Proceedings

computer graphics international 1997

Hasselt, Belgium
June 24–28, 1997

Sponsored by
CGS
EDM-LUC

IEEE Computer Society Press

THE INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, INC.

PROCEEDINGS

Computer Graphics International

June 23 – 27, 1997

Hasselt and Diepenbeek, Belgium

CGI'97 organized by

CGS (Computer Graphics Society), EDM (Expertise Centre for Digital Media)
LUC (Limburg University Centre)

In co-operation with

ACM SIGGRAPH, BCS (British Computer Society), Eurographics, IFIP WG 5.10
IEEE Computer Society Technical Committee on Computer Graphics, VRS (Virtual Reality Society)

With the support of

AGFA, ANDROME, BARCO, COMPUSOFT, CONCENTRA Group, European Commission — DG III
FORD Genk, FWO Vlaanderen, KINEPOLIS, SGI (Silicon Graphics International)
TANDEM, Touristic Service Limburg
Touristic Service of Hasselt, The Towns Hasselt and Diepenbeek

IEEE Computer Society Press
Los Alamitos, California

Washington • Brussels • Tokyo

IEEE Computer Society Press
10662 Los Vaqueros Circle
P.O. Box 3014
Los Alamitos, CA 90720-1314

IEEE Computer Society Press Order Number PR07825
ISBN 0-8186-7825-9
ISBN 0-8186-7826-7 (case)
ISBN 0-8186-7827-5 (microfiche)
IEEE Order Plan Catalog Number 97TB100104
Library of Congress Number 96-80274

Additional copies may be ordered from:

IEEE Computer Society Press
Customer Service Center
10662 Los Vaqueros Circle
P.O. Box 3014
Los Alamitos, CA 90720-1314
Tel: + 1-714-821-8380
Fax: + 1-714-821-4641
E-mail: cs.books@computer.org

IEEE Service Center
445 Hoes Lane
P.O. Box 1331
Piscataway, NJ 08855-1331
Tel: + 1-908-981-1393
Fax: + 1-908-981-9667
mis.custserv@computer.org

IEEE Computer Society
13, Avenue de l'Aquilon
B-1200 Brussels
BELGIUM
Tel: + 32-2-770-2198
Fax: + 32-2-770-8505
euro.ofc@computer.org

IEEE Computer Society
Ooshima Building
2-19-1 Minami-Aoyama
Minato-ku, Tokyo 107
JAPAN
Tel: + 81-3-3408-3118
Fax: + 81-3-3408-3553
tokyo.ofc@computer.org

Editorial production by Bob Werner

Cover design and art production by Joe Daigle/Studio Productions

Printed in the United States of America by Technical Communication Services

The Institute of Electrical and Electronics Engineers, Inc.

Preface

THESE PRINTED PROCEEDINGS CONTAIN the papers presented at the 15th Computer Graphics International conference, CGI'97, of the Computer Graphics Society. CGI'97 was held at Limburgs Universitair Centrum, Diepenbeek-Hasselt, Belgium, June 23-27, 1997. Previous conferences were held in Australia 1994, Japan 1983-1987 and 1992, Singapore 1990, Switzerland 1988 and 1993, the United Kingdom 1989 and 1995, the United States of America 1991. Future Computer Graphics conferences are planned in Germany (1998) and Canada (1999).

The truly international nature of these conferences is this year also reflected in the number of countries represented among the accepted papers, submitted from 24 different countries. A total of 31 contributions are included in the proceedings: one keynote presentation, 5 invited presentations and 25 reviewed papers, selected from a total of 47 submitted papers. Peer review of the submitted papers was undertaken by 70 reviewers; their invaluable time and expertise resulted in the selection of the 25 high-quality papers included in these proceedings of CGI'97.

The names of the cooperating societies, the sponsoring and supporting partners, as well as the committee members have been included in this volume. The much appreciated support and devotion of these organisations and individuals is gratefully acknowledged.

John Vince
Frank Van Reeth

Committee Members

Honorary Conference Chair: T.L. Kunii
Conference Co-Chairs: E. Flerackers, R. Earnshaw and D. Thalmann
Program Co- Chairs: F. Van Reeth and J. Vince

International Program Committee Members

Norman Badler, University of Pennsylvania, Philadelphia, PA, USA
Chandrajit L. Bajaj, Purdue University, West Lafayette, USA
Tat-Seng Chua, National University of Singapore
Michael F. Cohen, Microsoft Research, Redmond, WA, USA
Sabine Coquillart, INRIA-Rocquencourt, Le Chesnay, France
Rae A. Earnshaw, University of Bradford, West Yorkshire, Bradford, UK
Gershon Elber, Israel Institute of Technology, Israel
Jose L. Encarnação, Fraunhofer-Institut fuer Graphische Datenverarbeitung, Darmstadt, Germany
Gerald E. Farin, Arizona State University Tempe, AZ, USA
Olivier Faugeras, INRIA, Sophia-Antipolis, France
Steven K. Feiner, Columbia University, New York, NY, USA
Eddy Flerackers, Limburg University Center, Diepenbeek, Belgium
Charles Grimsdale, Division Ltd., North Carolina, USA
James K. Hahn, The George Washington University, Washington, DC, USA
Erik Jansen, Technische Universiteit Delft, The Netherlands
Arie E. Kaufman, State University of New York, Stony Brook, NY, USA
Myung-Soo Kim, Pohang University of Science and Technology, Pohang, South Korea
Stanislav V. Klimenko, Institute of High Energy Physics, Moscow, Russia
Tosiyasu L. Kunii, University of Aizu, Fukushima, Japan
Nadia Magnenat-Thalmann, University of Geneva, Switzerland
Nelson Max Lawrence, Livermore National Laboratory, Livermore, CA, USA
Eihachiro Nakamae, Hiroshima Institute of Technology, Hiroshima, Japan
Tomoyuki Nishita, Fukuyama University, Fukuyama, Japan
Nicholas M. Patrikalakis, MIT Cambridge, MA, USA
Yoshihisa Shinagawa, The University of Tokyo, Japan
Sung Yong Shin, KAIST Yusong-Ku, Taejon, Korea
Yasuhito Suenaga, NTT Human Interface Laboratories, Kanagawa-ken, Japan
Z. Tang, Tinghua University, China
Daniel Thalmann, Swiss Federal Institute of Technology, Lausanne, Switserland
Luiz Velho, IMPA Rio de Janeiro, Brasil
John Vince, School of Media Arts & Communication, Bournemouth University, Dorset, UK
Turner Whitted, Numerical Design Ltd, Chapel Hill, USA
George Wolberg, City College of New York, USA
Franz-Erich Wolter, Universität Hannover, Germany
Tony C. Woo, University of Washington, Seattle, USA
Brian Wyvill, University of Calgary, Alberta, Canada
Geoff Wyvill, University of Otago, Dunedin, New Zealand
Steve S.N. Yan, National Tsing Hua University, Taiwan
C. Yap, New York University, New York, USA
David Zeltzer, Zeltzer Sensory Communication Group, Cambridge, MA, USA
M.J. Zyda, Naval Postgraduate School, Monterey, CA, USA

Domestic Committee Members

Frans Arickx, University of Antwerp
Jan Broeckhove, University of Antwerp
Luc Claesen, IMEC Leuven
Bart Claeys, Kinepolis Hasselt
Karin Coninx, EDM-Limburg University Centre
Marcel De Coster, PIMC Philips Hasselt
An De Backer, Limburg University Centre
Luc Desimpelaere, BARCO Kortrijk
Theo D'Hondt, Free University, Brussels
Louis Ercken, EDM-Limburg University Centre
Eddy Flerackers, EDM-Limburg University Centre
Inge Jacobs, Limburg University Centre
Luc Smeets, VDAB Hasselt
Willy Smeulders, Philips Industrial Activities Hasselt
Frank Van Reeth, EDM-Limburg University Centre
Christophe Vermeulen, ALCATEL Antwerp
Yves Willems, Catholic University Leuven

Reviewers

Ken-ichi Anjyo
Frans Arickx
Norman I. Badler
Richard H. Bartels
Erik Behiels
Alexander Belyaev
Subhash Bhalla
Willem F. Bronsvoort
Tat-Seng Chua
Karin Coninx
Sabine Coquillart
Theo D'Hondt
Martin Duerst
Philip Dutre
Rae A. Earnshaw
Jose L. Encarnação
Wolfgang Felger
Eddy Flerackers
Kikuo Fujimura
Yukio Fukui
Jeff Ge
Hans Hagen
James K. Hahn
Werner Hansmann
Christoph M. Hoffmann
Masanori Idesawa
Hiroshi Imai
Hirofumi Ishida
Erik Jansen
Makoto Kato
Gerard Kim
Myung-Soo Kim
Tosiyasu L. Kunii
Tsuneya Kurihara
Wim Lamotte
Veronique Lang
Jintae Lee
Takashi Maekawa
Nadia Magnenat-Thalmann
Nelson Max
Stefan Muller
Masayuki Nakajima
Eichachiro Nakamae
Seiichi Nishihara
Tomoyuki Nishita
Jeff Norton
Seung-Woon Park
Nicolas M. Patrikalakis
Jorg Peters
Kaihuai Qin
Hassan Said
Muhammad Sarfraz
Vladimir V. Savchenco
Carlo H. Sequin
Evan C. Sherbrooke
Mikio Shinya
Yasuhito Suenaga
Tokiichiro Takahashi
Tiow-Seng Tan
Demitri Terzopoulos
Daniel Thalmann
Seamus T. Tuohy
Frank Van Reeth
John Vince
Turner Whitted
Yves Willems
Franz-Erich Wolter
Brian Wyvill
Geoff Wyvill

General Information

CGI'97 organized by

CGS (Computer Graphics Society)
EDM (Expertise Centre for Digital Media)
LUC (Limburg University Centre)

In co-operation with

ACM SIGGRAPH
BCS (British Computer Society)
Eurographics
IFIP WG 5.10
IEEE Computer Society Technical Committee on Computer Graphics
VRS (Virtual Reality Society)

With the support of

AGFA, ANDROME, BARCO, COMPUSOFT, CONCENTRA Group,
European Commission — DG III
FORD Genk, FWO Vlaanderen, KINEPOLIS, SGI (Silicon Graphics International),
TANDEM, Touristic Service Limburg
Touristic Service of Hasselt, The Towns Hasselt and Diepenbeek

Management Office

CGI'97 Secretariat
Expertise Centre for Digital Media
Limburg University Centre
Wetenschapspark 2
B-3590 Diepenbeek
e-mail: CGI97@luc.ac.be

Table of Contents

Computer Graphics International 1997 — CGI'97

Preface ix
Committee Members x
Reviewers xii
General Information xiii

Wednesday Paper Sessions

Welcome
H. Martens

Opening Session
E. Flerackers

Session 1: Keynote Lecture
Chair: T. Kunii

Enabling Superman's X-Ray Vision — Seeing is Believing
H. Fuchs

Session 2: Rendering
Chair: F. Van Reeth

An Algorithm for Polygon Subdivision Based on Vertex Normals 3
C. van Overveld, B. Wyvill

Calligraphic Character Synthesis using a Brush Model 13
H. Ip, H. Wong

Representing and Rendering Sweep Objects using Volume Models 22
G. Sealy, G. Wyvill

Session 3: Invited Lecture
Chair: T. D'Hondt

Autonomous Animated Interactive Characters: Do We Need Them? 29
B. Blumberg

Session 4: HCI in Virtual Environments
Chair: N. Magnenat-Thalmann

Integrating Flying and Fish Tank Metaphors with Cyclopean Scale 39
C. Ware, D. Fleet

A Hybrid 2D/3D User Interface for Immersive Object Modeling 47
K. Coninx, F. Van Reeth, E. Flerackers

Sketching Shadows and Highlights to Position Lights 56
P. Poulin, K. Ratib, M. Jacques

Session 5: Modeling
Chair: B. Blumberg

An Automatic Description of Volumetric Objects using Metaballs 65
J. Kim, E. Kim, S. Park

Reversibly Visible Polygons and Polygonal Approximation in Two Dimensional Space 74
S. Rao, K. Harada

VUEMS: A Virtual Urban Environment Modeling System 84
S. Donikian

Thursday Paper Sessions

Session 6: Invited Lecture
Chair: N. Patrikalakis

Characterizing Images Based on Lines for Image Indexing 94
T. Kunii, S. Murakami, Y. Shinagawa, M.Ohga

Session 7: Networked Virtual Environments
Chair: D. Thalmann

Model-Based View-Extrapolation for Interactive VR Web-Systems 104
D. Cohen-Or

DOOViE: An Architecture for Networked Virtual Environment Systems 113
J. Ahn, M. Lee, H. Lee

Session 8: Curves and Surfaces I
Chair: F. Wolter

Curve and Surface Design using Multiresolution Constraints 121
S. Takahashi, Y. Shinagawa, T. Kunii

An Interpolation Subspline Scheme Related to B-Spline Techniques 131
O. Röschel

Session 9: Invited Lecture
Chair: W. Lamotte

Internet Business Opportunities 138
C. Machover

Session 10: Panel on European Commission Research Projects in Computer Graphics

Session 11: Graphics Architectures
Chair: J. Vince

A Cached Frame Buffer System for Object-Space Parallel Processing Systems ____ 146
H. Kobayashi, T. Maeda, H. Yamauchi, T. Nakamura

An Advanced Graphics Chip with Bump-Mapped Phong Shading ____ 156
T. Ikedo, J. Ma

Session 12: Invited Lecture
Chair: G. Wyvill

A User-Friendly Texture-Fitting Methodology for Virtual Humans ____ 167
G. Sannier, N. Magnenat-Thalmann

Session 13: Simulation
Chair: R. Earnshaw

Controlling Fluid Animation ____ 178
N. Foster, D. Metaxas

Sensor Based Synthetic Actors in a Tennis Game Simulation ____ 189
H. Noser, D. Thalmann

Session 14: Virtual Reality Support Systems
Chair: E. Flerackers

The Virtual Display Case ____ 200
G. Wyvill, T. Mason, G. Downes, M. Williams

The Visorama System: A Functional Overview of a
New Virtual Reality Environment ____ 205
A. Matos, L. Velho, J. Gomes, A. Parente, H. Siffert

Distortion Correction of Magnetic Fields for Position Tracking ____ 213
G. Zachmann

Session 15: Invited Lecture
Chair: Y. Willems

VRML and Web Authoring
Silicon Graphics International (SGI) speaker

Session 16: Curves and Surfaces II
Chair: F. Arickx

Modifying Curvatures at Design Points for Convex B-spline Curves ____ 223
H. Guan, T. Torii

Geodesic Voronoi Diagrams on Parametric Surfaces ______ 230
R. Kunze, F. Wolter, T. Rausch

Rational Spline Interpolation Preserving the Shape of the Monotonic Data ______ 238
M. Sarfraz

Color Plates ______ 245

Index of Authors ______ 253

Wednesday Paper Sessions

Session 1

Keynote Lecture

Chair: T. Kunii

Session 2

Rendering

Chair: F. Van Reeth

An algorithm for polygon subdivision based on vertex normals

C.W.A.M. van Overveld
Department of Mathematics and Computing Science
Eindhoven University of Technology
Address: P.O.Box 513, 5600 MB, Eindhoven,
The Netherlands.
Email: wsinkvo@info.win.tue.nl

B.Wyvill
Department of Computer Science
University of Calgary
Address: 2500 University Drive N.W.,
Calgary, Alberta,Canada, T2N 1N4
Email: blob@cpsc.ucalgary.ca

Abstract

In order to achieve the impression of a smooth surface while rendering a polygon mesh, normal vector vectors may be provided in the vertices of the mesh that are the average of the surface normals of the adjacent polygons. Interpolation of these normal vectors while rendering of the polygons in the mesh, and using the interpolated normal vectors in the shading computations, yields a smoothly varying intensity distribution. There is an inherent mismatch, however, between the smoothness of the shading thus achieved and the non-smoothness of the geometry which is particularly visible at silhouettes, showing as straight edges and non-smooth edge junctions at the silhouette vertices. In this paper, a remedy for these artefacts is suggested. The remedy consists of subdividing each input polygon into a mesh of polygons prior to rendering. The shape of this resulting polygon mesh is controlled by the normal vectors that are provided in the vertices of the original polygon, unlike other subdivision schemes that make use of adjacent polygons. With our method, polygons equipped with vertex normal vectors can therefore be processed without further knowledge of neighbour polygons. This makes the method well-suited in the context of graphics libraries, such as OpenGl, that treat polygons typically on a per-polygon basis. So the proposed computation of the mesh which replaces the original polygon can be viewed as a filter which may operate as a process in front of a traditional polygon rendering pipeline.

Keywords geometric modelling, Bezier patches, subdivision, computer graphics

1. Introduction

Over the last years, efficient rendering algorithms, including hidden surface elimination and shading have found their way into the graphics work stations with the advent of sophisticated graphics hardware. This has emphasised the distinction between modelling and rendering: modelling takes place in the software domain, whereas rendering is taken care of by the underlying hardware. The interface between the two layers consists of a stream of data- and control signals. Several API's have been introduced recently, such as OpenGl, in order to standardise the format of the transfer of geometric modelling primitives. In particular, we will focus on polygons as modelling primitives in this paper: even though curved surfaces have been extensively used in geometric modelling, there is still a significant demand for visualisation of polygon models. Moreover, with the work of e.g. Mallet ([9]), Allan, Wyvill and Witten ([1]) and others, there even seems to be a revaluation for polygons in geometric modelling.

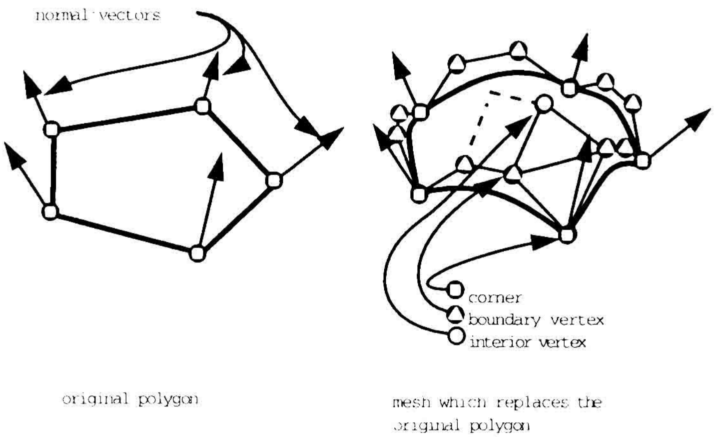
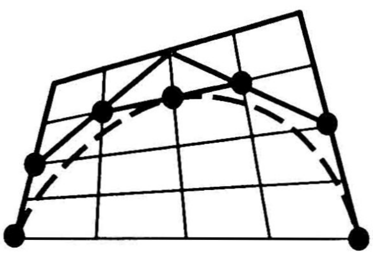


Figure 1. The original polygon; the resulting mesh; corners; boundary vertices; and interior vertices

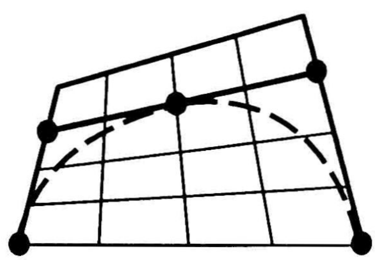
As far as geometry is concerned, the data that represents

0-8186-7825-9/97 $10.00 © 1997 IEEE

For color plate see page 246



A cubic Bezier segment
divided into two
cubic Bezier segments



A cubic Bezier segment replaced
by two quadratic Bezier segments with
equivalent boundary conditions

Figure 2. Reducing from a cubic to a quadratic subdivision scheme

polygons comprises vertex coordinates and vertex normal vectors. This data is passed through the graphics library API's in a pipe-lined fashion, i.e. each polygon is transformed, clipped and rendered in isolation without taking other polygons into account. In particular, the topological structure (neighbour relations) in the geometrical model is not represented in such API's[1], and hence not available, within the data structures of the rendering hardware. As a consequence, polygonal models will display straight silhouette edges and non-smooth edge junctions in silhouette vertices, even if their interior region is smoothly shaded. The most conspicuous aspect of straight silhouette edges are the non-G^1 edge junctions in the silhouette vertices.

Methods have been developed, such as the polygon subdivision algorithm by Catmull and Clark ([4]) that replace coarse polygon models by denser polygon meshes in order to remove straight silhouettes. However, these methods make use of the fact that for each polygon, its neighbours are available.

In this paper, we study the problem of removing straight-silhouette edges, and more in particular the non-G^1 edge junctions for geometric models that are communicated to the rendering hardware by means of a stream of isolated polygons. The only geometric information that is available, apart from the vertex coordinates, therefore, comprises the normal vectors in these vertices. Section 2 lists the requirements of the solution and reports on related work. A basic solution to the problem, which works for closed manifold polygon meshes, is presented in section 3. When the input polygons do not originate from a closed manifold mesh, but from a manifold mesh with one or more holes, a shortcoming of the method becomes visible, and we discuss a partial solution to this problem in section 4 (this solution is partial in the sense that it requires additional geometric information in the polygon vertices that should be provided by the application). Section 5 summarises the conclusions.

2. Requirements and conditions; Previous work.

An algorithm will be developed that converts a polygon, equipped with normal vectors in the vertices, into a triangle mesh. Assuming that the vertex normals of the input polygon are such that smooth shading of the original model would be achieved, the angles between adjacent triangles in the resulting mesh will be more obtuse than the angles between adjacent polygons in the original polygonal model.

[1] In some cases, regular neighbouhood structures such as triangle strips are supported for increased efficiency. In our study, however, we do not make any assumptions of the neighbourhood structure of the polygon to be rendered.

Therefore, the piecewise straight silhouettes in the resulting mesh will be less conspicuous. Also, the meshes of adjacent polygons will fit together in a C^0 fashion. The algorithm may operate recursively, by applying it again to this resulting mesh of triangles. A precondition for the algorithm is that normal vectors in shared edges of adjacent polygons are the same; this precondition is implied by conventional modelling techniques or post-modelling normal vector averaging, based on the neighbour relations between the polygons. The algorithm guarantees that the same condition holds for the output polygons.

Before listing the requirements and conditions for this algorithm, some nomenclature will be introduced (see also figure 1).

the polygon is the input polygon, i.e. a cyclic list of 3-D vertices with normal vectors (normalised). It is assumed that this polygon is part of a manifold polygon mesh, and for a given edge e in this mesh, the two polygons that share e have equal normal vectors in the two common vertices.

the mesh results from the polygon; the mesh vertices consist of the polygon vertices and additional new generated vertices. The locations of the new vertices are computed by taking the original vertices and normal vectors into account. The method for achieving this comprises the crucial aspect of the algorithm. In the sequel, polygons in the mesh will be called *sub-polygons*. After construction of the mesh, the sub-polygons will be triangulated.

the corners are those mesh vertices that equal the vertices of the polygon.

the boundary vertices are those mesh vertices that define, together with the corners, the boundaries of the mesh.

the interior vertices are the mesh vertices that are neither corners nor boundary vertices.

2.1. Requirements

The requirements that hold for a useful algorithm can be classified in three categories: architecture, functionality, and performance. Below we give these requirements; also, before explaining the algorithm in detail, some intuitive considerations are given as to how these requirements will be realised.

1. architecture: a pipeline-based, optional filter

   1.1 pipeline-based: the algorithm should deal with every polygon in isolation;

   1.2 optional filter: the algorithm should be a filter with congruent input- and output formats, i.e. both input and output should be a stream of polygons, equipped with normal vectors. This means that, e.g. for fast previewing, the filter can be passed by. Moreover it may be useful to have several instantiations of the filter in a row. This means that all sub-polygons that result after the first pass are replaced by sub-meshes of sub-sub-polygons after the second pass, etcetera.

2. functionality: continuity, consistency, affine invariance

   2.1 continuity: the filter's main purpose is to make the angles between adjacent triangles in the mesh more obtuse than the angles between the original polygons. This means:

   2.1.1 C^0 continuity: no cracks;

   2.1.2 G^0 normal vectors: no shading discontinuities;

   2.1.3 the shape of the mesh will be such that its periferal triangles will be (close) to perpendicular to the vertex normals of the original polygon, whereas the mesh will be 'smooth' in between (i.e. obtuse angles between adjacent triangles). Therefore, assuming that the normal vectors in the vertices of the input polygon resulted from averaging the surface normal vectors of this polygon and its neighbours, the meshes for adjacent input polygons will together form a 'smooth' mesh as well.

   2.2 consistency: the normal vector field in combination with the generated mesh is *consistent* with the geometry. This means the following. Linearly interpolating normal vectors over a polygon gives a shading pattern that suggests a monotonous curvature (either totally convex or totally concave) over this polygon. Such a monotonously curved surface, however, does not always exist (see e.g. figure 16.24 on page 741 in ([6]). Indeed, sometimesa curved surface that is perpendicular to the given vertex normals has to posses a curve of inflection (=the collection of points where the curvature changes sign). In this case, linear interpolation of normal vectors, suggesting a monotonous curvature is said to be *inconsistent*. The mesh generated by our algorithm, together with the vertex normals it provides, are such that each triangle represents either a totally convex or a totally concave surface fragment.

   2.3 affine invariance: the algorithm is transparent for linear transformations of the input vertices and normal vectors and for translations of the input vertices.

3. performance: generality, adaptiveness

   3.1 generality: the algorithm should work irrespective of the number of vertices in the input polygons. This means that the subdivision scheme used should be compatible both for triangles, quadrilaterals, and polygons with >4 sides, thereby taking into account continuity requirements as listed under 2.1.1-2.1.3,