FINAL version / DRAFT 1-14 , ref. No. 05-457340
Final version submitted to the International Journal of Image and Graphics (IJIG)
http://www.worldscinet.com/ijig/ijig.shtml

Length, Area and Volume Computation in Homogeneous Coordinates

Vaclav Skala
University of West Bohemia, Plzen
Department of Computer Science and Engineering
Czech Republic
skala@kiv.zcu.cz

Abstract

Many problems solved in computer graphics, computer vision, visualization etc. require fast and robust computation of an area of a triangle or volume of a tetrahedron. These very often used algorithms are well known and robust if vertices coordinates of triangles or tetrahedrons are given in Euclidean coordinates.

The homogeneous coordinates are often used for the representation of geometric transformations. They enable us to represent translation, rotation, scaling and projection operations in a unique way and handle them properly. Today's graphics hardware based on GPU offers very high computational power using pixel shaders and fragment shaders not only for graphical elements processing, but also for general computation using GPU as well.

This paper presents simple methods for the area of a triangle and the volume of a tetrahedron computation if vertices are given in homogeneous coordinates without the need to use the division operation for vertices coordinates transformation from the homogeneous coordinates to the Euclidian coordinates.

Area or volume computation is transferred to the cross product computation that is fast, simple, robust and can be supported in hardware or implemented on GPU that uses vector operations with homogeneous coordinates natively.

The presented formula can be used directly for Euclidean representation just setting w equal to 1.

Keywords: triangle area, tetrahedron volume, homogeneous coordinates, projective space, duality, cross product.

Notation

X = (X, Y) – a point in Euclidean coordinates
X – coordinate in the Euclidean coordinates
x = $[x,y,w]^T$ – point in the projective space represented by the homogeneous coordinates
x – coordinate in the homogeneous coordinates
p = $[a,b,c]^T$ – vector of a line p defined as ax+by+c=0
D(p) – dual representation of p
$\mathbf{a}^T\mathbf{b}$ – dot product of two vectors **a**, **b**
a × **b** – cross product of two vectors **a**, **b**
i = $[1,0,0]^T$, **j** = $[0,1,0]^T$, **k** = $[0,0,1]^T$
$a_i = (-b_i : b_4)$ $i = 1,\dots,3$ *Plücker notation for homogeneous coordinates e.g.* $a_1 = -\frac{b_1}{b_4}$

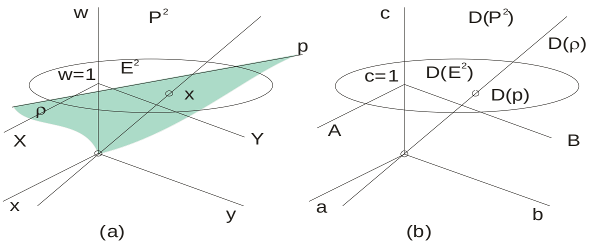
1. Introduction

Homogeneous coordinates are used in computer graphics, computer vision, visualization etc. in many ways. Many geometric methods used in computer graphics and related fields require the length of a transformed line, the area of a transformed triangle, or the volume of a transformed tetrahedron. The transformed objects are typically represented as homogeneous points. The quantities (length, area, and volume) are, however, typically computed in Euclidean space, requiring the transformed points to be converted from the homogenenous to the Euclidean form. This conversion requires division, which can lead to instability. The length, area, and voume calculations would be more stable and efficient if they utilized the homogeneous points directly.

This paper presents an improvement for the line segment length, the area of a triangle and the volume of a tetrahedron computation avoiding transformation to the Euclidean space. It is faster and more robust as it post-pones the division operation to the very last step if needed.

2. Projective geometry and duality

Homogeneous coordinates are widely used in computer graphics applications, usually associated with geometric transformations, such as rotation, scaling, translation and projection. In many cases, homogeneous coordinates are seen just as a "mathematical tool" that enables a simple description of geometric transformations. There are many "invisible" impacts to algorithm design that can lead to new, fast and robust algorithms that can be also supported in GPU hardware. Fig.1.a. presents a geometrical interpretation of the Euclidean and projective spaces.



Euclidean, projective and dual space representations
Figure 1

The point **x** is defined as a point in E^2 with coordinates **X**=(X,Y) or as a point with homogeneous coordinates $[x,y,w]^T$, where w = 1 usually. The point **x** is actually a "line" without the origin in the projective space P^2, and X = x/w and Y = y/w. It can be seen that a line $p \in E^2$ is actually a plane ρ without the origin in the projective space P^2, i.e. the Euclidean line p is defined as:

$$ax + by + cw = 0, \quad w \neq 0$$

Any $\xi \neq 0$ can multiply the equation without any effect to the geometry. In dual representation, see Fig.1.b, the plane ρ can be represented as a line $D(\rho) \in D(P^2)$ or as a point $D(p) \in D(E^2)$ when a projection is made, e.g. for c = 1. Complete theory on projective spaces can be found in [Sto01a], [Cox69a]

On the other hand, there is a principle of duality that is useful when deriving some formula. The principle states that any theorem remains true when we interchange the words "point" and line", "lie on" and "pass through", "join" and "intersection" and so on. Once the theorem has been established, the dual theorem is obtained as described above, see [Joh96a].

In other words, the principle of duality in E^2 says that in all theorems it is possible to substitute a term "point" by a term "line" and term "line" by the term "point" and the given theorem stays valid. This helps a lot in the solution of some geometrical cases.

Definition 1

The cross product of two vectors $\mathbf{x}_1 = [x_1,y_1,w_1]^T$ and $\mathbf{x}_2 = [x_2,y_2,w_2]^T$ is defined as

$$\mathbf{x}_1 \times \mathbf{x}_2 = \det\begin{bmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ x_1 & y_1 & w_1 \\ x_2 & y_2 & w_2 \end{bmatrix}$$

where: $\mathbf{i} = [1,0,0]^T$, $\mathbf{j} = [0,1,0]^T$, $\mathbf{k} = [0,0,1]^T$
Please, note that homogeneous coordinates are used.

Theorem 1
Let two points $\mathbf{x}_1$ and $\mathbf{x}_2$ be given in the projective space. Then the coefficients of the line p, which is defined by those two points, are the cross product of their homogeneous coordinates

$$\mathbf{p} = \mathbf{x}_1 \times \mathbf{x}_2$$

Proof 1
Let the line $p \in E^2$ be defined as

$$ax + by + cw = 0$$

Then

$$a = \det\begin{bmatrix} y_1 & w_1 \\ y_2 & w_2 \end{bmatrix} \qquad b = -\det\begin{bmatrix} x_1 & w_1 \\ x_2 & w_2 \end{bmatrix}$$

$$c = \det\begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \end{bmatrix}$$

Note: for w = 1 we get the standard cross-product formula and the cross product defines the line p, i.e.

$$\mathbf{p} = \mathbf{x}_1 \times \mathbf{x}_2$$

where: $\mathbf{p} = [a,b,c]^T$

Theorem 2
Let two lines p_1 and p_2 be given in the projective space. Then the homogeneous coordinates of the point **x** at the intersection of those two lines are given by the cross product of their coordinates

$$\mathbf{x} = \mathbf{p}_1 \times \mathbf{p}_2 = \det\begin{bmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \end{bmatrix}$$

where: $\mathbf{i} = [1,0,0]^T$, $\mathbf{j} = [0,1,0]^T$, $\mathbf{k} = [0,0,1]^T$

Proof 2
Immediate from Theorem 1 and the duality principle.

In E^3 case a plane ρ is given by three points **X**=(X,Y,Z) or by points in homogeneous coordinates $[x,y,z,w]^T$.

Theorem 3
Let three points $\mathbf{x}_1$, $\mathbf{x}_2$ and $\mathbf{x}_3$ be given in the projective space. Then the coefficients of the plane ρ, which is defined by those three points, are the cross product of their homogeneous coordinates

$$\boldsymbol{\rho} = \mathbf{x}_1 \times \mathbf{x}_2 \times \mathbf{x}_3$$

where: $\boldsymbol{\rho} = [a,b,c,d]^T$
and cross product is defined as follows:

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

where: $\mathbf{i} = [1,0,0,0]^T$, $\mathbf{j} = [0,1,0,0]^T$, $\mathbf{k} = [0,0,1,0]^T$, $\mathbf{l} = [0,0,0,1]^T$

The proof is left to the reader similarly to Proof 1.

Theorem 4

Let three planes ρ_1, ρ_2 and ρ_3 be given in the projective space. Then the homogeneous coordinates of the point **x** at the intersection of those three planes are given by the cross product of their coordinates

$$\mathbf{x} = \boldsymbol{\rho}_1 \times \boldsymbol{\rho}_2 \times \boldsymbol{\rho}_3$$

i.e.:

$$\boldsymbol{\rho}_1 \times \boldsymbol{\rho}_2 \times \boldsymbol{\rho}_3 = \begin{bmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} & \mathbf{l} \\ a_1 & b_1 & c_1 & d_1 \\ a_2 & b_2 & c_2 & d_2 \\ a_3 & b_3 & c_3 & d_3 \end{bmatrix}$$

where: $\mathbf{i} = [1,0,0,0]^T$, $\mathbf{j} = [0,1,0,0]^T$, $\mathbf{k} = [0,0,1,0]^T$, $\mathbf{l} = [0,0,0,1]^T$

The proof is left to the reader, as it is similar to Proof 2.

These theorems are very important as they enable us to handle some problems defined in the homogeneous coordinates efficiently and make computations quite robust and effective.

3. Euclidian coordinates

Let us consider a very simple case - a line segment in E^2 and a line p on which the line segment lies, see Fig.2.

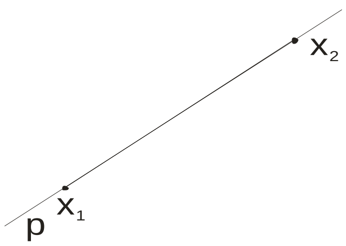


Figure 2

It is well known that the length l of the line segment is given as

$$l = \sqrt{(X_2 - X_1)^2 + (Y_2 - Y_1)^2} = \|\mathbf{X}_2 - \mathbf{X}_1\|$$

where $\mathbf{X}_1 = (X_1, Y_1)$, $\mathbf{X}_2 = (X_2, Y_2)$ are the end-points of the given line segment.

Nevertheless, what happen if the end-points of the line segment are expressed as homogeneous coordinates? Usually the end-points are converted to the Euclidean coordinates and the length l of the line segment is computed according to the equation shown.

Let us consider an area of a triangle in E^2 and E^3, which is a more general case, see Fig.3.

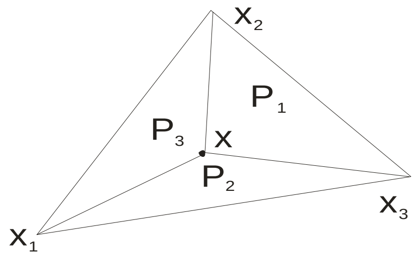


Figure 3

The formula

$$S = \frac{1}{2}\left\| (\mathbf{X}_2 - \mathbf{X}_1) \times (\mathbf{X}_3 - \mathbf{X}_1) \right\|$$

for the area of the triangle computation is known if the vertices are given in the Euclidean coordinates. Again difficulties arise if the vertices are given in the homogeneous coordinates.

Let us consider a case with a tetrahedron, see Fig.4.

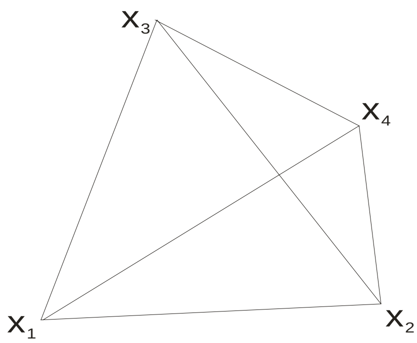


Figure 4

Again the well known formula

$$V = \frac{1}{6}\left\| (\mathbf{X}_2 - \mathbf{X}_1) \times (\mathbf{X}_3 - \mathbf{X}_1) \times (\mathbf{X}_4 - \mathbf{X}_1) \right\|$$

can be used, but it needs the division operation use if the vertices are given in the homogeneous coordinates.

Before we start deriving new formulas that are common for all the three cases, let us introduce some algebraic fundamentals, namely some specific operation with determinants.

4. Basic operations with determinants

Determinants are very often used in geometric algorithms and basic operations are well known. Let us take the opportunity and present the not-frequent operations.

Because determinants are multilinear in rows and columns we can write:

$$\det\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn}\end{bmatrix} \pm \det\begin{bmatrix} a_{11} & b_{12} & \cdots & a_{1n} \\ a_{21} & b_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & b_{n2} & \cdots & a_{nn}\end{bmatrix} =$$

$$\det\begin{bmatrix} a_{11} & a_{12}\pm b_{12} & \cdots & a_{1n} \\ a_{21} & a_{22}\pm b_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2}\pm b_{n2} & \cdots & a_{nn}\end{bmatrix}$$

we will use this property to derive some formula in order to understand the final solution.

It is well known that the determinant equals zero if two rows or two columns are linearly dependent

$$\det\begin{bmatrix} a_{11} & k.a_{12} & \cdots & a_{1n} \\ a_{21} & k.a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & k.a_{n2} & \cdots & a_{nn}\end{bmatrix} = 0$$

The above mentioned mathematical operations will be needed in the next sections.

5. Length of a line segment

Let us consider the line segment in E^2 from Fig.2. It is necessary to note that the end-points are given in the homogeneous coordinates, now.

We know that the line p is determined as

$$\mathbf{p} = \mathbf{x}_1 \times \mathbf{x}_2 \quad \text{and} \quad \mathbf{p} = [a,b,c]^T$$

i.e. the line p is defined by an equation

$$ax + by + cw = 0$$

or

$$\mathbf{p}^T \mathbf{x} = 0$$

where $\mathbf{x} = [x, y, w]^T$ and $\mathbf{p} = [a,b,c]^T$

This equation can also be expressed as

$$\det\begin{bmatrix} x & y & w \\ x_1 & y_1 & w_1 \\ x_2 & y_2 & w_2\end{bmatrix} = 0$$

The normal vector **n** of the line p is determined as

$$\mathbf{n} = [a,b]^T$$

where:

$$a = \det\begin{bmatrix} y_1 & w_1 \\ y_2 & w_2\end{bmatrix} = y_1 w_2 - y_2 w_1$$

$$b = -\det\begin{bmatrix} x_1 & w_1 \\ x_2 & w_2\end{bmatrix} = x_2 w_1 - x_1 w_2$$

We want to compute the length l of the given line segment.

Let us evaluate $\mathbf{n}^T\mathbf{n}$ expression, now. It can be seen that we get:

$$\mathbf{n}^T\mathbf{n} = (y_1 w_2 - y_2 w_1)^2 + (x_2 w_1 - x_1 w_2)^2 =$$

$$(Y_1 w_1 w_2 - Y_2 w_2 w_1)^2 + (X_2 w_2 w_1 - X_1 w_1 w_2)^2 =$$

$$(w_1 w_2)^2 \left[(Y_2 - Y_1)^2 + (X_2 - X_1)^2 \right]$$

Then the length l of the line segment is determined in the Euclidean coordinates as

$$l^2 = (X_2 - X_1)^2 + (Y_2 - Y_1)^2 = \frac{\mathbf{n}^T\mathbf{n}}{(w_1 w_2)^2}$$

It means that the length l of the line segment can be computed by the cross and dot products. If we use Plücker notation, we get a formula for the length l of the line segment as:

$$l = \left| \left(\sqrt{\mathbf{n}^T\mathbf{n}} : w_1 w_2 \right) \right|$$

It is a trivial result, but please note that if the points that define the line segment are given in the homogeneous coordinates, we do not need any division operation if the length l can be stored in the homogeneous coordinates. If the Euclidean representation is required, we need one division operation in total and we save 3 divisions in total.

It is a very promising result. Let us explore if this approach could be applied to the area of a triangle computation as well.

6. Area of a triangle

It is well known that an area of a triangle, see Fig.3, can be determined as:

$$S = \frac{1}{2} \left\| (\mathbf{X}_2 - \mathbf{X}_1) \times (\mathbf{X}_3 - \mathbf{X}_1) \right\| =$$

$$\frac{1}{2} \left\| \det\begin{bmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ X_2 - X_1 & Y_2 - Y_1 & Z_2 - Z_1 \\ X_3 - X_1 & Y_3 - Y_1 & Z_3 - Z_1\end{bmatrix} \right\|$$

The plane ρ on which the triangle lies can be determined as

$$\boldsymbol{\rho} = \mathbf{x}_1 \times \mathbf{x}_2 \times \mathbf{x}_3$$

where: $\boldsymbol{\rho} = [a,b,c,d]^T$ and the plane ρ is determined by an equation

$$\boldsymbol{\rho}^T \mathbf{x} = 0$$

i.e.

$$\det\begin{bmatrix} x & y & z & w \\ x_1 & y_1 & z_1 & w_1 \\ x_2 & y_2 & z_2 & w_2 \\ x_3 & y_3 & z_3 & w_3\end{bmatrix} = 0$$

where: $\mathbf{x} = [x, y, z, w]^T$

The normal vector **n** of the plane ρ is determined as $\mathbf{n} = [a,b,c]^T$.

The area S of the triangle can be computed as

$$S = \frac{1}{2}\left| \frac{\sqrt{\mathbf{n}^T\mathbf{n}}}{w_1 w_2 w_3} \right|$$

The area S of the given triangle can be computed using cross and dot products only. If we use Plücker notation, we get a formula for the length of the line segment as:

$$S = \left| \left(\sqrt{\mathbf{n}^T\mathbf{n}} : 2\ w_1 w_2 w_3 \right) \right|$$

It is an interesting result as we do not need division operation to determine the area of the given triangle at all if the result can be in the homogeneous coordinates and saving 9 division operation as we do not need to transfer the vertices coordinates to the Euclidean coordinates. If we need the standard scalar value, we need one division only at the very last step, so saving 8 division operations in total.

Let us consider a volume computation of the given tetrahedron.

7. Volume of a tetrahedron

Volume of a tetrahedron, see Fig.4, can be computed as

$$V = \frac{1}{6}\left\| (\mathbf{X}_2 - \mathbf{X}_1)\times(\mathbf{X}_3 - \mathbf{X}_1)\times(\mathbf{X}_4 - \mathbf{X}_1) \right\|$$

if the vertices $\mathbf{X}_i$ are given in the Euclidean coordinates of tetrahedron vertices. If the vertices coordinates are given in the homogeneous coordinates, the volume can be computed as (see Appendix for details)

$$V = \frac{1}{6}\left| \frac{1}{w_1 w_2 w_3 w_4} \det \begin{bmatrix} x_1 & y_1 & z_1 & w_1 \\ x_2 & y_2 & z_2 & w_2 \\ x_3 & y_3 & z_3 & w_3 \\ x_4 & y_4 & z_4 & w_4 \end{bmatrix} \right| =$$

$$\frac{1}{6}\left| \det \begin{bmatrix} X_1 & Y_1 & Z_1 & 1 \\ X_2 & Y_2 & Z_2 & 1 \\ X_3 & Y_3 & Z_3 & 1 \\ X_4 & Y_4 & Z_4 & 1 \end{bmatrix} \right| =$$

$$\frac{1}{6}|D|$$

where: $D = \det \begin{bmatrix} X_1 & Y_1 & Z_1 & 1 \\ X_2 & Y_2 & Z_2 & 1 \\ X_3 & Y_3 & Z_3 & 1 \\ X_4 & Y_4 & Z_4 & 1 \end{bmatrix}$

The given tetrahedron can be considered as a halfspace in the 5-dimensional projective space, and the vertices can be considered as

$$\mathbf{x}_i = [x_i, y_i, z_i, \xi_i, w_i]^T \qquad i = 1,...,4$$

Then the tetrahedron, actually half space in the 5-dimensional projective space, is determined as

$$\mathbf{q}^T\mathbf{x} = 0$$

where $\mathbf{x} = [x, y, z, v, w]^T$ and $\mathbf{q} = [a, b, c, d, e]^T$ i.e. using the determinant form we get

$$\det \begin{bmatrix} x & y & z & v & w \\ x_1 & y_1 & z_1 & 0 & w_1 \\ x_2 & y_2 & z_2 & 0 & w_2 \\ x_3 & y_3 & z_3 & 0 & w_3 \\ x_4 & y_4 & z_4 & 0 & w_4 \end{bmatrix} = 0$$

it can be seen that $[a, b, c, d, e]^T = [0, 0, 0, d, 0]^T$, where:

$$d = \det \begin{bmatrix} x_1 & y_1 & z_1 & w_1 \\ x_2 & y_2 & z_2 & w_2 \\ x_3 & y_3 & z_3 & w_3 \\ x_4 & y_4 & z_4 & w_4 \end{bmatrix}$$

It means that the normal $\mathbf{n}$ of the halfspace in the 5-dimensional projective space is given as

$$\mathbf{n} = [0, 0, 0, d, 0]^T$$

and the volume of the given tetrahedron can be determined as

$$V = \left| \left(\sqrt{\mathbf{n}^T\mathbf{n}} : 3 w_1 w_2 w_3 w_4 \right) \right|$$

It means that if the volume of the given tetrahedra can be represented in the projective coordinates, we save 12 division operations and if the scalar value is required, one division is needed and we save 11 division operations.

We have the following formula for direct computation in the homogeneous coordinates

$$Q_k = \left| \left(\sqrt{\mathbf{n}^T\mathbf{n}} : (k-1)! \prod_{i=1}^{k} w_i \right) \right|$$

where: k is a number of the end-points or vertices of the given object, Q_k is the line segment length or the area of the triangle or the volume of the tetrahedron.

8. New formula

The above derived equations for computations with the homogeneous coordinates enable us to derive new formula that could be useful is some algorithms.

It can be seen that the normalized normal vector $\bar{\mathbf{n}}$ for the given line segment can be computed as :

$$\bar{\mathbf{n}} = \frac{\mathbf{n}}{\sqrt{\mathbf{n}^T\mathbf{n}}} = (\mathbf{n} : l\ w_1 w_2)$$

where: l is the length of the line segment.

For a triangle the normal $\bar{\mathbf{n}}$ vector as

$$\bar{\mathbf{n}} = \frac{\mathbf{n}}{\sqrt{\mathbf{n}^T\mathbf{n}}} = (\mathbf{n} : 2S\ w_1 w_2 w_3)$$

where: S is the area of the triangle.

For a tetrahedron we get "hyper-normal" $\bar{\mathbf{n}}$ vector as

$$\bar{\mathbf{n}} = \frac{\mathbf{n}}{\sqrt{\mathbf{n}^T \mathbf{n}}} = \left(\mathbf{n} : 6\,V\ w_1 w_2 w_3 w_4\right)$$

where: V is the volume of the tetrahedra.

In general, we get

$$\bar{\mathbf{n}} = \frac{\mathbf{n}}{\sqrt{\mathbf{n}^T \mathbf{n}}} = \left(\mathbf{n} : (k-1)!\,Q_k \prod_{i=1}^{k} w_i\right)$$

It means that if we know the length of a line segment or the area of a triangle or the volume of a tetrahedron, we can save $\sqrt{\ }$ operation as well.

It is necessary to note that the above presented formulas are theoretical and should be optimized for specific conditions in order to get higher computational performance.

According to the author's knowledge, the above-mentioned relations have not been known yet.

9. Conclusion

This paper describes a robust computation of the area of a triangle and the volume of a tetrahedron if vertices are given directly in the homogeneous coordinates. The main advantages of the proposed approach are:

- The line segment length, the area of a triangle and the volume of a tetrahedron can be computed without the use of division operation; division operation use is postponed to the final evaluation step if needed.
- If vertices of the line segment, the triangle or the tetrahedron are given in the homogenenous coordinates and computation is done directly using the homogeneous coordinates no division operation is needed.
- Computation is more robust especially for slim triangles or tetrahedrons as we do not use the division operation that causes instability and decreases the robustness of computation in general.

This approach can help to resolve some problems with robustness and instability of some algorithms in specific cases. It can be also applied to an effective use of GPU within some algorithms, eg. linear system of equations can be solved using the cross and dot products only if a solution can be represented in homogeneous coordinates [Ska06a].

The presented approach, the principle of duality and the homogeneous coordinates use could bring new directions in algorithms design that can result into simple, robust and fast algorithm e.g. line clipping algorithm in E^2 [Ska94a].

10. Acknowledgments

The author would like to express his thanks to students and colleagues at the University of West Bohemia for recommendations, constructive discussions and hints that helped to finish the work. Many thanks belong to the anonymous reviewers for their valuable comments and suggestions that improved this paper significantly, to Milan Vasa, Martin Janda and Jan Patera for their critical comments and recommendations and Ivo Hanak for his hints in hardware aspects features evaluation.

This work was supported by the project 6FP NoE 3DTV and by the project LC-CPG No.06008 of the Ministry of Education of the Czech Republic.

11. References

[Bloo94a] Bloomenthal,J., Rokne,J. (1994) Homogeneous Coordinates. The Visual Computer, 11, pp. 15-26, Springer Verlag

[Cox69a] Coxeter H S M (1969) Introduction to Geometry, John Wiley.

[Hart00a] Hartley R, Zisserman A (2000) MultiView Geometry in Computer Vision, Cambridge Univ. Press.

[Fei01a] Segura,R.J., Feito,F.R.: Algorithms to Test Ray-Triangle Intersection, Comparative study, WSCG 2001 conf.proceedings, ISBN 80-7082-711-2, pp.76-81, 2005.

[Fer03a] Fernando,R., Kilgard,M.J.: Cg Tutorial: The Definitive Guide to Programmable Real-Time Graphics, Addison Wesley, 2003.

[Jim03a] Jimenez,J.J., Segura,R.J., Feito,F.R.: Efficient Collision Detection between 2D Polygons, Journal of WSCG, Vol.12, No.1-3, ISSN 1213-6972, 2003.

[John96a] Johnson M (1996) Proof by Duality: or the Discovery of "New" Theorems, Mathematics Today, December.

[Moll97a] Moller,T., Trumbore,B.: Fast, Minimum Storage Ray/Triangle Intersection, Journal of Graphics Tools, No.1., Vol.2, pp.-21-28,1997.

[Stol01a] Stolfi J (2001) Oriented Projective Geometry, Academic Press.

[Ska05b] Skala,V.: GPU Computation in Projective Space using Homogeneous Coordinates, accepted for publication in Game Programming, Gems 6, River Media, Vol.6, 2006

[Ska04a] Skala,V.: A New Line Clipping Algorithm with Hardware Acceleration, CGI'2004 conference proceedings, IEEE, Greece, 2004

[Ska05a] Skala,V.: New Approach to Line and Line Segment Clipping in Homogeneous Coordinates, The Visual Computer, Vol.21, No.11,pp.905-914, ISSN 0178-2789, 2005

[Ska06a] Skala,V.: GPU Computation in Projective Space Using Homogeneous Coordinates , Game Programming GEMS 6 (Ed.Dickheiser,M.), pp.137-147, ISBN 1-58450-450-1, Charles River Media, 2006

[Tho2002a] Thomas,F., Torras,C.: A Projective invariant intersection test for polyhedra, The Visual Computer, 2002

[Yama90a] Yamaguchi,F., Niizeki,M., Fukunaga,H.: Two robust point-in-polygon test based on 4x4 determinant metho, Advanced on design automation (ASME), Vol. 23, No.1,pp.89-95, 1990

[Yama02a] Yamaguchi,F.: Computer-Aided Geometric Design: A Totally Four-Dimensional Approach, Springer Verlag, ISBN 4-431-70340-3, 2002

Appendix

To be able to fully understand some steps, a detailed derivation of a formula for tetrahedron volume is included.

We know that

$$D_x = \det \begin{bmatrix} X_1 & Y_1 & Z_1 & 1 \\ X_1 & Y_2 & Z_2 & 1 \\ X_1 & Y_3 & Z_3 & 1 \\ X_1 & Y_4 & Z_4 & 1 \end{bmatrix} = 0$$

as the first column is linearly dependent on the last column. Now we can write

$$D = D - D_x = \det \begin{bmatrix} 0 & Y_1 & Z_1 & 1 \\ X_2 - X_1 & Y_2 & Z_2 & 1 \\ X_3 - X_1 & Y_3 & Z_3 & 1 \\ X_4 - X_1 & Y_4 & Z_4 & 1 \end{bmatrix}$$

We can again set

$$D_y = \det \begin{bmatrix} 0 & Y_1 & Z_1 & 1 \\ X_2 - X_1 & Y_2 & Z_2 & 1 \\ X_3 - X_1 & Y_3 & Z_3 & 1 \\ X_4 - X_1 & Y_4 & Z_4 & 1 \end{bmatrix} = 0$$

as the second column is linearly dependent on the last column. Then we obtain

$$D = D - D_y = \det \begin{bmatrix} 0 & 0 & Z_1 & 1 \\ X_2 - X_1 & Y_2 - Y_1 & Z_2 & 1 \\ X_3 - X_1 & Y_3 - Y_1 & Z_3 & 1 \\ X_4 - X_1 & Y_4 - Y_1 & Z_4 & 1 \end{bmatrix}$$

We can again set

$$D_z = \det \begin{bmatrix} 0 & 0 & Z_1 & 1 \\ X_2 - X_1 & Y_2 - Y_1 & Z_2 & 1 \\ X_3 - X_1 & Y_3 - Y_1 & Z_3 & 1 \\ X_4 - X_1 & Y_4 - Y_1 & Z_4 & 1 \end{bmatrix} = 0$$

as the third column is linearly dependent on the last column. Then we get

$$D = D - D_z = \det \begin{bmatrix} 0 & 0 & 0 & 1 \\ X_2 - X_1 & Y_2 - Y_1 & Z_2 - Z_1 & 1 \\ X_3 - X_1 & Y_3 - Y_1 & Z_3 - Z_1 & 1 \\ X_4 - X_1 & Y_4 - Y_1 & Z_4 - Z_1 & 1 \end{bmatrix}$$

It can be seen that we obtained the original formula for the Euclidean coordinates.

Appendix A

The cross product in 4D defined as

$$\mathbf{x}_1 \times \mathbf{x}_2 \times \mathbf{x}_3 = \det \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} & \mathbf{l} \\ x_1 & y_1 & z_1 & w_1 \\ x_2 & y_2 & z_2 & w_2 \\ x_3 & y_3 & z_3 & w_3 \end{vmatrix} \qquad \text{(A1)}$$

can be implemented in Cg/HLSL on 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));
      // or a.y=dot(x1.xzw, cross(x3.xzw, x2.xzw));
```

```
        a.z=dot(x1.xyw, cross(x2.xyw, x3.xyw));
        a.w=-dot(x1.xyz, cross(x2.xyz, x3.xyz));
        // or a.w=dot(x1.xyz, cross(x3.xyz, x2.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)) );
}
```

The code is simple and uses vector operations available on current GPU. The cross-product in E^3 is supported directly by Cg/HLSL.

Photo and Bibliography

Vaclav Skala is a full professor of Computer Science at the Faculty of Applied Sciences at the University of West Bohemia in Plzen, Czech Republic. He is responsible for courses on Computer Graphics, Algorithms for Computer Graphics, Visualization, Multimedia Systems, Programming in Windows, .NET Technologies at the Department of Computer Science. He is a member of The Visual Computer and Computers&Graphics editorial boards, Eurographics Executive Committee and member of program committees of established international conferences. He has been a research fellow or lecturing at the Brunel University (London, U.K.), Moscow Technical University (Russia), Gavle University (Sweden) and others institutions in Europe. He organizes the WSCG International Conferences in Central Europe on Computer Graphics, Visualization and Computer Vision (http://wscg.zcu.cz) held annually since 1992 and .NET Technologies conferences (http://dotnet.zcu.cz). He is interested in algorithms, data structures, mathematics, computer graphics, computer vision and visualization. He has been responsible for several research projects as well. Currently, he is a director of the Center of Computer Graphics and Visualization (http://herakles.zcu.cz).