points. The main advantages of the presented algorithms are their simplicity, which results in their easy implementation and modification, and the fact that properties of the curves and surfaces such as their order and their poles remain unchanged. Several tests were made to the obtained smoothed curves and surfaces, based on the tangent and normal vectors of the B-spline at their common point or border. The relative error between the components of the tangent and normal vectors was in all test cases below 10^{-5} degrees.

Several real ship hull surfaces have been modeled at the Design and Engineering Group (GED), Universidade de Vigo, following the discussed methodology. One of these models was presented in this paper.



(a) Vectors tangent to two adjacent B-spline curves, before and after being smoothed

(b) Vectors Normal to two adjacent surfaces, before and after being smoothed

FIGURE 3.9. Tangent and Normal vectors to B-spline curves and surfaces used for C^1 continuity testing

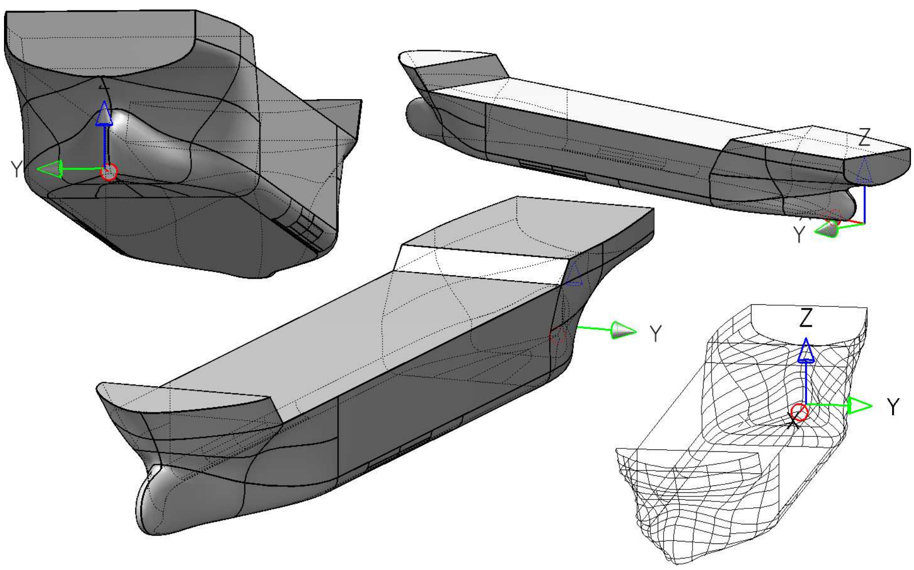


FIGURE 3.10. Ship hull surface obtained through the procedure described in section 2

Bibliography

[1] A. A. Ball and D. J. T. Storry. Conditions for tangent plane continuity over recursively generated b-spline surfaces. *ACM Trans. Graph.*, 7(2):83–102, 1988.

[2] Leonidas Bardis and Maria-Eleni Vafiadou. Ship-hull geometry representation with b-spline surface patches. *Computer-Aided Design*, 24(4):217–222, 1992.

[3] E. Catmull and J. Clark. Recursively generated b-spline surfaces on arbitrary topological meshes. *Computer Aided Design*, 10:350–355, 1978.

[4] D. Doo and M. A. Sabin. Behaviour of recursive subdivision surfaces near extraordinary points. *Computer Aided Design*, 10:356–360, 1978.

[5] Matthias Eck and Hugues Hoppe. Automatic reconstruction of b-spline surfaces of arbitrary topological type. In *SIGGRAPH '96*, pages 325–334, New York, NY, USA, 1996. ACM Press.

[6] N.G. Fog. Creative definition and fairing of ship hulls using a b-spline surface. *Computer-Aided Design*, 16(4):225–229, July 1984.

[7] Charles Loop. Smooth spline surfaces over irregular meshes. In *SIGGRAPH '94*, pages 303–310, New York, NY, USA, 1994. ACM Press.

[8] S. Percival, D. Hendrix, and F. Noblesse. Hydrodynamic optimization of ship hull forms. *Applied Ocean Research*, 23(6):337–355, December 2001.

[9] Jörg Peters and Ulrich Reif. Analysis of algorithms generalizing b-spline subdivision. *SIAM J. Numer. Anal.*, 35(2):728–748, 1998.

[10] D.F. Rogers. B-spline curves and surfaces for ship hull definition. In *SCAHD '77: Computer-aided hull surface definition symposium*, pages 1–25, Annapolis, MD, USA, 1977.

[11] D.F. Rogers, S.G. Satterfield, and F.A. Rodriguez. Ship hulls, b-spline surfaces and cad/cam. *IEEE Computer Graphics and Applications*, pages 37–45, December 1983.

Conclusion

Solutions to three geometric problems arising in real industrial applications have been presented in this work. Such solutions combine tools taken from different fields of computational geometry including combinatorial geometry, stochastic geometry, and numerical geometry. The successful application of the implemented solutions to real problems in the industries for which they were built is the most important contribution of this work.

Combination of stochastic and deterministic methods of computational geometry was an interesting exercise during the development of the projects. In particular, the use of Principal Component Analysis as a method to reduce the dimension of data sets proved to be essential in the treatment of problems where input data are generated from surface sampling and/or in an unknown coordinate system. In the first case, PCA is used to detect and eliminate the variance explained by the intrinsic noise of sampling devices. In the second case, PCA can be used to identify the direction of largest variance and for example to assign a principal axis to a solid in the direction of its largest expansion.

Academic fields and topics that were studied during the development of these projects include: (Discrete) Differential Geometry, Solid Geometric Modeling, Topological and Geometrical Correctness of Manifolds, Spatial Statistics, Stochastic Computational Geometry, Heuristic Methods in Dynamic Equations, and Prevention of Degeneracies of constructions in Descriptive Geometry. Experience in the use of Programming Languages, Application Programming Interfaces (APIs) of CAD Packages, and CAD packages was acquired. Skills in algorithm design, mathematical formulation of problems and methods, problem solving, literature reviewing, scientific rhetoric, paper writing, and oral presentations, have also been developed and/or strengthened throughout this work.

The valuable interaction with advisors, professors, and researchers at EAFIT University and Universities and Institutions abroad was essential in the successful development of this work. It was of particular importance the contact with other cultures, values and working environments.