Chapter 3

Hausdorff Distance Under Translations

3.1 Overview

In this chapter we consider the problem of matching two polyhedral sets in d dimensions under translations using the Hausdorff distance. We will use a configuration-space approach to solve this task.

3.1.1 Problem Statement

We identify $\mathbb{R}^d$ with its vector space, and identify a translation in $\mathbb{R}^d$ with its translation vector $t \in \mathbb{R}^d$. Let a be a k-simplex, i.e., $a = CH(p_1, \ldots, p_{k+1})$, where $p_1, \ldots, p_{k+1} \in \mathbb{R}^d$ are affinely independent points. We denote by $a + t := CH(t + p_1, \ldots, t + p_{k+1})$ the simplex translated by the translation t. Similarly, for the polyhedral set A we denote by $A + t := \{t + a_1, \ldots, t + a_m\}$ the polyhedral set translated by t.

We are now ready to formulate the matching problem we wish to solve:

Problem 1 (Hausdorff Optimization) *Let $A = \{a_1, \ldots, a_m\}$ and $B = \{b_1, \ldots, b_n\}$ in $\mathbb{R}^d$ be two polyhedral sets. We wish to find a translation $t^* \in \mathbb{R}^d$ such that*

$$\delta_{\mathrm{H}}(A + t^*, B) = \min_{t \in \mathbb{R}^d} \delta_{\mathrm{H}}(A + t, B) \; .$$

A way to attack an *optimization problem* such as Problem 1 is to consider the *decision variant* of it:

Problem 2 (Hausdorff Decision) *Let $A = \{a_1, \ldots, a_m\}$ and $B = \{b_1, \ldots, b_n\}$ in $\mathbb{R}^d$ be two polyhedral sets, and let $\varepsilon > 0$. We wish to decide if there exists a translation $t \in \mathbb{R}^d$ such that $\delta_{\mathrm{H}}(A + t, B) \leq \varepsilon$.*

The directed Hausdorff distance is often used to *partially* match the first object to the second one. Also the directed Hausdorff distance can be used to compute the undirected Hausdorff distance. We define the optimization and the decision problems for the directed case as well:

Problem 3 (Directed Hausdorff Optimization) *Let $A = \{a_1, \ldots, a_m\}$ and $B = \{b_1, \ldots, b_n\}$ in $\mathbb{R}^d$ be two polyhedral sets. We wish to find a translation $t^* \in \mathbb{R}^d$ such that*

$$\vec{\delta}_{\mathrm{H}}(A + t^*, B) = \min_{t \in \mathbb{R}^d} \vec{\delta}_{\mathrm{H}}(A + t, B) \; .$$

Problem 4 (Directed Hausdorff Decision) *Let $A = \{a_1, \ldots, a_m\}$ and $B = \{b_1, \ldots, b_n\}$ in $\mathbb{R}^d$ be two polyhedral sets, and let $\varepsilon > 0$. We wish to decide if there exists a translation $t \in \mathbb{R}^d$ such that $\vec{\delta}_{\mathrm{H}}(A + t, B) \leq \varepsilon$.*

Note that Problem 2 and Problem 4 are monotone in the parameter ε, which means that if there exists a translation that solves the problem for ε' then for all $\varepsilon \geq \varepsilon'$ there always exists a translation that solves the problem for ε. An efficient algorithm solving such a monotone decision problem can often be plugged as an oracle into an optimization scheme which solves the corresponding optimization problem by computing the optimal ε, and ideally introduces only a logarithmic or a poly-logarithmic multiplicative overhead in the runtime. Common optimization schemes are the parametric search [62], the expander-based approach of [54], or the randomized approach of [34]. See [4] for an overview. In practice one can always use unbounded binary search to compute the optimal ε to any precision, which introduces only the logarithm of the precision as an asymptotic factor in the runtime.

Recently more practical variants of the standard parametric search paradigm have been investigated by van Oostrum et al. [70]. They implemented an object-oriented framework in order to simplify the usually rather complicated implementation of parametric search. Furthermore, considering those decision problems for which a generic sorting algorithm can be used to produce the critical values for the parametric search, where usually Cole's trick [39] is applied, they show how to use the well-known Quicksort algorithm for this task. In the worst case this still introduces a quadratic logarithmic asymptotic factor in the running time. However experiments in [70] on computing the Fréchet distance for polygonal curves (see Definition 9 and Chapter 5) showed that in this case the usage of Quicksort even outperforms the always recommended binary search on the number space for ε.

3.1.2 Known Results

Different variants of the above matching problems have been investigated in the literature. For matching with respect to the Hausdorff distance under translations there are results for point sites in arbitrary dimensions or for sets of segments in two dimensions; see Table 3.1 for a summary. However in this setting there are almost no results so far in three or higher dimensions for more complicated sites than points. The only result we recently learned about is the result by Agarwal et al. [1], who compute the minimum Hausdorff distance under translations for two sets of L_2-disks in the plane in $O(mn(m+n)\log^3(mn))$ time and for two sets of L_2-balls in three dimensions in $O(m^2n^2(m+n)\log^3(mn))$ time.

For more complicated transformations, such as rigid motions or similarities for example, there are some results known in two dimensions, which are summarized in Table 3.2. Again, there are no results for higher dimensions known. However in different settings, such as exact point pattern matching, there are more results known for complicated transformation classes, also in higher dimensions, see for example [11, 69], and [15] for a survey.

Another strategy is, not to insist to find an optimal transformation, but to find an approximately optimal transformation with a considerably faster algorithm. Both of theoretical and

Table 3.1: Overview of known results for computing the Hausdorff distance under translations for sets of points or sets of non-intersecting segments of complexity m and n. $N := \max(m, n)$, and $\alpha(mn)$ denotes the so-called inverse Ackermann function, see [67].

Dim.	Metric	Points	Segments
$\mathbb{R}^1$	–	$O((m+n)\log(m+n))$ [64]	–
$\mathbb{R}^2$	L_∞	$O(nm\log^2(nm))$ [37]	$O(m^2n^2\alpha(mn))$ [53]
	L_2	$O(nm(n+m)\log(nm))$ [53]	$O(m^2n^2\log^3(mn)$ [7]
	L_p	$\Omega(N^3)$ [65]	$\Omega(N^4)$ [65]
$\mathbb{R}^3$	L_∞	$O(n^3\log^2 n)$ [38]	–
	L_2	$O(n^2m^2(n+m)^{1+\delta})$ [53]	–
$\mathbb{R}^d$	L_∞	$O(N^{\frac{4d-2}{3}}\log^2 N)$ [38]	–
		$\Omega(N^{\lfloor\frac{3d}{2}\rfloor})$ [38]	–
	L_2	$O(N^{\lceil\frac{3d}{2}\rceil+1}\log^3 N)$ [38]	–

Table 3.2: Known results for computing the Hausdorff distance between sets of sites in $\mathbb{R}^2$ under different transformation classes. Sites are either points or non-intersecting segments, and the complexity of the sets is m and n; $N := \max(m, n)$. †marks lower bounds for the directed Hausdorff distance.

Transformations	Metric	Points	Segments
Rigid motions	L_2	$O(m^2n^2N\log(mn))$ [36]	$O(m^3n^3\log(mn))$ [36]
	L_2	$\Omega(N^5)$ [65]	$\Omega(N^6)^\dagger$ [65]
Similarities	L_2, L_∞	random. expected for convex polygons: $O(n+m)$ [19]	
2 scaling factors	L_2	$\Omega(N^7)$ [65]	$\Omega(N^8)^\dagger$ [65]
2 scaling factors	L_∞	$\Omega(N^7)^\dagger$ [65]	$\Omega(N^8)^\dagger$ [65]
Affine maps	L_2, L_∞	$\Omega(N^9)$ [65]	$\Omega(N^{12})^\dagger$ [65]

practical interest for this are the so called *reference points*, see [12, 9] and also Section 3.5.1. They allow to find a transformation that yields a Hausdorff distance at most a constant times worse than the optimum distance. This constant is called the *loss factor*. In [9] it has been shown that the Steiner point is an optimal reference point, and algorithms were given for approximating the Hausdorff distance between sets of possibly intersecting segments or triangles in two or three dimensions, considering different transformation classes. Table 3.3 summarizes the results of [9]. It has to be noted that these approaches do not work for the directed Hausdorff distance.

For two sets of points, algorithms to approximate the directed Hausdorff distance under different transformation classes are given in [46]; the results are summarized in Table 3.4. They are based on a pinning strategy, that fixes one point of the first point set as a representative, and tries out all translations that map this representative to a point of the second point set. The algorithms there generalize in a straightforward way to also handle the undirected Hausdorff distance.

Table 3.3: Results from [9] for approximately matching sets of sets of m and n possibly intersecting segments or triangles in two or three dimensions under the Hausdorff distance for various transformation classes, using the Steiner point as a reference point. Let $H(m,n)$ denote the time to compute the Hausdorff distance between two sets.

Transformations	Dimension	Runtime	Loss Factor
Translations	2	$O((m+n)\log(m+n))$	$4/\pi+1$
	3	$O(H(m,n))$	2.5
Rigid motions	2	$O(mn\log(mn)\log^*(mn))$	$4/\pi+1$
	3	$O((mn)^3 H(m,n))$	2.5
Similarities	2	$O(mn\log(mn)\log^*(mn))$	$4/\pi+3$
	3	$O((mn)^3 H(m,n))$	4.5

Table 3.4: Results from [46] for approximately matching sets of m and n points under the directed Hausdorff distance for various transformation classes.

Transformations	Dimension	Runtime	Loss Factor
Translations	2	$O(nm\log n)$	2
	d	$O(nm\log n)$	$2+\varepsilon$
Rigid motions	2	$O(n^2 m\log n)$	4
	3	$O(n^3 m\log n))$	$8+\varepsilon$
Rotations	3	$O(n^2 m\log n)$	$4+\varepsilon$

3.1.3 Our Contribution

We consider the task of finding a translation which minimizes the Hausdorff distance between two sets of sites in higher dimensions. The approaches we consider construct the set of possible translations explicitly in one way or the other. This is what is called a configuration space approach. Therefore it is unfortunately not straightforward to generalize most of these ideas to rigid motions since this would involve the construction of arrangements of algebraic surface patches, for which there are only very limited results so far available. We consider a few different combinations of sites, metrics, the directed and the undirected Hausdorff distance, the decision problem and the optimization problem. For some of these cases there are different approaches applicable which allow a faster computation.

In Section 3.3 we investigate the task of finding a translation which does not exceed a given directed Hausdorff distance between a finite set A of points and a polyhedral set B. This approach, which utilizes an idea of [38], relies heavily on the fact that A consists of points only. In this case one can avoid the construction of the whole configuration space of all possible translations. We furthermore give a lower bound on the number of combinatorially different translations when B consists of k-simplices, which generalizes the bound by Rucklidge [65] for line segments in two dimensions. The results obtained in Section 3.3 are summarized in Table 3.5. The lower bound that we present in Section 3.3 relies on the fact that both sets are highly disconnected. In Section 3.4.4 we present the first lower bound for two curves. We show that for two polygonal curves in the plane of complexities m and n, respectively,

Table 3.5: Results of Section 3.3 concerning the decision problem of finding a translation which does not exceed a given directed Hausdorff distance between a set A of m points and a polyhedral set B of complexity n. The sites in B are non-intersecting. †marks lower bounds that use L_∞ only.

Metric	Dim.	Sites	Complexity	Runtime
polyh.	$d = 3$	line segments	$O(m^3n^2 \log n)$, $\Omega(m^3n^2)^{*\dagger}$	$O(m^4n^2 \log^2 n)$
polyh.	$d = 3$	triangles	$\theta(m^3n^3)$	$O(m^4n^3 \log n)$
L_2	$d = 3$	line segments	$O_\delta(m^3n^2)$, $\Omega(m^3n^2)$	$O_\delta(m^4n^2)$
polyh.	$d \geq 4$	simplices	$\theta(m^dn^d)$	$O(m^{d+1}n^d \log n)$
convex	$d \geq 2$	$(d-1)$-simplices	$\Omega(m^dn^d)$	–
L_∞, L_2	$d \geq 2$	k-simplices	$\Omega(m^dn^{k+1})$	–

Table 3.6: Results of Section 3.4 concerning the decision and optimization problem of finding a translation which minimizes the Hausdorff distance between two polyhedral sets A and B with complexity m and n, $N = \max(m, n)$. The underlying metric is polyhedral with constant description complexity.

Problem	Dim.	Complexity and Runtime
decision	$d = 3$	$O(N^9 \log^3 N)$
decision	$d \geq 4$	$O(N^{d^2+d})$
opt.	$d \geq 3$	$O(N^{d^2+d} \log^2 N)$

there are $\Omega(m^2n^2)$ combinatorially different translations for the directed Hausdorff distance between the two curves under translations.

We handle the case that both A and B are polyhedral sets and we wish to either decide or optimize the directed or undirected Hausdorff distance under translations, in Section 3.4. For this approach we have to construct the space of combinatorially different translations. The results of Section 3.4 are summarized in Table 3.6. In Section 3.5.1 we give a short introduction to reference points and raise the open problem to compute a reference point of optimal quality. In Section 3.5.2 we show a very simple generalization of the pinning technique of [46] to more general sites than points.

The sets A and B we consider in this chapter are either finite point sets or polyhedral sets. However for special polyhedral sets, such as terrains or convex polyhedra one can derive special algorithms which run much faster utilizing special properties of the sets. We investigate these cases in Chapter 4, where we consider the perpendicular distance for terrains, but also the Hausdorff distance.

3.2 Basic Properties of $\vec{\delta}_{\mathrm{H}}$

Now let us give some basic properties of the Hausdorff distance under translations. For this we need the notion of ε-neighborhoods.

Definition 12 (ε-Neighborhood) *Let $B \subseteq \mathbb{R}^d$, let ρ be a metric in $\mathbb{R}^d$, and let $\varepsilon > 0$.*

Then

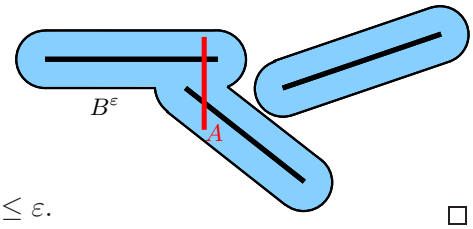
$$B^\varepsilon := B \oplus \mathbf{B}_d^\varepsilon$$

is the ε-neighborhood *of* B.

For example, the ε-neighborhood of a line segment in two dimensions has a racetrack-like shape whose boundary consists of line segments and circular arcs. The boundary of the ε-neighborhood of a triangle in $\mathbb{R}^3$ consists of copies of the triangle itself, and pieces of cylinders and spheres. Making use of ε-neighborhoods we can reformulate Definition 7 and obtain:

Lemma 1 *Let A and B be two compact sets, let ρ be a metric in $\mathbb{R}^d$, and let $\varepsilon > 0$. Then*

$$\vec{\delta}_{\mathrm{H}}(A, B) \leq \varepsilon \iff A \subseteq B^\varepsilon.$$



Proof: From Definition 7 follows that $\vec{\delta}_{\mathrm{H}}(A, B) \leq \varepsilon$ iff for each $x \in A$ there exists a $y \in B$ such that $\rho(x, y) \leq \varepsilon$. □

For the directed Hausdorff distance under translations we obtain the following reformulation, which has already been observed in [7, 36]:

Lemma 2 *Let A and B be two polyhedral sets in $\mathbb{R}^d$, t a translation in $\mathbb{R}^d$, and $\varepsilon > 0$. Then*

$$\vec{\delta}_{\mathrm{H}}(A + t, B) \leq \varepsilon \Longleftrightarrow t \in \bigcap_{a \in A} \overline{\overline{B^\varepsilon} \oplus (-a)}\ . \tag{3.1}$$

If A consists of points only, (3.1) simplifies to

$$t \in \bigcap_{a \in A} B^\varepsilon - a\ . \tag{3.2}$$

Proof:

$$\vec{\delta}_{\mathrm{H}}(A + t, B) \leq \varepsilon \iff A + t \subseteq B^\varepsilon \iff \textstyle\bigwedge_{a \in A} \bigwedge_{x \in a} x + t \in B^\varepsilon$$

$$\iff t \in \textstyle\bigcap_{a \in A} \bigcap_{x \in a} B^\varepsilon - x \iff t \in \overline{\bigcup_{a \in A} \bigcup_{x \in a} \overline{B^\varepsilon} - x} \iff t \in \textstyle\bigcap_{a \in A} \overline{\overline{B^\varepsilon} \oplus (-a)}$$

□

We call $L_a := \overline{\overline{B^\varepsilon} \oplus (-a)}$ the *a-layer*. One way to check if $\vec{\delta}_{\mathrm{H}}(A+t, B) \leq \varepsilon$ is to explicitly construct $\bigcap_{a \in A} L_a$. This is the approach that we follow in Section 3.4. This construction results in a sequence of set operations on polyhedral sets, when ρ is a convex polyhedral metric. When the L_2-metric is considered the boundaries of the ε-neighborhoods contain also parts of spheres and cylinders, such that the set operations have to be performed on these sets. The explicit computation of $\bigcap_{a \in A} L_a$ is very costly but can be applied to arbitrary sites. In Section 3.3 we generalize the approach by Chew et al. [38], which avoids an explicit construction and thus yields faster algorithms. However this approach works only in the restricted case that A consists of points only, and when the directed Hausdorff distance is considered.

3.3 Matching Points to Sites

In this section we present a combinatorial approach to decide, for a given $\varepsilon > 0$, if there exists a translation which brings a set of points $A = \{a_1, \ldots, a_m\} \subseteq \mathbb{R}^d$ to within directed Hausdorff distance ε from a polyhedral set $B = \{b_1, \ldots, b_n\} \subseteq \mathbb{R}^d$. In other words, we wish to solve Problem 4 for the special case that A consists of points only. Our approach is inspired by a result of Chew et al. [38], who considered the same problem for two point sets in $\mathbb{R}^2$. We show how to apply similar combinatorial ideas to our more general setting.

From Lemma 2 we know that solving Problem 4 is equivalent to deciding if $I := \bigcap_{a \in A} L_a \neq \emptyset$, where $L_a = B^\varepsilon - a$. The algorithm consists of a *generation phase* followed by a *decision phase*. In the generation phase a set M of candidate points is computed, which contains at least one point per connected component of I. In the decision phase for every point in M it is checked if it is contained in all layers L_a. This is done by performing point location queries in $B^\varepsilon - a$.

Consider an L_p-metric for fixed $1 \leq p \leq \infty$. Then the ε-neighborhood b^ε of an arbitrary k-dimensional simplex b in $\mathbb{R}^d$ is a semialgebraic set involving only polynomials of constant degree, assuming p and d are constant. A $(d-1)$-dimensional *face* (or $(d-1)$-*face*) of b^ε is a maximal connected closed subset of ∂b^ε which is contained in one algebraic surface. Hence, each $(d-1)$-face is an algebraic surface patch. A k-*face* of b^ε is a maximal connected closed subset of the intersection of a fixed subset of $(d-1)$-faces that avoids all other $(d-1)$-faces. Notice that if there are no degeneracies, meaning that all underlying algebraic surfaces involved in the intersection are distinct, then the intersection of $d-k$ such algebraic surface patches will result in (possibly several) k-faces. The *combinatorial complexity* of b^ε is the total number of faces of all dimensions. Consider for example the ε-neighborhood of a line segment in $\mathbb{R}^3$, with respect to the L_2 metric. Its boundary consists of a cylinder patch and two sphere patches which are the three 2-faces, and of two circles which are the two 1-faces.

This type of description of the boundary of a volume carries over to volumes obtained by set operations like unions and intersections of several b^ε, hence to the layers L_a as well as to I. See [8] for more information on algebraic surfaces or surface patches and their arrangements, and see [2] for a similar boundary representation.

Definition 13 ($\gamma_k(n)$, $\gamma'_k(n)$) *Let $\varepsilon > 0$ be fixed, and let ρ be a fixed metric defining $\mathbf{B}_\varepsilon$. Consider the union of n ε-neighborhoods of possibly intersecting k-simplices in $\mathbb{R}^d$. We denote by $\gamma_k(n)$ the maximum combinatorial complexity of such a union, and by $\gamma'_k(n)$ the worst-case time to compute one point on every connected component of such a union.*

Clearly $\gamma_k(n)$ and $\gamma'_k(n)$ are monotone increasing in n. In order to compute the set M of candidate points we need the following combinatorial lemma:

Lemma 3 *Let $\varepsilon > 0$ be fixed, and let ρ be a fixed metric defining $\mathbf{B}_\varepsilon$. Let $A = \{a_1, \ldots, a_m\} \subseteq \mathbb{R}^d$ be a set of points and let $B = \{b_1, \ldots, b_n\} \subseteq \mathbb{R}^d$ be a polyhedral set consisting of simplices of dimension at most k, $1 \leq k < d$. Let $L_a = B^\varepsilon - a$.*

Then $I := \bigcap_{a \in A} L_a$ has $O(m^d \gamma_k(dn))$ connected components, and one can construct one point per connected component of I in $O(m^d \gamma'_k(dn))$ time.

Proof: For the combinatorial part it suffices to count the number of μ-faces on ∂I, $0 \leq \mu < d$, which are not incident to another face of lower dimension. We call these μ-faces *minimal*.

Let f be a minimal μ-face of ∂I. There are $l \leq d - \mu$ layers $\mathcal{L} := \{L_{a_1}, \ldots, L_{a_l}\}$ such that f is contained in a μ-face $f_\mathcal{L}$ of the boundary of the intersection $I_\mathcal{L} := L_{a_1} \cap \ldots \cap L_{a_l}$. In fact,

f has to equal $f_{\mathcal{L}}$, because if there were a point x on $f_{\mathcal{L}}$ which was not on f, then x would have to be separated from f by a $(\mu-1)$-face in I, but this either contradicts the minimality of f, or the connectivity of $f_{\mathcal{L}}$. Hence, every minimal μ-face on ∂I is a minimal μ-face on $\partial I_{\mathcal{L}}$ for a set of layers $\mathcal{L}$. Since a μ-face $f \in \partial I$ cannot lie in the interior of any of the layers in $\mathcal{L}$, it is also a μ-face of the boundary of the union $U_{\mathcal{L}} := L_{a_1} \cup \ldots \cup L_{a_l}$ of the layers.

There are $\sum_{l=1}^{d} \binom{m}{l} = O(m^d)$ possibilities to choose $d-\mu$ layers, for $0 \leq \mu < d$. The union of those chosen layers consists of at most dn ε-neighborhoods of $b \in B$ and has the combinatorial complexity $\gamma_k(dn)$. One point on every connected component of the union can be constructed in $\gamma'_k(dn)$ time. Altogether the combinatorial complexity of I is $O(m^d\gamma_k(dn))$, and we can construct one point on every minimal μ-face on ∂I, $0 \leq \mu < d$, in $O(m^d\gamma'_k(dn))$ time. □

The candidate set M is the set of all points on I that are constructed according to Lemma 3. M contains at least one point per connected component of I, $|M| = O(m^d\gamma_k(dn))$, and M can be computed in $O(m^d\gamma'_k(dn))$ time.

In the decision step we compute B^ε and preprocess it into a point location data structure. Let $\beta_p(n)$ be the time for the computation and the preprocessing, and let $\beta_q(n)$ the time for a point location query. For each point in M we check if it is contained in $B^\varepsilon - a$ for every $a \in A$. The total time needed to check whether I is empty or not is therefore $O(m^d\gamma'_k(dn) + \beta_p(n) + m\beta_q(n)m^d\gamma_k(dn))$. Hence we obtain the following general result:

Theorem 1 *Let $\varepsilon > 0$ be fixed, and let ρ be a fixed metric. Let $A = \{a_1, \ldots, a_m\} \subseteq \mathbb{R}^d$ be a set of points and let $B = \{b_1, \ldots, b_n\} \subseteq \mathbb{R}^d$ be a polyhedral set consisting of simplices of dimension at most k, $1 \leq k < d$. Let $\beta_p(n)$ be the time for computing B^ε and for preprocessing it into a point location data structure, and let $\beta_q(n)$ be the time for a point location query.*

Then it can be decided whether there exists a translation t such that $\vec{\delta}_{\mathrm{H}}(A+t, B) \leq \varepsilon$ in $O(m^d\gamma'_k(dn) + \beta_p(n) + m^{d+1}\gamma_k(dn)\beta_q(n))$ time.

For reasonable instances† of γ_k, γ'_k, β_p and β_q the last summand $m^{d+1}\gamma_k(dn)\beta_q(n)$ will dominate the runtime. This is then a factor of $m\beta_q(n)$ times the combinatorial complexity $O(m^d\gamma_k(dn))$ of I, see Lemma 3.

Before we apply the results of Lemma 3 and Theorem 1 to the different special cases of sites and metrics, let us first show a lower bound on the combinatorial complexity of the intersection I of layers. It shows that the upper bound that can be derived on the combinatorial complexity on I is in most cases tight or almost tight. See Corollary 1, Corollary 2, and Corollary 4 below. This lower bound construction generalizes a result by Rucklidge [65] for line segments in two dimensions.

Lemma 4 *Let A be a set of m points in $\mathbb{R}^d$, $d \geq 2$, and let B be a polyhedral set of complexity n in $\mathbb{R}^d$ that consists exclusively of k-simplices for a fixed $k \in \{0, \ldots, d-1\}$. Let $\varepsilon > 0$. Then for $k = d-1$ and convex metrics of constant description complexity, as well as for arbitrary $k = 0, \ldots, d-1$ and the L_∞-metric and the L_2-metric the complexity of $I = \bigcap_{a \in A} B^\varepsilon - a$ can be $\Omega(m^d n^{k+1})$ in the worst case.*

Proof: For better readability we assume that m and n are powers of d; the following construction carries directly over to arbitrary values of m and n. For convenience we construct

†Which means that $\gamma_k(n) \approx \gamma'_k(n)$ and $\beta_p(n) \approx \gamma'_k(n)$.

a set of axis-parallel k-hyperrectangles. In order to obtain simplices we can either divide each hyperrectangle into a constant number of simplices, or construct for each hyperrectangle a big simplex of the same dimension that contains this simplex.

Let us first consider the case that $k = d - 1$. For given ε, m, n we choose $\delta < (k+1)\varepsilon/m$. Let $l := (n/(k+1) - 1)(2\varepsilon + \delta)$, and let $J := [0, l] \subseteq \mathbb{R}$. We construct $k+1$ groups of $n/(k+1)$ k-hyperrectangles, where the hyperrectangles inside each group are parallel, but hyperrectangles of different groups are orthogonal to each other. In particular, the i-th group of hyperrectangles, $i = 1, \ldots, k+1$, is defined as $f^{(i)} := \{f_1^{(i)}, \ldots, f_{n/(k+1)}^{(i)}\}$ where

$$f_j^{(1)} := \{j(2\varepsilon + \delta)\} \times J^k$$
$$f_j^{(i)} := ((i-1)(l + 2\varepsilon + \delta) + J) \times J^{i-2} \times \{j(2\varepsilon + \delta)\} \times J^{k+1-i} \quad \text{for } i = 2, \ldots, k+1\ ,$$

for $j = 0, \ldots, m/(k+1) - 1$. We let B consist of all these hyperrectangles.

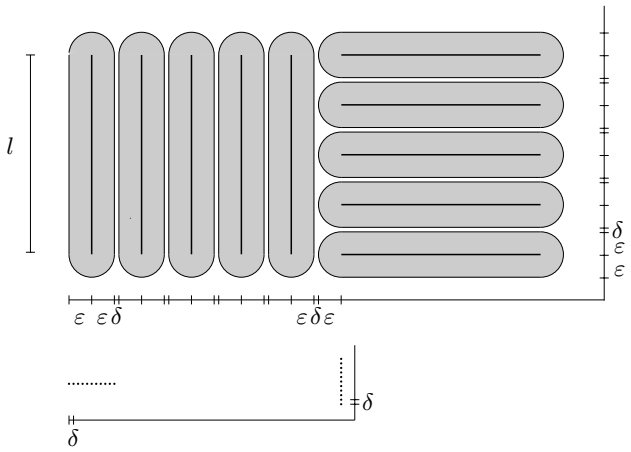


Figure 3.1: Lower bound for the overlay of ε-neighborhoods for the case of line segments in $d = 2$ and the Euclidean metric. The figure shows the segments, their ε-neighborhoods, and the point set.

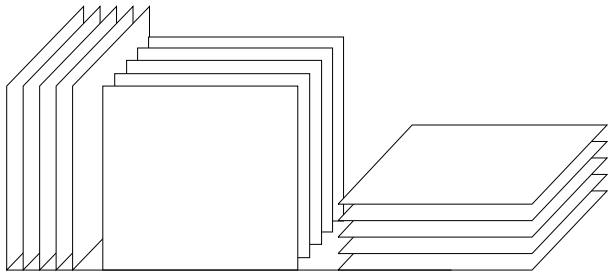


Figure 3.2: Lower bound for the overlay of ε-neighborhoods for the case of rectangles for $d = 3$, and the Euclidean metric. The figure shows only the three groups of rectangles.

For $d = 2$, B consists of $n/2$ vertical line segments spaced $(2\varepsilon + \delta)$ apart, and of $n/2$ horizontal line segments, see Figure 3.1. The construction for this particular 2-dimensional case has been presented by Rucklidge [65].

In $\mathbb{R}^3$ we have three groups of rectangles, see Figure 3.2. In general the k-hyperrectangles in the i-th group have the i-th unit vector $\vec{e}_i$ as their normal vector.

The set A of points consists of $(k+1)$ groups of $m/(k+1)$ points each. The i-th group of points, $i = 1, \ldots, k+1$, is defined as $p^{(i)} := \{p_1^{(i)}, \ldots, p_{m/(k+1)}^{(i)}\}$ where

$p_j^1 := j\delta\vec{e}_1$
$p_j^i := (i-1)(l+3\varepsilon+\delta)\vec{e}_1 + (j\delta-\varepsilon)\vec{e}_i \quad \text{for } i = 2,\dots,d$
for $j = 0,\dots,m/(k+1)-1$. Thus we have $m/(k+1)$ points spaced δ apart on a line in $\vec{e}_i$-direction, for $i = 1,\dots,k+1$. See Figure 3.1 and Figure 3.2 for two- and three-dimensional examples.

In B^ε each $f_j^{(i)}$ gives rise to a $(k+1)$-hyperrectangle of thickness ε in $\vec{e}_i$-direction, for any norm induced by a convex metric. The caps differ according to the different norms. By construction every two neighboring hyperrectangles $f_j^{(i)}$ and $f_{j+1}^{(i)}$ are spaced at distance $2\varepsilon+\delta$. Hence, there is a gap of width δ between the $(k+1)$-hyperrectangle of thickness ε in $\vec{e}_i$-direction induced by $f_j^{(i)}$ and $f_{j+1}^{(i)}$. In particular, each such gap is a $(k+1)$-hyperrectangle which is orthogonal to $\vec{e}_i$ and has thickness δ in $\vec{e}_i$-direction, and side-length l in all other directions. Note that this gap does not depend on the shape of the caps, and hence not on the norm.

The points in A are arranged in such a way that if $A \subseteq B^\varepsilon$ then the i-th point group has to be contained in the ε-neighborhood of the i-th hyperrectangle group. For fixed i, the i-th point group can be positioned such that the first of two particular consecutive points is to the left, and the second point to the right of a particular gap in the i-th hyperrectangle group. Left and right refer here to the $\vec{e}_i$-direction. Different choices of a consecutive point pair and gap are combinatorially different since there is no path in translation space between the two translations that is completely contained in I. There are $m/(k+1)-1$ consecutive point pairs, and $n/(k+1)-1$ gaps between ε-neighborhoods of hyperrectangles in one group; thus $(m/(k+1)-1)(n/(k+1)-1)$ different choices. Note that the choice of a point pair and a gap in the i-th group still allows to shift in all directions different from $\vec{e}_i$ while preserving the point pair / gap configuration. Therefore we can choose a point pair and a gap in each group independently, thus obtaining $((m/(k+1)-1)(n/(k+1)-1))^d = \Omega(m^{k+1}n^{k+1}) = \Omega(m^d n^d)$, since $d = k-1$, combinatorially different configurations, which is a lower bound for the combinatorial complexity of I.

Now let us consider the case of $k = 0,\dots,d-2$. In this case the caps of the ε-neighborhoods do have an influence on the construction. We show a construction each for the L_∞ and the L_2 metrics.

Let us first consider the easier case of the L_∞ metric. We choose $\delta < d\varepsilon/m$. We carry out the same construction as above in $\mathbb{R}^{k+1}$ for $m(k+1)/d$ points and $n/2^{d-k-1}$ k-hyperrectangles. We embed this construction canonically in $\mathbb{R}^d$, identifying $\mathbb{R}^{k+1}$ with $\mathbb{R}^{k+1} \times \{0\}^{d-k-1}$. We stack an identical copy of the constructed k-hyperrectangles at distance $2\varepsilon+\delta$ in $\vec{e}_{k+2}$-direction on top of the prior constructed hyperrectangles. We continue this copying process in directions $\vec{e}_{k+2},\dots,\vec{e}_d$, doubling the number of hyperrectangles in each step, thus obtaining n k-hyperrectangles altogether. To the already constructed point set in $\mathbb{R}^{k+1}$ we add m/d points in each of the directions $\vec{e}_{k+2},\dots,\vec{e}_d$, starting in the origin each and at a spacing of δ within each group. From the construction in $\mathbb{R}^{k+1}$ we obtain $\Omega(m^{k+1}n^{k+1})$ combinatorially different configurations. Since the ε-neighborhoods in the L_∞ metric are axis-parallel boxes, we can shift each such configuration by at most ε in each of the directions $\vec{e}_{k+2},\dots,\vec{e}_d$ without introducing combinatorial changes. In each of those remaining $d-k-1$ directions there is a gap of width δ between the two ε-neighborhoods of the copies of the hyperrectangles of one dimension lower. For each of the $m/d-1$ adjacent point pairs in the corresponding point group one can choose which of the points lies on which side of the gap, according to the direction under consideration. This choice can be done in each direction independently

since each point group has a diameter of $m/d\delta < \varepsilon$. We thus have altogether $\Omega(m^d n^{k+1})$ combinatorially different configurations.

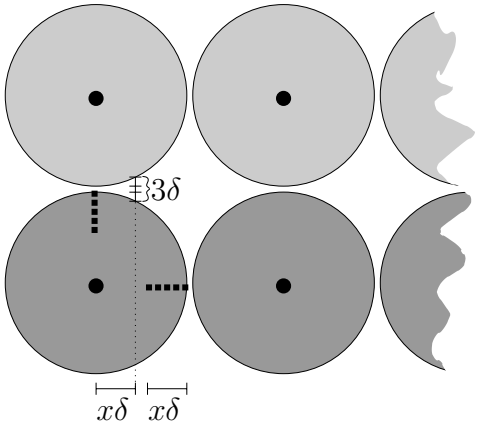


Figure 3.3: An example for 0-dimensional sites (points) in two dimensions. The first layer of disks is darker, and the copied layer is lighter.

The idea for the L_2 metric and $k = 0, \ldots, d-2$ is basically the same. We also carry out the first construction in $\mathbb{R}^{k+1}$ for a fraction of the points and the hyperrectangles, embed this construction canonically in $\mathbb{R}^d$. The construction for the hyperrectangles remains exactly the same, i.e., we recursively stack identical copies of the lower dimension in one dimension higher, just as for the case of the L_∞ metric. We will obtain as before $\Omega(m^{k+1}n^{k+1})$ combinatorially different configurations from the construction in $\mathbb{R}^{k+1}$. For the remaining directions $\vec{e}_{k+2}, \ldots, \vec{e}_d$ we now have to be careful with the cylindrical parts of the ε-neighborhood that are caused by the L_2 metric. Note that these have only to be considered for the remaining directions $\vec{e}_{k+2}, \ldots, \vec{e}_d$; the argument for the construction in $\mathbb{R}^{k+1}$ is not affected by the metric. The idea in this new setting is basically the same, however we enlarge the spacing between adjacent points of each point group slightly, and ensure that the diameter of each point group is small enough, such that shifts still allow two adjacent points in another group to be on two different sides of a gap.

For this let $\alpha := 3m/(3(k+1) + (d-k-1))$, and choose $\delta := 2\varepsilon/(\alpha^2+1)$. The point groups for the construction in $\mathbb{R}^{k+1}$ are of size α each. Note that with our choice of α and δ it still holds that for the diameter $\alpha\delta$ of such a point group $\alpha\delta < \varepsilon$ (for all $\alpha \geq 1$). For the remaining directions $\vec{e}_{k+2}, \ldots, \vec{e}_d$ we let the point groups be of size $\alpha/3$. The point group for direction $\vec{e}_i$, $k-2 \leq i \leq d$, starts at $(\varepsilon - \alpha\delta)\vec{e}_i$ and consists of $\alpha/3$ points spaced at distance 3δ in direction $\vec{e}_i$. The reason why the constant 3 appears is due to the shape of the L_2 sphere: The widths of the gaps between two spheres increases when shifting in some direction $\vec{e}_i$, see Figure 3.3. Precisely, a shift by ρ causes a gap of width $\delta + 2(\varepsilon - \sqrt{\varepsilon^2 - \rho})$. An easy calculation shows that with our choice of δ the gap will be at most 3δ when shifting no more than $\alpha\delta$ in any direction.

Thus we can, for each of the $\alpha/2 - 1$ adjacent point pairs in each of the new point groups choose which of the points lies on which side of the gap, according to the direction under consideration. From the earlier argument we see that this choice can indeed be done in each direction independently. Thus we have altogether $\Omega(m^d n^{k+1})$ combinatorially different configurations. □

Adjusting the technique of the lower bound construction of Lemma 4 to arbitrary other convex metrics is not quite straight-forward. Even for the L_1-metric the construction of the point sets cannot be carried out in the same way for $k < d-1$, since the gaps between the L_1-spheres are too large. An approach, which stacks the hyperrectangles in a better "packing" which reduces the space between the different layers, seems more promising for more general metrics. In general the stacking rule would then have to depend on the shape of the unit sphere and on the Minkowski sum of the unit ball with a hyperrectangle.

Note that the construction in Lemma 4 gives the first lower bounds for the directed Hausdorff distance under translations for sets of non-point sites in higher dimensions. Replacing each point in the construction by tiny simplices of higher dimensions, it follows that the lower bound for the directed Hausdorff distance under translations for two sets of k-dimensional sites in $\mathbb{R}^d$ is also $\Omega(m^d n^{k+1})$.

Now we are ready to consider special cases of sites and metrics and to apply Lemma 3, Theorem 1, and Lemma 4 to them.

Corollary 1 *Let A be a set of m points in $\mathbb{R}^2$ and let B consist of n line segments in $\mathbb{R}^2$. Then for any convex polyhedral metric of constant description complexity or any L_p-metric the maximum combinatorial complexity of $I = \bigcap_{a \in A} B^\varepsilon - a$ is $\theta(m^2 n^2)$ and the time to check if $I = \emptyset$, i.e., to solve Problem 4 is $O(m^3 n^2 \log n)$.*

Proof: Trivially an arrangement of n algebraic arcs of constant description complexity in $\mathbb{R}^2$ has complexity $\gamma(n) = O(n^2)$. With Lemma 3 and the lower bound of Lemma 4 and [65] follows the first statement. An arrangement of n algebraic arcs of constant description complexity in $\mathbb{R}^2$ can be computed in $O(n \log n + k)$ time [67], where k is the number of intersections. We therefore have $\gamma'(n) = O(n^2)$. In the case of polyhedral metrics a simple planar point location can be applied, whereas otherwise we have to apply a point location within an arrangement of algebraic surfaces, see [67], which needs $O(n^{2d-3+\varepsilon})$ preprocessing, where d is the dimension. In both cases a query needs $O(\log n)$ time, so that the term $O(m^3 n^2 \log n)$ dominates the running time. □

In two dimensions this approach is about a factor of m or n slower than the approach of [7]. But let us now turn to three dimensions.

Corollary 2 *Let A be a set of m points in $\mathbb{R}^3$ and let B consist of n non-crossing line segments or triangles in $\mathbb{R}^3$. Then for any convex polyhedral metric of constant description complexity the maximum combinatorial complexity of I is $\theta(m^3 n^3)$, and Problem 4 can be solved, i.e., it can be checked whether $I \neq \emptyset$ in randomized expected time $O(m^4 n^3 \log n)$.*

If B consists of line segments only, the maximum combinatorial complexity of I is $O(m^3 n^2 \log n)$ and the runtime to check whether $I \neq \emptyset$, i.e., to solve Problem 4, is randomized expected $O(m^4 n^2 \log^2 n)$. A lower bound for the combinatorial complexity of I in the case of the L_∞ metric is $\Omega(m^3 n^2)$.

Proof: Trivially $\gamma(n) = O(n^3)$, and since the b^ε are polyhedra we can use the randomized algorithm by Aronov et al. [21] to compute the union in randomized expected $\gamma'(n) = O(n^3)$ time. We apply the result for point location among hyperplanes by Chazelle et al. [35] with which $\beta_p(n) = n^d$ and $\beta_q(n) = \log n$. Applying Lemma 3 and Theorem 1 we obtain a combinatorial complexity of $O(m^3 n^3)$ and a randomized expected runtime of $O(m^4 n^3 \log n)$.

We are computing unions of Minkowski sums of disjoint simplices with the same ε-ball. For non-intersecting line segments we can apply the result of Aronov et al. [20] for the union of Minkowski sums, which states that $\gamma(n) = O(n^2 \log n)$ and $\gamma'(n) = O(n^2 \log^2 n)$ randomized expected. Although this result is stated for interior-disjoint polyhedra, it also applies to possibly crossing line segments, since a small perturbation of the line segments and points removes the crossings, but preserves the incidence relations of the faces forming the union under consideration. Applying again the result for point location among hyperplanes we obtain a total randomized expected runtime of $O(m^4 n^2 \log^2 n)$. The combinatorial complexity is, applying Lemma 3, $O(m^3 n^2 \log n)$. For the lower bound we apply Lemma 4.

□

Corollary 3 *Let A be a set of m points in $\mathbb{R}^3$ and let B consist of n non-crossing line segments in $\mathbb{R}^3$. Then for the L_2-metric it can be checked whether I is empty, which solves Problem 4, in $O_\delta(m^4 n^2)$ randomized expected time, for any $\delta > 0$. The maximum combinatorial complexity of I is $O_\delta(m^3 n^2)$ with a lower bound of $\Omega(m^3 n^2)$.*

Proof: We apply the result of Agarwal and Sharir [5, 6, 2] which states that in the case of line segments $\gamma(n) = \gamma'(n) = O_\delta(n^2)$, where $\gamma'(n)$ is the randomized expected runtime. Again we might need to perturb the input to assure that the line segments do not intersect. For the point location we apply the result for algebraic surfaces, see [67], which needs $O_\delta(n^{2d-3})$ preprocessing, where d is the dimension, and $O(\log n)$ query time. Plugging this into Theorem 1 we obtain a runtime of $O_\delta(m^4 n^2 \log n)$. The combinatorial part again follows from Lemma 3, and the lower bound from Lemma 4.
□

The following corollary states results for arbitrary dimension $d \geq 2$.

Corollary 4 *Let A be a set of m points in $\mathbb{R}^d$ and let B be a polyhedral set of complexity n in $\mathbb{R}^d$. Then for any polyhedral metric of constant description complexity the maximum combinatorial complexity of I is $\theta(m^d n^d)$ and the time to check whetehr I is empty, which solves Problem 4, is $O(m^{d+1} n^d \log n)$.*

Proof: In this case everything reduces to hyperplane arrangements. Trivially $\gamma(n) = \gamma'(n) = O(n^d)$, and the lower bound follows from Lemma 4. For point location among hyperplanes we apply again the result of Chazelle et al. [35] with $\beta_p(n) = n^d$ and $\beta_q(n) = \log n$. Applying Lemma 3 and Theorem 1 we obtain a combinatorial complexity of $O(m^d n^d)$ and a runtime of $O(m^{d+1} n^d \log n)$. The lower bound follows again from Lemma 4.

□

The results we have derived in this section are based on arrangement constructions. For arrangements of algebraic surfaces or surface patches in dimensions $d \geq 3$ there are only limited possibilities to assess their structure. It is possible to construct a point on every face of the arrangement [25, 27], or to construct a one-dimensional *road map* [28], but it is not possible to construct the whole arrangement. Therefore we could derive the higher dimensional result of Corollary 4 only for polyhedral distance functions for which arrangement constructions for hyperplanes can be used. Furthermore, we have derived results for the decision problem only. In order to determine an optimal translation we need to apply an optimization procedure. Theoretically one would want to apply the parametric search paradigm, but we are not aware of efficient parallel versions of our algorithms. In practice one would apply binary search in order to determine the minimum distance up to a given precision.

3.4 Matching Two Sets of Sites

Let $A = \{a_1, \ldots, a_m\}$ and $B = \{b_1, \ldots, b_n\}$ be two polyhedral sets in $\mathbb{R}^d$. We want to compute a translation $t \in \mathbb{R}^d$ that minimizes the Hausdorff distance $\delta_{\mathrm{H}}(A + t, B)$ with respect to any convex polyhedral metric of constant description complexity. Or in other words we wish to solve Problem 1 for these metrics. We follow the configuration space approach by Agarwal, Sharir and Toledo [7], in that we explicitly compute $\bigcap_{a \in A} L_a$ by constructing arrangements of hyperplanes. This explains the reason for considering polyhedral metrics only, since the surfaces needed to represent ε-neighborhoods of simplices are all linear, while other metrics would introduce algebraic surface patches of higher degree for which there is no known general way to construct a whole arrangement. We first show how to solve the decision problem for a given $\varepsilon > 0$, i.e., solve Problem 2. Afterwards we apply the parametric search technique for optimization in order to solve Problem 1. For this we have to assume an appropriate notion of general position for the simplices in the input sets.

3.4.1 Decision Problem

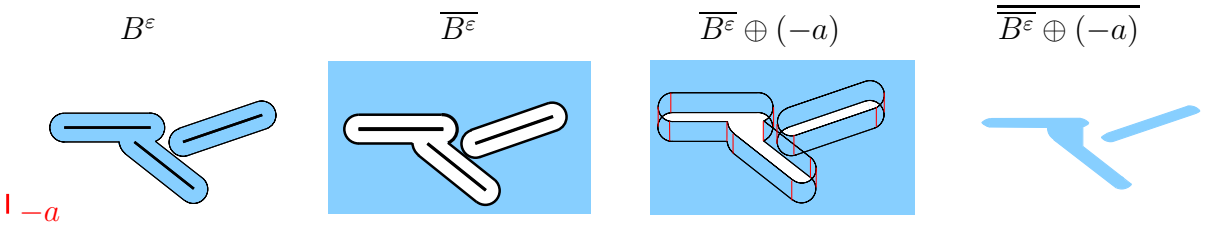


Figure 3.4: Construction of I.

For simplicity we first consider the decision problem for the directed Hausdorff distance only: Given an $\varepsilon > 0$ we wish to check if a $t \in \mathbb{R}^d$ exists such that $\vec{\delta}_{\mathrm{H}}(A + t, B) \leq \varepsilon$. As we have shown in (3.1), the existence of such a t is equivalent to $I := \bigcap_{a \in A} \overline{\overline{B^\varepsilon} \oplus (-a)} \neq \emptyset$. For each simplex $a \in A$ let $x_a \in a$ be an arbitrary fixed point in a. We call $L_a := \overline{\overline{B^\varepsilon} \oplus (-a)}$ the *a-layer*. Then from the definition of the Minkowski sum directly follows, taking into account that the facets of $\overline{B^\varepsilon}$ are exactly the facets of B^ε:

$$L_a = \overline{(\overline{B^\varepsilon} - x_a) \cup \bigcup_{F \text{ facet of } B^\varepsilon} F \oplus (-a)} \tag{3.3}$$

$$= (B^\varepsilon - x_a) \setminus \bigcup_{F \text{ facet of } B^\varepsilon} F \oplus (-a) \tag{3.4}$$

Each set b^ε or $F \oplus (-a)$ can be represented by a number of surface patches describing the set boundary. In fact, in the case of a polyhedral metric of constant description complexity each such set is a convex polyhedron of constant complexity, assuming as usual the dimension d to be constant. A straightforward way to check if $I \neq \emptyset$ is to construct I by applying the needed set operations (union, relative complement, intersection) to these polyhedra.

3.4.2 Arrangements of Volume Representations

Due to the lack of more efficient algorithms for the needed set operations in d dimensions we represent the volumes by a general arrangement construction. Each volume can be described by its bounding facets, i.e. $(d-1)$-faces, which in the case of a convex polyhedral metric are parts of hyperplanes. We thus represent each volume by an arrangement of hyperplanes, in which we equip each facet which is part of the volume describing boundary with a pointer telling on which side of the facet the described volume resides. In order to support the desired set operations we allow the pointers to be labeled: For each $a \in A$ we have two possible pointer labels p_a and $-p_a$, where the volume being pointed to by a p_a-labeled pointer belongs to L_a and a $-p_a$-label describes the complement of L_a. No label indicates that the described volume is neither part of L_a nor of $\overline{L_a}$.

Each layer L_a is built incrementally, according to (3.4). Simple volumes, like $b^\varepsilon - x_a$ or $F \oplus (-a)$, are constructed by a simple incremental arrangement construction algorithm, see [40], which constructs the incidence graph of the arrangement of M hyperplanes in $\theta(M^d)$ time. The facets describing $B^\varepsilon - x_a$ are equipped with p_a-labeled pointers, while those describing $F \oplus (-a)$ are equipped with $-p_a$-labeled pointers. The overlay of all these arrangements yields the volume representation for L_a. By traversing this final arrangement cell by cell, crossing cells via facets only, the pointer information can be utilized in order to maintain a counter of how many layers a certain cell belongs to. All cells contributing their volume to I can be enumerated in this way. For this and for the parametric search presented in Section 3.4.3 we need to make the assumption that all sites of A and B are in general position. All we need to guarantee is that every two volumes with different labels cannot touch each other in a lower dimensional face, other than a facet. A standard way to guarantee this is to assume that all coordinates of points are transcendentals that are algebraically independent over the rationals, see [22] for example.

The overlay of two volume representations, described by N and M hyperplanes, respectively, can be computed as follows: First the arrangement of all $N+M$ hyperplanes is computed incrementally in $\theta((N+M)^d)$ time. Then the volume-pointer information of the two original arrangements has to be transferred to the combined arrangement. This can be done by traversing the combined arrangment while keeping track of the corresponding position in the original arrangment to label each facet according to its volume-pointer information in the original arrangement. All this can be done in time proportional to the complexity of the collective arrangement.

In general $O(n^d)$ is an upper bound for the complexity of B^ε, and the incidence graph representation for it can also be computed in $O(n^d)$ time. We therefore have $O(n^d)$ facets of constant complexity describing one layer, hence $O(mn^d)$ facets describing all layers, which results in an arrangement of complexity $O(m^d n^{d^2})$ which can also be computed in this time. In order to compute the undirected Hausdorff distance we have to overlay the arrangements for both directed distances, which results in an arrangement of complexity and construction time $O(\max(m^d n^{d^2}, n^d m^{d^2}))$.

Assuming general position the traversing of the arrangement in order to find a cell which is covered by all layers can be done in a depth-first manner at asymptotically no extra cost.

For $d = 3$ we use the result of Aronov and Sharir [20] which states that the complexity of B^ε is $O(n^2 \log n)$ and that an incidence representation for it can be computed in $O(n^2 \log^2 n)$ time. Thus in this case we have $O(n^2 \log n)$ facets describing one layer. Hence, the complexity of the intersection is $O(\max(m^3 n^6 \log^3 n, n^3 m^6 \log^3 m)) = O(N^9 \log^3 N)$ and it can be

constructed and traversed in $O(\max(m^3n^6\log^6 n, n^3m^6\log^6 m)) = O(N^9\log^6 N)$ time, with $N = \max(m,n)$.

Theorem 2 *Let $A = \{a_1,\ldots,a_m\}$ and $B = \{b_1,\ldots,b_n\}$ be two polyhedral sets in $\mathbb{R}^d$ in general position. Let $\varepsilon > 0$, and let $N := \max(m,n)$. Considering a convex polyhedral metric of constant description complexity, Problem 2 can be solved, i.e., a translation t such that $\delta_{\mathrm{H}}(A+t,b) \le \varepsilon$ can be constructed, if it exists, in $O(N^{d^2+d})$ time. For $d = 3$ this improves to $O(N^9\log^6 N)$ time and $O(N^9\log^3 N)$ space.*

Of course Theorem 2 also holds for the directed Hausdorff distance, since the arrangement construction is the same, without overlaying the arrangements for both sides.

3.4.3 Optimization

For the optimization we apply the parametric search paradigm of Megiddo using the algorithm of Section 3.4.2 as an oracle. Furthermore, we need a parallel version of the oracle. However we are not aware of a parallel algorithm to compute the union of Minkowski-sum-polytopes. Therefore we consider only the case of an arbitrary d, and not the improved version for 3 dimensions. An arrangement of n hyperplanes can be constructed in parallel in $O(\log n)$ time using n^d processors, see Goodrich [47]. The boundary pointers can also be maintained at no extra cost in this setting. We can therefore implement the decision oracle in parallel analogously to Agarwal, Sharir, Toledo [7]: After the arrangement is constructed in parallel, we consider an auxiliary graph which has a node for each cell in the arrangement, and an arc for each $(d-1)$-face in the arrangement. Assuming general position, the depth of each cell can then be calculated by first traversing this auxiliary graph in an Euler path and then calculating the depths, i.e. the number of layers covering a cell, with a parallel prefix algorithm. As stated in [7] this can be done with $O(n^2)$ processors in $O(\log n)$ time. Using this algorithm as a parallel oracle in the parametric search paradigm we obtain a runtime of $O(\max(n^dm^{d^2}, m^dn^{d^2})\log^2 N) = O(N^{d^2+d}\log^2 N)$ with $N = \max(n,m)$.

Theorem 3 *Let $A = \{a_1,\ldots,a_m\}$ and $B = \{b_1,\ldots,b_n\}$ be two polyhedral sets in $\mathbb{R}^d$ in general position. A translation minimizing the Hausdorff distance between A and B under a convex polyhedral metric of constant description complexity can be computed in $O(\max(m^dn^{d^2}, n^dm^{d^2})\log^2(\max(m,n)))$ time.*

In order to apply this approach to other metrics, like the L_2-metric, it is necessary to be able to construct the incidence graphs of arrangements of surface patches in arbitrary dimensions. There is no algorithm known for dimensions $d \ge 3$, except an algorithm which preprocesses an arrangement of m surface patches into a point location data structure in $O(m^{2d-3+\varepsilon})$ time, see [67].

3.4.4 Lower Bounds

The lower bound of Lemma 4 also applies to the case of two polyhedral sets, by replacing the points by tiny higher-dimensional simplices. Thus Lemma 4 yields a lower bound of $\Omega(m^dn^{k+1})$ for the number of combinatorially different translations for the directed Hausdorff distance between A and B, if B consists of k-simplices. Unfortunately this leaves a very

large gap between the upper bound $O(m^d n^{d^2})$ for $d \geq 4$ and $O(m^3 n^6 \log^3 n)$ for $d = 3$, see Theorem 2.

The constructions in [65] and Lemma 4 rely on the fact that the point set A is disconnected. But what about connected shapes like curves – does the connectivity help to compute the minimum directed Hausdorff distance under translations faster? At least for curves in two dimension the answer is no. We present a lower bound construction for two polygonal curves for which the number of combinatorially different translations for the directed Hausdorff between them is $\Omega(m^2 n^2)$, where m and n are the complexities of the two polygonal curves.[‡] We use the L_∞ metric, but we are positive that the basic idea of the construction can be used to construct a lower bound for the Euclidean metric also.

Let $\varepsilon > 0$ and m and n be given, and set $\delta := \varepsilon/(mn)$. The lower bound construction is shown in Figure 3.5. P and Q are both polygonal curves, where P consists of $O(m)$ line segments and Q of $O(n)$ line segments. Q is constructed in that way that there are two groups of $(n-1)$ square holes of sidelength δ each in Q^ε. Those holes and another vertical hole of height ε are shown in grey in Figure 3.5. P consists of two groups of $m-1$ line segments of length $\varepsilon + (n-1)\delta$ each, where one group consists of horizontal line segments only, and the other of vertical line segments. The spacing between the line segments is δ, and they are arranged in such a way that the first line segment group has to lie in between the first group of holes, and the second line segment group has to lie in between the second group of holes in order for P to be contained in Q^ε. Of course the line segments in P are furthermore connected by auxiliary line segments in order to form a connected curve. Now one can choose one of the $O(n)$ holes in the first group to lie in between the line segments of the first group, and one of the $O(m)$ gaps between the line segments which should contain the hole. This yields $O(mn)$ combinatorially different placements of P, since the line segments are that long that there is no path in translation space between two such placements such that the translated P is always contained in Q^ε. But note that in horizontal direction the segments can be moved for at least a distance of ε, while maintaining the $O(mn)$ combinatorially different translations. This allows to independently find $O(mn)$ combinatorially different placements for the second group of line segments and the second group of holes. Thus there are altogether $O(m^2 n^2)$ combinatorially different translations for the directed Hausdorff distance between P and Q.

Lemma 5 *For every $\varepsilon, m, n > 0$ there exist two polygonal curves P and Q of complexities m and n, respectively, in $\mathbb{R}^2$, such that there are $\Omega(m^2 n^2)$ combinatorially different translations that bring P within directed Hausdorff distance at most ε to Q.*

3.5 Approximate Algorithms

The algorithms presented in the previous sections have rather high time complexities and are therefore more of theoretical than of practical interest. It is therefore interesting to devise approximate algorithms that do not compute the optimal Hausdorff distance under translations, but that compute some translation which results in a Hausdorff distance that is only a constant factor worse than the optimum. An easy and efficient way to obtain approximate results are the *reference point* methods which have been presented in [9]. For reasons of consistency we devote Section 3.5.1 to this approach, although we do not present new results in this area. However we raise the question to compute a reference point of optimal

[‡]We thank Alon Efrat for ideas and discussions regarding this example.

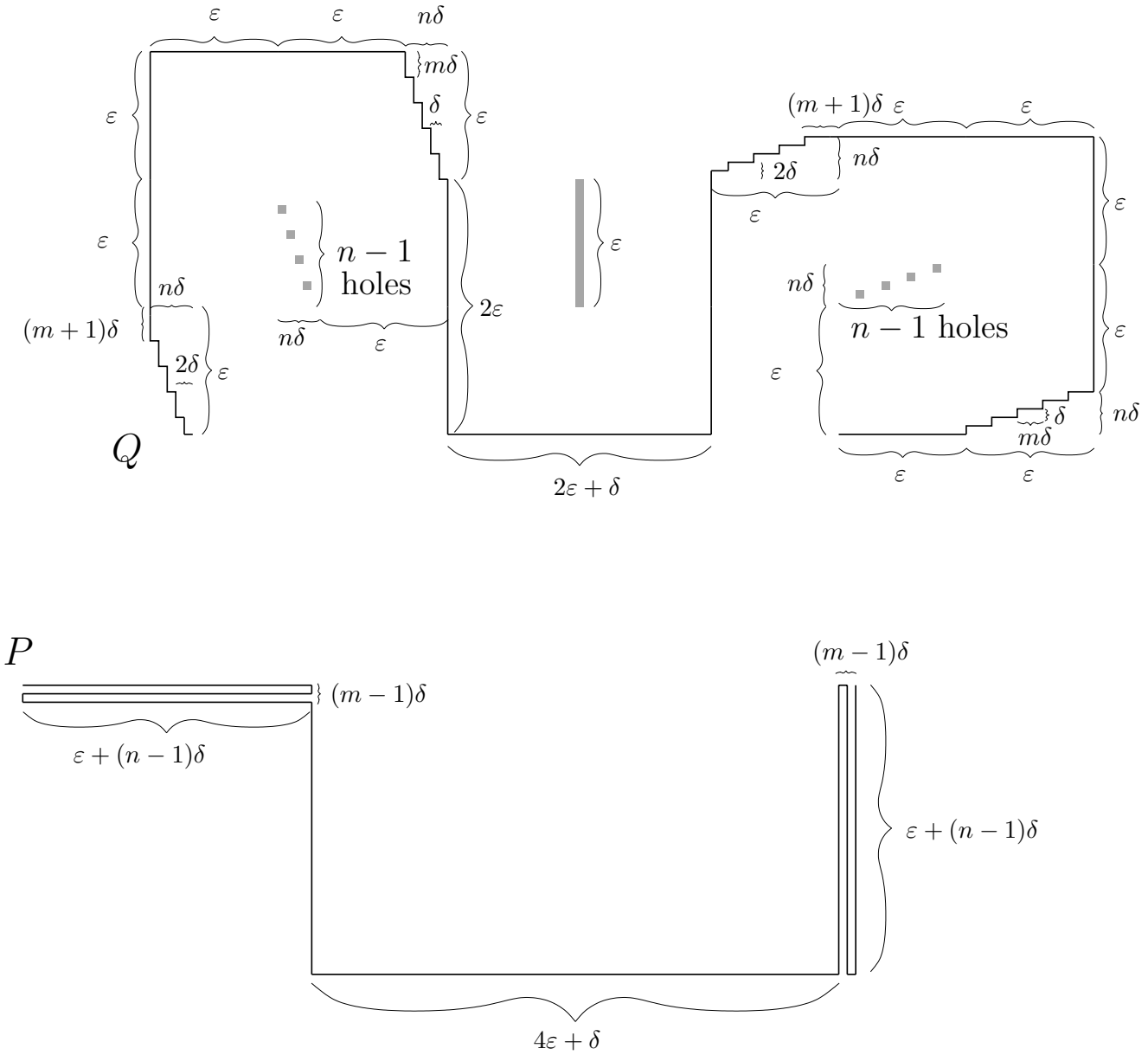


Figure 3.5: Lower bound for the number of combinatorially different translations for the directed Hausdorff distance between two polygonal curves of complexities m and n.

quality for the Hausdorff distance under translations in higher dimensions. In Section 3.5.2 we consider the approximation result by [46], and present a straightforward generalization to sites other than points.

For future research it is challenging to develop algorithms that give a constant factor approximation for the matching of two sets of sites other than points in dimensions larger than three. A constant factor approximation for the directed Hausdorff would be especially interesting, since the elegant reference points do not generalize to this setting.

3.5.1 Reference Point Methods

The reference points methods of [9] yield algorithms to approximate the Hausdorff distance under different transformation classes, involving only a constant loss factor. The theoretical results are shown for arbitrary dimension, however the computation of the reference points, as well as of the Hausdorff distance itself, has only been stated for two and three dimensions.

Definition 14 (Approximate matching / Loss factor) *Let A, B be two shapes, δ a distance measure, and $\mathcal{T}$ a class of transformations. Then the* approximate matching problem *with* loss factor *a, for some $a > 1$, is to find a $T' \in \mathcal{T}$ such that*

$$\delta(T'(A), B) \leq a \cdot \min_{T \in \mathcal{T}} \delta(T(A), B) \ .$$

An approximate solution is thus only a multiplicative factor a off from the optimal solution. In the case where the optimal solution is very time consuming to compute, the development of efficient approximation algorithms comes in handy for practical purposes, but also from a theoretical point of view. The general concept of reference points, see [12, 9], yields very elegant and efficient approximation algorithms for the matching problem.

Definition 15 (Reference point) *Let $\mathcal{C}^d$ be the set of all convex subsets of $\mathbb{R}^d$, δ a metric on $\mathcal{C}^d$, and $\mathcal{T}$ be a set of transformations on $\mathbb{R}^d$. Then a mapping $r : \mathcal{C}^d \to \mathbb{R}^d$ is called a* reference point with respect to *$\mathcal{T}$ if it has the following two properties:*

- **Equivariance:** *$r(T(A)) = T(r(A))$ for all $A \in \mathcal{C}^d$ and $T \in \mathcal{T}$*
- **Lipschitz continuity:** *There exists a $c \geq 0$ such that for all $A, B \in \mathcal{C}^d$*

$$||r(A) - r(b)|| \leq c \cdot \delta(A, B) \ .$$

The Lipschitz constant c is called the quality *of the reference point r.*

Using reference points, approximation algorithms for the Hausdorff distance under translations, rigid motions, and similarities have been given in [9]. Recently Knauer [56] has shown that there exist no reference points for the Hausdorff distance under arbitrary affine maps. Given a reference point of quality c with respect to translations and the Hausdorff distance it has been shown in [9] that the translation that maps the two reference points of the convex hull of two shapes onto each other is an approximation algorithm for the Hausdorff distance under translations with loss factor $c + 1$. Note that this approach however only works for the undirected Hausdorff distance and not for the directed Hausdorff distance.

Thus, in order to obtain an approximation algorithm with a small loss factor, one would like to use a reference point with a good (i.e. small) quality. In [12] it has been shown that

the lower left corner of the bounding box of the shape is a reference point of quality $\sqrt{2}$ for translations. For d dimension the quality of this reference point is $\sqrt{d}$. In [9] however it has been shown that the so called *Steiner point* is a reference point of much better quality, and that in the case of translations there is no other reference point of better quality. The quality of the Steiner point in d dimensions is $\chi_d = 2\Gamma(d/2+1)/(\sqrt{\pi}\Gamma(d/2+1/2))$, which lies in the range $\sqrt{2/\pi}\sqrt{d}$ and $\sqrt{2/\pi}\sqrt{d+1}$. This is roughly a factor of $\sqrt{2/\pi} \approx 0.7979$ better than the quality $\sqrt{d}$ of the lower left corner of the bounding box.

The runtime of the approximate matching algorithm consists of the time needed to compute the convex hulls, the reference points, and the Hausdorff distance. However, it is not known how to compute the Steiner point in dimensions $d > 3$. See [48] for more results on Steiner points. The computation following one possible definition of Steiner points involves the computation of higher dimensional external angles, which can be reduced to computing the volume of spherical simplices. The last problem however is an intriguing task which has already been attacked by Poincaré [63] and has not been solved since. Other approaches are by Sommerville [68] and Böhm and Hertel[32]. Poincaré [63] and Sommerville [68] generalize the approach of the well-known theorem of Girard, see e.g. [29], for the volume of spherical triangles. They establish recurrences for the volume, however the formula only holds for spherical simplices of even dimension. [§] So although the Steiner point is known to be a reference points with optimal quality for the Hausdorff distance under translations, it is not known how to compute it for higher dimensions, and there is no other reference point of the same quality known. Thus the current reference point of choice for the Hausdorff distance under translations is to take some corner of the bounding box which has quality $\sqrt{d}$ in d dimensions.

3.5.2 Sampling and Pinning

We present a very easy way to adapt the pinning approach of [46] that approximates the directed Hausdorff distance of two finite sets of points to the case of sites more general than points. For this we sample the sites uniformly by points, and apply the same simple pinning strategy. This approach does not yield a constant factor approximation, but adds an additive term, that depends on the sampling density, to the optimal distance. However in practice the Hausdorff distance is most commonly computed between sets of pixels, that are sampled from the actual surfaces under consideration. And this approach gives on the one hand an idea of how much the sampling density actually alters the true Hausdorff distance, but on the other hand gives a fast algorithm to approximate the Hausdorff distance in this setting (once one has decided to consider sampled points anyway). We use the Euclidean metric and the directed Hausdorff distance, but this approach generalizes as well to other metrics and to the undirected Hausdorff distance.

Let A and B be two polyhedral sets in $\mathbb{R}^d$ of complexities m and n respectively. And let B consist of simplices of dimension at most k. Let V be the maximum k-dimensional volume of any simplex in B, and let $\gamma > 0$ be a parameter. Then we sample each simplex $b \in B$ with a uniform grid of at most V/γ points. Denote the set of all of these points B'. Clearly $|B'| \leq nV/\gamma$. We pick an arbitrary representative point a_0 on some simplex of A. Then we try out all translations that map a_0 onto some point of B' (we *pin* a_0 to this point), and for

[§] By now the reader should have noticed that we have tried to extend this formula to work in all dimensions, and thus to compute the Steiner point in higher dimensions. However it seems to be an unexpectedly intriguing open problem.

each such translation t we compute $\vec{\delta}_{\mathrm{H}}(A+t, B')$. For the last part we use the same nearest neighbor algorithms as in [46] to compute for each $a \in A+t$ its nearest neighbor in B'. For two dimensions B' can be preprocessed into a Voronoi diagram with point location structure in $O(n \log n)$ such that nearest neighbor queries can be answered in $O(\log n)$ time. For $d > 2$ we use the data structure of Arya et al. [23] which preprocesses B' in $O(n \log n)$ time and answers approximate nearest neighbor queries in $O(\log n)$ time, always returning a neighbor whose distance is at most $(1+\varepsilon)$ larger than the distance to the nearest neighbor, for a fixed $\varepsilon > 0$. This algorithm takes $O(|B'| \log |B'|)$ time for the preprocessing and $O(|B'| \cdot |A| \cdot \log |B'|)$ time for all nearest neighbor queries, hence $O(nV/\gamma \log(nV/\gamma) + mnV/\gamma \log(nV/\gamma))$ time altogether. Let h_{opt} denote the minimum directed Hausdorff distance between A and B for any translation applied to A. We claim that this algorithm computes a translation t_{approx} such that $\vec{\delta}_{\mathrm{H}}(A + t_{\mathrm{approx}}, B) \leq 2 * h_{\mathrm{opt}} + \sqrt{k}\gamma$ for $d = 2$, and such that $\vec{\delta}_{\mathrm{H}}(A + t_{\mathrm{approx}}, B) \leq (2+\varepsilon) * h_{\mathrm{opt}} + \sqrt{k}\gamma$ for $d > 2$, and any $\varepsilon > 0$.

Indeed, consider an optimal translation t_{opt}, such that $\vec{\delta}_{\mathrm{H}}(A + t_{\mathrm{opt}}, B) = h_{\mathrm{opt}}$. Then each point $a \in A + t_{\mathrm{opt}}$ has a point b on some simplex of B in distance at most h_{opt}. Let b_0 on some simplex of B be the closest point to a_0. Let first $d = 2$. With t_{approx} we map a_0 to its closest point $b'_0 \in B'$, which has distance at most $\sqrt{k}$ to b_0. Thus each point $a \in A + t_{\mathrm{opt}}$ might get mapped by t_{approx} to a point at distance $h_{\mathrm{opt}} + h_{\mathrm{opt}} + \sqrt{k}$ from its corresponding point b. For $d > 2$, t_{approx} maps a_0 to a point in $b'_0 \in B'$ whose distance is at most $(1+\varepsilon)$ times the distance to the real nearest neighbor of a_0, thus the distance between a_0 and b'_0 is $(1+\varepsilon)h_{\mathrm{opt}}$. The same argument as before shows that $\vec{\delta}_{\mathrm{H}}(A + t_{\mathrm{approx}}, B) \leq (2+\varepsilon) * h_{\mathrm{opt}} + \sqrt{k}\gamma$.