Ray tracing multiple Minkowski sums: a CSG approach [1]

Jos B.T.M. Roerdink
Institute for Mathematics and Computing Science, University of Groningen,
P.O. Box 800, 9700 AV Groningen, The Netherlands
Tel. +31-50-3633931; Fax +31-50-3633800; Email: roe@cs.rug.nl
May 29, 1996

Abstract

The repertoire of set operators available in Constructive Solid Geometry (CSG) may be extended by the Minkowski addition and subtraction operators. In this way the problem of visualizing 3D solids defined as a Minkowski addition or subtraction of elementary objects can be solved by using CSG in combination with ray tracing. The method consists in transforming a Minkowski term with multiple operands into a union of simple objects which can be handled by a standard ray tracer. Applications of the method are found in solid modeling and shape description.

Keywords: Visualization of Minkowski operations, ray tracing, Constructive Solid Geometry

I. Introduction

In Constructive Solid Geometry (CSG), composite solid objects are formed by applying the elementary set operations union, intersection and set difference to a collection of simple solids, such as spheres, blocks, cylinders, etc. [19]. Visualization of these objects is possible by incorporating the CSG operations in the ray tracing technique, which is an established method in computer graphics to visualize 3D objects by simulating the physical processes of ray propagation, reflection and transmission [3,6].

The purpose of this paper is to show how another set of set operators, the Minkowski addition and subtraction operators [7,10,14], can be used as a tool in Constructive Solid Geometry which allows a more compact description of composite objects than is possible when using standard CSG operators. In addition, it makes possible the visualization of 3D solid objects (called 'Minkowski objects' from now on) defined as a Minkowski addition or subtraction of one or more elementary objects. Even when the elementary objects are very simple the composite objects can become very complicated higher degree surfaces without an explicit functional or parametric representation. A CSG representation is then a most useful tool to visualize these composites. In special cases the Minkowski addition reduces to the *sweep representation* in solid modeling [3,6], more in particular to the *translational sweep* defined by a 2D area swept

[1] Report CSR 9601, Department of Computing Science, University of Groningen, May 1996. Postscript version obtainable at ftp.cs.rug.nl/pub/cs-reports/CSR-9601.ps.gz

along a linear path normal to its plane, or to the *generalized cylinder* when the path is curved [22]. (In mathematics, surfaces which can be obtained by translating a curve along another curve are called *translation surfaces* [21]). Also, Minkowski operations can be implemented by a convolution of the characteristic functions of the sets, followed by thresholding [11]. In this sense, the translation surfaces may be considered as a special case of 'convolution surfaces', see e.g. [1].

Using a basic set of elementary shapes, we derive a decomposition of a multiple Minkowski sum of any number of objects, chosen from the basic set, into a union — called 'CSG object' from now on — of standard primitives used in ray tracing. These can then subsequently be visualized by using standard CSG combined with ray tracing. The transformation process involves the derivation and application of a number of decomposition rules, whose number depends on the size of the basic set. It is shown for the case of Minkowski addition, that the number of transformation rules needed is at most $2^K - 1$, where K is the number of elements of the basic set. For a particular choice of the basic set the decomposition process is carried out in detail below, and by using several simplifications the number of transformation rules is further reduced. Some examples of 3D visualizations of Minkowski sums are shown.

We mention a number of earlier approaches on the use of Minkowski operations for solid modeling. Kaul and Rossignac [9] study so-called PIP's (Parametric Interpolating Polyhedra) defined as binary Minkowski sums of the form $(1-t)A \oplus tB,\ t \in [0,1]$, with A and B polyhedra. In this way deforming polyhedra can be constructed by interpolation. Our method is different in that we do not confine ourselves to polyhedra, and consider an arbitrary number of Minkowski terms.

Rossignac and Requicha [18] use Minkowski sums of the form $(A \ominus B) \oplus B$ with B a ball[1] to achieve constant radius rounding of a solid A.

Menon et al. [12,13] use so-called 'ray-representations' in order to compute the Minkowski operations by brute force calculation using a specially designed parallel architecture.

Parry-Barwick and Bowyer [15] use a hierarchical approach which works in a higher dimensional space leading to a very time-consuming algorithm (order of hours in 2D, days in 3D).

In contrast, the present method uses an analytical decomposition of a Minkowski composite into primitives which can be simply visualized by standard CSG techniques. Thus our method is limited to special classes of objects for which the decompositions have been derived, but on the other hand it does not require special hardware and is much more efficient than the hierarchical approach.

In shape description and solid modeling the use of Minkowski operations has already proved to be fruitful, see e.g. Ghosh [4]. Other applications occur in morphological image processing, where the basic image transforms — dilations, erosions, openings, closings — are based upon the Minkowski operations [5,20]. Visualization of the results of such operations on 3D images might be a useful research tool.

The material of this paper has been presented in preliminary form in [16,17]. Here a much more concise and simplified derivation of the main results and the required transformation formulas is presented.

II. Preliminaries

In this section we first present the basics of ray tracing and Constructive Solid Geometry, followed by the definition of the Minkowski operations.

[1] In mathematical morphology [10,20] this operation is known as an *opening* of A with structuring element B.

II.A. Ray tracing

Ray tracing is a technique for image synthesis: creating 2-D pictures of a 3-D world. The first step in synthesizing such an image is a projection of the scene on the screen. The location of the projection point in the 3-D world is referred to as the *eye position*. The color of a pixel on the screen is determined by all light rays — emitted by one or more light sources — that strike that pixel and the average color of all these rays is assigned to it.

For efficiency reasons, one uses *backward ray tracing*: for every pixel on the screen, we follow back the light ray from the eye through the given pixel, and see where it first hits an object. The intersection point thus found in principle determines the color of the pixel. From this intersection point, a line (called a 'shadow ray') is traced towards every light source in the scene. When such a line intersects an object in front of the light, the point lies in the shadow of this object, unless the object is (partially) transparent, and the rays of that light source play no role in the color of the pixel. Otherwise the amount of light (of each color component) that is reflected is calculated and averaged with the colors of the rays coming from other light sources. To simulate multiple reflections or transmissions the ray tracing technique can be implemented recursively. Finally we include an ambient term, to account for diffuse light arriving via indirect paths.

II.B. Constructive Solid Geometry

In order to compute the intersection point of a ray and an object, a mathematical description of the object is needed. In CSG, simple primitives (like polygons) are combined by means of set operators. A composite object can be represented as a binary tree where each node contains a set operation and has two child nodes, which may also be composite solids. The leaves of this tree are primitive objects such as quadrics or polygons. Evaluation of set operations usually reduces to classification whether a point p lies on the *inside* or *outside* of the composite object. The rules for classification are given in Table 1.

TABLE *1: Classification of points for the CSG operations.*

operator	left	right	composite
Union	L	R	$L \vee R$
Intersection	L	R	$L \wedge R$
Difference	L	R	$L \wedge (\neg R)$

Using this classification, intersection points of composite objects are formed by merging intervals formed by the intersection points of the individual objects [19], cf. Fig. 1. It is to be noted that for surfaces with singularities a more elaborate method is required [6], but for ray tracing the above classification method usually leads to acceptable results.

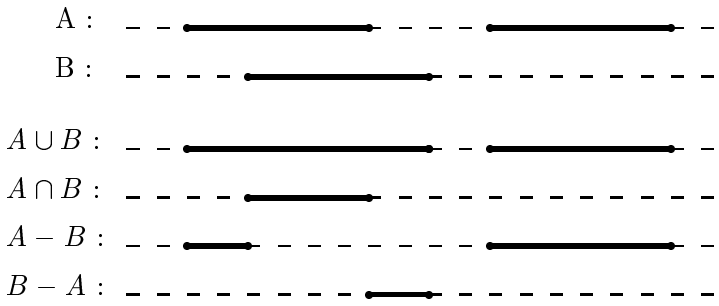

FIGURE *1: Combining lists of intersection intervals*

II.C. Minkowski operators

Minkowski addition and subtraction for subsets X, A of $\mathbb{R}^n$ are defined by

$$X \oplus A = \bigcup_{a \in A} X_a, \tag{1}$$

$$X \ominus A = \bigcap_{a \in A} X_{-a}, \tag{2}$$

where

$$X_a = \{x + a : x \in X\}$$

is the translate of X over the vector $a \in \mathbb{R}^n$, $x + y$ is the vector sum of x and y, and $-x$ the reflection of x. So the Minkowski sum of X and A is the union of translated copies of X with the translation vector running over the set A. A similar interpretation holds for the Minkowski subtraction.

The following properties of the Minkowski operators are needed below [7]. Here X, Y, A, B are arbitrary subsets of Euclidean space, $\mathsf{CH}(X)$ the convex hull of X.

$X \oplus A = A \oplus X$	commutativity	(3)
$(X \oplus A) \oplus B = X \oplus (A \oplus B)$	associativity	(4)
$(X \ominus A) \ominus B = X \ominus (A \oplus B)$		(5)
$(X \cup Y) \oplus A = (X \oplus A) \cup (Y \oplus A)$	distributivity	(6)
$(X \cap Y) \ominus A = (X \ominus A) \cap (Y \ominus A)$	distributivity	(7)
$X \ominus (A \cup B) = (X \ominus A) \cap (X \ominus B)$		(8)
$(X \oplus A)_h = X_h \oplus A$	translation invariance	(9)
$\mathsf{CH}(X) \oplus \mathsf{CH}(Y) = \mathsf{CH}(X \oplus Y)$		(10)

The last equality implies that for convex sets X and Y, $X \oplus Y$ is convex as well, and also that the following implication holds:

$$X = \mathsf{CH}(X'), Y = \mathsf{CH}(Y') \implies X \oplus Y = \mathsf{CH}(X' \oplus Y') \tag{11}$$

III. Transforming Minkowski objects into ray tracer primitives

In this section a description is given of the process needed for transforming 3D Minkowski objects into unions of simpler objects, which can be visualized via standard CSG operations combined with ray tracing. First some terminology is introduced.

III.A. Minkowski primitives

We assume a given basic set $\{A_1, A_2, \ldots, A_K\}$ of primitive shapes, referred to as *Minkowski primitives*. As a test case of the method a very small basic set will be used below (Sect. IV), consisting of the following elementary geometric objects: `xline`, `yline`, `zline`, `xdisc`, `ydisc`, `zdisc`, `sphere`. Here an `xline` is a line segment in the x-direction, and an `xdisc` is a disc with thickness zero in the plane through the origin with normal in the x-direction, etc.; finally, `sphere` is a solid sphere. All these Minkowski primitives are convex. To apply the technique of this paper in practice one probably would want to use a much larger set of primitives. However, the basic approach remains the same.

Primitives are by assumption (Minkowski-)irreducible, i.e. no decomposition of a Minkowski primitive into a finite number of shapes, whose type is different from this primitive, exists such that applying Minkowski summation to these shapes will result in this primitive. Here a decomposition of a line segment into two smaller line segments is not considered an admissible decomposition: the two operands are of the same type. In fact, the following basic assumption will be made:

Assumption : All primitives depend on a vector of parameters $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)$. *A Minkowski sum of primitives of type* ℓ *is again of type* ℓ*:*

$$A_\ell(\boldsymbol{\alpha}) \oplus A_\ell(\boldsymbol{\alpha}') = A_\ell(\boldsymbol{\alpha}''),$$

where $\boldsymbol{\alpha}''$ *depends on* $\boldsymbol{\alpha}$ *and* $\boldsymbol{\alpha}'$.

Examples of these parameters are position of solids, sizes of line segments, radii of circles, etc.

III.B. Secondary Minkowski objects

As with the ordinary boolean set operators *union*, *intersection* and *difference*, one can define terms — referred to as *Minkowski terms* — consisting of Minkowski primitives separated by Minkowski operators.

By *secondary* objects we will mean all non-empty Minkowski sums generated by choosing elements from the basic set at most once:

$$\begin{gathered} A_1, A_2, \ldots, A_K \\ A_1 \oplus A_2, A_1 \oplus A_3 \ldots, A_{K-1} \oplus A_K \\ \ldots \\ \ldots \\ A_1 \oplus A_2 \oplus A_3 \ldots \oplus A_K \end{gathered} \tag{12}$$

These sets will be denoted by $B_j(\boldsymbol{\beta}), j = 1, \ldots, 2^{K-1}$, where $\boldsymbol{\beta}$ depends on the parameter vectors of the summands $\{A_k\}$. Note that no braces have to be put around the Minkowski sums because of the associativity property (4). So the number of secondary Minkowski objects is 2^{K-1}, that is, one less than the cardinality of the power set over the set of Minkowski primitives.

Examples of secondary Minkowski objects for the basic set given above are *rectangle, block, cylinder, toroidal disc* (convex hull of a torus, a bit like a discus) and a *bidisc* (the Minkowski sum of two perpendicular flat discs), see Sect. IV.

III.C. Arbitrary Minkowski sums

Now consider a Minkowski sum of arbitrary size

$$S_N = C_1 \oplus C_2 \oplus \dots C_N \tag{13}$$

with each C_ℓ chosen from the basic set $\{A_1, A_2, \dots, A_K\}$. The following theorem will be proved.

Theorem 1: Let S_N be as in (13) with each C_ℓ chosen from the basic set. Then S_N equals one of the elements of the secondary set $\{B_j(\boldsymbol{\beta}) : j = 1, 2, \dots, 2^{K-1}\}$, cf. (12), for an appropriately chosen parameter vector $\boldsymbol{\beta}$.

Proof: Using commutativity of the Minkowski addition, we can group together all primitives of the same type, e.g. first all line segments in the x-direction, then those in the y-direction, etc. Then using the basic assumption, a Minkowski sum of primitives of the same type can be reduced to another primitive of this type with an appropriate choice of parameter values. So:

$$S_N = A_1(\boldsymbol{\gamma}_1) \oplus A_2(\boldsymbol{\gamma}_2) \oplus \dots A_K(\boldsymbol{\gamma}_K), \tag{14}$$

where a term A_ℓ has to be omitted if no primitives of type ℓ occur in (13). But the above expression for S_N precisely equals one of the secondary Minkowski primitives $B_j(\boldsymbol{\beta})$ where $\boldsymbol{\beta}$ depends on the parameters vectors $\boldsymbol{\gamma}_k, k = 1, 2, \dots, K$. ∎

Corollary 2: To study arbitrary Minkowski sums of the form (13) it is sufficient to restrict ourselves to the set of secondary objects.

III.D. Reduction of secondary Minkowski objects to ray tracer primitives

Our goal is to decompose 3D Minkowski objects into unions of elementary objects, referred to as *ray tracer primitives*, which are simple enough to be handled by a standard ray tracer. So the task is to find a set of objects (e.g., planes, quadrics, quartics)

$$R_1(\boldsymbol{\rho}_1), R_2(\boldsymbol{\rho}_2), \dots, R_M(\boldsymbol{\rho}_M) \tag{15}$$

depending on parameter vectors $\{\boldsymbol{\rho}_\ell\}$, which generate all secondary solids by *unions*:

$$B_j(\boldsymbol{\beta}) = \bigcup_{k=1}^{M} c_{jk} R_k(\boldsymbol{\rho}_k), \tag{16}$$

where $c_{jk} = 0$ or $c_{jk} = 1$. The union B_j can then be visualized using CSG in combination with ray tracing, as explained in Sect. II.B. Of course, for efficiency reasons the number M should be chosen as small as possible.

The reason of using a *union* for the representation is simplicity. Other possibilities are conceivable involving operators like set intersection or set difference, possibly combined with union. But our goal here is to demonstrate the possibility of a representation in terms of simple ray tracer primitives, not to derive the most general one.

IV. Reduction for the simple basic set

IV.A. Representation of Minkowski primitives

The process of reducing the secondary Minkowski objects B_j to ray tracer primitives R_k will be carried out for the very simple set of Minkowski primitives $\{A_1, A_2, \ldots, A_7\}$ introduced above, where

$$A_1 = \texttt{xline}(x,y,z,lx) : \text{line starting at } (x,y,z) \text{ and ending at } (x+lx,y,z)$$
$$A_2 = \texttt{yline}(x,y,z,ly) : \text{line starting at } (x,y,z) \text{ and ending at } (x,y+ly,z)$$
$$A_3 = \texttt{zline}(x,y,z,lz) : \text{line starting at } (x,y,z) \text{ and ending at } (x,y,z+lz)$$
$$A_4 = \texttt{xdisc}(x,y,z,rx) : \text{disc of radius } rx \text{ in the } YZ\text{-plane centered at } (x,y,z)$$
$$A_5 = \texttt{ydisc}(x,y,z,ry) : \text{disc of radius } ry \text{ in the } XZ\text{-plane centered at } (x,y,z)$$
$$A_6 = \texttt{zdisc}(x,y,z,rz) : \text{disc of radius } rz \text{ in the } XY\text{-plane centered at } (x,y,z)$$
$$A_7 = \texttt{sphere}(x,y,z,r) : \text{sphere with center } (x,y,z) \text{ and radius } r$$

Notice that no point primitive is needed, since all primitives have position parameters (x,y,z) to set their location. Also, in order to satisfy the basic assumption which says that addition of two Minkowski primitives of the same type should yield another primitive of the same type, we cannot use 'open' primitives like the circle, but must use a disc (convex hull of the circle) instead. Also, the sphere is considered a solid object (i.e., with interior), as is implicit in CSG anyhow.

Since this basic set contains 7 elements, there are $2^7 - 1 = 127$ secondary objects for which reduction formulas must be derived. The derivation of these formulas is obtained by geometric analysis which has to be done by hand. It would be very useful, especially if a larger basic set would be required, to automate the derivation of these reduction formulas.

In what follows, first a few examples of reduction formulas are presented to give the reader some idea of their nature. These are just a few specimens of an extended collection of such formulas, which however is too large to be presented in detail. This is followed by a formulation of the main result, whose proof can be found in Appendix B.

IV.B. Sketch of the reduction process

First we mention a few Minkowski sums for which no reduction is necessary because the objects are themselves standard ray tracer primitives. Consider a simple form, such as the sum of three orthogonal line segments, which is a block:

$$\texttt{block}(x,y,z,lx,ly,lz) = \texttt{xline}(x,0,0,lx) \oplus \texttt{yline}(0,y,0,ly) \oplus \texttt{zline}(0,0,z,lz).$$

If $lz = 0$ this reduces to a `zrectangle`, which is a rectangle with normal in the z-direction.

Another simple example is the addition of a line segment and a disc, resulting in a cylinder. For example,

$$\texttt{zline}(x,y,z,lz) \oplus \texttt{zdisc}(0,0,0,rz) = \texttt{zcylinder}(x,y,z,rz,lz),$$

where $\texttt{zcylinder}(x,y,z,rz,lz)$ is a cylinder with axis in the positive z-direction of length lz, and with base curve a circle of radius rz centered at (x,y,z).

To derive a formula for the the Minkowski sum of a disc and a solid sphere, which we call a *discus*, it is necessary to consider first the sum of circle and sphere, leading to a 3D object whose surface is a torus (a *quartic*, i.e., fourth degree surface):

$$\texttt{xtorus}(x, y, z, rr, rs) = \texttt{xcircle}(x, y, z, rr) \oplus \texttt{sphere}(0, 0, 0, rs),$$

where $\texttt{xcircle}(x, y, z, rr)$ is a circle of radius rr in the YZ-plane centered at (x, y, z) (`ycircle` and `zcircle` are defined similarly). A picture is shown in Fig. 2. Since `xdisc` is the convex hull of `xcircle`, the sum of the disc and the solid sphere is obtained from the torus by taking the convex hull (cf. (11)), which implies that a cylinder has to be added to the torus to fill up the 'hole', see Fig. 3.

The Minkowski sum of two orthogonal circles, one with radius r_z in the XY-plane and one with radius r_y in the XZ-plane leads to an object which we call a `xbicircle`, which is also quartic surface (with self-intersections), shown in Fig. 4. Its equation is [16]:

$$(x^2 - y^2 + z^2 + r_z^2 - r_y^2)^2 = 4x^2(r_z^2 - y^2).$$

Notice that a bicircle, being a quartic surface, is a ray tracer primitive. The Minkowski sum of two orthogonal discs is the convex hull of the corresponding bicircle, and is called a bidisc, cf. Fig. 5. For example, $\texttt{xbidisc}(x, y, z, ry, rz) = \texttt{ydisc}(x, y, z, ry) \oplus \texttt{zdisc}(0, 0, 0, rz)$.

FIGURE *2: Visualization of the Minkowski sum of a circle and a sphere (a torus).*

FIGURE *3: Visualization of the Minkowski sum of a flat disc and a sphere (a discus).*

FIGURE *4: Visualization of the Minkowski sum of two perpendicular circles (a bicircle).*

FIGURE *5: Visualization of the Minkowski sum of two perpendicular flat discs (a bidisc).*

Note that disci and bidiscs are solid, convex shapes. In order to produce ray tracer primitives for them, we have to take a union of one or two cylinders with a torus and a bicircle, respectively. To be precise, the following reductions have to be performed:

$$\begin{aligned}
\texttt{xdiscus}(x,y,z,rr,rs) \quad &\longmapsto \texttt{xtorus}\,(x,y,z,rr,rs) && \cup \\
&\quad\ \texttt{xcylinder}\,(x-rs,y,z,rr,2\,rs) \\
\texttt{xbidisc}(x,y,z,ry,rz) \quad &\longmapsto \texttt{xbicircle}\,(x,y,z,ry,rz) && \cup \\
&\quad\ \texttt{ycylinder}\,(x,y-rz,z,ry,2\,rz) && \cup \\
&\quad\ \texttt{zcylinder}\,(x,y,z-ry,rz,2\,ry)
\end{aligned}$$

with similar expressions for the rotated versions `ydiscus`, `ybidisc`, etc.

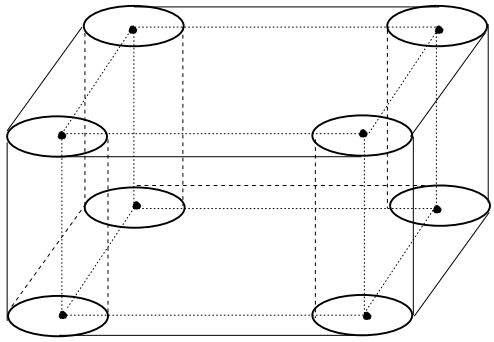

FIGURE *6: Sketch of a Minkowski sum of a* `block` *and a* `zdisc`*, from which the required ray tracer primitives can be inferred.*

Remark 3: Since $\texttt{xdiscus}(x,y,z,rr,rs) = \texttt{xdisc}(x,y,z,rr) \oplus \texttt{sphere}(0,0,0,rs)$ is a union of two ray tracer objects (torus and cylinder), it can itself be considered a ray tracer object. But there are advantages in working with torus and cylinder separately, see Sect. IV.D.

As a last example, consider the Minkowski sum of a `block` and a `zdisc`, cf. Fig. 6. By inspection we conclude that this sum can be represented by adding four cylinders and four blocks to the original block. The number of required blocks can be reduced to two by allowing some overlap between them. This leads to the following formula:

$$\begin{aligned}
&\texttt{block}\,(x1,y1,z1,lx,ly,lz) \oplus \texttt{zdisc}\,(x2,y2,z2,r) \quad \longmapsto \\
&\texttt{block}(x1+x2-r,y1+y2+r,z1+z2,lx+r,ly-r,lz) && \cup \\
&\texttt{block}(x1+x2+r,y1+y2-r,z1+z2,lx-r,ly+r,lz) && \cup \\
&\texttt{zcylinder}(x1+x2,y1+y2,z1+z2,r,lz) && \cup \\
&\texttt{zcylinder}(x1+x2+lx,y1+y2,z1+z2,r,lz) && \cup \\
&\texttt{zcylinder}(x1+x2,y1+y2+ly,z1+z2,r,lz) && \cup \\
&\texttt{zcylinder}(x1+x2+lx,y1+y2+ly,z1+z2,r,lz)
\end{aligned}$$

Remark 4: It may be surprising to the reader that the Minkowski addition of simple curves like circles produces such complicated surfaces. However, even in the plane, Minkowski addition may lead to complex results. As an example, consider the addition of a parabola and a disc of radius r, which can be computed in two steps. First, calculate the *parallel curve* of the parabola, i.e., the locus of points along the inner and outer normal at a distance r from the parabola. This

parallel curve contains two curve segments (one 'inside' the parabola and one 'outside'), which are two branches of a single algebraic curve of degree six [8, p. 179–181], cf. Fig. 7; see also Appendix A. Note that this curve is self-intersecting. The second step in computing the Minkowski sum consists in taking the union of all line segments from points on the parabola along the normals to the corresponding points on the inner and outer segments of the parallel curve (the hatched region in the right picture of Fig. 7).

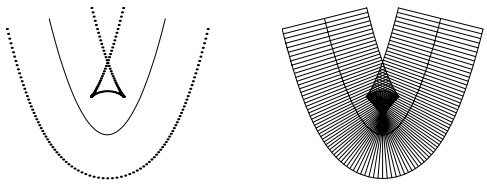

FIGURE *7: Left: parabola (solid line) and its parallel curve (dotted lines). Right: the corresponding Minkowski sum of parabola and disc (hatched region).*

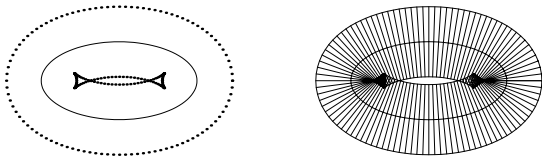

FIGURE *8: Left: ellipse (solid line) and its parallel curve (dotted lines). Right: the corresponding Minkowski sum of ellipse and disc (hatched region).*

Another example is provided by the *toroids*, which are the parallel curves of an ellipse, which can also be defined as the outline curves (silhouettes) of a torus when projected parallel to a plane [2]. Toroids are algebraic curves of degree eight. In Fig. 8 an example is given of a parallel curve of an ellipse, together with the corresponding Minkowski sum.

IV.C. Rotational Variants

Many ray tracer primitives exist in three variants (x, y, and z) that differ only in orientation. Therefore it is sufficient to derive a reduction formula for one particular variant (e.g. the x-variant), and apply an inverse transformation to the set of produced ray tracer primitives to get the correct rotational forms. This is based on the following property of the Minkowski addition:

$$A \oplus B \;=\; \mathcal{R}^{-1}(\; \mathcal{R}(A) \oplus \mathcal{R}(B)\;) \tag{17}$$

for every linear transformation $\mathcal{R}$ of $\mathbb{R}^d$. Of course one has to take into account that the reference location of primitives may be altered as a result of rotation.

IV.D. Optimizations

According to Remark 3, an equivalent representation arises if the tori are replaced by disci. A similar remark applies to replacing bicircles by bidiscs. However, it is preferable to use unions of ray tracer primitives in order to avoid the generation of 'redundant primitives'. For example, the Minkowski addition of an `xline` and an `xdiscus` results in the union of two parallel `xdisci` and an `xcylinder` in between. When the two disci are converted to ray tracer primitives, each of them will become the union of a torus and an cylinder. Both newly created cylinders share the same longitudinal axis and the cylindrical space in between lies completely interior to the embracing cylinder we already had. It is preferable to replace those two cylinders by an extended version that contains them both plus the space in between, thus reducing the number of cylinders to be traced by the ray tracer (time saving).

As another example, consider the Minkowski sum

$$\texttt{block} \oplus \texttt{zdisc} \oplus \texttt{sphere}$$

This can be reduced to ray tracer primitives in two ways. First we could reduce $\texttt{block} \oplus \texttt{zdisc}$, which yields 2 blocks and 4 cylinders, cf. Sect. IV.B. Then compute the Minkowski sum with the sphere: the sum of block and sphere yields 8 spheres, 3 blocks and 12 cylinders; the sum of cylinder and sphere produces 2 tori and 2 cylinders. All in all, we need 16 spheres, 6 blocks, 8 tori and 32 cylinders when proceeding this way. However, a more efficient representation results if we perform the reduction process in two steps by putting the line primitives in one group and the circle and sphere primitives in another group. In the present example this means that we first reduce $\texttt{zdisc} \oplus \texttt{sphere}$, yielding a `zdiscus` which is the union of a `ztorus` and a `zcylinder`. When taking the Minkowski sum with the block, we get an object which can be represented by the following collection of primitives: 8 ztori (one for each corner of the block); 4 zcylinders corresponding to translating the ztori along the Z-axis; 4 zcylinders corresponding to translating the zcylinders along the Z-axis; 4 xcylinders and 4 ycylinders corresponding to translating the ztori along the X- or Y-axis; and finally, 3 blocks. In all, this requires 27 primitives, which is substantially less than the 62 needed in the first reduction process.

IV.E. The main result

The complete set of reduction formulas for the basic set used is discussed in Appendix B. The result is given as a theorem.

Theorem 5: For the set of Minkowski primitives given in Sect. IV.A, all secondary Minkowski objects can be written as unions of the following 16 ray tracer primitives:

$R_1 = \texttt{block}(\mathbf{x}, lx, ly, lz)$: block with diagonal corners $\mathbf{x}$ and $\mathbf{x} + (lx, ly, lz)$

$R_2 = \texttt{xcylinder}(\mathbf{x}, r, lx)$: cylinder with axis $\texttt{xline}(\mathbf{x}, lx)$ and radius r

$R_3 = \texttt{ycylinder}(\mathbf{x}, r, ly)$: cylinder with axis $\texttt{yline}(\mathbf{x}, ly)$ and radius r

$R_4 = \texttt{zcylinder}(\mathbf{x}, r, lz)$: cylinder with axis $\texttt{zline}(\mathbf{x}, lz)$ and radius r

$R_5 = \texttt{sphere}(\mathbf{x}, r)$: sphere with center $\mathbf{x}$ and radius r

$R_6 = \texttt{xtorus}(\mathbf{x}, rr, rs)$: $\texttt{xcircle}(\mathbf{x}, rr) \oplus \texttt{sphere}(\mathbf{0}, rs)$

$R_7 = \texttt{ytorus}(\mathbf{x}, rr, rs)$: $\texttt{ycircle}(\mathbf{x}, rr) \oplus \texttt{sphere}(\mathbf{0}, rs)$

$R_8 = \texttt{ztorus}(\mathbf{x}, rr, rs)$: $\texttt{zcircle}(\mathbf{x}, rr) \oplus \texttt{sphere}(\mathbf{0}, rs)$

$R_9 = \texttt{xbicircle}(\mathbf{x}, ry, rz)$: $\texttt{ycircle}(\mathbf{x}, ry) \oplus \texttt{zcircle}(\mathbf{0}, rz)$

$R_{10} = \texttt{ybicircle}(\mathbf{x}, rx, rz)$: $\texttt{xcircle}(\mathbf{x}, rx) \oplus \texttt{zcircle}(\mathbf{0}, rz)$

$R_{11} = \texttt{zbicircle}(\mathbf{x}, rx, ry)$: $\texttt{xcircle}(\mathbf{x}, rx) \oplus \texttt{ycircle}(\mathbf{0}, ry)$

$R_{12} = \texttt{xbicirclesphere}(\mathbf{x}, ry, rz, rs)$: $\texttt{xbicircle}(\mathbf{x}, ry, rz) \oplus \texttt{sphere}(\mathbf{0}, rs)$

$R_{13} = \texttt{ybicirclesphere}(\mathbf{x}, rx, rz, rs)$: $\texttt{ybicircle}(\mathbf{x}, rx, rz) \oplus \texttt{sphere}(\mathbf{0}, rs)$

$R_{14} = \texttt{zbicirclesphere}(\mathbf{x}, rx, ry, rs)$: $\texttt{zbicircle}(\mathbf{x}, rx, ry) \oplus \texttt{sphere}(\mathbf{0}, rs)$

$R_{15} = \texttt{tricircle}(\mathbf{x}, rx, ry, rz)$: $\texttt{xcircle}(\mathbf{x}, rx) \oplus \texttt{ycircle}(\mathbf{0}, ry) \oplus \texttt{zcircle}(\mathbf{0}, rz)$

$R_{16} = \texttt{tricirclesphere}(\mathbf{x}, rx, ry, rz, rs)$: $\texttt{tricircle}(\mathbf{x}, rx, ry, rz) \oplus \texttt{sphere}(\mathbf{0}, rs)$

Here the shorthand notations $\mathbf{x} = (x, y, z)$ and $\mathbf{0} = (0, 0, 0)$ have been used. So the set of required ray tracer primitives consists of sixteen objects: a block, a sphere, 3 cylinders, 3 tori, 3 bicircles, and 5 objects (R_{12}–R_{16}) which are surfaces of degree higher than four. For such surfaces, no analytical formulas for the intersection points with a ray exist. To handle these higher-degree surfaces numerical root-finding techniques are required.

Remark 6: The line or disc primitives, which are infinitely thin objects, can be visualized by choosing a box or cylinder, respectively, with appropriate parameter values (block sizes, cylinder height) close to zero. Note also that sphere, tori, bicircles and tricircle can be obtained as special cases of a `tricirclesphere` by letting radii of the sphere and/or one or more circles go to zero in the definition of the tricirclesphere. However, for ray tracing purposes it is more efficient to work with these objects as separate ray tracer primitives.

IV.F. Complexity of the CSG representation

It is of interest to study the question how many ray tracer primitives can arise during the decomposition process. To answer this question consider the most complicated secondary Minkowski object, which is the sum of all the Minkowski primitives:

$$B^* = A_1' \oplus A_4 \oplus A_5 \oplus A_6 \oplus A_7, \tag{18}$$

where A_1' is the block produced by the Minkowski sum of three orthogonal line segments. That is, B^* is obtained by putting a `tricirclesphere` at each corner of the block, and taking the convex hull.

It is shown in Appendix B that B^* can be obtained as the union of the following ray tracer primitives: 3 blocks, 8 xcylinders, 8 ycylinders, 8 zcylinders and 8 tricirclespheres. Here special care has been taken to avoid redundant ray tracer primitives as much as possible.

Hence it may be concluded that the maximal number of ray tracer primitives which can arise during the decomposition of any secondary Minkowski object can be reduced to at most $3 + 24 + 8 = 35$. Of course for special values of the parameters the actual number needed may be (considerably) less.

IV.G. Minkowski Subtraction

Minkowski subtraction lacks, in contrast with Minkowski addition, some very useful properties like associativity, commutativity and certain forms of distributivity. This makes Minkowski subtraction a lot more difficult to implement, because conjuring with parentheses, permutations and distributions is not allowed. The only case which is comparable to the case of Minkowski sums is when we have an expression of the form

$$D_N = (\ldots((C_0 \ominus C_1) \ominus C_2)\ldots) \ominus C_N. \tag{19}$$

Using (5) this can be transformed to

$$D_N = C_0 \ominus (C_1 \oplus C_2 \ldots \oplus C_N) \tag{20}$$

Inserting the union representation for Minkowski sums and the property expressed by Eq. 8, this becomes

$$D_N = C_0 \ominus \Big(\bigcup_j R_j\Big) = \bigcap_j C_0 \ominus R_j. \tag{21}$$

Assuming that $C_0 \ominus R_j$ can be written as an intersection of R_k's we have a representation of D_N as a multiple intersection which is comparable to the union representation for Minkowski sums. Note however that for the expression

$$C_0 \ominus (C_1 \ominus (C_2 \ominus (\ldots \ominus C_N)\ldots))$$

this technique does not work.

When both Minkowski addition and subtraction are allowed within a single expression, many more secondary Minkowski objects arise than is the case when just Minkowski sums are allowed. For all combinations of Minkowski primitives reduction formulas must be derived. It is not even certain that the number of secondary objects, and therefore also the required number of reduction formulas, is finite in all cases.

IV.H. Implementation

A *preprocessor* routine has been developed which takes as input a file with a scene description, containing positions of light sources and objects, descriptions of the composite objects in terms of unions, intersections, or Minkowski sums of elementary objects and object properties (reflection and transmission coefficients, smoothness parameters, texture). A Minkowski sum may appear in the description file in the following form:

```
$ xline(5) <+> zline(5) <+> yline(100) <+> ydisc(2) $.          (22)
```

This is in fact the Minkowski addition of a block and a disc. The `$`-signs delimit a Minkowski sum in the input file. The symbol `<+>` represents the $\oplus$-operator. For all objects the origin is their reference location which is therefore omitted in the parameter list. During preprocessing these Minkowski expressions are translated into proper ray tracer code using the appropriate reduction formulas. The result is written to an output file, which is then itself used as input file for an ordinary ray tracer. Preprocessors for the ray tracers 'DKBTrace' and 'POV-Ray' were developed. The pictures in this paper have been produced with POV-Ray, an excellent ray tracer which offers many advanced features.

Some examples of scenes with objects modeled by Minkowski sums are given in Fig. 9 and Fig. 10.

FIGURE *9: Table on a wooden floor. Legs and table top modeled using Minkowski sums with discus and sphere.*

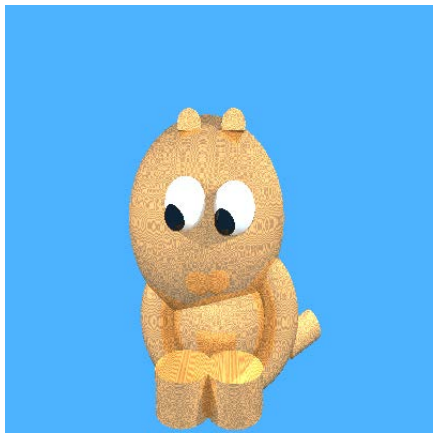

FIGURE *10: A toy animal. The arms and belly are composed of the union of a bidisc and a sphere.*

V. SUMMARY

The visualization of Minkowski set operations using ray tracing techniques is not a trivial matter. In this paper it has been shown that a multiple Minkowski sum with operands chosen from a given basic set of primitives can be reduced to a *secondary* Minkowski object, where the collection of secondary objects is formed by the power set of the basic set. By a decomposition

process wherein each secondary object is in turn expressed as a union of ray tracer primitives, ray tracing through standard CSG operations enables an efficient visualization.

The possibility of such a decomposition of Minkowski sums depends essentially upon useful properties like commutativity, associativity, distributivity over *union*, translation invariance and invariance under linear transformations.

To demonstrate the feasibility of this approach, we carried out the complete reduction process for a very simple set of Minkowski primitives consisting of three orthogonal line segments, three orthogonal disks and a sphere. In the process of computing Minkowski sums of disks and sphere some complicated surfaces arise (quartic or higher degree). The equations for these have to be derived analytically.

A preprocessor routine has been developed taking a scene description with Minkowski sums as input which delivers an output file in which the Minkowski terms have been replaced by a union of ray tracer primitives. This file is then itself used as input file for an ordinary ray tracer.

When the use of Minkowski *subtraction* is permitted in Minkowski terms too, then in general the decomposition process is a lot more complicated, requiring a large number of reduction formulas and yielding a much larger variety of primitive ray tracer objects.

In the present paper only a very simple class of primitive objects was considered, the Minkowski sums of which can be decomposed into unions of ray tracer objects. This may be extended to larger classes of Minkowski primitives, as long as the operands of the Minkowski sum themselves are described in terms of explicit mathematical equations. Then one may use a direct method involving envelope conditions (as used for a number of higher degree surfaces arising in this paper, see also Appendix A) to derive the intersection equations of a Minkowski sum from the equations of the individual objects.

The advantage of our method is twofold: (i) efficiency: once the decomposition has been carried out, the ray tracing process is comparatively fast; (ii) compactness of representation: an expression like (22) takes many lines after translation into ray tracer code; also the description in terms of Minkowski sums is more direct and intuitive.

The existing alternatives are brute force calculation requiring special purpose hardware [12, 13], or the hierarchical but very time-consuming approach of [15].

References

[1] Bloomenthal, J. and Shoemake, K., "Convolution Surfaces," *Siggraph'91, Computer Graphics* , 25, no. 4, pp. 251–256, July 1991.

[2] Brieskorn, E. and Knoerrer, H., *Plane algebraic curves.* Cambridge, MA, Birkhäuser Boston, 1986.

[3] Foley, J. D., Dam, A. V. and Feiner, S. K., *Computer Graphics : Principles and Practice.* Reading, MA, Addison Wesley, 1990.

[4] Ghosh, P. K., "A mathematical model for shape description using Minkowski operators," *Comp. Vis. Graph. Im. Proc.*, 44, pp. 239–269, 1988.

[5] Giardina, C. R. and Dougherty, E. R., *Morphological Methods in Image and Signal Processing.* Englewood Cliffs, NJ, Prentice-Hall, 1988.

[6] Glassner, A. S., Ed., *An Introduction to Ray Tracing*: Academic Press, New York, 1989.

[7] Hadwiger, H., *Vorlesungen über Inhalt, Oberfläche, und Isoperimetrie.* Springer, Berlin, 1957.

[8] Hilton, H., *Plane algebraic curves.* London, Milford, 1932, 2nd ed.

[9] Kaul, A. and Rossignac, J., "Solid-Interpolating Deformations: Constructions and Animation of PIPs ," in *Proc. Eurographics'91, Vienna*. pp. 493–505, 1991.

[10] Matheron, G., *Random Sets and Integral Geometry*. New York, NY, John Wiley & Sons, 1975.

[11] Mazille, J. E., "Mathematical morphology and convolutions," *J. Microscopy*, 156, pp. 3–13, 1989.

[12] Menon, J. P., Marisa, R. J. and Zagajac, J., "More powerful solid modeling through ray representations," *IEEE Computer Graphics and Applications*, 14, no. 3, pp. 22–35, 1994.

[13] Menon, J. P. and Voelcker, H. B., "Mathematical foundations I: set theoretic properties of ray representations and Minkowski operations on solids," Technical Report CPA 91-9, Sibley School of Mechanical and Aerospace Engineering, Cornell University, Ithaca New York, April 1992.

[14] Minkowski, H., "Volumen und Oberfläche," *Math. Ann.*, 57, pp. 447–495, 1903.

[15] Parry-Barwick, S. J. and Bowyer, A., "Minkowski sums of set-theoretic models," Preprint (`http://www.bath.ac.uk/-ensab/G_mod/MD/multi_theory.html`).

[16] Roerdink, J. B. T. M. and Blaauwgeers, G. S. M., "Visualization of Minkowski operations by computer graphics techniques," in *Mathematical Morphology and its Applications to Image Processing*, Serra, J. and Soille, P., Eds. Kluwer Acad. Publ., Dordrecht, pp. 289–296, 1994.

[17] Roerdink, J. B. T. M. and Blaauwgeers, G. S. M., "Minkowski operations as a tool in Constructive Solid Geometry," in *Proc. Computing Science in the Netherlands, 27-28 november, Utrecht*. pp. 194–204, 1995.

[18] Rossignac, J. R. and Requicha, A. A. G., "Offsetting operations in solid modeling," *Computer Aided Geometric Design*, 3, pp. 129–148, 1986.

[19] Roth, S., "Ray casting for modeling solids," *Comp. Graph. Im. Proc.*, 18, pp. 109–144, 1982.

[20] Serra, J., *Image Analysis and Mathematical Morphology*. Academic Press, New York, 1982.

[21] Struik, D. J., *Lectures on classical differential geometry*. Reading, MA, Addison Wesley, 1950.

[22] Wijk, J. J. van, "On new types of solid models and their visualization with ray tracing," PhD Thesis, Technical University, Delft, 1986.

Appendix

A. Parallel curve of a parabola

The equation for the parallel curve of a parabola can be obtained as follows. Let the equation of the parabola be given in coordinates (u, v) as

$$v = a\,u^2. \tag{23}$$

The points (x, y) on the parallel curve are at distance r from the parabola, so

$$(x-u)^2 + (y-v)^2 = r^2. \tag{24}$$

To complete the description we have to apply the *envelope* condition, which says that the desired points are on the boundary of the region swept out by the moving disc. This can be done by assuming a parameterization of the parabola in terms of a parameter t, and differentiating the equations (23)-(24) with respect to t. This yields

$$v' = 2a\,u\,u' \tag{25}$$

$$2(x-u)\,u' + 2(y-v)\,v' = 0 \tag{26}$$

where the primes denote differentiation with respect to t. Eliminating v' we get

$$(x-u) + 2a\,u(y-v) = 0. \tag{27}$$

The desired equation for the parallel curve can now be obtained by eliminating u and v from (23), (24) and (27). The result is:

$$\begin{gathered} 16\,a^4\,r^6 + 8\,a^2\,r^4\,(1 - 6\,a^2\,x^2 - 4\,a\,y - 2\,a^2\,y^2) \\ +r^2\,(1 + 20\,a^2\,x^2 + 48\,a^4\,x^4 - 8\,a\,y - 8\,a^3\,x^2\,y + 8\,a^2\,y^2 + \\ 32\,a^4\,x^2\,y^2 + 32\,a^3\,y^3) = (a\,x^2 - y)^2\,(1 + 16\,a^2\,x^2 - 8\,a\,y + 16\,a^2\,y^2) \end{gathered}$$

We note that a parametric representation of this curve is much easier to obtain: just find the points (x, y) at a distance r from the parabola along the normals:

$$\begin{aligned} x &= u \pm 2\,r\,a\,u/\sqrt{4\,a\,v+1} \\ y &= v \mp r/\sqrt{4\,a\,v+1} \end{aligned}$$

where u, v define the parabola as in (23).

B. Proof of Theorem 5

Using the representation in Sect. IV.A, there are 7 Minkowski primitives. To simplify the presentation let us replace the line primitives A_1, A_2, A_3 by their Minkowski sum

$$A_1' = \texttt{block}(x, y, z, lx, ly, lz) \tag{28}$$

which is a rectangular block with diagonal corners (x, y, z) and $(x + lx, y + ly, z + lz)$. The objects A_1, A_2, A_3 can be obtained from A_1' as limiting cases by letting lx, ly and/or lz go to zero. In this way only $2^5 - 1 = 31$ secondary objects need to be considered.

So the primitives used are (we drop the prime on A_1 from now on):

$$A_1 = \texttt{block}(x, y, z, lx, ly, lz)$$
$$A_2 = \texttt{xdisc}(x, y, z, rx)$$
$$A_3 = \texttt{ydisc}(x, y, z, ry)$$
$$A_4 = \texttt{zdisc}(x, y, z, rz)$$
$$A_5 = \texttt{sphere}(x, y, z, r)$$

Terms of order one

The first 5 secondary Minkowski objects are the primitives themselves. Block and sphere are directly ray-traceable, for the discs one can use cylinders with height going to zero. This yields the first 5 primitives R_1–R_5 in Theorem 5.

Terms of order two

These are 10 in number. Let us start with $A_1 \oplus A_2$. Using again the property expressed by (11), this sum can be obtained by putting an `xdisc` at each corner of the box, and taking the convex hull. This leads to 4 cylinders and two blocks. The sum of a block with a `ydisc` or `zdisc` leads to similar expressions, cf. Fig. 6.

Next take $A_1 \oplus A_5$. That is, put a `sphere` at each corner of the box, and take the convex hull. To compute the Minkowski sum of a sphere with a line segment, we make use of the fact that a sphere can be divided in two symmetric parts by a plane whose normal is along the translation direction. The cross-section of this plane with the sphere is a disc. This implies that the Minkowski sum of the sphere with a line segment consists of two half-spheres, one at each endpoint of the line segment, with a cylinder in between. Using ray tracer primitives only, this can also be represented with some redundancy by the union of two full spheres and a cylinder. Summing up, one now finds that 12 cylinders are needed (one for each side of the block), 3 blocks (minimizing their number) and finally 8 spheres (one for each corner of the block).

The sum $A_2 \oplus A_3$ is the object `zbidisc` defined earlier, cf. Fig. 5, which is itself the union of a bicircle and two cylinders. The same is true for the rotated versions $A_2 \oplus A_4$ and $A_3 \oplus A_4$ (cf. Sect. IV.C).

The last three sums ($A_2 \oplus A_5$, $A_3 \oplus A_5$ and $A_4 \oplus A_5$) lead to the previously defined `xdiscus`, `ydiscus` and `zdiscus`, which are unions of tori and cylinders.

In conclusion, the new ray tracer primitives (apart from those required for terms of order one) are bicircles and tori, i.e., the primitives R_6–R_{11} in Theorem 5.

Terms of order three

Again there are 10 of them. Start with

$$B_{16} = A_1 \oplus A_2 \oplus A_3 = \texttt{block} \oplus \texttt{xdisc} \oplus \texttt{ydisc} = \texttt{block} \oplus \texttt{zbidisc}.$$

This means putting a `zbicircle` at each corner of the block and taking the convex hull. The convex hull of the `zbicircle` (a `zbidisc`) is the union of the `zbicircle` and two cylinders. Using the same line of argument as for the sphere, we observe that a `zbidisc` has a symmetry plane perpendicular to the z-axis whose cross-section with itself is a rectangle, whereas it is the union of a rectangle and two discs for a cross-section perpendicular to the x- or y-axis, cf. Fig. 11. This implies that the Minkowski addition of a `zbidisc` with a line segment results in rectangles, blocks, discs or cylinders, with two `zbidiscs` at the endpoints. Therefore B_{16}

can be represented by bicircles, cylinders and blocks. This also holds for the rotated versions $A_1 \oplus A_2 \oplus A_4$ and $A_1 \oplus A_3 \oplus A_4$.

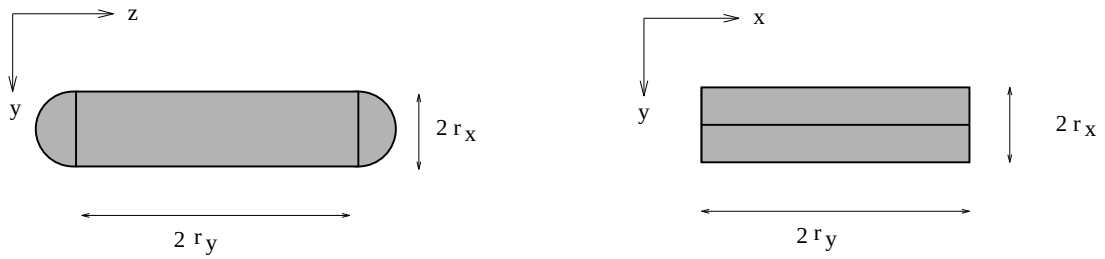

FIGURE *11: Cross section of a* `zbidisc` *with a symmetry plane perpendicular to: left: the x-axis; right: the z-axis.*

For the terms $A_1 \oplus A_2 \oplus A_5$, $A_1 \oplus A_3 \oplus A_5$ and $A_1 \oplus A_4 \oplus A_5$ we must do an addition of a block and a discus (convex hull of a torus). Since, as is easy to see, the cross-section of a discus with a symmetry plane perpendicular to the x-, y- or z- axis consists again of discs or unions of rectangles and discs, which generate blocks and cylinders under translation, these three terms can be represented by tori, cylinders and blocks.

The remaining 4 terms involve a sum of three discs, or two discs and a sphere. All these can be decomposed into a sum of three circles, or two circles and a sphere, leading to surfaces of degree higher than 4, whose equation can in principle be obtained by applying envelope conditions (compare the derivation of the Minkowski sum of a parabola and a disc in Appendix A), to which cylinders have to be added afterwards, just as in the case of discus and bidisc, cf. Sect. IV.B.

Terms of order four and five

All these terms involve higher degree surfaces. It is sufficient to consider the complete term $B_{31} = A_1' \oplus A_2 \oplus A_3 \oplus A_4 \oplus A_5$, since each of the operands can be reduced to a single point at the origin by choosing appropriate parameter values. That is, B_{31} is obtained by putting a `tricirclesphere` at each corner of the block, and taking the convex hull. First of all, we have to take the convex hull of the `tricirclesphere` itself; let us call it a `tridiscsphere`. Again it is easy to see that this object is the union of a `tricirclesphere` and an `xcylinder`, a `ycylinder` and a `zcylinder`. The two xcylinders in the two tridiscspheres on the endpoints of a side of the block parallel to the X-axis can be taken together in a single xcylinder; so 4 xcylinders are needed. The same holds for the y- and zcylinders. This yields 4 xcylinders, 4 ycylinders and 4 zcylinders. The cross-section of a `tridiscsphere` with a symmetry plane perpendicular to any of the three coordinate axes is the Minkowski sum of a rectangle with a disc. When translated along any such axis, this produces a block and 4 cylinders, of which 3 can be included in blocks. So at most 4 of such cylinders are needed per coordinate axis. This again yields 4 xcylinders, 4 ycylinders and 4 zcylinders. What remains are 8 tricirclespheres, one for each corner of the block (these reduce to bicirclespheres, bicircles, tori or spheres for appropriate values of the parameters). Using symmetry arguments, the number of blocks which is needed can be reduced to at most 3.

This concludes the proof of the theorem.