

Approximate Congruence in Nearly Linear Time

Piotr Indyk *

Suresh Venkatasubramanian †

Abstract

The problem of geometric point set matching has been studied extensively in the domain of computational geometry, and has many applications in areas such as computer vision, computational chemistry, and pattern recognition. One of the commonly used metrics is the *bottleneck distance*, which for two point sets P and Q is the minimum over all one-to-one mappings $f : P \rightarrow Q$ of $\max_{p \in P} d(p, f(p))$, where d is the Euclidean distance. Much effort has gone into developing efficient algorithms for minimising the bottleneck distance between two point sets under groups of transformations. However, the algorithms that have thus far been developed suffer from running times that are large polynomials in the size of the input, even for approximate formulations of the problem.

In this paper we define a point set similarity measure that includes both the bottleneck distance and the *Hausdorff distance* as special cases. This measure relaxes the condition that the mapping must be one-to-one, but guarantees that only a few points are mapped to any point. Using a novel application of Hall's Theorem to reduce the geometric matching problem to a combinatorial matching problem, we present near-linear time approximation schemes for minimising this distance between two point sets in the plane under isometries; we note here that the best known algorithms for congruence under the bottleneck measure run in time $\tilde{O}(n^{2.5})$.

We also obtain a combinatorial bound on the *metric entropy* of certain families of geometric objects. This result yields improved algorithms for approximate congruence, and may be of independent interest.

1 Introduction

Geometric point set matching in two and three dimensions is a well-studied area of computational geometry, with application to fields such as computer vision [22], pattern recognition [8, 18] and computational chemistry [12, 13, 23]. Given some choice of a space \mathcal{G} of transformations (e.g. all isometries) and a distance measure $d(P, Q)$ for two point sets P and Q in d -dimensional Euclidean space, we can formulate the basic problem as follows:

Problem 1.1 (Congruence). *Given point sets P, Q and $\epsilon > 0$, determine the transformation $T \in \mathcal{G}$ that brings P close to Q , i.e. such that $d(T(P), Q) \leq \epsilon$.*

A natural measure that has been studied is the *bottleneck distance*, defined for point sets P and Q as the minimum value r over all one-to-one mappings $f : P \rightarrow Q$ of $\max_{p \in P} d(p, f(p))$, where d is the Euclidean distance; we call r the *bottleneck value*. Alt, Mehlhorn, Wagener, and Welzl [3] initiated a comprehensive study of point set matching under the bottleneck distance, proposing a suite of polynomial time algorithms. The running times of their algorithms are large; for example, for two-dimensional point-sets and a transformation space restricted to isometries (which we call a *noisy congruence problem*) the running time is $\tilde{O}(n^8)^1$, where $n = |P| = |Q|$ (although they do not mention it explicitly, their algorithms also work for $|P| = k < n = |Q|$). Even though the running time can be reduced to $\tilde{O}(n^7)$ [11], it is still quite high. Moreover, as noted in the survey by Alt and Guibas [2] (also in the paper by Goodrich, Mitchell and Orletsky [15]), these algorithms are likely to be “difficult to implement and numerically unstable due to the necessary computation of intersections of complex algebraic surfaces”. In order

*MIT Laboratory for Computer Science Cambridge, Massachusetts 02139-3594. Email: indyk@theory.lcs.mit.edu. Work partially done while at Stanford University, (supported by a Stanford Graduate Fellowship and NSF Award CCR-9357849, with matching funds from IBM, Mitsubishi, Schlumberger Foundation, Shell Foundation, and Xerox Corporation.

†AT&T Research, 180 Park Ave, Florham Park, NJ 07932. Email: suresh@research.att.com. Work partially done while at Stanford University (supported by ARO MURI Grant DAAH04-96-1-0007 and NSF Award CCR-9357849, with matching funds from IBM, Mitsubishi, Schlumberger Foundation, Shell Foundation, and Xerox Corporation).

¹ $\tilde{O}(f(n))$ denotes $O(f(n) \log^{O(1)} n)$.

to obtain faster and more practical algorithms several authors proposed algorithms for restricted cases (e.g. [5]) or resorted to *approximations*, i.e. algorithms which guarantee solutions with cost at most $r(1+\epsilon)$, for some $\epsilon > 0$. This line of research was initiated by Heffernan and Schirra [17] who gave an $\tilde{O}(n^{2.5})$ -time algorithm for approximate noisy congruence for the case when the noise regions are “small”. The latter assumption can be removed by using the techniques of Efrat and Itai [11] with no change in the asymptotic complexity. To our knowledge this is the best algorithm for solving the approximate noisy congruence problem.

In the meantime, many algorithms have been proposed for other measures, most notably the *Hausdorff distance*, for which f is not restricted to be one-to-one. It was observed in [11] and by others that this lack of restriction is unsuitable in many situations (since it might happen that many feature points of the pattern are associated with only one point from the image); on the other hand, it seems to reduce the complexity of the problem. In particular, the exact problem can be solved in $\tilde{O}(n^5)$ time [8]. Goodrich *et al.* gave an approximate algorithm running in $\tilde{O}(n^3)$ -time. Both algorithms in fact solve the pattern matching problem (i.e. allow $|P| < |Q|$); if we assume $|P| = |Q|$ then (as in [17]) the running time can be reduced to $\tilde{O}(n^2)$. Recently, Cardoze and Schulman [7] and Indyk, Motwani and Venkatasubramanian [19] proposed a new paradigm for geometric point set matching based on algebraic convolutions. This reduced the complexity of pattern matching to $\tilde{O}(n^2)$; using their techniques congruence can be solved within the (ultimate) $\tilde{O}(n)$ bound. Unfortunately, the one-to-one restriction imposed by bottleneck matching distance seems hard to accommodate within the rigid framework of algebraic convolutions.

In this paper, we define a point set distance measure, the *generalized bottleneck distance*, that includes as special cases both the bottleneck distance and the Hausdorff distance. More specifically, we allow transformations for which the corresponding function $f : P \rightarrow Q$ maps a small constant fraction of points in P to non-unique elements of Q ; moreover, any given point in Q is the image (under f) of only a constant (between 3 and 6) number of points in P .

The above formulation generalizes both the Hausdorff distance and the bottleneck distance. Moreover, it allows us to achieve a tradeoff between the quality of match (not too many points can be contracted to one) and the complexity of the matching procedures (which in many cases is almost linear).

All our results are based on the reduction of geometric matching to combinatorial pattern matching; in each case the output is a bit vector o of translations such that $o[t] = 1$ iff $t(P)$ and Q have a match with the desired properties. All running times that we report are for algorithms that output such a bit vector. The algorithms are Monte Carlo; the probability of error in each case is $O(1/n^c)$, $c \geq 1$.

1.1 Our Techniques

The existence of a bottleneck matching between two sets of points is a global criterion. In order to apply techniques from combinatorial pattern matching, we need to express this in terms of local criteria. In order to do this, we invoke Hall’s theorem, which relates a global property (the size of a matching) to a local property (expansion of subsets).

The expansion of subsets can be captured by the problem of *smaller* matching [4] which is a combinatorial pattern matching problem where the pattern and text elements are numbers, and a pattern element is said to *match* a text element if its value is less than the value of the text element. We show that this problem can be solved approximately in $\tilde{O}(n)$ time; this result may be of independent interest.

In order to reduce the complexity of our algorithms we need to estimate a certain combinatorial parameter. Specifically, for a certain natural family of subsets of the unit square (call it F), we need to find a small subset F' of F such that for any element from F there exists a member of F' within distance $\epsilon > 0$ to it (with respect to the Hausdorff metric). For example, for a family of convex polygons, it is known that a sub-family of size roughly $2^{O(1/\epsilon^{0.5})}$ exists [10]; the exponent of $1/\epsilon$ (in this case 0.5) is called the *exponent of entropy* of F [10]. Our proof uses the result for convex polygons as well as some additional ideas to show that the exponent of entropy for our family is at most 1.25.

1.2 Outline of the Paper

In Section 2 we provide basic definitions and notation and summarize our main results. The following two sections describe the basic method. In Section 3 we start by describing a strongly polynomial time algorithm that provides a constant factor approximation to the bottleneck distance. This algorithm exemplifies the method by which we reduce the geometric matching problem to a combinatorial matching problem. In Section 4, we show how the use of Hall’s Theorem enables us to obtain the desired approximation scheme, albeit at the cost of making the dependence on ϵ exponential. In Section 5 (which may be omitted in a first reading), we exploit the notion of *metric entropy* mentioned above to improve the dependence of our algorithms on ϵ . Finally, in Section 6, we describe techniques to improve the running time of the above algorithms to nearly linear.

2 The Main Results

We start with some definitions. For a point p , let $B_r(p)$ denote the ball of radius r centered at p i.e $B_r(p) = \{q \mid \|q - p\| \leq r\}$. We define $B(p) = B_1(p)$. For a set of points S , $B_r(S) = \cup_{s \in S} B_r(s)$. Let $A|_B$ denote the set $A \cap B$. We define a *translation* $t = (t_x, t_y)$ as a mapping from $p = (x, y)$ to $t(p) = (x + t_x, y + t_y)$. For a set of points P , $t(P) = \cup_{p \in P} t(p)$. We say that a function $f : X \rightarrow Y$ is a ν -to-one mapping if $\max_{y \in Y} |\{x \in X \mid f(x) = y\}| \leq \nu$. We denote $[u]$, $u \in \mathbb{R}$ to be the value of u rounded to the nearest integer.

Given point sets $P, Q \subseteq \mathbb{R}^2$, we define the *cost* of a ν -to-one mapping f from P to Q as the length of the longest edge in the mapping defined by f i.e $c(f) = \max_{p \in P} \|p - f(p)\|$. The *generalized bottleneck distance* between point sets P, Q with parameter ν equals $\min_f c(f)$ where the minimum is taken over all ν -to-one mappings from P to Q . We note here that the bottleneck distance is a special case ($\nu = 1$) of the generalized bottleneck distance. Moreover, the *Hausdorff distance*, defined as the maximum (over $p \in P$) of the distance from p to its closest neighbour $q \in Q$, can also be viewed as a special case of the generalized bottleneck distance (with $\nu = |P|$).

In this paper, we will consider the approximate version of this problem, defined as:

Problem: Approximate Generalized Bottleneck Matching

Input: Point sets $P, Q \subseteq \mathbb{R}^2$, $\alpha, b \geq 1$, p, r such that there exists a ν -to-one mapping $f : P \rightarrow Q$ having cost $c(f) = r$.

Output: $\alpha \cdot \nu$ -to-one mapping f' having cost $c(f') \leq b \cdot r$.

Such a solution is called an (α, b) -approximate solution. If we moreover require that the fraction of points from P that are not mapped to a unique point is at most δ , we call it a (α, b, δ) -approximate solution.

All our results are for computing the approximate generalized bottleneck distance under translations. In the above definition, we thus assume that there exists a translation t such that the conditions on p and r hold with respect to the point sets $t(P)$ and Q . The output is the set of *all* translations and associated functions f' such that $c(f')$ satisfies the conditions specified (again with respect to $t(P)$ and Q).

For all our algorithms, given a solution translation we can compute the corresponding mapping from P to Q in linear time using a grid-based nearest neighbour search. The one exception to this is the scheme outlined in Section 6.2, where the algorithm that outputs the mapping takes time $O(n \log n)$ for a given translation.

We present a summary of results in Table 1.

Smaller Matching The combinatorial pattern matching problem that we use in our algorithms is called *Smaller Matching* [4], and is defined as follows:

Problem: Smaller Matching

Input: Text string $T = T[0], \dots, T[n - 1]$ and Pattern string $P = P[0], \dots, P[k - 1]$.

Output: Output string $O = O[0], \dots, O[n - k - 1]$, where $O[i] = 1$ if for all $j = 0 \dots k - 1$, $P[j] \leq T[i + j]$.

¹The exact dependence on ϵ is not shown. For pattern matching under translation, multiply the times by $1/\epsilon^2$; for congruence under translation and rotation, multiply the bounds by $1/\epsilon^4$.

Approximation	Running Time ¹
$(3, 5 + \epsilon)$	$\tilde{O}(\sqrt{kn})$
$(6, 5 + \epsilon)$	$\tilde{O}(n)$
$(4, 5 + \epsilon)$	$\tilde{O}(n)$
$(3, 1 + \epsilon)$	$\tilde{O}(2^{\log(1/\epsilon)/\epsilon^{1.25}} \sqrt{kn})$
$(6, 1 + \epsilon)$	$\tilde{O}(2^{\log(1/\epsilon)/\epsilon^{1.25}} n)$
$(4, 1 + \epsilon)$	$\tilde{O}(2^{\log(1/\epsilon)/\epsilon^{1.25}} n)$

Table 1: Our Results

In [4], an algorithm is presented for this problem that runs in time $O(n\sqrt{k} \log k)$.

3 A simple $O(1)$ -approximation algorithm

In this section we present a $(3, c + \epsilon)$ -approximation algorithms for computing the generalized bottleneck distance between two point sets. The basic idea of the algorithm is as follows. Let the optimal translation be t and assume that the cost of the matching is 1. For convenience, we shall assume that the function f presented in the input is one-to-one. Superimpose $t(P)$ and Q and impose a brick-wall grid structure on \mathbb{R}^2 as shown on Figure 1. Without loss of generality, we can assume that no point of $P \cup Q$ lies on a grid line (this can be achieved by displacing the grid by a small amount in the x and y dimensions).

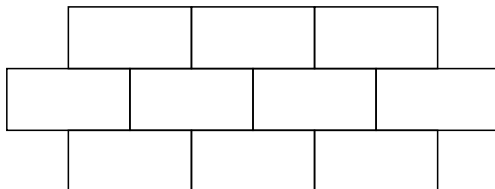


Figure 1: The brick-wall

The width of each grid cell C is 4 and the height is 2. Consider any cell and let $C' = B(C)$ (see Figure 2). Let $C(q)$ denote the closed cell C having center q .

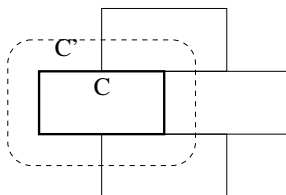


Figure 2: The brick-wall - one cell

Fact 3.1. *For any point $p \in \mathbb{R}^2$, the number of cells within distance 1 from p is at most 3.*

Since the function f is one-to-one, we know that

$$|t(P)_{|C}| \leq |Q_{|C'}| \quad (3.1)$$

(since any point from $t(P)_{|C}$ has to have a corresponding unique point in $Q_{|C'}$). We refer to Equation 3.1 as the *expansion inequality* (See Figure 3).

The expansion inequality guarantees that at the optimal translation, $|t(P)_{|C}| \leq |Q_{|C'}|$ for any cell C .

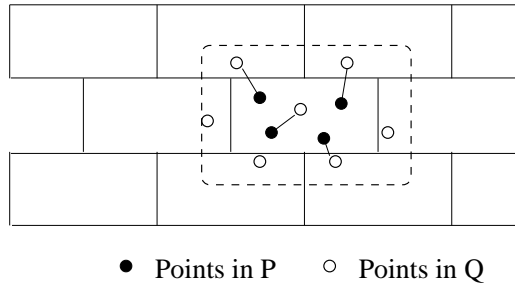


Figure 3: The expansion inequality

Now assume that we are given another translation (say t') such that the expansion inequality is true for every cell C . In this case one can find a function f' which is 3-to-one with bottleneck value equal to the diameter Δ of C plus 1. The construction of f' is as follows. For every cell C , we construct a one-to-one mapping f_C from $t(P)|_C$ to $Q|_C$ (since $|t(P)|_C| \leq |Q|_C|$, such a mapping always exists). The mapping f' is the union of all such f_C . The cost of f' is bounded by $\Delta + 1$. Furthermore, Fact 3.1 establishes that f' is a three-to-one mapping.

It remains to show how to find a translation t' such that the expansion inequality holds. For each cell C let $p_C = (x_C, y_C)$ denotes its center.

Let $g_t : \mathbb{R}^2 \rightarrow \mathbb{Z}$ be the function

$$g_t(p) = \begin{cases} |t(P)|_C & p = (x_C, y_C) \\ 0 & \text{otherwise} \end{cases}$$

Let $h : \mathbb{R}^2 \rightarrow \mathbb{Z}$ be the function $h(q) = |Q|_{B(C(q))}$

The problem can be now stated as: find a translation t such that for all $p \in \mathbb{R}^2$ we have $g_t(p) \leq h(p)$

Firstly, we approximate h by restricting its domain to $Z_\epsilon = \{\epsilon \cdot (i, j) | i, j \in \mathbb{Z}\}$. Let $g(p) = g_I(p)$ where I is the identity transform. We restrict the domain of g to its support in Z_ϵ . Let the corresponding functions be h' and g' . Set $P' = \{(i, j) | g(\epsilon i, \epsilon j) > 0\}$, $Q' = \{(i, j) | h(\epsilon i, \epsilon j) > 0\}$. Given that $|P| = k$ and $|Q| = n$, we have that $|P'| \leq k$ and $|Q'| = O(n/\epsilon^2)$. Also note that any two points in $P' \cup Q'$ are at least one unit away from each other.

Fact 3.2. *Let $t = (i\epsilon, j\epsilon)$ such that $\forall p \in P', g'_t(p) \leq h'(p)$. Then the generalized bottleneck distance between P and Q is at most $1 + \epsilon$.*

We now reduce this to a one-dimensional matching problem using the random projection technique of Cardoze and Schulman [7]. Select a random vector $u \in \mathbb{R}^2$ from the spherically symmetric Gaussian distribution with total variance 1. Let $\mu(p) = [u \cdot p]$. We denote the resulting point sets as $\mu(P')$, $\mu(Q')$. Fix a point $p_0 \in P'$ and let N be the set of vectors $t \in \mathbb{R}^2$ such that $t(p_0) \in Q'$ but $t(P') \not\subseteq Q'$. Observe that $|N| \leq |Q'|$ (because each translation $t \in N$ is of the form $q - p_0$, for $q \in Q'$).

For each $t \in N$ fix $A(t) \in P'$ such that $t(A(t)) \notin Q'$. For this fixed p_0 , the probability of a false match is the probability that for some translation $t \in N$, $\mu(t(P')) \subseteq \mu(Q')$. In this event, the point $A(t)$ must be matched to some point $q \in Q$. Note that prior to projection, $|A(t) - q| \geq 1$, and thus after projection, the probability that $|\mu(t(A(t))) - \mu(q)| \leq 1/(2\sqrt{\pi}n^3)$ is $O(1/n^3)$. Summing over all q, t , the probability of a false positive for a fixed p_0 is $O(1/n)$, yielding only a constant probability of a false positive over all p_0 (which can be reduced by repetitions).

We discretize the unit line at intervals of $1/3\sqrt{\pi}n^3$. Let $x[i] = g'(p)$, where $i = \lceil 3\mu(p)\sqrt{\pi}n^3 \rceil$. Similarly define $y[i]$ in terms of $h'(p)$. The domains of both vectors x, y is the interval $[0 \dots L = \Theta(n^3\epsilon^3)]$. In the next step, we reduce the size of this interval to $O(n)$ by hashing. Select a random prime $p \in [L \dots 2L]$; this can be done in time $O(\log^4 n)$ [6, Section 9.7]. Pick q uniformly at random from $[0 \dots p - 1]$. Fix $h = \Theta(n)$. Construct the vectors x', y' where $x'[(qi \bmod p) \bmod h] = x[i]$ and $y'[(qi \bmod p) \bmod h] = y[i]$. The number of non-zero entries in x', y' are $O(k)$ and $O(n/\epsilon^2)$ respectively. Let o' (rsp. o) be the solution to an instance of **Smaller Matching** with inputs x', y' (rsp. x, y).

Let t be fixed. Consider the entries $o[t]$ and $o'[t]$. The probability that the hashing yields a false match under the translation t is the probability $\Pr[o'[t] = 1 \mid o[t] = 0]$. Once again, we can find a location i such that $x[i + t] \neq 0$ and $y[t] = 0$. For a fixed position j such that $y[j] > 0$, the probability that $(i + t)q \bmod p \bmod h = jq \bmod p \bmod h$ is $O(1/n)$ by the fact that the hash function is a *nearly-universal function* [21, page 233]. Summing over all j , we conclude that $\Pr[o'[t] = 1 \mid o[t] = 0] = 1/c$ for some constant $c > 1$.

The resulting instance of **Smaller Matching** is then solved in time $\tilde{O}(\sqrt{kn}/\epsilon^2)$ by using the algorithm of [4]. Thus in time $\tilde{O}(\sqrt{kn}/\epsilon^2)$ we obtain an approximation that is correct with probability $1 - \Theta(1)$. By repeating the randomization steps a logarithmic number of times, we can achieve any success probability $1 - 1/n^{\Theta(1)}$. Observing that the diameter Δ of a cell is $2\sqrt{5}$, we obtain the following theorem:

Theorem 3.1. *Given point sets P, Q and $\epsilon > 0$, we can compute a $(3, 2\sqrt{5} + 1 + \epsilon)$ -approximation to the generalized bottleneck distance under translations between P, Q in time $\tilde{O}(n\sqrt{k}/\epsilon^2)$. The algorithm is correct with probability $1 - 1/n^{\Theta(1)}$.*

A Better Tiling. We can obtain a better approximation bound by considering other tilings. Consider the hexagonal tiling (with side length 2) depicted on Figure 4.

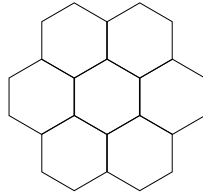


Figure 4: A hexagonal tiling

One can observe that Fact 3.1 holds for this tiling as well. Since the diameter of a hexagonal cell is 4, we obtain the following:

Theorem 3.2. *There is a $(3, 5 + \epsilon)$ -approximation for generalized bottleneck matching under translation that runs in time $\tilde{O}(n\sqrt{k}/\epsilon^2)$.*

4 The Use Of Hall's Theorem

The approximation to the optimal generalized bottleneck distance achieved by the algorithm developed in the previous section is given by the diameter of a cell in the tiling used. The key is in the construction of the mapping f of points in P to points in Q . The expansion inequality (3.1) only guarantees that for every point $p \in t(P)_{|C}$, there exists *some* point $q \in Q_{|C'}$ within distance Δ . However, we know that for the optimal matching, a stronger expansion inequality can be stated (see Figure 5):

$$|t(P)_{|C}| \leq |B(t(P)_{|C}) \cap Q| \tag{4.1}$$

We will employ Hall's theorem to exploit this fact.

Theorem 4.1. Hall's Matching Theorem [16] *Let $G = (A \cup B, E)$ be a bipartite graph. Then G has a matching of A into B iff $|\Gamma(X)| \geq |X|$, for all $X \subseteq A$.*

We divide each cell C into a grid of cell size ϵ ; we will assume that $1/\epsilon = E$ is an integer. Let the (i, j) th grid cell be denoted by g_{ij}^C . An *indexed subset* I of C is a subset of $\mathcal{I} = \{g_{(i,j)}^C \mid i \leq kE, j \leq lE\}$ (where a cell C is of dimension $k \times l$). Such a subset can also be represented in terms of its characteristic function χ_I , where $\chi_I(i, j) = 1 \iff g_{ij}^C \in I$.

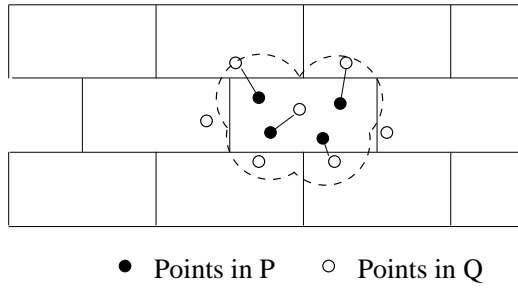


Figure 5: Strong expansion

Assume that P and Q are aligned according to the optimal translation t . Consider $S \subseteq C$. From Theorem 4.1, it follows that $(\forall S \subseteq C) |B(S) \cap Q|_{|C|} \geq |S \cap t(P)|_{|C|}$. This implies a similar (although slightly weaker) property that can be stated in terms of indexed subsets. Let $B_E(I) = \{g_{uv}^C \mid |u - i|^2 + |v - j|^2 \leq E^2, \chi_I(i, j) = 1\}$. Then we have²

$$(\forall I \subseteq \mathcal{I}) |B_{E+1}(I) \cap Q| \geq |I \cap t(P)| \quad (4.2)$$

Our algorithm will find a translation t for which equation (4.2) is true for every cell C . By Hall's theorem, this implies the existence of a bottleneck matching from $t(P) \cap C$ to Q with value $1 + O(\epsilon)$, for every cell C .

A solution to this problem will be used to check the validity of equation (4.2). Thus, our algorithm for generalized bottleneck matching is as follows:

1. For all index subsets I ,
2. Let $P' = Q' = \{(i, j) \mid \chi_I(i, j) = 1\}$.
3. Let $g'((i, j)) = |P \cap g_{ij}^C| \cdot \chi_I(i, j)$, $h'((i, j)) = |B_{E+1}(g_{ij}^C) \cap Q| \cdot \chi_I(i, j)$.
4. As in Section 3, find solution vector o_I and function T such that $o_I[t] = 1 \iff \forall p, g'_{t'}(p) \leq h'(p)$, where $t' = T(t)$.
5. Compute $o = \bigwedge_I o_I$.

A cell C of G is decomposed into a grid having E^2 cells. Thus, there are 2^{E^2} invocations of the *Smaller Matching* algorithm, yielding a running time of $\tilde{O}(2^{1/\epsilon^2} \sqrt{kn}/\epsilon^2)$.

All translations $T(t)$ such that $o[t] = 1$ yield bottleneck matches. The mapping f induced by such a match will not be one-to-one in general. However, consider any point $q \in Q$. We know that any point $p \in P$ mapped to q has to belong to a cell within distance 1 from q . By Fact 3.1 there are at most 3 such cells, and therefore f is 3-to-one.

We thus have:

Theorem 4.2. *There is a $(3, 1 + \epsilon)$ -approximate algorithm for generalized bottleneck matching under translations running in time $\tilde{O}(2^{1/\epsilon^2} \sqrt{kn}/\epsilon^2)$.*

Tricriterion approximation bounds. In order to limit (say to δ) the fraction of points which are assigned to non-unique image points, we can enlarge the cells by a factor of $1/\delta$ and translate the grid by a random vector. It is easy to see that the fraction of matching edges close to the boundary of cells is at most $O(\delta)$ for any fixed translation t . The remaining edges will be assigned one-to-one. In this way we obtain $(\alpha, O(1/\delta), \delta)$ -approximation algorithms for constants α as before.

Approximate congruence. Here we discuss how to extend our result for translations to isometries. To this end, we first translate P such that the centroids of P and Q align (say at point s). Then, we find a rotation of P around s which minimizes its distance to Q ; we perform it by reducing it to $O(\log \Delta)$ instances of pattern matching under

²The subscript of $E + 1$ (instead of E) comes from the rounding error of coordinates of t to nearest multiples of ϵ .

translations, as in [7, 19]. By the argument of [1] this approach results in a constant factor approximation algorithm; this can be further reduced to $1 + \epsilon$ by exploring $O(1/\epsilon^2)$ such points [1]. In this way we obtain an algorithm for approximate congruence with essentially the same performance guarantees as the algorithm for translations and with running time larger by a factor of $\log \Delta/\epsilon^2$.

5 A Combinatorial Lemma

In this section we improve the dependence of the running time on ϵ from $2^{O(1/\epsilon^2)}$ to $2^{O(\frac{\log 1/\epsilon}{\epsilon^{1.25}})}$. The basic idea is to reduce the number of sets for which we verify the expansion by using geometry. More specifically, assume that $I_1 \dots I_s$ are indexed sets such that the Hausdorff distance between $(B_{E+1}(I_i))$ and $(B_{E+1}(I_j))$ is $O(1)$.

We replace all tests

$$|B_{E+1}(I_k)|_C \cap Q_{|C'}| \geq |I_k|_C \cap t(P)|_C|$$

by one test

$$|B_{E+O(1)}(I)|_C \cap Q_{|C'}| \geq |I|_C \cap t(P)|_C|$$

where $I = \cup_k I_k$. Notice that the second test is implied by the first one. Moreover, if the second test holds for all cells for a translation t' , then by Hall's theorem we are guaranteed a matching with bottleneck value $1 + O(\epsilon)$. Therefore, if we prove that the number of such tests is small, we improve the running time of the algorithm. This number can be bounded from above by using the concept of a cover defined as follows.

Definition 5.1 (λ -cover). Let (X, d) be a metric space. A set $S \subset X$ is called a λ -cover ($\lambda \geq 0$) for $Y \subset X$ if for any $p \in Y$ there exists $q \in S$ such that $d(p, q) \leq \lambda$.

Let $R = E + 1$ and $R' = \gamma \cdot R$ for some constant γ . Let the set \mathcal{X} (with associated metric d) be the Hausdorff space consisting of subsets of $[0, R']^2$. Consider $\mathcal{Y} \subset \mathcal{X}$ defined as $\mathcal{Y} = \{B_R(S) | S \subset \{0, \dots, R'\}^2\}$. Observe that the number of tests can be bounded by the size of a 1-cover for \mathcal{Y} . Moreover, for any set Y from the cover, we can easily find a union of all S such that $d(B_R(S), Y) \leq 1$ (corresponding to I above) simply by taking all $p \in \{0, \dots, R'\}^2$ such that $B_R(p) \subset B_1(Y)$. Therefore, it remains to construct a small 1-cover for \mathcal{Y} .

Theorem 5.1. *There exists a 1-cover for \mathcal{Y} of size $2^{O(R^{1.25} \log R)}$.*

Proof. Note that by a scaling argument it is sufficient to show the existence of an $O(1)$ -cover. Let $r = \sqrt{R}$. Impose a square grid on $[0, R']^2$ of side r . We will show that for each grid cell G the family $\mathcal{Y}_G = \{Y \cap G | Y \in \mathcal{Y}\}$ has an $O(1)$ -cover of size $c = 2^{O(\sqrt{r} \log r)}$. Since there are only $O(R)$ cells, this implies a $O(1)$ -cover for \mathcal{Y} of size $c^{O(R^{1.25})}$, which is what we need.

Consider a fixed cell G and take any $S \subset \{-\gamma R, \dots, \gamma R\}^2$. We assume that all points $p \in S$ are *significant*, i.e. $B_R(S) \cap G \neq B_R(S - \{p\}) \cap G$ (otherwise we can remove insignificant points and apply the same arguments). We also assume that G is not completely included in $B_R(S)$. Notice that in this case for any significant point p there is a part of a boundary (an arc) of $B_R(S)$ which belongs to G . For each $p \in S$, let H_p be any half-space containing $B_R(p)$ and with boundary tangent to the arc of $B_R(p)$. We will approximate $B_R(p) \cap G$ by $H_p \cap G$ using the following claim.

Claim 5.1. $d(B_R(p) \cap G, H_p \cap G) = O(1)$.

Proof. Follows from the fact that the length of the arc is $O(\sqrt{R})$ while its radius is R . □

As a corollary we obtain that

$$d(\cup_{p \in S} B_R(p) \cap G, \cup_{p \in S} H_p \cap G) = O(1)$$

Therefore, it is sufficient to construct an $O(1)$ -cover for the latter set. In fact, we will construct such cover for sets $\overline{\cup_{p \in S} H_p} \cap G = \cap_{p \in S} \overline{H_p} \cap G$ (where \overline{A} denotes the complement of A). Notice that this set is a subset of the set CP of convex polygons contained in G . Thus it is sufficient to find a $O(1)$ -cover for CP . However, it follows from the

theorem by Dudley [10] that for any $P \in \text{CP}$ there exists $P' \in \text{CP}$ with only $O(\sqrt{\tau})$ vertices such that $d(P, P') \leq 1$. Each vertex can be rounded to the nearest point of $\{-R, \dots, R\}$ and thus can be represented only $O(\log R)$ bits. This implies that we can represent P' using only $O(\sqrt{\tau} \log R)$ bits. Since the set of all P' 's is an $O(1)$ -cover for CP , the theorem follows. \square

We mention that the exponent of 1.25 is probably not tight. One can observe however that a 1-cover must have size at least $2^{\Omega(\sqrt{R})}$. To see this, consider S containing \sqrt{R}/l points (for $l > 1$) uniformly spaced on a horizontal line with $y = -R$. One can observe that for sufficiently large l we can independently move each point up or down by 2 units while always ensuring that the topmost point of a ball centered at any point is not covered by any other ball. Therefore, we can obtain $2^{\sqrt{R}/l}$ sets such that the distance between any pair of distinct sets is at least 2.

6 Near-Linear Matching Schemes

In this section we describe improvements to our algorithms that yield running times of the form $\tilde{O}(n)$ (the dependency on ϵ remains unchanged). These improvements come at the cost of the approximation in the fanout of the matching; a $(3, 1 + \epsilon)$ -approximation becomes a $(4, 1 + \epsilon)$ -approximation. The improvements can be applied as modules over the previous algorithms. Therefore, we will describe them in terms of the simpler algorithm of Section 3; they extend analogously to the algorithm of Section 4.

In Section 6.1, we will outline a scheme that outputs a $(6, 5 + \epsilon)$ -approximate solution and has running time $\tilde{O}(n/\epsilon^2)$. In Section 6.2 we then show how to improve it to a $(4, 5 + \epsilon)$ -approximation with the same running time. Although this result appears to strictly dominate the previous one, this is in fact not the case; we explain this in more detail in Section 6.2.

6.1 Approximate Smaller Matching

The key routine in the previous algorithm was an algorithm to solve the *Smaller matching* problem. Here, we define an approximate version of this problem.

Problem: *Approximate Smaller Matching*

Input: Text string $T = T[0], \dots, T[n-1]$, Pattern string $P = P[0], \dots, P[k-1]$ and $0 < \gamma \leq 1$.

Output: Output string $O = O[0], \dots, O[n-k-1]$, where $O[i] = 1$ if for all $j = 0 \dots k-1$, $P[j] \leq (1 + \gamma)T[i+j]$.

Replacing all instances of *Smaller Matching* by instances of *Approximate Smaller Matching* yields all translations where for each cell C , $|t(P)|_C| \leq (1 + \gamma)|Q|_C$. Unfortunately, this *approximate* expansion property does not yield the required matching; some points of P may remain unmatched. However, suppose we replace each point in Q by two points arbitrarily close to each other. Call the resulting set Q^2 . Observe that the expansion property is preserved with respect to Q^2 ; specifically, for $\gamma \leq 1$, $|t(P)|_C| \leq (1 + \gamma)|Q|_C| \leq (1 + \gamma)|Q^2_C|/2 \leq |Q^2_C|$.

Running the algorithm from Section 3 on P, Q^2 yields a $(3, 5 + \epsilon)$ -approximation algorithm for P, Q^2 which can be easily converted into a $(6, 5 + \epsilon)$ -approximation algorithm for P, Q by merging the two points in Q^2 corresponding to each point in Q .

It remains to prove that we can solve the one-dimensional *Approximate Smaller Matching* in near-linear time. Recall that the instance consists of two vectors $x[i], y[i]$, where $0 \leq i \leq O(n)$. For $j \in \{0, \dots, O(\log n)/\log(1 + \gamma)\}$, define binary vectors x_{\leq}^j, y_{\geq}^j as:

$$\begin{aligned} x_{\leq}^j[i] &= 1 && \text{if } (1 + \gamma)^j \leq x[i] < (1 + \gamma)^{j+1} \\ y_{\geq}^j[i] &= 1 && \text{if } (1 + \gamma)^j \leq y[i] \end{aligned}$$

Vector x matches vector y at position i if $x[i] \leq (1 + \gamma)y[i]$. Fix j . If $x_{\leq}^j[i] = 1$, then $x[i] \geq (1 + \gamma)^j$ and then for x to match y at i , $y[i]$ must be at least $(1 + \gamma)^{j-1}$, which is equivalent to the condition $y_{\geq}^{j-1}[i] = 1$.

The condition $\forall i x_{\leq}^j[i] = 1 \Rightarrow y_{\geq}^{j-1}[i] = 1$ can be written as

$$\sum_i x_{\leq}^j[i] \cdot \overline{y_{\geq}^{j-1}[i]} = 0$$

where $\overline{}$ is the complement operator. Define $O^j[t] = 1$ if

$$\sum_i x_{\leq}^j[i] \cdot \overline{y_{\geq}^{j-1}[i+t]} = 0$$

It is easy to see that the desired vector O can now be computed by taking the bitwise **and** of the vectors O^j .

For a fixed j , $O^j[t]$ can be computed deterministically for all t in $O(n \log n)$ time using the convolution-based scheme of Fischer and Paterson [14]. There are $O(\log n/\gamma)$ such instances, yielding a running time of $O(n \log^2 n/\gamma)$ for solving *Approximate Smaller Matching*.

6.2 An Expander-based Matching Algorithm

In our next improvement, we show how to improve the approximation guarantee achievable from $(6, 5 + \epsilon)$ to $(4, 5 + \epsilon)$, with no significant change in the running time of the algorithm. As before, we employ the algorithm for *Approximate Smaller Matching* on the sets P, Q to determine all translations where for each cell C , $|t(P)_{|C}| \leq (1 + \gamma)|Q_{|C}'|$. However, here we replicate each point in Q four times. Let the resulting set be Q^4 . For a set $S \subseteq P$, let \mathcal{C}_S be the set of cells C such that $S \cap t(P)_{|C} \neq \emptyset$.

From Fact 3.1 we know that each point in Q has at most three cells that are adjacent to it. Therefore, $|\cup_{C \in \mathcal{C}_S} Q_{|C}'| \geq \sum_{C \in \mathcal{C}_S} |Q_{|C}'|/3$. The guarantee provided by the matching algorithm implies that

$$\begin{aligned} \sum_{C \in \mathcal{C}_S} |t(P)_{|C}| &\leq \sum_{C \in \mathcal{C}_S} (1 + \gamma)|Q_{|C}'| \\ &\leq 3(1 + \gamma)|\cup_{C \in \mathcal{C}_S} Q_{|C}'| \\ &\leq 3/4(1 + \gamma)|\cup_{C \in \mathcal{C}_S} Q^4_{|C}'| \end{aligned}$$

Since $|S| \leq \sum_{C \in \mathcal{C}_S} |t(P)_{|C}|$, we have that $B(S) \geq 4/(3(1 + \gamma))|S|$ for any $S \subseteq P, |S| \leq 3(1 + \gamma)/4|Q|$. In other words, the bipartite graph defined by P and Q^4 is an expander (with expansion $4(1 + \gamma)/3$). Consider Dinic's algorithm [9] for computing a perfect matching in a graph via augmenting paths. In [20], Motwani proved that there always exists an augmenting path of length $O(\log n/\log \Delta)$ in a bipartite expander with expansion Δ . Furthermore, Efrat and Itai [11] showed that for bipartite graphs defined by points in the plane, augmenting paths can be computed in time $O(\tilde{O}(n))$ (as opposed to $O(m)$ for general graphs). Combining these two results, we obtain a procedure running in time $\tilde{O}(n/\gamma \log \gamma)$ that returns a matching between P, Q^4 . As before, we then merge the four points in Q^4 that correspond to a single point in Q , yielding a $(4, 5 + \epsilon)$ -approximate matching. Notice that the process of computing the actual match takes $\tilde{O}(n/\gamma)$ time, compared to the linear-time simple approximate nearest-neighbour procedures which suffice for the earlier schemes.

7 Conclusions

It seems likely that the 1.25 exponent of $1/\epsilon$ (from Section 5) can be improved. We believe that the right value is 0.5 (as for convex polygons). Unfortunately, the proof appears complex since the (complement) of a union of balls is not convex; in fact, the shape even does not have to be connected.

It might be possible to reduce the dependence on $\epsilon > 0$ from exponential to polynomial, and still maintain the $(1 + \epsilon)$ -approximation of the matching cost. One possibility is to encode the RNC algorithm for bipartite matching testing using algebraic convolutions (in fact, one can view our method as a constant time parallel algorithm with exponential number of processors). However, at present we do not know if this is possible.

We would like to thank the anonymous referees for correcting many errors and suggesting changes that improved the presentation.

References

- [1] ALT, H., BEHREND, B., AND BLÖMER, J. Approximate matching of polygonal shapes. In *Proc. 11th ACM Symp. Comp. Geom.* (June 1995), pp. 186–193.
- [2] ALT, H., AND GUIBAS, L. Discrete geometric shapes: Matching, interpolation, and approximation. A Survey. Tech. Rep. B96-11, Freie Universität Berlin, December 1996.
- [3] ALT, H., MELHORN, K., WAGENER, H., AND WELZL, E. Congruence, similiarity, and symmetries of geometric objects. *Discrete Computational Geometry* 3 (1988), 237–256.
- [4] AMIR, A., AND FARACH, M. Efficient 2-dimensional approximate matching of non-rectangular figures. *Information and Computation* 118, 1 (1995), 1–11.
- [5] ARKIN, E., KEDEM, K., MITCHELL, J., SPRINZAK, J., AND WERMAN, M. Matching points into pairwise-disjoint noise regions: Combinatorial bounds and algorithms. *ORSA Journal on Computing* 4, 4 (1992), 375–386.
- [6] BACH, E., AND SHALLIT, J. *Algorithmic Number Theory, Volume I: Efficient Algorithms*. MIT Press, 1996.
- [7] CARDOZE, D., AND SCHULMAN, L. Pattern matching for spatial point sets. In *Proc. 39th IEEE Symp. Foundations of Computer Science* (November 1998), pp. 156–165.
- [8] CHEW, L., GOODRICH, M., HUTTENLOCHER, D., KEDEM, K., KLEINBERG, J., AND KRAVETS, D. Geometric pattern matching under euclidean motion. In *Proceedings of the Fifth Canadian Conference on Computational Geometry* (1993), pp. 151–156.
- [9] DINIC, E. A. Algorithm for solution of a problem of maximum flow in networks with power estimation. *Sov. Math. Doklady* 11 (1970), 1277–1280.
- [10] DUDLEY, R. M. Metric entropy of some classes of sets with differentiable boundaries. *Approximation Theory* 10 (1974), 227–236.
- [11] EFRAT, A., AND ITAI, A. Improvements on bottleneck matching and related problems using geometry. In *Proc. 12th ACM Symp. Comp. Geom.* (June 1996), pp. 301–310.
- [12] FINN, P., HALPERIN, D., KAVRAKI, L. E., LATOMBE, J. C., MOTWANI, R., SHELTON, C., AND VENKATASUBRAMANIAN, S. Geometric manipulation of flexible ligands. *LNCS Series - 1996 ACM Workshop on Applied Computational Geometry* 1148 (1996), 67–78.
- [13] FINN, P., KAVRAKI, L. E., LATOMBE, J. C., MOTWANI, R., SHELTON, C., VENKATASUBRAMANIAN, S., AND YAO, A. Rapid: Randomized pharmacophore identification for drug design. In *Proc. 13th ACM Symp. Comp. Geom.* (1997), pp. 324–333.
- [14] FISCHER, M. J., AND PATERSON, M. S. *Complexity of Computation*. SIAM-AMS, 1974, ch. String Matching and Other Products, pp. 113–125.
- [15] GOODRICH, M. T., MITCHELL, J. B., AND ORLETSKY, M. W. Practical methods for approximate geometric pattern matching under rigid motions. In *Proc. 5th ACM-SIAM Symp. Disc. Algorithms* (Jan. 1994), pp. 103–113.
- [16] HALL, P. On representatives of subsets. *Journal of London Mathematics Society* 10 (1935), 26–30.
- [17] HEFFERNAN, P. J., AND SCHIRRA, S. Approximate decision algorithms for point set congruence. *Computational Geometry: Theory and Applications* 4, 3 (1994), 137–156.
- [18] HUTTENLOCHER, D. P., KEDEM, K., AND KLEINBERG, J. M. On dynamic Voronoi diagrams and the minimum Hausdorff distance for points sets under Euclidean motion in the plane. In *Proc. 8th ACM Symp. Comp. Geom.* (June 1992), pp. 110–120.
- [19] INDYK, P., MOTWANI, R., AND VENKATASUBRAMANIAN, S. Geometric matching under noise: Combinatorial bounds and algorithms. In *Proc. 10th ACM-SIAM Symp. Disc. Algorithms* (Jan. 1999), pp. 457–465.

- [20] MOTWANI, R. Average-case analysis of algorithms for matchings and related problems. *Journal of the ACM* 41 (1994), 1329–1356.
- [21] MOTWANI, R., AND RAGHAVAN, P. *Randomized Algorithms*. Cambridge University Press, 1995.
- [22] MOUNT, D., NETANYAHU, N., AND LEMOIGNE, J. Efficient algorithms for robust point pattern matching and applications to image registration. *Pattern Recognition*, 32 (1999), 17–38.
- [23] NOREL, R., FISCHER, D., WOLFSON, H. J., AND NUSSINOV, R. Molecular surface recognition by a computer vision-based technique. *Protein Engineering* 7, 1 (1994), 39–46.