Journal for Geometry and Graphics
Volume 29 (2025), No. 2, 199–221

3-D Shadows of 4-D Algebraic Hypersurfaces in a 4-D Perspective

Michal Zamboj[1], Jakub Řada[2]

[1] *Charles University, Prague, Czech Republic*
michal.zamboj@pedf.cuni.cz

[2] *Czech Technical University, Prague, Czech Republic*
jakub.rada@cvut.cz

Abstract. Visualizing a scene of objects in 4-D space faces several challenges. Mere projections into 3- or less-dimensional spaces usually contain overlapping parts, making them difficult to comprehend or study. Illuminating the scene can enhance intuition about its "dimensionality." Our contribution describes a geometric approach to creating visualizations of 4-D hypersurfaces represented by implicit algebraic equations without their parametrization. By geometric, we mean methods using constructions of geometric objects without their approximation, for example, by polyhedral meshes. Therefore, instead of sets of many points and operating with meshes, we work with implicitly represented hypersurfaces, their projections, contours, intersections, etc. We provide a general algorithm to find shadow boundaries in an arbitrary dimension and apply it in a 4-D space. Furthermore, we design a system of polynomial equations to construct occluding contours of algebraic surfaces in a 4-D perspective. The results of our algorithm are components of the 3-D model of a scene image represented by polynomial equations and inequalities prepared for plotting by standard computer algebra systems with visualization tools. The method is presented on three 4-D scenes with gradual complexity created in *Wolfram Mathematica*. Since algebraic methods preserve many properties of the visualized shapes, they are suitable for precise mathematical or scientific visualization. On the other hand, processing higher-degree polynomials using elimination methods places greater demands on computational time.

Key Words: four-dimensional visualization, algebraic hypersurface, shadow, implicit equation, elimination methods

MSC 2020: 51N15 (primary), 68U05, 14J70

1 Introduction

Visualizing shapes embedded in more-dimensional spaces faces several challenges. In many cases, mere projections into three- or less-dimensional spaces contain overlapping parts, making them difficult to understand. One technique that enhances intuition about the properties of shapes and their mutual relations is to visualize shadows cast on themselves and on other objects. Apart from the general case, choosing algebraic hypersurfaces defined by polynomials is often convenient. These are good candidates for visualization using computational methods of algebraic geometry and elimination theory. In this sense, instead of sets of many points and operating with meshes, we can work with implicitly represented hypersurfaces, their projections, contours, intersections, etc. Since algebraic methods preserve many properties of the visualized shapes, they are suitable for precise mathematical visualization. The disadvantage of implicit representation is the computational speed when processing polynomials of higher degrees or adding more variables.

This paper aims to improve understanding spatial properties in a four-dimensional scene containing algebraic hypersurfaces. To do so, we join theoretical geometric construction and algebraic computational methods and provide concrete examples of visualizations of four-dimensional hypersurfaces and their shadows based on implicit representations.

In particular, we show visualizations of four-dimensional algebraic hypersurfaces (3-surfaces), their contours (2-surfaces), terminators, and 3-D shadows cast on other 3-surfaces with respect to a point light source.

The process consists of two main parts – construction of shadows from an arbitrary point light source and central projection of the scene into a 3-D modeling space (usually a virtual 3-D environment in some software, AR, CR, or even a real 3-D model). First, we find the terminator with respect to an arbitrary point light source and its projection to the 3-surface on which the shadow is cast. Next, to construct a 4-D perspective image, a 3-surface given by a polynomial is intersected by its first polar (3-surface) with respect to the center of projection, and its (2-surface) contour generator is centrally projected into a modeling 3-space. While geometrically, we describe intersections of surfaces, algebraically, we need to find a polynomial that solves systems of polynomial equations with several variables (7 to 9 in the 4-D case). This leads to the use of standard computational methods such as finding a Gröbner basis or Dixon resultant. Finally, to complete the shadow, especially for 3-surfaces of degrees higher than two, we need to find the regions in their own shade. Such regions are not algebraically omitted in the elimination procedure; hence, we need to carry out further selection.

1.1 Related Work

The algebraic concepts, in particular, the use of Gröbner basis and Dixon resultant for finding solutions of polynomial systems, are described in detail in [13]. A similar technique to find the implicit representation of an occluding contour in 3-D through the Dixon resultant was applied in [16, 17]. In our experiments, we used *Wolfram Mathematica 13* implementations of algorithms for finding Gröbner basis — WM-GB [35] and the Dixon resultant — WM-Dix [23], and also Dixon resultant — Fer-Dix-KSY or improved Fer-Dix-EDF [18] implemented in software *Fermat 6.5* [19] by Lewis, see [20, 21].

In addition to our approach, where we start with surfaces given implicitly, a considerable part of previous research on computational aspects of surfaces deals with implicitization from parametric representation (e.g., [4, 21, 22, 33]). The algebraic derivation of perspective images of surfaces and reconstruction with respect to further applications is shown in [24].

Four-dimensional projections through parameterization or point coordinates are described in [25, 26, 37]. A 4-D perspective projection was also used to visualize implicitly given surfaces that arise in a complex number plane [31]. A descriptive geometry approach for constructing 4-D perspective images of a 3-sphere in 3-D from orthogonal projections is discussed in [30]. Visualizations of the 4-space, including hypersurfaces, are treated comprehensively in [1]. In [11, 39], the authors created projections of several examples of surfaces in 4-space, examined their properties geometrically and algebraically, and showed various applications. An illumination method applied to 2-D implicit surfaces embedded in 4-space is presented in [3]. Four-dimensional lighting was used to study the shades of some mathematically interesting 3-surfaces in [10]. Interactive manipulation with four-dimensional objects based on their projections or shadows in a hyperplane is elaborated in [2, 38] and through a tetrahedral mesh construction in [7, 8].

1.2 Contribution

Our approach emphasizes algebraic methods in four-dimensional visualization. Throughout the paper, we work purely with implicit representations of surfaces without the necessity of their parametrization. Visualizing shadows between 3-surfaces, we discuss their mutual relations in the 4-space. In this way, compared to previous attempts, we offer a more comprehensive perception of complex 4-D scenes projected into 3-D space. We also provide a direct method for visualization of 3-surfaces in a 4-D perspective. After all, the designs of polynomial systems for constructing tangent cones and shadow boundaries as intersections are general for any dimension.

1.3 Paper Organization

The rest of the article is organized as follows: we start with a 3-D example to describe the algorithm to find the terminator of a 2-surface and its shadow cast on another 2-surface in Section 2. Next, we generalize it into 4D and describe the 4-D perspective from a 4-space into the modeling 3-space. Section 3 is focused on concrete examples. The 3-D scene from the previous explanation is supplemented with technical details. Next, we consecutively examine and comment three 4-D scenes with respect to their geometric and computational complexity. In Section 4, we discuss the critical points of our method and propose further research directions. Section 5 summarizes the results of this paper. The polynomials in their full forms, code in *Mathematica*, computation times, and videos are provided in an online *GitHub* repository referenced in the Appendices.

2 Method

2.1 Constructing Shadow of an Algebraic Hypersurface

In the first part, we examine the process of computing a shadow in arbitrary dimension, particularly suited to 4-D scenarios but illustrated in the 3-D case.[1]

[1] The upcoming 3-D visualizations are created in *Wolfram Mathematica 13* from outputs based on implicit equations (or inequalities) formulated in variables x, y, z. Therefore, they are also usable as 3-D graphics in an interactive environment.

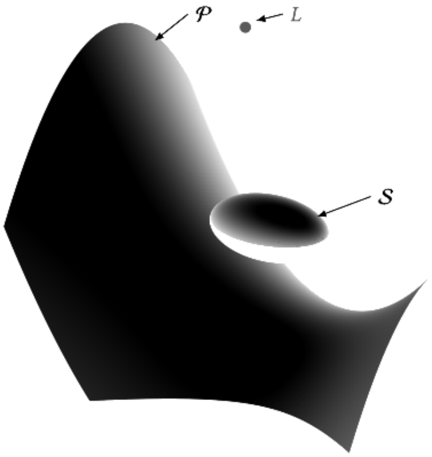


Figure 1: Initial setting of hypersurfaces $\mathcal{S}, \mathcal{P}$, and a point light source L.

2.1.1 Preliminaries

Let us have a hypersurface $\mathcal{S}$ (Fig. 1), i.e., an $(n-1)$-surface embedded in a real n-space $(n \geq 1)$ given by a polynomial equation in n variables

$$\mathcal{S}\colon \sigma = 0. \tag{1}$$

By introducing a point light source into the scene, our next goal is to determine the tangents from this point to the hypersurface. To this end, we employ polar properties together with a convenient representation in homogeneous coordinates within the projectively extended real space. The tangent points of the illuminated hypersurface then lie on its polar hypersurface.

Let us have $(x_1', x_2', \ldots, x_n', x_0')$, $(y_1', y_2', \ldots, y_n', y_0') \in \mathbb{R}^{n+1} \setminus \{(0, 0, \ldots, 0)\}$. We define the equivalence $(x_1', x_2', \ldots, x_n', x_0') \sim (y_1', y_2', \ldots, y_n', y_0')$, if there exists $\lambda \in \mathbb{R} \setminus \{0\}$ such that $(x_1', x_2', \ldots, x_n', x_0') = (\lambda y_1', \lambda y_2', \ldots, \lambda y_n', \lambda y_0')$. The projective n-space is defined as equivalance classes of $\mathbb{R}^{n+1} \setminus \{(0, 0, \ldots, 0)\}$. A point $\overline{P}(p_1', p_2', \ldots, p_n', p_0')$ of the projective space has homogeneous projective coordinates $(p_1', p_2', \ldots, p_n', p_0')$. Additionally, for $p_0' \neq 0$, we can obtain the Cartesian coordinates of $P(p_1, p_2, \ldots, p_n)$ by substituting $p_1 = \frac{p_1'}{p_0'}, p_2 = \frac{p_2'}{p_0'}, \ldots, p_n = \frac{p_n'}{p_0'}$, or vice versa. In case $p_0' = 0$, the point $\overline{P}$ represents a point at infinity. Throughout the text, $\overline{\sigma}$, $\overline{P}$,... denote representations of σ, P, ... in homogeneous coordinates.

A point $\overline{P}$ on the surface $\mathcal{S}$ is called singular if all the partial derivatives $\nabla\overline{\sigma}$ vanish at that point. Let $\overline{P}$ be a non-singular point of $\mathcal{S}$ (i.e., a regular point on the surface, or point not on the surface) and assume a polynomial $\overline{\sigma_P} = \overline{P}^T \nabla\overline{\sigma}$. The surface $\mathcal{S}_P : \overline{\sigma_P} = 0$ is called the first polar of the point $\overline{P}$ with respect to the hypersurface $\mathcal{S}$.

2.1.2 Terminator

Roughly speaking, the terminator is the boundary between the illuminated and shaded regions of a hypersurface — a curve on a 2-surface in the 3-D case, or a 2-surface on a 3-surface in the 4-D case. More precisely, it is the set of points of tangency of all tangent lines drawn from the light source.

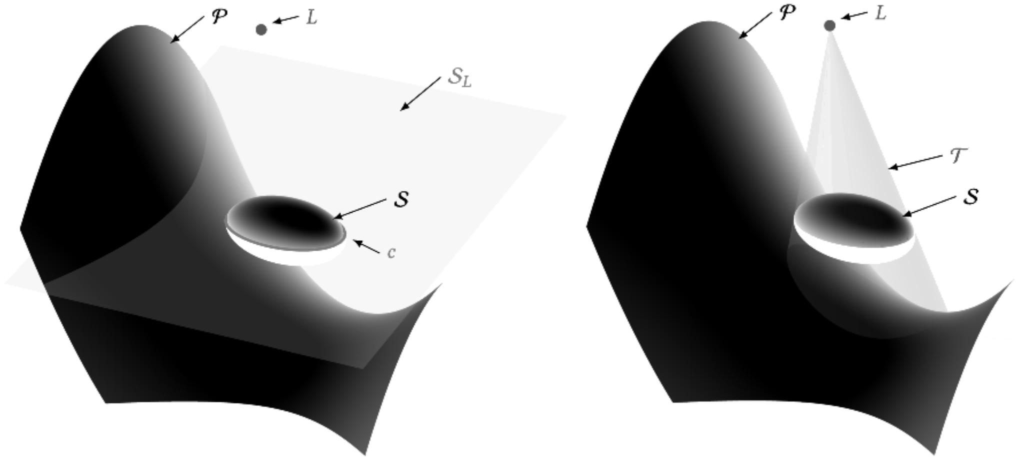


Figure 2: (Left) The polar hypersurface $\mathcal{S}_L$ of a hypersurface $\mathcal{S}$ with respect to a point light source L and its terminator c. (Right) The tangent hypercone $\mathcal{T}$ to a hypersurface $\mathcal{S}$ through a light source L and the shadow cast on $\mathcal{P}$.

Let us have a point light source $L(l_1, \dots, l_n)$ with homogeneous coordinates $\overline{L}(\overline{l_1}, \dots, \overline{l_n}, \overline{l_0})$. The terminator c (Fig. 2a, left) of the hypersurface $\mathcal{S}$ with respect to L is the intersection of the first polar

$$\mathcal{S}_L\colon \overline{\sigma_L} = \overline{L}^T \nabla \overline{\sigma} = 0 \tag{2}$$

of the hypersurface $\mathcal{S}$ with the hypersurface $\mathcal{S}$:

$$c\colon \overline{\sigma} = 0 \wedge \overline{\sigma_L} = 0. \tag{3}$$

2.1.3 Tangent Hypercones

The next step is to find an implicit representation of the tangent hypercones[2] from the light source L.

They will be used to create shadows cast on other surfaces. These are the hypercones through the terminator c. The tangent cone $\mathcal{T}$ is the set of lines LQ for all points $Q(q_1, \dots, q_n)$. The line LQ can be (parametrically) represented in a general dimension with the parameter $a \in \mathbb{R}$ as the set of points $X(x_1, \dots, x_n)$ satisfying the following n equations (one equation for each coordinate):

$$aL + (1-a)Q = X. \tag{4}$$

Thus, the implicit equation of $\mathcal{T}$ is the solution of the system:

$$\mathcal{T}\colon \sigma(Q) = 0 \wedge \sigma_L(Q) = 0 \wedge aL + (1-a)Q - X = 0, \tag{5}$$

where $\sigma(Q)$ and $\sigma_L(Q)$ denote polynomials σ and σ_L in variables Q. By computing the Gröbner basis or Dixon resultant of the system

$$\{\sigma(Q), \sigma_L(Q), aL + (1-a)Q - X\}$$

[2] For the sake of readability, we use terms cones and hypercones instead of more proper terms conical surfaces, conical hypersurfaces, …, over the paper.

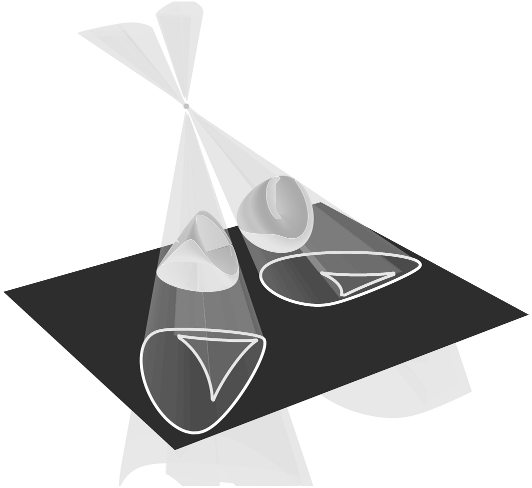


Figure 3: (Left) Roman surface: $-2(x-2)(y-1)z+(x-2)^2(y-1)^2+(x-2)^2z^2+(y-1)^2z^2=0$ and (right) Cross-cap: $\Big((x+1)^2+y^2\Big)\Big((x+1)^2+z^2\Big)+(x+1)^2+\frac{y^4}{4}+y^2z=0$, their parts separated by the first polars, and polar boundaries projected to a plane $z+2=0$. Without omission of the self-shaded parts.

and eliminating variables a, q_1, …, q_n, we obtain the polynomial θ in variables x_1, …, x_n representing the tangent hypercone

$$\mathcal{T}\colon \theta = 0 \text{ (Fig. 2b, right)} \tag{6}$$

2.1.4 Shadow Cast on an Algebraic Hypersurface

Now, we find the shadow cast by $\mathcal{S}$ on itself and on another algebraic hypersurface $\mathcal{P}$ given by a polynomial equation

$$\mathcal{P}\colon \pi = 0. \tag{7}$$

The boundary of the shadow cast by $\mathcal{S}$ on itself — terminator, is the intersection of $\mathcal{T}$ with $\mathcal{S}$. The selection of illuminated parts is carried out in the following steps:

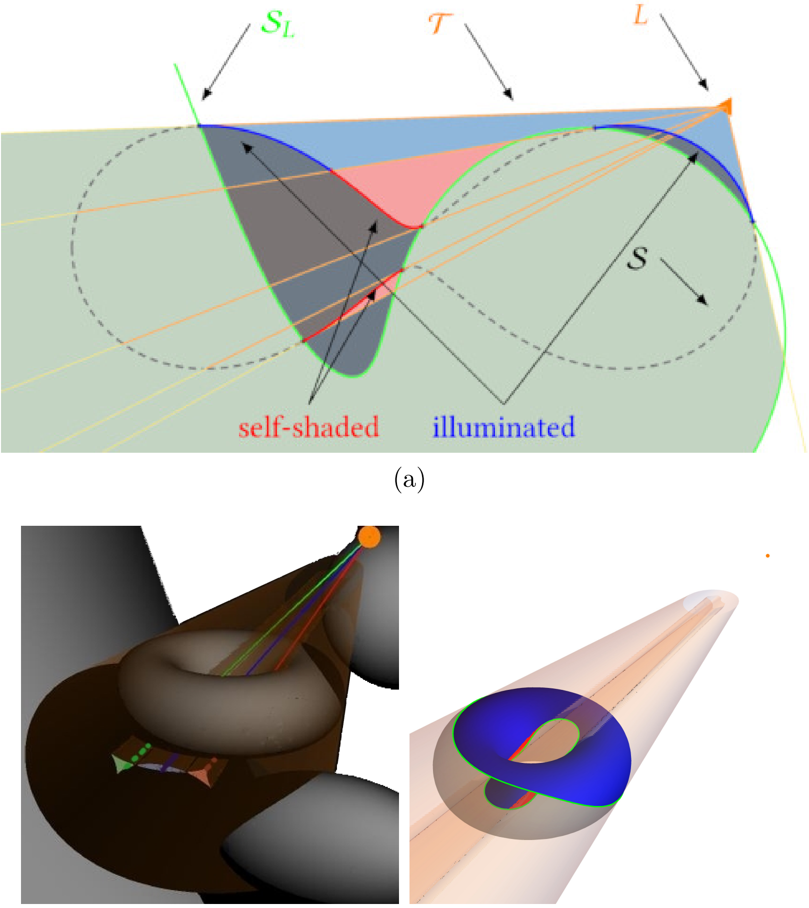
1. The n-space is divided by the first polar to subsets ($\sigma_L > 0$ or < 0), and the illuminated part is in the same subset as the source of light L. The second part is in the shade.[3]

Since we cannot always divide the inner and outer parts, the selection fails with non-orientable or self-crossing surfaces (e.g., see Fig. 3). Hence, for simplicity, we assume non self-crossing orientable bounded hypersurfaces (or at least their terminators are bounded) (see discussion in Section 4). For polynomials of degrees higher than two, some regions of the hypersurface $\mathcal{S}$ can still be in their own shade (the tangent hypercone intersects itself), and we have to omit them from the final selection of the illuminated parts (Figs. 4a, 4b).

2. Decompose the tangent hypercone into conical subregions (subcones) divided by the terminator.[4] Eliminate empty subcones.
3. Select the parts of the hypersurface closest to the light source in non-empty subcones. This is carried out by constructing rays from the light source in each subcone and finding

[3] The method is implemented in 4-D scenes up to this point. The upcoming decomposition to subcones seems, in most cases, computationally unfeasible.

[4] We use cylindrical algebraic decomposition. The results of the decomposition are distinct subregions represented by polynomial equations and inequalities (see [34] for details and implementation).



(a)

(b)

Figure 4: (a) A 2-D situation of a Cassini oval (degree 4) and its self-shading from the point light source. The green curve is the first polar, dividing the plane into two areas. The area that does not contain the light source is excluded. The subcones in 2-D case are plane angles bounded by the rays from the light source. The blue arcs represent illuminated parts, and the red arcs are in the self-shade. ((b) left) A shadow of the torus cast on a surface without the self-shaded parts omitted. ((b) right) Decomposition of a torus. The green curve is the terminator. The blue region is illuminated. The red regions are in the accepted subregion divided by the first polar but in the shade of the blue region.

the region with the intersection nearest to the source.[5] The selected nearest parts are illuminated, and the rest is in shade.

The boundary of the shadow cast by $\mathcal{S}$ on $\mathcal{P}$ is the intersection of its tangent hypercone $\mathcal{T}$ with $\mathcal{P}$. The final shadow contains inner points in the shade, i.e., inside the subcones containing previously selected regions.

If a scene contains more hypersurfaces, some of them might intersect, so we would not be able to distinguish their order with respect to the light source. In such cases, the selection algorithm can be further generalized for a hypersurface $\mathcal{Z}$ as a composition of hypersurfaces

[5] In our *Mathematica* implementation, this is carried out by finding points inside the subregions using the function ‘FindInstance‘. Subsequently, ‘Solve‘ or ‘NSolve‘ (for numerical approximations) is used to determine the intersections of rays with other subregions. The remaining steps consist of sorting, for each ray, the intersection closest to the light source.

$\mathcal{S}_1, \dots, \mathcal{S}_k$, $k \geq 1$:

$$\mathcal{Z}\colon \sigma_1 \cdots \sigma_k = 0. \tag{8}$$

However, this generally works when the light source is outside (noting that the notion of “outside” is not yet defined) all composed hypersurfaces. Otherwise, the selection of the illuminated subsets divided by the first polars must be carried out for some factors separately. For example, in Fig. 5, we can define a surface $\mathcal{Z}$ as the composition of a sphere, ellipsoid, and torus. The unbounded hyperbolic paraboloid would be treated separately.

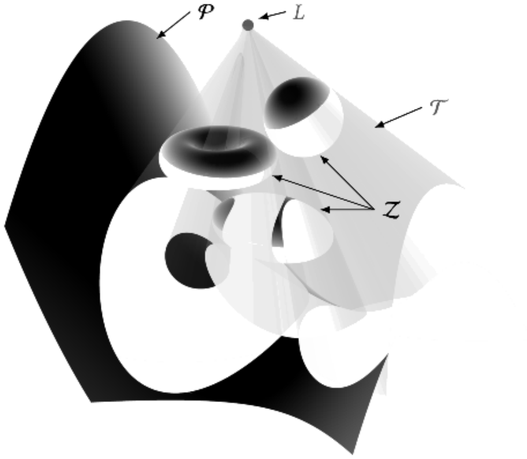


Figure 5: The final visualization of a scene with a sphere, torus, and ellipsoid casting shadows on themselves and on a hyperbolic paraboloid.

2.2 The Principle of a 4-D Perspective

Figure 6 shows a 2-D view of the correspondence between the coordinates of a real point $P(p_x, p_y, p_z, p_w)$ and its centrally projected image $P^\nu(p^\nu_x, p^\nu_y, p^\nu_w)$ into the modeling 3-space $\nu(x, y, w)$ (an analogy to a picture plane in a 3-D perspective). Assume an orthogonal coordinate system (x, y, z, w) placed in the point O in ν, with the z-axis perpendicular to ν and the center of projection C in the oriented perspective distance d from ν such that $CO \perp \nu$. Observing the homothety, the coordinates of the 4-D perspective image P^ν of $P \neq C$ are

$$(p^\nu_x, p^\nu_y, p^\nu_w) = \left(d\frac{p_x}{d - p_z}, d\frac{p_y}{d - p_z}, d\frac{p_w}{d - p_z} \right) \tag{9}$$

(in the case of $p_z = 0$ the coordinates do not change; if $p_z = d$ the image is improper).

To find the implicit representation of the 4-D perspective image (3-D occluding contour) of an algebraic surface

$$\mathcal{S}\colon \sigma = 0 \tag{10}$$

(in variables x, y, z, w), we use a similar idea as in constructing shadows. Let

$$\mathcal{S}_C\colon \overline{\sigma_C} = \overline{C}^T \nabla \overline{\sigma} = 0, \tag{11}$$

in homogeneous coordinates be the first polar of $\mathcal{S}$ with respect to C and

$$\sigma_C = 0 \tag{12}$$

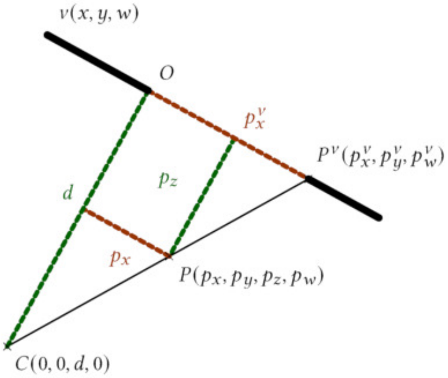


Figure 6: The principle of a 4-D perspective projection from a center C onto a modeling 3-space $\nu(x,y,w)$. 2-D orthogonal projection view on the 4-D perspective shows the homothety between the pre-image in a 4-space and its perspective image in the modeling 3-space from the center of projection.

its dehomogenized equation. Since we are projecting to a 3-space instead of an arbitrary hypersurface, we can conveniently use the derived rules of the 4-D perspective mapping (Eq. (9)).

Let $Q(q_x, q_y, q_z, q_w)$ be a point on a contour generator

$$c_C\colon \sigma = 0 \wedge \sigma_C = 0. \tag{13}$$

Elimination of q_x, q_y, q_z, and q_w from the system of polynomials

$$\left\{\sigma(Q), \sigma_C(Q), x - d\frac{q_x}{d-q_z}, y - d\frac{q_y}{d-q_z}, w - d\frac{q_w}{d-q_z}\right\}$$

leads to a polynomial σ^ν in (x,y,w) such that its zero set represents the 4-D perspective image $\mathcal{S}^\nu$ of the surface $\mathcal{S}$.

Assuming a point light source L, the terminator c^ν is, in this case, a 2-surface projected via the system of polynomials

$$\left\{\sigma(Q), \sigma_L(Q), x - d\frac{q_x}{d-q_z}, y - d\frac{q_y}{d-q_z}, w - d\frac{q_w}{d-q_z}\right\},$$

similarly as above.

For the sake of better understanding, we also visualize the tangent hypercones $\mathcal{T}$ of $\mathcal{S}$ from L, when possible (cf. Subsections 3.2.4 and 3.2.5). The tangent hypercones $\mathcal{T}$ are 3-surfaces given by Eq. (6), and the contours of their images $\mathcal{T}^\nu$ are created by the same procedure as images $\mathcal{S}^\nu$ of $\mathcal{S}$.

The final shadows are three-dimensional regions bounded by 2-surfaces in the 4-space. Conveniently, in a 4-D perspective, we can find implicit equations of the 2-surface boundaries in the modeling 3-space. To do so, we need to find the zero set of the system of polynomials

$$\left\{\sigma(Q), \theta(Q), x - d\frac{q_x}{d-q_z}, y - d\frac{q_y}{d-q_z}, w - d\frac{q_w}{d-q_z}\right\},$$

representing the perspective image of the intersection of the surface $\mathcal{S}$ and the tangent cone $\mathcal{T}$. The last step is to select the regions according to Section 2.1.4. In a 4-D perspective, we only highlight shadow boundaries so that we can see through 3-D images.

3 Experimental Results and Technical Details

In this section, we review several examples and comment on technical details.

3.1 3-D Scene

3.1.1 “Implicit Bakery”, Fig. 5

See the text file with equations and video in Attachment 1.

The 3-D scenario from Section 2 shows a surface $\mathcal{Z}$ composed of three implicitly given factor surfaces: sphere $\mathcal{S}_1$, torus $\mathcal{S}_2$, and ellipsoid $\mathcal{S}_3$

$$\mathcal{S}_1\colon (x-1)^2+(y+4)^2+(z-5)^2-4=0, \tag{14}$$

$$\mathcal{S}_2\colon \Big((x-1)^2+(y-1)^2+(z-2)^2+3\Big)^2-16(x-1)^2+16(y-1)^2=0, \tag{15}$$

$$\mathcal{S}_3\colon 4(x-3)^2+(y+1)^2+4(z+2)^2-12=0. \tag{16}$$

We describe constructions of shadows cast between them and their shadows cast on a hyperbolic paraboloid $\mathcal{P}$

$$\mathcal{P}\colon 2(y+3)^2-2(x-5)^2-25(z+7)=0 \tag{17}$$

from the light source $L(-1,-2,10)$.

The polynomial of $\mathcal{Z}$ has degree eight, but we treat its decomposition into factors (factor surfaces $\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3$). In this case, the factor surface of the highest degree, four, is the torus.

A terminator line of each “factor surface” is its intersection with the first polar with respect to L. Explicitly, terminators are the four sets of points that satisfy the following pairs of equations:

factor surface		first polar	
$\mathcal{S}_1$: Eq. (14)	$\wedge$	$19+2x-2y-5z=0$	(18)
$\mathcal{S}_2$: Eq. (15)	$\wedge$	$128-30x+12x^2+2x^3-29y-10xy$ $+3x^2y+10y^2+2xy^2+3y^3-104z+8xz$ $-8x^2z+4yz-8y^2z+40z^2+2xz^2+3yz^2$ $-8z^3=0$	(19)
$\mathcal{S}_3$: Eq. (16)	$\wedge$	$16x+y-48z-131=0$	(20)
$\mathcal{P}$: Eq. (17)	$\wedge$	$24x+4y-25z-708=0$	(21)

If a surface has a degree n, its first polar has a degree $(n-1)$, and the degree of the terminator line is due to Bézout’s theorem $n(n-1)$. Therefore, the terminator line of the entire scene $\mathcal{Z}$ would have the degree $8\cdot 7=56$. However, after factorization, the factor surfaces will have degrees 2 for the quadrics and 12 for the torus. The same holds for degrees of tangent cones [5].

Next, we omit the shaded subsets divided by the first polar. In this case, the points on illuminated regions have non-negative values in the equations of the first polars.

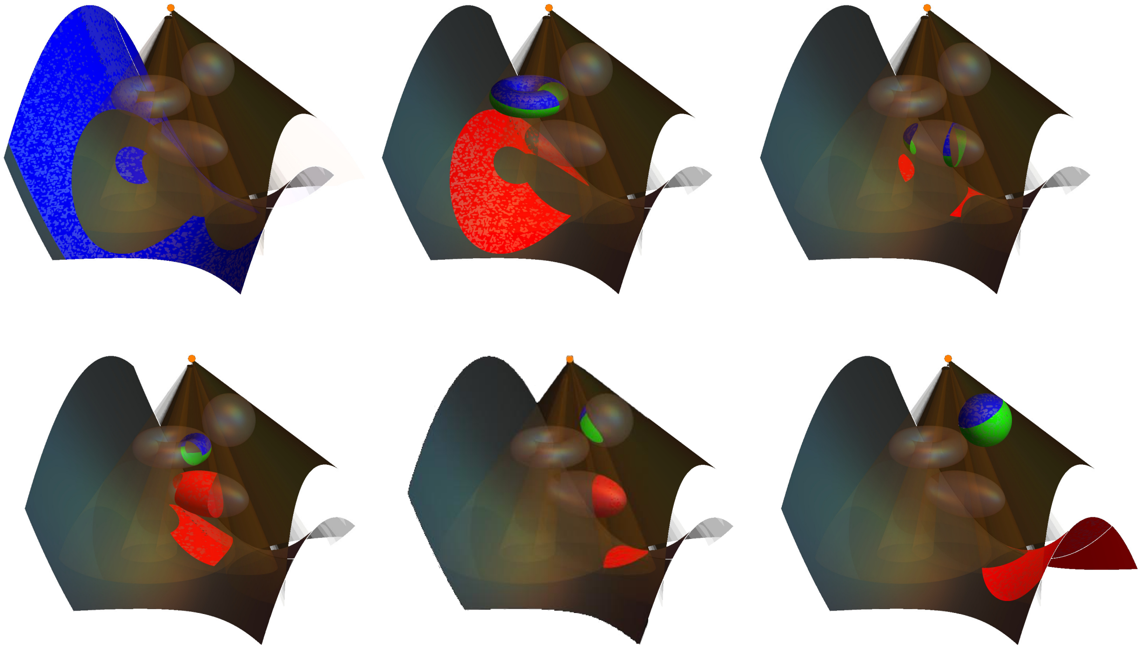


Figure 7: A selection of the regions contained in subcones. Each picture shows a different subcone of the scene $\mathcal{Z}$ consisting of a sphere $\mathcal{S}_1$, torus $\mathcal{S}_2$, and ellipsoid $\mathcal{S}_3$. The blue regions are illuminated, the green regions are excluded by the first polars, and the red regions are in the shade of the blue regions with a shorter distance to the light source. The top left figure shows the complement of the union of all subcones in the 3-space; hence it distinguishes the shadow of the scene on the hyperbolic paraboloid $\mathcal{P}$. The rest of the empty regions are not shown.

Tangent cones are also treated separately for each factor surface. For example, for the torus $\mathcal{S}_2$, we have the following system of polynomial equations:

$$\begin{aligned} &\text{Eq. (15)},\\ &\text{Eq. (19)},\\ aq_1 - (a-1) - x &= 0,\\ aq_2 - 2(a-1) - y &= 0,\\ aq_3 + 10(1-a) - z &= 0. \end{aligned} \tag{22}$$

Eliminating q_1, q_2, q_3, and a leads to an 8th-degree polynomial of the tangent cone. Similarly, we find the rest of the tangent cones of the surfaces in the scene.

Next, we decompose the regions bounded by the system of tangent cones into all subcones and obtain five nonempty and three empty intersecting regions in this case (Fig. 7). As a result of cylindrical algebraic decomposition, the subcones are represented by implicit equations and inequalities.

In the last step, we trace subsets separated by the first polars over each subcone and choose the region nearest to the light source.

3.2 4-D Scenes

3.2.1 Understanding the 4-D Visualizations

Let us give a few remarks on how to understand the 4-D visualizations below:

- The visualizations are 3-D models (occluding contours), and the figures in the paper are only 2-D images of the 3-D scenes.
- Standing in a gallery in front of an actual 2-D painting in the 3-D linear perspective, we can find the position of our eye such that the picture makes an illusion of 3-D space. This is unreachable in the 4-D perspective because we cannot leave the 3-D space of the 3-D image (of a 4-D object).
- One could think of observing the picture from the inside, i.e., we can reach any location in the 3-D static image. On the contrary, the change of the 4-D eye/camera position would change the contours of the objects.
- For better orientation, we always attach an image of a reference hypercube with the center in $(0, 0, 0, 0)$ (Fig. 8).
- The default value of the oriented eye distance in figures is $d = -6$, and its coordinates are $(0, 0, d, 0)$.
- To understand the (3-D) spatial properties of the modeling 3-space, we keep the software lighting properties of the 3-D graphics.

3.2.2 Technical Notes on the Implementation of the 4-D Perspective

Let us reflect on some pros and cons of the 4-D perspective:

+ Intersections of 3-surfaces in a 4-space are 2-surfaces; hence, we can visualize them using only one implicit equation in the modeling 3-space. The equation can be obtained directly from the corresponding polynomial system.

– 4-D perspective images should include "inner points" of hypersurfaces, but we only show their occluding contours. Otherwise, we would not see images that overlap in the modeling 3-space. An analogous problem occurs in 3-D perspective, where we can imagine a smaller object in front of a bigger object, so the perspective image of the smaller object would lie inside the image of the bigger object. Although we can easily decompose figures in the picture plane, we would not see much in the modeling 3-space. The understanding becomes very unclear with non-closed surfaces, where parts of hypersurfaces might seem to go through their contours (see Subsection 3.2.4).

– It is important to note that in 4-D, we only visualize contours of surfaces and their shadows, not their interior. The algorithm used to identify and fill unshaded and shaded regions of surfaces within tangent cones has been successfully applied in 3-D scenes. However, it encounters technical challenges in 4-D. In our experiments, the computation time required to locate points on the hypersurface (defined by a polynomial equation in four variables) and within the subcone (described by a system of polynomial inequalities in four variables) exceeded reasonable limits. Whether a parametric representation might be more suitable for this purpose remains an open question.

3.2.3 4-D Scene: "HyperQuadrics" Fig. 10b

See the text file with equations, code, and video in Attachment 2.

In the first 4-D case (Fig. 9), we have a 3-ellipsoid

$$\mathcal{S}\colon \frac{(x+2)^2}{4} + \frac{(y+1)^2}{2} + z^2 + (w-4)^2 - 1 = 0 \tag{23}$$

casting a shadow on a 3-sphere

$$\mathcal{P}\colon (x+5)^2 + (y+6)^2 + (z-2)^2 + (w+3)^2 - 36 = 0 \tag{24}$$

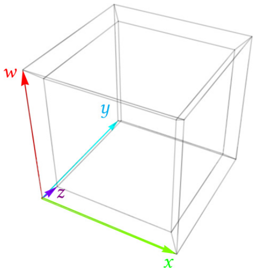


Figure 8: A reference hypercube in the 4-D perspective. The colors of the arrows mark the directions of the coordinate axes: Green → x, Blue → y, Purple → z, Red → w.

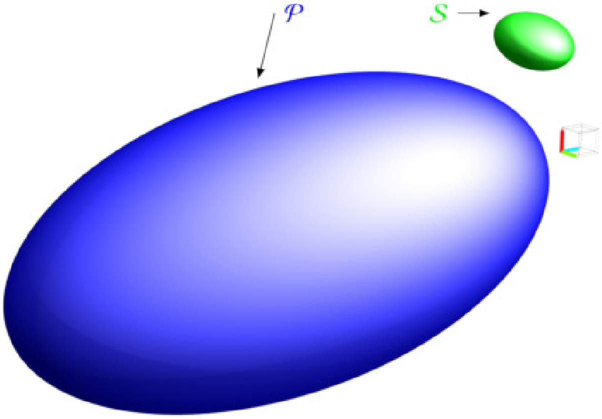


Figure 9: A 4-D scene with a 3-sphere $\mathcal{P}$ and a 3-ellipsoid $\mathcal{S}$ in a 4-D perspective.

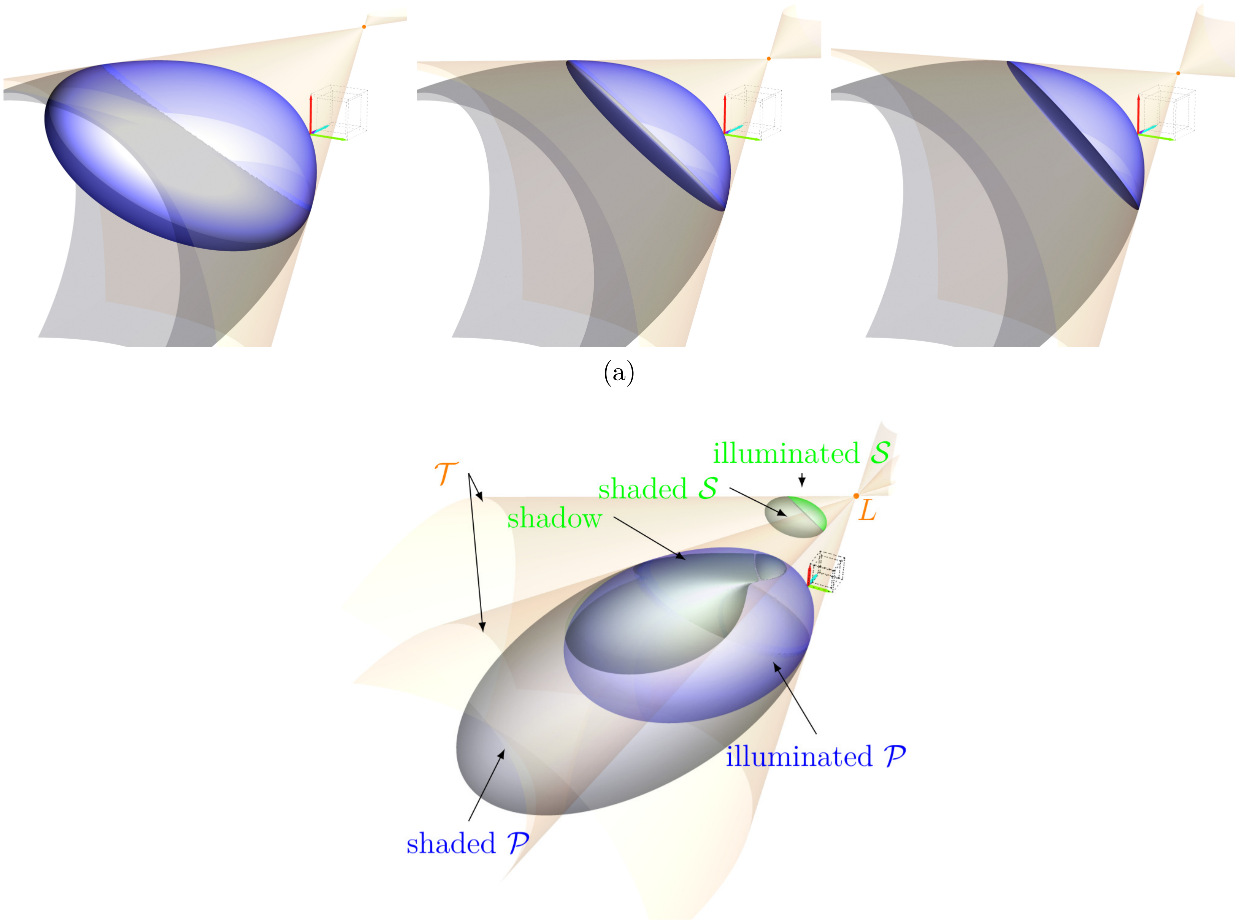
from the light source $L(1, 1, -1.5, 5)$.

In this scene, the given 3-surfaces and their tangent hypercones have degrees 2. The highest degree, 4, has the two-dimensional boundary of the shadow cast by the 3-ellipsoid on the 3-sphere. The occluding contours $\mathcal{S}^\nu$ and $\mathcal{P}^\nu$ of $\mathcal{S}$ and $\mathcal{P}$ are ellipsoids, and the same holds for their terminator 2-surfaces $c_\mathcal{S}, c_\mathcal{P}$ and their images $c^\nu_\mathcal{S}, c^\nu_\mathcal{P}$.

In the case of hyperquadrics, we can easily deduce the transition of the visible illuminated parts with respect to the given 4-D perspective (Fig. 10a). The first polar of the 3-sphere $\mathcal{P}$ with respect to the perspective center C is a 3-space, dividing the 4-space into two half-4-spaces. Thus, we have the following three cases:

1. The shape of the illuminated part is in a special position when the light source L is in the polar 3-space (with respect to C), i.e., the terminator 2-surface of the 3-sphere with respect to L degenerates to an ellipse including inner points.
2. When L is in the same half-4-space as the center C, the visible illuminated part includes the inner points of the terminator 2-surface.
3. If the light source L and the center C lie in the opposite half-4-spaces, we must exclude the inner points of the terminator 2-surface.

Visual interpretation: Let us visually examine Fig. 10b. First, we should remind ourselves that each surface in the figure represents only the boundary of a three-dimensional volume, which itself is a projection of a (hyper-)surface in four-dimensional space. Furthermore, we are no longer observers viewing the scene in 3-D perspective; instead, we are situated within the 3-D model depicted in the figure. Hence, we cannot simply stand at a single point in a


(a)

(b)

Figure 10: (a) Transitions of the illuminated part of the 3-sphere $\mathcal{P}$ from the point light source $L(1, 1, z_L, 5)$ moving in the z-direction: (left) $z_L = 2.5$, (center) $z_L = 7.5$, (right) $z_L = 12.5$. The point L lies in the first polar 3-space for $z_L = 9.5$. The illuminated part is bounded by the occluding contours of the 3-sphere and terminator 2-surface with respect to L. (b) A 4-D scene with the shadow of a 3-ellipsoid $\mathcal{S}$ on a 3-sphere.

gallery to see the objects from the figure appearing as real. However, within the 3-D image, we can move to any location and, therefore, see the interiors of objects.

In this case, the light source L is closer to the 3-ellipsoid $\mathcal{S}$, and we can easily see its green illuminated part. The boundary of the illuminated region on the 3-sphere $\mathcal{P}$ is depicted in blue. The reader can imagine a 3-D analogy involving a light source in front of a sphere: the illuminated spherical disk appears as an elliptical disk within the elliptical contour of the sphere.

Returning to the figure, we observe a self-intersecting surface within the blue quadric. This surface represents the boundary of the shadow cast by the 3-ellipsoid onto the 3-sphere. In this case, the self-intersection of the shadow indicates that, from the 4-D viewer's perspective, the shadow overlaps itself. In a 3-D analogy, we can imagine one sphere casting a shadow onto another, where the shadow falls partly on the far side of the second sphere and is therefore partially hidden from our point of view. If we visualize only the shadow, including its hidden parts, it may appear to overlap itself.

We encourage the reader to explore this situation further in the video referenced in At-

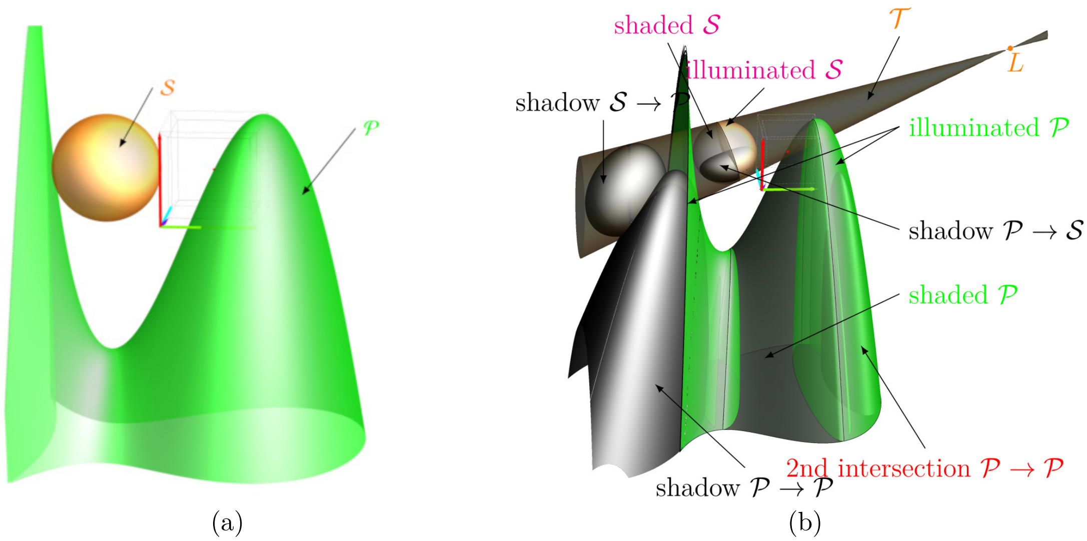

$\mathcal{S}$
$\mathcal{P}$
(a)
shaded $\mathcal{S}$
$\mathcal{T}$
L
illuminated $\mathcal{S}$
shadow $\mathcal{S} \to \mathcal{P}$
illuminated $\mathcal{P}$
shadow $\mathcal{P} \to \mathcal{S}$
shaded $\mathcal{P}$
2nd intersection $\mathcal{P} \to \mathcal{P}$
shadow $\mathcal{P} \to \mathcal{P}$
(b)

Figure 11: (a) A 4-D scene with a 3-sphere $\mathcal{S}$ and a 3-surface $\mathcal{P}$ of degree 3 in a 4-D perspective. (b) An illumination of a 3-sphere $\mathcal{S}$ and its shadow on a 3-surface $\mathcal{P}$ of degree 3 from a point light source L in a 4-D perspective. The figure contains the excess intersection of the tangent cone (not visualized) of $\mathcal{P}$, and the self-shaded region of $\mathcal{P}$ is not excluded from the illuminated part.

tachment 2, which shows a light source moving in the z-direction.

3.2.4 4-D Scene: "Full HyperMoon Between HyperMountain", Fig. 11b

See the text file with equations and code in Attachment 3.

The second 4-D scene (Fig. 11a) shows a more complicated situation. Let us have a 3-sphere

$$\mathcal{S}\colon w^2 + (x+1)^2 + y^2 + (z+1)^2 - \frac{1}{4} = 0 \tag{25}$$

casting a shadow on a 3-surface of degree 3

$$\mathcal{P}\colon (x-1)(x+2)x + y^2 + z^2 + w = 0 \tag{26}$$

from the light source $L(5, 1, 1, 2)$.

The contours of the occlusion are 2-surfaces of degree 2 for $\mathcal{S}^\nu$ and degree 6 for $\mathcal{P}^\nu$. In this case, it is hard to perceive the 4-D spatial properties of the scene from the contours. In particular, we cannot intuitively grasp the unbounded 3-surface $\mathcal{P}$.

Let us bring more light to this scene. After finding the terminator 2-surfaces and tangent hypercones[6] to the 3-surfaces through the vertex L, we can create shadows between the 3-surfaces. The 3-sphere casts a shadow on $\mathcal{P}$. The contour of the intersection of the tangent hypercone to $\mathcal{S}$ with $\mathcal{P}$ is a 2-surface of degree 6. It consists of two disjoint parts, and the part closer to L is omitted in Fig. 11b. The image of the 2-surface boundary of the shadow of $\mathcal{P}$ cast on $\mathcal{S}$ is given by a polynomial of degree 14 and the contour of the shadow of $\mathcal{P}$ on itself is a surface of degree 18.

[6] The computation of the projection of the tangent hypercone to $\mathcal{P}$ was terminated after an excessively long runtime (5–10h).

Visual interpretation: The power of illumination is evident when comparing Figs. 11a and 11b. In the figure without illumination, we cannot make any statement about the relative positions of $\mathcal{S}$, $\mathcal{P}$, and L. However, adding illumination clarifies the scene, revealing both the self-shaded regions and the shadow of $\mathcal{S}$ cast onto $\mathcal{P}$. Again, we must keep in mind that we, as three-dimensional beings, are able to move freely within the 3-D scene. However, from the perspective of a 4-D viewer, it is possible to see the 3-D shadows inside the surface $\mathcal{P}$.

3.2.5 4-D Scene: "HyperRing", Fig. 13a

See the text file with equations, code, and video in Attachment 4.

The last 3-surface (Fig. 11a) is given by a polynomial of degree 4:

$$\mathcal{S}\colon\, (x-1)^2 + ((w-2)^2 + y^2 - 4)^2 + z^2 - 1 = 0. \tag{27}$$

The first polar $\mathcal{S}_L$ with respect to the light source $L(0, 2, -2, 4)$ is:

$$\mathcal{S}_L\colon\, 4w^3 - 3x + x^2 + 4w^2(-8+y) - 16y^2 + 4y^3 + 4w(16 - 4y + y^2) - 2z + z^2 = 0. \tag{28}$$

Let us have a 3-space

$$\mathcal{P}\colon\, w + 2 = 0. \tag{29}$$

The occluding contour $\mathcal{S}^\nu$ is after elimination given by a polynomial of degree eight in 72 terms (in variables x, y, z). After elimination, the terminator surface c^ν is given by a polynomial of degree eight in 146 terms (in variables x, y, z). The tangent cone $\mathcal{T}$ generated by the terminator c (intersection of $\mathcal{S}$ and $\mathcal{S}_L$) is after elimination given by a polynomial of degree eight in 483 terms (in variables x, y, z, w). The occluding contour of the shadow of $\mathcal{S}$ on $\mathcal{P}$ is a 2-surface of degree eight (Fig. 13a).

Visual interpretation: In this case (Fig. 13a), the surface $\mathcal{S}$ casts a shadow onto the "ground 3-space" $\mathcal{P}$. The figure represents a scene analogous to Fig. 3. We can see the boundary of the surface $\mathcal{S}$, the boundary of its illuminated part c_L, and the projections of this boundary onto $\mathcal{P}$. A rotation of the 3-D model can be seen in the video referenced in Attachment 4. A closer look at the detail of the terminator in Fig. 13b reveals that the contour of the terminator c_L intersects itself for certain positions of the light source L. From the 4-D perspective, the terminators might overlap, self-intersect, or even twist.

However, based on the sequence of terminator contours — where intersections appear and disappear in pairs — we assume that the terminator overlaps itself in the 4-D perspective, in a manner similar to the situation described in Subection 3.2.3 (Fig. 10b). Nevertheless, this intuitive visual interpretation still remains to be algebraically validated.

4 Discussion and Future Work

Throughout the paper, we tried to bring the "most universal" solution to visualizing shadows of algebraic hypersurfaces. In this sense, the method presented in Subsection 2.1 works in a general dimension; the key point is to construct tangent cones with Eq. (5). Illumination of a scene, both in 3-D and 4-D, is a complex process, and we should always consider the properties and positions of objects in the scene. The critical features of objects in our approach are the boundedness, orientability, and degree of a hypersurface and its terminator. We have made a pragmatic decision in Section 2.1.4 to restrict our analysis to non-intersecting,

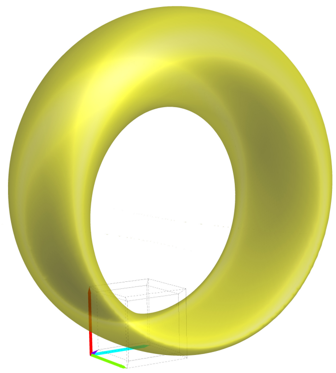
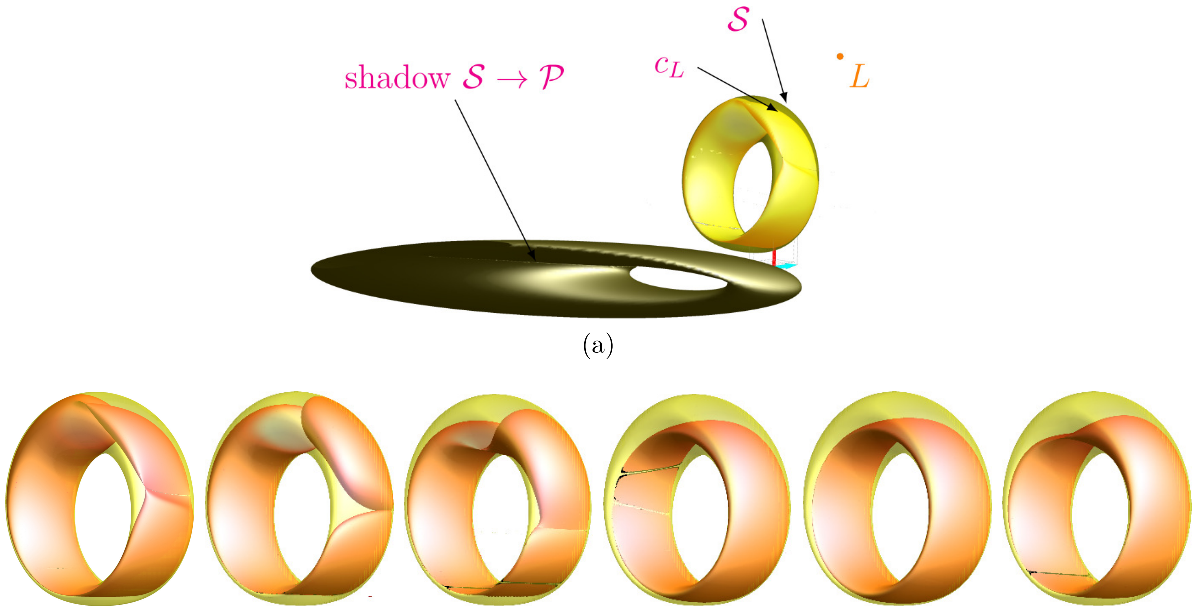


Figure 12: A surface $\mathcal{S}$ of degree 4 in a 4-D perspective.



(a)

(b)

Figure 13: (a) The shadow cast by a 3-surface $\mathcal{S}$ on a 3-space $\mathcal{P}$. Since the terminator overlaps itself, we cannot properly distinguish the illuminated visible part of $\mathcal{S}$. (b) A detail of the contours of the surface $\mathcal{S}$ and its terminator c_L when illuminated by a point light source L moving away from the perspective center in the z-direction. In the leftmost figure, the coordinates of L are $(0, 2, -2, 4)$. The z-coordinates of L in the sequence are 0, 2, 6, 21, and 30.

orientable, and bounded hypersurfaces. The assumptions of non-intersection and orientability are necessary, since without them it is impossible to consistently distinguish the interior from the exterior of an object. This ambiguity complicates the selection of illuminated regions and, in the 3-D case, corresponds to the classical problem of sidedness. In the 4-D setting, however, we construct only contours, so sidedness does not arise. These restrictions could, in principle, be relaxed by employing more computationally demanding algorithms. In particular, applying cylindrical algebraic decomposition to the hypersurface before polarity condition would eliminate singularities and yield locally orientable regions, after which a consistent interior–exterior distinction could be established. The assumption of boundedness may appear overly restrictive, but it prevents complications arising from terminators at infinity, which would again obscure the notion of interior. We foresee a possible theoretical solution in projectivization via homogeneous coordinates, although this would require introducing an additional variable into the tangent-hypercone computation. This, in turn, would increase the computational cost of the elimination methods. Additionally, the position of the point light source and the projection center is crucial to define the inside and outside of a hypersurface or its part and, consequently, its visibility. Elimination methods such as Gröbner basis assume that coefficients lie in a field (in our case, the rationals). Hence, if we wish to use an arbitrary floating-point representation as a light source, we must first rationalize it using the standard procedures implemented in computer algebra systems. Otherwise, the coefficients of the polynomials describing the polars may contain non-rational and inexact values, which can lead to incorrect results. The degree of a hypersurface is another key factor influencing computational complexity. Evidently, the appropriate setting and the choice of elimination method (Gröbner basis or the Dixon resultant) plays a crucial role in the computation time for higher-degree polynomial systems. The computational complexity also depends on the form of the polynomial, i.e., transformations of the hypersurface. In the worst case, the complexity of Gröbner basis computation is doubly exponential in the degree of the input polynomials (see, e.g., [9]), while the complexity of the Dixon resultant is factorial in the number of variables and polynomial in the degrees (see [29]). Nevertheless, methods for reducing the complexity of elimination-based computations remain an active area of research and are widely implemented in modern computer algebra systems. Since our visualizations were created in *Mathematica*, we restricted ourselves to its implementation of the Buchberger and Gröbner walk algorithms for Gröbner basis computation (last updated in *Mathematica* in 2007), which typically perform well for low-degree polynomials over the rationals. For future work, it might be interesting to compare these methods with those of other computer algebra systems, such as *Maple* or *Magma*, using implementations of Faugère's F4 or F5 algorithms on polynomial systems arising from illumination applications (see, e.g., [28]). In summary, these methods are not suitable for real-time manipulation, but they are effective when algebraic precision is required in low-degree polynomial scenarios, such as ours. The computations with hypersurfaces of degree four and higher in 4-D scenarios ly failed led to too long computation times after minor adjustments in our experiments. Furthermore, the diversion between dimensions became important in the final visualizations of the inner points of shadows. While in 3-D, we could cover the regions with a satisfying number of points, our solution in 4-D was defeated by time to solve a system of equations and inequalities in four variables. We dropped this case because filling the 3-dimensional volumes would not clarify our visualizations in any way.

The question of perceiving four- and more-dimensional spaces is very challenging. We only added one more "perspective" open for further investigation. One of the possibilities is to study the properties of 3-surfaces through their projections. Since we developed our method on implicit surfaces, it is convenient for mathematical visualization. Second, we can pursue more "natural" illumination details, including 4-dimensional light intensity, specularity, reflections, etc. In our cases, polynomial systems that created tangent cones and occluding contours contained first-degree polynomials. Hence, there might be possibilities for improving the algorithms tailored to our situation and consequently reducing the computation time.

Although our exploration focuses on seemingly abstract higher-dimensional objects, it remains closely connected to the study of natural phenomena and real-world applications. Such problems typically emerge when systems involve multiple variables. For example, the visual aspects of higher-dimensional surfaces in kinematics are emphasized in [12, 27]. In physics, projections of 3- and 4-dimensional surfaces and their topological features are employed to visualize and gain intuition about the phase spaces of dynamical systems [14, 15, 32]. A natural way to extend the dimension is to introduce time as a geometric property. This approach is demonstrated in [6], where numerical simulations yield visualizations of moving 3-D geometries. Likewise, interpreting a 4-D scene through illumination may aid in motion planning with time-dependent obstacles, where the objective is to design feasible paths [36]. In many such scenarios, a crucial first step is to identify at least one valid solution, which can then be refined numerically. The existence of connecting paths and the connectivity of workspace components are of utmost importance—and, in principle, accessible to visual inspection. Last but not least, illuminated four-dimensional scenes might be used to study, understand, and train four-dimensional spatial ability (if possible).

5 Conclusion

This paper focuses on a four-dimensional visualization based on implicit representations of hypersurfaces. We have described a general method to find shadow boundaries in an arbitrary dimension and applied it in a three- and four-dimensional space. Furthermore, we have designed a system of polynomial equations to construct occluding contours of hypersurfaces in a 4-D perspective. The method was presented on a composed 3-D scene and three 4-D cases with gradual complexity. Our experimental journey was extensively commented on throughout the article.

Acknowledgment

We thank the community at mathematica.stackexchange.com for support with minor technical details in *Mathematica*. We also thank Daniel Lichtblau from *Wolfram Research* for his remarks leading to a significant acceleration of our computations in *Mathematica* and an update of the Dixon resultant implementation during our research.

Appendices

The attachments are available at GitHub repository:
`https://github.com/mbzamboj/4-D-shadows/tree/50fde9cb27eef46849a67fb82b6c992c0b03b491/v2`

Attachment 1

- Equations: attachment01-3D-bakery.rtf
- Video: `https://youtu.be/tKqZn7tzAQE`

Attachment 2

- Equations: attachment02-4D-hyperquadrics.rtf
- Code: code02-4D-HyperQuadrics-v2.nb
- Video: `https://youtu.be/01kYwSblPEY`

Attachment 3

- Equations: attachment03-4D-hypermoon.rtf
- Code: code03-4D-HyperMoon-v2.nb

Attachment 4

- Equations: attachment04-4D-hyperring.rtf
- Code: code04-4D-HyperRing-v2.nb
- Video: `https://youtu.be/6ZdwJ-P18Gw`

Attachment 5

Table 1 (Computation times: 4-D Scene: “Full HyperMoon Between HyperMountains” (Subsection 3.2.4))

Table 2 (Computation times: 4-D Scene: “HyperRing” (Subsection 3.2.5))

Notation:
deg – degrees of polynomials
WM-GB – Wolfram Mathematica implementation of the function GroebnerBasis with the attributes:

> LO: default lexicographic monomial order
> MO (BEO): MonomialOrder → EliminationOrder, Method → ”Buchberger”
> MO (EE): MonomialOrder → EliminationOrder, Method → {”GroebnerWalk”, ”EarlyElimination→True”}
> MO (GWEE): MonomialOrder → EliminationOrder, Method → {”GroebnerWalk”, ”EarlyElimination→True”}

WM-Dix – function DixonResultant in Wolfram Mathematica
Fer-Dix-EDF – Dixon resultant implementation in Fermat
T – terminated after 5 hours or more
F – failed

Table 1: Computation times: 4-D Scene: “Full HyperMoon Between HyperMountains” (Subsection 3.2.4)

object	deg	WM-GB LO	WM-GB MO	WM-Dix	Fer-Dix-EDF
occ. cont. $\mathcal{S}^\nu, \mathcal{P}^\nu$	3, 6	0.03s	0.01s (EO)	0.14s	—
terminators $c^\nu_{\mathcal{S}}, c^\nu_{\mathcal{P}}$	2, 6	0.26s	0.01s (EO)	0.19s	—
tang. hypcon. τ_S, τ_P	2, 6	0.1s	0.03s (GWEE)	F	—
shadow $\mathcal{S} \to \mathcal{P}$	6	25s	0.03s (GWEE)	0.33s	—
shadow $\mathcal{P} \to \mathcal{P}$	18	T	F	11s	26s
shadow $\mathcal{P} \to \mathcal{S}$	14	T	0.24s (BEO)	15966s	79s

Table 2: Computation times: 4-D Scene: "HyperRing" (Subsection 3.2.5)

object	deg	WM-GB LO	WM-GB MO	WM-Dix	Fer-Dix-EDF
occ. cont. $\mathcal{S}^{\nu}$	8	1.2s	0.02s (EO)	0.27s	—
terminator $c^{\nu}_{\mathcal{S}}$	8	516s	0.06s (EO)	0.40s	—
tang. hypcon. $\mathcal{T}$	8	2701s	0.20s (EO)	75s	3.08s
shadow $\mathcal{S} \to \mathcal{P}$	8	0.06s	0.05s (BEO)	8.6s	—

References

[1] T. Banchoff: *Beyond the Third Dimension: Geometry, Computer Graphics, and Higher Dimensions.* Scientific American Library; W. H. Freeman and Company, New York, 1990.

[2] D. Banks: *Interactive Manipulation and Display of Two-Dimensional Surfaces in Four-Dimensional Space.* In *Proc. Symp. Interactive 3D Graphics*, Part F1296, 197–207. 1992. doi: 10.1145/147156.147205.

[3] A. L. Bordignon, L. Sá, H. Lopes, S. Pesco, and L. H. de Figueiredo: *Point-based rendering of implicit surfaces in* $\mathbb{R}^4$. Comput. Graph. **37**(7), 873–884, 2013. doi: 10.1016/j.cag.2013.06.005.

[4] L. Buse, D. Cox, and C. D'Andrea: *Implicitization of Surfaces in* $\mathbb{P}^3$ *in the Presence of Base Points.* J. Algebra Appl. **2**(2), 189–214, 2003. doi: 10.1142/S0219498803000489.

[5] B. Bydžovský: *Úvod do algebraické geometrie.* Prometheus, Praha, 1948.

[6] P. C. Caplan: *Tessellation and interactive visualization of four-dimensional spacetime geometries.* Comput.-Aided Des. **178**, 103792, 2025. doi: 10.1016/j.cad.2024.103792.

[7] M. Cavallo: *Higher dimensional graphics: conceiving worlds in four spatial dimensions and beyond.* Comput. Graph. Forum **40**(2), 51–63, 2021. doi: 10.1111/cgf.142614.

[8] A. Chu, C.-W. Fu, A. J. Hanson, and P. A. Heng: *GL4D: A GPU-based architecture for interactive 4D visualization.* IEEE Trans. Vis. Comput. Graph. **15**(6), 1587–1594, 2009. doi: 10.1109/TVCG.2009.147.

[9] D. A. Cox, J. Little, and D. O'Shea: *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra.* Undergraduate Texts in Mathematics. Springer, Cham, 5 ed., 2025. doi: 10.1007/978-3-031-91841-4.

[10] A. J. Hanson and P. A. Heng: *Visualizing the Fourth Dimension Using Geometry and Light.* In *Proc. 2nd Conf. on Visualization '91*, 321–328. IEEE, San Diego, 1991. doi: 10.1109/VISUAL.1991.175821.

[11] C. M. Hoffmann and J. Zhou: *Some techniques for visualizing surfaces in four-dimensional space.* Comput.-Aided Des. **23**(1), 83–91, 1991. doi: 10.1016/0010-4485(91)90083-9.

[12] M. Husty: *On some surfaces in kinematics.* J. Geom. Graph. **16**(1), 47–58, 2012.

[13] D. Kapur and Y. N. Lakshman: *Elimination methods: An introduction.* In B. R. Donald, D. Kapur, and J. L. Mundy, eds., *Symbolic and Numerical Computation for Artificial Intelligence*, 45–89. Academic Press, 1992.

[14] M. Katsanikas and P. A. Patsis: *The structure of invariant tori in a 3D galactic potential.* Int. J. Bifurcation Chaos **21**(2), 467–496, 2011. doi: 10.1142/S0218127411028520.

[15] M. Katsanikas, P. A. Patsis, and A. D. Pinotsis: *Chains of rotational tori and filamentary structures close to high multiplicity periodic orbits in a 3D galactic potential.* Int. J. Bifurcation Chaos **21**(8), 2331–2342, 2011. doi: 10.1142/S0218127411029823.

[16] N. Khan: *Silhouette-based 2D–3D pose estimation using implicit algebraic surfaces.* Master's thesis, Saarland University, 2007.

[17] N. Khan, B. Rosenhahn, R. H. Lewis, and J. Weickert: *Silhouette-based 2D–3D Pose Estimation Using Algebraic Surfaces*, 2014. https://www.researchgate.net/publication/255455984_Silhouette-based_2D-3D_Pose_Estimation_Using_Algebraic_Surfaces. Accessed November 12, 2025.

[18] R. H. Lewis: *Example of Dixon-EDF*, 2008. http://home.bway.net/lewis/dixon. Accessed November 12, 2025.

[19] R. H. Lewis: *Fermat computer algebra system.* http://home.bway.net/lewis/, 2008. http://home.bway.net/lewis/. Accessed November 12, 2025.

[20] R. H. Lewis: *Heuristics to accelerate the Dixon resultant.* Math. Comput. Simul. **77**(4), 400–407, 2008. doi: 10.1016/j.matcom.2007.04.007.

[21] R. H. Lewis: *Resultants, implicit parameterizations, and intersections of surfaces.* In J. H. Davenport, M. Kauers, G. Labahn, and J. Urban, eds., *Mathematical Software – ICMS 2018*, 310–318. Springer, Cham, 2018.

[22] Q. Li, S. Zhang, and X. Ye: *Algebraic algorithms for computing intersections between torus and natural quadrics.* Comput.-Aided Des. Appl. **1**(1–4), 459–467, 2004. doi: 10.1080/16864360.2004.10738288.

[23] D. Lichtblau: *DixonResultant, Wolfram Function Repository, version 1.1.0*, 2023. https://resources.wolframcloud.com/FunctionRepository/resources/DixonResultant/. Accessed November 12, 2025.

[24] T. Liu: *Geometric, kinematic and radiometric aspects of image-based measurements.* In *22nd AIAA Aerodynamic Measurement Technology and Ground Testing Conference.* AIAA, Reston, Virginia, 2002. doi: 10.2514/6.2002-3239.

[25] T. Miwa, Y. Sakai, and S. Hashimoto: *Four-dimensional viewing direction control by principal vanishing points operation and its application to four-dimensional fly-through experience.* In *Proc. 25th Australas. Comput.-Human Interaction Conf.: Augmentation, Application, Innovation, Collaboration*, 95–104. ACM, New York, 2013. doi: 10.1145/2541016.2541029.

[26] A. M. Noll: *A computer technique for displaying n-dimensional hyperobjects.* Commun. ACM **10**(8), 469–473, 1967. doi: 10.1145/363534.363544.

[27] F. Pernkopf and M. L. Husty: *Workspace analysis of Stewart–Gough-type parallel manipulators.* Proc. Inst. Mech. Eng., Part C: J. Mech. Eng. Sci. **220**(7), 1019–1032, 2006. doi: 10.1243/09544062JMES194.

[28] S. Poslavsky: *Computer algebra with Rings library.* J. Phys.: Conf. Ser. **1525**(1), 012020, 2020. doi: 10.1088/1742-6596/1525/1/012020.

[29] X. Qin, D. Wu, L. Tang, and Z. Ji: *Complexity of constructing Dixon resultant matrix.* Int. J. Comput. Math. **94**(10), 2074–2088, 2017. doi: 10.1080/00207160.2016.1276572.

[30] J. Řada and M. Zamboj: *3-sphere in a 4-perspective.* In *Proc. moNGeometrija 2021*, 52–61. Planeta Print, Belgrade, 2021.

[31] J. Řada and M. Zamboj: *Four-dimensional visual exploration of the complex number plane.* In L.-Y. Cheng, ed., *ICGG 2022 – Proc. 20th Int. Conf. on Geometry and Graphics*, 138–149. Springer, Cham, 2023. doi: 10.1007/978-3-031-13588-0_12.

[32] M. Richter, S. Lange, A. Bäcker, and R. Ketzmerick: *Visualization and comparison of classical structures and quantum states of four-dimensional maps.* Phys. Rev. E **89**(2), 022902, 2014. doi: 10.1103/PhysRevE.89.022902.

[33] T. W. Sederberg, D. C. Anderson, and R. N. Goldman: *Implicit representation of parametric curves and surfaces.* Comput. Vis. Graph. Image Process. **28**(1), 72–84, 1984. doi: 10.1016/0734-189X(84)90140-3.

[34] A. Strzebonski: *Cylindrical algebraic decomposition, MathWorld—A Wolfram Web Resource*, 2023. https://mathworld.wolfram.com/CylindricalAlgebraicDecomposition.html. Accessed November 12, 2025.

[35] Wolfram Research: *GroebnerBasis, Wolfram Language function*, 1991. https://reference.wolfram.com/language/ref/GroebnerBasis.html. Accessed November 12, 2025.

[36] E. K. Xidias, P. N. Azariadis, and N. A. Aspragathos: *Path planning of holonomic and non-holonomic robots using bump-surfaces.* Comput.-Aided Des. Appl. **5**(1–4), 497–507, 2008. doi: 10.3722/cadaps.2008.497-507.

[37] S. Zachariáš and D. Velichová: *Projection from 4D to 3D.* J. Geom. Graph. **4**(1), 55–69, 2000.

[38] H. Zhang and A. J. Hanson: *Shadow-driven 4D haptic visualization.* IEEE Trans. Vis. Comput. Graph. **13**(6), 1688–1695, 2007. doi: 10.1109/TVCG.2007.70593.

[39] J. Zhou: *Visualization of four dimensional space and its applications.* Ph.D. thesis, Purdue University, 1991.

Received December 12, 2025; final form January 1, 2026.