Matching Point Sets with respect to the Earth Mover's Distance

Sergio Cabello* Panos Giannopoulos† Christian Knauer† Günter Rote†

Abstract

Let A and B be two sets of m resp. n weighted points in the plane, with $m \leq n$. We present $(1+\epsilon)$ and $(2+\epsilon)$-approximation algorithms for the minimum Euclidean Earth Mover's Distance between A and B under translations and rigid motions respectively. In the general case where the sets have unequal total weights the algorithms run in $O((n^3 m/\epsilon^4)\log^2(n/\epsilon))$ time for translations and $O((n^4 m^2/\epsilon^4)\log^2(n/\epsilon))$ time for rigid motions. When the sets have equal total weights, the respective running times decrease to $O((n^2/\epsilon^4)\log^2(n/\epsilon))$ and $O((n^3 m/\epsilon^4)\log^2(n/\epsilon))$. We also show how to compute a $(1+\epsilon)$ and $(2+\epsilon)$-approximation of the minimum cost Euclidean bipartite matching under translations and rigid motions in $O((n^{3/2}/\epsilon^{7/2})\log^5 n)$ and $O((n/\epsilon)^{7/2}\log^5 n)$ time respectively.

1 Introduction

Let $A = \{a_1, \ldots, a_m\}$ and $B = \{b_1, \ldots, b_n\}$ be two planar weighted point sets with $m \leq n$. A weighted point $a_i \in A$ is defined as $a_i = \{(x_{a_i}, y_{a_i}), w_i\}, i = 1, .., m$, where $(x_{a_i}, y_{a_i}) \in \mathbb{R}^2$ and $w_i \in \mathbb{R}^+ \cup \{0\}$ is its weight. A weighted point $b_j \in B$ is defined similarly as $b_j = \{(x_{b_j}, y_{b_j}), u_j\}, j = 1, .., n$. Let $W = \sum_{i=1}^m w_i$ and $U = \sum_{j=1}^n u_j$ be the total weight, or simply weight, of A and B respectively.

The Earth Mover's Distance (EMD) is a similarity measure for weighted point sets with applications in colour-based image retrieval [7], shape matching [7, 2, 1] and music score matching [9]. In a typical scenario, a pattern is reduced to a set of feature weighted points; the larger the weight, the more important the point for the whole pattern. Informally, a weighted point a_i can be seen as an amount (supply) of earth or mass, equal to w_i units, positioned at (x_{a_i}, y_{a_i}); alternatively it can be taken as an empty hole (demand) of w_i units of earth capacity. We assign arbitrarily the role of the supplier to A and that of the receiver/demander to B, setting, in this way, a direction of earth transportation. The Earth Mover's Distance (EMD) of A to B measures the minimum amount of work needed to fill the holes with earth. A formal definition of the EMD will be given shortly.

In order to measure the similarity of two sets A and B independently of transformations, one wants to find a transformed version of, say, A that attains the minimum possible distance to B. In this paper we are interested in transformations that change only the position of the points, not their weights; in particular, we focus on translations and rigid motions – sometimes referred to as isometries. We consider B to be fixed, while A can be translated and/or rotated relative to B. We assume some initial positions for both sets, denoted simply by A and B. Let $\mathcal{I}$ be the set of all possible rigid motions in the plane. We denote by R_θ a rotation about the origin by some angle $\theta \in [0, 2\pi)$ and by $T_{\vec{t}}$ a translation by some $\vec{t} \in \mathbb{R}^2$. Any rigid motion $I \in \mathcal{I}$ can be uniquely defined as a translation followed by a rotation, that is, $I = I_{\vec{t},\theta} = R_\theta \circ T_{\vec{t}}$, for some $\theta \in [0, 2\pi)$ and $\vec{t} \in \mathbb{R}^2$. In general, transformed versions of A are denoted by $A(\vec{t}, \theta) = \{a_1(\vec{t}, \theta), \ldots, a_m(\vec{t}, \theta)\}$ for some $I_{\vec{t},\theta} \in \mathcal{I}$. For simplicity, translated only versions of A are denoted by $A(\vec{t}) = \{a_1(\vec{t}), \ldots, a_m(\vec{t})\}$.

The EMD between $A(\vec{t}, \theta)$ and B, is a function $\mathrm{EMD} : \mathcal{I} \to \mathbb{R}^+ \cup \{0\}$ defined as

$$\mathrm{EMD}(\vec{t}, \theta) = \min_{F \in \mathcal{F}(A,B)} \frac{\sum_{i=1}^m \sum_{j=1}^n f_{ij} d_{ij}(\vec{t}, \theta)}{\min\{W, U\}},$$

where $d_{ij}(\vec{t}, \theta)$ is the distance of $a_i(\vec{t}, \theta)$ to b_j, and $F = \{f_{ij}\} \in \mathcal{F}(A, B)$ with $\mathcal{F}(A, B)$ being the set of all feasible flows between A and B defined by the constraints: (i)$f_{ij} \geq 0, i = 1, ..., m, j = 1, ..., n$, (ii)$\sum_{j=1}^n f_{ij} \leq w_i, i = 1, ..., m$, (iii)$\sum_{i=1}^m f_{ij} \leq u_j, j = 1, ..., n$, and (iv)$\sum_{i=1}^m \sum_{j=1}^n f_{ij} = \min\{W, U\}$. In case that $\vec{t}$ or θ or both are constant, we simply write $\mathrm{EMD}(\theta)$, $\mathrm{EMD}(\vec{t})$ and EMD respectively. The EMD is a metric when d_{ij} is a metric and $W = U$ [7]. When $W \neq U$ the EMD inherently performs partial matching since a portion of the weight of the 'heavier' set remains unmatched. We deal with the Euclidean EMD where d_{ij} is given by the L_2-norm. For simplicity, and without loss of generality, we assume that $\min\{W, U\} = 1$. We study the following problem:

Given two weighted point sets A, B compute a rigid motion $I_{\vec{t}_{opt},\theta_{opt}}$ that minimizes $\mathrm{EMD}(\vec{t}, \theta)$.

This problem was first studied by Cohen [7] who

*IMFM, Department of Mathematics, Jadranska 19, SI-1000 Ljubljana, Slovenia, sergio.cabello@imfm.uni-lj.si; partially supported by the European Community Sixth Framework Programme under a Marie Curie Intra-European Fellowship.

†Institut für Informatik, Freie Universität Berlin, Takusstraße 9, D-14195 Berlin, Germany, {panos, knauer, rote}@inf.fu-berlin.de

presented a Flow–Transformation iteration which alternates between finding the optimum flow for a given transformation, and the optimum transformation for a given flow. They showed that this iterative procedure converges, but not neccessarily to the global optimum. Computing the EMD for a given transformation is actually the transportation problem, a special minimum cost network flow problem for the solution of which there is a variety of polynomial time algorithms [3, 4]. However, as we discuss later on, the task of finding the optimal transformation for a given flow is not trivial. Cohen also gave simple algorithms that compute the optimum translation for the special case where $W = U$ and d_{ij} is the squared Euclidean distance. This case is quite restrictive since, in general, the sets need not have the same weight, and the use of squared Euclidean distance is statistically less robust than Euclidean distance [5]. Currently, no algorithm that computes the optimal translation and/or rotation is known for the Euclidean EMD.

Observe that the objective function is not linear in $\vec{t}$ and θ but it is still linear in F. Thus, the minimum EMD occurs at some vertex of the convex polytope $\mathcal{F}(A, B)$. This suggests the following straightforward algorithm: for every vertex $F = \{f_{ij}\}$ of $\mathcal{F}(A, B)$ compute the optimal rigid motion, i.e., the one that minimizes $\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}d_{ij}(\vec{t}, \theta)$. For translations, the latter problem reduces to the $Fermat - Weber$ problem where one wants to find a point that minimizes the sum of weighted distances to a set of given points. No exact solution to this problem is known even in the real RAM model of computation [5]. However, Bose et al. [5] gave a $O(n \log n)$-time $(1 + \epsilon)$-approximation algorithm for any fixed dimension. Using their algorithm for every vertex of $\mathcal{F}(A, B)$ gives only a $(1 + \epsilon)$-approximation of the minimum EMD under translations in exponential time.

In this paper, we give simple polynomial-time algorithms that achieve a $(1+\epsilon)$ and $(2+\epsilon)$-approximation for translations and rigid motions respectively.

2 Lower bounds and approximations of the EMD

First, we give two simple lower bounds on the EMD that are vital for the approximation algorithms given in the next sections. In these algorithms we need to compute the EMD for a given transformation. Computing the EMD exactly is expensive, and unecessary since we opt for approximations of the minimum EMD under transformations. We show how to get a $(1+\epsilon)$-approximation of the EMD in almost quadratic time.

The following lower bound comes directly from the definition of the EMD.

Observation 1 *Given two weighted point sets A and B,* $\mathrm{EMD} \geq \min_{i,j} d_{ij}$.

The next lower bound is due to Cohen [7]. The *center of mass* $C(A)$ of a planar weighted point set $A = \{(x_{a_i}, y_{a_i}), w_i\}, i = 1, \ldots, m$ is defined as $C(A) = \left(\sum_{i=1}^{m} w_i(x_{a_i}, y_{a_i})\right) / \sum_{i=1}^{m} w_i$.

Theorem 1 *[7, Theorem 6] Let A and B be two weighted point sets with equal weights. Then* $\mathrm{EMD} \geq |C(A) - C(B)|$.

Currently, the fastest strongly polynomial-time algorithm for the minimum cost flow problem on a graph $G(V, E)$ is due to Orlin [3], and runs in $O((|E| \log |V|)(|E| + |V| \log |V|))$ time. Using the algorithm of Callahan and Kosaraju [6], we can construct, in $O(n \log n + (n/\epsilon^2) \log 1/\epsilon)$ time, a linear size $(1 + \epsilon)$-spanner G_s, i.e., a graph $G_s(V, E')$ with $|E'| = O(n/\epsilon)$ such that the shortest path between any two points in G_s is at most $(1 + \epsilon)$ times the Euclidean distance of the points. Running the algorithm of Orlin on G_s produces an approximate value EMD_s such that $\mathrm{EMD} \leq \mathrm{EMD}_s \leq (1 + \epsilon)\mathrm{EMD}$ in $O((n^2/\epsilon^2) \log^2(n/\epsilon))$ time; we refer to this procedure as $\textsc{ApxEMD}(A, B, \epsilon)$.

Lemma 2 *For any given $\epsilon > 0$, a value EMD_s with $\mathrm{EMD} \leq \mathrm{EMD}_s \leq (1 + \epsilon)\mathrm{EMD}$ can be computed in $O((n^2/\epsilon^2) \log^2(n/\epsilon))$ time.*

Next, consider the case where $|A| = |B| = n$ and $w_i = u_j = 1, i = 1, ..., n, j = 1, ..., n$. The integer solutions property of the minimum cost flow problem and the fact that $0 \leq f_{ij} \leq 1$ imply that there is a minimum cost flow on G that results in a (perfect) matching between A and B. Hence, we can restrict ourselves to finding a minimum cost matching–usually called the *assignment* problem. Varadarajan and Agarwal [8] presented an algorithm that finds a matching with cost at most $(1+\epsilon)$ times that of an optimal matching in $O((n/\epsilon)^{3/2} \log^5 n)$ time; we refer to this algorithm as $\textsc{ApxMATCH}(A, B, \epsilon)$.

3 Approximation algorithms for translations

We denote by $\vec{t}_{i \to j}$ the translation which matches a_i and b_j; we call such a translation a *point-to-point* translation. Observation 1 implies that the point-to-point translation that is closest to $\vec{t}_{opt}$ gives a 2-approximation of $\mathrm{EMD}(\vec{t}_{opt})$. Hence, we have the following:

Lemma 3 *Given two weighted point sets A and B,* $\mathrm{EMD}(\vec{t}_{opt}) \leq \min_{i,j} \mathrm{EMD}(\vec{t}_{i \to j}) \leq 2\mathrm{EMD}(\vec{t}_{opt})$.

According to Observation 1, the point-to-point translation which is closest to $\vec{t}_{opt}$ can be at most $\mathrm{EMD}(\vec{t}_{opt})$ away from $\vec{t}_{opt}$. This bound is crucial for the $(1 + \epsilon)$-approximation algorithm given in Figure 1. Using a uniform square grid of suitable size we compute the EMD for a limited number of grid translations within a small neighborhood – of size $\mathrm{EMD}(\vec{t}_{opt})$ – of every point-to-point translation. Note that we do not know $\mathrm{EMD}(\vec{t}_{opt})$ but we can compute $\min_{i,j} \mathrm{EMD}(\vec{t}_{i\to j})$ which, according to Lemma 3, approximates $\mathrm{EMD}(\vec{t}_{opt})$ well-enough. In order to save time, rather than computing EMD exactly, we will approximate it using the procedure ApxEMD.

Translation(A, B, ϵ):

1. Let $\alpha = \min_{i,j}\mathrm{ApxEMD}(A(\vec{t}_{i\to j}), B, 1)$ and let G be a uniform square grid of spacing $c\epsilon\alpha$, where $c = 1/\sqrt{72}$.
2. For each pair of points $a_i \in A$ and $b_j \in B$ do:
   (a) Place a disk D of radius α around $\vec{t}_{i\to j}$.
   (b) For every grid point $\vec{t}_g \in D \cap G$ compute a value $\widetilde{\mathrm{EMD}}(\vec{t}_g) = \mathrm{ApxEMD}(A(\vec{t}_g), B, \epsilon/3)$.
3. Report the grid point $\vec{t}_{apx}$ that minimizes $\widetilde{\mathrm{EMD}}(\vec{t}_g)$.

Figure 1: Algorithm Translation(A, B, ϵ).

Theorem 4 *For any given $\epsilon > 0$, a translation $\vec{t}_{apx}$ such that $\mathrm{EMD}(\vec{t}_{apx}) \le (1+\epsilon)\mathrm{EMD}(\vec{t}_{opt})$ can be computed in $O((n^3m/\epsilon^4)\log^2(n/\epsilon))$ time.*

Next, consider the case of equal weight sets. Let $\vec{t}_{C(A)\to C(B)}$ be the translation that matches the centers of mass $C(A)$ and $C(B)$. Theorem 1 suggests the following trivial 2-approximation algorithm: compute $\mathrm{EMD}(\vec{t}_{C(A)\to C(B)})$. According to Theorem 1, $\vec{t}_{opt}$ is at most $\mathrm{EMD}(\vec{t}_{opt})$ far away from $\vec{t}_{C(A)\to C(B)}$. Hence, we need to search for $\vec{t}_{opt}$ only within a small neighborhood of $\vec{t}_{C(A)\to C(B)}$. We modify algorithm Translation(A, B, ϵ) as follows: First we compute $C(A)$ and $C(B)$. Then, we run $\mathrm{ApxEMD}(A(\vec{t}_{C(A)\to C(B)}), B, 1)$ and set α to the value returned. Next, we run $\mathrm{ApxEMD}(A(\vec{t}_g), B, \epsilon/3)$ for all the grid points $\vec{t}_g$ which are at most α away from $\vec{t}_{C(A)\to C(B)}$. The minimum over all these approximations gives the desired approximation bound. Hence, we have managed to save an nm term from the time bound of Theorem 4.

Theorem 5 *If A and B have equal total weights then, for any given $\epsilon > 0$, a translation $\vec{t}_{apx}$ such that $\mathrm{EMD}(\vec{t}_{apx}) \le (1+\epsilon)\mathrm{EMD}(\vec{t}_{opt})$ can be computed in $O((n^2/\epsilon^4)\log^2(n/\epsilon))$ time.*

For the assigment problem under translations, we can use the above algorithm for equal weight sets , running ApxMATCH instead of ApxEMD. This reduces the running time further.

Theorem 6 *For any given $\epsilon > 0$, a $(1+\epsilon)$-approximation of the minimum cost assignment under translations can be computed in $O((n^{3/2}/\epsilon^{7/2})\log^5 n)$ time.*

Note that, the latter algorithm can be also applied to equal weight sets with bounded integer point weights by replacing each point by as many points as its weight.

4 Approximation algorithms for rigid motions

We first give a $(2+\epsilon)$-approximation algorithm for rotations. Then, we combine this $(2+\epsilon)$-approximation algorithm with the $(1+\epsilon)$-approximation algorithms for translations to get $(2+\epsilon)$-approximation algorithms for rigid motions.

Rotations. Let $a_i\widehat{o}b_j$ be the angle between the segments $\overline{oa_i}$ and $\overline{ob_j}$ such that $0 \le a_i\widehat{o}b_j \le \pi$. Also, let $\theta_{i\to j}$ be the rotation of a_i by $a_i\widehat{o}b_j$ that aligns the origin o and points a_i and b_j such that both a_i and b_j are on the same side of o. Note that this is the rotation that minimizes $d_{ij}(\theta)$; we call such a rotation an *alignment rotation*. We have the following simple lemma.

Lemma 7 *Let a_i and b_j be two points in the plane with $a_i\widehat{o}b_j = \phi$. If a_i is rotated by an angle $\theta \le \phi$, then $d_{ij}(\theta) < 2d_{ij}$.*

Similarly to Lemma 3, and using Lemma 7, we can prove that the alignment rotation that is closest to θ_{opt} gives a 2-approximation of $\mathrm{EMD}(\theta_{opt})$. Hence, we have the following:

Lemma 8 *Given two weighted point sets A and B, $\mathrm{EMD}(\theta_{opt}) \le \min_{i,j}\mathrm{EMD}(\theta_{i\to j}) \le 2\mathrm{EMD}(\theta_{opt})$.*

By approximating $\mathrm{EMD}(\theta_{i\to j})$ with ApxEMD or ApxMATCH we can get a $(2+\epsilon)$-approximation of $\mathrm{EMD}(\theta_{opt})$. We call this algorithm Rotation(A, B, ϵ); from the context it will be always clear whether ApxEMD or ApxMATCH is used. Apart from the cost value, Rotation returns the corresponing rotation $\theta_{i\to j}$ as well.

Lemma 9 *For any given $\epsilon > 0$, a rotation θ_{apx} such that $\mathrm{EMD}(\theta_{apx}) \le (2+\epsilon)\mathrm{EMD}(\theta_{opt})$ can be computed in $O((n^3m/\epsilon^2)\log^2(n/\epsilon))$ time. For the minimum cost assignment problem under rotations the same approximation can be computed in $O((n^{7/2}/\epsilon^{3/2})\log^5 n)$ time.*

Rigid Motions. We can combine the algorithms implied by Lemma 3 and Lemma 9 to get a $(4+\epsilon)$-approximation of the minimum EMD under rigid motions in the following way: for each point-to-point translation $\vec{t}_{i\to j}$, compute a $(2+\epsilon)$-approximation of the optimum EMD between $A(\vec{t}_{i\to j})$ and B under rotations about b_j. The minimum over all these approximations gives a $2(2+\epsilon)$-approximation of $\mathrm{EMD}(\vec{t}_{\mathrm{opt}}, \theta_{\mathrm{opt}})$.

Lemma 10 *For any given $\epsilon > 0$, a $(4+\epsilon)$-approximation of the minimum EMD under rigid motions can be computed in $O((n^4m^2/\epsilon^2)\log^2(n/\epsilon))$ time.*

The $(2+\epsilon)$-approximation algorithm for rigid motions is based on similar ideas. Accoring to Observation 1, there exist two points a_i, b_j whose distance at $I_{\vec{t}_{\mathrm{opt}},\theta_{\mathrm{opt}}}$ is at most $\mathrm{EMD}(\vec{t}_{\mathrm{opt}}, \theta_{\mathrm{opt}})$. We place a grid of suitable size around each $\vec{t}_{i\to j}$. For each grid point $\vec{t}_g$ that is at most $\mathrm{EMD}(\vec{t}_{\mathrm{opt}}, \theta_{\mathrm{opt}})$ away from $\vec{t}_{i\to j}$ we compute a $(2+\epsilon)$-approximation of the optimum EMD betwenn $A(\vec{t}_g)$ and B under rotations about b_j. The minimum over all these approximations is within a factor of $(2+\epsilon)$ of $\mathrm{EMD}(\vec{t}_{\mathrm{opt}}, \theta_{\mathrm{opt}})$. Since we do not know $\mathrm{EMD}(\vec{t}_{\mathrm{opt}}, \theta_{\mathrm{opt}})$, we first compute a $(4+\epsilon)$-approximation of it as shown above. Algorithm RigidMotion(A, B, ϵ) is shown in Figure 2.

RigidMotion(A, B, ϵ):

1. For each pair of points $a_i \in A$ and $b_j \in B$ do:
   (a) Set the center of rotation, i.e. the origin, to be b_j by translating B appropriately.
   (b) Run Rotation$(A(\vec{t}_{i\to j}), B, 1)$ and let α_{ij} the cost value returned.

   Let $\alpha = \min_{ij} \alpha_{ij}$.
2. Let G be a uniform grid of spacing $c\alpha\epsilon$, where c is a suitable constant. For each pair of points $a_i \in A$ and $b_j \in B$ do:
   (a) Set the center of rotation, i.e. the origin, to be b_j by translating B appropriately.
   (b) Place a disk D of radius α around $\vec{t}_{i\to j}$.
   (c) For every grid point $\vec{t}_g \in D \cap G$ run Rotation$(A(\vec{t}_g), B, \epsilon/3)$ Let $\widetilde{\mathrm{EMD}}(\vec{t}_g)$ and θ^g_{apx} be the cost value and angle returned respectively.
3. Report the grid point $\vec{t}_{apx}$ that minimizes $\widetilde{\mathrm{EMD}}(\vec{t}_g)$, and the corresponding angle θ_{apx}.

Figure 2: Algorithm RigidMotion(A, B, ϵ).

Theorem 11 *For any given $\epsilon > 0$, a rigid motion $I_{\vec{t}_{apx},\theta_{apx}}$ such that $\mathrm{EMD}(\vec{t}_{apx}, \theta_{apx}) \le (2+\epsilon)\mathrm{EMD}(\vec{t}_{opt}, \theta_{opt})$ can be computed in $O((n^4m^2/\epsilon^4)\log^2(n/\epsilon))$ time.*

As in the case of translations, for equal weight sets we need to search for the optimal translation only around $\vec{t}_{C(A)\to C(B)}$. We set the center of rotation to be $C(B)$. Computing the 6-approximation of $\mathrm{EMD}(\vec{t}_{opt}, \theta_{opt})$ can be done simply by running Rotation$(A(\vec{t}_{C(A)\to C(B)}), B, 1)$. Similarly, we need to run Rotation$(A(\vec{t}_g), B, \epsilon/3)$ only for grid points $\vec{t}_g$ that are close to $\vec{t}_{C(A)\to C(B)}$.

Theorem 12 *If A and B have equal total weights, then, for any given $\epsilon > 0$, a rigid motion $I_{\vec{t}_{apx},\theta_{apx}}$ such that $\mathrm{EMD}(\vec{t}_{apx}, \theta_{apx}) \le (2+\epsilon)\mathrm{EMD}(\vec{t}_{opt}, \theta_{opt})$ can be computed in $O((n^3m/\epsilon^4)\log^2(n/\epsilon))$ time. For the minimum cost assignment problem under rigid motions the same approximation can be computed in $O((n/\epsilon)^{7/2}\log^5 n)$ time.*

References

[1] K. Grauman and T. Darell. Fast contour matching using approximate Earth Mover's Distance. In *Proc. of the IEEE Conf. Comp. Vision and Pattern Recognition*, pages I: 220-227, 2004.

[2] P. Giannopoulos and R. C. Veltkamp. A pseudometric for weighted point sets. In *Proc. of the 7th European Conf. Comp. Vision*, vol. 2352 of *LNCS*, pages 715–731, 2002.

[3] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows: Theory, Algorithms and Applications.* Prentice-Hall, 1993.

[4] D.S. Atkinson and P.M. Vaidya. Using geometry to solve the transportation problem in the plane. *Algorithmica*, 13:442–461, 1995.

[5] P. Bose, A. Maheshwari, and P. Morin. Fast approximations for sums of distances clutering and the Fermat–Weber problem. *Comput. Geom. Theory and Appl.*, 24:135–146, 2003.

[6] P. B. Callahan and S. R. Kosaraju. Faster algorithms for some geometric graph problems in higher dimensions. In *Proc. of the 4h Ann. ACM-SIAM Sympos. Discr. Algorithms*, pages 291–300, 1993.

[7] S. Cohen. *Finding Color and Shape Patterns in Images.* PhD thesis, Stanford University, Department of Computer Science, 1999.

[8] K. R. Varadarajan and P. K. Agarwal. Approximation algorithms for bipartite and non-bipartite matching in the plane. In *Proc. of the 10th Ann. ACM-SIAM Sympos. Discr. Algorithms*, pages 805–814, 1999.

[9] R. Typke, P. Giannopoulos, R. C. Veltkamp, F. Wiering, and R. van Oostrum. Using transportation distances for measuring melodic similarity. In *Proc. of the 4th Int. Conf. Music Inf. Retrieval* , pages 107–114, 2003.