In many applications, the vertices of the given triangle and the given point $\mathbf{X}$ can be given in homogeneous coordinates. Let us explore how the barycentric coordinates could be computed in this case.

The linear system of equations for the barycentric coordinates can be rewritten as:

$$a_1 \frac{x_1}{w_1} + a_2 \frac{x_2}{w_2} + a_3 \frac{x_3}{w_3} = \frac{x}{w}$$
$$a_1 \frac{y_1}{w_1} + a_2 \frac{y_2}{w_2} + a_3 \frac{y_3}{w_3} = \frac{y}{w}$$
$$a_1 + a_2 + a_3 = 1$$

where: $\mathbf{x}_i = [x_i, y_i : w_i]^T$ represents the i-th vertex triangle in the homogeneous coordinates and $\mathbf{x} = [x, y, w]^T$ is the given point in the homogeneous coordinates.

We can multiply the linear system by $w \neq 0$, $w_i \neq 0 \quad i = 1,...,3$ and substitute:

$$b_1 = -a_1 w_2 w_3 w \qquad b_2 = -a_2 w_1 w_3 w$$
$$b_3 = -a_3 w_1 w_2 w \qquad b_4 = w_1 w_2 w_3 w$$

Thus we get:

$$b_1 x_1 + b_2 x_2 + b_3 x_3 + b_4 x = 0$$
$$b_1 y_1 + b_2 y_2 + b_3 y_3 + b_4 y = 0$$
$$b_1 w_1 + b_2 w_2 + b_3 w_3 + b_4 w = 0$$

and in the matrix notation:

$$\begin{bmatrix} x_1 & x_2 & x_3 & x \\ y_1 & y_2 & y_3 & y \\ w_1 & w_2 & w_3 & w \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix} = \mathbf{0}$$

We are looking for a vector $\boldsymbol{\tau}$ that satisfies the following equation:

$$\boldsymbol{\tau}^T \mathbf{b} = 0$$

where: the vector $\boldsymbol{\tau}$ is defined as $\boldsymbol{\tau} = [\tau_1, \tau_2, \tau_3 : \tau_4]^T$

Then the solution is defined as

$$\det \begin{vmatrix} \tau_1 & \tau_2 & \tau_3 & \tau_4 \\ x_1 & x_2 & x_3 & x \\ y_1 & y_2 & y_3 & y \\ w_1 & w_2 & w_3 & w \end{vmatrix} = 0$$

and we can formally write

$$\mathbf{b} = \boldsymbol{\xi} \times \boldsymbol{\eta} \times \mathbf{w}$$

where: $\mathbf{b} = [b_1, b_2, b_3, b_4]^T \quad \boldsymbol{\xi} = [x_1, x_2, x_3, x\ ]^T$

$\boldsymbol{\eta} = [y_1, y_2, y_3, y\ ]^T \quad \mathbf{w} = [w_1, w_2, w_3, w\ ]^T$

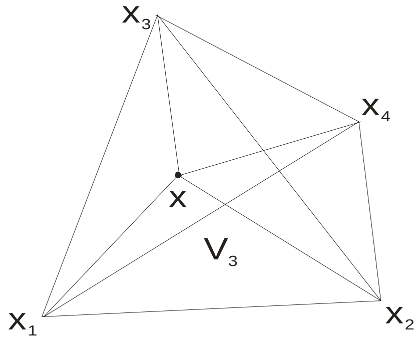
Of course, the conditions in the case that the point is inside the given triangle are slightly more complex, and the condition $0 \le a_i \le 1 \ \ i = 1,...,3$ can be expressed by the following criteria:

$$0 \le (-b_1 : w_2 w_3 w\ ) \le 1$$
$$0 \le (-b_2 : w_1 w_3 w\ ) \le 1$$
$$0 \le (-b_3 : w_1 w_2 w\ ) \le 1$$

This means that the barycentric coordinates can be computed **without using the division operation** even if the vertices of the given triangle and the point **x** are given in homogeneous coordinates. Therefore the approach presented here is more robust than the direct computation, i.e. normalizing the vertices and point coordinates into Euclidean coordinates and standard barycentric coordinates computation. In addition, the test if a point is inside the given triangle is consequently more robust.

Of course, there is a natural question: is it possible to extend the above mentioned approach to the E^3 case?

Let us consider the E^3 case, where the "point in a tetrahedron" test is similar to the "point in a triangle" test in E^2, see Fig.7.



Barycentric coordinates in E^3

Figure 7

It can be seen that the barycentric coordinates are given as

$$a_1 X_1 + a_2 X_2 + a_3 X_3 + a_4 X_4 = X$$
$$a_1 Y_1 + a_2 Y_2 + a_3 Y_3 + a_4 Y_4 = Y$$
$$a_1 Z_1 + a_2 Z_2 + a_3 Z_3 + a_4 Z_4 = Z$$
$$a_1 + a_2 + a_3 + a_4 = 1$$

It is useful to know that

$$a_i = \frac{V_i}{V} \quad \text{i} = 1,...,3$$

where: V is the volume of the given tetrahedron and V_i is the volume of the i-th sub-tetrahedron.

The non-homogeneous system of linear equations can be transformed into a homogeneous linear system of equations

$$b_1X_1 + b_2X_2 + b_3X_3 + b_4X_4 + b_5X = 0$$
$$b_1Y_1 + b_2Y_2 + b_3Y_3 + b_4Y_4 + b_5Y = 0$$
$$b_1Z_1 + b_2Z_2 + b_3Z_3 + b_4Z_4 + b_5Z = 0$$
$$b_1 + b_2 + b_3 + b_4 + b_5 = 0$$

where: $b_5 \neq 0$ and $b_i = -a_i b_5 \quad i = 1,...,4$

Rewriting this system in matrix form, we get

$$\begin{bmatrix} X_1 & X_2 & X_3 & X_4 & X \\ Y_1 & Y_2 & Y_3 & Y_4 & Y \\ Z_1 & Z_2 & Z_3 & Z_4 & Z \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \end{bmatrix} = \mathbf{0}$$

i.e.

$$\mathbf{B}\,\mathbf{b} = \mathbf{0} \qquad or \qquad [\mathbf{A} \mid \mathbf{X}][\mathbf{b}] = \mathbf{0}$$

where: $\mathbf{b} = [b_1, b_2, b_3, b_4 : b_5]^T$, $\mathbf{X} = [X, Y, Z : 1]^T$,

$$\mathbf{A} = \begin{bmatrix} X_1 & X_2 & X_3 & X_4 \\ Y_1 & Y_2 & Y_3 & Y_4 \\ Z_1 & Z_2 & Z_3 & Z_4 \\ 1 & 1 & 1 & 1 \end{bmatrix} \text{ and } \mathbf{B} = [\mathbf{A} \mid \mathbf{X}]$$

Again, we are looking for a vector $\boldsymbol{\tau}$ that satisfies the equation

$$\boldsymbol{\tau}^T \mathbf{b} = 0$$

where: $\boldsymbol{\tau} = [\tau_1, \tau_2, \tau_3, \tau_4 : \tau_5]^T$

The equation can be expressed using a determinant form as:

$$\det \begin{vmatrix} \tau_1 & \tau_2 & \tau_3 & \tau_4 & \tau_5 \\ X_1 & X_2 & X_3 & X_4 & X \\ Y_1 & Y_2 & Y_3 & Y_4 & Y \\ Z_1 & Z_2 & Z_3 & Z_4 & Z \\ 1 & 1 & 1 & 1 & 1 \end{vmatrix} = 0$$

It can be seen that we can formally write again:

$$\mathbf{b} = \boldsymbol{\xi} \times \boldsymbol{\eta} \times \boldsymbol{\zeta} \times \mathbf{w}$$

where: $\mathbf{b} = [b_1, b_2, b_3, b_4 : b_5]^T$ $\boldsymbol{\xi} = [X_1, X_2, X_3, X_4 : X]^T$

$\boldsymbol{\eta} = [Y_1, Y_2, Y_3, Y_4 : Y]^T$ $\boldsymbol{\zeta} = [Z_1, Z_2, Z_3, Z_4 : Z]^T$

$\mathbf{w} = [1, 1, 1, 1 : 1]^T$

This means that the barycentric coordinates of the point $\mathbf{X}$ are given as:

$$a_1 = -\frac{b_1}{b_5}, \quad a_2 = -\frac{b_2}{b_5},$$
$$a_3 = -\frac{b_3}{b_5}, \quad a_4 = -\frac{b_4}{b_5}$$

or if we use the Plücker coordinates notation, they are given as

$$a_i = (-b_i : b_5) \quad i = 1,...,4 .$$

The given point $\mathbf{X}$ is inside the given tetrahedron **if and only if** $0 \le a_i \le 1$, i = 1,…,4.

This condition can be expressed by the following sequence

if $b_5 > 0$ **then** $0 \le -b_i \le b_5$

else $b_5 \le -b_i \le 0$

If $b_5 = 0$, the tetrahedron is degenerated to a triangle or to a line segment or to a point, i.e. singular cases that can be correctly detected.

Let us again consider a case when the tetrahedron vertices and the given point are in homogeneous coordinates.

The linear system of equations can be rewritten as:

$$a_1\frac{x_1}{w_1} + a_2\frac{x_2}{w_2} + a_3\frac{x_3}{w_3} + a_4\frac{x_4}{w_4} = \frac{x}{w}$$
$$a_1\frac{y_1}{w_1} + a_2\frac{y_2}{w_2} + a_3\frac{y_3}{w_3} + a_4\frac{y_4}{w_4} = \frac{y}{w}$$
$$a_1\frac{z_1}{w_1} + a_2\frac{z_2}{w_2} + a_3\frac{z_3}{w_3} + a_4\frac{z_4}{w_4} = \frac{z}{w}$$
$$a_1 + a_2 + a_3 + a_4 = 1$$

where: $\mathbf{x}_i = [x_i, y_i, z_i : w_i]^T$ represents the i-th vertex coordinates in the homogeneous coordinates.

We can multiply the linear system of equations by $w \neq 0$, $w_i \neq 0 \quad i = 1,...,4$ and substitute

$$b_1 = -a_1 w_2 w_3 w_4 w \qquad b_2 = -a_2 w_1 w_3 w_4 w$$
$$b_3 = -a_3 w_1 w_2 w_4 w \qquad b_4 = -a_4 w_1 w_2 w_3 w$$
$$b_5 = w_1 w_2 w_3 w_4$$

This results into a standard homogeneous linear system:

$$b_1x_1 + b_2x_2 + b_3x_3 + b_4x_4 + b_5x = 0$$
$$b_1y_1 + b_2y_2 + b_3y_3 + b_4y_4 + b_5y = 0$$
$$b_1z_1 + b_2z_2 + b_3z_3 + b_4z_4 + b_5z = 0$$
$$w_1b_1 + w_2b_2 + w_3b_3 + w_4b_4 + w\,b_5 = 0$$

that can be expressed in the matrix form as:

$$\begin{bmatrix} x_1 & x_2 & x_3 & x_4 & x \\ y_1 & y_2 & y_3 & y_4 & y \\ z_1 & z_2 & z_3 & z_4 & z \\ w_1 & w_2 & w_3 & w_4 & w \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \end{bmatrix} = \mathbf{0}$$

We are looking for a vector $\boldsymbol{\tau}$ that satisfies the equation

$$\boldsymbol{\tau}^T \mathbf{b} = 0$$

where: the vector $\boldsymbol{\tau} = [\tau_1, \tau_2, \tau_3, \tau_4 : \tau_5]^T$ is defined as

$$\det\begin{vmatrix}\tau_1 & \tau_2 & \tau_3 & \tau_4 & \tau_5\\ x_1 & x_2 & x_3 & x_4 & x\\ y_1 & y_2 & y_3 & y_4 & y\\ z_1 & z_2 & z_3 & z_4 & z\\ w_1 & w_2 & w_3 & w_4 & w\end{vmatrix}=0$$

It can be seen that we can formally write:

$$\mathbf{b}=\boldsymbol{\xi}\times\boldsymbol{\eta}\times\boldsymbol{\zeta}\times\mathbf{w}$$

where: $\mathbf{b}=\left[b_1,b_2,b_3,b_4:b_5\right]^T$ $\boldsymbol{\xi}=\left[x_1,x_2,x_3,x_4:x\right]^T$

$\boldsymbol{\eta}=\left[y_1,y_2,y_3,y_4:y\right]^T$ $\boldsymbol{\zeta}=\left[z_1,z_2,z_3,z_4:z\right]^T$

$\mathbf{w}=\left[w_1,w_2,w_3,w_4:w\right]^T$

The conditions – if the point is inside the given triangle – are slightly more complex and the condition $0\le a_i\le 1 \;\; i=1,\ldots,4$ can be expressed by the following criteria:

$$0\le\left(-b_1:w_2w_3w_4w\right)\le 1$$
$$0\le\left(-b_2:w_1w_3w_4w\right)\le 1$$
$$0\le\left(-b_3:w_1w_2w_4w\right)\le 1$$
$$0\le\left(-b_4:w_1w_2w_3w\right)\le 1$$

It is worth noting that the equations for the computation of barycentric coordinates given above can be simplified for special cases, e.g. if the tetrahedron vertices are expressed in the Euclidean coordinates or the given point $\mathbf{x}$ is expressed in the Euclidean coordinates. Such simplifications will increase the speed of computation significantly without compromising the robustness of the computation. Nevertheless, the resulting barycentric coordinates are generally in the projective space, i.e. the homogeneous coordinate is not equal to '1' in general.

5.2 Dual Barycentric Coordinates

Barycentric coordinates of a point $\boldsymbol{x}=[x,y:w]^T$ in the triangle given by points $\boldsymbol{x}_1,\boldsymbol{x}_2,\boldsymbol{x}_3$ in E^2 can be computed directly using homogeneous coordinates as $\tilde{\boldsymbol{x}}\times\tilde{\boldsymbol{y}}\times\tilde{\boldsymbol{w}}$,

where: $\tilde{\boldsymbol{x}}=[x_1,x_2,x_3,x]^T$, $\tilde{\boldsymbol{y}}=[y_1,y_2,y_3,y]^T$, $\tilde{\boldsymbol{x}}=[w_1,w_2,w_3,w]^T$

$$\tilde{\boldsymbol{x}}\times\tilde{\boldsymbol{y}}\times\tilde{\boldsymbol{w}}=det\begin{pmatrix}\boldsymbol{i} & \boldsymbol{j} & \boldsymbol{k} & \boldsymbol{l}\\ x_1 & x_2 & x_3 & x\\ y_1 & y_2 & y_3 & y\\ w_1 & w_2 & w_3 & w\end{pmatrix}=[\xi_1 \quad \xi_2 \quad \xi_3 \quad \xi_w]^T$$

where: $\lambda_i=-\,\xi_i/\xi_w,\, i=1,\ldots,3$ [24].

The P area of a triangle given by three points in E^2 can be easily computed as

$$P=\frac{1}{2}\boldsymbol{x}_1^T.(\boldsymbol{x}_2\times\boldsymbol{x}_3)/(w_1w_2w_3)=det\begin{pmatrix}x_1 & x_2 & x_3\\ y_1 & y_2 & y_3\\ w_1 & w_2 & w_3\end{pmatrix}\Big/(w_1w_2w_3)$$

We can rewrite the result in the projective notation as "projective" scalar value as:

$$P=[det\begin{pmatrix}x_1 & x_2 & x_3\\ y_1 & y_2 & y_3\\ w_1 & w_2 & w_3\end{pmatrix}:w_1w_2w_3]^T$$

Similarly a volume of a tetrahedron given by four points in E^3 can be computed as

$$V=\frac{1}{6}\boldsymbol{x}_1^T.(\boldsymbol{x}_2\times\boldsymbol{x}_3\times\boldsymbol{x}_4)/(w_1w_2w_3w_4)$$

It means that the projective formulation is simple and matrix-vector GPU architecture supports fast computations without using division operation, as the result can be represented by homogeneous coordinates, in general.

As the principle of duality is valid, one could ask: *What is a "dual" value G to a computation of the area P if the triangle is given by three lines in the "normalized" form, e.g.* $\boldsymbol{a}_1^T.(\boldsymbol{a}_2\times\boldsymbol{a}_3)$ *instead of three points?*

$$G=\boldsymbol{a_1^T}.(\boldsymbol{a_2}\times\boldsymbol{a_3})=det\begin{pmatrix}a_1 & a_2 & a_3\\ b_1 & b_2 & b_3\\ c_1 & c_2 & c_3\end{pmatrix}=det\begin{pmatrix}cos\alpha_1 & cos\alpha_2 & cos\alpha_3\\ sin\alpha_1 & sin\alpha_2 & sin\alpha_3\\ d_1 & d_2 & d_3\end{pmatrix}$$

It can be seen that we can apply some transformations so that one vertex of the given triangle is in the origin and the line $\boldsymbol{a}_1$ is on the axis x, the edge $\boldsymbol{a}_2$ passes the origin and line $\boldsymbol{a}_3$ is in the general position.

$$G=(\boldsymbol{T}\boldsymbol{a}_1)^T(\boldsymbol{T}^{-1})^T(\boldsymbol{a}_2\times\boldsymbol{a}_3)/det(\boldsymbol{T})=\boldsymbol{a}_1^T\boldsymbol{T}^T(\boldsymbol{T}^{-1})^T(\boldsymbol{a}_2\times\boldsymbol{a}_3)/det(\boldsymbol{T})=\boldsymbol{a}_1^T(\boldsymbol{a}_2\times\boldsymbol{a}_3)/det(\boldsymbol{T})$$

As for the "standard" transformations $\det(\boldsymbol{T})=1$ and we can write:

$$G=det\begin{pmatrix}1 & cos\alpha_2 & cos\alpha_3\\ 0 & sin\alpha_2 & sin\alpha_3\\ 0 & 0 & d_3\end{pmatrix}=d_3sin\alpha_2=d_3.a/(2R)=P/R$$

It can be seen that $G=d_3sin\alpha_2=P/R$, where: a is the length of the line segment on $\boldsymbol{a}_3$ and R is a radius of the circumscribing circle. It can be seen that the value G can be used as criterion for a quality triangular meshes.

Of course, we have to prove that the proposed transformation of the given triangle is invariant to the G value. As $det(\boldsymbol{T}) = 1$ for translation and rotation operations, those transformations are invariant and value G is not changed by those transformations. The value G has a property of a distance, i.e. it is in measured in [m], in general.

In geometric modeling a skewnees factor S is used for quality evaluation of triangular meshes

$$S = 1 - 2r/R$$

where r is a radius of the inscribed circle.

It seems to that the value G can be used for an effective evaluation for quality of triangular meshes in E^2 or tetrahedron meshes in E^3.

6 Conclusion

This paper briefly described some problems in numerical computations, advantages of the projective space representation use and some well known disasters caused by impropriate use on numerical computations. Geometrical applications and computational methods require robust algorithms. Some above described principles have been applied recently, e.g. in clipping algorithms [23], [34], [39].

The projective space representation and reformulation of geometrical problems leads to more robust algorithms and simpler formulations as shown. The matrix-vector operations lead to more compact algorithms and due to the today's hardware also to computation acceleration, especially if GPU is used.

7 Acknowledgments

The author would like to express his thanks to MSc. students and colleagues at the University of West Bohemia, Plzen and to Dr.Rongjinang Pan, Shandong University, China for their critical and constructive comments and suggestions for their suggestions that helped to improve the manuscript.

This work was supported by the Ministry of Education – projects No.ME10060 and LH12181.

References:

[1] Cuyt,A., Verdonk,B., Becuwe,S., Kuterna,P.: A remarkable Example of Catastrophics Cancellation Unraveled, Computing 66, pp.309-320, 2011

[2] Fu,H., Mencer,O., Luk,W.: Comparing Floating-point and Logarithmic Number Representations for Reconfigurable Acceleration

[3] Gibson,C.G., Hunt,K.H.: Geometry of Screw Systems, Mechanical Machine Theory, Vol.12, pp.1-27, 1992

[4] Hanrahan,P.: Ray-Triangle and Ray-Quadrirateral Intersections in Homogeneous Coordinates, http://graphics.stanford.edu/courses/cs348b-04/rayhomo.pdf, (unpublished) 1989

[5] Hartley,R, Zisserman,A.: MultiView Geometry in Computer Vision, Cambridge Univ. Press, 2000.

[6] Hildenbrand,D., Fontijne,D., Perwass,C., Dorst,L: Geometric Algebra and its Application to Computer Graphics, Eurographics 2004 Tutorial, pp.1-49, 2004.

[7] Hill,F.S.: Computer Graphics using OpenGL, Prentice Hall, pp.827, 2001

[8] Hradek,J., Skala,V.: Hash Function and Triangular Mesh Reconstruction, Vol.29, Computers&Geosciences, Pergamon Press, No.6., pp.741-751, 2003

[9] Chevalley,C.: Fundamental Concepts of Algebra, Academic Press, pp.201-203, 1956,

[10] Jimenez,J.J., Segura,R.J., Feito,F.R.: Efficient Collision Detection between 2D Polygons, Journal of WSCG, Vol.12, No.1-3, 2003

[11] Johnson,M.: Proof by Duality: or the Discovery of "New" Theorems, Mathematics Today, December 1996.

[12] Klein,F.: Notiz Betreffend dem Zusammenhang der Liniengeometrie mit der Mechanik starrer KÄorper.,Math. Ann. 4 , pp.403- 415, 1871

[13] Leclerc,A.P.: Efficient and Reliable Global Optimization, PhD Thesis, Ohio State University, 1992

[14] Lorentzen,L.: Continued Fractions, Atlanties Studies in Mathematics for Engineering and Science, World Scientific Publ., 2008

[15] Ma,Y., Soatto,S., Kosecka,J., Sastry,S.S.: An Invitation to 3D Vision, Springer Verlag, 2004

[16] Matousek,R., Tichy,M., Pohl,Z., Kadlec,J., Softley,C., Coleman,N.: Logarithmic Number System and Floating-point Arithmetic on FPGA, in Proc. FPL, 2002, pp. 627–636.

[17] Mohr,R., Triggs,B.: Projective Geometry for Image Analysis, Tutorial notes, http://www.inrialpes.fr/movi, 1996

[18] Oh,E., Walster,W.G.: Rump's Example Revisited, Reliable Computing, Kluwer Academic Publ., Vol.9., pp.245-248, 2002

[19] Press,W.H., Teukolsky,S.A., Vetterling,W.T., Flannery,B.P.: Numerical recipes in C, Cambridge University Press, 1999
[20] Rump,S.M.: Realiability in Computing, The role of Interval Methods in Scientific Computing, Academic Press, 1988
[21] Skala,V., Kaiser,J., Ondracka,V.: Library for Computation in the Projective Space, 6th Int.Conf. Aplimat, Bratislava, pp. 125-130, 2007
[22] Skala,V., Ondracka,V.: A Precision of Computation in the Projective Space, Recent Researches in Computer Science, pp.35-40, 15th WSEAS Int.Conference on Computers, Corfu, Greece, 2011
[23] Skala,V.: A New Line Clipping Algorithm with Hardware Acceleration, CGI'2004 conference proceedings, IEEE, Greece, 2004
[24] Skala,V.: Barycentric Coordinates Computation in Homogeneous Coordinates, Computers & Graphics, Elsevier, Vol. 32, No.1, pp.120-127, 2008
[25] Skala,V.: Computation in Projective Space, MAMECTIS conference, La Laguna, Spain, WSEAS, pp.152-157, 2009
[26] Skala,V.: Duality and Intersection Computation in Projective Space with GPU support, Applied Mathematics, Simulation and Modeling - ASM 2010 conference, NAUN, Corfu, Greece, pp.66-71, 2010
[27] Skala,V.: Geometric Computation, Duality and Projective Space, ICGG 2010 conference, pp.363-364, Kyoto, Japan, 2010
[28] Skala,V.: Geometric Computation, Duality and Projective Space, IW-LGK workshop proceedings, pp.105-111, Dresden University of Technology, 2011
[29] Skala,V.: Intersection Computation in Projective Space using Homogeneous Coordinates, Int.Journal on Image and Graphics, Vol.8, No.4, pp.615-628, 2008
[30] Skala,V.: Length, Area and Volume Computation in Homogeneous Coordinates, International Journal of Image and Graphics, Vol.6., No.4, pp.625-639, 2006.
[31] Skala,V.: Mathematical Foundations for Computer Graphics and Computer Vision, Tutorial CGI 2008 conference, Istanbul, Turkey, 2008.
[32] Skala,V.: Mathematical Foundations for Computer Graphics and Virtual Reality, Tutorial Intuition 2008 conference, Torino, Italy, 2008.
[33] Skala,V.: Mathematical Foundations for Computer Graphics and Vision and Computations in Projective Spaces, Tutorial 3DTV conference, 2007.
[34] Skala,V: A new Approach to Line and Line Segment Clipping in Homogenenous Coordinates, The Visual Computer, Vol.21, No.11, pp.905-914, 2005
[35] Skala,V: Duality and Intersection Computation in Projective Space with GPU Support, WSEAS Trans.on Mathematics, Vol.9., No.6., pp.407-416, 2010
[36] Skala,V.: Fast Information Retrieval for Textual and Geometrical Applications, 16th WSEAS Conf.on Computers, Kos, accepted for publication, WSEAS, Greece, 2012
[37] Skala,V., Hradek,J., Kuchar,M.: New Hash Fuction Construction for Textual and Geometrical Data Retrieval, Latest Trends on Computers, CSCC conference, Corfu, Vol.2, pp.483-489, Greece, 2010
[38] Stolfi,J.: Oriented Projective Geometry, Academic Press, 2001.
[39] Thomas,F., Torras,C.: A Projective invariant intersection test for polyhedra, The Visual Computer, Vol.18, No.1, pp.405-414, 2002
[40] Vince,J.: Geometry for Computer Graphics, Springer Verlag, 2004
[41] Yamaguchi,F., Niizeki,M.: Some basic geometric test conditions in terms of Plücker coordinates and Plücker coefficients, The Visual Computer, Vol.13, pp.29-41, 1997
[42] Yamaguchi,F.: Computer-Aided geometric Design – A Totally Four Dimensional Approach, Springer Verlag, 1996
[43] Zapletal,J., Vanecek,P., Skala,V.: RBF-based Image Restoration Utilising Auxiliary Points, CGI 2009 proceedings, pp.39-44, 2009

WEB resources

[44] Ackermann function, http://en.wikipedia.org/wiki/Ackermann_function, <retrieved 2012-02-23>
[45] Ariane 501 report http://www.esa.int/esaCP/Pr_33_1996_p_EN.html <retrieved 2012-02-02>
[46] Arnold,D.A.: The sinking of the Sleipner A offshore Platform http://www.ima.umn.edu/~arnold/disasters/sleipner.html <retrieved 2012-02-02>
[47] Arnold,D.A.: Two disasters caused by computer arithemtic error,

http://www.ima.umn.edu/~arnold/455.f96/disasters.html <retrieved 2012-02-02>

[48] Continuous Fractions http://www.numericana.com/answer/fractions.htm <retrieved 2012-01-29>

[49] IEEE-754 Data Format, http://en.wikipedia.org/wiki/IEEE_754-2008 <retrieved 2012-01-29>

[50] Patriot Missile Defense: Software Problem Led to System Failure at Hhara, Saudi Arabia, GAO/IMTEC-92-26 Report, House of Representatives , USA, 2009 http://www.gao.gov/assets/220/215614.pdf <retrieved 2012-01-29>

[51] SINTEF, http://www.sintef.no/ <retrieved 2012-02-23>

[52] Tucker,W.: Automatic Differentiation, http://www.sintef.no/project/eVITAmeeting/2010/vn2010.pdf <retrieved 2012-01-29>

[53] Materials Digital Library Pathway - MATDL, http://matdl.org/failurecases/Main_Page, <retrieved 2012-02-23>

Appendix A

The cross product in 4D is defined as

$$\mathbf{x}_1 \times \mathbf{x}_2 \times \mathbf{x}_3 = \det \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} & \mathbf{l} \\ x_1 & y_1 & z_1 & w_1 \\ x_2 & y_2 & z_2 & w_2 \\ x_3 & y_3 & z_3 & w_3 \end{vmatrix}$$

and can be implemented in Cg/HLSL on a GPU as follows:

```
float4 cross_4D(float4 x1, float4 x2, float4 x3)
{     float4 a;
a.x = dot(x1.yzw, cross(x2.yzw, x3.yzw));
a.y = - dot(x1.xzw, cross(x2.xzw, x3.xzw));
a.z = dot(x1.xyw, cross(x2.xyw, x3.xyw));
a.w = - dot(x1.xyz, cross(x2.xyz, x3.xyz));
return a;
}
```

or more compactly

```
float4 cross_4D(float4 x1, float4 x2, float4 x3)
{
return (
  dot(x1.yzw, cross(x2.yzw, x3.yzw)),
- dot(x1.xzw, cross(x2.xzw, x3.xzw)),
  dot(x1.xyw, cross(x2.xyw, x3.xyw)),
- dot(x1.xyz, cross(x2.xyz, x3.xyz)) );
}
```