Approximate Minimum-Weight Matching with Outliers Under Translation

Pankaj K. Agarwal[1]

Department of Computer Science, Duke University, Durham, NC 27708, USA
pankaj@cs.duke.edu

Haim Kaplan

School of Computer Science, Tel Aviv University, Tel Aviv 69978, Israel
haimk@tau.ac.il

Geva Kipper

School of Computer Science, Tel Aviv University, Tel Aviv 69978, Israel
gevakip@gmail.com

Wolfgang Mulzer[2]

Institut für Informatik, Freie Universität Berlin, 14195 Berlin, Germany
mulzer@inf.fu-berlin.de
🆔 https://orcid.org/0000-0002-1948-5840

Günter Rote

Institut für Informatik, Freie Universität Berlin, 14195 Berlin, Germany
rote@inf.fu-berlin.de
🆔 https://orcid.org/0000-0002-0351-5945

Micha Sharir[3]

School of Computer Science, Tel Aviv University, Tel Aviv 69978, Israel
michas@tau.ac.il

Allen Xiao[4]

Department of Computer Science, Duke University, Durham, NC 27708, USA
axiao@cs.duke.edu

Abstract

Our goal is to compare two planar point sets by finding subsets of a given size such that a minimum-weight matching between them has the smallest weight. This can be done by a translation of one set that minimizes the weight of the matching. We give efficient algorithms (a) for finding approximately optimal matchings, when the cost of a matching is the L_p-norm of the tuple of the Euclidean distances between the pairs of matched points, for any $p \in [1, \infty]$, and (b) for constructing small-size approximate minimization (or matching) diagrams: partitions of the translation space into regions, together with an approximate optimal matching for each region.

2012 ACM Subject Classification Mathematics of computing → Combinatorics

■ **Figure 1** Two sets A and B, and a matching of size $k = 6$ after translation.

1   Introduction

The following problem arises in pattern matching: given point sets A, B, with $|A| = m$ and $|B| = n$, and $k \leq \min\{m, n\}$, find subsets $A' \subseteq A$ and $B' \subseteq B$ with $|A'| = |B'| = k$ and a transformation R that matches $R(A)$ and B as closely as possible, see Figure 1. We think of A as a collection of features, or interest points of some pattern, that we want to match, bijectively, with similar features in a large image B. Moreover, since the coordinate frames for A and B are not necessarily aligned, we want to transform A to get the best possible fit.

This problem comes in many variants, depending on the class of permissible transformations R and on the similarity measure for the match. Here, we want to match A' and B' in a one-to-one manner, where the cost of a matching depends on the distances between matched points. Moreover, we only consider translations as permissible transformations, and write $A + t$ for the set A translated by a vector $t \in \mathbb{R}^2$. A feasible solution is given by a translation $t \in \mathbb{R}^2$ and by a *matching $M \subset A \times B$* of size k (in short, a *k-matching*): a set of k pairs $(a, b) \in A \times B$ so that any point $a \in A$ or $b \in B$ occurs in at most one pair. The parameter k is part of the input. We consider the L_p-cost of such a solution, for some $p \in [1, \infty]$:

$$\text{cost}_p(M, t) = \text{cost}(M, t) := \begin{cases} \left[\frac{1}{k} \sum_{(a,b) \in M} \|a + t - b\|^p \right]^{1/p} & \text{for finite } p, \\ \max_{(a,b) \in M} \|a + t - b\| & \text{for } p = \infty. \end{cases} \quad (1)$$

We will regard p as a fixed constant and will omit it from the notation. Noteworthy special cases arise when $p = 1$ (sum of distances, minimum-weight Euclidean matching), $p = 2$ (root-mean-square matching, in short RMS matching), and $p = \infty$ (bottleneck matching). In (1), we always measure the distances $\|a + t - b\|$ by the Euclidean norm. It is not hard to extend the treatment to other norms, but we stick with Euclidean distances for simplicity.

One important special case occurs when we have a small point set A (the *pattern*) that we want to locate within a larger set B (the *image*), and $k = |A| < |B|$. This problem was considered for $p = 2$ by Rote [11] and in subsequent work [3, 8], under the name *RMS partial matching*. Another important instance has $|A| \approx |B|$ and k slightly smaller than $|A|, |B|$. Now, we want to discard a few *outliers* from each set, to allow for some erroneous data.

For a fixed translation vector $t \in \mathbb{R}^2$, we define $\text{cost}^*(t) = \min_M \text{cost}(M, t)$ to be the cost of the minimum-cost k matching between $A + t$ and B. We set $M_t = \arg\min_M \text{cost}(M, t)$ to be an optimal matching from $A + t$ to B, i.e., $\text{cost}^*(t) = \text{cost}(M_t, t)$.

Let Π be the set of all k-matchings from A into B. The function cost^* is the *lower envelope* (i.e., the pointwise minimum) of the set of functions $F = \{t \mapsto \text{cost}(M, t) \mid M \in \Pi\}$. The vertical projection of this lower envelope induces a planar subdivision, called the *minimization diagram* of F. It is denoted by $\mathcal{M} := \mathcal{M}(A, B)$. Each face σ of $\mathcal{M}$ is a maximal connected set of points t for which $\text{cost}^*(t)$ is realized by the same matching M_σ. The *combinatorial complexity* of $\mathcal{M}$ is the number of its faces. We refer to $\mathcal{M}$ as the *(k-)matching diagram* of A and B. We are interested in two questions:

(P1) Compute $t^* = \arg\min_t \text{cost}^*(t)$ and $M^* := M_{t^*}$.

(P2) What is the combinatorial complexity of $\mathcal{M}(A, B)$, and how quickly can it be computed?

These questions have been studied, $p = 2$, by Rote [11] and by Ben-Avraham et al. [3]. Two challenging, still open problems are whether the size of $\mathcal{M}$ is polynomial in both m and n, and whether t^* and M^* can be computed in polynomial time. These previous works have raised the questions only for the case $p = 2$, but they are open for arbitrary $p < \infty$. There is extensive work on pattern matching and on computing similarity between two point sets. We refer the reader to [2, 15] for surveys. Here, we confine ourselves to a brief discussion of work directly related to the problem at hand.

Much work has been done on computing a minimum-cost *perfect matching* in geometric settings. Here, $n = |A| = m = |B| = k$. A minimum-cost perfect matching, for any L_p-norm, can be found in $\widetilde{O}(n^2)$ time [1, 9, 10].[5] These algorithms are based on the Hungarian algorithm for a minimum-cost maximum matching in a bipartite graph, and are made more efficient than the general technique by using certain efficient geometric data structures. Thus, they also work when the two point sets A and B have different sizes, say, $\lfloor A \rfloor = n$ and $|B| = m$, with $k = m \leq n$. In this case, the running time of the algorithm is $\widetilde{O}(mn)$.

Approximation algorithms for the minimum-weight perfect matching in geometric settings have been developed in a series of papers; see, e.g., [12] and the references therein. For the case when the weight of a matching is the sum of the Euclidean lengths of its edges, a near-linear algorithm is known [12]. If the weight is the L_p-norm of the Euclidean lengths of the edges, for some $p > 1$, then the best known algorithm runs in $\widetilde{O}(n^{3/2})$ time [13, 14]. In particular, for RMS matching ($p = 2$) and for $p = 1, \infty$, the time for finding a $(1 + \varepsilon)$-approximate optimal matching is $\widetilde{O}(n^{3/2})$, and for a general p, the running time is $\widetilde{O}\left(\frac{n^{3/2}}{\varepsilon^{3/2}}\right)$. These algorithms use the scaling method by Gabow and Tarjan [6] that at each scale computes a minimum-weight matching by finding n augmenting paths in $O(\sqrt{n})$ phases, where each phase takes $\widetilde{O}(n)$ time (see also [7]). If $|A| = n$, $|B| = m$, and $k = m \leq n$, then the m augmenting paths can be found in $O(\sqrt{m})$ phases, each of which takes $\widetilde{O}(n)$ time. Hence, the total running time in this case is $\widetilde{O}(\sqrt{m}n)$, for $p = 1, 2, \infty$, or $\widetilde{O}(\sqrt{m}n/\varepsilon^{3/2})$, for general p. When $k \leq m \leq n$, the minimum-weight k-matching is constructed, using the geometrically enhanced version

---

[5] The notation $\widetilde{O}(\cdot)$ hides polylogarithmic factors in n, m, and also polylogarithmic factors in $1/\varepsilon$, when we only seek a $(1 + \varepsilon)$-approximate solution.

of the Hungarian algorithm, in k augmenting steps, each of which can be performed in $O(n \operatorname{polylog}(n))$ time. That is, the exact minimum-weight k-matching can be computed in $\widetilde{O}(kn)$ time. The case of computing an approximate k-matching is somewhat trickier. If $k = \Theta(m)$, one can show, adapting the technique in [13], that the running time remains $O(\sqrt{mn} \operatorname{polylog}(n))$. For smaller values of k, one can still get a bound depending on k, but we do not treat this case in the paper. It is also much less motivated from the point of view of applications.

Cabello et al. [4] considered optimal shape matching under translations and/or rotations. They considered the more general setting of *weighted* point sets, where each point of A and B comes with a multiplicity or "weight". Accordingly, the similarity criterion is the *earth-mover's distance*, or *transportation distance*, which measures the minimum amount of work necessary to transport all the weight from A to B, where transporting a weight w by distance δ costs $w \cdot \delta$. For the special case of unit weights, this reduces, via the integrality of the minimum-cost flows, to one-to-one matching.

We apply several ideas from Cabello et al.'s paper: (1) the use of *point-to-point translations* to get constant-factor approximations, (2) the selection of a random subset of these transformations to get fast Monte Carlo algorithms, and (3) tiling the vicinity of these transformations in the parameter space by an ε-grid to get $(1 + \varepsilon)$-approximations. We go beyond the results of Cabello et al. in the following aspects.

- We give a greedy "disk-eating" algorithm in the space of translations to get an improved deterministic approximation (Theorem 4.5). This idea could be useful for other problems.
- We introduce *approximate matching diagrams*: Such a diagram is a subdivision of the translation plane together with a matching for each cell. This matching is approximately optimal for every translation in the cell. As a consequence, this diagram provides approximate optimal matchings for *all* translations. We show that there is an approximate matching diagram of small size, and we describe how to compute it efficiently (Section 2.1).
- Less importantly, our results cover a broader class of similarity measures: The lengths of the k matching edges can be aggregated in the objective function using any L_p norm, $p \geq 1$, whereas Cabello et al. only dealt with the L_1 norm. By indentifying the crucial property that lies at the basis of the approximation, namely Lipschitz continuity (Corollary 2.2), this generalization comes without much additional effort. Our results are also slightly more general because we allow outliers (i.e., $k < \min\{m, n\}$), whereas Cabello et al. match the smaller set completely.
- By using better data structures, some of our algorithms are more efficient.

We present approximate solutions for (P1) and (P2). They use approximation algorithms for matching between *stationary* sets as a black box. We write $W(m, n, k, \varepsilon)$ for the time that is needed to compute a $(1 + \varepsilon)$-approximate minimum-weight matching of size k between two given (stationary) sets A and B of m and n points in the plane, where the weight is the L_p-norm of the vector or Euclidean edge lengths, for $k \leq \min\{m, n\}$ and for a given $\varepsilon \geq 0$. We abbreviate $W(m, n, k, 0)$ as simply $W(m, n, k)$. Table 1 summarizes the known running times. We obtain two main results:

(i) We present an $\widetilde{O}(mn + \frac{mn}{\varepsilon^2 k} W(m, n, k, \varepsilon/2))$-time algorithm for computing a translation vector $\tilde{t}$ and a k-matching $\widetilde{M}$ between A and B such that $\operatorname{cost}(\widetilde{M}, \tilde{t}) \leq (1 + \varepsilon) \operatorname{cost}^*(t^*)$.

(ii) We present an $\widetilde{O}(mn + \frac{mn}{\varepsilon^2 k} W(m, n, k, \varepsilon/2))$-time algorithm for computing a $(1 + \varepsilon)$-approximate matching diagram of size $O\left(\frac{n}{\varepsilon^2} \log \frac{1}{\varepsilon}\right)$, i.e., a planar subdivision $\widetilde{\mathcal{M}}$ and a collection of k-matchings M_σ, one matching for each face σ of $\widetilde{\mathcal{M}}$, such that for each face σ of $\widetilde{\mathcal{M}}$ and for every $t \in \sigma$, $\operatorname{cost}(M_\sigma, t) \leq (1 + \varepsilon) \operatorname{cost}^*(t)$.

▪ **Table 1** Known time bounds for various matching problems between stationary sets. We assume $m \le n$, and in the last two rows $k = \Theta(m)$.

norm		time	reference
$p \in [1, \infty]$	exact	$W(m, n, k) = \widetilde{O}(kn)$	Hungarian method, geometric version [1, 9, 10]
$p \in \{1, 2, \infty\}$	$(1+\varepsilon)$-approximate	$W(m, n, k, \varepsilon) = \widetilde{O}(\sqrt{m}n)$	[13]
$p \in [1, \infty]$	$(1+\varepsilon)$-approximate	$W(m, n, k, \varepsilon) = \widetilde{O}(\sqrt{m}n/\varepsilon^{3/2})$	[14]

The paper is organized as follows. We start with simple solutions to (P1) and (P2) with constant-factor approximations (Section 2). We then refine them to obtain $(1+\varepsilon)$-approximate solutions, in Section 3. Finally, we present improved algorithms, which attain the bounds claimed in (i) and (ii), in Section 4. All our statements hold for $p = \infty$. In some cases, the proofs require a special treatment for this case, but for brevity, we will mostly omit the treatment for $p = \infty$.

2   Simple Constant-Factor Approximations

The following lemma establishes a Lipschitz condition for the cost of a matching of size k.

▶ **Lemma 2.1.** *Let $M \subset A \times B$ be a matching of size k, and let $t, \Delta \in \mathbb{R}^2$ be two translation vectors. Then, for any $p \in [1, \infty]$, the cost under the L_p-norm satisfies*

$$\text{cost}(M, t + \Delta) \le \text{cost}(M, t) + \|\Delta\|. \tag{2}$$

Proof. Let $M = \{(a_1, b_1), \ldots, (a_k, b_k)\}$, and define two nonnegative k-dimensional vectors $\vec{v}$ and $\vec{w}$ by $\vec{v}_i = \|a_i + t - b_i\|$ and $\vec{w}_i = \|a_i + t + \Delta - b_i\|$, for $1 \le i \le k$. By the triangle inequality for the Euclidean norm, we have, for each i, $\vec{w}_i = \|a_i + t + \Delta - b_i\| \le \|a_i + t - b_i\| + \|\Delta\| = \vec{v}_i + \|\Delta\|$. Thus, we obtain the component-wise inequality $\vec{w} \le \vec{v} + \|\Delta\| \cdot \vec{1}$, where $\vec{1}$ denotes the k-dimensional vector in which all components are 1. Now,

$$\text{cost}(M, t + \Delta) = \frac{\|\vec{w}\|_p}{k^{1/p}} \le \frac{\left\|\vec{v} + \|\Delta\| \cdot \vec{1}\right\|_p}{k^{1/p}} \le \frac{\|\vec{v}\|_p}{k^{1/p}} + \|\Delta\| \cdot \frac{\|\vec{1}\|_p}{k^{1/p}} = \text{cost}(M, t) + \|\Delta\|,$$

using the definition (1) of cost, the fact that the L_p-norm is a monotone function in the components whenever they are nonnegative, and the triangle inequality for the L_p-norm.   ◀

Here is an immediate corollary of Lemma 2.1:

▶ **Corollary 2.2** (Lipschitz continuity of the optimal cost). *For any two translation vectors $t_1, t_2 \in \mathbb{R}^2$, $\text{cost}^*(t_2) \le \text{cost}^*(t_1) + \|t_2 - t_1\|$.*

Proof. For the respective optimal k-matchings M_1 and M_2 between $A + t_1$ and B and $A + t_2$ and B,

$$\text{cost}^*(t_2) = \text{cost}(M_2, t_2) \le \text{cost}(M_1, t_2) \le \text{cost}(M_1, t_1) + \|t_2 - t_1\| = \text{cost}^*(t_1) + \|t_2 - t_1\|. ◀$$

Approximating t^* by point-to-point translations.   As in [4], we consider the set $T = \{b - a \mid a \in A, b \in B\}$ of at most *mn point-to-point translations* where some point in A is moved to some point in B. The following simple observation turns out to be very useful:

▶ **Lemma 2.3** ([4, Observation 1]). *Let $t \in \mathbb{R}^2$ be an arbitrary translation vector, and let $t_0 \in T$ be the nearest neighbor of t in T. Then $\mathrm{cost}^*(t) \geq \|t - t_0\|$.*

Proof. By definition, $t_0 = b - a$ is the translation in T with $\|t - t_0\| = \min_{(a',b') \in A \times B} \|t - b' + a'\|$. Thus, for $p < \infty$, all summands in the definition (1) of $\mathrm{cost}^*(t)$ are at least $\|t - t_0\|$, implying $\mathrm{cost}^*(t) \geq \|t - t_0\|$. The last conclusion is trivially valid for $p = \infty$ as well.   ◀

▶ **Lemma 2.4** ([4, Lemma 1]). *There is a translation $t_0 \in T$ with $\mathrm{cost}^*(t_0) \leq 2\,\mathrm{cost}^*(t^*)$.*

Proof. Let t^* be an optimal translation and M^* a corresponding matching of size k. Take the translation $\Delta = b - a - t^* \in \mathbb{R}^2$ for which $\|a + t^* - b\|$ is minimized, over $(a, b) \in M^*$. By Lemma 2.3, $\|\Delta\| \leq \mathrm{cost}^*(t^*)$. The claim now follows from Lipschitz continuity (Corollary 2.2) with $t_1 = t^*$ and $t_2 = t^* + \Delta$, where the latter translation is the desired $t_0 \in T$.   ◀

We remark that for RMS matching ($p = 2$), the factor 2 in the lemma can be improved to $\sqrt{2}$. Lemma 2.4 leads to the following simple algorithm for 2-approximating the optimum matching. Compute T, and iterate over its elements. For each $t_0 \in T$ compute $\mathrm{cost}^*(t_0)$ (exactly), and return the matching with the minimum weight, in $O(mnW(m, n, k))$ time.

If we are willing to tolerate a slightly larger approximation factor, we can compute, for any $\delta > 0$ and for each $t_0 \in T$, a $(1 + \delta)$-approximate matching, resulting in a $2(1 + \delta)$-approximation. This approach has overall running time $O(mnW(m, n, k, \delta))$.

▶ **Theorem 2.5.** *Let $A, B \subset \mathbb{R}^2$, with $|A| = m$ and $|B| = n$, $m \leq n$, and let $k \leq m$ be a size parameter. A translation vector $\tilde{t} \in \mathbb{R}^2$ can be computed in $O(mnW(m, n, k))$ time, such that $\mathrm{cost}^*(\tilde{t}) \leq 2\,\mathrm{cost}^*(t^*)$, where t^* is the optimum translation. Alternatively, for any constant $\delta > 0$, one can compute a translation vector $\tilde{t} \in \mathbb{R}^2$ and a k-matching $\widetilde{M}$ between A and B, in $O(mnW(m, n, k, \delta))$ time, such that $\mathrm{cost}(\widetilde{M}, \tilde{t}) \leq 2(1 + \delta)\,\mathrm{cost}^*(t^*)$.*

2.1   An Approximate Matching Diagram

We construct a planar subdivision $\widetilde{\mathcal{M}}$ that approximates the matching diagram $\mathcal{M}$ up to factor 3. This means that, for each face σ of $\widetilde{\mathcal{M}}$, there is a single matching M_σ (that we provide) so that, for each $t \in \sigma$, we have $\mathrm{cost}^*(t) \leq \mathrm{cost}(M_\sigma, t) \leq 3\,\mathrm{cost}^*(t)$.

We need a lemma that relates the best matching for a given translation t to the closest translation in T.

▶ **Lemma 2.6.** *Let t be an arbitrary translation, and let $t_0 \in T$ be its nearest neighbor in T, i.e., the translation in T that minimizes the length of $\Delta = t_0 - t$. Then,*

$$\mathrm{cost}^*(t) \leq \mathrm{cost}(M_{t_0}, t) \leq 3\,\mathrm{cost}^*(t). \tag{3}$$

(Recall that M_{t_0} denotes the optimal matching for t_0.)

Proof. Since M_{t_0} is a k-matching between A and B, we have, by definition, $\mathrm{cost}^*(t) \leq \mathrm{cost}(M_{t_0}, t)$. We prove the second inequality. By Corollary 2.2, $\mathrm{cost}^*(t_0) \leq \mathrm{cost}^*(t) + \|\Delta\|$, and by Lemma 2.3, $\|\Delta\| \leq \mathrm{cost}^*(t)$. Applying Lemma 2.1, we obtain

$$\begin{aligned}
\mathrm{cost}(M_{t_0}, t) &\leq \mathrm{cost}(M_{t_0}, t_0) + \|t - t_0\| = \mathrm{cost}^*(t_0) + \|\Delta\| \\
&\leq \mathrm{cost}^*(t) + 2\|\Delta\| \leq \mathrm{cost}^*(t) + 2\,\mathrm{cost}^*(t) = 3\,\mathrm{cost}^*(t).
\end{aligned}$$ ◀

Our approximate map $\widetilde{\mathcal{M}}$ is simply the Voronoi diagram $\mathrm{VD}(T)$, where each cell $\mathrm{VC}(t_0)$, for $t_0 \in T$, is associated with the optimal matching M_{t_0} at t_0. Correctness follows immediately from Lemma 2.6. Since the complexity of $\mathrm{VD}(T)$ is $O(|T|) = O(mn)$, we have a diagram

of complexity $O(mn)$. For each point $t_0 \in T$, we can either directly compute an optimal k-matching between $A + t_0$ and B and associate the resulting map with $\text{VC}(t_0)$, or use the $(1 + \delta)$-approximation algorithm of [13]. In the former case, $\text{VD}(T)$ is a 3-approximate matching diagram, and in the latter case it is a $3(1 + \delta)$-approximate matching diagram. We thus conclude the following:

▶ **Theorem 2.7.** *Let $A, B \subset \mathbb{R}^2$, with $|A| = m$ and $|B| = n$, $m \leq n$, and let $k \leq m$ be a size parameter. There is a 3-approximate k-matching diagram of A and B of size $O(mn)$, and it (and the matchings in each cell) can be computed in $O(mnW(m, n, k))$ time. Alternatively, a $3(1 + \delta)$-approximate matching diagram, for constant $\delta > 0$, of size $O(mn)$ can be computed, using the same planar decomposition, in $O(mnW(m, n, k, \delta))$ time.*

For $p = 2$, there is an alternative, potentially better approximating, construction. For each $t \in T$, define the function $f_t(s) := \text{cost}(M_t, s)$, and set $F = \{f_t \mid t \in T\}$. We let $\widetilde{\mathcal{M}}_0$ be the minimization diagram of the functions in F. Simple algebraic manipulations, similar to those for Euclidean Voronoi diagrams, show that $\widetilde{\mathcal{M}}_0$ is the minimization diagram of a set of $|T| \leq mn$ linear functions, namely, the functions $\tilde{f}_t(s) = 2 \sum_{(a,b) \in M_t} \langle a - b, s \rangle + \sum_{(a,b) \in M_t} \|a - b\|^2$, for $t \in T$. The resulting map $\widetilde{\mathcal{M}}_0$ is a 3-approximate diagram of complexity $O(mn)$. To see this, consider a Voronoi cell $\text{VC}(t_0)$ in $\widetilde{\mathcal{M}}$. We divide it into subcells in $\widetilde{\mathcal{M}}_0$, each associated with some matching. All these matchings, other than M_{t_0}, have smaller weights than the matching computed for t_0, over their respective subcells. Note that this subdivision is only used for the analysis, the algorithm outputs the original minimization diagram. We emphasize that this construction works only for $p = 2$, while the Voronoi diagram applies for any $p \in [1, \infty]$.

For $p = 2$, using the fact that the Euclidean norm is derived from a scalar product, we can improve the constant factors in Lemma 2.4 and Lemma 2.6. However, we chose to present the more general results, since they are simpler and since we derive a more powerful approximation below anyway.

## 3   Improved Approximation Algorithms

Computing a $(1 + \varepsilon)$-approximation of the optimum matching.   This algorithm uses the same technique that was used by Cabello et al. [4, Section 4.1, Theorem 6] in a slightly different setting. We include the description of this algorithm as a preparation for the approximate minimization diagram, and for the improved solutions in the following section.

Let t^* be the optimum translation, as above. Our goal is to compute a translation $\tilde{t}$ and a matching $\widetilde{M}$ so that $\text{cost}(\widetilde{M}, \tilde{t}) \leq (1 + \varepsilon) \text{cost}^*(t^*)$.

Suppose we know the translation $t_0 \in T$ that minimizes the length of $\Delta = t_0 - t^*$. By Lemma 2.3 and Lipschitz continuity (Corollary 2.2), $\|\Delta\| \leq \text{cost}^*(t^*) \leq \text{cost}^*(t_0) \leq \text{cost}^*(t^*) + \|\Delta\| \leq 2 \text{cost}^*(t^*)$. Using Theorem 2.5 with $\delta = 1/2$, we compute a 3-approximation for $\text{cost}^*(t^*)$, This allows us to choose some radius r_0 with $2 \text{cost}^*(t^*) \leq r_0 \leq 6 \text{cost}^*(t^*)$. We take the disk D_0 of radius r_0 centered at t_0, and we tile it with the vertices of a square grid of side-length $\delta := \frac{\varepsilon\sqrt{2}}{18} r_0 \leq \frac{\varepsilon\sqrt{2}}{3} \text{cost}^*(t^*)$. We define G_0 as the set of vertices of all grid cells that lie in D_0 or that overlap D_0 at least partially. G_0 contains $O(r_0/\delta)^2 = O(1/\varepsilon^2)$ vertices.

We compute, by [13], a $(1 + \varepsilon/2)$-approximate minimum-weight matching at each translation in G_0 and return the one that achieves the smallest weight. Since t^* has distance at most $\delta/\sqrt{2}$ from some grid vertex $g \in G_0$, we have, again by Lipschitz continuity (Corollary 2.2),

$$\text{cost}^*(g) \leq \text{cost}^*(t^*) + \frac{\delta}{\sqrt{2}} \leq \text{cost}^*(t^*) + \frac{\varepsilon}{3} \text{cost}^*(t^*) \leq \left(1 + \frac{\varepsilon}{3}\right) \text{cost}^*(t^*).$$
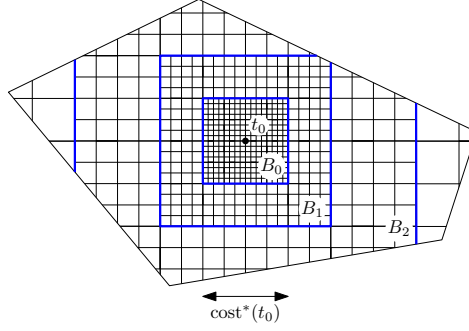
Figure 2 Partition of a Voronoi cell into nested grids, for the (unrealistically large) choice $\varepsilon = 1/2$.

Since we compute a $(1 + \varepsilon/2)$-approximate matching for each grid point, the best computed matching has cost at most $(1 + \varepsilon/3)(1 + \varepsilon/2) \operatorname{cost}^*(t^*) \leq (1 + \varepsilon) \operatorname{cost}^*(t^*)$, assuming $\varepsilon \leq 1$.

Since we do not know t_0, we apply this procedure to all mn translations of T, for a total of $O(mn/\varepsilon^2)$ approximate matching calculations for fixed sets.

▶ **Theorem 3.1.** *Let $A, B \subseteq \mathbb{R}^2$, $|A| = m \leq |B| = n$, and let $k \leq m$ be a size parameter and $0 < \varepsilon \leq 1$ a constant. A translation vector $\tilde{t} \in \mathbb{R}^2$ and a matching $\widetilde{M}$ of size k between A and B can be computed in $O\left(\frac{mn}{\varepsilon^2} \cdot W(m, n, k, \frac{\varepsilon}{2})\right)$ time, such that $\operatorname{cost}(\widetilde{M}, \tilde{t}) \leq (1 + \varepsilon) \operatorname{cost}^*(t^*)$.*

Cabello et al. [4, Theorem 4] give an $O\left(\frac{n^3 m}{\varepsilon^4} \log^2 n\right)$-time algorithm for the weighted problem, which includes the matching problem with $k = m \leq n$ as a special case. It follows the same technique: it solves $O(mn/\varepsilon^2)$ problems, each with a fixed translation, but each such problem takes longer than in our case because it uses the earth mover's distance.

A $(1+\varepsilon)$-approximation of $\mathcal{M}$.   We now construct a $(1+\varepsilon)$-approximate matching diagram $\widetilde{\mathcal{M}}$ of A and B by refining $\operatorname{VD}(T)$. Without loss of generality, we assume that $\varepsilon = 2^{-\alpha}$, for some natural number α, and we set $u := \log_2(1/\varepsilon) + 2 = \alpha + 2$. We subdivide each Voronoi cell of $\operatorname{VD}(T)$ into smaller subcells, as follows. Fix $t_0 \in T$. For $i = 0, \ldots, u$, let B_i be the square of side-length $2^i \operatorname{cost}^*(t_0)$, centered at t_0. Set $B_{-1} = \emptyset$. For $i = 0, \ldots, u$, we partition $B_i \setminus B_{i-1}$ into a uniform grid with side-length $\varepsilon 2^{i-3} \operatorname{cost}^*(t_0)$. We clip each grid cell τ to $\operatorname{VC}(t_0)$, i.e., if $\tau \cap \operatorname{VC}(t_0) \neq \emptyset$, we take $\tau \cap \operatorname{VC}(t_0)$ as a face of $\widetilde{\mathcal{M}}$. Let t_τ be the center of the grid cell τ. We associate $M_\tau := M_{t_\tau}$ with the face $\tau \cap \operatorname{VC}(t_0)$. Finally, each connected component of $\operatorname{VC}(t_0) \setminus B_u$ becomes a (possibly non-convex) face of $\widetilde{\mathcal{M}}$. There are at most four such faces, and we associate M_{t_0} with each of them.

The above procedure partitions $\operatorname{VC}(t_0)$ into $O(\frac{1}{\varepsilon^2} \log \frac{1}{\varepsilon})$ cells, and their total complexity is $O(k_0 + \frac{1}{\varepsilon^2} \log \frac{1}{\varepsilon})$, where k_0 is the number of vertices on the boundary of $\operatorname{VC}(t_0)$. We repeat our procedure for all Voronoi cells of $\operatorname{VD}(T)$. Since the total complexity of $\operatorname{VD}(T)$ is $O(mn)$, the total complexity of $\widetilde{\mathcal{M}}$ is $O(\frac{mn}{\varepsilon^2} \log \frac{1}{\varepsilon})$.

▶ **Lemma 3.2.** *$\widetilde{\mathcal{M}}$ is a $(1 + \varepsilon)$-approximate matching diagram of A and B.*

Proof. Let $t \in \mathbb{R}^2$ be an arbitrary translation vector, and let $t_0 \in T$ be the nearest neighbor of t in T, i.e., $t \in \operatorname{VC}(t_0)$. First, consider the case when $t \notin B_u$. Then $\|t - t_0\| \geq 2 \operatorname{cost}^*(t_0)/\varepsilon$ and M_{t_0} is the matching associated with the cell of $\widetilde{\mathcal{M}}$ containing t. Hence, using Lemmas 2.1 and 2.3, we obtain

$$\operatorname{cost}^*(t) \leq \operatorname{cost}(M_{t_0}, t) \leq \operatorname{cost}^*(t_0) + \|t - t_0\| \leq \left(1 + \frac{\varepsilon}{2}\right) \|t - t_0\| \leq \left(1 + \frac{\varepsilon}{2}\right) \operatorname{cost}^*(t).$$

Suppose $t \in B_0$. Then $\|t - t_0\| \leq \mathrm{cost}^*(t_0)/\sqrt{2}$. Therefore, by Corollary 2.2,

$$\mathrm{cost}^*(t) \geq \mathrm{cost}^*(t_0) - \|t - t_0\| \geq \mathrm{cost}^*(t_0) - \frac{1}{\sqrt{2}}\mathrm{cost}^*(t_0) = \left(1 - \frac{1}{\sqrt{2}}\right)\mathrm{cost}^*(t_0).$$

Let τ be the grid cell inside B_0 containing t, and let t_τ be the center of τ. Then $\|t - t_\tau\| \leq \frac{\varepsilon}{8\sqrt{2}}\mathrm{cost}^*(t_0)$. By Corollary 2.2, $\mathrm{cost}^*(t_\tau) \leq \mathrm{cost}^*(t) + \|t - t_\tau\|$. Furthermore,

$$\begin{aligned}
\mathrm{cost}(M_\tau, t) &\leq \mathrm{cost}(M_\tau, t_\tau) + \|t - t_\tau\| = \mathrm{cost}^*(t_\tau) + \|t - t_\tau\| \\
&\leq \mathrm{cost}^*(t) + 2\|t - t_\tau\| \leq \mathrm{cost}^*(t) + \frac{\varepsilon}{4\sqrt{2}}\mathrm{cost}^*(t_0) \\
&\leq \mathrm{cost}^*(t) + \frac{\varepsilon}{4\sqrt{2}} \cdot \frac{\sqrt{2}}{\sqrt{2} - 1}\mathrm{cost}^*(t) \leq (1 + \varepsilon)\mathrm{cost}^*(t).
\end{aligned}$$

Finally, suppose $t \in B_i \setminus B_{i-1}$, for some $i \geq 1$. Since $t \notin B_{i-1}$, we have $\|t - t_0\| \geq 2^{i-2}\mathrm{cost}^*(t_0)$. Let τ be the grid cell of $B_i \setminus B_{i-1}$ containing t, and let t_τ be its center. Then $\|t - t_\tau\| \leq \frac{2^{i-3}}{\sqrt{2}}\varepsilon \cdot \mathrm{cost}^*(t_0)$. Starting with the inequality that was established above, we get

$$\begin{aligned}
\mathrm{cost}(M_\tau, t) &\leq \mathrm{cost}^*(t) + 2\|t - t_\tau\| \leq \mathrm{cost}^*(t) + 2\frac{2^{i-3}\varepsilon}{\sqrt{2}}\mathrm{cost}^*(t_0) \\
&\leq \mathrm{cost}^*(t) + \frac{\varepsilon}{\sqrt{2}}\|t - t_0\| \leq \mathrm{cost}^*(t) + \frac{\varepsilon}{\sqrt{2}}\mathrm{cost}^*(t) \leq (1 + \varepsilon)\mathrm{cost}^*(t). \quad \blacktriangleleft
\end{aligned}$$

Similar to the $O(1)$-approximate matching diagram, we can improve the construction time by setting $\varepsilon' = \varepsilon/3$ instead of ε and computing a $(1 + \varepsilon/2)$-approximate optimal matching (instead of the exact matching) for the center of every cell:

▶ **Theorem 3.3.** *Let $A, B \subseteq \mathbb{R}^2$, with $|A| = m$, $|B| = n$, $m \leq n$ and a size parameter $k \leq m$. For $0 < \varepsilon \leq 1$, one can compute a $(1 + \varepsilon)$-approximate k-matching diagram of A and B, of size $O(\frac{mn}{\varepsilon^2}\log\frac{1}{\varepsilon})$, in $O(\frac{mn}{\varepsilon^2}\log\frac{1}{\varepsilon})W(m, n, k, \frac{\varepsilon}{2})$ time.*

4   Improved Algorithms

We now present techniques that improve, by a factor of m or of k, both algorithms for computing an approximate optimal matching and an approximate matching diagram. These algorithms work well for the case $k \approx m$, and they deteriorate when k becomes small. The first algorithm is based on an idea of Cabello et al. [4, Lemma 2]: The best matching contains a substantial number of edges whose length does not exceed the optimum cost by more than a constant factor (cf. Lemma 4.1). This gives a randomized constant-factor approximation algorithm that requires $O(mn/k)$ approximate matching computations between stationary sets in order to succeed with probability $\frac{1}{2}$ (Theorem 4.2). We proceed to an improved algorithm that computes a constant-factor approximation with the same number of fixed-translation matching calculations *deterministically*. By tiling the vicinity of each candidate translation by an ε-grid, we then obtain a $(1 + \varepsilon)$-approximation (Theorem 4.5).

Markov's inequality bounds the number of items in a sample that are substantially above average. We will use the following consequence of it:

▶ **Lemma 4.1.** *Let M be a matching of size k between a (possibly translated) set A and a set B, with cost μ. Let $0 < c \leq 1$. Then the number of pairs $(a, b) \in M$ for which $\|a - b\| < (1 + c)\mu$ is at least $k - k/(1 + c)^p$.*

Proof. For $p = \infty$, we interpret $(1 + c)^p$ as ∞, and the result is obvious because $\|a - b\| < (1 + c)\mu$ for *all* pairs (a, b). For $1 \le p < \infty$, we argue by contradiction. The total number of pairs is k. If there were more than $k/(1 + c)^p$ pairs $(a, b) \in M$ with $\|a - b\| \ge (1 + c)\mu$, the total cost would be

$$\mu = \text{cost}(M) = \left[\tfrac{1}{k} \cdot \sum_{(a,b) \in M} \|a - b\|^p \right]^{1/p} > \left[\tfrac{1}{k} \cdot k/(1 + c)^p \cdot ((1 + c)\mu)^p \right]^{1/p} = \mu. \qquad \blacktriangleleft$$

Consider the optimal translation t^* and the corresponding optimal matching M^*. By the lemma, the fraction of the pairs $(a, b) \in M^*$ that satisfy $\|a + t^* - b\| \le (1 + c)\,\text{cost}^*(t^*)$ is at least $1 - 1/(1 + c)^p \ge 1 - 1/(e^{c/2})^p = 1 - e^{-cp/2}$, since $c \le 1$. Hence, with probability at least $(1 - e^{-cp/2})\frac{k}{m}$, a randomly chosen $a \in A$ will participate in such a "close" pair of M^*. We do not know the $b \in B$ with $(a, b) \in M^*$, so we try all n possibilities. That is, we choose a single random point $a_0 \in A$, and we try all n translations $b - a_0 \in T$, returning the minimum-weight partial matching over these translations. With probability at least $(1 - e^{-cp/2})\frac{k}{m}$, we get, by Lemma 2.6, a matching whose weight is at most $\text{cost}^*(t^*) + (1 + c)\,\text{cost}^*(t^*) = (2 + c)\,\text{cost}^*(t^*)$. The runtime of this procedure is $n \cdot W(m, n, k)$, or $n \cdot W(m, n, k, \delta)$ if we compute at each of the above translations t_0 a $(1+\delta)$-approximation to $\text{cost}^*(t_0)$. To boost the success probability, we repeat this drawing process s times and obtain a $(2 + c)(1 + \delta)$-approximation to the best matching, with probability at least $1 - \left(1 - (1 - e^{-cp/2})\frac{k}{m}\right)^s$. By setting $c = \delta = \varepsilon/4$, we get the following theorem.

▶ **Theorem 4.2.** *Let $A, B \subset \mathbb{R}^2$ with $|A| = m$ and $|B| = n$, $m \le n$, and let $k \le m$ and $s \ge 1$ be parameters. Then, a translation vector $\tilde{t} \in \mathbb{R}^2$ and a matching $\widetilde{M}$ of size k between A and B can be computed in $O(sn \cdot W(m, n, k, \varepsilon/4))$ time, such that $\text{cost}(\widetilde{M}, \tilde{t}) \le (2 + \varepsilon)\,\text{cost}^*(t^*)$ with probability at least $1 - \left(1 - (1 - e^{-\varepsilon p/8})\frac{k}{m}\right)^s$, for any ε with $0 < \varepsilon \le 1$.*

If εp is small, the probability is approximately equal to the simpler expression $1 - e^{-s \cdot \varepsilon p k/8m}$.

Cabello et al. [4] proceeded from this result to a $(1 + \varepsilon)$-approximation by tiling the vicinity of each selected translation with an ε-grid [4, Theorem 7]. We will first replace the randomized algorithm by a deterministic one, and apply the ε-grid refinement afterwards.

We now describe a deterministic algorithm for approximating t^* and the corresponding matching M^*. At a high level, the mn points of T are partitioned into $O(mn/k)$ clusters of size $\Omega(k)$, and one point, not necessarily from T, is chosen to represent each cluster. We will argue that the point in the resulting set X of representatives that is nearest to t^* yields a matching whose value at t^* is an $O(1)$-approximation of $\text{cost}^*(t^*)$.

Here is the main idea of how we cluster the points in T and construct X, in an incremental manner. In step i, we greedily choose the smallest disk D_i that contains $k/2$ points of T (or all of T, if $|T| \le k/2$), add the center of D_i to X, delete the points of $D_i \cap T$ from T, and repeat. Carmi et al. [5] have described an efficient algorithm for this clustering problem. It preprocesses T into a data structure (consisting of three compressed quadtrees) in $O(mn \log n)$ time, so that in step i, the disk D_i can be computed in $\widetilde{O}(k^2)$ time and $D_i \cap T$ can be deleted from the data structure in $\widetilde{O}(k^2)$ time, leading to an $\widetilde{O}(mnk)$-time algorithm. They also present a faster approximation algorithm for this clustering problem: in step i, instead of computing the smallest enclosing disk D_i, they show that a disk of radius at most twice that of D_i that still contains $k/2$ points of T can be computed in $\widetilde{O}(k)$ time, and that $D_i \cap T$ can be deleted in $\widetilde{O}(k)$ time, thereby improving the overall running time to $\widetilde{O}(mn)$. This approximation algorithm is sufficient for our purpose. We next give a more formal description of our method:

At the beginning of step i, we have a set $P_i \subseteq T$ and the current set X. Initially, $P_1 = T$ and $\mathsf{X} = \emptyset$. We preprocess P_1, in $\widetilde{O}(|T|) = \widetilde{O}(mn)$ time, into the data structure described by Carmi et al. [5]. We perform the following operations in step i: if $P_i = \emptyset$, the algorithm

terminates. If $0 < |P_i| \le k/2$, we compute the smallest disk D_i containing P_i. If $|P_i| > k/2$, then let ρ_i^* be the radius of the smallest disk that contains at least $k/2$ points of P_i. Using the algorithm in [5], we compute a disk D_i of radius $\rho_i \le 2\rho_i^*$ containing at least $k/2$ points of P_i. We add the center ξ_i of D_i to X, and we set $P_{i+1} := P_i \setminus D_i$. We remove $P_i \cap D_i$ from the data structure, as described in [5]. Let $\mathcal{D}$ be the set of disks computed by the above procedure. By construction, $\rho_i^* \le \rho_{i+1}^*$, $\rho_i \le 2\rho_i^* \le 2\rho_{i+1}^* \le 2\rho_{i+1}$, and $|\mathsf{X}| = |\mathcal{D}| \le 2mn/k$. The following two lemmas establish the correctness of our method.

▶ **Lemma 4.3.** *Let $t \in \mathbb{R}^2$ be a translation vector, and let ξ_0 be its nearest neighbor in X. Then $\|t - \xi_0\| \le 3 \cdot 2^{1/p} \operatorname{cost}^*(t)$.*

Proof. Let D be the disk of radius $2^{1/p} \operatorname{cost}^*(t)$ centered at t, and let $S = D \cap T$. By Lemma 4.1 with $1 + c = 2^{1/p}$, we have $|S| \ge k/2$. Let D_i be the first disk chosen by the above procedure that contains a point t_0 of S, so $S \subseteq P_i$. We must have $\rho_i^* \le 2^{1/p} \operatorname{cost}^*(t)$, because the smallest disk that contains at least $k/2$ points of P_i is not larger than D. Hence, $\rho_i \le 2 \cdot 2^{1/p} \operatorname{cost}^*(t)$, and

$$\|t - \xi_i\| \le \|t - t_0\| + \|t_0 - \xi_i\| \le 2^{1/p} \operatorname{cost}^*(t) + \rho_i$$
$$\le 2^{1/p} \operatorname{cost}^*(t) + 2 \cdot 2^{1/p} \operatorname{cost}^*(t) = 3 \cdot 2^{1/p} \operatorname{cost}^*(t). \qquad \blacktriangleleft$$

▶ **Lemma 4.4.** $\min_{\xi \in \mathsf{X}} \operatorname{cost}^*(\xi) \le (1 + 3 \cdot 2^{1/p}) \operatorname{cost}^*(t^*)$.

Proof. Let ξ_0 be the nearest neighbor to t^* in X. Applying Lemma 4.3 with $t = t^*$, we obtain $\|t^* - \xi_0\| \le 3 \cdot 2^{1/p} \operatorname{cost}^*(t^*)$. By Corollary 2.2, we then have $\operatorname{cost}^*(\xi_0) \le \operatorname{cost}^*(t^*) + \|t^* - \xi_0\| \le (1 + 3 \cdot 2^{1/p}) \operatorname{cost}^*(t^*)$. $\blacktriangleleft$

We fix a constant $\delta \in (0, 1]$. We compute a $(1 + \delta)$-approximate k-matching M_ξ between $A + \xi$ and B, for every $\xi \in \mathsf{X}$, and choose the best among them. This will give an $O(1)$-approximation of the minimum-cost k-matching under translation. We can extend this algorithm to yield a $(1 + \varepsilon)$-approximation algorithm following the same procedure as in Section 3: We draw a disk of radius $(1 + 3 \cdot 2^{1/p} + 4\varepsilon) \operatorname{cost}^*(t^*)$ around each point of X. We draw a uniform grid of cell size $O(\varepsilon)$ and look at all vertices t of grid cells that overlap one of these disks at least partially. We compute a $(1 + \varepsilon/2)$-approximation for the best matching of size k between $A + t$ and B for each of the grid point t under consideration, and we choose the best matching among them. Putting everything together, we obtain the following:

▶ **Theorem 4.5.** *Let $A, B \subset \mathbb{R}^2$, with $|A| = m$ and $|B| = n$, and let $0 < \varepsilon \le 1$ and $k \le \min\{m, n\}$ be parameters. Then, a translation vector $\tilde{t} \in \mathbb{R}^2$ and a matching $\widetilde{M}$ of size k between A and B can be computed in $\widetilde{O}(mn + \frac{mn}{\varepsilon^2 k} W(m, n, k, \frac{\varepsilon}{2}))$ time, such that $\operatorname{cost}(\widetilde{M}, \tilde{t}) \le (1 + \varepsilon) \operatorname{cost}^*(t^*)$.*

We show that $\mathrm{VD}(\mathsf{X})$ is indeed an $O(1)$-approximate matching diagram of A and B. This is analogous to Section 2.1 (Lemma 2.6).

▶ **Lemma 4.6.** *Let $t \in \mathbb{R}^2$ be a translation vector, and let ξ_0 be its nearest neighbor in X. Then, $\operatorname{cost}^*(t) \le \operatorname{cost}(M_{\xi_0}, t) \le (1 + 6 \cdot 2^{1/p}) \operatorname{cost}^*(t)$.*

Proof. Since M_{ξ_0} is a matching of size k between A and B, we have, by definition, $\operatorname{cost}^*(t) \le \operatorname{cost}(M_{\xi_0}, t)$. We now prove the second inequality. By Corollary 2.2, $\operatorname{cost}^*(\xi_0) \le \operatorname{cost}^*(t) + \|t - \xi_0\|$, Lemma 2.1, and Lemma 4.3,

$$\operatorname{cost}(M_{\xi_0}, t) \le \operatorname{cost}(M_{\xi_0}, \xi_0) + \|t - \xi_0\|$$
$$= \operatorname{cost}^*(\xi_0) + \|t - \xi_0\| \le \operatorname{cost}^*(t) + 2\|t - \xi_0\| \le (1 + 6 \cdot 2^{1/p}) \operatorname{cost}^*(t). \qquad \blacktriangleleft$$

The combinatorial complexity of VD(X) is $O(mn/k)$. We can now construct a $(1+\varepsilon)$-approximate matching diagram by refining each Voronoi cell of VD(X), as in Section 3, but the constants have to be chosen differently. The diagram has $O(\frac{mn}{k\varepsilon^2} \log \frac{1}{\varepsilon})$ cells, and we need $W(m, n, k, \frac{\varepsilon}{2})$ time per cell. We obtain the following:

▶ **Theorem 4.7.** *Let* $A, B \subset \mathbb{R}^2$, $|A| = m \le |B| = n$, *and let* $k \le m$, $\varepsilon \in (0, 1]$ *be parameters. There exists a* $(1+\varepsilon)$-*approximate* k-*matching diagram of* A *and* B *of size* $O(\frac{mn}{k\varepsilon^2} \log \frac{1}{\varepsilon})$, *and it can be computed in* $\widetilde{O}(mn) + O\big(\frac{mn}{k\varepsilon^2} \log \frac{1}{\varepsilon} W(m, n, k, \frac{\varepsilon}{2})\big)$ *time.*

For the case when $cm \le k \le (1-c)n$ for some constant $c > 0$, we can show that the bound in Theorem 4.7 on the size of the diagram is tight in the worst case in terms of m, n, and k (but not of ε): If A is a unit grid of size $\sqrt{m} \times \sqrt{m}$ and B is a unit grid of size $\sqrt{n} \times \sqrt{n}$, then there are $\Omega(n)$ translation vectors at which A and B are perfectly aligned and have at least k points in common. Thus, any $O(1)$-approximate matching diagram of A and B needs to have $\Omega(n)$ distinct faces.

References

1 Pankaj K. Agarwal, Alon Efrat, and Micha Sharir. Vertical decomposition of shallow levels in 3-dimensional arrangements and its applications. *SIAM J. Comput.*, 29(3):912–953, 1999.

2 Helmut Alt and Leonidas J. Guibas. Discrete geometric shapes: Matching, interpolation, and approximation. In J.R. Sack and J. Urrutia, editors, *Handbook of Comput. Geom.*, pages 121–153. Elsevier, Amsterdam, 1999.

3 Rinat Ben-Avraham, Matthias Henze, Rafel Jaume, Balázs Keszegh, Orit E. Raz, Micha Sharir, and Igor Tubis. Partial-Matching RMS Distance Under Translation: Combinatorics and Algorithms. *Algorithmica*, 80(8):2400–2421, 2018.

4 Sergio Cabello, Panos Giannopoulos, Christian Knauer, and Günter Rote. Matching point sets with respect to the Earth Mover's Distance. *Comput. Geom.*, 39(2):118–133, 2008.

5 Paz Carmi, Shlomi Dolev, Sariel Har-Peled, Matthew J. Katz, and Michael Segal. Geographic Quorum System Approximations. *Algorithmica*, 41(4):233–244, 2005.

6 Harold N. Gabow and Robert Endre Tarjan. Faster Scaling Algorithms for Network Problems. *SIAM J. Comput.*, 18(5):1013–1036, 1989.

7 Andrew V. Goldberg, Sagi Hed, Haim Kaplan, and Robert E. Tarjan. Minimum-Cost Flows in Unit-Capacity Networks. *Theory Comput. Syst.*, 61(4):987–1010, 2017.

8 Matthias Henze, Rafel Jaume, and Balázs Keszegh. On the complexity of the partial least-squares matching Voronoi diagram. In *Proc. 29th European Workshop Comput. Geom. (EWCG)*, pages 193–196, 2013.

9 Haim Kaplan, Wolfgang Mulzer, Liam Roditty, Paul Seiferth, and Micha Sharir. Dynamic Planar Voronoi Diagrams for General Distance Functions and their Algorithmic Applications. In *Proc. 28th Annu. ACM-SIAM Sympos. Discrete Algorithms (SODA)*, pages 2495–2504, 2017.

10 Jeff M. Phillips and Pankaj K. Agarwal. On Bipartite Matching under the RMS Distance. In *Proc. 18th Canad. Conf. Comput. Geom. (CCCG)*, pages 143–146, 2006.

11 Günter Rote. Partial least-squares point matching under translations. In *Proc. 26th European Workshop Comput. Geom. (EWCG)*, pages 249–251, 2010.

12 R. Sharathkumar and Pankaj K. Agarwal. A near-linear time ε-approximation algorithm for geometric bipartite matching. In *Proc. 44th Annu. ACM Sympos. Theory Comput. (STOC)*, pages 385–394, 2012.

13 R. Sharathkumar and Pankaj K. Agarwal. Algorithms for the transportation problem in geometric settings. In *Proc. 23rd Annu. ACM-SIAM Sympos. Discrete Algorithms (SODA)*, pages 306–317, 2012.

14 Kasturi R. Varadarajan and Pankaj K. Agarwal. Approximation Algorithms for Bipartite and Non-Bipartite Matching in the Plane. In *Proc. 10th Annu. ACM-SIAM Sympos. Discrete Algorithms (SODA)*, pages 805–814, 1999.

15 Remco C. Veltkamp. Shape matching: Similarity measures and algorithms. In *Proc. Intl. Conf. Shape Modeling and Applications*, pages 188–197, 2001.