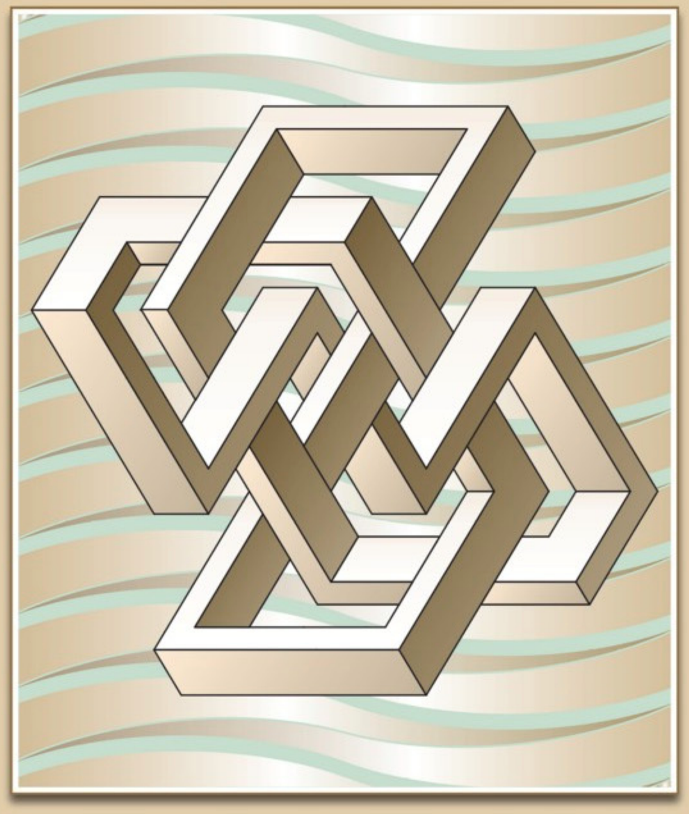
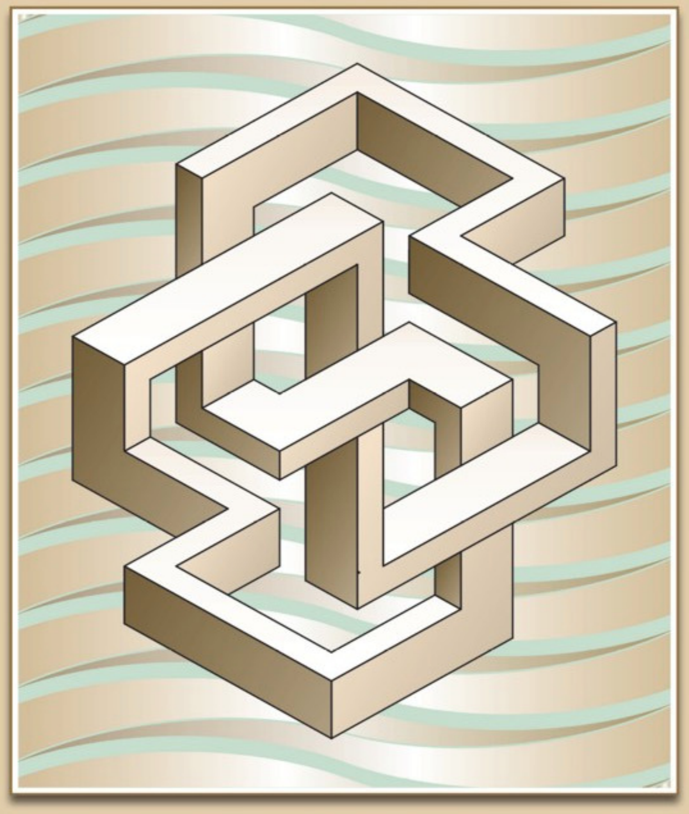
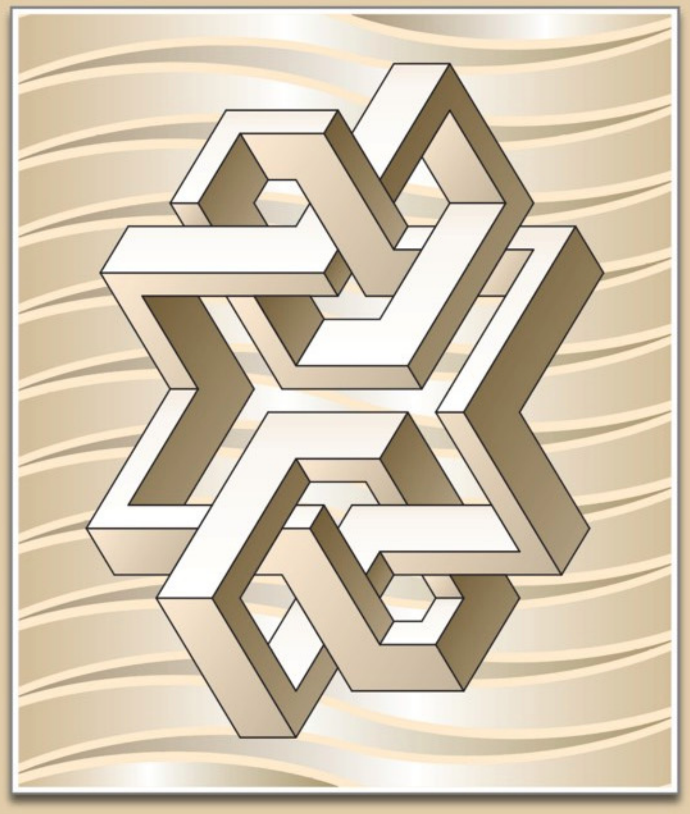
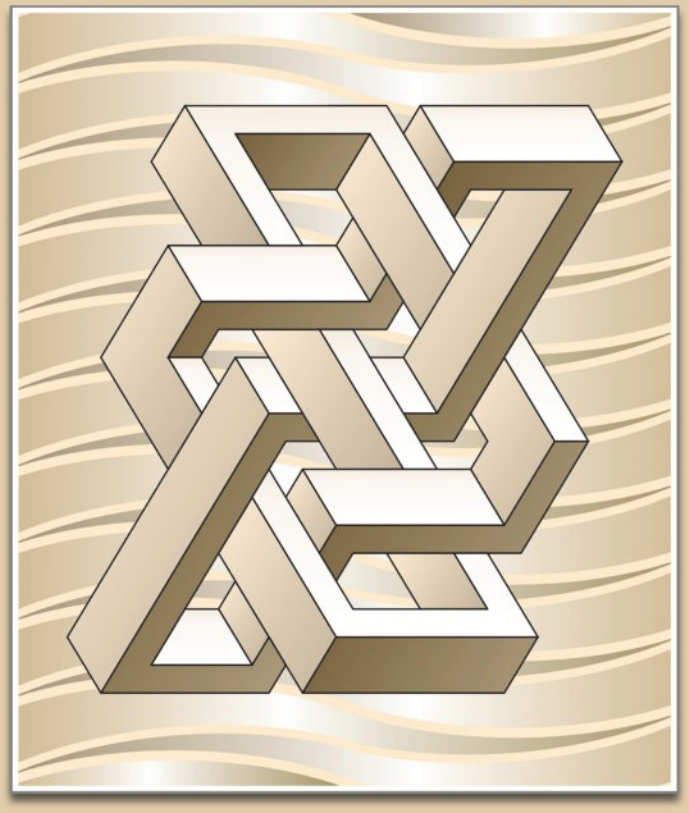
KoG

No. 10. 2006
ISSN 1331-1611

SCIENTIFIC-PROFESSIONAL JOURNAL OF
CROATIAN SOCIETY FOR GEOMETRY AND GRAPHICS

Official publication of the Croatian Society for Geometry and Graphics publishes scientific and professional papers from the fields of geometry, applied geometry and computer graphics.

Founder and Publisher

Croatian Society for Geometry and Graphics

Editors

SONJA GORJANC, Faculty of Civil Engineering, University of Zagreb, Croatia (Editor-in-Chief)
JELENA BEBAN-BRKIĆ, Faculty of Geodesy, University of Zagreb, Croatia
EMA JURKIN, Faculty of Mining, Geology and Petroleum Engineering, University of Zagreb, Croatia
MARIJA ŠIMIĆ, Faculty of Architecture, University of Zagreb, Croatia

Editorial Board

SONJA GORJANC, Faculty of Civil Engineering, University of Zagreb, Croatia
MILJENKO LAPAINE, Faculty of Geodesy, University of Zagreb, Croatia
EMIL MOLNÁR, Institute of Mathematics, Tehnical University of Budapest, Hungary
LIDIJA PLETENAC, Faculty of Civil Engineering, University of Rijeka, Croatia
HELLMUTH STACHEL, Institute of Geometry, Tehnical University of Vienna, Austria
NIKOLETA SUDETA, Faculty of Architecture, University of Zagreb, Croatia
VLASTA SZIROVICZA, Faculty of Civil Engineering, University of Zagreb, Croatia
VLASTA ŠČURIĆ - ČUDOVAN, Faculty of Geodesy, University of Zagreb, Croatia
GUNTER WEISS, Institute of Geometry, Tehnical University of Dresden, Germany

Design

Miroslav Ambruš-Kiš

Layout

Sonja Gorjanc, Ema Jurkin

Cover Illustration

Tamás F. Farkas, *Graphics* (see the last page)

Print

“O-TISAK”, d.o.o., Zagreb

URL address

http://www.hdgg.hr/kog
http://hrcak.srce.hr

Edition

250

Published annually

Guide for authors
Please see the page 2.

KoG is cited in: Mathematical Review, MathSci, Zentralblatt für Mathematik

---

This issue has been financially supported by The Ministry of Science, Education and Sport of the Republic of Croatia.

ISSN 1331–1611

No. 10

Zagreb, 2006

KoG

SCIENTIFIC AND PROFESSIONAL JOURNAL OF
CROATIAN SOCIETY FOR GEOMETRY AND GRAPHICS

CONTENTS

GUIDE FOR AUTHORS

SCOPE. "KoG" publishes scientific and professional papers from the fields of geometry, applied geometry and computer graphics.

SUBMISSION. Scientific papers submitted to this journal should be written in English or German, professional papers should be written in Croatian, English or German. Only unpublished material can be accepted.

Two single-side copies of the manuscript with wide margins and double spaced should be sent to the one of the editors.

Sonja Gorjanc	Jelena Beban - Brkić
sgorjanc@grad.hr	jbeban@geof.hr
Faculty of Civil Engineering	Faculty of Geodesy

Kačićeva 26, 10000 Zagreb, Croatia

The first page should contain the article title, author and coauthor names, affiliation, a short abstract in English, a list of keywords and the Mathematical subject classification.

ELECTRONIC FORMATS. Accepted papers should be sent by electronic mail as ASCII files (LaTeX format is recommended) to the address: sgorjanc@grad.hr

OFFPRINTS. The total of 20 reprints of each contribution will be sent to its first mentioned author (if not otherwise desired) free of charge.

UPUTE ZA AUTORE

PODRUČJE. "KoG" objavljuje znanstvene i stručne radove iz područja geometrije, primijenjene geometrije i kompjutorske grafike.

UPUTSTVA ZA PREDAJU RADA. Znanstveni radovi trebaju biti napisani na engleskom ili njemačkom jeziku, a stručni na hrvatskom, engleskom ili njemačkom. Rad treba biti neobjavljen.

Dva primjerka rukopisa sa širokim marginama i dvostrukim proredom treba poslati na adresu jedne od urednica:

Sonja Gorjanc	Jelena Beban - Brkić
sgorjanc@grad.hr	jbeban@geof.hr
Građevinski fakultet	Geodetski fakultet

Kačićeva 26, 10000 Zagreb

Prva stranica treba sadržavati naslov rada, podatke o autoru i koautorima, sažetak na hrvatskom i engleskom, ključne riječi i MSC broj.

ELEKTRONIČKI FORMATI. Prihvaćene radove autori dostavljaju elektronskom poštom kao ASCII datoteke (preporučuje se LaTeX format) na adresu: sgorjanc@grad.hr

POSEBNI OTISCI. Autori dobivaju 20 otisaka svog rada koji se šalju prvom koautoru, ukoliko nije dukčije dogovoreno.

How to get KoG?

The easiest way to get your copy of KoG is by contacting the editor's office:

Nikoleta Sudeta, nsudeta@arhitekt.hr
Faculty of Architecture
Kačićeva 26, 10 000 Zagreb, Croatia
Tel: (+385 1) 4639 219, Fax: (+385 1) 4828 079

The price of the issue is €15 + mailing expenses €5 for European countries and €10 for other parts of the world.

The amount is payable to:

ACCOUNT NAME: Hrvatsko društvo za geometriju i grafiku
Kačićeva 26, 10000 Zagreb, Croatia
IBAN: HR862360000-1101517436

Kako nabaviti KoG?

KoG je najjednostavnije nabaviti u uredništvu časopisa:

Nikoleta Sudeta, nsudeta@arhitekt.hr
Arhitektonski fakultet
Kačićeva 26, 10 000 Zagreb
Tel: (01) 4639 219, Fax: (01) 4828 079

Za Hrvatsku je cijena primjerka 100 KN + 10 KN za poštarinu.

Nakon uplate za:
HDGG (za KoG), Kačićeva 26, 10000 Zagreb
žiro račun broj **2360000-1101517436**
poslati ćemo časopis na Vašu adresu.

Ako Vas zanima tematika časopisa i rad našega društva, preporučamo Vam da postanete članom HDGG (godišnja članarina iznosi 100 KN). Za članove društva časopis je besplatan.

ISSN 1331–1611

KoG

BROJ 10

Zagreb, 2006

ZNANSTVENO-STRUČNI ČASOPIS
HRVATSKOG DRUŠTVA ZA GEOMETRIJU I GRAFIKU

SADRŽAJ

Review
Accepted 13.12.2006.

Vladimir Benić
Sonja Gorjanc

(1,n) Congruences

(1,n) Congruences

ABSTRACT

The first order algebraic congruences are classified into two basic classes which depend on their directing curves. By the method of synthetic geometry, we investigated the basic properties for each of these classes: the construction of rays, singularities, decomposition into developable surfaces, focal properties and the types of rays. The paper ends with a short analytical approach, which enables the visualizations of these congruences in the program *Mathematica*. Some exmples are shown.

Key words: congruence, decomposition on developable surfaces, focal lines, singularities, visualization

MSC 2000: 51M15,51M30,51N10,51N35,68U05

(1,n) kongruencije

SAŽETAK

Algebarske kongruencija prvoga reda razvrstane su u dvije osnovne klase, ovisno o njihovim ravnalicama. Za svaku od tih klasa, metodologijom sintetičke geometrije, istražuju se osnovna svjstva: konstrukcija zraka, singulariteti, dekompozicija na razvojne plohe, žarišne osobine te vrste zraka. Na kraju se daje i kratki analitički pristup koji je omogućio izradu programa za vizualizaciju ovih kongruencija u programu *Mathematica*. Pokazano je nekoliko primjera.

Ključne riječi: kongruencija, dekompozicija na razvojne plohe, žarišna linija, singulariteti, vizualizacija

1 Introduction

A congruence $\mathcal{C}$ is a double infinite line system, i.e. it is a set of lines in a three-dimensional space (projective, affine or Euclidean) depending on two parameters. A line $l \in \mathcal{C}$ is said to be a *ray* of the congruence.

The *order* of a congruence is the number of its rays which pass through an arbitraty point; the *class* of a congruence is the number of its rays which lie in an arbitrary plane. mth *order*, nth *class* congruence is signed $\mathcal{C}_n^m$.

A point is called the *singular point* of a congruence if ∞^1 rays pass through it. A plane is called the *singular plane* of a congruence if it contains ∞^1 rays.

Rays in a congruence can be decomposed in two ways into a one-parameter family of developable surfaces (torses) so that through every ray $l \in \mathcal{C}$ pass two torses that are real and different (the case of *hyperbolic* ray), or imaginary (an *elliptic* ray), or real and coincident (a *parabilic* ray).

The points of contact of a ray $l \in \mathcal{C}$ with the edges of regression (cuspidal edges) of these torses are called the *foci* of l. The foci of l are the intersection points of l with consecutive rays of a congruence. The surfaces formed by the foci of the rays of a congruence are called its *focal surfaces*. Each ray of a congruence touches its focal surface at the foci. Two planes defined as the planes containing a ray and consecutive rays are the *focal planes* of a congruence. The focal surface is the envelope of the focal planes. A congruence clearly reciprocates into a congruence. The focal planes and points are interchanged and the focal surface reciprocates into the new focal surface.
[11], [5], [9], [1]

Since the lines of the congruence are bitangents of the focal surface, every congruence of lines may be regarded as the system of bitangents of a surface. The surface may, however, break up into two separate surfaces, and the original surface, or each or either of the component surfaces may degenerate into a curve; we have thus as congruences the following systems of lines:

1. the bitangents of a surface,
2. common tangents of two surfaces,
3. tangents to a surface from the points of a curve,
4. common transversales of two curves,
5. lines “through two points” of a curve,

where the last four cases being degenerate cases of the first, which is the general one. [9, p.37]

2 1st order congruences

It was proved that the rays of the first order congruences are always tranversales of two curves, or they intersect the same space curve twice. Beside that it was proved that the only congruence of the first order, consisting of a system of lines meeting a proper curve twice, is when the curve is a twisted cubic. ([9, p. 64], [14, pp. 1184-1185], [13, p. 32])

If a congruence $\mathcal{C}$ is a system of lines meeting two directing curves of the orders m and m' which have α common points, the order of a congruence is $mm' - \alpha$. The only congruence of the first order of this kind is when the directing curves are a curve of the nth order and a straight line meeting it $n-1$ times. [9, p. 64]

Therefore, we have only two types of the congruences of the first order:

Type I 1st odrer nth class congruences $\mathcal{C}_n^1$ are the systems of lines which intersect a space curve c^n of the order n and a straight line d, where c^n and d have $n-1$ common points.

Type II 1st order 3rd class congruence $\mathcal{B}_3^1$ is the system of lines which meet a twisted cubic twice.

2.1 Congruences of the type I

2.1.1 Directing lines of $\mathcal{C}_n^1$

The directing lines of a congruence $\mathcal{C}_n^1$ are a space curve c^n of the order n and a straight line d which intersects c^n in $n-1$ points. If all intersection points are the regular points of c^n we will sign them $D_1^1, \ldots\ldots, D_{n-1}^1$. Some of these points can coincide. There are cases when the line d is the tangent line of c^n, the tangent at inflection, etc. If c^n and d have s-ple contact at one regular common point it is signed $D_i^{1,s}$, where $i \leq n-s$. (See Fig. 1)

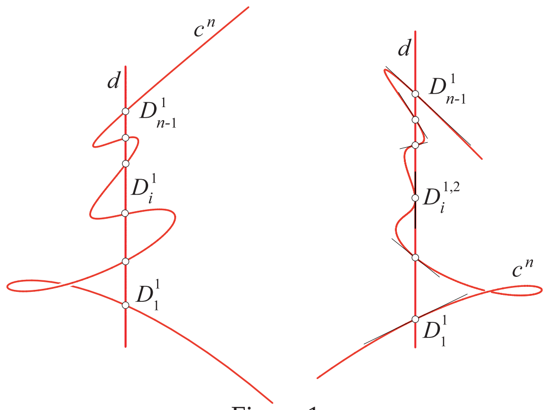


Figure 1

The nth order space curve can possess singular points with the highest multiplicity $n-2$. If the directing curve c^n has a multiple point it must lie on the line d because if c^n had a mutiple point out of the line d, the plane through that point and the line d would cut the curve c^n in more than n points, which is impossible. The k-ple point of the curve c^n which lies on the line d is signed D_i^k, where $i \leq n-k$. Three examples of 7th and 8th order curves with dobule and triple points are shown in Fig. 2.

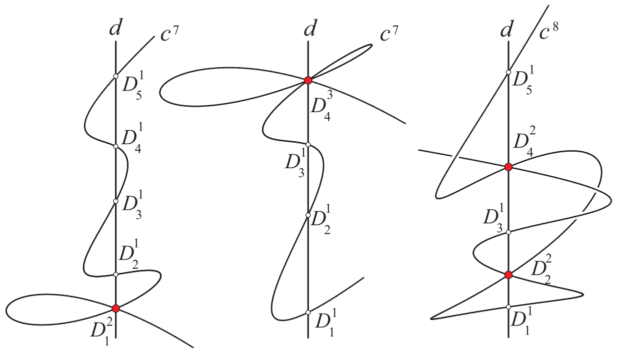


Figure 2

2.1.2 Singular points of $\mathcal{C}_n^1$

a) All singular points of $\mathcal{C}_n^1$ (the points which contain ∞^1 rays of $\mathcal{C}_n^1$) lie on its directing lines c^n and d.

If a point C lies on the curve c^n and $C \neq D_i^j$, then the rays of C_n^1 which pass trough C form the pencil of lines (C) in the plane $\delta \in [d]$ which contains C and d. (See Fig. 3)

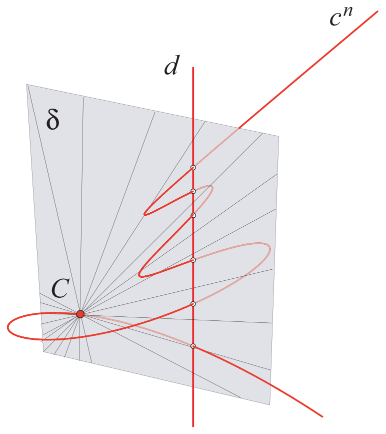


Figure 3

b) If a point D lies on the line d and $D \neq D_i^j$, then all the lines which join D with the points of the curve c^n are the rays of $\mathcal{C}_n^1$. They form nth degree cone ζ_D^n with the vertex D. Since c^n and d have $n-1$ common points, this cone intersects (or tuoches) itself $n-1$ times through the line d, thus the line d is $(n-1)$-ple generatrix of ζ_D^n. (See Fig. 4)

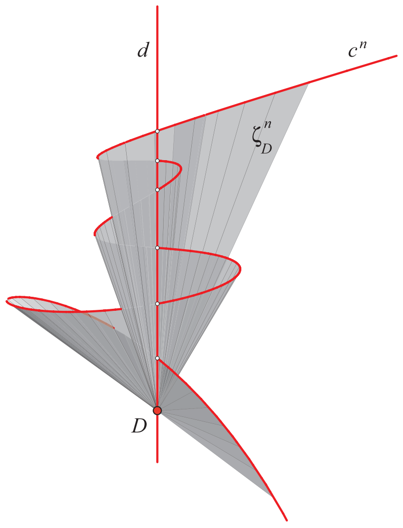


Figure 4

c) If a point D_i^k is the intersection point of c^n and d and if it is a k-ple point of the curve c^n, then the rays through D_i^k which cut c^n form $(n-k)$th degree cone $\zeta_{D_i^k}^{n-k}$ with the vertex D_i^k. The line d is $(n-k-1)$-ple genetartix of $\zeta_{D_i^k}^{n-k}$. Besides that the rays through the point D_i^k form k pencils of lines (D_i^k) in the planes determined by the line d and k tangent lines of c^n at D_i^k. If the intersection point is $D_i^{1,s}$, then the pencil of lines $(D_i^{1,s})$ in the rectifying plane of c^n at $D_i^{1,s}$ are also the rays of a congruence.

The other lines of the sheefs $\{D_i^k\}$ and $\{D_i^{1,s}\}$ are not regarded as the rays of a congruence.

The example of the rays through the regular intersection point D_1^1 is shown in Fig. 5.

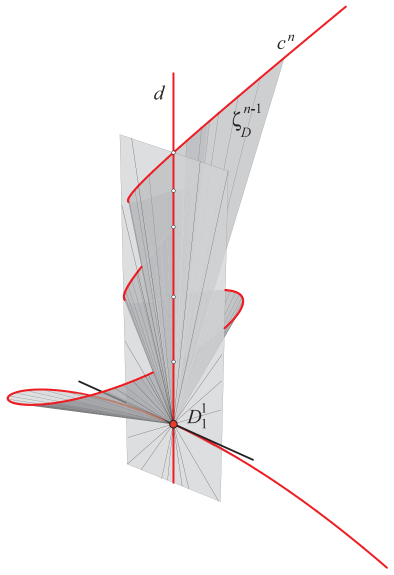


Figure 5

2.1.3 Rays of $\mathcal{C}_n^1$ through an arbitrary point

Every point A which is not the singular point of $\mathcal{C}_n^1$, i.e. $A \notin c^n, A \notin d$, determines the plane $\delta_A \in [d]$ which cuts c^n in only one point C which in general does not lie on the line d. The line AC, which cut d in one point D, is the unique ray of $\mathcal{C}_n^1$ through the point A. If the plane δ_A contains one of the the tangent lines of c^n at the intersection point D_i^k (or if it is the rectifying plane at $D_i^{1,s}$), then the points C and D cioncide with D_i^k $(D_i^{1,s})$ and the line AD_i^k $(AD_i^{1,s})$ is the unique ray of $\mathcal{C}_n^1$ through A. (See Fig. 6)

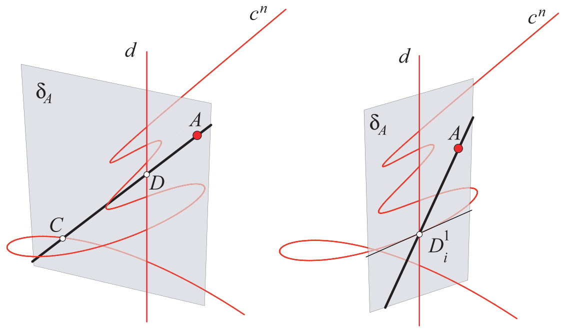


Figure 6

2.1.4 Singular planes of $\mathcal{C}_n^1$

All singular planes of $\mathcal{C}_n^1$ (the planes which contain ∞^1 rays of $\mathcal{C}_n^1$) are the planes of the pencil $[d]$. From 2.1.2. it is clear that in every plane $\delta \in [d]$ lie thes pencil of rays (C) or (D_i^k) or $(D_i^{1,s})$. (See Fig. 7)

It is possible that some of the tangent lines at intersection points D_j^k lie in the same plane of the pencil $[d]$. In such case there is more than one pencil of lines in the plane determined by these coplanar tangent lines.

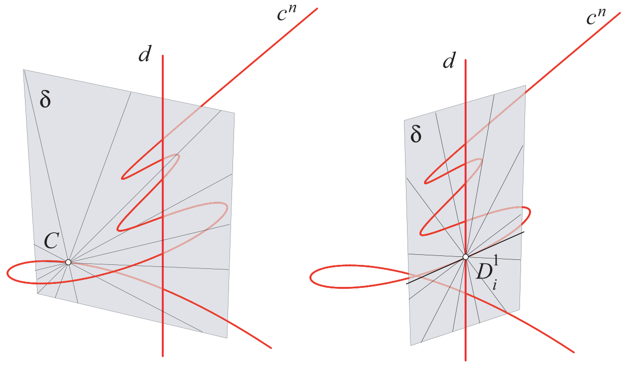


Figure 7

2.1.5 Rays of $\mathcal{C}_n^1$ through an arbitrary plane

Every plane α which is not the singular plane of $\mathcal{C}_n^1$, i.e. $\alpha \notin [d]$, contains n rays of the congruence. The plane α cuts line d in one point D and nth order space curve c^n in n points $C_j, j = 1, \ldots, n$. The lines DC_j are n rays of the congruence $\mathcal{C}_n^1$ in the plane α. They are the intersection of the plane α and nth degree cone ζ_D^n and can be real and different, coinciding or imaginary. If α cuts the line d in D_i^k, then $n-k$ rays are the intersection of α and the cone $\zeta_{D_i^k}^{n-k}$ and other k rays are the intersection of α and the planes through d determined by the tangent lines of c^n at D_i^k. (See Fig 8)

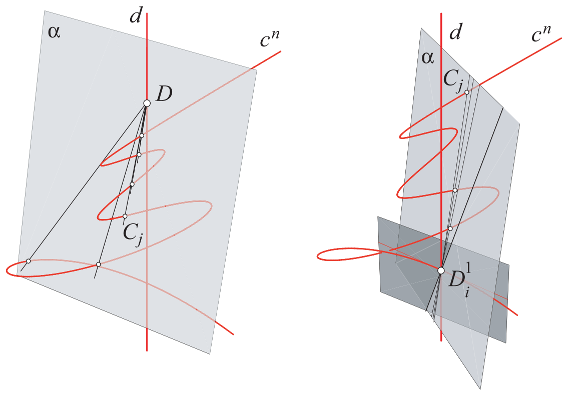


Figure 8

2.1.6 Decomposition of $\mathcal{C}_n^1$ into developable surfaces

As mentioned in the introduction every congruence can be decomposed in two ways into a one-parameter family of developables. These two families arise if one of the two parameters of which a congruence depends, is fixed.

In the case of the 1st order congruences of the type I the devlopables are the sets of rays through singular points, i.e. one family is formed by the nth degree cones ζ_D^n, and the other by the planes of the pencil $[d]$. Every ray of $\mathcal{C}_n^1$ is the intersection of two developables, one from each family. Since d is $(n-1)$-ple generatrix of ζ_D^n, every plane of $[d]$ cuts it into only one more generatrix which is the ray of a congruence. For the rays through the intercestion points D_i^k the nth degree cones split into $(n-k)$th degree cone and k planes through the line d. (See Figures 4, 5 and 7)

2.1.7 Focal properties – hyperbolic, elliptic and parabolic rays of $\mathcal{C}_n^1$

In general case the ray of a congruence touches the focal surface at foci which lie on the cuspidal edges of the developables. If the developables through the ray are real and different, the ray is hyperbolic and the foci are real and distinct. If the developables are imaginary, the ray is elliptic and the foci are imaginary. If the developables coincide, the ray is parabolic and the foci coincide. A congruence or the partition thereof is said to be hyperbolic, elliptic or parabolic if its rays are hyperbolic, elliptic or parabolic.

In the case of the 1st order surfaces of the type I the focal surface degenerates into the directing curves c^n and d. The developables have not the cuspidal edges, only cuspidal points: the verteces D of the cones ζ_D^n and the points C which are the intersections of the planes $\delta \in [d]$ and the curve c^n. Thus, each ray of $\mathcal{C}_n^1$ has the foci on the directing lines c^n and d. If they are real, the congruence is hyperbolic with parabolic rays in $n-1$ planes through the points D_i^k where the developables and foci coincide. If the directing lines are imaginary the congruence is elliptic.

2.2 Congruences of the type II

If a congruence $\mathcal{B}$ is the set of lines which cut a proper curve twice, this curve must be a twisted cubic. Since through an arbitrary point only one ray of the 1st order congruence passes, the projection of the directing curve from this point onto an arbitrary plane has only one double point. Thus this projection is the 3rd order plane curve. As the original curve and its projection have the same order, then the directing line of a congruence $\mathcal{B}$ is a twisted cubic. The projection of a twisted cubic onto a plane from a point on a secant line yields a nodal cubic and from a point on a tangent line a cuspidal cubic [4, p. 54]. (See Fig. 9)

The tangent and a secant lines of a twisted cubic b^3 fill up the projective space and are pairwise disjoint, except at points at curve itself [4, p. 90]. Thus through an arbitrary point unique ray of the congruence $\mathcal{B}$ passes.

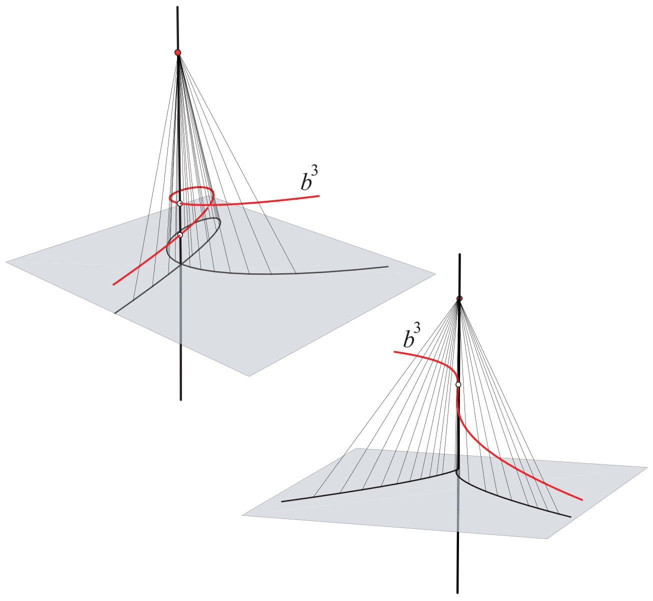


Figure 9

It is clear that such congruence is of the 3rd class ($\mathcal{B}_3^1$), because every plane cuts the curve b^3 in 3 points and the lines joining them are three rays of a congruence. These three rays can be: three real and different (a), one real and two imaginary (b), three real where two of them coincide (c) and three real and coinciding (d). (See Fig. 10)

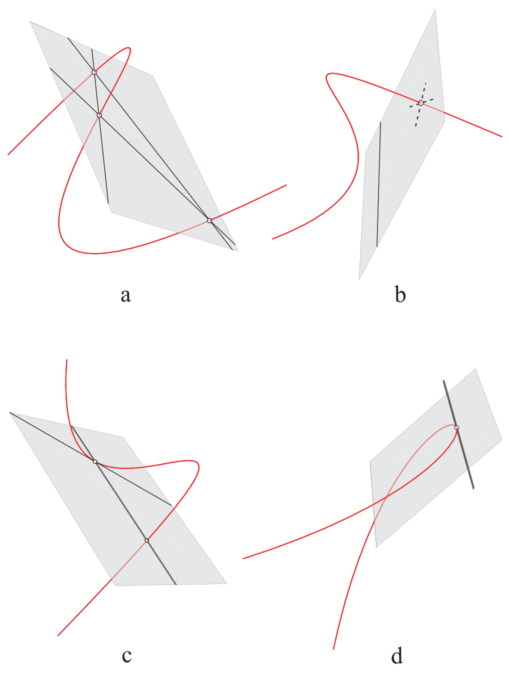


Figure 10

All singular points of $\mathcal{B}_3^1$ lie on the twisted cubic b^3. The lines which join the point $B \in b^3$ with the other points of b^3 form 2nd degree cone ζ_B^2. (See Fig. 11)

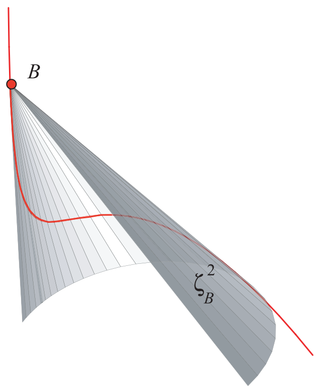


Figure 11

Since every plane contains exactly three rays of $\mathcal{B}_3^1$ there are no singular planes of $\mathcal{B}_3^1$.

$\mathcal{B}_3^1$ can be decomposed into one family of developables. This family consists of the cones ζ_B^2, $B \in b^3$. Every ray of $\mathcal{B}_3^1$ which cuts b^3 at the points B_1, B_2 is the part of the intersecion of the cones $\zeta_{B_1}^2$, $\zeta_{B_2}^2$. Namely, the intersection of $\zeta_{B_1}^2$ and $\zeta_{B_2}^2$ is $b^3 \cup B_1B_2$.

The curve b^3 is the focal curve of $\mathcal{B}_3^1$, it contains the cuspidal points of ζ_B^3, $B \in b^3$. The ray of $\mathcal{B}_3^1$ is hyperbolic if the intersection points $B_1, B_2 \in b^3$ are real and different, it is parabolic if they coincide (the ray is a tangent of b^3) and it is elliptic if they are imaginary.

The parabolic rays of $\mathcal{B}_3^1$ form the tangent developable of b^3. (See Fig. 12)

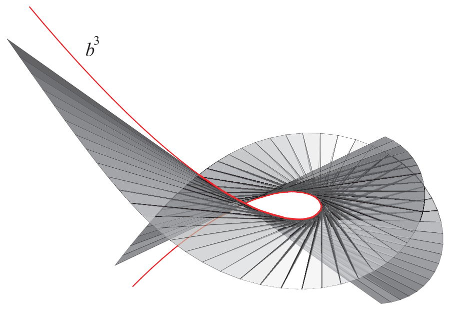


Figure 12

The rays of such congruences are also the intersections of the corresponding elements of two collinear bundles of planes $\{B_1\}$, $\{B_2\}$, [7, p. 135], [14, p. 1185]. In this case the basic points of the bundles (B_1, B_2) lie on a twisted cubic b^3, and the unique ray through an arbitrary point can be constructed as the part of the intersection of two ruled quadrics. These quadrics pass through b^3 and their rulings are determined by the collineation between the bundles $\{B_1\}$ and $\{B_2\}$, [7, p. 136]. In the special case when one plane in the collineation between $\{B_1\}$ and $\{B_2\}$ corresponds to itself, the basic cubic b^3 splits into one straight line and a conic which have one common point, and the congruence $\mathcal{B}_3^1$ splits into the 2nd class congruence $\mathcal{C}_2^1$ and the field of lines in the plane of the conic. These congruences are elaborated in detail in [2].

3 Analytical approach and *Mathematica* visualizations

If two algebraic space curves c_1 and c_2 are given by the following parametric equations

$$\begin{aligned} &c_1 \ldots x = x_1(u), \quad y = y_1(u), \quad z = z_1(u),\\ &\qquad x_1, y_1, z_1 : I_1 \to \mathbb{R},\ I_1 \subseteq \mathbb{R},\ x_1, y_1, z_1 \in C^1(I_1)\\ &c_2 \ldots x = x_2(v), \quad y = y_2(v), \quad z = z_2(v),\\ &\qquad x_2, y_2, z_2 : I_2 \to \mathbb{R},\ I_2 \subseteq \mathbb{R},\ x_2, y_2, z_2 \in C^1(I_2), \qquad (1)\end{aligned}$$

then the set of lines which join the points of c_1 and c_2 are given by the following equations

$$\frac{x-x_1(u)}{x_1(u)-x_2(v)}=\frac{y-y_1(u)}{y_1(u)-y_2(v)}=\frac{z-z_1(u)}{z_1(u)-z_2(v)}, \qquad (2)$$
$$(u,v)\in I_1\times I_2\subseteq\mathbb{R}^2.$$

In the previous section, for drawing the directing lines of $\mathcal{C}_n^1$, we used the following parametric functions:

$$\begin{aligned} d\ldots x_1(u)&=0,\quad y_1(u)=0,\quad z_1(u)=u,\quad u\in\mathbb{R},\\ c^n\ldots x_2(v)&=a_x(v-v_1)\cdots(v-v_{n-1}),\\ y_2(v)&=a_y v x_2(v),\\ z_2(v)&=v,\qquad v,a_x,a_y,v_1,\ldots,v_{n-1}\in\mathbb{R}. \end{aligned} \qquad (3)$$

It is clear that the line d is the axis z and c^n is the nth order space curve which cuts the axis z at the points $D_i(0,0,v_i)$, $i\in\{1,...,n-1\}$.

If the polynomial $x_2(v)$, from (3), contains the factor $(v-v_i)^s$, then $i\le n-s$ and d and c^n have $s-$ple contact at the point $D_i(0,0,v_i)$.

If the polynomial $z_2(v)$, from (3), takes the form

$$z_2(v)=v(v-v_{i_1})\cdots(v-v_{i_k}),\ v_{i_j}\neq 0, \qquad (4)$$
$$i_1,\ldots,i_k\in\{1,\ldots,n-1\},\quad k\le n-2,$$

then $(0,0,0)$ is the $k-$ple singular point of c^n and the coordinates of intersection points of c^n and d are $(0,0,z_2(v_i))$.

(3) and (2) give the following equations of the rays of $\mathcal{C}_n^1$

$$\frac{x}{x_2(v)}=\frac{y}{y_2(v)}=\frac{u-z}{u-z_2(v)},\qquad (u,v)\in\mathbb{R}^2. \qquad (5)$$

The above equations enable computer visualization of the rays of $\mathcal{C}_n^1$ and $\mathcal{B}_3^1$. Based on this we made the program in *webMathematica* which enables interactive visualizations of $\mathcal{C}_n^1$ on the internet. It is available at the following address:
www.grad.hr/itproject_math/Links/webmath/indexeng.html

3.1 Examples

In the following examples the graphics are produced with the program *Mathematica*.

EXAMPLE 1

Two dsiplays of the same 2nd class congruence are shown in Fig. 13. The directing lines of this $\mathcal{C}_2^1$ are the axis z and the circle given by the following parametric equations:

$$x(v)=\cos v+1,\ y(v)=\sin v,\ z(v)=0,\ v\in[0,2\pi].$$

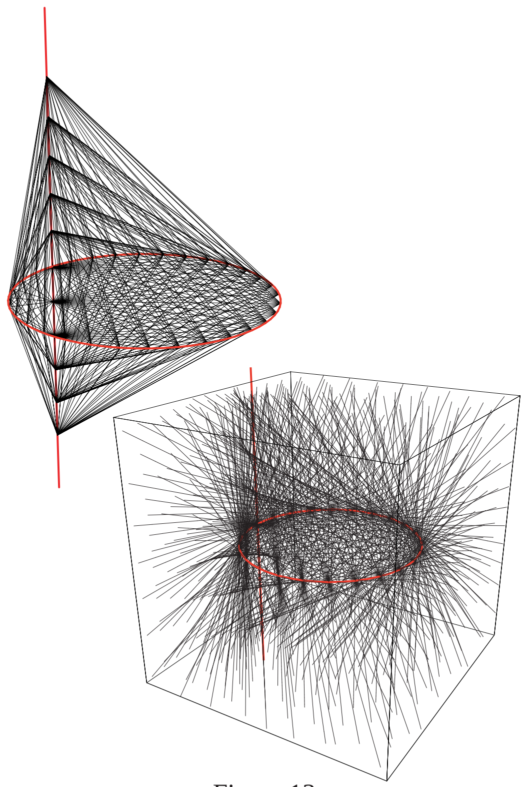


Figure 13

EXAMPLE 2

The rays of the 4th class congruence are shown in Fig. 14. The directing lines of this congruence are the Viviani's curve c^4 which cuts the axis z in two points, but one of the intersection points is the double point of c^4. The parametric equations of c^4 are:

$$x(v)=\frac{\sqrt{2}}{2}(\cos v-2\sin\frac{v}{2}-1),\quad y(v)=\sin v,$$
$$z(v)=\frac{\sqrt{2}}{2}(\cos v+2\sin\frac{v}{2}-1),\quad v\in[-2\pi,2\pi].$$

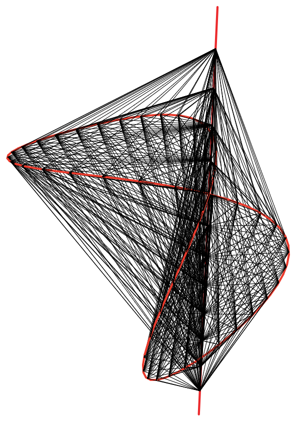


Figure 14

EXAMPLE 3

Two displays of the same 7th class congrence are shown in Fig. 15. The directing lines of this $\mathcal{C}_7^1$ are the axis z and the curve c^7 which is given by the following parametric equations:

$$x(v) = \frac{1}{10}v(v-1)(v-2)(v-3)(v-3.5)(v-4),$$
$$y(v) = 2vx_2(v),$$
$$z(v) = v, \quad v \in \mathbb{R}.$$

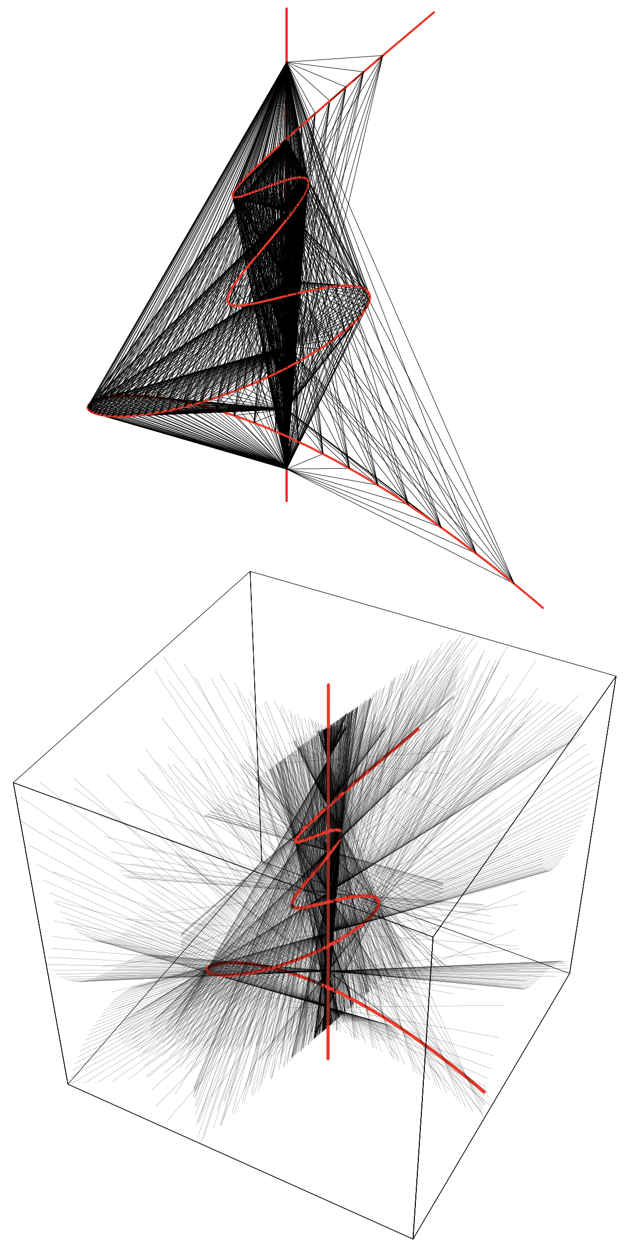


Figure 15

EXAMPLE 4

The visualization of $\mathcal{B}_3^1$ whose directing curve is given by the following parametric equations

$$x(v) = v,$$
$$y(v) = (v-1)(v+1),$$
$$z(v) = (v-1)^2(v+1), \quad v \in \mathbb{R}$$

is shown in Fig. 16. The same congruence, with red parabolic rays, is shown in Fig. 17 for two different view points.

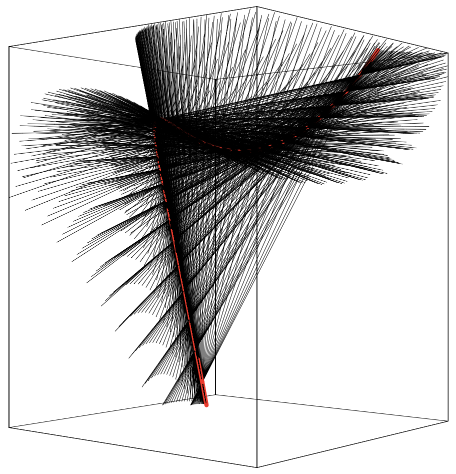


Figure 16

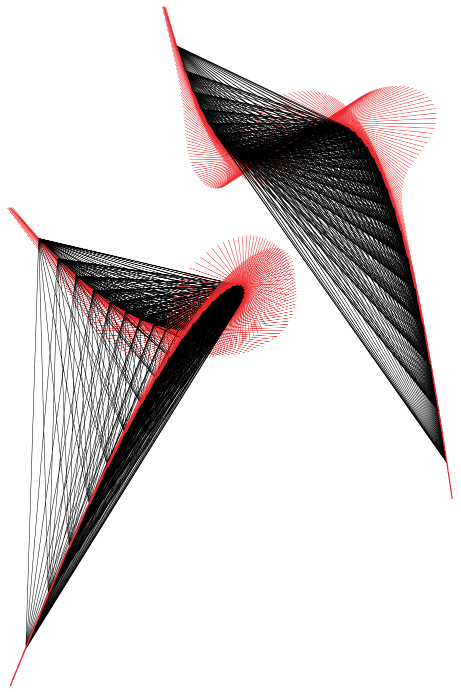


Figure 17

References

[1] S. P. FINIKOV: *Theorie der Kongruenzen*. Akademie-Verlag-Berlin, Berlin, 1959. (translation from Russian)

[2] S. GORJANC: *The Classification of the Pedal Surfaces of (1,2) Congruences*. Dissertation, Department of Mathematics, University of Zagreb, 2000. (in Croatian)

[3] A. GRAY: *Modern Differential Geometry of Curves and Surfaces with Mathematica*. CRC Press, Boca Raton, 1998.

[4] J. HARRIS: *Algebraic Geometry: A First Course* Springer, New York, 1992.

[5] C. M. JESSOP: *A Treatise on the Line Geometry*. Chelsea Publishing Company, New York 1969 (reprint).

[6] E. MÜLLER, J. L. KRAMES: *Konstruktive Behandlung der Regelflächen*. Franc Deuticke, Leipzig und Wien, 1931.

[7] V. NIČE: *Introduction to Synthetic Geometry*. Školska knjiga, Zagreb, 1956. (in Croatian)

[8] G. SALMON: *A Treatise on the Analytic Geometry of Three Dimensions, Vol.I* . Chelsea Publishing Company, New York, 1958. (reprint)

[9] G. SALMON: *A Treatise on the Analytic Geometry of Three Dimensions, Vol.II* . Chelsea Publishing Company, New York, 1965. (reprint)

[10] G. SALMON: *Higher Plane Curves*. Chelsea Publishing Company, New York, 1960. (reprint)

[11] *Encyclopaedia of Mathematics - SpringerLink*. Edited by M. Hazewinkel, available online: `http://eom.springer.de/C/c024910.htm`

[12] R. STURM: *Die Lehre von den geometrischen Verwandtschaften, Band IV*. B. G. Taubner, Leipzig-Berlin, 1909.

[13] R. STURM: *Liniengeometrie, II. Teil*. B. G. Taubner, Leipzig, 1893.

[14] K. ZINDLER: *Algebraische Liniengeometrie*. Encyklopädie der Mathematischen Wissenschaften, Band III, 2. Teil, 2. Hälfte. A., pp. 1184-1185, B. G. Teubner, Liepzig, 1921-1928.

Vladimir Benić
e-mail: benic@grad.hr
Sonja Gorjanc
e-mail: sgorjanc@grad.hr

Faculty of Civil Engineering University of Zagreb
Kačićeva 26, 10000 Zagreb

Original scientific paper
Accepted 1. 12. 2006.

MÁRTA SZILVÁSI-NAGY

About Curvatures on Triangle Meshes

About Curvatures on Triangle Meshes

ABSTRACT

A face-based curvature estimation on triangle meshes is presented in this paper. A flexible disk is laid on the mesh around a given triangle. Such a bent disk is used as a geodesic neighborhood of the face for approximating normal and principal curvatures. The radius of the disk is free input data in the algorithm. Its influence on the curvature values and the stability of estimated principal directions are investigated in the examples.

Key words: triangle mesh, curvature

MSC 2000: 68U05, 68U07, 65D20

O zakrivljenostima na trokutnim mrežama

SAŽETAK

U članku je prikazana procjena zakrivljenosti na trokutnim mrežama, bazirana na stranicama. Gipki disk položen je na mrežu oko danog trokuta. Takav prilagodljiv disk koristi se kao geodetska okolina stranice za aproksimaciju normalnih i glavnih zakrivljenosti. Polumjer diska je nezavisni ulazni podatak u algoritmu. U primjerima se istražuje njegov utjecaj na vrijednosti zakrivljenosti i na stabilnost procijenjenih glavnih smjerova.

Ključne riječi: trokutna mreža, zakrivljenost

1 Introduction

Triangle meshes are the most frequently used surface representations in many surface-oriented applications. Surface curvature properties have been successfully employed for solution of different practical problems, as smoothing or simplifying meshes in modeling and manufacturing, also for surface classification and 3D object recognition in computer vision research, etc. Discrete counterparts of continuous definitions of differential operators, curvature values, geodesic curves and Dirichlet energy, etc. have been given and derived for arbitrary triangle meshes in [3], [10], [13] and [14].

Almost all methods for surface derivative and curvature estimations on meshes have been vertex connectivity based. In these approaches a specified neighborhood of vertices formed by adjacent vertices, edges and faces is used to approximate the surface normal, surface derivatives and curvature values at a vertex. The algorithms use either analytic methods based on surface fitting, or they work with discrete differential operators. The crucial first step in these algorithms is the computation of a vector at each vertex that approximates the true normal vector at this point of the surface represented by the mesh. This problem is equivalent to the computation of the best tangent plane to the mesh at a given vertex. Most methods compute a weighted average of facet normals in a one-ring neighborhood of the vertex.

$$N = \frac{\sum \omega_j N_j}{m_i}, \quad j = 1 \dots m_i,$$

where m_i is the number of edges emanating from the vertex v_i, $\omega_j = k \cdot Area(triangle_j)$, $k > 0$ and $Area(triangle_j)$ is either surface area or Voronoi-surface area or a mixed surface area of the triangle $\{v_i, v_j, v_{j+1}\}$ (Fig. 1).

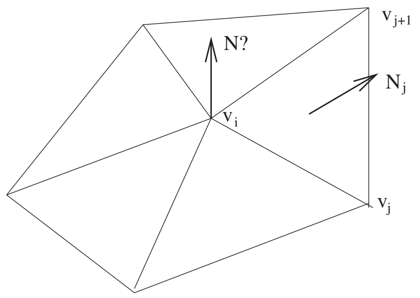


Figure 1: *One-ring neighborhood of a vertex*

A number of proposals have been published for choosing the weights or the neighborhood for determining the best normal vector ([15], [11], [9]).

An other problem is to estimate normal curvatures, which is equivalent to the definition of osculating circles producing a second-order approximation to those curvature values ([12], [19]).

A normal curvature estimation in a one-ring neighborhood can be given simply by defining the osculating circle in a normal plane through the vertices v_i, v_j and the normal vector N (Fig. 2).

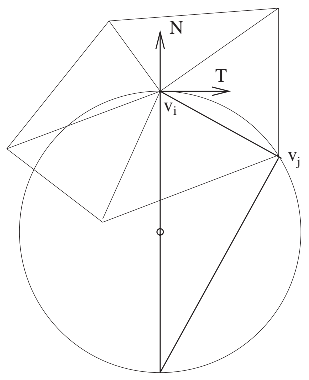


Figure 2: *Osculating circle in a normal plane*

The curvature of this circle is

$$\kappa_n(v_i) \approx \frac{2 < N, (v_j - v_i) >}{|v_j - v_i|^2},$$

where $<,>$ denotes the dot product.

Instead of circle in a normal plane, interpolating quadratic polynomial curve is also used e.g. in [7]. The one-ring neighborhood of a specified vertex is replaced by a Voronoi or mixed surface area [10], it is extended in [7] and geodesic neighborhoods are also used ([12], [16]) in the computations. The selection of neighborhood size can affect results significantly: small neighborhoods provide better estimates for clean data, while increasing the neighborhood size smoothes the estimates, leading to less sensitivity to noise. Obviously, small errors in these approximations lead rapidly to unreliable, noisy curvature values. Comparisons of five frequently used methods are given in [5].

Analytic methods are also applied for curvature estimations by fitting a surface to the mesh in the neighborhood of the point of interest and evaluating its curvatures ([5], [6], [19]). Principal curvatures and principal directions can be determined on the base of Euler theorem ([11], [2], [4]). Mean and Gauss curvature values can be computed from the curvature tensor, i.e. from the Weingarten matrix or from its symmetric extension by eigen-decomposition ([12], [18]). The Gauss-Bonnet theorem gives a direct method for the computation of Gauss curvature. It has two different discrete forms which provide good approximations for special triangulations of surfaces [20].

In this paper we define normal curvatures on each face of the triangle mesh in order to estimate the principal directions and to characterize elliptical, umbilical, flat and hyperbolic regions. The proposed new method is presented in Chapter 2. In the examples (Chapter 3) we show the proposed method on ‘ synthetic’ and real triangulated surfaces.

2 Curvatures defined on faces

2.1 Geodesic circle of a triangle

Instead of computing surface properties at vertices in vertex neighborhoods we define curvature values ordered to faces. The center of the defined region is the barycentric center of the given triangle. We intersect the mesh with normal planes passing through the face normal of the triangle, then we measure a given geodesic radius along the polygonal lines of intersection in both directions from the center point. In this way we get a number of curved diameters of the geosedic circle bent on the mesh around the face. We call this geodesic neighborhood “splat” after Kobbelt [8] and the set of the polygonal diameters “spider” after Simari [16] (Fig. 3). Then we compute the chord lengths of the constructed diameters in order to estimate normal curvature values.

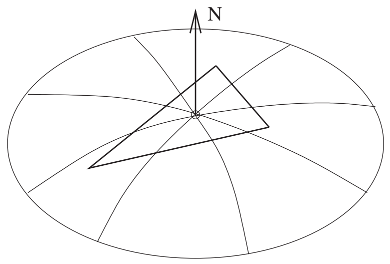


Figure3: *Geodesic circle of a face*

2.2 Osculating circle and normal curvature

We define in each normal plane an osculating circle to the face determined by the endpoints of the bent diameter and the face normal (Fig. 4).

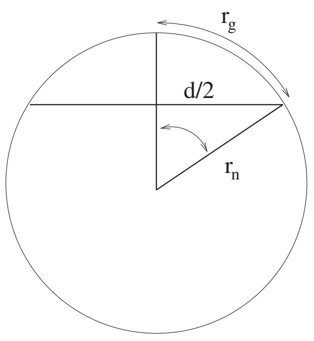


Figure 4: *Osculating circle in a normal section*

Denote r_g the given geodesic radius, d the chord length between the endpoints of the curved diameter, 2α the unknown central angle and r_n the required radius of the osculating circle. From the equations

$$r_n\alpha = r_g \qquad \text{and} \qquad r_n \sin\alpha = \frac{d}{2}$$

we get for α

$$\frac{d}{2r_g} = \frac{\sin\alpha}{\alpha}.$$

We apply the approximation

$$\sin\alpha \approx \alpha - \frac{\alpha^3}{6}, \quad 0 < \alpha << 1$$

and get

$$\alpha \approx \sqrt{(1-\frac{d}{2r_g})6},$$

consequently,

$$r_n \approx \frac{r_g}{\alpha} \qquad \text{if} \qquad \alpha \neq 0 \qquad \text{and} \qquad \kappa_n \approx \frac{\alpha}{r_g}$$

is the radius of the osculating circle and the normal curvature, respectively.

2.3 Principal directions and principal curvatures

Repeating this computation for a set of normal sections in the geodesic neighborhood of the given triangle we obtain normal curvature values $\kappa_{n,i}$, $i = 1,\ldots k$. If the mesh is a dense triangulation of a regular surface then the normal planes belonging to the minimal and maximal normal curvatures are orthogonal to each other. They determine the principal directions.

We select the maximal curvature and define it as first principal curvature κ_1 and the corresponding direction T_1 in the plane of the current triangle as first principal direction. This direction is fairly stable, even if we compute with smaller geodesic circles. The second principal direction T_2 is orthogonal to it. In the case of properly defined geodesic circle and nearly regular triangulation it is the direction belonging to the maximal chord length (Fig. 5).

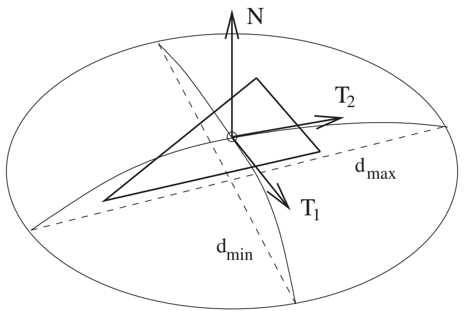


Figure 5: *Principal directions*

3 Examples

In order to compute a geodesic disk around a triangle face in a mesh we have to construct a flexible polyhedral data structure on the mesh which differentiates inner and boundary edges, moreover "feature edges" along sharp ridges. Then we have to implement an algorithm for computing the lines of intersection of the mesh and the defined normal planes [17]. The normal sections on the mesh are polygonal lines, and the arc length on the approximated surface is measured along these polygonal lines.

The triangulated cylindrical meshes in Fig. 6 and 7 are generated from the analytical description of a cylindrical surface. The vertices of such a "synthetic" mesh are lying exactly on the surface approximated by the mesh. The geodesic radius in Fig 6 is 3.5 times the average size of the triangles. The number of the computed diameters is 24. In Fig 7 the same cylindrical surface is shown with different triangulation. The geodesic radius is 0.6 times the average size of the triangles in the mesh. The largest curvature computed by the proposed method are 0.0402 and 0.0405, respectively, instead of 0.0400. Hence the relative errors in the curvature estimation are 0.5% and 1.2%, respectively. The corresponding principal directions are in both cases the most perpendicular ones to the axis of the cylinder among the 24 tangent directions. These directions are shown in the figures with longer segments.

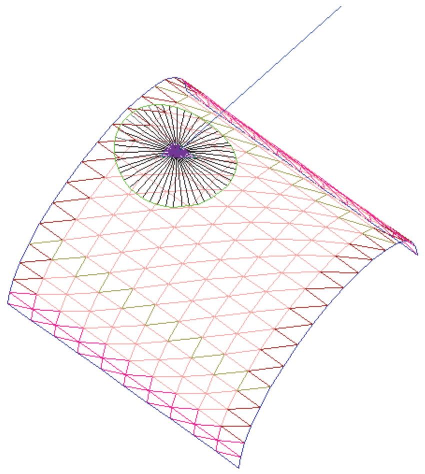


Figure 6: *Splat on a cylindrical surface*

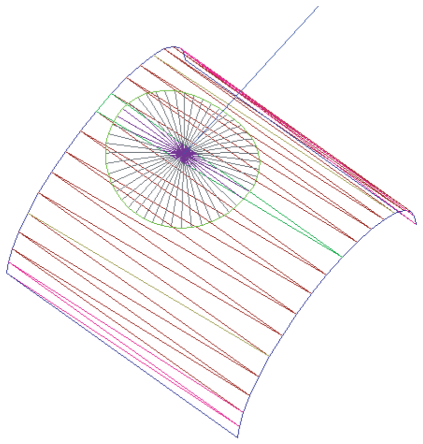


Figure 7: *The same cylindrical surface*

The second example shows that our curvature estimation works also in the case, where one-ring neighborhoods cannot be applied.

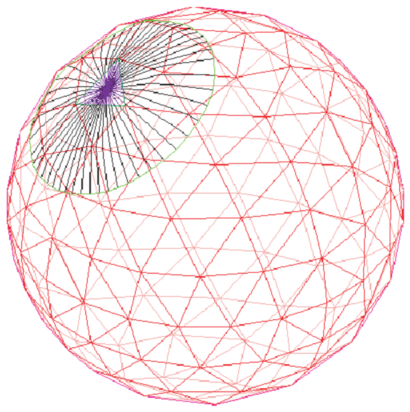


Figure 8: *Splat on the sphere*

In Fig. 8 a real triangle mesh approximating a sphere is shown. In this mesh there is a noise in the vertex data, and in the size of the triangles. The chosen geodesic radius is three times the average triangle size. The difference in the computed normal curvatures is about 3%, and this relative error does not change when the geodesic radius is varying between 2.5 and 7 times the average triangle size. This accuracy in the computation is comparable to the published results in the literature analysing curvature estimation methods [5].

In Fig. 9 and 10 a synthetic mesh of one eighth of a torus is shown. In Fig 9 the geodesic radius is three times the average triangle size, and the results in the principal directions are very good, considering the relative coarse approximation with 24 directions in the disk. In Fig 10 the geodesic disk is apparently too big to get reliable approximations (5 times the average triangle size) for the specified triangle face. In this picture only boundary and silhouette edges are shown besides the 24 diameters of the geodesic disk.

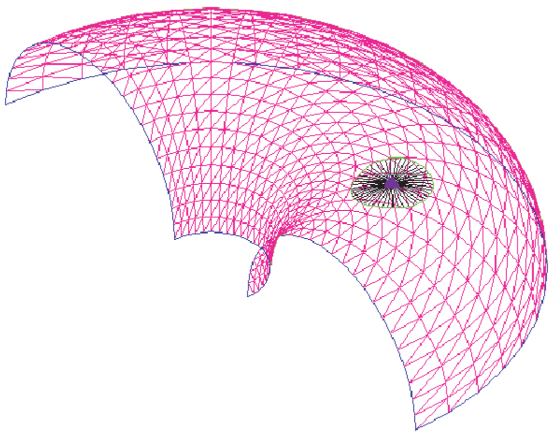


Figure 9: *Splat on the torus*

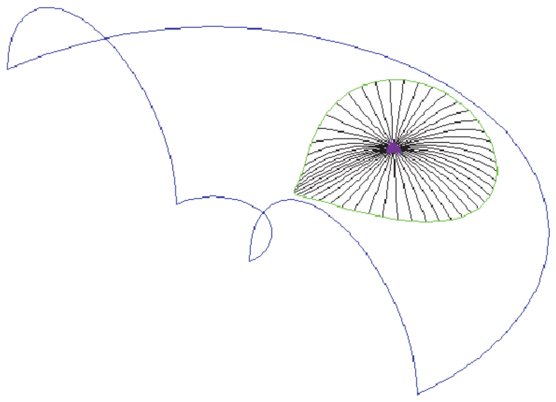


Figure 10: *The same facette with a bigger splat*

According to our experience the best measurement of the radius of a geodesic disk is between 2.5 and 3.5 times the average triangle size in a dense mesh. Of course, the mesh should provide a proper surface approximation. The examples have shown that our face-based curvature estimation works better than vertex-oriented methods using a one-ring neighborhood, especially in the case when the mesh contains long narrow triangles.

Some more investigations have to be done in the future, e.g., in the analysis of relative errors and in further applications.

4 Conclusion

We have introduced curvature values ordered to faces in triangle meshes by laying a flexible circular disk with user-specified radius onto each face of the mesh. From the chords of such a bent disk and from the face normal we have defined normal curvatures of the current face. The examples have shown that the obtained principal curvature values and the corresponding principal directions are quite reliable if the radius of the disk achieves an optimal size. Our method provides a good classification of elliptic, parabolic, flat and hyperbolic regions of the mesh.

We have implemented the algorithm for constructing geodesic disks on triangle meshes and for estimating normal curvatures and principal directions in Java.

References

[1] DO CARMO, M.P., *Differential Geometry of Curves and Surfaces*, Prentice-Hall, Englewood Cliffs, NJ., 1976.

[2] CHEN, X., SCHMITT, F., *Intrinsic surface properties from surface triangulation*, Proceedings of the European Conference on Computer Vision, 1992, pp.739-743.

[3] DESBRUN, M., MEYER, M., SCHRÖDER, P., BARR, A.H., *Implicit fairing of irregular meshes using diffusion and curvature flow*, Computer Graphics Proceedings (SIGGRAPH'99), 1999, pp.317-324.

[4] DONG, C.S, WANG, G.Z., *Curvatures estimation on triangular mesh*, Journal of Zhejiang University SCIENCE 6ASuppl.I (2006) 128-136.

[5] FLYNN, P.J., JAIN, A.K., *On reliable curvature estimation*,IEEE Conf. Comput. Vision Patt. Recogn. 1989, pp. 110-116.

[6] GOLDFEATHER, J., INTERRANTE, V., *A novel cubic-order algorithm for approximating principal direction vectors*, ACM Transactions on Graphics 23/1 (2004), 45-63.

[7] HAMEIRI, E., SHIMSHONI, I., *Estmating the principal curvatures and the Darboux frame from real 3D range data*, Computers & Graphics 28 (2004), 801-814.

[8] KOBBELT, L., BOTSCH, M., *A survey of point-based techniques in computer graphics*, Computers & Graphics 28 (2004), 801-814.

[9] MAX, N., *Weights for computing vertex normals from facet normals*, Journal of Graphics Tools 4 (2000), 1-6.

[10] MEYER, M., DESBRUN, M., SCHRÖDER, P., BARR, A.H., *Discrete differential-geometry operators for triangulated 2-manifolds*, Visualization and Mathematics III. (H.C. Hege and K. Polthier, Ed.) Springer Verlag 2003, pp. 35-57.

[11] MORETON, H.P., SÉQUIN, C.H., *Functional optimization for fair surface design*, SIGGRAPH'92 Conference Proceedings 1992, pp. 167-176.

[12] PAGE, D.L., SUN, Y., KOSHAN, A.F., PAIK, J., ABIDI, M.A., *Normal vector voting: Crease detection and curvature estimation on large, noisy meshes*, Graphical models 64 (2002), 199-229.

[13] PINKALL, U., POLTHIER, K., *Computing discrete minimal surfaces and their conjugates*, Experimental Math. 2 (1993), 15-36.

[14] POLTHIER, K., SCHMIES, M., *Straightest geodesics on polyhedral surfaces*, Mathematical visualization (K. Polthier and H.C. Hege Ed.) pp.391-409, Springer Verlag 1998.

[15] RUSINKIEWICZ, S., *Estimating curvatures and their derivatives on triangle meshes*, Proc. 3DPVT'04 2004, 486-493.

[16] SIMARI, P., SINGH, K., PEDERSEN, H., SPIDER, *A rpbust curvature estimator for noisy, irregular meshes*, Tecnical report CSRG-531, Dynamic Graphics Project, Dept. of Comp.Sci., Univ. Toronto, 2005.

[17] SZILVÁSI-NAGY, M., *Removing errors from triangle meshes by slicing*, Third Hungarian Conference on Computer Graphics and Geometry, 17-18. Nov. 2005 Budapest, Hungary, 125-127.

[18] TAUBIN, G., *Estimating the tensor of curvature of a surface from a polyhedral approximation*, ICCV'95 Proceedings of the Fifth International Conference on Computer Vision, IEEE Computer Society, Washington DC, USA, 902.

[19] TODD, P.H., MCLEOD, R.J.Y., *Numerical estimation of the curvature of surfaces*, computer-aided design 18 (1986) 33-37.

[20] XU, G., *Convergence analysis of a discretization scheme for Gaussian curvature over triangular surfaces*,Computer Aided Geometric Design 23 (2006) 193-207.

Márta Szilvási-Nagy
e-mail: szilvasi@math.bme.hu

Department of Geometry
Budapest University of Technology and Economics
H-1521 Budapest, Hungary

Supported by a joint project between the TU Berlin and the BUTE, by the Hungarian National Foundation OTKA No. T047276 and by the Hungarian-Croatian Intergovernmental S & T Cooperation Programme

Original scientific paper
Accepted 6. 12. 2006.

LÁSZLÓ VÖRÖS

Two- and Three-dimensional Tilings Based on a Model of the Six-dimensional Cube

Two- and Three-dimensional Tilings Based on a Model of the Six-dimensional Cube

ABSTRACT

A central-symmetric three-dimensional model of the six-dimensional cube can give us the idea of filling the space with mosaics of zonotopes. This model yields also plane tilings by its intersections. Using the parts of the model the mosaic and the tiling can further be dissected by projections, associations and Boolean operations. Further constructions are also indicated in the paper.

Key words: 3-dimensional models of the hypercube, plane-tiling, space-filling

MSC 2000: 51M20, 68U07

2-dimenzionalno i 3-dimenzionalno popločavanje zasnovano na modelu 6-dimenzionalne kocke

SAŽETAK

Centralno simetrični 3-dimenzionalni model 6-dimenzionalne kocke može nam dati ideju kako prostor ispuniti s mozaicima zonotopa. Pomoću presjeka, ovaj model vodi također i ka ravninskom popločavanju. Koristeći dijelove modela, mozaik i popločavanje mogu biti razdijeljeni projekcijama, asocijacijama i Boolovim operacijama. U članku se također navode i daljnje konstrukcije.

Ključne riječi: 3-dimenzionalni model hiperkocke, ravninsko popločavanje, prostorno popunjavanje

Lifting the vertices of a *k* sided regular polygon from their plane perpendicularly by the same height and joining them with the centre of the polygon, we get the *k* edges of the hypercube (*k*-cube) modelled in the three-dimensional space (3-model). From these the 3-models or their polyhedral surface (Fig. 1) can be generated as well in different procedures [3, 4, 5]. Each polyhedron from these will be a so called zonotope [6], i.e. a "translational sum" (Minkowski-sum) of some segments.

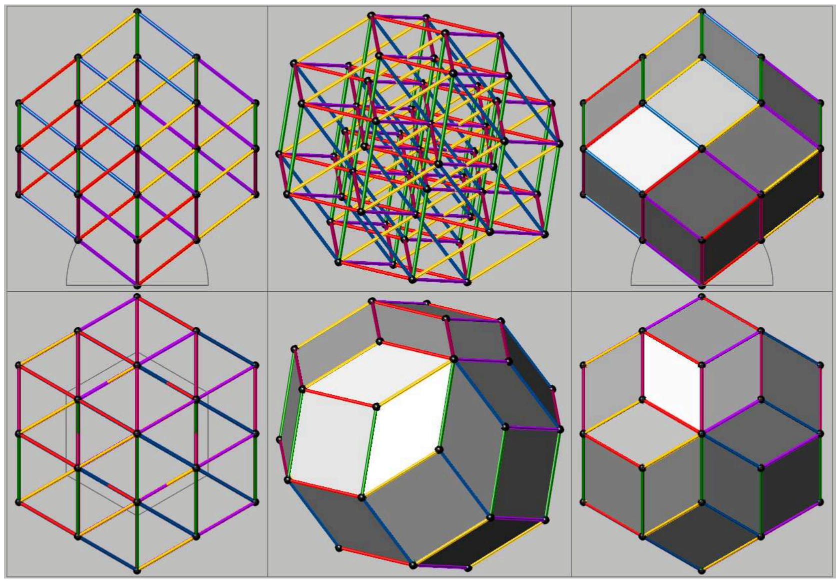


Figure 1

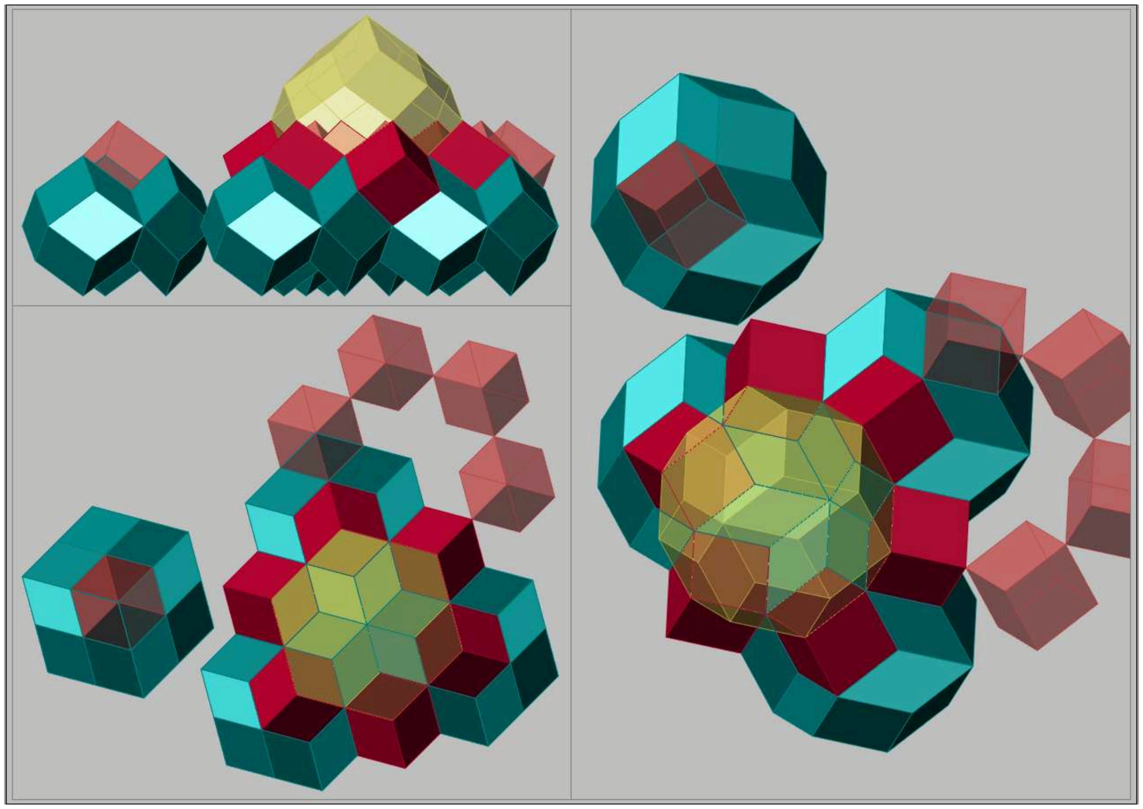


Figure 2

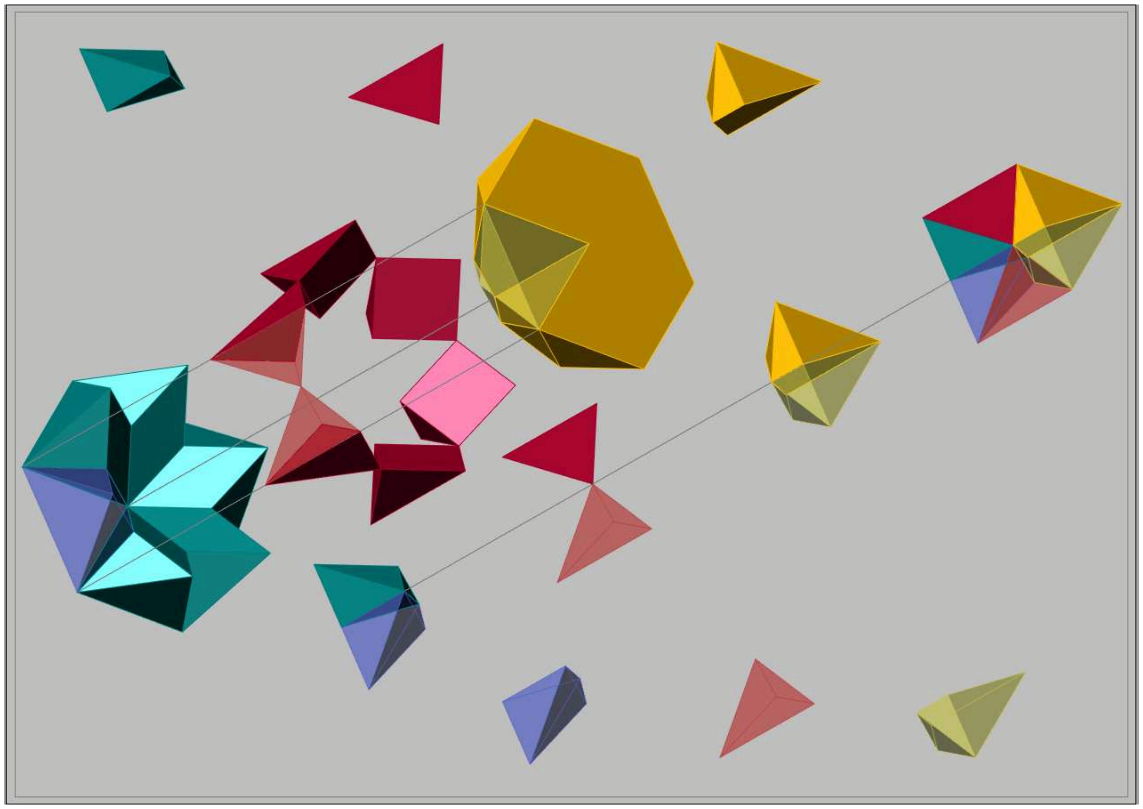


Figure 3

The 2-dimensional ortogonal projection of these 3-models indicates the idea how to construct space-filling with this model. However our 3-model of the 6-cube for example does not fill the space. The projected grid of the 3-cube joins our grid above and the cube fills the space well known. The edges of the cube can be selected from the conveniant lifted edges of the 6-cube's 3-model. With the selected four edges of the grid we can build the 3-model of the 4-cube. The shell of this is a rhombic dodekahedron which fills the space but this arrangement has not any rotational symmetry without additional assumptions. We can however replace a cube in the hole of the rotational-symmetrically arranged rhombic dodekahedra and continue the filling in a sixfold polar array with a rhombic triacontahedron which contains our 3-model of the 6-cube (Fig. 2).

It can be seen, that we can fill the space with these solids. The basic stones are to cut from a honeycomb by symmetry-planes. If the cutting process has been completed, we have the basic stones from the three starting solids (Fig. 3).

Another possibility is to rearrange our space-filling, assembling the 3-models of the k- and j-cubes from lower-dimensional cube 3-models. From the given 6 edges we can combine the 3-models of $2 < j < k$ cubes: 4 of the 3-cubes, 3 of the 4-cubes and 1 of the 5-cubes. Their additions (Fig. 4) can replace the 3-models of the above k- and j-cubes in our mosaic.

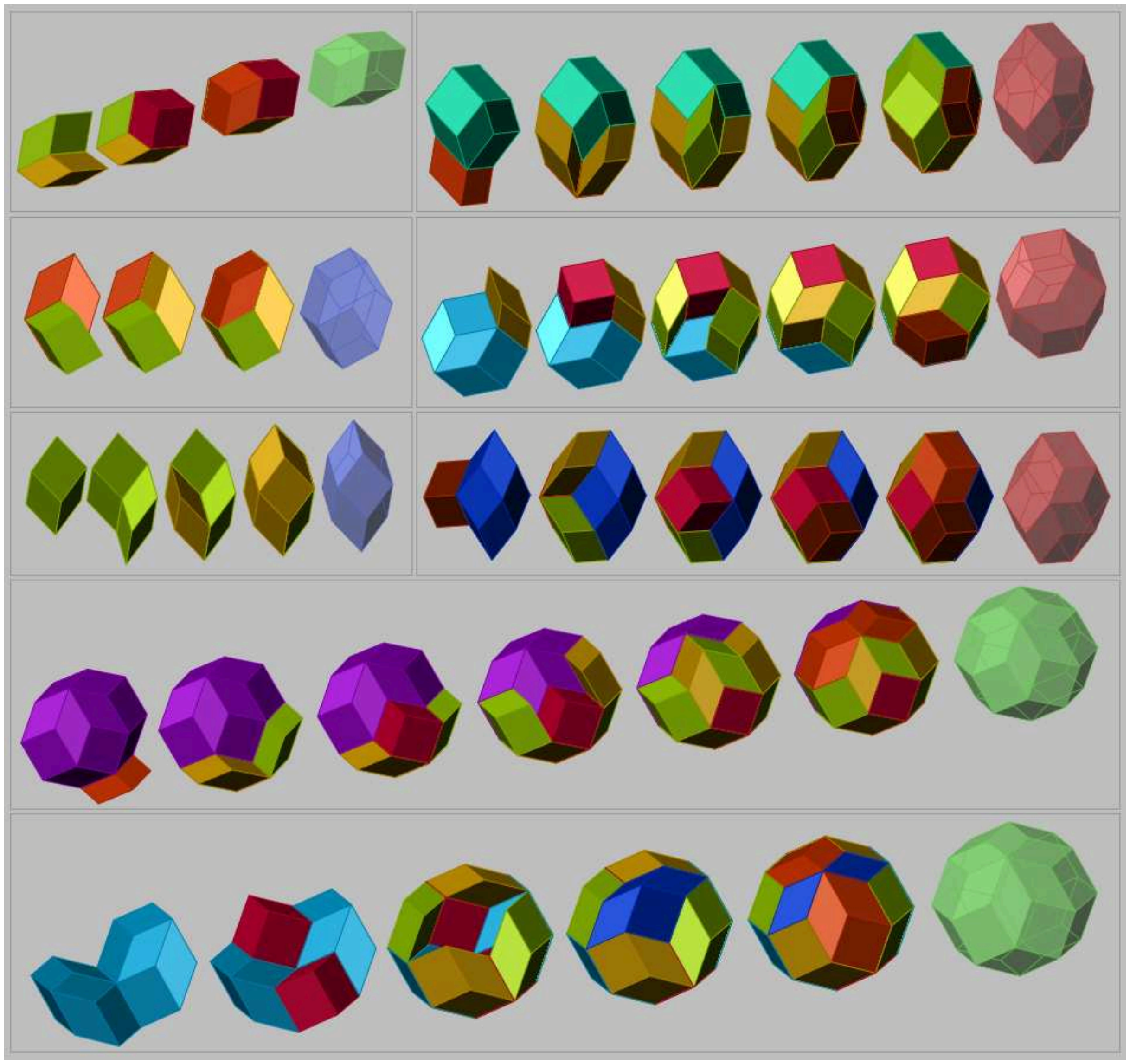


Figure 4

Interpreting the starting construction of the k-cube 3-model as a sequence of dispositions, the increasing dimensional inner $2 < j < k$ cube 3-models can "easily" be separated. The edges of the $0,1,...,k$ cube model-sequence are parallel to the k-segment chain approaching a starting helix, and the disposition vectors are joining each other along this segment chain. The model $0,1,...,k-1$ parts can also be interpreted as intersections of two full models so that the equal dimensional parts are positioned around the main diagonal of a full model, symmetrically to its centre point. More on this full model (3-model of the n-cube) can be read in [4], [5], [7], and on periodic and aperiodic tilings, based on d-dimensional crystallographic space groups, you find references in [1]. A further related topic might be: To what extent are these 3-models certain axonometric pictures of higher-dimensional cubes, created by a sequence of parallel projections? The Pohlke-theorem has surely limited validity in higher dimensions [2].

As it follows from our construction, the vertices lie in planes parallel to the basic plane of the construction, therefore a plane-tiling appears on these horizontal intersections of our space-filling solid-mosaics based on the 3-model of the 6-cube (Fig. 5). This has rotational symmetries but the diagonal intersections can be identical with the longitudinal and cross-intersections (Fig. 6).

We can see in Fig. 7 the horizontal intersections alternating one another $(0,1,2,3,2,1,0,1,...)$ in the space-filling mosaic based on the 3-model of the 6-cube. The tiling of the intersections can further be dissected by the perpendicular projected edges of the intersected solids (Fig. 8). A similar phenomenon could be seen in the projection of the inner edges of the j-cube 3-models.

Projecting the combination of the intersection grids, the tiling can further be dissected (Fig. 9). The coloring here is kept to one intersection and the grid of another one is projected into this plane.

In Fig. 10 we have combined the grids of three and finally of all four horizontal intersections. This is further dissected by the projected edges of the intersected solids.

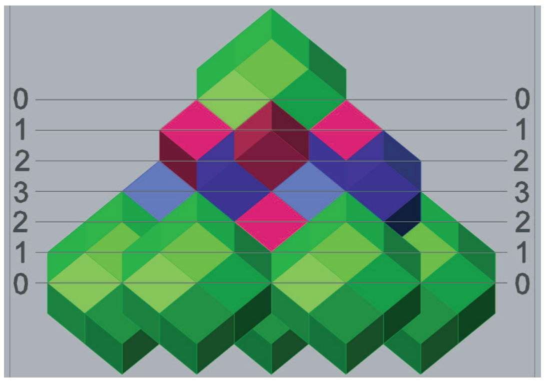


Figure 5

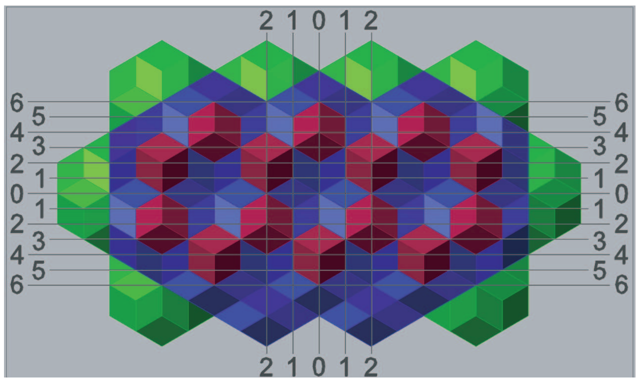


Figure 6

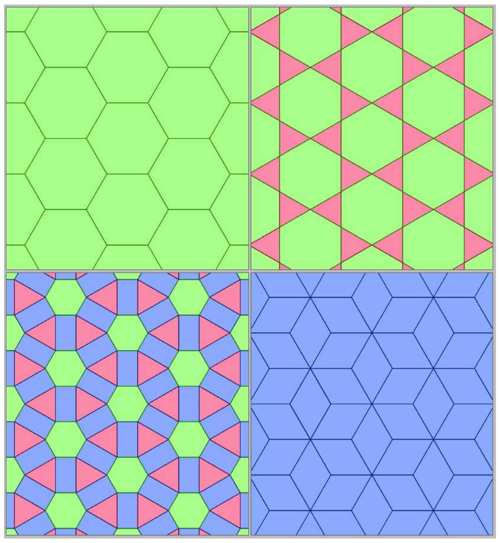


Figure 7

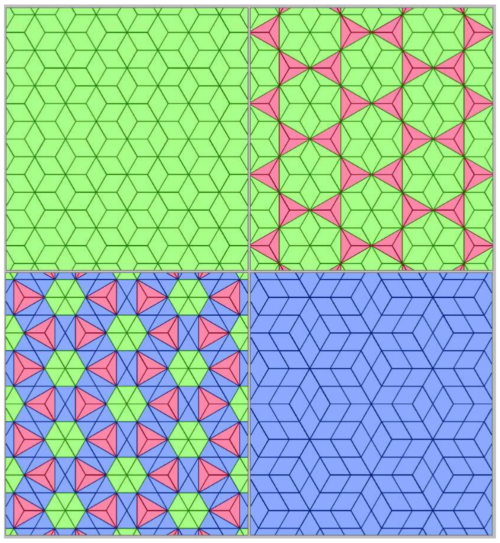


Figure 8

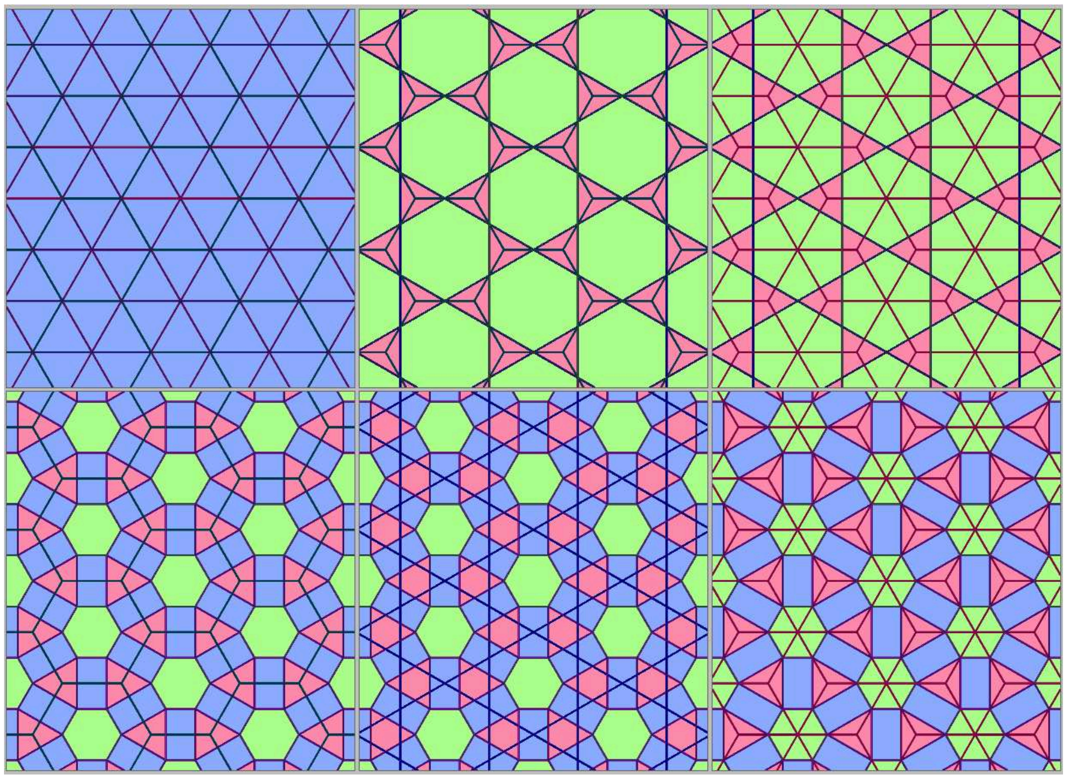


Figure 9

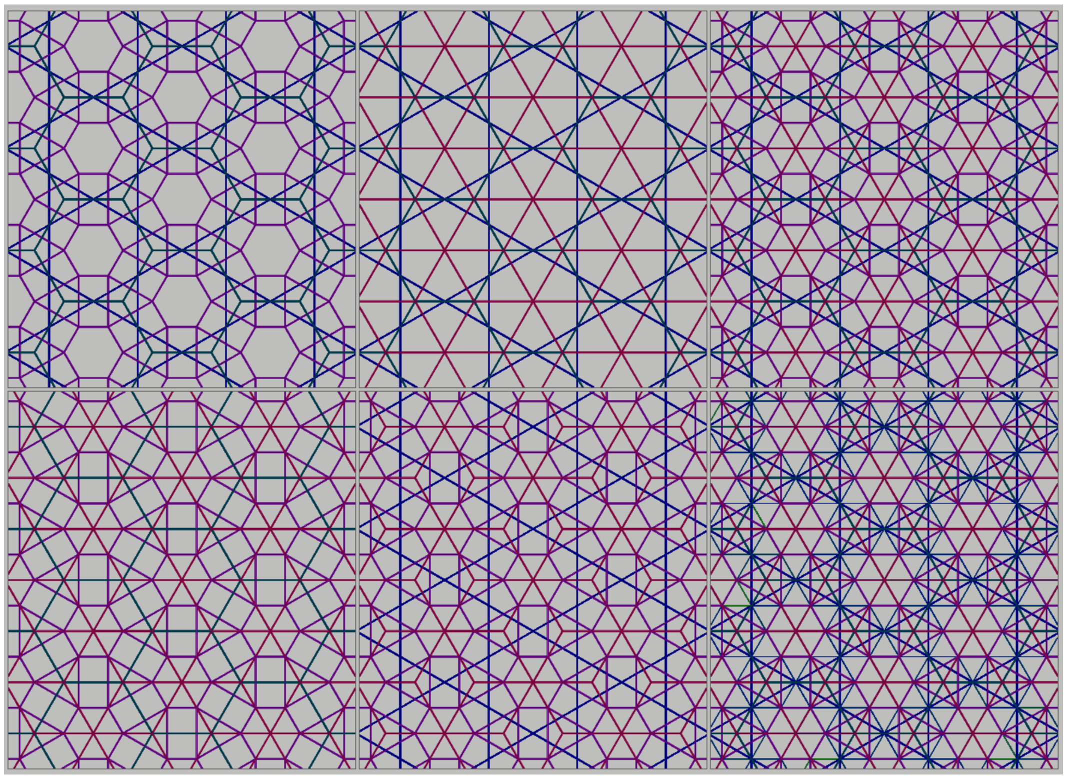


Figure 10

We can see in Fig. 11 the cross-intersections alternating one another $(0,1,2,1,0,1,...)$ in the space-filling mosaic based on the 3-model of the 6-cube. In the bottom row are the intersections supplemented by the projected edges of the intersected solids.

The alternating $(0,1,2,3,4,5,6,5,...)$ longitudinal intersections of our mosaic are descripted in Fig. 12. The methods of the further dissections could be applied here similarly to the horizontal intersections.

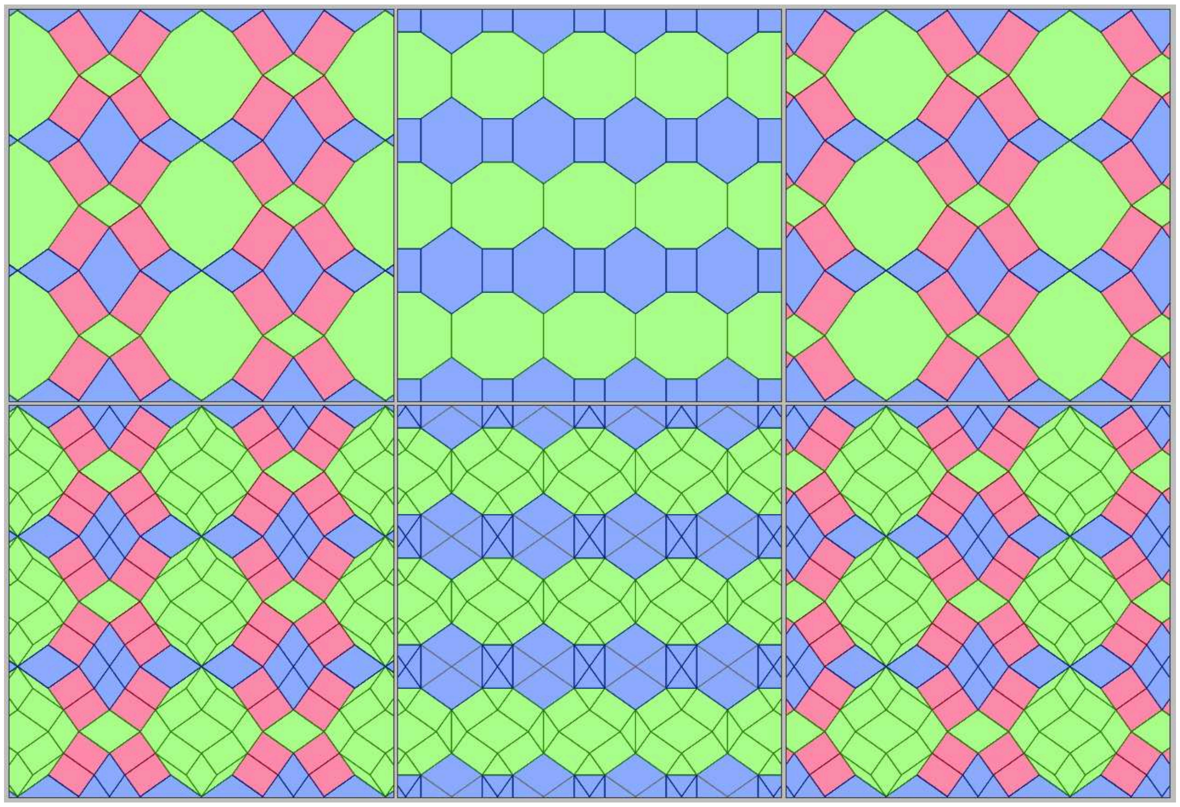


Figure 11

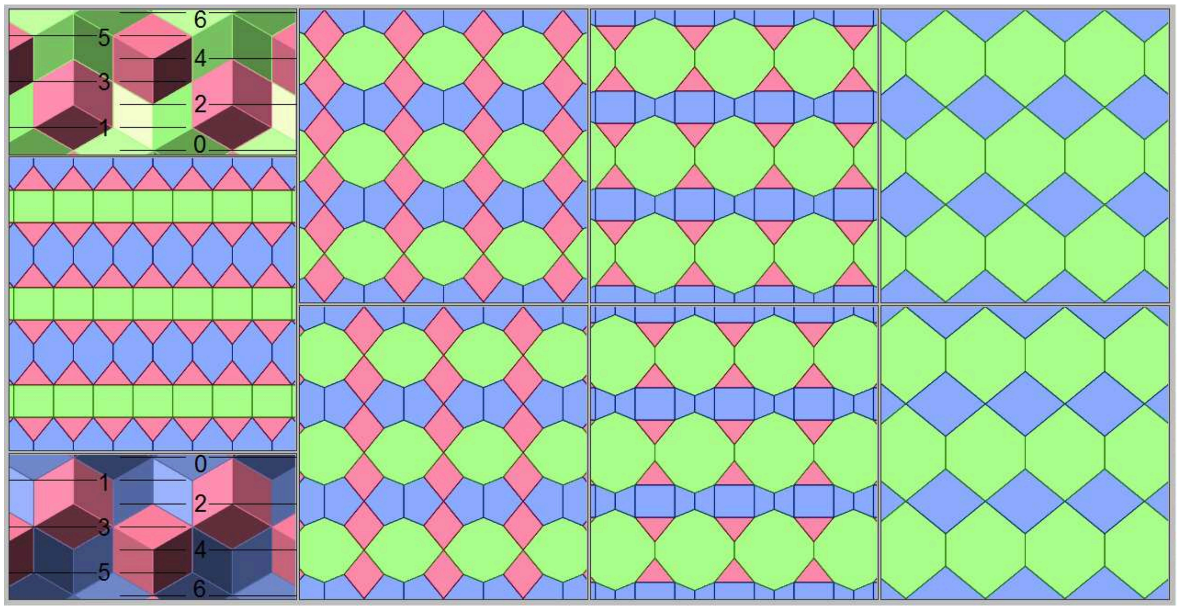


Figure 12

With the above methods two- and three-dimensional tilings based on the 3-models of k-cubes, can surely be made up to $k = 10$ and probably furthermore, too. These cases are just examined but not displayed yet in all details by the author.

The creation of the constructions and figures required for the paper was aided by the AutoCAD program and the Autolisp routines developed by the author.

References

[1] MOLNÁR, E., SCHULZ, T., SZIRMAI, J., *Periodic and aperiodic figures on the plane by higher dimensions*, Journal for Geometry and Graphics, Vol. 5 (2001), No. 2, 133-144

[2] STACHEL, H., *Mehrdimensionale Axonometrie*, Proceedings of the Congress of Geometry, Thessaloniki, 159-168 (1987)

[3] VÖRÖS L., *Reguläre Körper und mehrdimensionale Würfel*, KoG 9, Zagreb (2005), 21-27.

[4] VÖRÖS L., *A Symmetric Three-dimensional Model of the Hypercube*, Conference: Symmetry-Festival 2006. Budapest, to appear in Symmetry: Culture and Science, special issue

[5] VÖRÖS L., *Some Ways to Construct a Symmetric 3-Dimensional Model of the Hypercube*, 1st Croatian Conference on Geometry and Graphics, Bjelolasica, September 17-21 2006, Abstracts, 31

[6] http://home.inreach.com/rtowle/Zonohedra.html

[7] http://icai.voros.pmmf.hu

László Vörös

University of Pécs

M. Pollack Technical Faculty

Institut of Architecture

e-mail: vorosl@witch.pmmf.hu

Original scientific paper
Accepted 21. 12. 2006.

ATTILA BÖLCSKEI*
BRIGITTA SZILÁGYI

Visualization of Curves and Spheres in *Sol* Geometry

Visualization of Curves and Spheres in *Sol* Geometry

ABSTRACT

The paper makes an attempt to visualize one of the homogeneous geometries, the *Sol* geometry, by illustrating first the geodesic curves and spheres then the so-called translation curves and spheres. We've collected their basic properties, too.

Key words: visualization of Thurston's geometries

MSC 2000: 53B20, 53C30

Vizualizacije krivulja i ploha u *Sol* geometriji

SAŽETAK

Ovaj članak je pokušaj vizualizacije jedne od homogenih geometrija, *Sol* geometrije. Prvo se ilustriraju geodetske krivulje i sfere, a zatim i tzv. translatirajuće krivulje i sfere. Također su navedena njihova osnovna svojstva.

Ključne riječi: vizualizacija Thurstonovih geometrija

"*This* (the *Sol* geometry) *is the real weird. Unlike the previous geometries, solve geometry isn't even rotationally symmetric. I don't know any good intrinsic way to understand it.*" (J. R. WEEKS) [7]

1 Introduction

In [5] W. P. Thurston formulated a geometrization conjecture for three-manifolds which states that every compact orientable three-manifold has a canonical decomposition into pieces, each of which admits a canonical geometric structure from among the 8 maximal simply connected homogeneous Riemannian 3-geometries E^3, H^3, S^3, $S^2 \times R$, $H^2 \times R$, $\widetilde{SL(2,R)}$, Nil and Sol. Obviously, the Poincaré conjecture (a compact three-manifold with trivial fundamental group is necessarily homeomorphic to the 3-sphere) is a special case of the Thurston conjecture. In the past thirty years, many mathematicians have contributed to the understanding of this problem, maybe the most important attempts are due to R. Hamilton. In 2006 a scoop went round the world claiming that a Russian mathematician, G. I. Perelman could give a complete proof of the Thurston conjecture and so the Poincare conjecture, too. Followed by the complex and knotty proof (using modern differential geometry of Ricci flows) the interest has turned to homogeneous spaces. This paper tries to help in understanding one of the above geometries, the *Sol*.

Let (M,g) be a Riemannian manifold. If for any $x,y \in M$ there does exist an isometry $\Phi : M \to M$ such that $y = \Phi(x)$, then the Riemannian manifold is called *homogeneous*.

The visualization of the three possible two-dimensional homogeneous Riemann geometries E^2, H^2, S^2 is familiar to anyone, but in higher dimensions we face a lot of open questions. Even in three dimensions, where first time anisotropic cases also appear we have difficulties in the imagination. No doubt, the standard models work for E^3, H^3, S^3, moreover real-time interactive graphics algorithms have been developed by J. R. WEEKS that can be extended even more for the product spaces $S^2 \times R$, $H^2 \times R$ [8]. The remaining three Thurston's homogeneous 3-dimensional geometries $\widetilde{SL(2,R)}$, Nil and Sol, however are difficult to handle. From these the twisted spaces $\widetilde{SL(2,R)}$ and Nil need multiple imaging and there are just a few results about them, whilest the Sol (mentioned also as *solv* in the literature) is the most unusual as our motto above indicates, as well (for more information consult [5], [6], [4]). We note that in the paper [2] Emil MOLNÁR elaborated the projective interpretations of all the eight geometries, we only cite his model for Sol.

Sol geometry can be obtained by giving a group structure T to be a semi-direct product $R \ltimes R^2$ as follows:

$$\begin{pmatrix} 1 & a & b & c \end{pmatrix} \begin{pmatrix} 1 & x & y & z \\ 0 & e^{-z} & 0 & 0 \\ 0 & 0 & e^{z} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} =$$

$$\begin{pmatrix} 1 & x+ae^{-z} & y+be^{z} & z+c \end{pmatrix}$$

* Supported by the Öveges József Programme OMFB-01525/2006

is the right action by a translation (x,y,z) on an affine point (a,b,c) yielding also a point of *Sol* expressed in homogeneous (projective) coordinates after choosing a fixed origin $O(1,0,0,0)$.

Then an invariant metric on $Sol(O,T)$ is given by

$$(ds)^2 = e^{2z}(dx)^2 + e^{-2z}(dy)^2 + (dz)^2,$$

as infinitesimal arc length square, now in any point $(1,x,y,z)$ [5], [2].

2 Geodesics and their representation

In the following we briefly recall from [1] the standard procedure yielding the geodesics of *Sol*.

Consider first the fundamental (metric) tensor from the above mentioned equation

$$(g_{ij}) = \begin{pmatrix} e^{2z} & 0 & 0 \\ 0 & e^{-2z} & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

The well-known equation of geodesics

$$\frac{d^2u^k}{dt^2} + \Gamma^k_{ij}\frac{du^i}{dt}\frac{du^j}{dt} = 0$$

containing the Christoffel symbols of second kind: $\Gamma^k_{ij} = \frac{1}{2}\left(\frac{\partial g_{jl}}{\partial u^i} + \frac{\partial g_{li}}{\partial u^j} - \frac{\partial g_{ij}}{\partial u^l}\right) g^{lk}$ turns [1] to

$$\begin{aligned} &\ddot{x} + 2\dot{x}\dot{z} = 0 \\ &\ddot{y} - 2\dot{y}\dot{z} = 0 \\ &\ddot{z} - e^{2z}(\dot{x})^2 + e^{-2z}(\dot{y})^2 = 0. \end{aligned}$$

Solving this differential equation system as a Cauchy problem

$$\begin{matrix} x(0) = 0 & & \dot{x}(0) = u \\ y(0) = 0 & \text{and} & \dot{y}(0) = v \\ z(0) = 0 & & \dot{z}(0) = w \end{matrix}$$
$$u^2 + v^2 + w^2 = 1$$

we could arrange the following table that contains our results:

(1)	$u \neq 0$ $v \neq 0$ $0 < \lvert w\rvert = \sqrt{1-u^2-v^2} < 1$	$x(t) = u\int_0^t e^{-2z(\tau)}d\tau$ $y(t) = v\int_0^t e^{2z(\tau)}d\tau$ $z(t)$ comes from the separable differential equation $\frac{dz}{\pm\sqrt{1-u^2e^{-2z}-v^2e^{2z}}} = dt$, for $w \gtrless 0$ whose solution is non-elementary function.
(2)	$u \neq 0$ $v \neq 0$ $w = 0$	$x(t) = ut$ $y(t) = vt$ $z(t) = 0$
(3)	$v = 0$ $0 < \lvert w\rvert = \sqrt{1-u^2} < 1$	$x(t) = u\frac{\sinh t}{\cosh t + w\sinh t}$ $y(t) = 0$ $z(t) = \ln(\cosh t + w\sinh t)$
(4)	$u = 0$ $0 < \lvert w\rvert = \sqrt{1-v^2} < 1$	$x(t) = 0$ $y(t) = v\,\frac{\sinh t}{\cosh t - w\sinh t}$ $z(t) = -\ln(\cosh t - w\sinh t)$
(5)	$u = 0$ $v = 0$ $\lvert w\rvert = 1$	$x(t) = 0$ $y(t) = 0$ $z(t) = \pm t$, for $w = \pm 1$

Table 1: *Table of geodesics in Sol geometry, depending on the initial velocity parameters* (u,v,w), $u^2+v^2+w^2 = 1$.

The forthcoming pictures try to visualize the most general cases (1) and (3). As we easily see the change $(u,v,w) \leftrightarrow (v,u,-w)$ leads to the isometry of the corresponding geodesic curves.

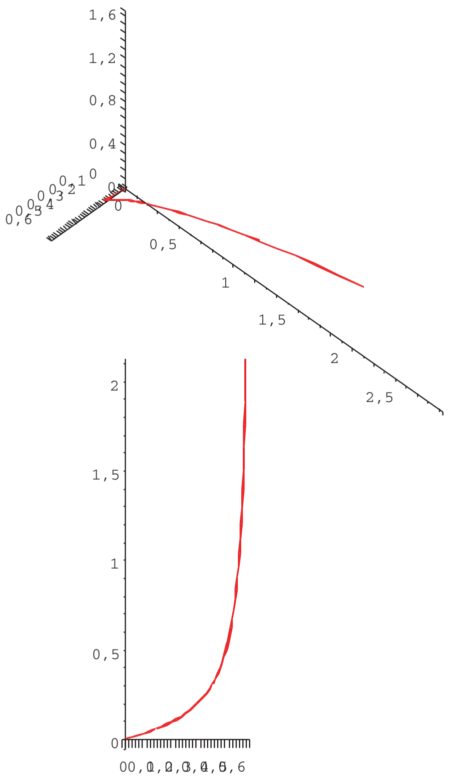


Fig. 1: *The approximate view of the most general geodesic curve with initial velocity parameters $u = 0.9$ and $v = 0.25$ in the parameter interval $t \in [0,2]$. The first picture shows the curve in a general view, the other from the direction of z-axis.*

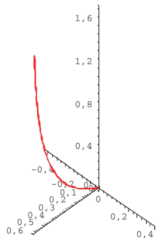


Fig. 2: *Geodesic curve starting in the [x,z] coordinate plane with $u = 0.9$ in a general view in the parameter interval $t \in [0,2]$.*

Clearly, the geodesic sphere unfortunately can not be expressed in a closed explicite form. We give approximations by plotting the endpoints of many geodesic curves (of the first type) with different initial unit velocities, according to geographic parameters

$$\begin{aligned} u &= \cos\vartheta\cos\varphi & -\pi \le \varphi \le \pi \\ v &= \cos\vartheta\sin\varphi & -\frac{\pi}{2} \le \vartheta \le \frac{\pi}{2} \\ w &= \sin\vartheta \ . \end{aligned}$$

That means, if ϑ is fixed and φ varies, then the endpoints of geodesics describe an altitude circle. Similarly we get longitude half-circle for fixed φ. The following figures show our results.

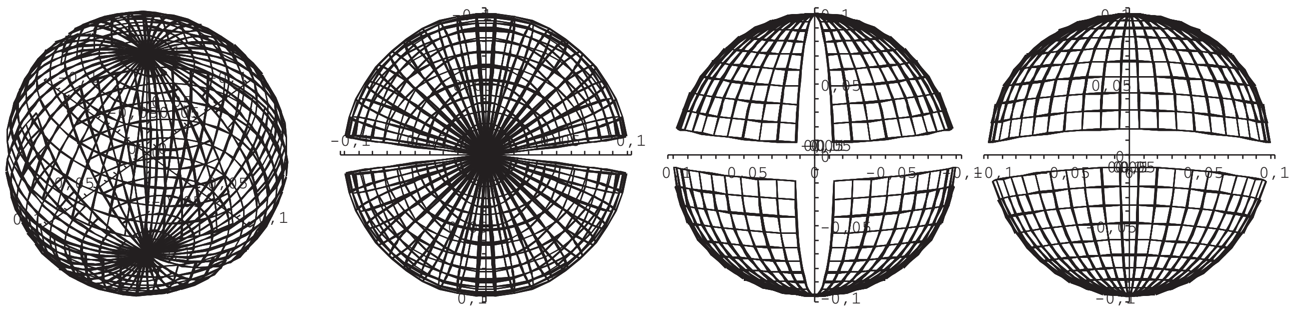


Fig. 3: *Geodesic sphere of radius 0.1. The first picture shows the sphere in a general view, then from the direction of axes z, y and x, respectively.*

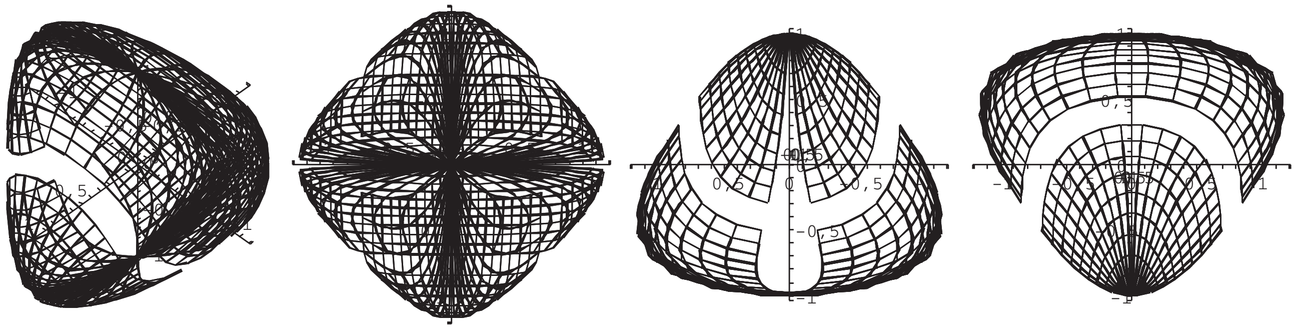


Fig. 4: *The same arrangement as in Fig. 4 but now the radius is 1.*

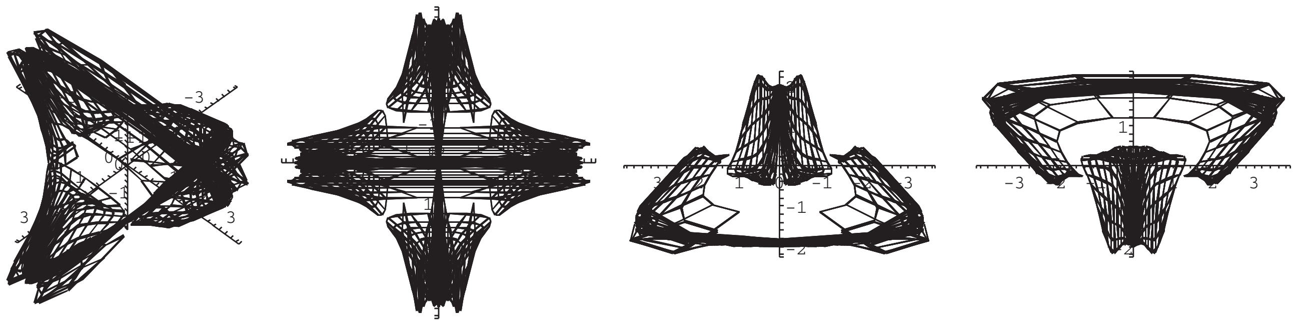


Fig. 5: *Geodesic sphere of radius 2.*

3 Translation curves and spheres

A Riemannian manifold with a transitive group of isometries is called homogeneous. In a homogeneous space there are postulated isometries, mapping each point to any point. Translations can be introduced in a natural way. Consider a unit vector at the origin. Translations, postulated at the beginning carry this vector to any point by its tangent mapping. If a curve $t \mapsto (x(t), y(t), z(t))$ has just the translated vector as tangent vector in each point, then the curve is called a *translation curve*. This assumption leads to a system of first order differential equations, thus translation curves are simpler than geodesics and differ from them in most cases (except in spaces of constant curvature).

From [3] we have already known the solution of the above defined system

$$\begin{aligned}\dot{x}(t) &= ue^{-z(t)} \quad , \\ \dot{y}(t) &= ve^{z(t)} \quad , \\ \dot{z}(t) &= w \quad ,\end{aligned}$$

of differential equation which holds for a curve starting at the origin in direction (u, v, w):

$$\begin{aligned}x(t) &= -\frac{u}{w}\left(e^{-wt} - 1\right) \quad . \\ y(t) &= \frac{v}{w}\left(e^{wt} - 1\right) \quad , \\ z(t) &= wt \quad ,\end{aligned}$$

.
In the following -as illustration- we show how a translation curve looks like.

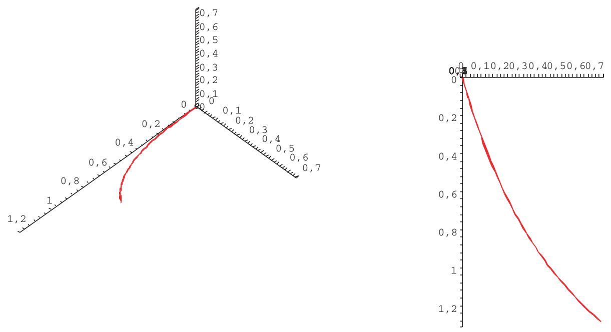


Fig. 6: *Translation curve with the same initial velocity parameters as for geodesics above, ($u = 0.9$ and $v = 0.25$) in the parameter interval $t \in [0, 2]$. The first picture shows the curve in a general view, the other from the direction of z-axis.*

With unit velocity translation curves we can define the *translation sphere* of radius r with centre in the origin of usual longitude and altidude parameters φ and ϑ, respectively ([3]):

$$\begin{aligned} u &= \cos\vartheta\cos\varphi & -\pi \le \varphi \le \pi \\ v &= \cos\vartheta\sin\varphi & -\frac{\pi}{2} \le \vartheta \le \frac{\pi}{2} \\ w &= \sin\vartheta\ ; & \end{aligned}$$

$$\begin{aligned} x(\vartheta,\varphi) &= -\cot\vartheta\cos\varphi\left(e^{-r\sin\vartheta}-1\right) \\ y(\vartheta,\varphi) &= \cot\vartheta\sin\varphi\left(e^{r\sin\vartheta}-1\right) \\ z(\vartheta,\varphi) &= r\sin\vartheta\ . \end{aligned}$$

As illustrations we give the following nice pictures.

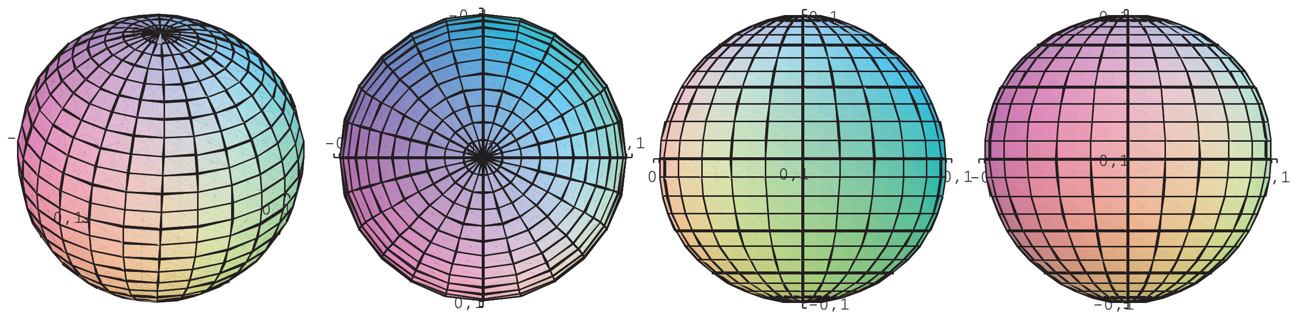


Fig. 7: *Translation sphere of radius 0.1. The first picture shows the sphere in a general view, then from the direction of axes z, y and x, respectively.*

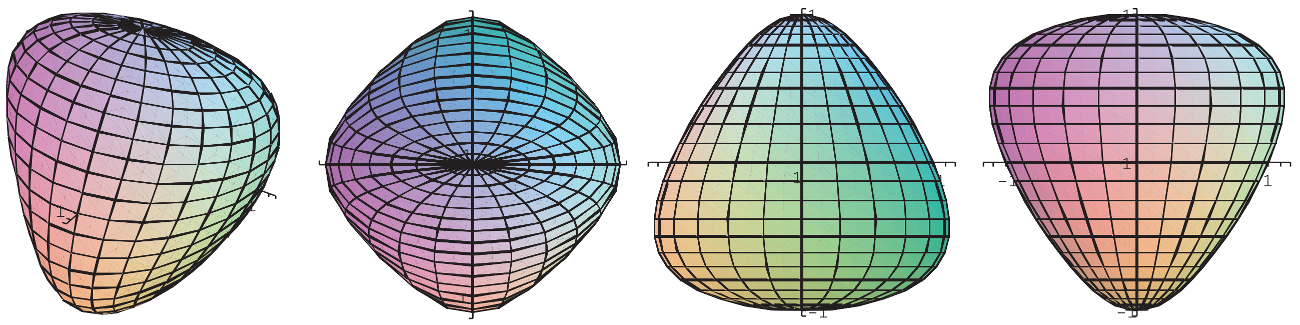


Fig. 8: *The same arrangement as above but now the radius is 1.*

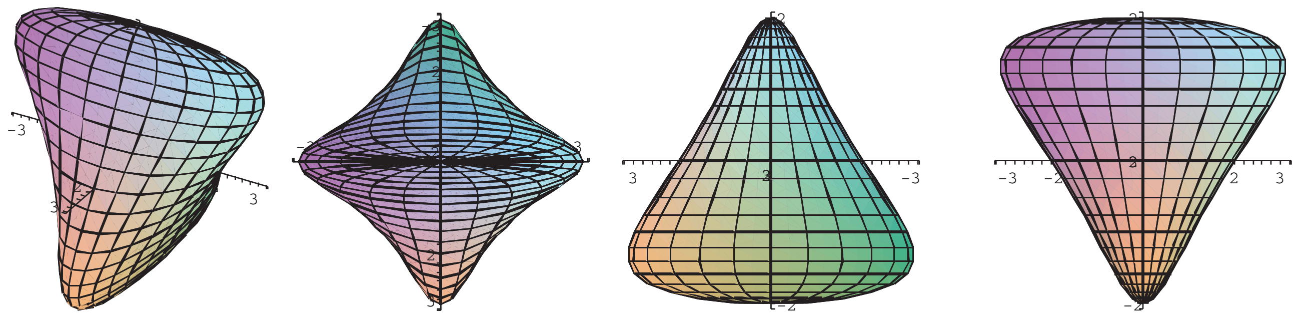


Fig. 9: *Translation sphere of radius 2.*

References

[1] BÖLCSKEI A., SZILÁGYI B., *Frenet Formulas and Geodesics in Sol geometry*, (to appear in Beiträge zur Algebra und Geometrie)

[2] MOLNÁR, E., *The projective interpretation of the eight* 3*-dimensional homogeneous geometries*, Beiträge zur Algebra und Geometrie 38 (1997), 261-288. p

[3] MOLNÁR, E., SZILÁGYI, B., *Translation curves and their spheres in homogeneous geometries*, (manuscript)

[4] SCOTT, P., *The geometries of 3-manifolds*, Bull. of the Lon. Math. Soc. 15 (1983), 401-487.

[5] THURSTON, W.P., *Three dimensional manifolds, Kleinian groups and hyperbolic geometry*, Bull. Amer. Math. Soc. 6 (1982), 357-381.

[6] THURSTON, W.P., *Three-Dimensional Geometry and Topology*, Vol. 1 (edited by S. Levy) Princeton University Press, Princeton, New Yersey (1997)

[7] WEEKS, J.F., *The Shape of Space*, Marcel Dekker Inc. (2001)

[8] WEEKS, J.F., *Real-Time Animation in Hyperbolic, Spherical and Product Geometries*, Non-Euclidean Geometries (edited by A. Prékopa and E. Molnár) Mathematics and Its Applications 581, Springer, (2006), 287-306.

Attila Bölcskei
Szent István University
Ybl Miklós College
Dept. of Representation and Informatics
H 1146 Budapest XIV. Thököly I. u. 74.
e-mail: bolcskei.attila@ymmfk.szie.hu

Brigitta Szilágyi
Budapest University of Technology and Economics
Dept. of Geometry
H 1521 Budapest XI. Egry J. u. 1, H.II.22.
e-mail: szilagyi@math.bme.hu

Professional paper
Accepted 28. 11. 2006.

**Özcan Gelişgen,
Rustem Kaya**

Generalization of $\alpha-$distance to $n-$dimensional Space

Generalization of $\alpha-$distance to $n-$dimensional Space

ABSTRACT

In this study, we generalize the concept of $\alpha-$ distance which contains both of Taxicab distance and Chinese Checker distance as special cases to $n-$dimensional space.

Key words: Taxicab distance, CC-distance, α-distance, metric, non-Euclidean geometry

MSC 2000: 51K05, 51K99

Poopćenje $\alpha-$udaljenosti u $n-$dimenzionalnom prostoru

SAŽETAK

U članku se poopćuje pojam $\alpha-$ udaljenosti koji taxi udaljenost i CC-udaljenost sadrži kao posebne slučajeve u $n-$dimenzionalnom prostoru.

Ključne riječi: Taxi udaljenost, CC-udaljenost, α-udaljenost, metrika, neeuklidska geometrija

Tian [9] gave a generalization of both Taxicab and Chinese Checker distances in the plane, and named it as $\alpha-$distance. In [6], $\alpha-$distance have been extended to three dimensional space. In this work the concept of $\alpha-$distance is generalized to $n-$dimensional space.

In the following definition, we introduce a family of distances in $\mathbb{R}^n$, which include Taxicab and Chinese Checker distances as special cases.

Definition:

Let $P_1=(x_1,x_2,\dots,x_n)$ and $P_2=(y_1,y_2,\dots,y_n)$ be two points in $\mathbb{R}^n$. If

$$\Delta_{P_1P_2}=\max\{|x_1-y_1|,|x_2-y_2|,\dots,|x_n-y_n|\}=|x_j-y_j|$$

and

$$\delta_{P_1P_2}=\sum_{i\in I}|x_i-y_i|,\quad I=\{1,2,\dots,n\}\setminus\{j\},$$

then the function $d_\alpha:\mathbb{R}^n\times\mathbb{R}^n\to\mathbb{R}$ such that

$$d_\alpha(P_1,P_2)=\Delta_{P_1P_2}+(\sec\alpha-\tan\alpha)\delta_{P_1P_2},\quad \alpha\in[0,\pi/4],$$

is called generalized $\alpha-$distance between points P_1 and P_2.

Generalized Taxicab and Chinese Checker distances between points P_1 and P_2 in $\mathbb{R}^n$ are $d_T(P_1,P_2)=\Delta_{P_1P_2}+\delta_{P_1P_2}$ and $d_c(P_1,P_2)=\Delta_{P_1P_2}+(\sqrt{2}-1)\delta_{P_1P_2}$, respectively. (See [1], [2], [3], [4], [5], [8]).

Notice that
$d_0(P_1,P_2)=d_T(P_1,P_2)$ and $d_{\frac{\pi}{4}}(P_1,P_2)=d_c(P_1,P_2)$.

Also, if $\delta_{P_1P_2}>0$, then for all $\alpha\in(0,\pi/4)$,
$d_E(P_1,P_2)<d_c(P_1,P_2)<d_\alpha(P_1,P_2)<d_T(P_1,P_2)$,
where d_E, d_c and d_T stand for the Euclidean, Chinese Checker and Taxicab distances, respectively.

Further, if $\delta_{P_1P_2}=0$, then P_1 and P_2 lie on a line which is parallel to one of coordinate axes, and for all $\alpha\in[0,\pi/4]$,
$d_c(P_1,P_2)=d_\alpha(P_1,P_2)=d_T(P_1,P_2)=d_E(P_1,P_2)$.

Let l be a line through P_1 and parallel to jth-coordinate axis and $l_1,\dots,l_n$ denote lines each of which is parallel to a coordinate axis distinct from jth-axis. Geometrically, the shortest way between the points P_1 and P_2 is the union of a line segment parallel to l_j and line segments each making α angle with one of $l_1,\dots,l_n$, as shown in Figure 1. Thus, the shortest distance d_α from P_1 to P_2 is sum of the Euclidean lengths of such n line segments.

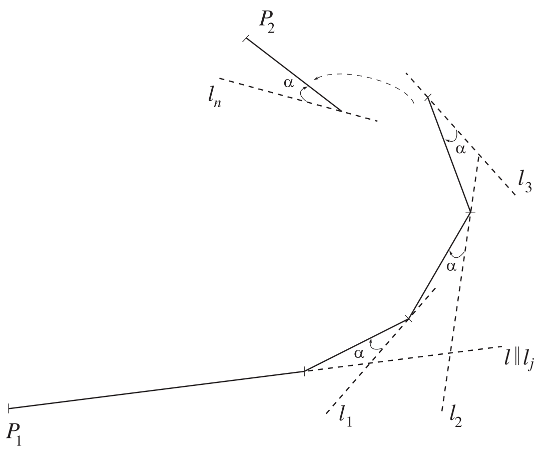


Figure 1

The next two propositions follow directly from the definition of the generalized α−distance.

Proposition 1.

The generalized α−distance is invariant under all translation in $\mathbb{R}^n$. That is, $T:\mathbb{R}^n\to\mathbb{R}^n$, where $T(x_1,x_2,\ldots,x_n)=(x_1+a_1,x_2+a_2,\ldots,x_n+a_n)$, $a_i\in\mathbb{R}$, does not change the distance between any two points in $\mathbb{R}^n$.

Proposition 2.

Let $P_1=(x_1,x_2,\ldots,x_n)$ and $P_2=(y_1,y_2,\ldots,y_n)$ be two points in $\mathbb{R}^n$. If $\Delta_{P_1P_2}=\left|x_{j_1}-y_{j_1}\right|$ for $j_1\in\{1,2,\ldots,n\}$, then

$$\left|x_{j_1}-y_{j_1}\right|+(\sec\alpha-\tan\alpha)\sum_{i\in I}|x_i-y_i|\geq$$
$$\geq\left|x_{j_2}-y_{j_2}\right|+(\sec\alpha-\tan\alpha)\sum_{i\in I'}|x_i-y_i|,$$

for $I=\{1,2,\ldots,n\}\setminus\{j_1\}$, $I'=\{1,2,\ldots,n\}\setminus\{j_2\}$, $j_2\in I$ and $\alpha\in[0,\pi/4]$.

Proof: Let P_1=$(x_1,x_2,\ldots,x_n)$ and P_2=$(y_1,y_2,\ldots,y_n)$.
If $\Delta_{P_1P_2}$=$\left|x_{j_1}-y_{j_1}\right|$,
then $a=\left|x_{j_1}-y_{j_1}\right|+(\sec\alpha-\tan\alpha)\sum_{i\in I}|x_i-y_i|$.
Let $b=\left|x_{j_2}-y_{j_2}\right|+(\sec\alpha-\tan\alpha)\sum_{i\in I'}|x_i-y_i|$, for $j_2\in I$.

$$\begin{aligned}a-b &= \left|x_{j_1}-y_{j_1}\right|+(\sec\alpha-\tan\alpha)\sum_{i\in I}|x_i-y_i|-\\&\quad-\left|x_{j_2}-y_{j_2}\right|-(\sec\alpha\text{-}\tan\alpha)\sum_{i\in I'}|x_i-y_i|\\&= \left|x_{j_1}-y_{j_1}\right|+(\sec\alpha-\tan\alpha)\left|x_{j_2}-y_{j_2}\right|-\\&\quad-\left|x_{j_2}-y_{j_2}\right|-(\sec\alpha-\tan\alpha)\left|x_{j_1}-y_{j_1}\right|\\&= (1-(\sec\alpha-\tan\alpha))(\left|x_{j_1}-y_{j_1}\right|-\left|x_{j_2}-y_{j_2}\right|).\end{aligned}$$

Notice that $(1-(\sec\alpha-\tan\alpha))\geq 0$ for all $\alpha\in[0,\pi/4]$ and $(\left|x_{j_1}-y_{j_1}\right|-\left|x_{j_2}-y_{j_2}\right|)\geq 0$. Thus $a-b\geq 0$.
That is, $\left|x_{j_1}-y_{j_1}\right|+(\sec\alpha-\tan\alpha)\sum_{i\in I}|x_i-y_i|\geq$

$$\geq\left|x_{j_2}-y_{j_2}\right|+(\sec\alpha-\tan\alpha)\sum_{i\in I'}|x_i-y_i|$$

for $I=\{1,2,\ldots,n\}\setminus\{j_1\}$, $I'=\{1,2,\ldots,n\}\setminus\{j_2\}$, $j_2\in I$ and $\alpha\in[0,\pi/4]$.

The following theorem shows that generalized α−distance is a metric.

Theorem 3.

For each $\alpha\in[0,\pi/4]$, generalized α−distance determines a metric for $\mathbb{R}^n$.

Proof: We have to show that d_α is positive definite and symmetric, and d_α holds triangle inequality. Let $P_1=(x_1,x_2,\ldots,x_n)$, $P_2=(y_1,y_2,\ldots,y_n)$ and $P_3=(z_1,z_2,\ldots,z_n)$ be three points in $\mathbb{R}^n$. Generalized α−distance between points P_1 and P_2 is $d_\alpha(P_1,P_2)=\Delta_{P_1P_2}+(\sec\alpha-\tan\alpha)\delta_{P_1P_2}$, $\alpha\in[0,\pi/4]$.

$d_\alpha(P_1,P_2)\geq 0$ since $|x_i-y_i|\geq 0$ and $(\sec\alpha-\tan\alpha)\geq 0$ for each $\alpha\in[0,\pi/4]$. Obviously, $d_\alpha(P_1,P_2)=0$ if and only if $P_1=P_2$. So d_α is positive definite.

Clearly $d_\alpha(P_1,P_2)=d_\alpha(P_2,P_1)$ follows from $|x_i-y_i|=|y_i-x_i|$. That is, d_α is symmetric.

Now, we try to prove that $d_\alpha(P_1,P_2)\leq d_\alpha(P_1,P_3)+d_\alpha(P_3,P_2)$ for all $P_1,P_2,P_3\in\mathbb{R}^n$ and $\alpha\in[0,\pi/4]$. For each $\alpha\in[0,\pi/4]$, and I=$\{1,2,\ldots,n\}\setminus\{j\}$,

$$\begin{aligned}d_\alpha(P_1,P_2) &= \left|x_j-y_j\right|+(\sec\alpha-\tan\alpha)\sum_{i\in I}|x_i-y_i|\\&= \left|x_j-z_j+z_j-y_j\right|+\\&\quad+(\sec\alpha-\tan\alpha)\sum_{i\in I}|x_i-z_i+z_i-y_i|\\&\leq \left|x_j-z_j\right|+\left|z_j-y_j\right|+\\&\quad+(\sec\alpha-\tan\alpha)\sum_{i\in I}(|x_i-z_i|+|z_i-y_i|)\\&= k.\end{aligned}$$

One can easily see that d_α satisfies the triangle inequality by examining the following cases:

Case I: If $\left|x_j-z_j\right|\geq|x_i-z_i|$ and $\left|z_j-y_j\right|\geq|z_i-y_i|$, $i,j\in\{1,2,\ldots,n\}$, $i\neq j$, then for each $\alpha\in[0,\pi/4]$, and

$I=\{1,2,\dots,n\}\setminus\{j\}$,

$$\begin{aligned} d_\alpha(P_1,P_2) &\le k \\ &= \left|x_j-z_j\right|+\left|z_j-y_j\right|+ \\ &\quad +(\sec\alpha-\tan\alpha)\sum_{i\in I}\left(|x_i-z_i|+|z_i-y_i|\right) \\ &= \left|x_j-z_j\right|+(\sec\alpha-\tan\alpha)\sum_{i\in I}|x_i-z_i|+ \\ &\quad +\left|z_j-y_j\right|+(\sec\alpha-\tan\alpha)\sum_{i\in I}|z_i-y_i| \\ &= d_\alpha(P_1,P_3)+d_\alpha(P_3,P_2). \end{aligned}$$

Case II: If $\left|x_j-z_j\right|\ge|x_i-z_i|$ and $\left|z_j-y_j\right|\le|z_i-y_i|$, $i,j\in\{1,2,\dots,n\}$, $i\ne j$, then there are two possible situations:

(i) Let $\left|x_j-z_j\right|+\left|z_j-y_j\right|\ge|x_i-z_i|+|z_i-y_i|$. Then for each $\alpha\in[0,\pi/4]$, and $I=\{1,2,\dots,n\}\setminus\{j\}$,

$$\begin{aligned} d_\alpha(P_1,P_2) &\le k \\ &= \left|x_j-z_j\right|+\left|z_j-y_j\right|+ \\ &\quad +(\sec\alpha-\tan\alpha)\sum_{i\in I}\left(|x_i-z_i|+|z_i-y_i|\right) \\ &= \left|x_j-z_j\right|+(\sec\alpha-\tan\alpha)\sum_{i\in I}|x_i-z_i|+ \\ &\quad +\left|z_j-y_j\right|+(\sec\alpha-\tan\alpha)\sum_{i\in I}|z_i-y_i| \\ &= d_\alpha(P_1,P_3)+\left|z_j-y_j\right|+ \\ &\quad +(\sec\alpha-\tan\alpha)\sum_{i\in I}|z_i-y_i| \\ &\le d_\alpha(P_1,P_3)+d_\alpha(P_3,P_2), \end{aligned}$$

where $\left|z_j-y_j\right|+(\sec\alpha-\tan\alpha)\sum\limits_{i\in I}|z_i-y_i|\le d_\alpha(P_3,P_2)$ because of Proposition 2.

(ii) Let $\left|x_j-z_j\right|+\left|z_j-y_j\right|\le|x_i-z_i|+|z_i-y_i|$. One can easily give a proof for the situation (ii) as in situation (i).

Case III: If $\left|x_j-z_j\right|\le|x_i-z_i|$ and $\left|z_j-y_j\right|\ge|z_i-y_i|$, $i,j\in\{1,2,\dots,n\}$, $i\ne j$, then there are two possible situations:

(i) Let $\left|x_j-z_j\right|+\left|z_j-y_j\right|\ge|x_i-z_i|+|z_i-y_i|$.

(ii) Let $\left|x_j-z_j\right|+\left|z_j-y_j\right|\le|x_i-z_i|+|z_i-y_i|$.

One can easily give a proof for the Case III as in the Case II.

References

[1] AKCA, Z., KAYA, R., *On The Distance Formulae in Three Dimensional Taxicab Space*, Hadronic Journal, Vol. 27, No. 5 (2004), 521-532.

[2] AKCA, Z., KAYA, R., *Lines and Circles in Taxicab Geometry*, Hadronic Journal Supplement, Vol. 19 (2004), 491-501.

[3] CHEN, G., *Lines and Circles in Taxicab Geometry*, Master Thesis, Department of Mathematics and Computer Science, Centered Missouri State University, 1992.

[4] DIVJAK, B., *Notes on Taxicab Geometry*, Scientific and Professional Information Journal of Croatian Society for Constructive Geometry and Computer Graphics (KoG), Vol. 5 (2000/01), 5-9.

[5] GELIŞGEN, O., KAYA, R., OZCAN, M., *Distance Formulae in the Chinese Checker Space*, Int. Jour. of Pure and Appl. Math. (IJPAM), Vol. 26, No. 1 (2006)

[6] GELIŞGEN, O., KAYA, R., *On α−distance in three dimensional space*, Applied Sciences, Vol. 8 (2006), 65-69.

[7] KAYA, R., GELIŞGEN, O., EKMEKCI, S.,BAYAR, A., *Group of Isometries of CC-Plane*, Missouri Journal of Mathematical Sciences, (To appear in 2006)

[8] KRAUSE, E.F., *Taxicab Geometry*, Addison - Wesley Publishing Company, Menlo Park, CA, 1975.

[9] TIAN, S., *Alpha Distance-A Generalization of Chinese Checker Distance and Taxicab Distance*, Missouri Journal of Mathematical Sciences, Vol. 17, No. 1, (2005), 35-40.

Özcan Gelişgen
e-mail: gelisgen@ogu.edu.tr

Rustem Kaya
e-mail:rkaya@ogu.edu.tr

Department of Mathematics
Faculty of Science and Arts
University of Eskișehir Osmangazi
Eskișehir, Türkiye

Professional paper
Accepted 4. 12. 2006.

GÜNTER WALLNER

Geometry of Real Time Shadows

Geometry of Real Time Shadows

ABSTRACT

Shadows provide important visual hints about the spatial relationship between objects. Shadow volumes are one way to generate sophisticated shadows for use in real time environments. This paper focuses on the geometric aspects which are involved in the creation of the shadow volume. Speed up techniques like shaders and dual space approaches for silhouette determination are discussed. Finally the application of the described methods in a software for shadow profile calculation is explained.

Key words: Shadow volumes, Dual space, Silhouette determination, Shader, Real time

MSC 2000: 51-04

Geometrija sjenâ u realnom vremenu

SAŽETAK

Sjene pružaju važne vizualne informacije o prostornom odnosu među objektima. Tijelo sjene je jedan način kako generirati profinjene sjene za prkaze u realnom vremenu. U ovom članku usredotočilo se na geometrijski aspekt uključen u stvaranje tijela sjene. Razmatraju se brze i efikasne tehnike za određivanje rastavnice preko dva pristupa: dualnog prostora i programa za sjenčanje (shadera). Na kraju je prikazana primjena opisanih metoda u softveru za određivanje oblika sjene.

Ključne riječi: tijelo sjene, dualni prostor, određivanje rastavnice, program za sjenčanje, realno vrijeme

1 Introduction

Shadows are an important part in computer graphics because they can reveal information that otherwise would not be ascertainable. Foremost, they reveal the spatial relationship between objects in the scene. They also disclose new angles on an object that otherwise might not be visible and they can also indicate the presence of off-screen objects. These and other visual functions of shadows in computer graphics are described by Birn in [5].

Shadow volumes were first proposed by Crow in 1977 [8]. With the advent of modern day computer graphic cards, shadow volumes are now possible in real time. Heidmann [14] adapted Crow's algorithm to hardware acceleration. His method is now known as the z-pass method (because the stencil buffer is incremented/decremented when a polygon passes the depth test). However, the z-pass method does not work correctly if the near clipping plane intersects the shadow volume. Carmack [6] solved the problem by using z-fail testing (the stencil buffer is incremented/decremented when a polygon fails the depth test). The z-fail method still yields incorrect results if the shadow volume is intersected by the far clipping plane. This problem can be circumvented by moving the far clipping plane to infinity, as proposed by [9].

Shadow maps (introduced by [26]) are image based alternatives to shadow volumes (which operate on the object geometry). In the meantime several different shadow map algorithms have been developed. Both methods have their benefits and drawbacks. For a comparison of the pros and cons of both methods see for example [25].

”Classic” shadow volume algorithms create hard shadows. A shadow region is divided into two parts: the region which is fully in shadow (umbra) and the region which is partially in shadow (penumbra). Hard shadows only consist of the umbra area. Soft shadow volume algorithms have been developed among others by Ulf Assarsson and Tomas Akenine-Möller [21, 1].

2 Assumptions and Definitions

The shadow volume algorithm requires that the shadow casting objects must be a 2-manifold polygon mesh and free of non-planar polygons. 2-manifold means that every edge of the mesh must be shared exactly by two polygons. It is also useful to restrict oneself to triangular meshes, because modern graphics hardware is optimized for triangle rendering.

Furthermore, all triangles must have the same winding order. For the following discussion a counter clockwise winding order and outward pointing normals are assumed.

A silhouette edge is an edge adjacent to one front-facing and one back-facing polygon. A polygon is called front-

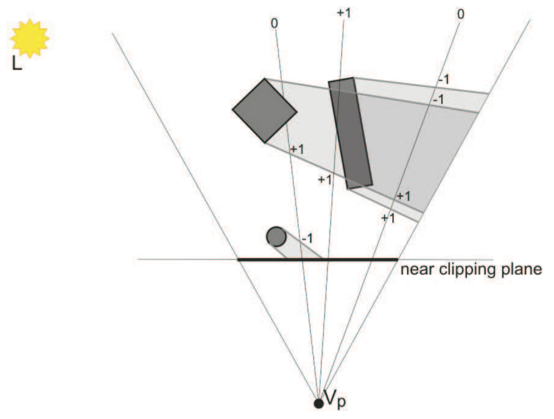


Figure 1: *The z-pass method. The values at the end of the rays represent the values left in the stencil buffer. Note that the stencil value of the leftmost ray is wrong due to the clipping of the shadow volume of the sphere at the near clipping plane.*

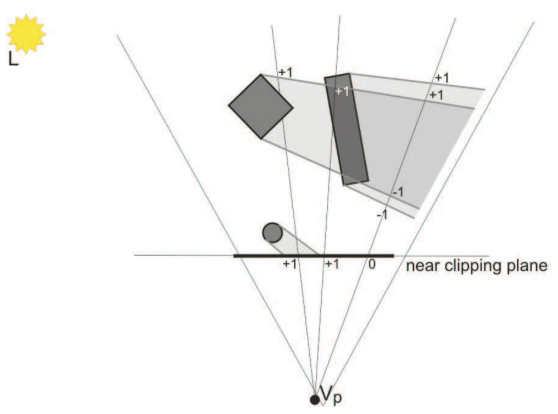


Figure 2: *The z-fail method. The values at the intersection of the ray and the near clipping plane represent the values left in the stencil buffer. This time the stencil value for the ray passing through the sphere is correct.*

facing with respect to the light if the dot-product of its normal and the vector from the light position and a point on the polygon is positive. Respectively a polygon is called back-facing with respect to the light if the dot-product is negative.

A border edge is an edge which is only adjacent to one face (which implies that the mesh is open). It should be noted that we can handle open meshes if we treat border edges as part of the silhouette. The silhouette is the set of all silhouette edges (and border edges).

3 Overview

I will first give an overview of the z-pass algorithm and then point out the differences with respect to the z-fail algorithm. The basic concept [...] is to use the stencil buffer as a masking mechanism to prevent pixels in shadow from being drawn during the render pass for a particular light source [17]. First of all the stencil buffer is initialized with zero and the z-buffer is initialized with the depth values of the visible objects during a first rendering pass. In this pass only light-independent attributes are considered (e.g. ambient light). Then the shadow volume is rendered with writes to the color buffer and depth buffer disabled. This is usually done in two steps. First, the front faces of the shadow volume (with respect to the camera position) are rendered and the stencil buffer is incremented each time the fragment passes the depth test. Second, the back faces are rendered. This time decrementing the value in the stencil buffer when a fragment passes the depth test.

As shown in figure 1, this leaves non-zero values in the stencil buffer wherever the shadow volume intersects a visible object. Figure 1 in addition shows why this approach fails if the shadow volume intersects the near clipping plane.

As noted by [3] the front faces must be rendered before the back facing polygons to avoid shadow counting overflow. That is, because under OpenGL the result of the increment and decrement functions is clamped to lie between 0 and the maximum unsigned integer value ($2^n - 1$ if the stencil buffer holds n bits) [22]. However, rendering the shadow volume geometry twice is a suboptimal solution. The OpenGL extension EXT_stencil_two_side [11] allows separate stencil states for front faces and back faces to be specified simultaneously. Therefore front faces as well as back faces can be rendered at once. Though this time it is not guaranteed that the front facing polygons will be rendered before the back faces. Consequently the feasibility exists that the stencil value for a particular pixel is decremented before it is incremented. We can account for that option by using another OpenGL extension, namely EXT_stencil_wrap [12], which allows stencil values to wrap when they exceed the maximum and minimum stencil values.

Several authors [4, 3, 6] proposed methods to cap the shadow volume at the near plane. However, these are computationally expensive and they have robustness problems.

Carmack [6] and others therefore suggested the z-fail algorithm. Instead of counting the shadow faces in front of a particular pixel, the shadow faces behind are counted. This time the near clipping plane problem is avoided because shadow volume geometry between the eye and the pixel is nonrelevant. Figure 2 shows the z-fail approach. As already mentioned in the introduction the z-fail approach moves the near clipping plane problem to the far plane, which can be prohibited by using an infinite projection matrix (see section 6).

4 Silhouette Detection

To calculate the shadow volume, we first have to determine the silhouette of the shadow casting object. The so-called brute force method for detecting silhouette edges is to loop through all edges and check the dot-product of the adjacent triangles. Since silhouette detection is one of the two major bottlenecks (beside fill rate consumption), as pointed out by [16], it is appropriate to use more sophisticated methods. [2] developed a dual space approach for silhouette extraction in 3D and [15] used a similar method but moved to four dimensions. Most recently [24] presented a paper about silhouette extraction in Hough space.

Because [15] are concerned with non photorealastic rendering, they determine the silhouette with respect to the viewpoint. However, in case of shadows the silhouette depends on the light position. Therefore the viewpoint must be substituted with the light position. The algorithm in [15] is based on the geometric concept of duality in a projective space and the following characterization of the silhouette: If $\mathbf{L}$ is the homogeneous light position, the set of silhouette points determines a general cone C (with apex $\mathbf{L}$) tangent to the differentiable surface M. If $\mathbf{L}'$ is the image plane of $\mathbf{L}$ when applying the duality map, the image C' of C is the intersection of the plane $\mathbf{L}'$ with the dual surface M'. C' can be identified with the silhouette set of surface M. A point $\mathbf{v} = (v_x, v_y, v_z, 1)$ of M belongs to the silhouette set if $(\mathbf{a} - \mathbf{v}) \cdot \mathbf{n} = 0$, where $\mathbf{n} = (n_x, n_y, n_z, 0)$ is the unit normal vector to M at $\mathbf{v}$ and $\mathbf{a} = (a_x, a_y, a_z, 1)$ is a point on the tangent plane at $\mathbf{v}$. The tangent plane at $\mathbf{v}$ is mapped onto a dual point $\mathbf{v}' = (v_x, v_y, v_z, -\mathbf{v} \cdot \mathbf{n})$. Therefore the silhouette set of M is characterized by the equation $\mathbf{L} \cdot \mathbf{v}' = (\mathbf{L} - \mathbf{v}) \cdot \mathbf{n} = 0$. Consequently, the problem of finding the silhouette of a differentiable surface is reduced to the problem of intersecting a plane with a surface.

Since I am concerned with polyhedral surfaces the problem can be reformulated in a way as described by [15]. The dual surface is built by mapping each vertex $\mathbf{v}$ of the mesh onto a homogeneous point $\mathbf{v}' = (v_x, v_y, v_z, -(\mathbf{v} \cdot \mathbf{n}))$. The dual surface has the same connectivity but different vertex positions. A dual edge e' of an edge $e = (\mathbf{v_1}, \mathbf{v_2})$ is a tuple $(\mathbf{v_1'}, \mathbf{v_2'})$. An edge e belongs to the set of silhouette edges if $\mathbf{L} \cdot \mathbf{v_1'} >= 0$ and $\mathbf{L} \cdot \mathbf{v_2'} < 0$ or vice versa. Each $\mathbf{v}'$ is then normalized (using the Euclidean norm) to make sure that each point of the dual surface lies inside the unit hypercube. This allows us to store each dual edge in a 4D variant of an octtree (I will call it hextree in the further discussion) as pointed out by [7]. At the highest level this hextree ranges from $(-1, -1, -1, -1)$ to $(1, 1, 1, 1)$. The space can be repeatedly divided into 16 smaller hextrees until a small enough partition is reached. A dual edge e' is then inserted into the smallest subcube which encloses $\mathbf{v_1'}$ as well as $\mathbf{v_2'}$.

Instead of using two bounding boxes per subcube to determine if the dual edges have to be verified [7] I use a different approach. For testing if an AABB[1] and a plane intersect, the box diagonal which is most aligned with the normal of the plane has to be found first. Second the diagonals vertices ($\mathbf{v_{min}}$ and $\mathbf{v_{max}}$) are inserted into the plane equation. If the signs of the results differ or at least one of them is zero, then the plane intersects the box [20]. [20] also points out that the two vertices can be found directly. The signs of the components of the plane normal are used as a bit mask. If this mask is interpreted as a number it can be used as index to an array of AABB vertices. This approach can easily be extended to four dimensions. Each of the 16 vertices of a 4D cube is stored in an array so that the minimum vertex is located at index 0 and the maximum vertex at position 15. Instead of the plane normal we interpret the signs of the components of $\mathbf{L}$ as a bit mask. The index i of $\mathbf{v_{min}}$ can then be calculated as $i = 8 \cdot sgn(L_x) + 4 \cdot sgn(L_y) + 2 \cdot sgn(L_z) + sgn(L_h)$ where

$$sgn(x) = \begin{cases} 0 & x >= 0 \\ 1 & \text{otherwise.} \end{cases}$$

The $\mathbf{v_{max}}$ vertex can be found by inverting the bit mask. The dual edges of a subcube must only be tested if $\mathbf{L} \cdot \mathbf{v_{min}} >= 0$ and $\mathbf{L} \cdot \mathbf{v_{max}} < 0$ or vice versa.

Building the dual surface and inserting the dual edges into the hextree can be done once in a preprocessing step as long as the connectivity of the object does not change. Furthermore silhouette detection must only be performed if the object position changes with respect to the light position.

5 Shadow Volume Construction

Once the set of silhouette edges is determined the edges must be extruded to form the shadow volume. As described by [17], no matter what finite distance silhouette edges are extruded, it is still possible that the shadow volume does not reach far enough to cast a shadow on every object in the scene that should intersect the volume. This problem worsens when the light source is very near to the shadow casting object, but it can be circumvented by using an infinite projection matrix. How this matrix can be obtained is described in section 6.

[1]AABB stands for Axis Aligned Bounding Box. Assuming an AABB is valid in our case because the hextree is axis aligned.

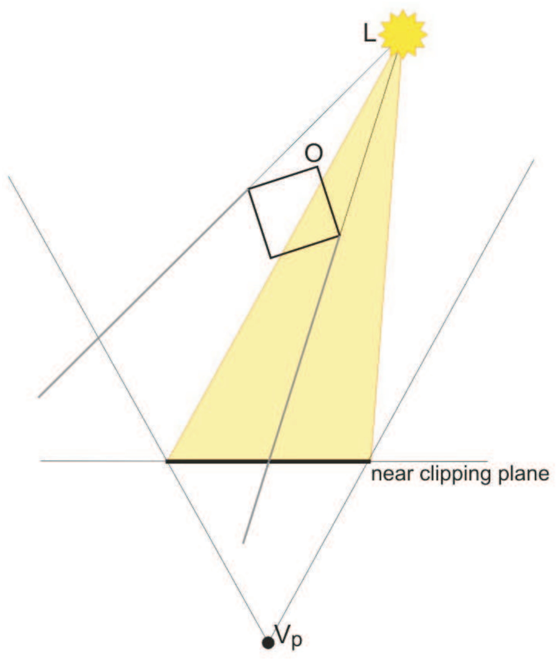


Figure 3: *An Object O is casting a shadow onto the near clip plane since it partially intersects the near clip volume (shaded)*

To make the z-fail algorithm work correctly, the shadow volume must be a closed volume where all polygons must have a consistent winding order. A complete shadow volume consists of: (1) the front cap (consisting of all front-facing polygons), (2) the extruded silhouette edges and (3) the back cap. It is notable that the extrusion of the geometry depends on the light source. For a point light the vertices of the silhouette edge must be extruded to infinity along the vector from the location of the point light to the vertex (see figure 4). If $\mathbf{v} = (v_x, v_y, v_z, 1)$ is the position of the vertex to be extruded and $\mathbf{L}$ is the position of the point light then the extruded vertex $\mathbf{v}_e = (v_x - L_x, v_y - L_y, v_z - L_z, 0)$.

For a directional light all extruded points converge to a single point in infinity (see figure 4) at position $(-L_x, -L_y, -L_z, 0)$. This implies that the back cap is not necessary for directional light sources. The back cap conventionally consisted of all back-facing polygons projected away from the light [9, 17]. But since the back cap is at infinity the shape does not matter [19]. The only constrain which remains is that the back cap must actually close the volume. This can be achieved with a simple triangle fan constructed from the extruded silhouette edges [19, 16].

The z-pass algorithm doesn't use caps, therefore the incorrect results if the shadow volume intersects the near clip plane (see [9] for details) or the viewpoint is inside the volume. From this point it is clear that the z-fail method is computationally more expensive and should only be used when necessary. To determine whether the shadow volume is clipped by the near plane the near clip volume has to be constructed. The near clip volume is bound by the planes which connect the near rectangle to the light position, as shown in figure 3. The near rectangle is the area cut out of the near plane by the four side planes of the view frustum. Only an object which is inside this near clip volume can cast a shadow onto the near clipping plane. For a comprehensive description see [17].

Silhouette edge extrusion can now be done on graphics hardware to remove the burden from the CPU. The following Cg vertex shader extrudes a vertex $\mathbf{v} = (v_x, v_y, v_z, v_w)$ if $v_w = 0$ otherwise the position is just passed through.

```
float4 lightToVertex = IN.position −
    lightPos;

float m = 1 − IN.position.w;
float4 outx = IN.position*(1−m) +
              lightToVertex*m;
outx.w = IN.position.w;

// transform position to homogeneous clip
    space
OUT.HPOS = mul(ModelViewProj, outx);
```

IN.position is the vertex coordinate and lightPos is the position of the point light. If shaders are used, one has to take care of transforming the vertex position into homogenous clip space, therefore the multiplication with the modelview-projection matrix. To make this approach work correctly, each vertex of the silhouette must be passed twice to the shader. Once with $v_w = 1$ and once with $v_w = 0$. The extrusion for a directional light looks similar.

6 Infinite Projection Matrix

The OpenGL projection matrix is defined as [22]:

$$P = \begin{vmatrix} \frac{2\cdot n}{r-l} & 0 & \frac{r+l}{r-l} & 0 \\ 0 & \frac{2\cdot n}{t-b} & \frac{t+b}{t-b} & 0 \\ 0 & 0 & \frac{-(f+n)}{f-n} & \frac{-2\cdot f\cdot n}{f-n} \\ 0 & 0 & -1 & 0 \end{vmatrix}$$

In this matrix f is the distance from the viewer to the far clip plane, n the distance to the near clip plane and r and l are the respective distances to the left and right clip plane. t and b are the distances to the top and bottom clip plane.

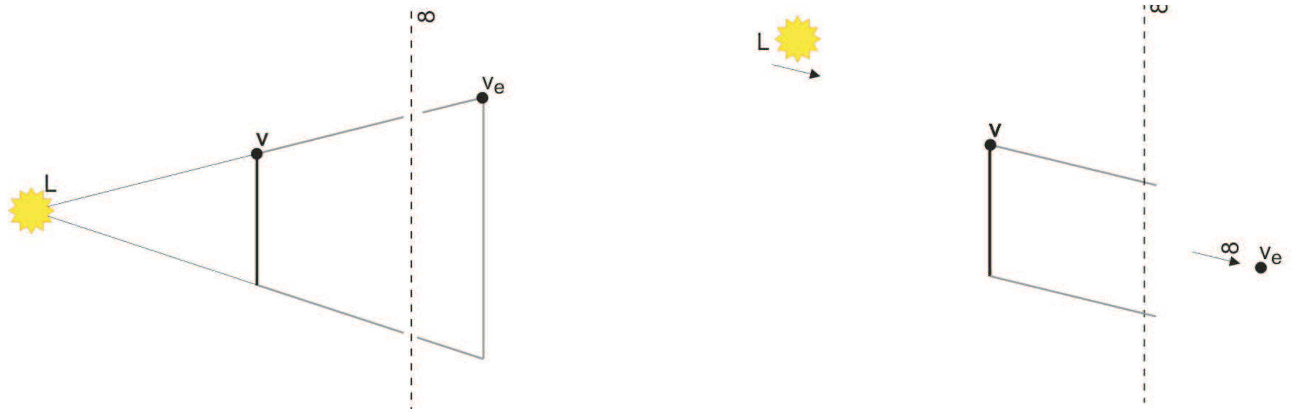


Figure 4: *Silhouette extrusion for a point light (left) and for a directional light (right)*

We can obtain the infinite projection matrix by calculating $P_\infty = \lim_{f\to\infty} P$ which yields

$$P_\infty = \begin{vmatrix} \frac{2\cdot n}{r-l} & 0 & \frac{r+l}{r-l} & 0 \\ 0 & \frac{2\cdot n}{t-b} & \frac{t+b}{t-b} & 0 \\ 0 & 0 & -1 & -2\cdot n \\ 0 & 0 & -1 & 0 \end{vmatrix}$$

An infinite projection matrix reduces the depth buffer precision only marginally as pointed out by [9]. However, if you are concerned about this loss you can use Nvidia's NV_depth_clamp [23] extension. If depth clamping is enabled, the near and far clipping plane are disabled for rasterizating geometry primitives.

7 Rendering

Here I present the necessary steps to render shadow volumes with OpenGL. First we render the scene with enabled depth writes, backface culling and with ambient lighting only (light independent attributes). This guarantees that the depth buffer is initialized with the correct depth values. Afterwards we disable writes to the depth buffer and turn off ambient lighting.

```
glEnable(GL_LIGHTING);
glLightModelfv(GL_LIGHT_MODEL_AMBIENT,
     ambient);
glEnable(GL_DEPTH_TEST);
glDepthFunc(GL_LESS);
glEnable(GL_CULL_FACE);
glCullFace(GL_BACK);

drawScene();

glDepthMask(GL_FALSE);
glLightModelfv(GL_LIGHT_MODEL_AMBIENT,
     zero);
```

The stencil mask has to be calculated separately for each light source.

```
for each light source
{
```

First we clean the stencil buffer, configure the stencil test so that it always passes and disable writes to the color buffer. We will take advantage of two side stencil testing so that we only have to render the shadow volume of each occluder once. Therefore the stencil operation is set to increment and decrement for front- and back-facing polygons respectively – if the depth test fails. Culling is also turned off because front as well as back faces must be rendered at the same time.

```
glClear(GL_STENCIL_BUFFER_BIT);
glEnable(GL_STENCIL_TEST);
glStencilFunc(GL_ALWAYS, 0, ~0);
glStencilMask(~0);

glColorMask(GL_FALSE, GL_FALSE, GL_FALSE,
      GL_FALSE);

glActiveStencilFaceEXT(GL_BACK);
glStencilOp(GL_KEEP, GL_INCR_WRAP_EXT,
     GL_KEEP);
glActiveStencilFaceEXT(GL_FRONT);
glStencilOp(GL_KEEP, GL_DECR_WRAP_EXT,
     GL_KEEP);

glDisable(GL_CULL_FACE);
glEnable(GL_STENCIL_TEST_TWO_SIDE_EXT);
```

Now the shadow volume of each occluder in the scene is rendered. Afterwards culling is turned on and the stencil test is disabled. At that time the stencil buffer holds the correct information about which pixels are in shadow and which aren't.

```
  for each occluder
  {
    renderShadowVolume(occluder);
  }

  glEnable(GL_CULL_FACE);
  glDisable(GL_STENCIL_TEST_TWO_SIDE_EXT);
```

The whole scene is now rendered again. This time the current light is enabled and configured (all light dependent attributes). Stencil testing is configured so that only pixels with a zero stencil value are rendered. Equal depth testing is used so that only visible fragments are updated. Since this pass adds to the ambient scene already in the color buffer, additive blending must be enabled as well as writes to the color buffer. After rendering the scene, blending is disabled and the depth function is restored to less depth testing.

```
  glEnable(light);
  configureLight(light);

  glEnable(GL_BLEND);
  glBlendFunc(GL_ONE, GL_ONE);
  glColorMask(GL_TRUE, GL_TRUE, GL_TRUE,
      GL_TRUE);

  glStencilFunc(GL_EQUAL, 0, ~0);
  glStencilOp(GL_KEEP, GL_KEEP, GL_KEEP);
  glDepthFunc(GL_EQUAL);

  renderScene();

  glDisable(GL_BLEND);
  glDepthFunc(GL_LESS);
}
```

After the above steps have been carried out for all lights, stencil testing is disabled and writes to the depth buffer are enabled.

```
glDisable(GL_STENCIL_TEST);
glDepthMask(GL_TRUE);
```

8 Application: Shadow Profiles

I have succesfully applied shadow volumes in an application for calculating shadow profiles in real time. A shadow profile shows the cast shadow of an object over a specific time period. This is, for example, of concern for architects to find out how long the surroundings are obscured by a building. After providing the required information needed for computing the position of the sun[2] (latitude, date, time) and the time period, the shadow profile is calculated.

[2]See [13] for a description of the calculation

Scene	Number of triangles[a]	~time [ms]
Eiffel Tower	11353 (11155)	4.584
Industry Area	13615 (13585)	7.299
Uniqua Building	182038 (147296)	44.486
Uniqua Building	182038 (182038)	58.666

Table 1: Performance with brute force silhouette detection

[a]First number: total triangles in the scene. Second number: triangles of shadow casting objects

Scene	Number of triangles	~time [ms]
Eiffel Tower	11353 (11155)	4.236
Industry Area	13615 (13585)	5.799
Uniqua Building	182038 (147296)	39.331
Uniqua Building	182038 (182038)	48.872

Table 2: Performance with dual space silhouette detection

The application can detect the silhouette either by brute force or with the above described dual space approach. If the graphics card supports vertex and fragment shaders, silhouette extrusion and per pixel lighting is performed on the GPU. Otherwise the CPU handles the extrusion and standard OpenGL lighting is used. Double sided stencil testing is performed if EXT_stencil_two_side is supported. The z-fail algorithm is only applied if necessary (see section 3).

Figures 5 to 7 show some sample scenes. Table 1 shows the time needed for brute force silhouette detection for each scene and table 2 for dual space silhouette detection, respectively. All measurements were taken on a Pentium 4 3.4Ghz processor with 1GB memory. For each scene a hextree with a fixed depth of four was chosen for the dual space approach.

9 Future Work

The results show that silhouette detection can greatly improve performance. As future work it would be interesting to see how Hough space silhouette finding [24] can further speed up the process. At this time no techniques to reduce fill rate consumption are implemented. Lengyel [17] describes how OpenGLs scissor rectangle support can be used to cut down the fill rate penalty for rendering the shadow volumes. That is because the hardware does not generate fragments outside the scissor rectangle. The scissor rectangle can be applied on a per light basis or per geometry basis, as pointed out by [18]. [10] suggest a depth bounds test for stencil writes. This idea is based on the observation that some depth values can never be in shadow, so incrementing and decrementing the stencil buffer is needless.

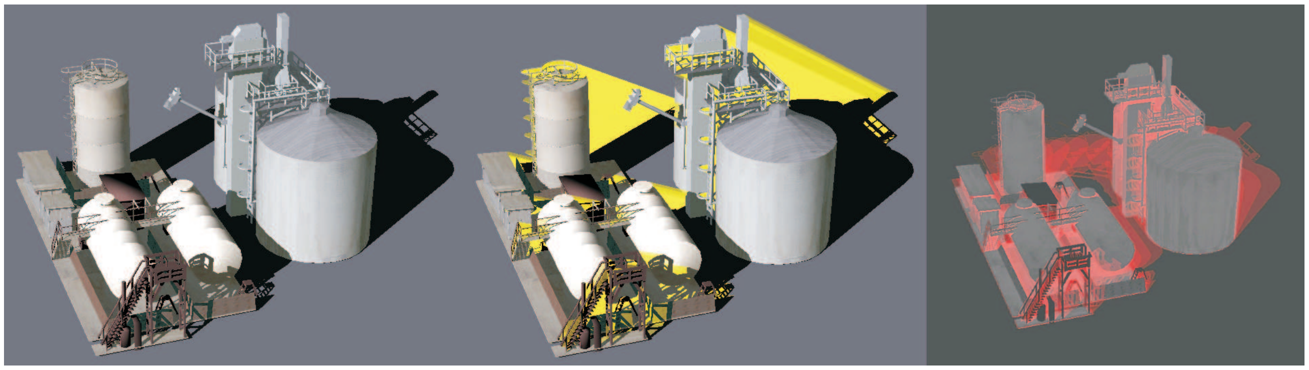


Figure 5: *left: Casted shadows of an industry area located at a latitude of 45.2° north on September 18th at 3pm. middle: Visualization of the shadow volumes (yellow). right: The shadow profile of the scene over a time period of three hours (12pm until 3pm in 30 minutes time steps).*

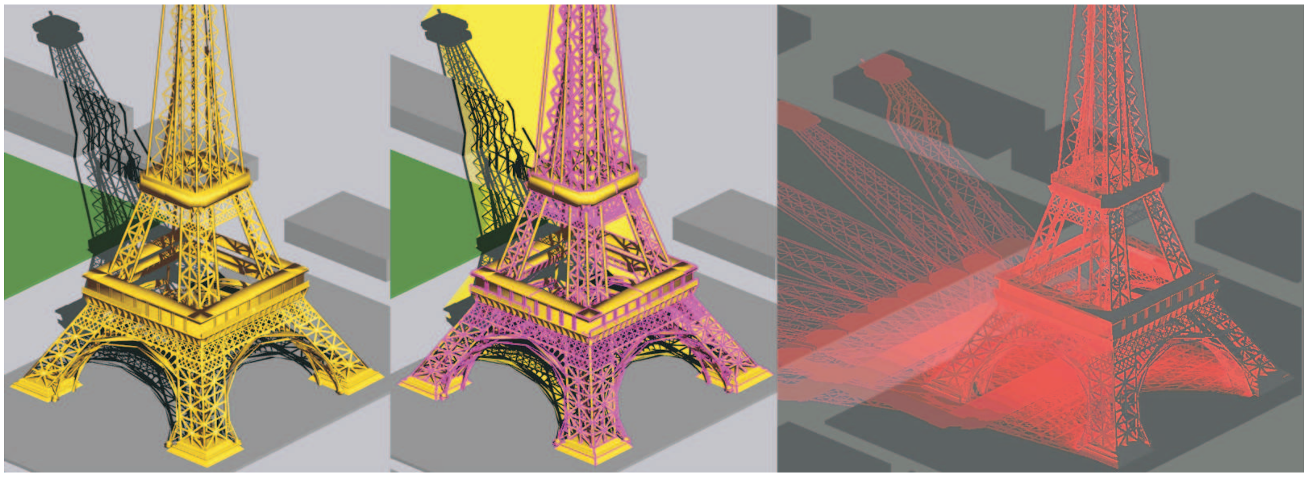


Figure 6: *left: Shadow of the Eiffel Tower in Paris (latitude of 48.8° north) on September 18th at 2pm. middle: Visualization of the shadow volume (yellow) and the silhouette edges (pink). right: Shadow profile over a time period of four hours (10am until 2pm in 30 minutes intervals).*

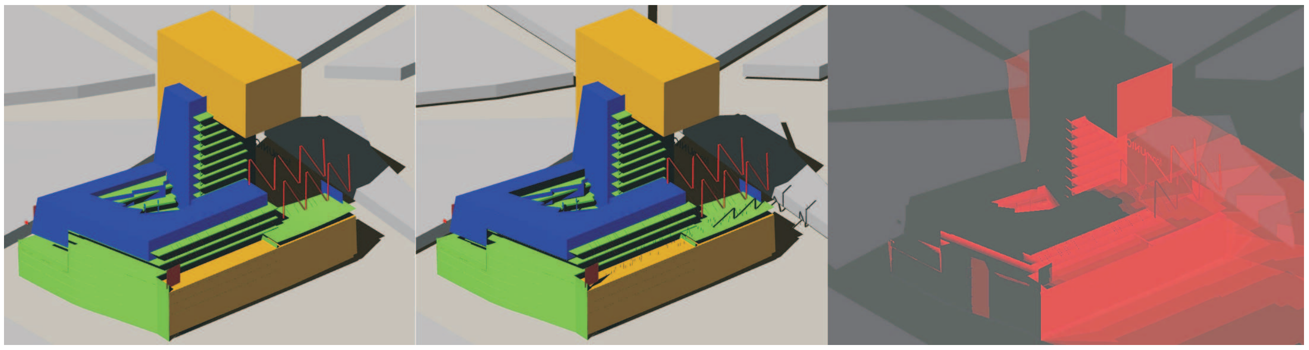


Figure 7: *Proposal for the Uniqua building in Vienna (48.2° north) by Hans Hollein. A color was assigned to each structural component. left: Only the facade (yellow) and the concrete (green) is casting a shadow. middle: The complete building is casting a shadow. right: The shadow profile over a time period of eight hours (9am until 5pm in 1 hour intervals) on September 18th.*

10 Conclusions

In this paper I have presented the necessary steps for a robust implementation of stencil shadow volumes. Stencil shadow volumes suffer mainly from two bottlenecks: (a) fill rate and (b) silhouette detection. The latter was discussed in section 4. Modern graphics hardware can take over computations which formerly had to be performed on the CPU, e.g. silhouette extraction. Code snippets showed how stencil shadows can be implemented with OpenGL. Extensions to OpenGL provide further ways to improve performance.

References

[1] ULF ASSARSSON AND TOMAS AKENINE-MÖLLER, *A Geometry-based Soft Shadow Volume Algorithm using Graphics Hardware*, Siggraph Proceedings **22**, 511–520, 2003, available online: `http://www.cs.lth.se/home/Tomas_Akenine_Moller/pubs/soft_sig2003.pdf`

[2] G. BAREQUET AND C. A. DUNCAN AND M. T. GOODRICH AND S. KUMAR AND M. POP, *Efficient perspective-accurate silhouette computation*, Proceedings of the 15th annual symposium on Computational geometry, 417–418, 1999

[3] HARLEN COSTA BATAGELO AND ILAIM COSTA JUNIOR, *Real-Time Shadow Generation using BSP Trees and Stencil Buffers*, XII Brazilian Symposium on Computer Graphics and Image Processing, 93–102, 1999

[4] JASON BESTIMT AND BRYANT FREITAG, *Real-Time Shadow Casting Using Shadow Volumes*, 1999, available online: `http://www.gamasutra.com/features/19991115/bestimt_freitag_01.htm`

[5] JEREMY BIRN, *Digital Lighting & Rendering*, New Riders, 2006

[6] JOHN CARMACK, *E-Mail to private list*, 2000, available online: `http://developer.nvidia.com/attach/6832`

[7] JOHAN CLAES AND FABIAN DI FIORE AND GERT VANSICHEM AND FRANK VAN REET, *Fast 3D Cartoon Rendering with Improved Quality by Exploiting Graphics Hardware*, Proceedings of Image and Vision Computing New Zealand (IVCNZ), 13–18, 2001, available online: `http://www.cs.utah.edu/npr/papers/Claes_IVCNZ2001.pdf`

[8] F.C. CROW, *Shadow Algorithms for Computer Graphics*, Siggraph Proceedings **11**, 242–248, 1977

[9] CASS EVERITT AND MARK J. KILGARD, *Practical and Robust Stenciled Shadow Volumes for Hardware-Accelerated Rendering*, 2002, available online: `http://developer.nvidia.com/object/robust_shadow_volumes.html`

[10] CASS EVERITT AND MARK J. KILGARD, *Optimized Stencil Shadow Volumes*, Game Developer Conference Presentation, 2003, available online: `http://developer.nvidia.com/object/GDC_2003_Presentations.html`

[11] OPENGL EXTENSION REGISTRY, *EXT_stencil_two_side*, 2003, available online: `http://oss.sgi.com/projects/ogl-sample/registry/EXT/stencil_two_side.txt`

[12] OPENGL EXTENSION REGISTRY, *EXT_stencil_wrap*, 2002, available online: `http://oss.sgi.com/projects/ogl-sample/registry/EXT/stencil_wrap.txt`

[13] GEORG GLAESER, *Der mathematische Werkzeugkasten*, Elsevier, 2006

[14] TIM HEIDMANN, *Real Shadows, Real Time*, Silicon Graphics Inc. **18**, 23–31, 1991

[15] AARON HERTZMANN AND DENIS ZORIN, *Illustrating smooth surfaces*, Proceedings of the 27th annual conference on Computer graphics and interactive techniques, 517–526, 2000, available online: `http://mrl.nyu.edu/publications/illustrating-smooth/hertzmann-zorin.pdf`

[16] HUN YEN KWOON, *The Theory of Stencil Shadow Volumes*, available online: `http://www.gamedev.net/reference/articles/article1873.asp`

[17] ERIC LENGYEL, *The Mechanics of Robust Stencil Shadows*, 2002, available online: `http://www.gamasutra.com/features/20021011/lengyel_01.htm`

[18] ERIC LENGYEL, *Advanced Stencil Shadow and Penumbral Wedge Rendering*, Game Developer Conference Presentation, 2005, available online: `http://www.terathon.com/gdc_lengyel.ppt`

[19] MORGAN MCGUIRE AND JOHN F. HUGHES AND KEVIN T. EGAN AND MARK J. KILGARD AND CASS EVERITT, *Fast, Practical and Robust Shadows*, 2003, available online: `http://developer.nvidia.com/object/fast_shadow_volumes.html`

[20] TOMAS MÖLLER AND ERIC HAINES, *Real-Time Rendering*, A K Peters, 1999

[21] Tomas Akenine-Möller and Ulf Assarsson, *Approximate Soft Shadows on Arbitrary Surfaces using Penumbra Wedges*, Proceedings of the 13th Eurographics workshop on Rendering, 297–306, 2002, available online: `http://www.ce.chalmers.se/~uffe/softshadows_egrw.pdf`

[22] Jackie Neider and Tom Davis and Mason Woo, *OpenGL Programming Guide*, Addison Wesley, 1993

[23] OpenGL Extension Registry, *NV_depth_clamp*, 2003, available online: `http://oss.sgi.com/projects/ogl-sample/registry/NV/depth_clamp.txt`

[24] Matt Olson and Hao Zhang, *Silhouette Extraction in Hough Space*, Eurographics Proceedings **25**, 2006, available online: `http://www.cs.sfu.ca/~haoz/pubs/06_eg_hough.pdf#search=%22silhouette%20extraction%20hough%20space%22`

[25] Ashu Rege, *Shadow Considerations*, available online: `http://download.nvidia.com/developer/presentations/2004/6800_Leagues/6800_Leagues_Shadows.pdf`

[26] Lance Williams, *Casting Curved Shadows on Curved Surfaces*, Siggraph Proceedings **12**, 270–274, 1978, available online: `http://accad.osu.edu/~waynec/history/PDFs/shadowmaps.pdf#search=%22casting%20curved%20surfaces%22`

Günter Wallner
email: gw@autoteles.org

Department of Geometry
University of Applied Arts Vienna

Stručni rad
Prihvaćeno 8. 12. 2006.

MATIJA HERCEG
DRAGO ŠPOLJARIĆ

Primjena *webMathematice* i *LiveGraphics3D* u sfernoj astronomiji

Application of *webMathematica* and *LiveGraphics3D* in Spherical Astronomy

ABSTRACT

This paper describes development of interactive web application designed for coordinate's recalculation within the celestial coordinate systems. Furthermore, this application is also a simple tool for the graphical display of the celestial objects location (coordinates). Recalculation of coordinates and visualization of the celestial coordinate systems capable of magnification, rotation and perspective change makes this interactive application suitable for e-learning.

Key words: celestial coordinate systems, recalculation of coordinates, visualization, webMathematica, LiveGraphics3D, e-learning.

MSC 2000: 97U80, 85-04

Primjena *webMathematice* i *LiveGraphics3D* u sfernoj astronomiji

SAŽETAK

U radu je opisana interaktivna internetska aplikacija namijenjena preračunavanju koordinata u nebeskim koordinatnim sustavima. Aplikacija je ujedno jednostavan alat za grafički prikaz položaja (koordinata) nebeskih tijela uz mogućnost rotacije, povećanja i promjene perspektive grafičkog prikaza. Preračunavanje koordinata i vizualizacija nebeskih koordinatnih sustava omogućuje primjenu ove interaktivne aplikacije u e-obrazovanju.

Ključne riječi: nebeski sferni koordinatni sustavi, preračunavanje koordinata, vizualizacija, webMathematica, LiveGraphics3D, e-obrazovanje.

1 Uvod

Informacijske i komunikacijske tehnologije (ICT) u suvremenim oblicima učenja važno su sredstvo za poboljšanje kvalitete obrazovanja. Ostvarenjem potrebne infrastrukture (tehnička opremljenost, brze veze za pristup internetu i sustavi za upravljanje učenjem) stvoren je temelj za primjenu e-obrazovanja - učenja i podučavanja potpomognutog ICT-eom i internetom [4]. Stručnjaci predviđaju da će se e-obrazovanje na različite načine upotrebljavati u svim vidovima obrazovanja.

Akademske godine 2005/06, tadašnji studenti treće godine Geodetskog fakulteta u Zagrebu, izradili su rad [1] nagrađen Dekanovom nagradom. Na temelju tog rada izrađena je internetska interaktivna (on-line) aplikacija pomoću koje korisnici mogu jednostavno i trenutačno preračunati koordinate u različitim nebeskim koordinatnim sustavima i grafički ih prikazati. Osim toga, naknadno je izrađena i interaktivna aplikacija koja omogućuje vizualizaciju odabranog nebeskog koordinatnog sustava uz mogućnosti rotacije, povećanja, promjene perspektive i pomicanja/mijenjanja položaja nebeskog tijela. Aplikacija je razvijena u *Mathematici* i *webMathematici* i pomoću Java applet-a *LiveGraphics3D*.

Aplikacija je izrađena sa ciljem poboljšanje kvalitete nastave iz geodetske astronomije na Geodetskom fakultetu u Zagrebu.

2 *Mathematica* i *webMathematica*

Za točnost grafičkog prikaza nebeske sfere s traženim kružnicama programski jezik mora sadržavati vektorski način prikaza grafičkih elemenata u trodimenzionalnom prostoru, određene naredbe potrebne za pretvorbu koordinata iz sfernog u Kartezijev koordinatni sustav te mogućnost implementacije grafičkog prikaza i rješenja na internetu.

Mathematica je softver tvrtke Wolfram Research koji u sebi sadrži numeričko i simboličko računalo, grafički sustav, programski jezik, dokumentaciju i napredну mogućnost spajanja s drugim aplikacijama. Jedna od takvih aplikacija je *webMathematica*, još jedan proizvod tvrtke Wolfram Research [5].

webMathematica omogućuje izvršavanje interaktivnih računanja i vizualizacija na internetskim stranicama, a također i brzo stvaranje te distribuiranje rješenja računanja u mreži servera na kojem je postavljena. Nadalje, sadrži mogućnost računanja funkcija za razvijanje tehničkih rješenja koja dopuštaju izradu tehničkog mrežnog servisa koji uključuje numeričke, simboličke i grafičke aplikacije za rješavanje svakodnevnih računalnih problema.

3 Izrada programskog kôda i implementacija na internet

Početnu ideju o prikazu nebeske sfere kao cjelovitog objekta (mogućnost koju podržava programski jezik *Mathematice*, `Sphere[r,m,n]`), zamjenjujemo prikazom nebeske sfere konturnom kružnicom u prostoru, zbog lakše manipulacije dijelovima grafičkog prikaza. Toj kružnici pridružujemo pravac zenita (pravac koji spaja zenit i nadir), pravac nebeske ili svjetske osi, te ravninu nebeskog ekvatora. *Zenit* je točka nebeske sfere, točno iznad motritelja koji se nalazi u središtu nebeske sfere. *Nadir* je točka nebeske sfere dijametralno suprotna zenitu. *Nebeska os* je zamišljena os koja nebesku sferu probada u sjevernom i južnom nebeskom polu, a na kojoj leži Zemljina os rotacije. *Ravnina nebeskog ekvatora* je ravnina okomita na nebesku os i u njoj leži stajalište. [3]

Sve ravnine ili kružnice na grafičkom prikazu dobivene su pomoću trigonometrijskih funkcija, a iscrtane su pomoću malih dužina čije granične točke određujemo pomicanjem kuta na nebeskoj sferi za po jedan stupanj.

Slijedi kôd za iscrtavanje konture horizonta (horizont je velika kružnica nebeske sfere koja nastaje presjekom ravnine koja prolazi stajalištem, a okomita je na pravac zenita).

```
x2=Table[Cos[i*Pi/180]*Cos[0],{i,-1,360}];
y2=Table[Sin[i*Pi/180]*Cos[0],{i,-1,360}];
z2=Table[Sin[0],{i,-1,360}];
horizont1=Transpose[{x2,y2,z2}];
horizont=Graphics3D[{AbsoluteThickness[2],
          RGBColor[0,0.5,0],Line[horizont1]}]
```

U ispisanom kôdu `x2`, `y2` i `z2` su liste koordinata točaka na konturi horizonta. Lista prvih 5 članova x koordinate:

$$\{\cos[\frac{\pi}{180}],1,\cos[\frac{\pi}{180}],\cos[\frac{\pi}{90}],\cos[\frac{\pi}{60}]\}.$$

Lista prvih 5 članova y koordinate:

$$\{-\sin[\frac{\pi}{180}],0,\sin[\frac{\pi}{180}],\sin[\frac{\pi}{90}],\sin[\frac{\pi}{60}]\}.$$

Lista prvih 5 članova z koordinate:

$$\{0,0,0,0,0\}.$$

Vizualizacija je moguća tek kada sve tri koordinate iz sve tri liste spojimo u uređene triplete funkcijom *Transpose*. Horizont iscrtavamo spajanjem tih točaka dužinama. Prikazani su tripleti prvih pet točaka:

$$\{\cos[\frac{\pi}{180}],-\sin[\frac{\pi}{180}],0\},$$
$$\{1,0,0\},$$
$$\{\cos[\frac{\pi}{180}],\sin[\frac{\pi}{180}],0\},$$
$$\{\cos[\frac{\pi}{90}],\sin[\frac{\pi}{90}],0\},$$
$$\{\cos[\frac{\pi}{60}],\sin[\frac{\pi}{60}],0\}.$$

Zadatak je prikazati nebesku sferu s ishodištem u stajališnoj točki, glavne točke, pravce i kružnice nebeske sfere i položaj nebeskog tijela sa zadanim/izračunanim sfernim koordinatama. Koordinate nebeskih tijela mogu biti zadane u različitim koordinatnim sustavima. Ovom aplikacijom moguća su preračunavanja koordinata između horizontalnog (A,z), mjesnog ekvatorskog (t,δ) i nebeskog ekvatorskog (α,δ) koordinatnog sustava.

Malu kružnicu nebeske sfere, paralelnu s ravninom horizonta, nazivamo *almukantaratom*. Sve točke almukantarata jednako su udaljene od točke zenita Z. Kutnu udaljenost između zenita i almukantarata nazivamo *zenitna daljina* z. Veliku kružnicu nebeske sfere koja prolazi kroz zenit Z i nadir Z', a okomita je na horizont nazivamo *vertikal*. Kut između stajališnog meridijana i vertikala (od južne točke horizonta S u smjeru kazaljke na satu) nazivamo *azimutom* A. *Deklinacijska kružnica* je velika kružnica nebeske sfere koja prolazi nebeskim polovima P_N i P_S, a okomita je na nebeski ekvator. Kutnu udaljnost u smjeru zapada uzduž nebeskog ekvatora od stajališnog meridijana do deklinacijske kružnice zovemo *satnim kutom* t. Dnevna paralela je kružnica nebeske sfere paralelna s nebeskim ekvatorom, a kutnu udaljenost od nebeskog ekvatora do dnevne paralele nazivamo *deklinacijom* δ. Kutnu udaljenost mjerenu u suprotnom smjeru od kazaljke na satu uzduž nebeskog ekvatora, od proljetnog ekvinocija (presjecište ekliptike i nebeskog ekvatora) do satne kružnice, nazivamo *rektascenzijom* α. *Stajališni meridijan* je velika kružnica nebeske sfere koja prolazi nebeskim polovima, zenitom i nadirom, najvišom Q i najnižom Q' točkom nebeskog ekvatora te točkom sjevera N i juga S.

Neke kružnice nije bilo jednostavno matematički definirati, na primjer kružnicu satnog kuta. To je izvedeno pomoću središta kružnice i dviju njenih točaka čije koordinate možemo jednostavno trigonometrijski definirati. Ravnina kružnice određena je radij vektorima točaka na kružnici iz njenog središta.

Slijedi dio koda za iscrtavanje kružnice satnog kuta.

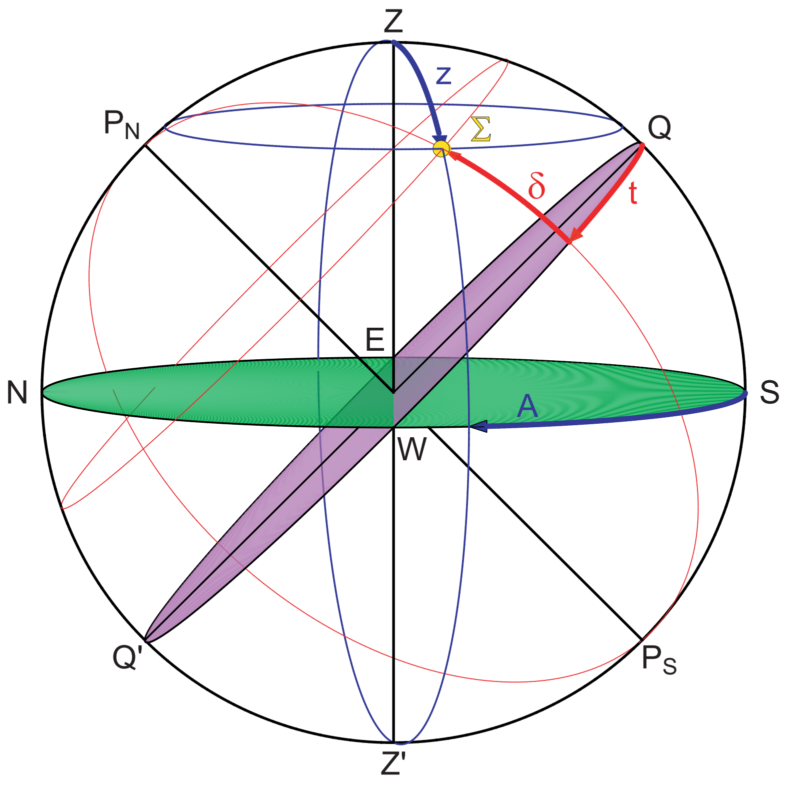
```
n1=Cross[r1,r3];
nm1=Sqrt[(n1[[1]])^2+(n1[[2]])^2+(n1[[3]])^2];
n01=n1/nm1; ar=r1; br=Cross[n01,ar];
ar1=Sqrt[(ar[[1]])^2+(ar[[2]])^2+(ar[[3]])^2];
br1=Sqrt[(br[[1]])^2+(br[[2]])^2+(br[[3]])^2];


a=ar[[1]]*Cos[(i)*Pi/180]+br[[1]]*Sin[(i)*Pi/180];
b=ar[[2]]*Cos[(i)*Pi/180]+br[[2]]*Sin[(i)*Pi/180];
c=ar[[3]]*Cos[(i)*Pi/180]+br[[3]]*Sin[(i)*Pi/180];


x9=Table[a,{i,-1,360}];
y9=Table[b,{i,-1,360}];
z9=Table[c,{i,-1,360}];
satniKUT1=Transpose[{x9,y9,z9}];
satniKUT=
    Graphics3D[{RGBColor[1,0,0],Line[satniKUT1]}];
```

Konačno grafičko rješenje zadatka sadrži ravninu horizonta (zeleno) i ravninu nebeskog ekvatora (ljubičasto), kružnice zadanih (obojene plavom bojom) i računanih veličina (obojene crvenom bojom), mjesni meridijan (to je ujedno i kontura nebeske sfere), pravac zenita (ZZ') i nebesku ili svjetsku os ($P_N P_S$). Podebljane linije označavaju zadane ili računane veličine, uz koje stoje slova koja ih opisuju. Nebeska sfera sadrži oznake za strane svijeta, nebeske polove te položaj zvijezde obojene žutom bojom (Slika 1).

Potrebno je napomenuti da će zadane veličine uvijek biti obojene plavom, a računane veličine crvenom bojom. Stoga će na grafičkom rješenju boja pojedinog elementa (kružnica, dio kružnice, slovo, strelica) ovisiti o vrsti zadatka.



Opis oznaka

- ravnina horizonta
- ravnina nebeskog ekvatora
- almukantarat
- **zenitna daljina (z)**
- vertikal
- **azimut (A)**
- deklinacijska kružnica
- **satni kut (t)**
- dnevna paralela
- **deklinacija (δ)**
- nebesko tijelo (Σ)
- pravac koji spaja zenit (Z) i nadir (Z')
- pravac koji spaja sjeverni (P_N) i južni pol (P_S)
- najviša (Q) i najniža (Q') točka nebeskog ekvatora
- glavne strane svijeta (N, W, E, S)

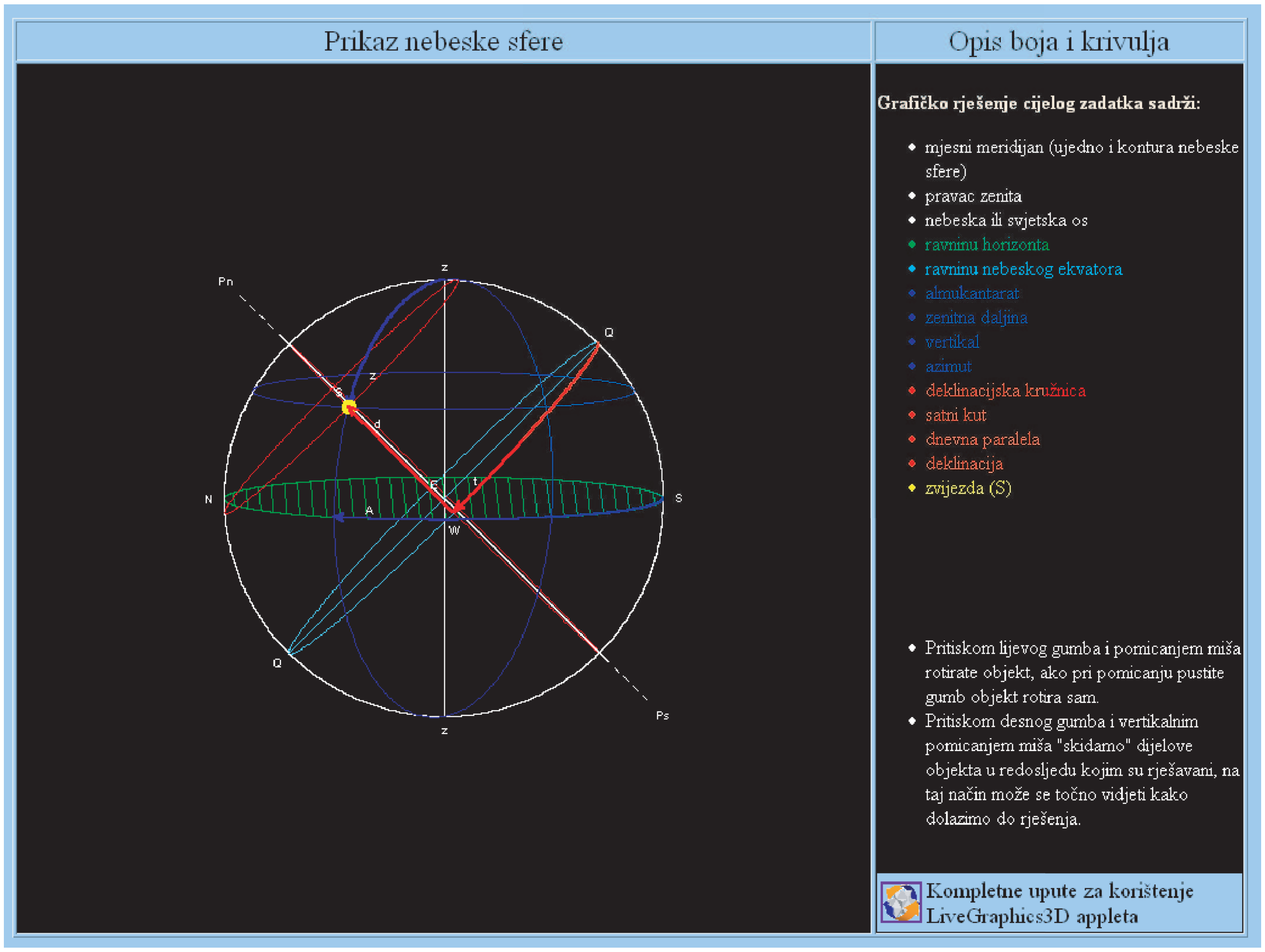
Slika 1

Nakon izrade programskog kôda u *Mathematici* bilo je potrebno ostvariti vizualizaciju preko interneta. Izvršavanje programskog kôda *Mathematice* preko interneta (on-line) moguće je samo uz upotrebu *webMathematice*.

Kako bi programski kôd bio prepoznatljiv *webMathematici*, koja predstavlja vezu između *web* servera i programa *Mathematica*, potrebno ga je preraditi te prebaciti u *jsp* (JavaServer Pages) datoteku. Na taj način omogućujemo serveru da uz pomoć *webMathematice* izdvoji iz stranice matematičke naredbe i proslijedi ih *Mathematici*, koja serveru vraća rezultat. Dobiveni rezultat, prikazan na internetskoj stranici, namijenjen je korisniku.

Aplikacija za preračunavanje koordinata i njihovu vizualizaciju nalazi se na adresi *http://webmath.grad.hr:8180/webMathematica/geodezija/stranica/ga.html*.[1]

Na stranici nalazimo i legendu - opis i objašnjenje boja linija, točaka, slova, šrafura i strelica. Također i upute za rukovanje trodimenzionalnim grafičkim objektom kao npr. povećanje i smanjenje, promjena perspektive i rotacija (Slika 2).



Opis boja i krivulja

Grafičko rješenje cijelog zadatka sadrži:

- mjesni meridijan (ujedno i kontura nebeske sfere)
- pravac zenita
- nebeska ili svjetska os
- ravninu horizonta
- ravninu nebeskog ekvatora
- almukantarat
- zenitna daljina
- vertikal
- azimut
- deklinacijska kružnica
- satni kut
- dnevna paralela
- deklinacija
- zvijezda (S)

- Pritiskom lijevog gumba i pomicanjem miša rotirate objekt, ako pri pomicanju pustite gumb objekt rotira sam.
- Pritiskom desnog gumba i vertikalnim pomicanjem miša "skidamo" dijelove objekta u redosljedu kojim su rješavani, na taj način može se točno vidjeti kako dolazimo do rješenja.

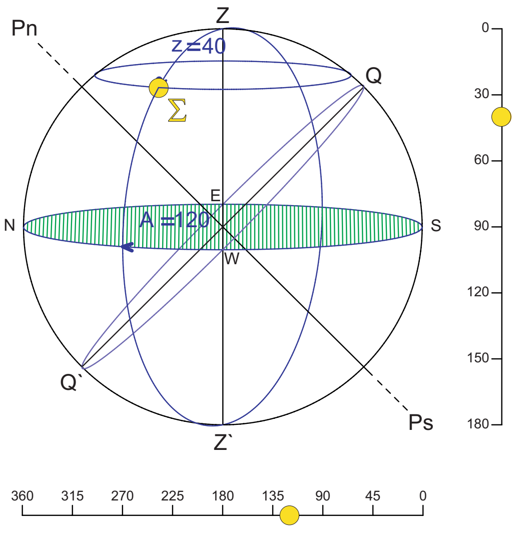
Kompletne upute za korištenje LiveGraphics3D appleta

Slika 2

[1]Autori zahvaljuju dr. sc. Sonji Gorjanc i Vladimiru Beniću, dipl. ing. mat. s Građevinskog fakulteta Sveučilišta u Zagrebu koji su im omogućili realizaciju projekta na web serveru njihovog fakulteta.

4 Vizualizacija nebeskih koordinatnih sustava

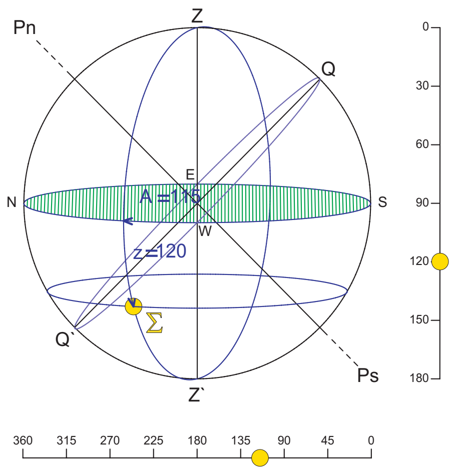
Za vizualizaciju pojedinačnih nebeskih koordinatnih sustava (bez preračunavanja koordinata) upotrebljavan je Java applet *LiveGraphics3D*. S njime možemo ubrzati proces stvaranja zahtjevnijih interaktivnih prikaza te definirati kompleksne manipulacije nad njima, bez potrebe kreiranja grafičkog sadržaja u Javi, Flash-u ili nekom drugom programskom jeziku.



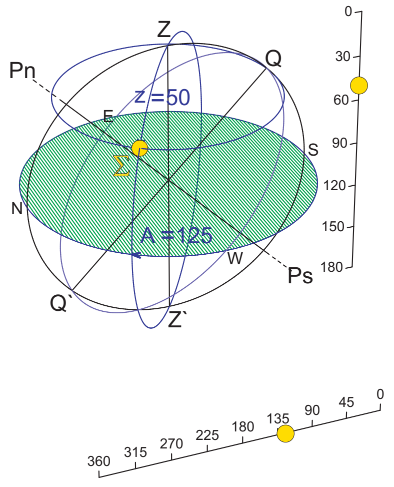
Slika 3a

Statičan prikaz odabranog koordinatnog sustava (Slika 3a) nije osobito zanimljiv. Ono što ga čini "živim" (Slike 3b, 3c i 3d) je mogućnost pomicanja točke (koja predstavlja zvijezdu) neovisno o cijelom prikazu, ali ovisno o kružnicama koje ta točka određuje (vertikal i almukantarat). Sljedeće kontrole to omogućuju ("Click" predstavlja pritisak lijevog gumba miša):

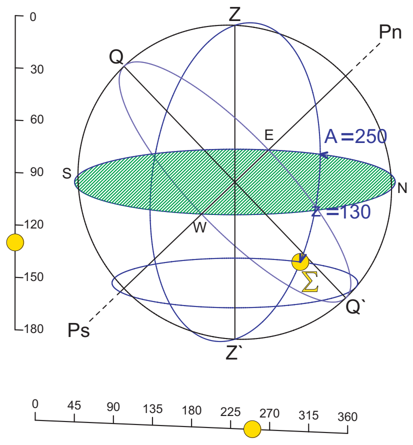
- Click na žutu točku i njenim pomicanjem mijenjamo položaj svih kružnica koje ovise o položaju te točke
- Click negdje drugdje i pomicanjem miša rotiramo cijeli prikaz
- Puštanjem gumba miša dok ga pomičemo dovodimo cijeli prikaz u rotaciju; click bilo gdje na prikazu i rotacija prestaje
- Pritiskom Shift tipke, click i vertikalnim pomicanjem miša povećavamo i smanjujemo prikaz
- Pritiskom Shift tipke, click i horizontalnim pomicanjem miša rotiramo prikaz oko osi okomite na ekran
- Pritiskom tipke Home vraćamo na originalni prikaz



Slika 3b



Slika 3c



Slika 3d

Pisanje programa u bilo kojem softveru koji omogućava ovakvu "živu" vizualizaciju bilo bi vrlo složeno. Proces stvaranja *LiveGraphics3D* prikaza ne zahtijeva znanje Jave. Bitno je napomenuti da *Mathematica* nije potrebna za samu kreaciju *LiveGraphics3D* appleta, jer se ulazni podaci mogu i ručno upisati. No, pomoću *Mathematice* lakše je generirati ulazne podatke za složenije objekte. Da bi izradili takav applet potrebno je datoteku `live.jar` sa stranice "LiveGraphics3D Homepage" staviti u isti direktorij kao i HTML datoteku koju smo prethodno kreirali. HTML datoteka treba sadržavati, osim izgleda stranice, i potreban računalni kôd koji će omogućiti prikaz appleta [2].

```
<applet archive="live.jar" code="Live.class"
   width="500" height="500">
<param name="INPUT_FILE"
   value="pomicanje_fi_z_A.lg3d"/>
<param name="INDEPENDENT_VARIABLES"
   value="{xK -> 0.383022,
   yK -> 0.663414,
   zK -> 0.642788}" />
<param name="DEPENDENT_VARIABLES"
   value="{
   z -> -(ArcSin[zK]*180/Pi - 90),
   A -> ArcCos[xK/(Cos[(90 + z)*Pi/180])]*180/Pi,
   zVrijednost-> Round[10*z]/10,
   AVrijednost->Round[10*A]/10}" />
</applet>
```

5 Zaključak

Pretraživanje i prikupljanje informacija, rješavanje problema i samostalno učenje oslanjati će se u budućnosti najvećim dijelom na internet. Stoga je vrlo važna primjena informacijske tehnologije (ICT) u znanstvenim, nastavnim i drugim aktivnostima.

Razvoj aplikacije za preračunavanje astronomskih koordinata i grafički prikaz rješenja, uz mogućnost rukovanja grafičkim prikazom (rotacija, povećanje, promjena perspektive i drugo), te "živom" vizualizacijom nebeskih koordinatnih sustava omogućuje primjenu ove interaktivne aplikacije u e-obrazovanju. Precizno preračunavanje koordinata u nebeskim koordinatnim sustavima potrebno je i profesionalnim astronomima ali i naprednim amaterima.

Literatura

[1] HERCEG, M., KELEMINEC, S., VIHER, D., *Preračunavanje koordinata u nebeskim koordinatnim sustavima i njihova vizualizacija preko interneta*, Zagreb, 2006.

[2] *Journal of Online Mathematics and its Applications* http://mathdl.maa.org/mathDL/4/?pa=content& sa= viewDocument& nodeId=1143

[3] ROŠA, D., ŠPOLJARIĆ, D., *Astronomski rječnik*, BOLID, 86, (1/2001), Zvjezdarnica Zagreb - Zagrebački astronomski savez, Zagreb, 2001.

[4] TUTEK, Ž., *E-učenje - vežite se, polijećemo!*, Ekscentar, 8, (2006), 120-121.

[5] *webMATHEMATICA 2* http://www.wolfram.com/products/webmathematica

Matija Herceg
e-mail: mherceg@geof.hr

Drago Špoljarić
e-mail:drago.spoljaric@geof.hr

Geodetski fakultet Sveučilišta u Zagrebu
Kačićeva 26, 10000 Zagreb

The art of Tamás F. Farkas

http://www.farkas-tamas.hu/

Tamas F. Farkas has been conducting researches on the boarder between non-traditional geometry and spatial visualisation for 30 years.

His art opened new vistas into an impossible world, partly in higher dimensions, partly in geometric structures not realisable even in higher dimensions. Making visible spaces, that are non-Euclidean, beyond-traditional geometries, multi-dimensional, and not realisable in the real world, his works attract the interest not only of geometry, but also of physics and crystallography, i.e., of sciences searching for the structure of matter

His graphical life-work follows individual paths, he did not join any designated artistic movement, school. The roots of his graphical works go back to different directions. They are fed on partly from the arts (e.g., Escher), partly from the sciences (e.g., multidimensional geometry, the world of abstract symmetries).

His works represent the joint beauty of human thinking and manual creativity. His graphical world is not one of simple play with forms and colours, rather it is built by grave scientific regularities.

The shape, composed of continuous quadratic prisms, returning into itself, partly is a self-imposed visual delimitation, a self-chosen basic unit of a set symbols, partly it determines the nearly infinite abundance of regularities, what he is able to depict on the plane, within and beyond the limits of traditional geometry.

His form- world is composed of one of more (2, 3, 4, 6) continuous, closed quadratic prism lines woven into themselves, each returning into itself. These lines, woven also into each other, compose a system characterised by an internal regularity. Applying the regularities developed by himself, he creates more and more complicated structures, within the limits of the representation in 2 dimensions. At the same time, his forms seem to emerge from the plane into the space before the spectator's eyes. These complicated structures keep their perspicuity and can easily be surveyed by the general observer, because order and symmetry prevail in them. He applies most often the 2-, 3-, 4- and 6- fold rotational symmetries, and in certain cases mirror-symmetry from among the possible geometrical symmetry transformations.

Another cluster of Tamás F. Farkas' works is manifested in architectural-like graphics. He follows the road paved by the impossible building representations of M.C. Escher... In his graphics there appear clear structures, without any reference to a real environment. The pure geometric shapes make him free to construct a richer form-world. Thus he operates with a wide set of symmetry transformations, mainly rotations in multiple dimensions. The result is a fantastic beautiful, imaginary world, much more variegated that our real one, built of impossible structures. A third cluster of his works, that deserve special mentioning here, is a spatial world, whose various structures are formed by the rotation of virtual cubic units.

Any catalogue or exhibition can display only a selection from among the spatial variations and colour world of his graphical units. This beauty, painted by him on canvas, has been admired in many exhibitions both in his native country, Hungary, and the worldwide, from Washington, D.C. to Israel, and from Japan to Italy.

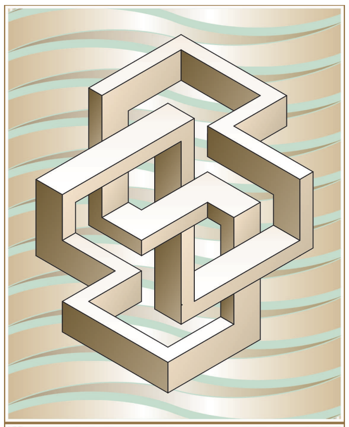
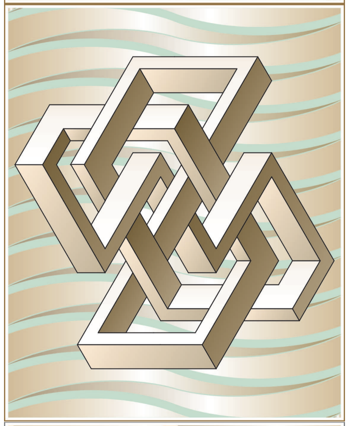
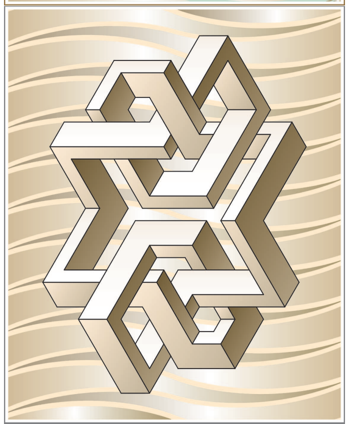
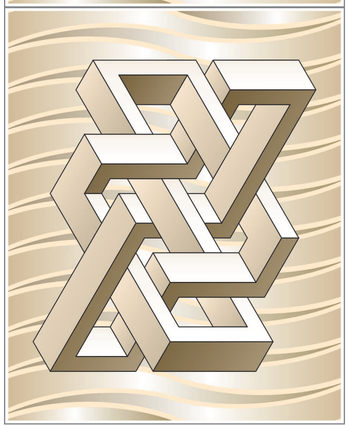
The different graphics, hyperspace structures by Farkas, being either stairs-like or constructed by line tracing, allow many interpretations. They were presented in two books dedicated to M.C. Escher (ed. D. Schattschneider) and in two volumes on visual illusions (by A . Seckel). Several issues of the journal Symmetry have been illustrated by his graphics. Logos of the L'Oreal Art and Science Foundation, the International Symmetry Foundation, and the Symmetry Festival series are designed by T.F. Farkas.

According to the common belief, this world is too complicated to be represented simply, not even to attribute an internal structure to the represented objects. The pictorial world of Tamás F. Farkas breaks this taboo.

His images preserve openness for new developments in science. They evoke associations in the mind. They generate thoughts in laypeople and inspire ideas in specialist spectators.

He provides us with the appearent illusions that we understand something of nature, although we are 'only' enjoying his art.

Extracts from the text by GYÖRGY DARVAS

ISSN 1331-1611

9 771331 161005