

Article

Lightweight Implicit Approximation of the Minkowski Sum of an N-Dimensional Ellipsoid and Hyperrectangle

Martijn Courteaux , Bert Ramlot , Peter Lambert  and Glenn Van Wallendael * 

IDLab, Ghent University—imec, 9052 Ghent, Belgium; martijn.courteaux@ugent.be (M.C.); bert.ramlot@ugent.be (B.R.); peter.lambert@ugent.be (P.L.)

* Correspondence: glenn.vanwallendael@ugent.be

Abstract: This work considers the Minkowski sum of an N-dimensional ellipsoid and hyperrectangle, a combination that is extremely relevant due to the usage of ellipsoid-adjacent primitives in computer graphics for work such as 3D Gaussian splatting. While parametric representations of this Minkowski sum are available, they are often difficult or too computationally intensive to work with when, for example, performing an inclusion test. For performance-critical applications, a lightweight approximation of this Minkowski sum is preferred over its exact form. To this end, we propose a fast, computationally lightweight, non-iterative algorithm that approximates the Minkowski sum through the intersection of two carefully constructed bounding boxes. Our approximation is a super-set that completely envelops the exact Minkowski sum. This approach yields an implicit representation that is defined by a logical conjunction of linear inequalities. For applications where a tight super-set of the Minkowski sum is acceptable, the proposed algorithm can substantially improve the performance of common operations such as intersection testing.

Keywords: Minkowski sum; ellipsoid; hyperrectangle; box; algorithm; approximation; Gaussian splatting; implicit representation; Mahalanobis distance; bounding boxes

MSC: 52A27



Academic Editor: Valery G. Romanovskij

Received: 11 March 2025

Revised: 11 April 2025

Accepted: 16 April 2025

Published: 18 April 2025

Citation: Courteaux, M.; Ramlot, B.; Lambert, P.; Van Wallendael, G. Lightweight Implicit Approximation of the Minkowski Sum of an N-Dimensional Ellipsoid and Hyperrectangle. *Mathematics* **2025**, *13*, 1326. <https://doi.org/10.3390/math13081326>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The Minkowski sum, a binary operation in geometry, combines two subsets of a Euclidean space to form a new set. Formally, the Minkowski sum $A \oplus B$ of two sets of vectors A and B is defined as follows:

$$A \oplus B = \{\vec{a} + \vec{b} \mid \vec{a} \in A, \vec{b} \in B\}.$$

Additionally, the Minkowski difference $A \ominus B$ of two sets of vectors A and B is defined as follows:

$$A \ominus B = A \oplus (-B) = \{\vec{a} - \vec{b} \mid \vec{a} \in A, \vec{b} \in B\}.$$

The Minkowski sum and difference play a critical role in both theoretical research and practical applications across a diverse range of fields, including robotics [1,2], energy [3–5], algebraic geometry [6], and computational biology [7], to name a few. In mathematical morphology, the Minkowski sum is better known as dilation, one of the four basic operations. Mathematical morphology has its roots in 2D image processing but has since expanded to related fields, such as computer graphics [8]. The Minkowski sum is also used

more broadly in computer graphics for tasks such as constraint enforcement for vector graphics [9] and collision detection [10]. These works typically focus on traditional representations, e.g., vector graphics [9] and meshes [10], with little attention being paid to newer representations. One class of non-traditional representations that has become popular is those based on multivariate Gaussian primitives, e.g., the Steered Mixture of Experts [11] and 2D/3D/4D Gaussian splatting [12–14]. Multivariate normal distributions are closely related to ellipsoids, as the level sets of their probability density function (or Mahalanobis distance) form concentric ellipsoids, which are uniformly scaled versions of one another. Ellipsoids have therefore become of more and more interest within computer graphics.

When an origin-centered ellipsoid is the second operand in a Minkowski difference, it becomes equivalent to a Minkowski sum due to the point symmetry of ellipsoids. Numerous methods for computing the boundary of a Minkowski sum involving ellipsoids exist, but most are parametric [15–18]. Although parametric representations provide precise solutions, they fall short in their applicability in tests that must be run extremely quickly. This is especially relevant in computer graphics applications, where we want to conduct millions to billions of tests every second [19]. To achieve such speeds, we are willing to accept a more crude approximation of the Minkowski sum, given that the approximation is guaranteed to envelop the entire Minkowski sum. Furthermore, we are especially interested in the Minkowski sum of an ellipsoid and hyperrectangle (commonly referred to as a box), as this operation naturally occurs during common computer graphics operations, such as during the frustum culling and block-based rasterization of 3D Gaussian splatting [13]. The construction of the exact Minkowski sum used in this scenario is visualized in Figure 1. An ellipsoid and a box only intersect when the origin (i.e., zero vector) lies within their Minkowski difference. Note that this is true for any pair of shapes [20]. More intuitively, taking advantage of the ellipsoid's point symmetry, the ellipsoid intersects with the box only if its center lies within the Minkowski sum of the box and its origin-centered counterpart. An example of how this overlap is detected when relying on the Minkowski sum is demonstrated in Figure 2.

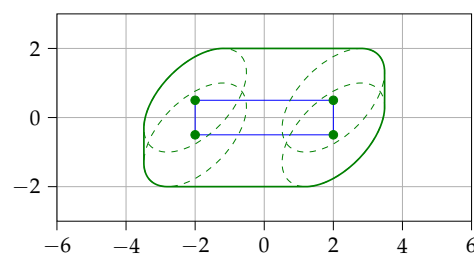


Figure 1. Illustration of the construction of the exact Minkowski sum boundary (shown in solid green) constructed for a rectangle (shown in blue) and an ellipse. For visual clarity, only ellipses positioned at the rectangle's corner points (shown as dashed green outlines) are shown as they define the boundary's extremes.

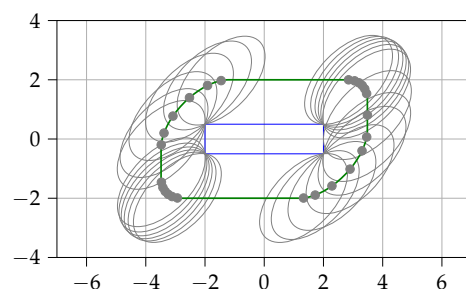


Figure 2. Illustration showing how the Minkowski sum of an ellipse and a rectangle can be used to determine their intersection. When the center point of the ellipse (shown in gray) lies within the Minkowski sum (depicted in green), it indicates that the ellipse intersects or touches the rectangle (shown in blue).

To this end, we propose a non-iterative, fast, and computationally lightweight test to determine whether an N -dimensional ellipsoid overlaps with an N -dimensional box's domain. Our method creates an implicit representation by defining the approximated sum as a logical conjunction of $2N$ linear inequalities. This approach not only guarantees a 'tightly fitted' approximation that fully contains the exact solution, but it also offers a simple-to-evaluate and computationally efficient solution. One significant practical implication of this method is its efficiency in identifying, within a specified tolerance, the multivariate normal distributions that intersect a given hyperrectangular subdomain from a large set.

2. Background and Related Work

In this section, we provide an overview of the fundamental mathematical concepts that underlie our work and review relevant existing methods for computing Minkowski sums.

The Minkowski sum, named after Hermann Minkowski, is a binary operation that combines two subsets of a Euclidean space by adding each vector in the first set to each vector in the second set. While this operation has closed-form solutions for certain combinations of shapes, some are non-trivial. Examples of cases where the Minkowski sum has straightforward solutions and for which algorithms are available include the following:

- Convex polygons [21];
- Non-convex polygons (through their decomposition into convex polygons) [21,22];
- Three-dimensional triangle meshes [21];
- Algebraic planar curves [23].

Existing methods for computing Minkowski sums often involve complex mathematical procedures or algorithms. Some methods yield results in a parametric form [15–17,23–25], while others result in a set of vertices with connectivity information [21,22]. While these solutions are mathematically precise, their form or algorithmic complexity limits their practical application. In general, implicit formulations are preferred for tasks such as intersection testing.

This work focuses on the Minkowski sum of a box and an ellipsoid. To our knowledge, there is no prior work directly addressing this specific combination that has led to a practically efficient and implicit representation. A key insight involves leveraging the properties of zonohedra. A box is a zonohedron and can be expressed as the Minkowski sum of its N orthogonal generator line segments, where N is the dimensionality of the space. Let the i -th generator line segment, centered at the origin, be aligned with the i -th axis and have a half-length of s_i . We can represent this line segment as a 'collapsed' ellipsoid E_i whose covariance matrix R_i has exactly one non-zero eigenvalue:

$$R_i = \vec{e}_i \vec{e}_i^T s_i^2, \quad i = 1, \dots, N, \quad (1)$$

where \vec{e}_i is the i -th standard basis vector. This transforms the problem of computing the Minkowski sum of an ellipsoid E (with covariance matrix R) and a box into computing the Minkowski sum of $m = N + 1$ ellipsoids: $E \oplus E_1 \oplus \dots \oplus E_N$.

The boundary of the Minkowski sum of m ellipsoids E_i (defined by the covariance matrices R_i) can be parameterized exactly [15] as follows:

$$\vec{S}(\vec{n}) = \sum_{i=1}^m \frac{R_i \vec{n}}{\sqrt{\vec{n}^T R_i \vec{n}}}, \quad \forall \vec{n} : \|\vec{n}\| = 1, \quad (2)$$

where parameter \vec{n} represents the normal vector to the boundary surface pointing outward [15]. However, this parametric form is generally impractical for efficiently determining point containment (i.e., whether a point lies inside the sum). Furthermore, the presence of collapsed ellipsoids (with rank-deficient R_i) requires careful handling, as $\vec{n}^T R_i \vec{n}$ can be zero for normals that are orthogonal to the i -th standard basis vector.

Given the limitations of the exact parametric form, approximation techniques are often employed, particularly for constructing an outer ellipsoidal bound. A fundamental tool for constructing the outer bound of a Minkowski sum of convex shapes is the support function h_K of the convex body K :

$$h_K(\vec{u}) = \sup_{\vec{x} \in K} \vec{x} \cdot \vec{u}, \quad \vec{u} \in \mathbb{R}^N. \quad (3)$$

We restrict ourselves to normalized vectors, i.e., $\vec{u} \in \mathbb{S}^{N-1}$. As a result, the support function describes the signed distance from the origin to the supporting hyperplane of K with outward normal \vec{u} , i.e., the farthest extent of K in direction \vec{u} . It possesses the additivity property for Minkowski sums [26,27]:

$$h_{K_1 \oplus K_2} = h_{K_1} + h_{K_2}. \quad (4)$$

For an ellipsoid E defined by a positive definite covariance matrix R , the support function is [15] as follows:

$$h_E(\vec{u}) = \sqrt{\vec{u}^T R \vec{u}}. \quad (5)$$

As for the origin-centered collapsed ellipsoid E_i with the covariance matrix from Equation (1), we interpret it as an origin-centered line segment extending from $-s_i \vec{e}_i$ to $s_i \vec{e}_i$ to allow us to apply the known support function for a line segment [28]:

$$h_{E_i}(\vec{u}) = \max\{(-s_i \vec{e}_i) \cdot \vec{u}, (s_i \vec{e}_i) \cdot \vec{u}\} = |s_i u_i|. \quad (6)$$

Leveraging the additivity of support functions, a commonly studied family of ellipsoids guaranteed to contain the Minkowski sum of a collection of ellipsoids with covariance matrices R_1, \dots, R_m can be constructed. An ellipsoid within this family $E(\gamma)$ is parameterized by a vector $\gamma = (\gamma_1, \dots, \gamma_m) \in \mathbb{R}_{>0}^m$, with its covariance matrix $R(\gamma)$ given by [29–31]:

$$R(\gamma) = \sum_{i=1}^m \gamma_i R_i \quad (7)$$

where the parameters must satisfy

$$\sum_{i=1}^m \frac{1}{\gamma_i} = 1. \quad (8)$$

Note that this constraint implies that $\gamma_i \geq 1$. Durie et al. [31] show that this ensures that the support function of the resulting ellipsoid $h_{E(\gamma)}(\vec{u}) = \sqrt{\vec{u}^T R(\gamma) \vec{u}}$ bounds the sum of the individual support functions $\sum h_{E_i}(\vec{u})$, thus guaranteeing containment: $E_1 \oplus \dots \oplus E_m \subseteq E(\gamma)$ [31].

The primary objective is often to find the tightest bound within this family, specifically the Minimum-Volume Outer Ellipsoid (MVOE), which is obtained by minimizing the determinant of $R(\gamma)$:

$$\gamma_{\text{MVOE}} = \arg \min_{\gamma} \det(R(\gamma)) \quad \text{subject to (8)}. \quad (9)$$

While this is a convex optimization problem [31], γ_{MVOE} generally lacks a closed-form solution. It can be found using numerical optimization algorithms, such as Sequential Least Squares Programming (SLSQP) [32], but this can be computationally demanding. For the specific case of $m = 2$, a fast fixed-point algorithm exists [33]. Halder proposed using this fixed-point algorithm as the basis for a heuristic for $m > 2$ by recursively applying this pairwise optimal algorithm to achieve an outer ellipsoid bound (γ_{halder}) [33]. While

this fixed-point algorithm simplifies for collapsed ellipsoids, the recursive nature of the heuristic means that it remains iterative.

An alternative approach involves minimizing the trace of the covariance matrix, $\text{tr}(R(\gamma))$, instead of its determinant. This alternative optimization problem does have a known closed-form solution for the $m = 2$ case [30]:

$$\gamma_1 = 1 + \sqrt{\frac{\text{tr}(R_2)}{\text{tr}(R_1)}} \quad \text{and} \quad \gamma_2 = 1 + \sqrt{\frac{\text{tr}(R_1)}{\text{tr}(R_2)}} \quad (10)$$

Inspired by this, a computationally tractable heuristic choice for the general m case is as follows [15]:

$$\gamma_{\text{chirikjian},i} = \frac{\sum_{j=1}^m \sqrt{\text{tr}(R_j)}}{\sqrt{\text{tr}(R_i)}}, \quad i \in \{1, \dots, m\} \quad (11)$$

This provides a non-iterative method for obtaining the γ of an outer ellipsoid, potentially sacrificing tightness compared to the MVOE in exchange for computational speed.

3. Proposed Approximation Algorithm

This section details our proposed algorithm for efficiently approximating the Minkowski sum of an N -dimensional ellipsoid and an axis-aligned box, $\Omega_M = \Omega_{\text{ellipsoid}} \oplus \Omega_{\text{box}}$. The core strategy of this algorithm is to construct two distinct oriented bounding boxes, Ω_{bbA} and Ω_{bbB} , which are each guaranteed to contain the Minkowski sum Ω_M . The intersection of these two boxes, $\Omega_{\sim M} = \Omega_{\text{bbA}} \cap \Omega_{\text{bbB}}$, yields a tighter approximation of Ω_M than either box alone while retaining a simple implicit representation suitable for efficient queries.

We begin by formally defining the input shapes, which are centered at the origin. We assume, without a loss of generality, that the box is axis-aligned. If the initial box were arbitrarily oriented using an invertible linear transformation $B \in \mathbb{R}^{N \times N}$, the problem could be transformed to the axis-aligned case by applying an inverse transformation to the ellipsoid, resulting in a modified covariance matrix $R' = B^{-1}R(B^{-1})^T$.

The axis-aligned box is defined by its per-dimension half-sizes, $\vec{s}_{\text{box}} \in \mathbb{R}_{>0}^N$. The region Ω_{box} can then be defined as

$$\Omega_{\text{box}} = \left\{ \vec{x} \in \mathbb{R}^N : \bigwedge_{i=1}^N (|x_i| \leq s_{\text{box},i}) \right\}. \quad (12)$$

The ellipsoid, defined by its covariance matrix R , occupies the region $\Omega_{\text{ellipsoid}}$:

$$\Omega_{\text{ellipsoid}} = \left\{ \vec{x} \in \mathbb{R}^N : \vec{x}^T R^{-1} \vec{x} \leq 1 \right\}. \quad (13)$$

Our approximation Ω_M relies on constructing two tight bounding boxes, Ω_{bbA} and Ω_{bbB} . The key tool for determining the minimal size of a bounding box that encloses Ω_M in any given direction is the support function of the Minkowski sum h_{Ω_M} . Using the additivity property (Equation (4)), the support function of the ellipsoid (Equation (5)), and the support function of the box (Equation (6)), we derive the following:

$$\begin{aligned} h_{\Omega_M}(\vec{u}) &= h_{\Omega_{\text{ellipsoid}}}(\vec{u}) + h_{\Omega_{\text{box}}}(\vec{u}) \\ &= h_{\Omega_{\text{ellipsoid}}}(\vec{u}) + \sum_{i=1}^N h_{E_i}(\vec{u}) \\ &= \sqrt{\vec{u}^T R \vec{u}} + \sum_{i=1}^N |s_{\text{box},i} u_i| \\ &= \sqrt{\vec{u}^T R \vec{u}} + \vec{s}_{\text{box}} \cdot |\vec{u}|. \end{aligned} \quad (14)$$

where $|\vec{u}|$ denotes the element-wise absolute value of \vec{u} .

An oriented bounding box Ω_{bb} , aligned with an orthonormal basis $\{\vec{b}_1, \dots, \vec{b}_N\}$ and tightly enclosing Ω_M , can be constructed using the support function h_{Ω_M} . Recall that $h_{\Omega_M}(\vec{u})$ gives us the maximum projection of Ω_M onto the \vec{u} direction. For the box to tightly contain Ω_M , its boundary planes orthogonal to \vec{b}_i must align with the maximum extents of Ω_M in the $\pm\vec{b}_i$ directions, which are given by $h_{\Omega_M}(\vec{b}_i)$ and $h_{\Omega_M}(-\vec{b}_i)$. Thus, the smallest bounding box Ω_{bb} is given by

$$\begin{aligned}\Omega_{\text{bb}} &= \left\{ \vec{x} \in \mathbb{R}^N : \bigwedge_{i=1}^N \left((\vec{b}_i \cdot \vec{x} \leq h_{\Omega_M}(\vec{b}_i)) \wedge ((-\vec{b}_i) \cdot \vec{x} \leq h_{\Omega_M}(-\vec{b}_i)) \right) \right\} \\ &= \left\{ \vec{x} \in \mathbb{R}^N : \bigwedge_{i=1}^N \left(|\vec{b}_i \cdot \vec{x}| \leq h_{\Omega_M}(\vec{b}_i) \right) \right\}\end{aligned}\quad (15)$$

since $h_{\Omega_M}(\vec{u}) = h_{\Omega_M}(-\vec{u})$ due to the symmetry of the box and ellipsoid about the origin. We choose two specific orientations for the basis vectors \vec{b}_i to construct Ω_{bbA} and Ω_{bbB} , aiming to capture different geometric features of Ω_M .

3.1. The Axis-Aligned Bounding Box (bbA)

The first bounding box, Ω_{bbA} , is aligned with the standard coordinate axes, meaning that its basis vectors are $\vec{b}_i = \vec{e}_i$. Its half-sizes, $s_{\text{bbA},i}$, are determined by evaluating the support function (Equation (14)) along these axes:

$$s_{\text{bbA},i} = h_{\Omega_M}(\vec{e}_i) = \sqrt{R_{ii}} + s_{\text{box},i}, \quad i = 1, \dots, N. \quad (16)$$

Intuitively, the half-size along the i -th axis is the sum of the ellipsoid's projected extent onto that axis ($\sqrt{R_{ii}}$) and the box's half-size along that axis ($s_{\text{box},i}$). The region Ω_{bbA} is then implicitly defined according to Equation (15):

$$\Omega_{\text{bbA}} = \left\{ \vec{x} \in \mathbb{R}^N : \bigwedge_{i=1}^N \left(|x_i| \leq s_{\text{bbA},i} \right) \right\}. \quad (17)$$

This construction is illustrated for $N = 2$ in Figure 3.

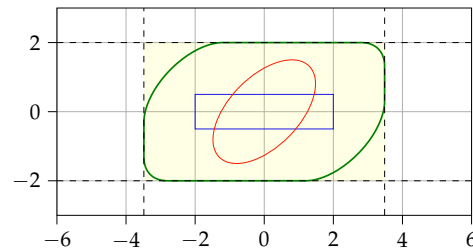


Figure 3. Example construction of the 2D axis-aligned bounding box Ω_{bbA} .

3.2. The Covariance-Oriented Bounding Box (bbB)

The second bounding box, Ω_{bbB} , is aligned with the principal axes of the ellipsoid. These axes are defined by the normalized eigenvectors $\{\vec{v}_1, \dots, \vec{v}_N\}$ of the ellipsoid's covariance matrix R and correspond to the eigenvalues $\{\lambda_1, \dots, \lambda_N\}$. These are obtained via an eigendecomposition: $R = V\Lambda V^T$, where $V = [\vec{v}_1 | \dots | \vec{v}_N]$ and $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_N)$. (We note that in applications like 3D Gaussian splatting [13], R is often stored in terms of its eigensystem, avoiding the need for explicit decomposition).

The half-sizes of Ω_{bbB} , denoted as $s_{\text{bbB},i}$, are determined by evaluating the support function (Equation (14)) along the eigenvectors $\vec{b}_i = \vec{v}_i$:

$$s_{\text{bbB},i} = h_{\Omega_M}(\vec{v}_i) = \sqrt{\lambda_i} + s_{\text{box}} \cdot |\vec{v}_i|, \quad i = 1, \dots, N, \quad (18)$$

for which we used the definition of an eigenvector: $R \vec{v}_i = \lambda_i \vec{v}_i$. Intuitively, this is the sum of the ellipsoid's semi-axis length along \vec{v}_i and the maximal extent of the box projected onto the direction \vec{v}_i . The region Ω_{bbB} is implicitly defined by using the basis V according to Equation (15):

$$\Omega_{\text{bbB}} = \left\{ \vec{x} \in \mathbb{R}^N : \bigwedge_{i=1}^N (|(V^T \vec{x})_i| \leq s_{\text{bbB},i}) \right\}. \quad (19)$$

The construction of this bounding box is visualized in Figure 4.

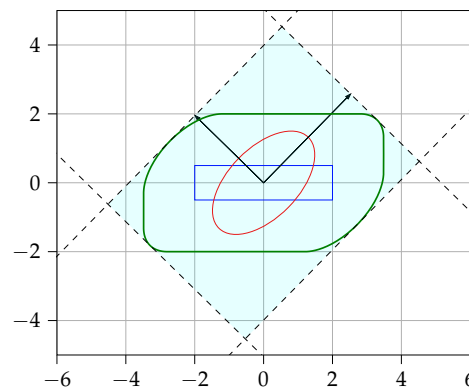


Figure 4. Example construction of the 2D eigenvector-aligned bounding box Ω_{bbB} .

3.3. Intersecting the Bounding Boxes

To compute the approximated Minkowski sum $\Omega_{\sim M}$, we intersect the two bounding boxes:

$$\begin{aligned} \Omega_{\sim M} &= \Omega_{\text{bbA}} \cap \Omega_{\text{bbB}} \\ &= \left\{ \vec{x} \in \mathbb{R}^N : \bigwedge_{i=1}^N (|x_i| \leq s_{\text{bbA},i}) \wedge (|(V^T \vec{x})_i| \leq s_{\text{bbB},i}) \right\}. \end{aligned} \quad (20)$$

The resulting intersection is illustrated in Figure 5.

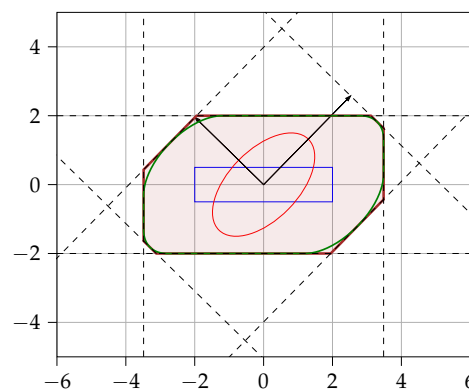


Figure 5. Example construction of the 2D intersection of Ω_{bbA} and Ω_{bbB} .

Testing whether a point lies within the approximated Minkowski sum region, as shown in Equation (20), is straightforward and computationally efficient. Specifically, the required operations are as follows:

- 1 eigendecomposition (*);
- $2N$ square root computations (*);
- 1 matrix–vector multiplication;
- $2N$ absolute value computations;
- $2N$ comparisons.

Operations marked with (*) can be precomputed when the box and ellipsoid are static. Additionally, depending on the use case, the eigenvectors and corresponding eigenvalues might already be known, making the eigendecomposition unnecessary.

In NumPy [34], this logic can be implemented as in Listing 1:

Listing 1. Reference implementation of our proposed algorithm in Python 3 using NumPy 2.

```

1 import numpy as np
2
3 # Input definitions
4 cov = np.array([\dots]) # NxN matrix
5 half_block_size = np.array([\dots]) # N-vector
6 x = np.array([\dots]) # point to test, N-vector
7
8 # Compute bounding box sizes (precomputable):
9 evals, evecs = np.linalg.eigh(cov)
10 bbA = np.sqrt(np.diag(cov)) + half_block_size
11 bbB = np.sqrt(evals) + np.sum(half_block_size[:,None] * np.abs(evecs), axis=0)
12
13 # Testing the point:
14 in_bbA = np.all(np.abs(x) <= bbA)
15 in_bbB = np.all(np.abs(np.dot(evecs.T, x)) <= bbB)
16 in_minkowski_approx = np.logical_and(in_bbA, in_bbB)

```

The above implementation can be trivially extended to efficiently test an array of points.

4. Results and Discussion

This section uses a Monte Carlo estimation to evaluate the performance of the proposed Minkowski sum approximation, $\Omega_{\sim M} = \Omega_{bbA} \cap \Omega_{bbB}$, and compare it with alternative approaches. It also discusses the approximation's characteristics and identifies areas for future investigation.

4.1. Evaluation Methodology

To quantify the tightness of the approximation, we use the Volume Ratio (VR), which is defined as $VR(A) = Vol(A) / Vol(\Omega_M)$, where A is a fully encompassing approximation of the exact Minkowski sum Ω_M . A perfect approximation has a VR of 1.

A direct comparison with prior work is challenging due to the lack of specific methods for approximating the box–ellipsoid Minkowski sum. To address this, we establish comparisons by viewing the box as a sum of degenerate ellipsoids (as discussed in Section 2) and by comparing it against approximations from the literature on multi-ellipsoid summation. Here, we note that the computational complexity of the point containment tests for a single box is roughly comparable to that of a single ellipsoid. Therefore, we report VR values not only for the proposed intersection $\Omega_{\sim M}$ but also for its components, Ω_{bbA} and Ω_{bbB} .

The evaluation was performed via Monte Carlo simulations with the dimensions $N = 2$ to $N = 6$. In each trial, the lengths of the ellipsoid semi-axes ($\sqrt{\lambda_i}$) and the half-sizes of the box ($s_{box,i}$) were independently sampled from $\mathcal{U}(0.2, 1.8)$, simulating a balanced scenario. The ellipsoids were also given a random orientation. All VRs were computed using 100,000 trials.

4.2. Results and Analysis

The estimated average VRs are presented in Table 1. The results clearly show that the proposed intersection $\Omega_{\sim M}$ yields the lowest average VR across the tested dimensionalities in this balanced scenario, outperforming all discussed outer ellipsoid bounds (even without considering Ω_{bbB}), validating this intersection approach.

Table 1. Monte Carlo estimates (100,000 samples per N) of the average Volume Ratio (VR), with 99.9% confidence intervals, for $N = 2$ to 6 dimensions. The proposed intersection method ($\Omega_{\sim M}$) and its components (Ω_{bbA} and Ω_{bbB}) are compared against several of the outer ellipsoid approximations discussed in Section 2. Lower VR values indicate tighter approximations.

Method		Iterative Creation?	Number of Dimensions (N)				
			2	3	4	5	6
Ellipsoid bounds $E(\gamma)$	γ_{MVOE}	Yes (Slow)	1.231 ± 0.001	1.597 ± 0.002	2.134 ± 0.003	2.913 ± 0.005	4.034 ± 0.008
	γ_{halder}	Yes (Fast)	1.255 ± 0.001	1.684 ± 0.002	2.350 ± 0.005	3.382 ± 0.009	4.972 ± 0.015
	$\gamma_{\text{chirikjian}}$	No	1.256 ± 0.001	1.656 ± 0.002	2.250 ± 0.003	3.126 ± 0.006	4.413 ± 0.009
Ours	Ω_{bbA}	No	1.094 ± 0.001	1.261 ± 0.001	1.513 ± 0.002	1.872 ± 0.003	2.371 ± 0.005
	Ω_{bbB}	No	1.390 ± 0.002	2.362 ± 0.004	4.669 ± 0.010	10.40 ± 0.025	25.54 ± 0.068
	$\Omega_{\sim M}$	No	1.028 ± 0.000	1.160 ± 0.001	1.388 ± 0.001	1.725 ± 0.002	2.205 ± 0.004

While generally effective, the performance of $\Omega_{\sim M}$ exhibits sensitivity to the specific geometric configuration considered. Firstly, if the ellipsoid's principal axes align closely with the coordinate axes, Ω_{bbA} and Ω_{bbB} become similar, reducing the benefit gained from their intersection. Secondly, their relative scales matter. If the size of the ellipsoid is negligible compared to that of the box, Ω_{bbA} is a near-perfect bound by itself. Conversely, if the size of the box is negligible, the true sum is nearly an ellipsoid, and single-ellipsoid approximations, such as $E(\gamma_{\text{halder}})$ and $E(\gamma_{\text{chirikjian}})$, will offer a tighter fit than our box-based intersection.

Furthermore, Table 1 reveals that the VR tends to increase with the number of dimensions N . We hypothesize that this trend arises from two main factors. First, as N increases, a larger portion of the Minkowski sum's volume lies near the edges, where approximations introduce errors. Second, the boundary of the Minkowski sum involves interactions between the ellipsoid and all k -faces of the N -dimensional box. Since the number of these faces grows exponentially with N , this significantly increases geometric complexity. Our approximation only accounts for the box's vertices (0-faces) and facets ($(N - 1)$ -faces), intentionally simplifying the structure to maintain computational efficiency. This simplification becomes increasingly limiting in higher dimensions, where intermediate k -faces contribute more substantially to the true boundary.

4.3. Future Directions

The observed performance characteristics of our method, and particularly its degradation in higher dimensions, suggest avenues for future work. A promising direction would be to enhance the approximation by intersecting more than two bounding volumes. Investigating systematic methods for selecting additional oriented bounding boxes, potentially targeting different k -faces or even incorporating other shapes like ellipsoids into the intersection, could yield tighter bounds. The challenge lies in balancing the improved accuracy with the added computational cost of constructing and querying a more complex intersection region.

5. Conclusions

This paper presents a novel, non-iterative, computationally efficient algorithm for approximating the Minkowski sum of an N -dimensional ellipsoid and box. The proposed method defines the approximate sum as the logical conjunction of the conditions defining two carefully constructed bounding boxes: one axis-aligned and the other oriented along the ellipsoid's eigenvectors. These bounding boxes are designed to each fully contain the exact Minkowski sum, ensuring that the approximation fully envelopes the exact solution. By considering the overlap between the bounding boxes, the algorithm provides

an approximation that is closely fitted to the exact solution. Its computational simplicity and non-iterative nature make it suitable for performance-critical applications, such as the efficient culling of multivariate normal distributions in computer graphics.

Author Contributions: Conceptualization, M.C., P.L. and G.V.W.; methodology, M.C.; software, M.C.; validation, M.C. and B.R.; formal analysis, M.C., B.R., P.L. and G.V.W.; investigation, M.C., B.R., P.L. and G.V.W.; resources, M.C., P.L. and G.V.W.; data curation, M.C.; writing—original draft preparation, M.C. and B.R.; writing—review and editing, B.R., P.L. and G.V.W.; visualization, M.C.; supervision, P.L. and G.V.W.; project administration, P.L. and G.V.W.; funding acquisition, M.C., P.L. and G.V.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded in part by the Research Foundation–Flanders (FWO) under Grant 1SA7919N, IDLab (Ghent University–imec), Flanders Innovation & Entrepreneurship (VLAIO), and the European Union.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Ruan, S.; Poblete, K.L.; Wu, H.; Ma, Q.; Chirikjian, G.S. Efficient Path Planning in Narrow Passages for Robots With Ellipsoidal Components. *IEEE Trans. Robot.* **2023**, *39*, 110–127. [\[CrossRef\]](#)
2. Liao, Y.; Wu, Y.; Zhao, S.; Zhang, D. Unmanned Aerial Vehicle Obstacle Avoidance Based Custom Elliptic Domain. *Drones* **2024**, *8*, 397. [\[CrossRef\]](#)
3. Wu, Z.; Zou, Y.; Zheng, F.; Liang, N. Research on Optimal Scheduling Strategy of Microgrid Considering Electric Vehicle Access. *Symmetry* **2023**, *15*, 1993. [\[CrossRef\]](#)
4. Wang, X.; Wang, X.; Liu, Y.; Xiao, C.; Zhao, R.; Yang, Y.; Liu, Z. A Sustainability Improvement Strategy of Interconnected Data Centers Based on Dispatching Potential of Electric Vehicle Charging Stations. *Sustainability* **2022**, *14*, 6814. [\[CrossRef\]](#)
5. Öztürk, E.; Rheinberger, K.; Faulwasser, T.; Worthmann, K.; Preißinger, M. Aggregation of Demand-Side Flexibilities: A Comparative Study of Approximation Algorithms. *Energies* **2022**, *15*, 2501. [\[CrossRef\]](#)
6. Đapić, P.; Pavkov, I.; Crvenković, S.; Tanackov, I. Generating Integrally Indecomposable Newton Polygons with Arbitrary Many Vertices. *Mathematics* **2022**, *10*, 2389. [\[CrossRef\]](#)
7. Bernholt, T.; Eisenbrand, F.; Hofmeister, T. Constrained Minkowski sums: A geometric framework for solving interval problems in computational biology efficiently. *Discret. Comput. Geom.* **2009**, *42*, 22–36. [\[CrossRef\]](#)
8. Suriyababu, V.K.; Vuik, C.; Möller, M. Towards a High Quality Shrink Wrap Mesh Generation Algorithm Using Mathematical Morphology. *Comput.-Aided Des.* **2023**, *164*, 103608. [\[CrossRef\]](#)
9. Minarčík, J.; Estep, S.; Ni, W.; Crane, K. Minkowski Penalties: Robust Differentiable Constraint Enforcement for Vector Graphics. In Proceedings of the SIGGRAPH'24: ACM SIGGRAPH 2024 Conference Papers, Denver, CO, USA, 27 July–1 August 2024; Association for Computing Machinery: New York, NY, USA, 2024. [\[CrossRef\]](#)
10. Govindaraju, N.K.; Lin, M.C.; Manocha, D. Fast and reliable collision culling using graphics hardware. In Proceedings of the ACM Symposium on Virtual Reality Software and Technology, New York, NY, USA, 10–12 November 2004; VRST'04, pp. 2–9. [\[CrossRef\]](#)
11. Verhack, R.; Sikora, T.; Van Wallendael, G.; Lambert, P. Steered Mixture-of-Experts for Light Field Images and Video: Representation and Coding. *IEEE Trans. Multimed.* **2020**, *22*, 579–593. [\[CrossRef\]](#)
12. Huang, B.; Yu, Z.; Chen, A.; Geiger, A.; Gao, S. 2D Gaussian Splatting for Geometrically Accurate Radiance Fields. In Proceedings of the SIGGRAPH'24: ACM SIGGRAPH 2024 Conference Papers, Denver, CO, USA, 27 July–1 August 2024; Association for Computing Machinery: New York, NY, USA, 2024. [\[CrossRef\]](#)
13. Kerbl, B.; Kopanas, G.; Leimkuehler, T.; Drettakis, G. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Trans. Graph.* **2023**, *42*, 4. [\[CrossRef\]](#)
14. Yang, Z.; Yang, H.; Pan, Z.; Zhang, L. Real-time Photorealistic Dynamic Scene Representation and Rendering with 4D Gaussian Splatting. *arXiv* **2024**, arXiv:2310.10642. [\[CrossRef\]](#)
15. Chirikjian, G.S.; Shiffman, B. Applications of convex geometry to Minkowski sums of m ellipsoids in \mathbb{R}^N : Closed-form parametric equations and volume bounds. *Int. J. Math.* **2021**, *32*, 2140009. [\[CrossRef\]](#)
16. Ruan, S.; Chirikjian, G.S. Closed-form Minkowski sums of convex bodies with smooth positively curved boundaries. *Comput.-Aided Des.* **2022**, *143*, 103133. [\[CrossRef\]](#)

17. Lee, I.K.; Kim, M.S.; Elber, G. Polynomial/Rational Approximation of Minkowski Sum Boundary Curves. *Graph. Model. Image Process.* **1998**, *60*, 136–165. [\[CrossRef\]](#)
18. Ruan, S.; Poblete, K.L.; Li, Y.; Lin, Q.; Ma, Q.; Chirikjian, G.S. Efficient Exact Collision Detection between Ellipsoids and Superquadrics via Closed-form Minkowski Sums. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 1765–1771. [\[CrossRef\]](#)
19. Courteaux, M.; Mareen, H.; Ramlot, B.; Lambert, P.; Van Wallendael, G. Dimensionality Reduction for the Real-Time Light-Field View Synthesis of Kernel-Based Models. *Electronics* **2024**, *13*, 4062. [\[CrossRef\]](#)
20. Gilbert, E.; Johnson, D.; Keerthi, S. A fast procedure for computing the distance between complex objects in three-dimensional space. *IEEE J. Robot. Autom.* **1988**, *4*, 193–203. [\[CrossRef\]](#)
21. Ghosh, P.K. A unified computational framework for Minkowski operations. *Comput. Graph.* **1993**, *17*, 357–378. [\[CrossRef\]](#)
22. Agarwal, P.K.; Flato, E.; Halperin, D. Polygon decomposition for efficient construction of Minkowski sums. *Comput. Geom.* **2002**, *21*, 39–61. [\[CrossRef\]](#)
23. Kaul, A.; Farouki, R.T. Computing Minkowski Sums of Plane Curves. *Int. J. Comput. Geom. Appl.* **1995**, *05*, 413–432. [\[CrossRef\]](#)
24. Yan, Y.; Chirikjian, G.S. Closed-form characterization of the Minkowski sum and difference of two ellipsoids. *Geom. Dedicata* **2015**, *177*, 103–128. [\[CrossRef\]](#)
25. Fahim Golestaneh, A. A Closed-Form Parametrization and an Alternative Computational Algorithm for Approximating Slices of Minkowski Sums of Ellipsoids in R3. *Mathematics* **2023**, *11*, 137. [\[CrossRef\]](#)
26. Gruber, P.M. *Convex and Discrete Geometry*; Springer: Berlin/Heidelberg, Germany, 2007; Volume 336. [\[CrossRef\]](#)
27. Schneider, R. *Convex Bodies: The Brunn–Minkowski Theory*; Cambridge University Press: Cambridge, UK, 2013; Volume 151. [\[CrossRef\]](#)
28. Ghosh, P.K.; Kumar, K.V. Support function representation of convex bodies, its application in geometric computing, and some related representations. *Comput. Vis. Image Underst.* **1998**, *72*, 379–403. [\[CrossRef\]](#)
29. Sholokhov, O. Minimum-volume ellipsoidal approximation of the sum of two ellipsoids. *Cybern. Syst. Anal.* **2011**, *47*, 954–960. [\[CrossRef\]](#)
30. Kurzhanski, A.; Vályi, I. *Ellipsoidal Calculus for Estimation and Control*; Springer: Berlin/Heidelberg, Germany, 1997.
31. Durieu, C.; Walter, E.; Polyak, B. Multi-input multi-output ellipsoidal state bounding. *J. Optim. Theory Appl.* **2001**, *111*, 273–303. [\[CrossRef\]](#)
32. Kraft, D. A software package for sequential quadratic programming. In *Forschungsbericht- Deutsche Forschungs- und Versuchsanstalt für Luft- und Raumfahrt*; DLR German Aerospace Center—Institute for Flight Mechanics: Köln, Germany, 1988.
33. Halder, A. On the Parameterized Computation of Minimum Volume Outer Ellipsoid of Minkowski Sum of Ellipsoids. In Proceedings of the 2018 IEEE Conference on Decision and Control (CDC), Miami Beach, FL, USA, 17–19 December 2018; pp. 4040–4045. [\[CrossRef\]](#)
34. Harris, C.R.; Millman, K.J.; van der Walt, S.J.; Gommers, R.; Virtanen, P.; Cournapeau, D.; Wieser, E.; Taylor, J.; Berg, S.; Smith, N.J.; et al. Array programming with NumPy. *Nature* **2020**, *585*, 357–362. [\[CrossRef\]](#)

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.