Analytical calculation of the solid angle subtended by an arbitrarily positioned ellipsoid

Eric Heitz

Analytical calculation of the solid angle subtended by an arbitrarily positioned ellipsoid

Eric Heitz

Unity Technologies

Abstract

We present a geometric method for computing an ellipse that subtends the same solid-angle domain as an arbitrarily positioned ellipsoid. With this method we can extend existing analytical solid-angle calculations of ellipses to ellipsoids. Our idea consists of applying a linear transformation on the ellipsoid such that it is transformed into a sphere from which a disk that covers the same solid-angle domain can be computed. We demonstrate that by applying the inverse linear transformation on this disk we obtain an ellipse that subtends the same solid-angle domain as the ellipsoid. We provide a MATLAB implementation of our algorithm and we validate it numerically.

Note: This is the author's version of an article accepted for publication in *Nuclear Inst. and Methods in Physics Research, A*. Changes were made to this version by the publisher prior to publication. The publisher's version is available at: http://www.sciencedirect.com/science/article/pii/S0168900217301857

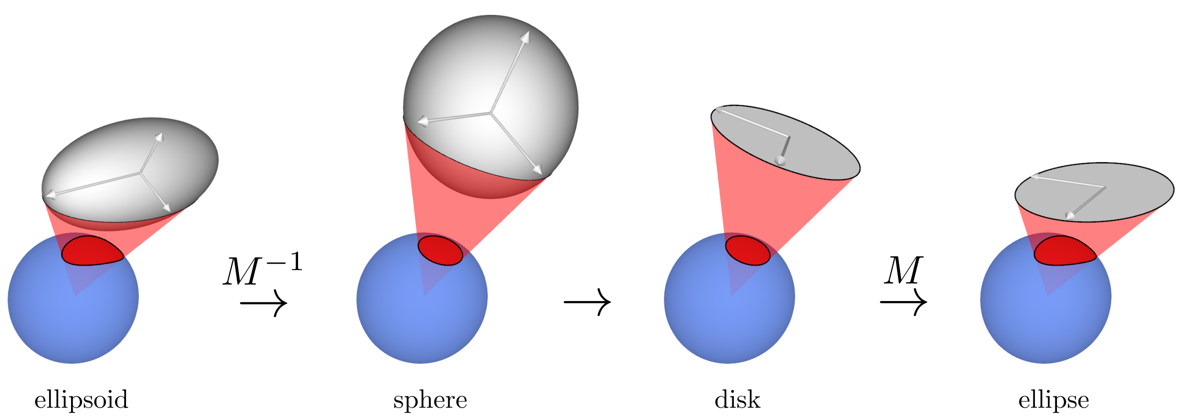

Figure 1: Illustration of our method.

1 Introduction

While analytic calculations of the solid angle subtended by ellipses exist [Con10], we are not aware of solutions for the solid angle subtended by ellipsoids in the general case where they can be arbitrarily positioned and/or oriented. In this paper, we present a simple geometric invariance property (Section 2) that allows for extracting an ellipse that subtends the same solid-angle domain as an ellipsoid in the general case (Section 3) and we obtain the solid angle of the ellipsoid by computing the solid angle of the ellipse with an analytical method. We validate our method against numerical solid angle computations (Section 4).

2 Linear invariance of solid-angle domains

In this section, we demonstrate the property our method is built on.

Property The property is illustrated in Figure 2. Let A and B be two objects that subtend the same solid-angle domain $\Omega_A = \Omega_B$ with respect to the origin $O = (0,0,0)$ and M a 3×3 invertible matrix. Then MA and MB, the objects obtained by applying the linear transformation M on respectively A and B, subtend the same solid-angle domain $\Omega_{MA} = \Omega_{MB}$ with respect to the origin O.

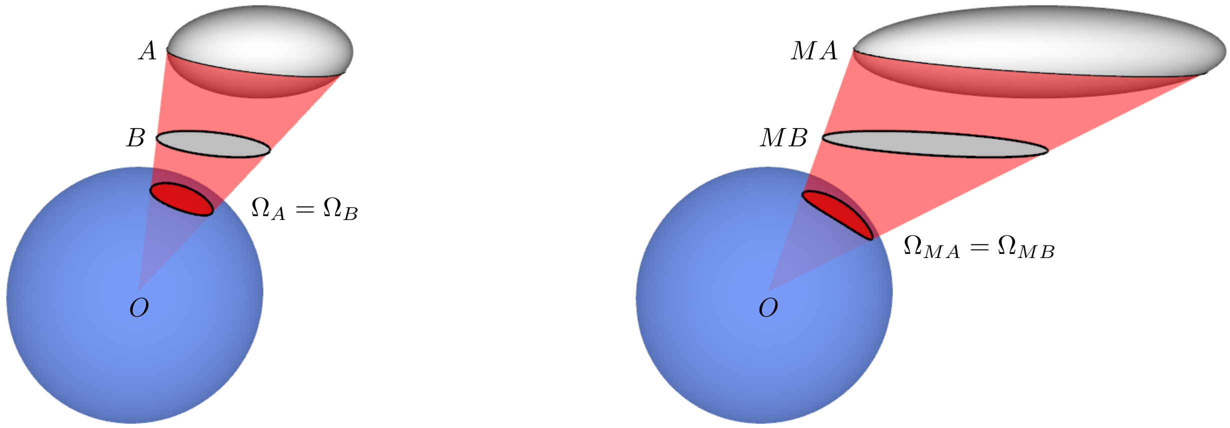

Figure 2: Property: if two objects A and B subtend the same solid-angle domain, this equality holds after applying a linear transformation M to them.

Proof The proof is illustrated in Figure 3. The solid-angle domain Ω_A subtended by an object A with respect to the origin is given by the set of half lines that start from the origin and intersect object A. Since a linear transformation applied on both a line and an object does not change their intersections, the set of lines intersecting MA is the set of lines intersecting A multiplied by matrix M, and the set of lines intersecting MB is the set of lines intersecting B multiplied by matrix M. Since A and B subtend the same solid-angle domain, i.e. $\Omega_A = \Omega_B$, the set of half lines intersecting A is the same as the set of half lines intersecting B, which means that the set of half lines intersecting MA is the same as the set of half lines intersecting MB, i.e. $\Omega_{MA} = \Omega_{MB}$.

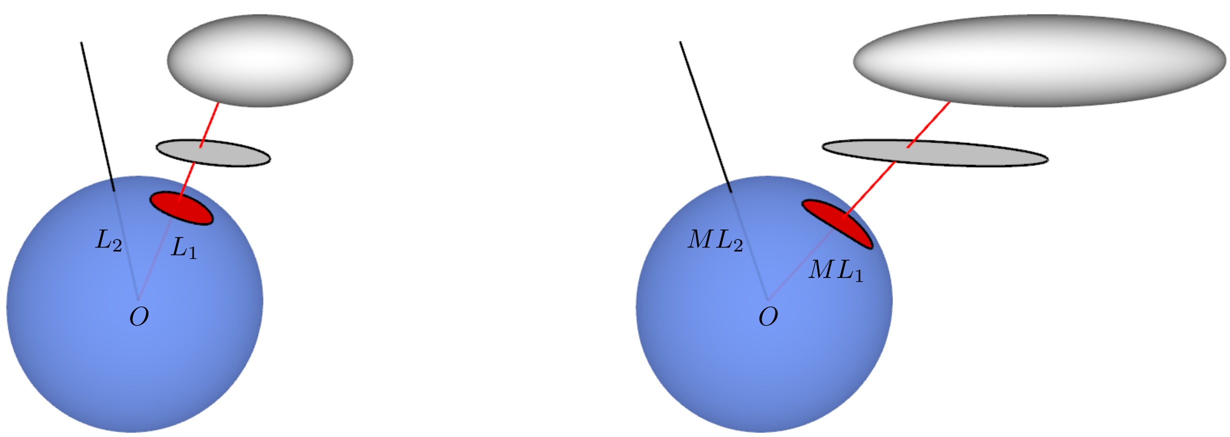

Figure 3: Proof: if two objects intersect the same half lines, then they intersect the same half lines after a linear transformation. Hence, if two objects subtend the same solid-angle domain, they subtend the same solid-angle domain after a linear transformation.

3 An ellipse that subtends the same solid angle as an ellipsoid

In this section, we explain our algorithm for computing an ellipse that subtends the same solid angle as an ellipsoid.

3.1 Linearly transforming the ellipsoid into a sphere

We consider an ellipsoid of center P_a and of principal axes given by an orthonormal basis (A_1, A_2, A_3) and lengths a_1, a_2, and a_3. Let M be the matrix

$$M = (A_1 \; A_2 \; A_3) \begin{bmatrix} a_1 & 0 & 0 \\ 0 & a_2 & 0 \\ 0 & 0 & a_3 \end{bmatrix} (A_1 \; A_2 \; A_3)^T , \tag{1}$$

then by multiplying the ellipsoid by the inverse matrix M^{-1} we obtain a sphere of radius 1 and of center

$$P_b = M^{-1} \, P_a. \tag{2}$$

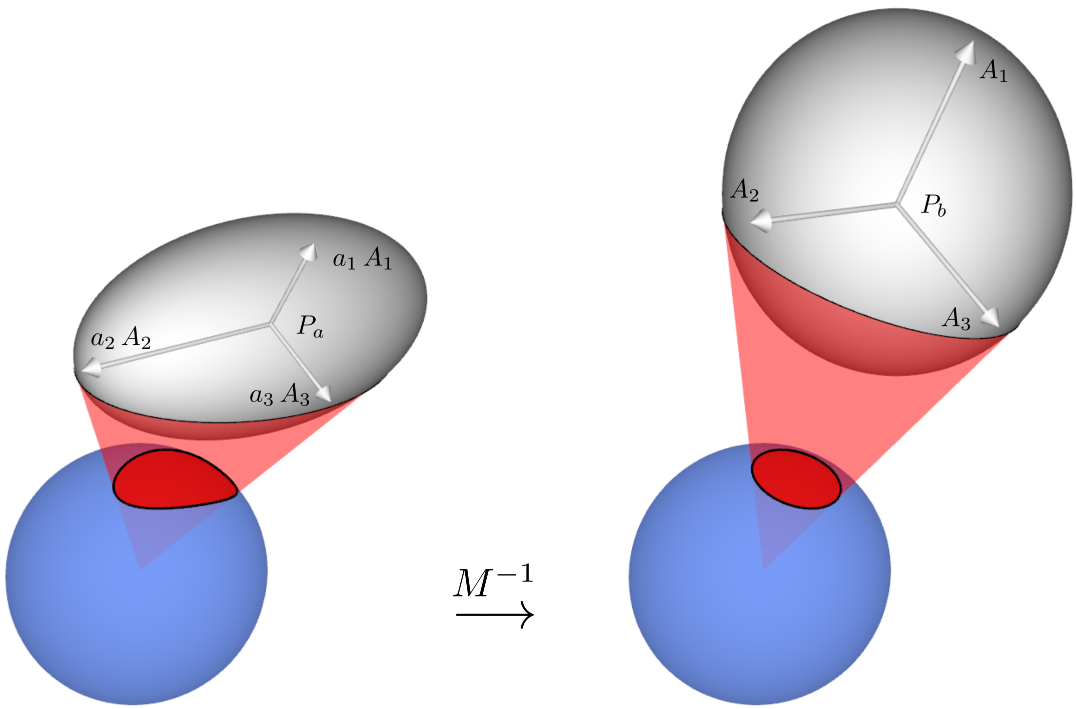

Figure 4: Linearly transforming the ellipsoid into a sphere of radius 1.

3.2 Extracting a disk that subtends the same solid-angle domain as the sphere

We extract a disk that subtends the same solid-angle domain as the sphere. Since the scale of this disk can be arbitrarily chosen, we choose the disk that lies on the surface of the sphere. The sphere subtends a spherical cap of angle

$$\theta = \operatorname{asin}\left(\frac{1}{\|P_b\|}\right). \tag{3}$$

The center of the disk is

$$P_c = \cos(\theta)^2 \ P_b \tag{4}$$

and its radius

$$c_1 = c_2 = \tan(\theta) \ \|P_c\| \tag{5}$$

The normal of the disk is $\frac{P_c}{\|P_c\|}$ and we use it to compute two tangent directions C_1 and C_2, for instance with Frisvad's method [Fri12], such that $\left(\frac{P_c}{\|P_c\|}, C_1, C_2\right)$ is an orthonormal basis.

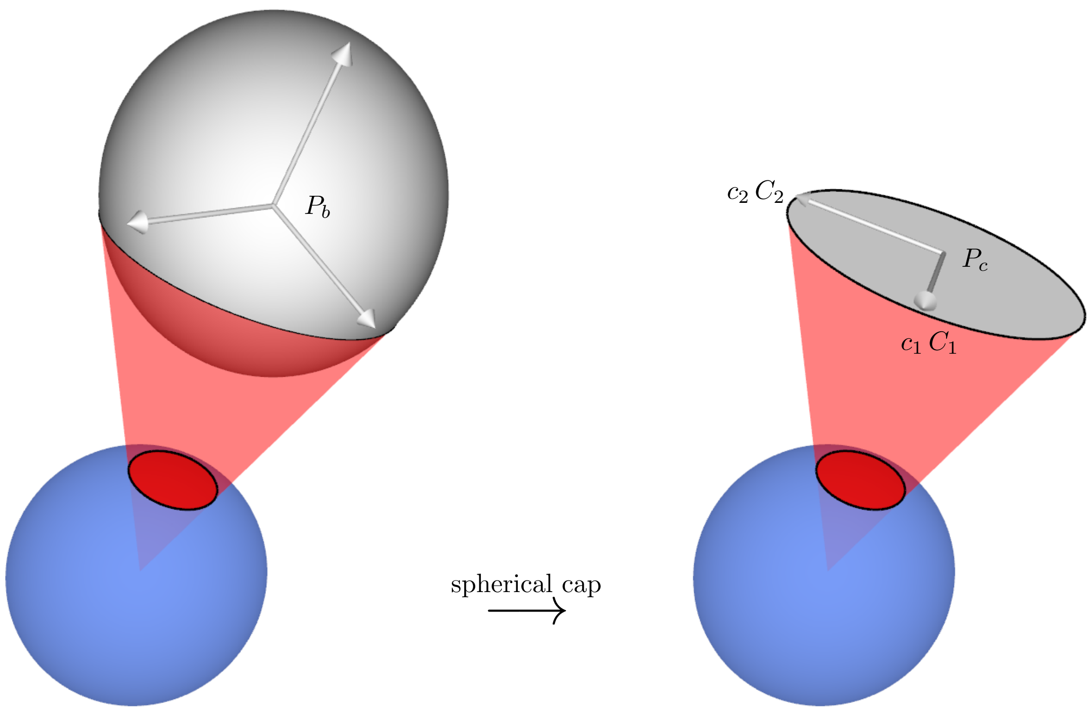

Figure 5: Extracting a disk that subtends the same solid-angle domain as the sphere.

3.3 Computing the ellipse obtained by applying the inverse linear transformation on the disk

Thanks to the property provided in Section 2, we know that since the disk and the sphere subtend the same solid-angle domain, any linear transform of them subtends the same solid-angle domain. If we multiply the sphere by matrix M we obtain the ellipsoid. Hence, the disk multiplied by matrix M is an ellipse that subtends the same solid angle as the ellipsoid. This ellipse is defined by its center

$$P_d = M\, P_c \tag{6}$$

and the linearly-transformed tangent vectors of the disk scaled by its radius

$$D_1' = M\, c_1\, C_1 \tag{7}$$
$$D_2' = M\, c_2\, C_2 \tag{8}$$

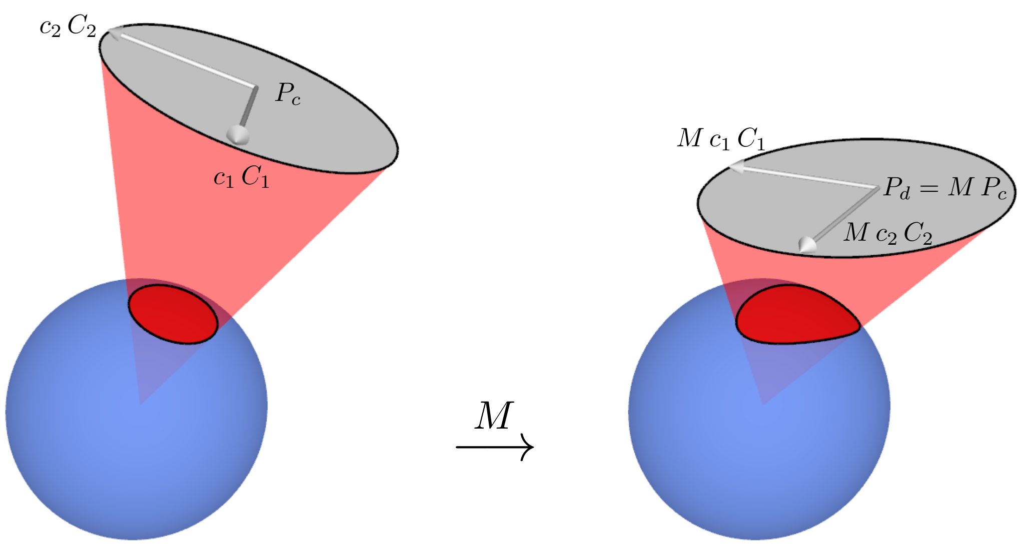

Figure 6: Computing the ellipse obtained by applying the inverse linear transformation on the disk.

3.4 Computing the principal axes of the ellipse

We obtain the principal axes of the ellipse by computing the 2D eigenvectors (p_1, p_2) and (q_1, q_2) and respective eigenvalues v_p and v_q of the dot-product matrix

$$Q = \begin{bmatrix} D'_1 \cdot D'_1 & D'_1 \cdot D'_2 \\ D'_1 \cdot D'_2 & D'_2 \cdot D'_2 \end{bmatrix} = \begin{bmatrix} p_1 & q_1 \\ p_2 & q_2 \end{bmatrix} \begin{bmatrix} v_p & 0 \\ 0 & v_q \end{bmatrix} \begin{bmatrix} p_1 & q_1 \\ p_2 & q_2 \end{bmatrix}^T . \tag{9}$$

The directions of the principal axes are the eigenvectors

$$D_1 = \frac{p_1\, D'_1 + p_2\, D'_2}{\|p_1\, D'_1 + p_2\, D'_2\|}, \tag{10}$$

$$D_2 = \frac{q_1\, D'_1 + q_2\, D'_2}{\|q_1\, D'_1 + q_2\, D'_2\|}, \tag{11}$$

and the lengths of the principal axes are the square roots of the eigenvalues

$$d_1 = \sqrt{v_p} \tag{12}$$

$$d_2 = \sqrt{v_q}. \tag{13}$$

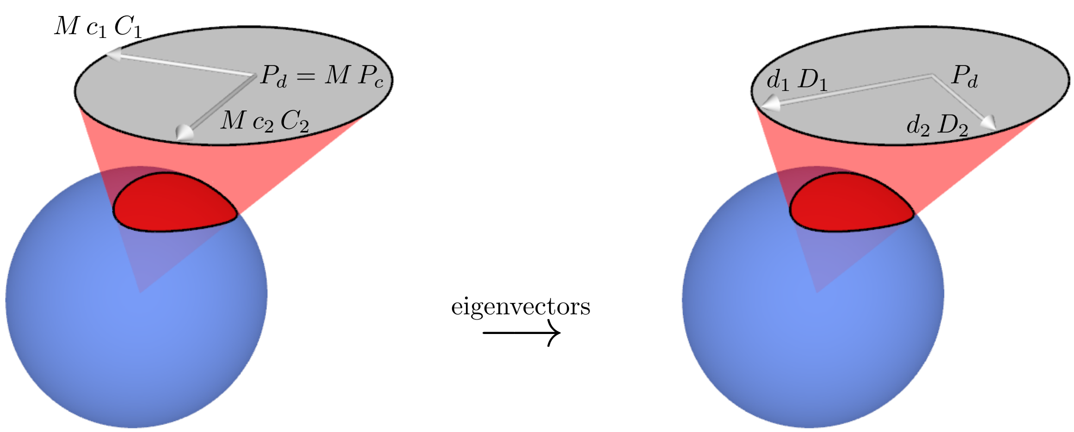

Figure 7: Computing the principal axes of the ellipse.

4 Results

Figure 8 shows results computed with our method. We computed the solid angles Ω of the ellipsoid with a Monte Carlo method and of the ellipse with Conway's analytical method [Con10]. We obtained the ellipse from the input ellipsoids with the MATLAB implementation of our method provided in Listing 1.

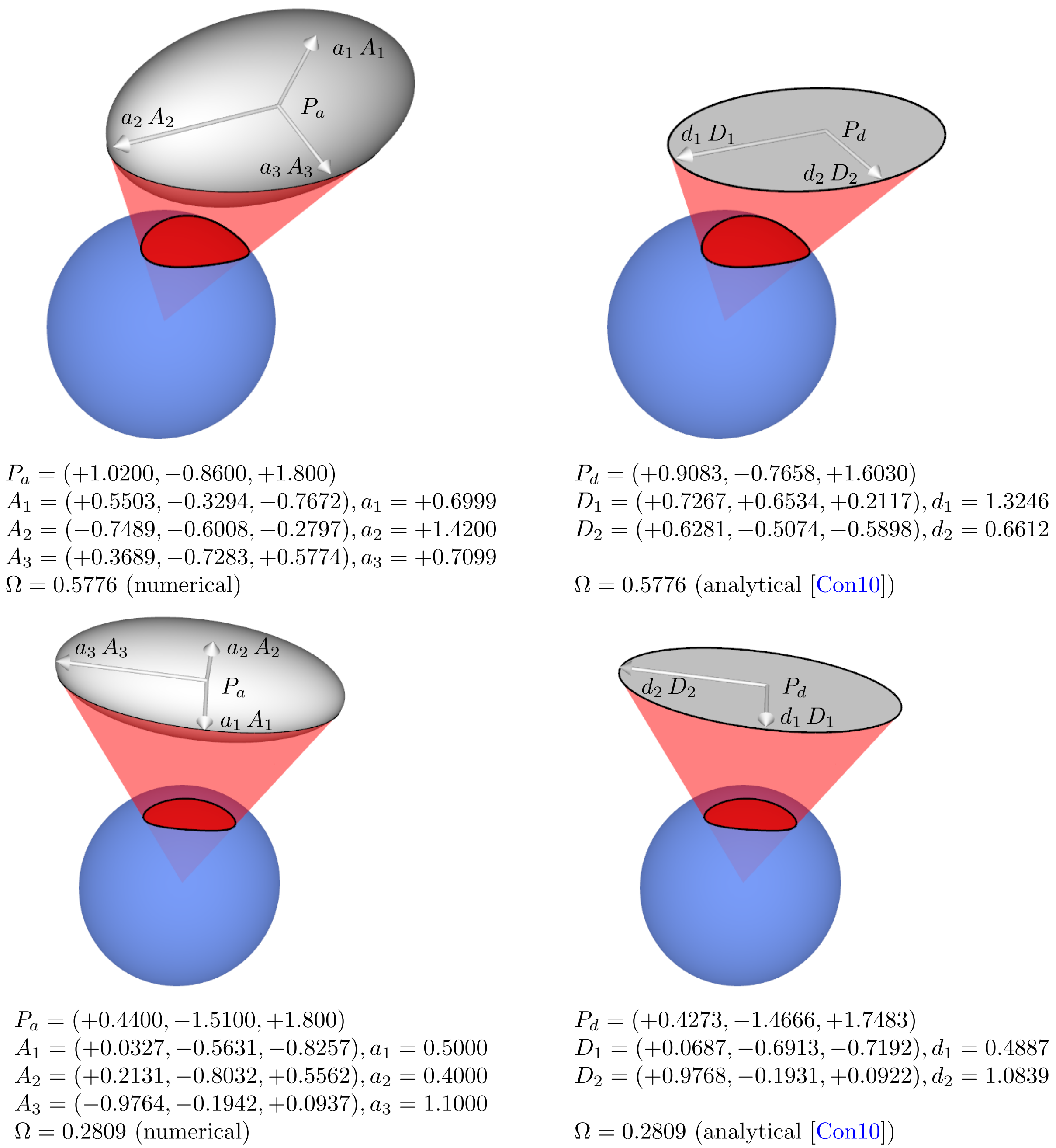

Figure 8: Result computed with our method.

5 MATLAB Implementation

```
% This function computes the ellipse that subtends the same solid angle
% as the input ellipsoid with respect to the origin O = [0 0 0]
% INPUT
% Pa: center of the ellipsoid
% A1, A2, A3: normalized directions of the principal axes of the ellipsoid
% a1, a2, a3: lengths of the principal axes of the ellipsoid
% OUTPUT
% Pd: center of the ellipse
% D1, D2: normalized directions of the principal axes of the ellipsoid
% d1, d2: lengths of the principal axes of the ellipsoid
function [Pd, D1, D2, d1, d2] = ellipsoid2ellipse(Pa, A1, A2, A3, a1, a2, a3)

    % 3.1 ellipsoid to sphere
    M = [A1' ; A2' ; A3']' * [a1 0 0 ; 0 a2 0 ; 0 0 a3] * [A1' ; A2' ; A3'];
    Minv = inv(M);
    Pb = Minv * Pa;

    % 3.2 sphere to disk
    theta = asin(1/norm(Pb));
    Pc = cos(theta)^2 * Pb;
    radius = tan(theta) * norm(Pc);
    [C1, C2] = buildOrthonormalBasis(Pc/norm(Pc));

    % 3.3 disk to ellipse
    Pd = M * Pc;
    D1_ = M * radius * C1;
    D2_ = M * radius * C2;

    % 3.4 ellipse principal axes
    Q = [dot(D1_, D1_) dot(D1_, D2_) ; dot(D1_, D2_) dot(D2_, D2_)];
    [eigenvectors, eigenvalues] = eig(Q);
    D1 = eigenvectors(1,1)*D1_ + eigenvectors(2,1)*D2_;
    D1 /= norm(D1);
    D2 = eigenvectors(1,2)*D1_ + eigenvectors(2,2)*D2_;
    D2 /= norm(D2);
    d1 = sqrt(eigenvalues(1,1));
    d2 = sqrt(eigenvalues(2,2));

endfunction

%% code from [Frisvad2012]
function [X, Y] = buildOrthonormalBasis(Z)

    if Z(3) < -0.999999
        X = [0 -1 0]';
        Y = [-1 0 0]';
        return;
    endif

    a = 1 / (1 + Z(3));
    b = -Z(1)*Z(2)*a;
    X = [1 - Z(1)*Z(1)*a, b, -Z(1)]';
    Y = [b, 1 - Z(2)*Z(2)*a, -Z(2)]';

endfunction
```

Listing 1: MATLAB implementation of our method.

References

[Con10] John T. Conway. Analytical solution for the solid angle subtended at any point by an ellipse via a point source radiation vector potential. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 614(1):17 – 27, 2010. 1, 4

[Fri12] J. R. Frisvad. Building an orthonormal basis from a 3D unit vector without normalization. *Journal of Graphics Tools*, (16):151–159, 2012. 3.2