LNCS 3604

Ralph Martin
Helmut Bez
Malcolm Sabin (Eds.)

Mathematics of Surfaces XI

11th IMA International Conference
Loughborough, UK, September 2005
Proceedings

Springer

Ralph Martin Helmut Bez
Malcolm Sabin (Eds.)

Mathematics of Surfaces XI

11th IMA International Conference
Loughborough, UK, September 5-7, 2005
Proceedings

Springer

Volume Editors

Ralph Martin
Cardiff University
School of Computer Science
5 The Parade, Roath, Cardiff, UK, CF24 3AA
E-mail: Ralph.Martin@cs.cf.ac.uk

Helmut Bez
Loughborough University
Department of Computer Science
Loughborough, UK, LE11 3TU
E-mail: h.e.bez@lboro.ac.uk

Malcolm Sabin
Numerical Geometry Ltd
26 Abbey Lane, Cambridge, UK, CB5 9EP
E-mail: malcolm@geometry.demon.co.uk

Library of Congress Control Number: 2005930889

CR Subject Classification (1998): I.3.5, I.3, I.1, G.2, G.1, F.2, I.4

ISSN 0302-9743
ISBN-10 3-540-28225-4 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-28225-9 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media

springeronline.com

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11537908 06/3142 5 4 3 2 1 0

Preface

This volume collects the papers accepted for presentation at the 11th IMA Conference on the Mathematics of Surfaces, held at Loughborough University, 5th–7th September 2005. As with all earlier conferences in the series, contributors to this volume come from many countries. The papers presented here reflect the interest in a subject of relevance to mathematics, engineering, and computer science, especially in domains such as computer-aided design, computer vision, and computer graphics.

The papers in the present volume include eight invited papers, as well as a larger number of submitted papers. They cover a range of ideas from underlying theoretical tools to industrial and medical uses of surfaces. The latter category includes such diverse topics as surfaces in car design, and modelling of teeth, while the former includes papers on Voronoi diagrams, linear systems, estimation of curvatures on meshes, operators on meshes, intersection of subdivision surfaces, approximate parameterization, condition numbers, Pythagorean hodographs, artifacts in B-spline surfaces, Bézier surfaces of minimal energy, line subdivision, subdivision surfaces, level sets and symmetry, the topology of algebraic surfaces, curve analysis, interpolation with positivity, and conversion of cyclides to NURBS. Other papers concentrate on particular algorithms arising from applications, such as embedding graphs in manifolds, recovery of 3D shape from shading, finding optimal feedrates for machining, detection of creases in range data, and filling holes in range data.

We would like to thank all those who attended the conference and helped to make it a success. We are particularly grateful to Lucy Nye at the Institute of Mathematics and Its Applications for her hard work in organizing many aspects of the conference, and to Alfred Hofmann and Frank Holzwarth of Springer for their help in publishing this volume. Following this Preface is a list of distinguished researchers who formed the International Programme Committee, and who freely gave their time in helping to assess papers for these proceedings. Due to their work, many of the papers have been considerably improved. Our thanks go to all of them, and to other people who they called upon to help with refereeing.

June 2005

Ralph Martin,
Helmut Bez,
Malcolm Sabin

Organization

The Mathematics of Surfaces XI conference was organized by the Insitute of Mathematics and Its Applications (Catherine Richards House, 16 Nelson St., Southend-on-Sea, Essex, SS1 1EF, UK), and Loughborough University.

Organizing Committee

Helmut Bez	Loughborough University, UK
Ralph Martin	Cardiff University, UK
Malcolm Sabin	Numerical Geometry Ltd., UK

Program Committee

Bob Cripps	University of Birmingham
Gerald Farin	Arizona State University
Peter Giblin	University of Liverpool
Ron Goldman	Rice University
Tim Goodman	University of Dundee
Hans Hagen	University of Kaiserslautern
Edwin Hancock	University of York
Shimin Hu	Tsinghua University
Bert Jüttler	Johannes Kepler University
Myung-Soo Kim	Seoul National University
Tom Lyche	University of Oslo
Nick Patrikalakis	MIT
Jorg Peters	University of Florida
Hong Qin	State University of New York
Martin Rumpf	University of Duisberg
Georg Umlauf	University of Kaiserslautern
Wenping Wang	Hong Kong University
Joe Warren	Rice University
Mike Wilson	University of Leeds
Franz-Erich Wolter	University of Hannover
Guoliang Xu	Chinese Academy of Sciences

Lecture Notes in Computer Science

For information about Vols. 1–3542

please contact your bookseller or Springer

Vol. 3659: J.R. Rao, B. Sunar (Eds.), Cryptographic Hardware and Embedded Systems - CHES 2005. XIV, 458 pages. 2005.

Vol. 3654: S. Jajodia, D. Wijesekera (Eds.), Data and Applications Security XIX. X, 353 pages. 2005.

Vol. 3653: M. Abadi, L.d. Alfaro (Eds.), CONCUR 2005 – Concurrency Theory. XIV, 578 pages. 2005.

Vol. 3649: W.M.P. van der Aalst, B. Benatallah, F. Casati, F. Curbera (Eds.), Business Process Management. XII, 472 pages. 2005.

Vol. 3645: D.-S. Huang, X.-P. Zhang, G.-B. Huang (Eds.), Advances in Intelligent Computing, Part II. XIII, 1010 pages. 2005.

Vol. 3644: D.-S. Huang, X.-P. Zhang, G.-B. Huang (Eds.), Advances in Intelligent Computing, Part I. XXVII, 1101 pages. 2005.

Vol. 3639: P. Godefroid (Ed.), Model Checking Software. XI, 289 pages. 2005.

Vol. 3638: A. Butz, B. Fisher, A. Krüger, P. Olivier (Eds.), Smart Graphics. XI, 269 pages. 2005.

Vol. 3636: M.J. Blesa, C. Blum, A. Roli, M. Sampels (Eds.), Hybrid Metaheuristics. XII, 155 pages. 2005.

Vol. 3634: L. Ong (Ed.), Computer Science Logic. XI, 567 pages. 2005.

Vol. 3633: C. Bauzer Medeiros, M. Egenhofer, E. Bertino (Eds.), Advances in Spatial and Temporal Databases. XIII, 433 pages. 2005.

Vol. 3632: R. Nieuwenhuis (Ed.), Automated Deduction – CADE-20. XIII, 459 pages. 2005. (Subseries LNAI).

Vol. 3627: C. Jacob, M.L. Pilat, P.J. Bentley, J. Timmis (Eds.), Artificial Immune Systems. XII, 500 pages. 2005.

Vol. 3626: B. Ganter, G. Stumme, R. Wille (Eds.), Formal Concept Analysis. X, 349 pages. 2005. (Subseries LNAI).

Vol. 3625: S. Kramer, B. Pfahringer (Eds.), Inductive Logic Programming. XIII, 427 pages. 2005. (Subseries LNAI).

Vol. 3624: C. Chekuri, K. Jansen, J.D.P. Rolim, L. Trevisan (Eds.), Approximation, Randomization and Combinatorial Optimization. XI, 495 pages. 2005.

Vol. 3623: M. Liśkiewicz, R. Reischuk (Eds.), Fundamentals of Computation Theory. XV, 576 pages. 2005.

Vol. 3621: V. Shoup (Ed.), Advances in Cryptology – CRYPTO 2005. XI, 568 pages. 2005.

Vol. 3620: H. Muñoz-Avila, F. Ricci (Eds.), Case-Based Reasoning Research and Development. XV, 654 pages. 2005. (Subseries LNAI).

Vol. 3619: X. Lu, W. Zhao (Eds.), Networking and Mobile Computing. XXIV, 1299 pages. 2005.

Vol. 3615: B. Ludäscher, L. Raschid (Eds.), Data Integration in the Life Sciences. XII, 344 pages. 2005. (Subseries LNBI).

Vol. 3614: L. Wang, Y. Jin (Eds.), Fuzzy Systems and Knowledge Discovery, Part II. XLI, 1314 pages. 2005. (Subseries LNAI).

Vol. 3613: L. Wang, Y. Jin (Eds.), Fuzzy Systems and Knowledge Discovery, Part I. XLI, 1334 pages. 2005. (Subseries LNAI).

Vol. 3612: L. Wang, K. Chen, Y. S. Ong (Eds.), Advances in Natural Computation, Part III. LXI, 1326 pages. 2005.

Vol. 3611: L. Wang, K. Chen, Y. S. Ong (Eds.), Advances in Natural Computation, Part II. LXI, 1292 pages. 2005.

Vol. 3610: L. Wang, K. Chen, Y. S. Ong (Eds.), Advances in Natural Computation, Part I. LXI, 1302 pages. 2005.

Vol. 3608: F. Dehne, A. López-Ortiz, J.-R. Sack (Eds.), Algorithms and Data Structures. XIV, 446 pages. 2005.

Vol. 3607: J.-D. Zucker, L. Saitta (Eds.), Abstraction, Reformulation and Approximation. XII, 376 pages. 2005. (Subseries LNAI).

Vol. 3606: V. Malyshkin (Ed.), Parallel Computing Technologies. XII, 470 pages. 2005.

Vol. 3604: R. Martin, H. Bez, M. Sabin (Eds.), Mathematics of Surfaces XI. IX, 473 pages. 2005.

Vol. 3603: J. Hurd, T. Melham (Eds.), Theorem Proving in Higher Order Logics. IX, 409 pages. 2005.

Vol. 3602: R. Eigenmann, Z. Li, S.P. Midkiff (Eds.), Languages and Compilers for High Performance Computing. IX, 486 pages. 2005.

Vol. 3599: U. Aßmann, M. Aksit, A. Rensink (Eds.), Model Driven Architecture. X, 235 pages. 2005.

Vol. 3598: H. Murakami, H. Nakashima, H. Tokuda, M. Yasumura, Ubiquitous Computing Systems. XIII, 275 pages. 2005.

Vol. 3597: S. Shimojo, S. Ichii, T.W. Ling, K.-H. Song (Eds.), Web and Communication Technologies and Internet-Related Social Issues - HSI 2005. XIX, 368 pages. 2005.

Vol. 3596: F. Dau, M.-L. Mugnier, G. Stumme (Eds.), Conceptual Structures: Common Semantics for Sharing Knowledge. XI, 467 pages. 2005. (Subseries LNAI).

Vol. 3595: L. Wang (Ed.), Computing and Combinatorics. XVI, 995 pages. 2005.

Vol. 3594: J.C. Setubal, S. Verjovski-Almeida (Eds.), Advances in Bioinformatics and Computational Biology. XIV, 258 pages. 2005. (Subseries LNBI).

Vol. 3593: V. Mařík, R. W. Brennan, M. Pěchouček (Eds.), Holonic and Multi-Agent Systems for Manufacturing. XI, 269 pages. 2005. (Subseries LNAI).

Vol. 3592: S. Katsikas, J. Lopez, G. Pernul (Eds.), Trust, Privacy and Security in Digital Business. XII, 332 pages. 2005.

Vol. 3591: M.A. Wimmer, R. Traunmüller, Å. Grönlund, K.V. Andersen (Eds.), Electronic Government. XIII, 317 pages. 2005.

Vol. 3590: K. Bauknecht, B. Pröll, H. Werthner (Eds.), E-Commerce and Web Technologies. XIV, 380 pages. 2005.

Vol. 3589: A M. Tjoa, J. Trujillo (Eds.), Data Warehousing and Knowledge Discovery. XVI, 538 pages. 2005.

Vol. 3588: K.V. Andersen, J. Debenham, R. Wagner (Eds.), Database and Expert Systems Applications. XX, 955 pages. 2005.

Vol. 3587: P. Perner, A. Imiya (Eds.), Machine Learning and Data Mining in Pattern Recognition. XVII, 695 pages. 2005. (Subseries LNAI).

Vol. 3586: A.P. Black (Ed.), ECOOP 2005 - Object-Oriented Programming. XVII, 631 pages. 2005.

Vol. 3584: X. Li, S. Wang, Z.Y. Dong (Eds.), Advanced Data Mining and Applications. XIX, 835 pages. 2005. (Subseries LNAI).

Vol. 3583: R.W. H. Lau, Q. Li, R. Cheung, W. Liu (Eds.), Advances in Web-Based Learning – ICWL 2005. XIV, 420 pages. 2005.

Vol. 3582: J. Fitzgerald, I.J. Hayes, A. Tarlecki (Eds.), FM 2005: Formal Methods. XIV, 558 pages. 2005.

Vol. 3581: S. Miksch, J. Hunter, E. Keravnou (Eds.), Artificial Intelligence in Medicine. XVII, 547 pages. 2005. (Subseries LNAI).

Vol. 3580: L. Caires, G.F. Italiano, L. Monteiro, C. Palamidessi, M. Yung (Eds.), Automata, Languages and Programming. XXV, 1477 pages. 2005.

Vol. 3579: D. Lowe, M. Gaedke (Eds.), Web Engineering. XXII, 633 pages. 2005.

Vol. 3578: M. Gallagher, J. Hogan, F. Maire (Eds.), Intelligent Data Engineering and Automated Learning - IDEAL 2005. XVI, 599 pages. 2005.

Vol. 3577: R. Falcone, S. Barber, J. Sabater-Mir, M.P. Singh (Eds.), Trusting Agents for Trusting Electronic Societies. VIII, 235 pages. 2005. (Subseries LNAI).

Vol. 3576: K. Etessami, S.K. Rajamani (Eds.), Computer Aided Verification. XV, 564 pages. 2005.

Vol. 3575: S. Wermter, G. Palm, M. Elshaw (Eds.), Biomimetic Neural Learning for Intelligent Robots. IX, 383 pages. 2005. (Subseries LNAI).

Vol. 3574: C. Boyd, J.M. González Nieto (Eds.), Information Security and Privacy. XIII, 586 pages. 2005.

Vol. 3573: S. Etalle (Ed.), Logic Based Program Synthesis and Transformation. VIII, 279 pages. 2005.

Vol. 3572: C. De Felice, A. Restivo (Eds.), Developments in Language Theory. XI, 409 pages. 2005.

Vol. 3571: L. Godo (Ed.), Symbolic and Quantitative Approaches to Reasoning with Uncertainty. XVI, 1028 pages. 2005. (Subseries LNAI).

Vol. 3570: A. S. Patrick, M. Yung (Eds.), Financial Cryptography and Data Security. XII, 376 pages. 2005.

Vol. 3569: F. Bacchus, T. Walsh (Eds.), Theory and Applications of Satisfiability Testing. XII, 492 pages. 2005.

Vol. 3568: W.-K. Leow, M.S. Lew, T.-S. Chua, W.-Y. Ma, L. Chaisorn, E.M. Bakker (Eds.), Image and Video Retrieval. XVII, 672 pages. 2005.

Vol. 3567: M. Jackson, D. Nelson, S. Stirk (Eds.), Database: Enterprise, Skills and Innovation. XII, 185 pages. 2005.

Vol. 3566: J.-P. Banâtre, P. Fradet, J.-L. Giavitto, O. Michel (Eds.), Unconventional Programming Paradigms. XI, 367 pages. 2005.

Vol. 3565: G.E. Christensen, M. Sonka (Eds.), Information Processing in Medical Imaging. XXI, 777 pages. 2005.

Vol. 3564: N. Eisinger, J. Małuszyński (Eds.), Reasoning Web. IX, 319 pages. 2005.

Vol. 3562: J. Mira, J.R. Álvarez (Eds.), Artificial Intelligence and Knowledge Engineering Applications: A Bioinspired Approach, Part II. XXIV, 636 pages. 2005.

Vol. 3561: J. Mira, J.R. Álvarez (Eds.), Mechanisms, Symbols, and Models Underlying Cognition, Part I. XXIV, 532 pages. 2005.

Vol. 3560: V.K. Prasanna, S. Iyengar, P.G. Spirakis, M. Welsh (Eds.), Distributed Computing in Sensor Systems. XV, 423 pages. 2005.

Vol. 3559: P. Auer, R. Meir (Eds.), Learning Theory. XI, 692 pages. 2005. (Subseries LNAI).

Vol. 3558: V. Torra, Y. Narukawa, S. Miyamoto (Eds.), Modeling Decisions for Artificial Intelligence. XII, 470 pages. 2005. (Subseries LNAI).

Vol. 3557: H. Gilbert, H. Handschuh (Eds.), Fast Software Encryption. XI, 443 pages. 2005.

Vol. 3556: H. Baumeister, M. Marchesi, M. Holcombe (Eds.), Extreme Programming and Agile Processes in Software Engineering. XIV, 332 pages. 2005.

Vol. 3555: T. Vardanega, A.J. Wellings (Eds.), Reliable Software Technology – Ada-Europe 2005. XV, 273 pages. 2005.

Vol. 3554: A. Dey, B. Kokinov, D. Leake, R. Turner (Eds.), Modeling and Using Context. XIV, 572 pages. 2005. (Subseries LNAI).

Vol. 3553: T.D. Hämäläinen, A.D. Pimentel, J. Takala, S. Vassiliadis (Eds.), Embedded Computer Systems: Architectures, Modeling, and Simulation. XV, 476 pages. 2005.

Vol. 3552: H. de Meer, N. Bhatti (Eds.), Quality of Service – IWQoS 2005. XVIII, 400 pages. 2005.

Vol. 3551: T. Härder, W. Lehner (Eds.), Data Management in a Connected World. XIX, 371 pages. 2005.

Vol. 3548: K. Julisch, C. Kruegel (Eds.), Intrusion and Malware Detection and Vulnerability Assessment. X, 241 pages. 2005.

Vol. 3547: F. Bomarius, S. Komi-Sirviö (Eds.), Product Focused Software Process Improvement. XIII, 588 pages. 2005.

Vol. 3546: T. Kanade, A. Jain, N.K. Ratha (Eds.), Audio- and Video-Based Biometric Person Authentication. XX, 1134 pages. 2005.

Vol. 3544: T. Higashino (Ed.), Principles of Distributed Systems. XII, 460 pages. 2005.

Vol. 3543: L. Kutvonen, N. Alonistioti (Eds.), Distributed Applications and Interoperable Systems. XI, 235 pages. 2005.

¥566.40元

Table of Contents

Free-Form Surface Construction in a Commercial CAD/CAM System

Florian Albat and Rainer Müller

Tebis AG, Lademannbogen 128, D-22339 Hamburg, Germany
{Florian.Albat, Rainer.Mueller}@tebis.com

Abstract. In automobile industry, free-form surfaces often have to be constructed and even more often have to be modified. Frequently, a surface model is given as a triangular mesh, which is converted to (polynomial) spline surfaces (reverse engineering). We show some arising problems and how they are solved in our software. Furthermore, we present some open theoretical problems.

1 Surface Construction in CAD

There are two main types of surfaces in CAD: free-form surfaces and standard surfaces. The latter describe spatial objects, that can be defined uniquely by quite simple rules, often in terms of two-dimensional drawings. Typically, such objects consist of planes, cylinders and similarly simple surfaces. The sharp edges between the single surfaces are rounded by simulating a rolling ball (fillet surfaces). These models dominate in machine-building. The hood or roof of a car, however, is shaped in such a way, that it cannot be defined by easy rules like "cylinder of height 10cm with radius 3cm". Such surfaces are called free-form surfaces.

The automobile industry is not the only, but a very important field of industrial application for free-form surfaces. Many of our customers belong to this industry, both the car companies themselves and many suppliers of different kind. It is the main application field for surfaces of the highest quality requirements ('Class A', this implies e.g. curvature continuity). They are needed for the visible outer skin, that determines the potential buyer's impression of a car. But these high-quality surfaces represent only a minority of all constructed surfaces in a car, most of which lie invisible in the inside like the one shown in Fig. 1 and do not need to be of such a high quality. However, even such surfaces need to be G^1 continuous (within a tolerance of about $0.5°$), because they will finally be produced as sheet metals, and sheet metals cannot be shaped into sharp edges.

Besides sheet metals a car contains metal parts like the crank shaft, exhaust pipe and clutch shell plus plastic parts like the dashboard, which are molded. For all these parts CAD models must be constructed.

New objects account for only about 10% of construction work, roughly 90% of all constructions are modifications of existing models. So we will focus on such modifications. One reason for them is, that first versions of the fabrication

R. Martin, H. Bez, M. Sabin (Eds.): Mathematics of Surfaces 2005, LNCS 3604, pp. 1–13, 2005.

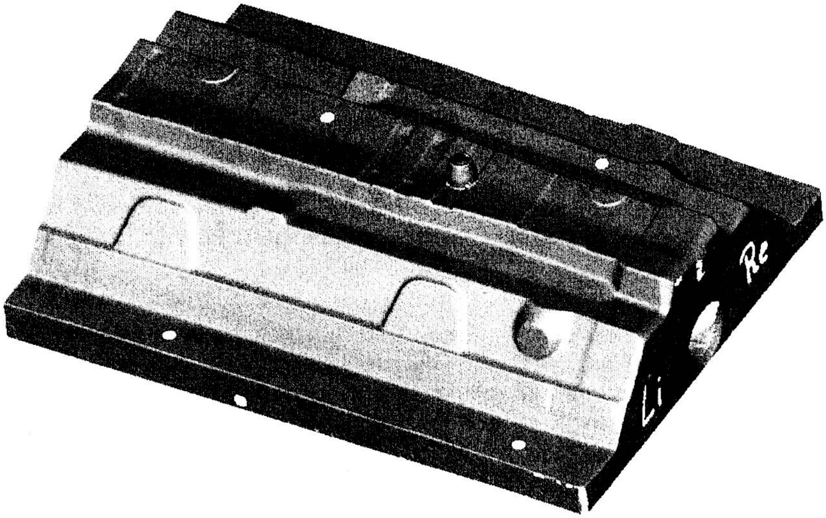

Fig. 1. Technical sheet metal from the inside of a car (scanned data)

tools are milled before the final design with all details is done. Another frequent reason is the compensation of a metal's springback.

Modifications can be either *real* or *virtual*. As real modifications we denote such, that are based on a really existing object. For example, the outer skin of a car is usually milled as a prototype, and the shape of this real model is modified by sanding off and spreading of material to satisfy certain aesthetic visions. This modified model is scanned, the scan points are triangulated and from this triangular mesh a new CAD model consisting of spline surfaces is constructed (reverse engineering). Virtual modifications we call such, that are performed directly on the CAD model without any real object being modified and scanned. A typical requirement is to raise a certain position of the roof for 1 cm. To do this, mostly neighbouring surfaces need to be adapted.

We will treat virtual modifications in the third section. Beforehand, we will elaborately treat real modifications, i.e. free-form surface construction by reverse engineering based on a triangular mesh. The question arises whether it is at all necessary to convert the mesh of scanned data into polynomial surfaces, or whether suitable new algorithms make it possible to work with meshes just as effectively. This question is interesting from a theoretical standpoint and not easily answered. However, it is (for the time being) irrelevant from a practical point of view, since industrial process chains are designed for work with 'smooth' surfaces. Changing to a continuous technique using meshes will not be possible until not only solutions for single steps exist, but really all design and production processes can be converted without suffering a loss in quality. Even then, the change will be done only when the savings gained outweigh the high cost of conversion. As a result, at least for the present and in the foreseeable future, it will be unavoidable in industrial applications to convert mesh data into polynomial surfaces. Hence, there is a need for easily useable software to do reverse engineering.

Among others, this paper shall also show, that our point of view as a provider of commercial software differs in some points from that of more theoretically interested scientists. We have to sell the used algorithms in such a way, that our customers, of whom hardly any has studied mathematics or computer science, can use them as easily as possible. Some of the discussed points may be trivial for academically educated readers, but these are questions, which must be considered if one wants to have success in the market.

This paper does not want to give an overview of the vast literature about possible algorithms or similar, but is aimed to inform about practical experiences of different software users, which is not written in common scientific papers. Hence, we restrict our list of references to [1], which is a good short introduction to reverse engineering. More references can be found therein.

2 Surface Construction with Reverse Engineering

2.1 Structuring the Object

Fig. 2 shows a triangular mesh of the front part of a car. Because of the symmetry, it suffices to construct the left half of the car. A surface construction of this model is shown in Fig. 3. This model looks good, matches the mesh very well and is G^1 continuous, which are very important properties of a surface model from reverse engineering. This model was constructed with our software package RSC (*Rapid Surface Creation*).

In Fig. 3 the border lines of the single surfaces are drawn. We denote their entirety as *wire frame*. Many curves follow the object's *feature lines*, each of which separates different 'uniform' areas, e.g. the relatively flat part of the hood from the more curved fillet. As we will delve into later, it is very important for the quality of the approximation surfaces, that the feature lines can be found in the wire frame. In a classical construction without mesh, the feature lines are naturally incorporated in the process: the larger uniform surfaces, the so called main surfaces, are constructed first. Then they are intersected with each other, and the sharp edges are rounded by fillets. The border lines of fillets are feature lines.

When constructing the fillets, the main surfaces are trimmed by the fillet boundaries. Likewise, the wire frame in Fig. 3, which originated from reverse engineering, contains many trimmed surfaces, i.e. surfaces, whose border is not a quadrangle of isoparametric lines. This is the normal case, one can hardly find CAD constructions, which do not contain trimmed surfaces. Exceptions are constructions, that are made by some reverse engineering programs, who cannot handle trimmed surfaces, but only *natural surfaces*, i.e. quadrangular surfaces, whose boundaries are isoparametric lines. Of course, every wire frame can be subdivided into quadrangles, but such a face layout is not suitable for some subsequent process steps. As an example, fig. 4 shows on the left a part of the

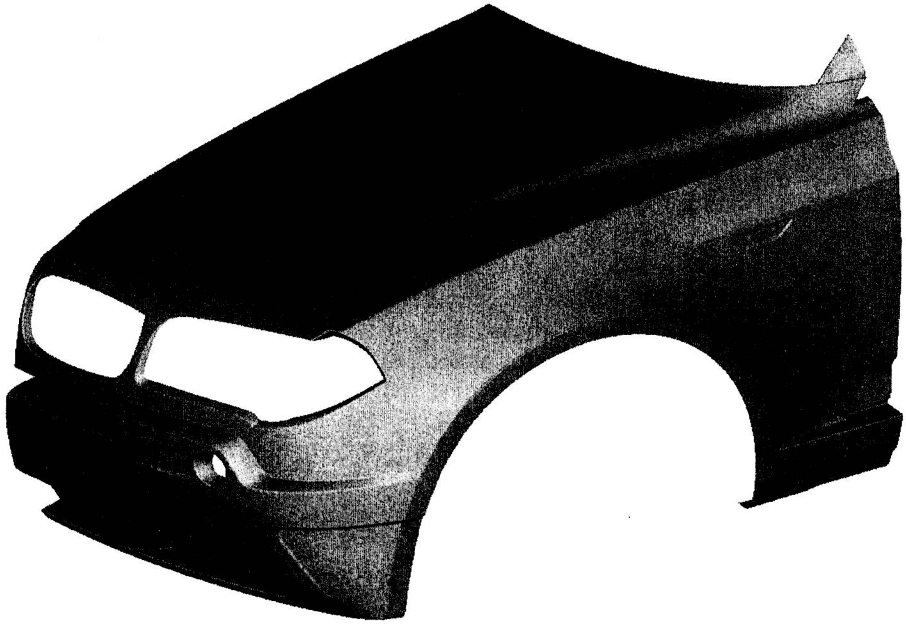

Fig. 2. Triangular mesh of a car

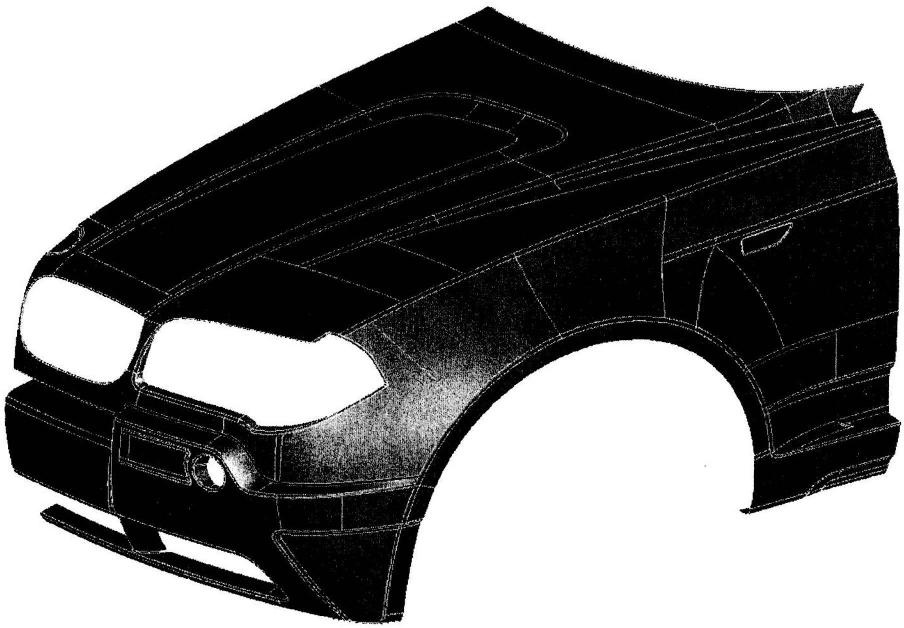

Fig. 3. Surface construction from the mesh in Fig. 2

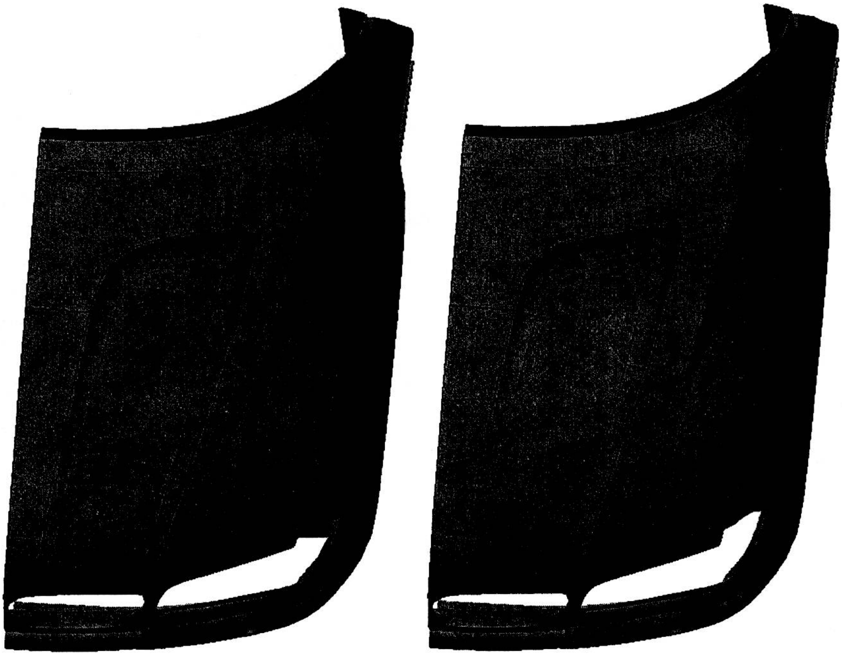

Fig. 4. Part of the wire frame from Fig. 3(left) and example for quadrangulation (right)

wire frame from fig. 3 and on the right a subdivision of this wire frame, so that it does only contain quadrangles.

The most obvious disadvantage of quadrangulation is the 'unnaturalness': the human user is disturbed by the many additional border lines. Another disadvantage is, that it is difficult to enlarge a main surface, when it was divided into several quadrangles. Such an enlargement is e.g. necessary, when a neighbouring approximated fillet surface has to be replaced by an exact fillet surface with bigger or smaller radius (reduction of radius).

Especially for the visible surfaces of a car (roof etc.) a surface calculated automatically by approximation of digitized data is hardly ever good enough for the high quality standards. Here is always manual postprocessing necessary, e.g. by manipulation of Bézier points. For that, a subdivision of the surfaces into quadrangles is completely unsuitable, one needs a face layout like in Fig. 3.

The designer's great dream is of course an algorithm, that is given the mesh and creates a wire frame like that in Fig. 3. We are working on this and are optimistic, that there will be considerable progress soon, but until now, such an algorithm does not exist. As a surface model implies a wire frame, no automatic procedure exists, that constructs a surface model like the one shown directly from the mesh. Sure, there are commercial programs for this task, but the programs known to us can only handle quadrangles and therefore have the mentioned disadvantages.

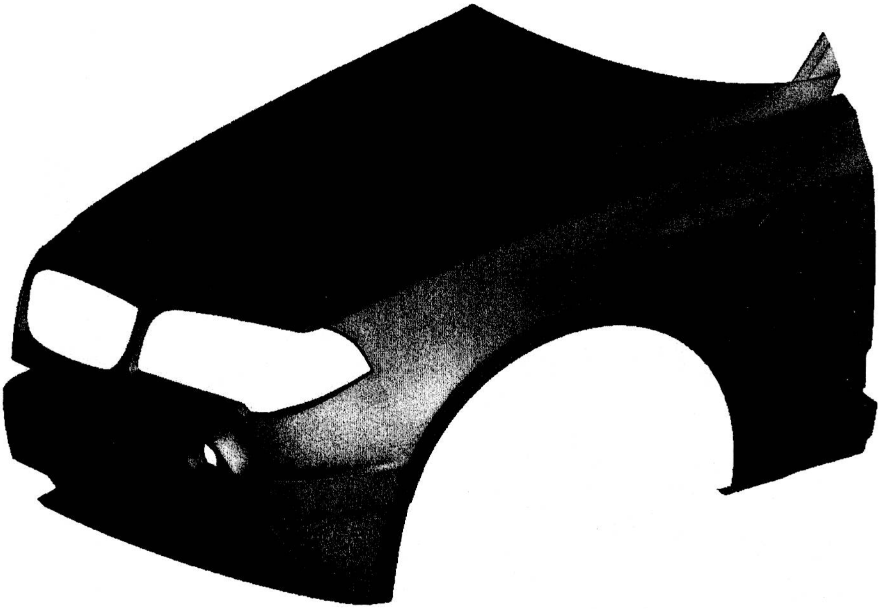

Fig. 5. Curvature representation of the mesh in Fig. 2

Until now, no program can create a wire frame nearly as well as the human designer, so the most reliable method, which is also implemented in our software, is to let the user construct the wire frame by himself. In general, it is difficult to construct spatial free-form curves, but in this application, it is much easier, as the curves are not absolutely free, but must lie on the mesh. By projecting the input automatically, our software enables the user to create curves directly on the mesh. This has proven to be both intuitive and reliable.

As support, we offer half-automatic functions for construction of feature lines, which e.g. recognize pair of fillet border curves or curves with constant (approximated) mesh curvature on positions selected by the user. For visualization of the object's structure it has proven useful to approximate the mesh's curvature and show it as a color spectrum, see Fig. 5.

2.2 Calculation of Surfaces

By construction of the wire frame the user decomposes the mesh into several facets, in which then surfaces can be calculated.

The surface, which is calculated for a facet, depends on the mesh and the facet's boundary curves, that define the approximated region. To achieve G^1 continuity, it also depends on surfaces in adjacent facets, if such exist. Letting the user specify all these parameters explicitly, is not only cumbersome, but also prone to error. A user-friendly program should ease this as far as possible. To do so, we use in our software a data structure 'RSC manifold', which contains besides geometric data also topological data, namely neighbourhood