SCHOOL OF OPERATIONS RESEARCH
AND INDUSTRIAL ENGINEERING
COLLEGE OF ENGINEERING
CORNELL UNIVERSITY
ITHACA, NY 14853-3801

TECHNICAL REPORT NO. 940

November 1990

THE TRANSLATION SQUARE MAP AND APPROXIMATE CONGRUENCE

by

Paul J. Heffernan*

*Supported by an NSF graduate fellowship.

The Translation Square Map and Approximate Congruence

Paul J. Heffernan*
SORIE, Cornell University
Ithaca, NY 14853

November 1, 1990

Abstract

We consider the point matching problem under translation with the L_∞ or L_1 metric. We introduce the *translation square map* (TSM), which provides an algorithm for testing ϵ-congruence of equal cardinality sets A and B. The TSM can be used to form a $((2\delta/(\epsilon+2\delta))\epsilon_{opt}(A,B), (2\delta/\epsilon)\epsilon_{opt}(A,B))$-approximate point matching algorithm, where $\epsilon_{opt}(A,B)$ represents the smallest ϵ such that A and B are ϵ-congruent under the L_∞ (or L_1) metric. The approximate algorithm runs in time $O((\epsilon/\delta)^6 n^3)$; we know of no other (exact or approximate) algorithm for point matching under translation with run time $o(n^4)$.

Keywords: Computational geometry, computer vision, point matching.

1 Introduction

An important problem in computer vision is determining whether two point sets are equivalent. In this paper we ask whether two planar point sets of equal cardinality are congruent under translation. We introduce a new tool, the *translation square map* (TSM), to help answer this question.

In real instances, exact congruence is an elusive pursuit, because of errors in measurement and computational imprecision. This limitation led Baird [Ba] and Alt, et al. [AMWW] to introduce the concept of *approximate congruence*. Two equal cardinality point sets A and B are approximately congruent with tolerance ϵ, or *ϵ-congruent*, under a given metric, if there exists a bijection $l : B \to A$ and an isometric mapping M such that $dist(M(b), l(b)) \leq \epsilon$ for every $b \in B$. (An isometric mapping, or congruence, is a mapping from $\Re^2$ to itself that preserves Euclidean distance.) The decision problem for a tolerance

*Supported by an NSF graduate fellowship.

ϵ asks for such a bijection l and isometry M, or for a statement that no such pair exists. In this paper, our introduction of the translation square map leads to efficient decision algorithms for ϵ-congruence for the isometry M restricted to the family of translations, and the metric fixed as the L_∞ or L_1 metric.

The approximate congruence problem has been studied by many researchers. In [AMWW], algorithms for the ϵ-congruence decision problem are given for the L_2 and L_∞ metrics, under both the general class of isometries and the class of translations. An algorithm in [AMWW] solves the ϵ-congruence optimization problem under translation, which asks for $\epsilon_{opt}(A, B)$, the minimum value of ϵ such that the point sets A and B are ϵ-congruent under translation. In [AMZ] and [Sp], output-sensitive algorithms are given that generalize the approach of [AMWW] by considering sets A and B of unequal cardinality, and by generalizing the metric. Specifically, [AMZ] allows the "noise regions" (i.e. the ϵ-balls around the points of A) to be arbitrary nonconvex polygons; it also considers piecewise linear noise functions. In [AKMSW], algorithms are given for the decision problem for numerous metrics under various classes of isometries and under similarity (an isometry plus a change of scale), but with the assumption that the ϵ-balls around point set A are pairwise-disjoint or have limited overlap. In [AKMSW], [AMZ], and [Sp], combinatorial upper and lower bounds are given on the number of distinct bijections that can satisfy the decision problem. A problem related to ϵ-congruence is that of finding a translation that minimizes the Hausdorff distance between two point sets; this problem is studied in [HuKe] and [HKS].

The high run-times of algorithms for approximate congruence motivate the search for approximate algorithms. The word "approximate" is used twice here, with two different purposes. For point sets A and B under a metric and a class of isometries, define $\epsilon_{opt}(A, B)$ to be the smallest value of ϵ such that A and B are ϵ-congruent. Schirra [Sc] calls a decision algorithm (α, β)-*approximate* if, for every ϵ outside of the interval $[\epsilon_{opt}(A, B) - \alpha, \epsilon_{opt}(A, B) + \beta]$, the algorithm returns a correct decision (for values of ϵ inside this interval, the algorithm may either return a correct decision or return no decision). Schirra [Sc] presents approximate algorithms for general metrics and isometries. The translation square map produces an asymptotically faster algorithm for the special case of the L_∞ metric and the isometries restricted to translations.

For the general case of point sets A and B of size n, the TSM method solves the ϵ-congruence decision problem for the L_∞ metric under translation in $O(n^6)$ time. If, for each $x \in \Re^2$, the number of points $a \in A$ such that $dist(x, a) \leq \epsilon$ is bounded by k, then the algorithm runs in $O(k^3 n^3)$ time. A $((2\delta/(\epsilon + 2\delta))\epsilon_{opt}(A, B), (2\delta/\epsilon)\epsilon_{opt}(A, B))$-approximate algorithm runs in time $O((\epsilon/\delta)^6 n^3)$.

2 The Translation Square Map

We describe now the translation square map (TSM). We are given planar point sets A and B of size n, and a tolerance parameter ϵ. The decision problem asks whether there exists a bijection $l : B \to A$ and a translation T such that $dist(T(b), l(b)) \leq \epsilon$ for every

$b \in B$, where $dist(\cdot, \cdot)$ represents the L_∞ distance between two points in $\Re^2$. In other words, we "grow" each point $a \in A$ to an orthogonal square of side length 2ϵ, centered at a. We then search for a bijection l and a translation T such that the point $T(b)$ lies in the square $a = l(b)$, for every $b \in B$. In the following discussion, we let A denote the original point set and A_ϵ the corresponding set of 2ϵ-size squares.

Let b_x and b_y represent the points of B with the least x- and y-coordinates, respectively. Similarly, a_x and a_y are the left-most and bottom-most squares of A_ϵ. Let T_0 be the translation of B such that b_x has the same x-coordinate as the left side of a_x and b_y the same y-coordinate as the bottom of a_y. Clearly any translation T that admits ϵ-congruence is of the family $T_0 + (x', y')$, for $0 \leq x', y' \leq 2\epsilon$. This limits our attention to an orthogonal square of side length 2ϵ that represents all candidate translations; that is, each point in this "square map" is of the form $(x', y') \in [0, 2\epsilon] \times [0, 2\epsilon]$, and therefore represents a candidate translation $T = T_0 + (x', y')$. It is natural to call this map of candidate translations the *translation square map* (TSM). The question thus amounts to determining whether there exists a point in the TSM that admits a valid matching of A and B. For each $(a, b) \in A_\epsilon \times B$, determine the region of points of the TSM representing translations $T = T_0 + (x', y')$ that place the point b inside or on the square a. This region is called the *overlap region* (or *overlap rectangle*) of a and b, and is denoted $OR(a, b)$; we say that a pair $(a, b) \in A_\epsilon \times B$ *overlap* if $OR(a, b) \neq \emptyset$. If a and b overlap, their overlap region is an orthogonal rectangle touching the TSM boundary on two or three sides. In Figure 1, it is shown that an overlap region with three sides on the TSM boundary has one vertical (Figure 1(a)) or horizontal (1(b)) edge interior to the TSM, and a region with two sides on the boundary has an internal vertical edge and an internal horizontal edge (1(c)).

Suppose, for each $(a, b) \in A_\epsilon \times B$, we place $OR(a, b)$ onto the TSM, as shown in Figure 2. This partitions the TSM into (possibly nonconvex) orthogonal polygons (called blocks). For the remainder of this paper, we will use TSM to denote this partition. Each block of the TSM represents a bipartite graph $G = (V_1, V_2; E)$, where the vertex sets V_1 and V_2 represent the elements of A_ϵ and B, respectively, and $e = (a, b) \in E$ if and only if the overlap rectangle of a and b covers the block. The decision problem is thereby reduced to asking whether there exists a block of the TSM that defines a graph with a perfect matching. Note that every point in a block defines the same bipartite graph, and that neighboring blocks define graphs that differ by only one edge.

While we defined the TSM in terms of the left-most and bottom-most elements of A_ϵ and B, we could instead use the centroids c_A and c_B of A and B. Since every translation that allows ϵ-congruence of two point sets maps the centroids of the sets within distance ϵ [Sc], the TSM can be defined as an orthogonal square of side length 2ϵ, where the translation mapping c_B to c_A lies at the center of the square.

The TSM algorithm for ϵ-congruence can now be described. First, we determine the initial translation T_0 in $O(n)$ time (we could instead determine the translation moving c_B to c_A in $O(n)$ time). Next, for each pair $(a, b) \in A_\epsilon \times B$, we construct the overlap rectangles ($O(n^2)$ time). We separately sort the horizontal and vertical edges of the overlap rectangles to form the translation square map ($O(n^2 \log n)$ time). We now walk

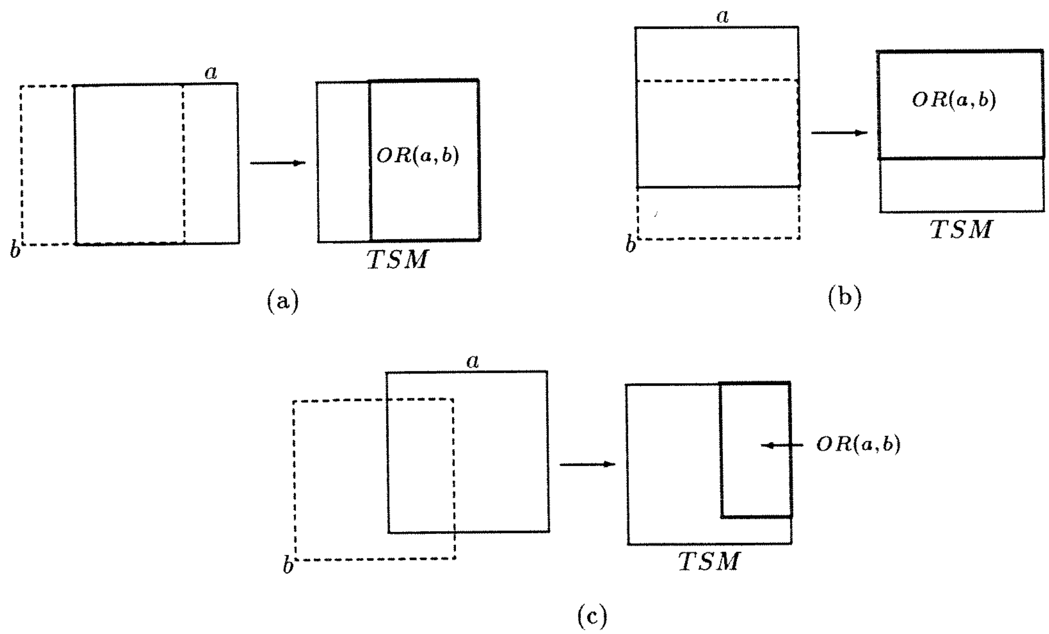

Figure 1: Examples of overlapping pairs and their corresponding overlap rectangles.

through the blocks of the TSM, solving the matching problem on the bipartite graph of each block. Neighboring blocks (blocks sharing an edge boundary) define graphs that vary by one edge (assuming no two overlap rectangles share an internal edge). Therefore, after solving an initial matching problem, for the lower-left block, say, we can walk through the blocks in such a way that, when solving the matching problem for the current block, we can use the already-computed optimal matching from a neighboring block. In this manner, the work in each block except the first consists of updating an optimal solution for a graph with one edge inserted or deleted.

The complexity of this procedure depends on the size of the TSM, and the number of edges in the bipartite graphs. Let e represent the total number of overlapping pairs $(a,b) \in A_\epsilon \times B$. In other words, e is the number of rectangles that form the TSM arrangement. Therefore the numbers of vertical and horizontal segments in the TSM are both bounded by e. This shows that the TSM consists of $O(e^2)$ blocks. Since a pair (a,b) produces an edge in a graph G only if a and b overlap, the number of edges in the graph G for any block is bounded by e. The total work of the TSM procedure consists of constructing the TSM ($O(n^2 \log n)$ time, as shown above), solving an initial bipartite matching problem to optimality (in $O(n^{2.5})$ time [HoKa]), and then performing $O(e^2)$ updates. Updating an optimal matching on a bipartite graph with one edge inserted or deleted can be performed in $O(e)$ time [HoKa], giving a $O(n^{2.5} + e^3)$ time bound for the TSM algorithm.

If a point of A (B) does not overlap any point of B (A), then the point sets are not ϵ-congruent, since the corresponding node in G is incident to no edge. Since an initial

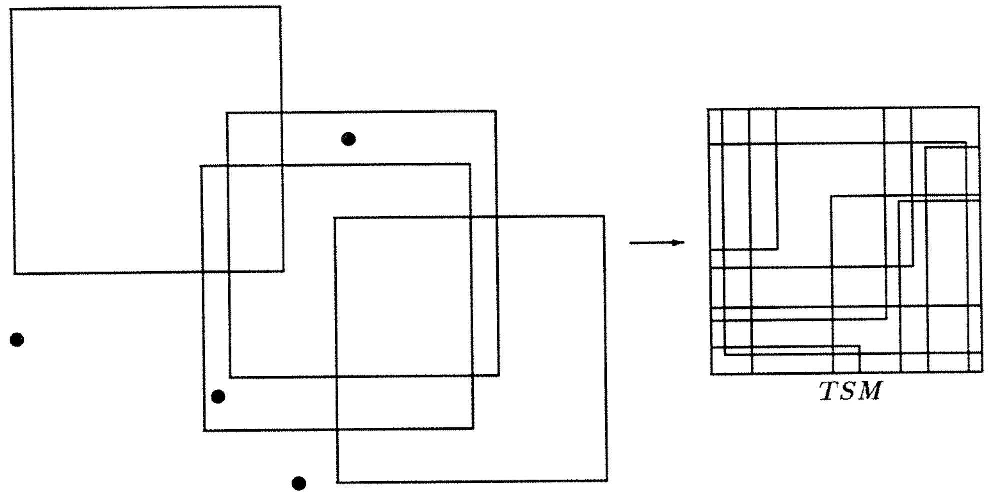

Figure 2: Sets A and B, and the Translation Square Map.

check will reveal such a situation, we assume that every point of A (B) overlaps some point of B (A). Therefore $n \leq e$, so the above time complexity can be written as $O(e^3)$.

The space bound depends on our manner of traversing the TSM blocks. A simple scheme is to extend each segment in the TSM until it touches the TSM boundary. Now the TSM partition consists only of rectangles, which allows an easy walk through the blocks (Figure 3), and requires that only one $O(n)$ space maximum matching be stored at a time. Of course, this creates more blocks, but the number is still $O(e^2)$. Note that neighboring blocks define graphs that differ by one edge or are identical.

For the general case, $e = O(n^2)$, so the TSM algorithm runs in $O(n^6)$ time. This matches the time bound given in [AMWW]. If every point of B overlaps no more than

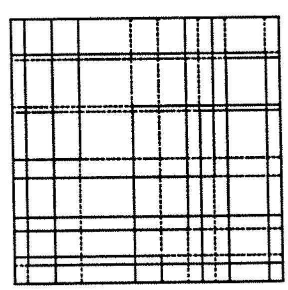

Figure 3: The example of Figure 2, with extended segments providing an easy walk through the blocks.

k squares of A_ϵ, then $e = O(kn)$, giving an $O(k^3n^3)$ time algorithm. Limited overlap is implied when the points of A are spaced somewhat apart. For example, if no point of the plane is covered by more than k squares of A_ϵ, then no point of B overlaps more than $4k$ squares of A_ϵ, giving an $O(k^3n^3)$ algorithm. The special case of no intersections among squares of A_ϵ is solved in $O(n \log n)$ time in [AMWW].

3 An Approximate Algorithm

In an effort to improve run-times, we turn now to approximate algorithms for ϵ-congruence. We describe an algorithm which, given point sets A and B, and ϵ and δ such that $\epsilon/\delta \in \mathcal{Z}$, either returns a matching of A and B within tolerance $\epsilon + 2\delta$, or certifies that A and B are not ϵ-congruent. The algorithm runs in time $O((\epsilon/\delta)^6 n^3)$. We will also show how the algorithm can be used in a $((2\delta/(\epsilon+2\delta))\epsilon_{opt}(A,B), (2\delta/\epsilon)\epsilon_{opt}(A,B))$-approximate algorithm (according to the definition of [Sc]) in the same time bound.

The original ϵ-approximate algorithm took each point a of A, and turned a into an orthogonal square with side length 2ϵ centered at a. The approximate algorithm lets the square "grow" to be a square of side length $2(\epsilon + \delta)$ in the following manner. Set A down on graph paper with horizontal and vertical lines spaced 2δ apart. Consider the collection of squares of side length $2(\epsilon + \delta)$ with sides on the graph-paper lines. Assuming nondegeneracy, each orthogonal square of side length 2ϵ is completely contained in exactly one of these $2(\epsilon + \delta)$-size squares. "Grow" a into this square a' (Figure 4). In other words, for each point $a \in A$, a is in one of the 2δ-size squares of the graph paper. Move a to the center of this small square and then construct a', a $2(\epsilon + \delta)$-size square about a. Let $A'_{(\epsilon,\delta)}$ (or simply A') represent the set of these $2(\epsilon + \delta)$-size squares.

Now use the TSM method to test for $(\epsilon + \delta)$-congruence between $A'_{(\epsilon,\delta)}$ and B. By our clever "standardization" of the squares A', as a point $b \in B$ is translated to the right by $2(\epsilon+\delta)$ units, it crosses vertical edges of squares of A' at no more than $2(\epsilon+\delta)/2\delta = \epsilon/\delta + 1$ distinct x-coordinates. Therefore b intersects the interiors of at most $2(\epsilon/\delta + 1)$ squares of A' with distinct x-coordinates, as b moves through all candidate translations. Similarly, b overlaps at most $2(\epsilon/\delta + 1)$ squares of A' with distinct y-coordinate, bounding the total number of squares of A' that overlap b at

$$(2(\epsilon/\delta + 1))^2 = O((\epsilon/\delta)^2).$$

This bounds e by $O((\epsilon/\delta)^2 n)$.

Applying the TSM method to test for $(\epsilon + \delta)$-congruence of $A'_{(\epsilon,\delta)}$ and B is different in one respect: several squares of A' may coincide. Having squares with multiplicity greater than 1 requires a slightly different formulation. Any bipartite matching problem can be formulated as a max-flow problem; for our approximate algorithm we will formulate, for each block of the TSM, a max-flow problem that looks similar to a bipartite matching problem. Given a block, we create a source and sink node, a set V_2 of n nodes corresponding to the points of B, and a set V_1 of $\leq n$ nodes, each corresponding to a *distinct* square of A'. Draw an arc of capacity 1 from the source node to each node of V_2, and an

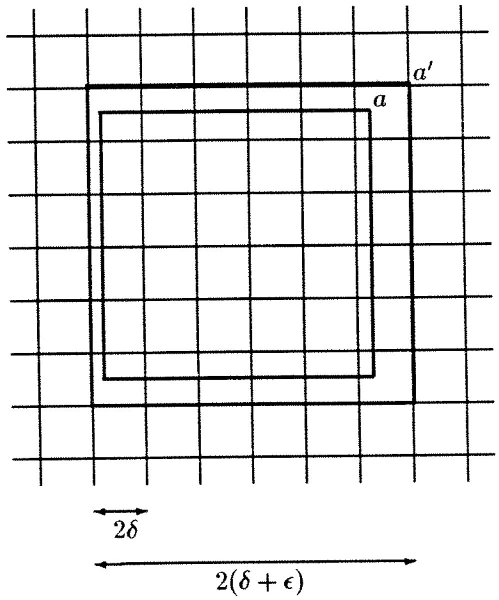

Figure 4: Square a "growing" into square a'.

arc (b, a') of capacity 1 for each pair $(b, a') \in V_2 \times V_1$ such that $OR(a', b)$ covers the block. For each node of V_1, draw an arc to the sink whose capacity equals the multiplicity of the corresponding square $a' \in A'$. We first solve the max-flow problem for the lower-left block of the TSM (in $O(n^3)$ time [MKM]), and then walk through the blocks of the TSM. Because at most one edge is inserted or deleted in each step, each update can be performed in $O(e)$ time (this is the complexity of an iteration of the Ford and Fulkerson max-flow algorithm [FF], as analyzed in [Ch]). Therefore the TSM method runs in $O((\epsilon/\delta)^6 n^3)$ time for the approximate algorithm.

Since we use the squares $A'_{(\epsilon,\delta)}$, instead of A_ϵ, this method is approximate for the original sets A and B. A square $a' \in A'$ of size $2(\epsilon + \delta)$ contains the corresponding 2ϵ-size square $a \in A_\epsilon$ that generated it; therefore if A and B are ϵ-congruent, A' and B are $(\epsilon+\delta)$-congruent, and the procedure returns a matching of A' and B. If an $(\epsilon + \delta)$-congruent matching of A' and B is returned, then for each pair (a', b) in the matching, b lies within distance $\epsilon + 2\delta$ of the corresponding point $a \in A$, as shown in Figure 5.

Let $\epsilon_{opt}(A, B)$ be the minimum value ϵ such that A and B are ϵ-congruent. Schirra [Sc] provides the following definition: a decision algorithm for ϵ-congruence of A and B is (α, β)-approximate if, for $\epsilon \notin [\epsilon_{opt}(A, B) - \alpha, \epsilon_{opt}(A, B) + \beta]$, the algorithm returns a correct answer. That is, for $\epsilon > \epsilon_{opt}(A, B) + \beta$, the algorithm returns an ϵ-congruent matching, and for $\epsilon < \epsilon_{opt}(A, B) - \alpha$, it answers that A and B are not ϵ-congruent. For ϵ in the interval $[\epsilon_{opt}(A, B) - \alpha, \epsilon_{opt}(A, B) + \beta]$, the algorithm may provide a correct answer (i.e. provide an ϵ-congruent matching or state that one does not exist), or may give no answer. Schirra [Sc] gives a $((1/2)\epsilon_{opt}(A, B), \epsilon_{opt}(A, B))$-approximate algorithm

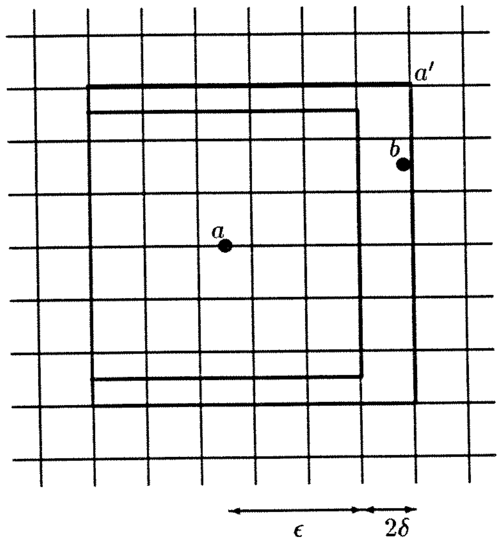

Figure 5: $dist(a,b) \leq \epsilon + 2\delta$.

for the general metric and the general class of isometries, with time complexity $O(n^4)$, and a (γ,γ)-approximate algorithm with run-time $O((\epsilon/\gamma)^2 n^4)$. Our algorithm for the special case of the L_∞ metric and the class of translations reduces the exponent of n; it is $((2\delta/(\epsilon+2\delta))\epsilon_{opt}(A,B), (2\delta/\epsilon)\epsilon_{opt}(A,B))$-approximate and runs in time $O((\epsilon/\delta)^6 n^3)$.

Theorem 1 *The following algorithm is* $((2\delta/(\epsilon+2\delta))\epsilon_{opt}(A,B), (2\delta/\epsilon)\epsilon_{opt}(A,B))$*-approximate for* ϵ*-congruence of* A *and* B*:*

Test $A'_{(\epsilon,\delta)}$ and B for $(\epsilon+\delta)$-congruence.
  If not $(\epsilon+\delta)$-congruent, answer A *and* B *are not* ϵ*-congruent.*
Test $A'_{(\epsilon/\gamma,\delta/\gamma)}$ and B for $(1/\gamma)(\epsilon+\delta)$-congruence, where $\gamma = (\epsilon+2\delta)/\epsilon > 1$.
  If a $(1/\gamma)(\epsilon+\delta)$-congruent matching of $A'_{(\epsilon/\gamma,\delta/\gamma)}$ and B is returned, it corresponds to an ϵ-congruent matching of A and B.

Proof. The algorithm returns a "not ϵ-congruent" answer only if $A'_{(\epsilon,\delta)}$ and B are not $(\epsilon+\delta)$-congruent; however, this condition implies that A and B are not ϵ-congruent. If a $(1/\gamma)(\epsilon+\delta)$-congruent matching of $A'_{(\epsilon/\gamma,\delta/\gamma)}$ and B is returned, it corresponds to a $(1/\gamma)(\epsilon+2\delta)$-congruent matching of A and B. Since $1/\gamma = \epsilon/(\epsilon+2\delta)$, this is an ϵ-congruent matching of A and B. Thus, if the algorithm returns an answer, the answer is correct.

Suppose $\epsilon < (1/\gamma)\epsilon_{opt}(A,B)$. If we find an $(\epsilon+\delta)$-congruent matching of $A'_{(\epsilon,\delta)}$ and B, it corresponds to an $(\epsilon+2\delta)$-congruent matching for A and B. But $\epsilon+2\delta = \gamma\epsilon < \epsilon_{opt}(A,B)$, a contradiction. Therefore we find no $(\epsilon+\delta)$-congruent matching of $A'_{(\epsilon,\delta)}$ and B, and answer that there exists no ϵ-congruent matching for A and B.

Suppose $\epsilon > \gamma\epsilon_{opt}(A,B)$. If we determine that $A'_{(\epsilon/\gamma,\delta/\gamma)}$ and B are not $(1/\gamma)(\epsilon+\delta)$-congruent, then A and B are not (ϵ/γ)-congruent. But $\epsilon/\gamma > \epsilon_{opt}(A,B)$, a contradiction. Therefore we do find a $(1/\gamma)(\epsilon+\delta)$-congruent matching of $A'_{(\epsilon/\gamma,\delta/\gamma)}$ and B, which corresponds to a $(1/\gamma)(\epsilon+2\delta)$-congruent matching, i.e. an ϵ-congruent matching, of A and B. ∎

The time complexity of the algorithm is clearly $O((\epsilon/\delta)^6 n^3)$.

If δ is chosen to equal $\epsilon/2$, use of the TSM can be avoided. The following lemma is a modification of a lemma given in [Sc]:

Lemma 2 *If A and B are ϵ-congruent, then there exists a 2ϵ-congruent matching of A and B that uses the translation $T_{c_B c_A}$ (which is the translation that moves the centroid of B, c_B, to the centroid of A, c_A).*

By fixing our translation to be $T_{c_B c_A}$, the TSM shrinks to a single point. Evaluating a single bipartite matching problem tells us whether A and B are 2ϵ-congruent under the translation $T_{c_B c_A}$. This gives the following $O(n^{2.5})$-time, $((1/2)\epsilon_{opt}(A,B), \epsilon_{opt}(A,B))$-approximate algorithm:

Test A and B for 2ϵ-congruence under translation $T_{c_B c_A}$.
  If not 2ϵ-congruent under $T_{c_B c_A}$, then answer A *and* B
  are not ϵ-congruent.
Test A and B for ϵ-congruence under translation $T_{c_B c_A}$.
  If an ϵ-congruent matching is returned for A and B under $T_{c_B c_A}$,
  then it is an ϵ-congruent matching for A and B.

We have so far restricted our discussion to the L_∞ metric, because it produces ϵ-balls around points of A that are orthogonal squares. The L_1 metric gives ϵ-balls that are diamonds: squares with edges with slopes of 1 and -1. Since every translation that allows ϵ-congruence of A and B maps the centroids c_A and c_B within distance ϵ of each other, we can create a diamond-shaped TSM of side length $\sqrt{2}\epsilon$, with diamond-shaped overlap regions forming the partition. The TSM method, therefore, works as well for the L_1 metric as it does for L_∞.

The translation square map can be extended to higher dimensions. For point sets A and B in $\Re^d$, the d-dimensional translation hypercube map for metric L_∞ consists of $O(e^d)$ regions, each corresponding to a bipartite graph with $O(e)$ edges. This results in an $O(e^{d+1})$ time algorithm for testing ϵ-congruence of A and B. The approximate algorithm exhibits an $O((\epsilon/\delta)^{d(d+1)} n^{d+1})$ time bound.

An open question arises from the updating stage of the algorithms. Currently, $O(e^2)$ updates of a maximum bipartite matching or maximum-flow on a graph G are performed. Since G has $O(e)$ edges, these updates are performed in $O(e)$ time each by known methods, which use an optimal solution for a neighboring block. In fact, the TSM problem exhibits more structure: in Figure 3, horizontal rows of blocks are shown updated in succession. If we process each row from left to right, then the sequence of edge insertions and deletions encountered is almost the same for each row. When updating a row, we have at our disposal optimal solutions for the row below. Perhaps we can perform bipartite matching and max-flow updates in better than $O(e)$ amortized time when performing a large number of updates with the structure of the TSM.

Acknowledgement: The author thanks Joseph S.B. Mitchell for introducing him to this topic.

References

[AMWW] H. Alt, K. Mehlhorn, H. Wagener, and E. Welzl, "Congruence, Similarity, and Symmetries of Geometric Objects", *Discrete and Computational Geometry*, **3** (1988), pp. 237-256.

[AKMSW] E.M. Arkin, K. Kedem, J.S.B. Mitchell, J. Sprinzak, M. Werman, "Matching Points into Noise Regions: Combinatorial Bounds and Algorithms", to appear in *Symposium on Discrete Algorithms*, Jan. 1991.

[AMZ] E.M. Arkin, J.S.B. Mitchell, and K. Zikan, "Algorithms for Generalized Matching of Point Sets", Manuscript, 1988. Presented at the CORS/ORSA/TIMS National Meeting, Vancouver, B.C., May 1989.

[Ba] H.S. Baird, *Model-Based Image Matching Using Location*, MIT Press, 1984.

[Ch] V. Chvatal, *Linear Programming*, W.H. Freeman and Co., 1983, pp. 374-380.

[FF] L.R. Ford, Jr., and D.R. Fulkerson, "A Simple Algorithm for Finding Maximal Network Flows and an Application to the Hitchcock Problem", *Canadian Journal of Mathematics*, **9** (1957), pp. 210-218.

[HoKa] J.E. Hopcroft and R.M. Karp, "An $n^{5/2}$ Algorithm for Maximum Matchings in Bipartite Graphs", *SIAM Journal of Computing*, **2** (1973), pp. 225-231.

[HuKe] D.P. Huttenlocher and K. Kedem, "Computing the Minimum Hausdorff Distance for Point Sets Under Translation", in *Proc. of 6th ACM Symp. on Computational Geometry* (1990), pp. 340-349.

[HKS] D.P. Huttenlocher, K. Kedem, and M. Sharir, "The Upper Envelope of Voronoi Surfaces: Theory and Applications", in preparation.

[MKM] V.M Malhotra, M.P. Kumar, and S.N. Maheshwari, "An $O(|V|^3)$ Algorithm for Finding Maximum Flows in Networks", *Information Processing Letters*, **7** (1978), pp. 277-278.

[Sc] S. Schirra, "Approximate Algorithms for Approximate Congruence", Manuscript, Universitat des Saarlandes, 1990.

[Sp] J. Sprinzak, "Thesis title", M.S. Thesis, Hebrew University, Dept. of Computer Science, 1990.