Similaritiy Search in Semialgebraic Pattern Spaces

(Extended Abstract)

Christian Knauer [a]

[a] *Institut für Informatik, Freie Universität Berlin, Takustraße 9, D-14195 Berlin, Germany*

Abstract

We describe a general technique to construct data structures for similarity search in semialgebraic pattern spaces. These spaces capture most known combinations of geometric patterns (e.g., point sets, polygons, polygonal curves) and geometric distance measures for them (e.g. Hausdorff-distance, area of overlap, Fréchet-distance) together with their quotients under various transformation classes (e.g., translations, rigid motions) and they provide the first non-trivial exact search structures in these settings.

Key words: Computational geometry, Shape matching, Similarity queries, Data structures

1. Introduction

Similarity search is a much-studied and practically important type of problem: *Given a set $\mathcal{D}$ of n patterns from a suitable class of valid geometric patterns (e.g., polygonal curves), preprocess them in such a way that we can determine quickly for a query pattern Q, which of the n preprocessed patterns is most similar to the query object (this is called a* similarity query*). We assume that we have an appropriate distance measure δ (e.g., the smallest Fréchet distance that can be achieved under translations) to asses the similarity of two patterns.*

Satisfactory algorithmic results exist only in the case that the patterns can be encoded in a Euclidean space, or an L_1- or L_∞-space such that δ is the corresponding metric [9], or in the case that they can be embedded in such spaces with a low distortion [7,6]. Then we have available the very powerful techniques of Voronoi decompositions. Some algorithms were formulated with 'vantage points' or similar devices [2,3,5,10], but then there are no general performance bounds: only under additional assumptions that enforce in some way that the distance measure is a metric that is 'similar' to a Euclidean metric it is possible to obtain nontrivial bound on the performance of these algorithms [9].

In this paper, we study the case where the size of the individual patterns is small, compared to n. To be more precise, we assume that $|I| = O(1)$ for all $I \in \mathcal{D}$. In that case the distances $\delta(Q, I)$, can be computed in $O(1)$ time (for reasonable δ). So the query can be answered in $O(n)$ time without additional storage and preprocessing, but up to now there are no algorithms and data structures that allow such queries with a nontrivial query time among *preprocessed* pattern sets. We describe a general technique to construct data structures for similarity queries for many combinations of geometric patterns (e.g., point sets, polygons, polygonal curves) and geometric distance measures (e.g. Hausdorff-distance, area of overlap, Fréchet-distance) together with their quotients under various transformation classes (e.g., translations, rigid motions). The solution achieves sublinear query time with quadratic preprocessing time and storage and provides the first non-trivial search structures in these settings.

A *pattern space* $\Pi = (\Omega, \delta)$ of dimension d is a set of geometric objects Ω, where each object can be described by exactly d real parameters, together with a distance measure δ that maps pairs of objects to non-negative real numbers. Usually

Email address: Christian.Knauer@inf.fu-berlin.de (Christian Knauer).

we identify Ω with $\mathbb{R}^d$, the *parameter space* of Π. The objects in Ω are called *patterns*. Intuitively, when $\delta(P, I)$ is small, we consider the patterns P and I to be similar. Note that we do not demand that δ has some special properties (like being a metric, etc.). As an example consider polygonal chains in the plane with at most r vertices and the Fréchet-distance as a distance measure. Each such chain can be encoded by a sequence of at most $2r$ real numbers, the coordinates of the vertices of the chain. If a chain has less than r vertices we can simply pad this description by repeating the last vertex; note that this padding does not interfere with the distance function.

A pattern space $\Pi = (\mathbb{R}^d, \delta)$ is called *semialgebraic*, if the set $\mathcal{F}_\Pi := \{(P, I, \epsilon) \in \mathbb{R}^d \times \mathbb{R}^d \times \mathbb{R} \mid \delta(P, I) \leq \epsilon\}$ is semialgebraic, i.e., there is a boolean formula B in s boolean variables $z_1, \ldots, z_s$, and there are s polynomials $g_1, \ldots, g_s$ in $(2d + 1)$ real variables $\boldsymbol{x}_P, \boldsymbol{x}_I, x_\epsilon$ ($\boldsymbol{x}_P$ and $\boldsymbol{x}_I$ are actually sequences of d variables each), such that

$$\begin{aligned}&\delta(P, I) \leq \epsilon \iff \\ &\qquad B(z_1 \leftarrow [g_1(\boldsymbol{x}_P \leftarrow P, \boldsymbol{x}_I \leftarrow I, x_\epsilon \leftarrow \epsilon) \geq 0], \\ &\qquad\qquad \ldots \\ &\qquad\quad z_s \leftarrow [g_s(\boldsymbol{x}_P \leftarrow P, \boldsymbol{x}_I \leftarrow I, x_\epsilon \leftarrow \epsilon) \geq 0]) \\ &\text{is true}\end{aligned}$$

($\leftarrow$ denotes variable substitution and $[X]$ is the truth value of the predicate X).

As an example consider the set of all point sets in the plane with at most r points each and the directed Hausdorff-distance as a distance measure. Recall that for compact sets $P, I \subseteq \mathbb{R}^d$ the *directed Hausdorff distance from P to I*, is defined as $h(P, I) := \max_{x \in P} \min_{y \in I} ||x - y||$.

This is a pattern space of dimension $2r$, since each pattern P can be encoded by a sequence of at most $2r$ real numbers, the coordinates of the points in P. If P has less than r points we can simply pad this description by repeating some point; note that this padding does not interfere with the distance function. To see that it is actually a semialgebraic pattern space, note that for two finite point sets P, I and $\epsilon > 0$ we have that

$$\begin{aligned} h(P, I) \leq \epsilon &\iff \forall p \in P \exists i \in I : ||p - i||^2 - \epsilon^2 \leq 0 \\ &\iff \bigwedge_{p \in P} \exists i \in I : ||p - i||^2 - \epsilon^2 \leq 0 \\ &\iff \bigwedge_{p \in P} \bigvee_{i \in I} ||p - i||^2 - \epsilon^2 \leq 0. \end{aligned}$$

We will see in Section 3 that the notion of a semialgebraic pattern space captures most known combinations of geometric patterns (e.g., point sets, polygons, polygonal curves) and geometric distance measures (e.g. Hausdorff-distance, area of overlap, Fréchet-distance) together with their quotients under various transformation classes.

In this paper, we study the problem of similarity search among patterns from a semialgebraic pattern space where the size of the individual patterns is small, compared to their number: *Given a data set $\mathcal{D}$ that consists of* n *patterns from a semialgebraic pattern space $\Pi = (\Omega, \delta)$ of dimension* d*, where* d *is* constant*, preprocess $\mathcal{D}$ into a data structure to answer the following kind of* similarity queries*: For a query pattern $Q \in \Omega$, determine*

- $\Delta(Q, \mathcal{D}) := \operatorname{minarg}_{I \in \mathcal{D}} \delta(Q, I)$*, the set of patterns in $\mathcal{D}$ to which Q has the smallest possible distance, and*
- $\Delta(\mathcal{D}, Q) := \operatorname{minarg}_{I \in \mathcal{D}} \delta(I, Q)$*, the set of patterns in $\mathcal{D}$ that have the smallest possible distance to Q.*

Since d is constant, the distances $\delta(Q, I)$, can be computed in $O(1)$ time (for reasonable δ). So these queries can be answered in $O(n + k)$ time (where k is the size of the answer) without additional storage and preprocessing, but up to now there are no algorithms and data structures that allow such queries with a nontrivial query time among *preprocessed* pattern sets.

We will also consider the decision version of the similarity queries, called *ϵ-similarity queries*, where we are given an additional parameter $\epsilon > 0$, and we want to determine

- $\Delta(Q, \mathcal{D}, \epsilon) := \{I \in \mathcal{D} \mid \delta(Q, I) \leq \epsilon\}$, the set of patterns in $\mathcal{D}$ to which Q has distance at most ϵ, and
- $\Delta(\mathcal{D}, Q, \epsilon) := \{I \in \mathcal{D} \mid \delta(I, Q) \leq \epsilon\}$, the set of patterns in $\mathcal{D}$ that have distance at most ϵ to Q.

Again the brute-force approach can answer these queries in $O(n + k)$ time, but there are no data structures that allow such queries with a nontrivial query time among preprocessed pattern sets.

2. Similaritiy queries in pattern spaces

In this abstract we only consider the decision version of the similarity queries. We describe a data structure that answers $\Delta(Q, \mathcal{D}, \epsilon)$-queries (the case of $\Delta(\mathcal{D}, Q, \epsilon)$-queries is completely symmetric). The same techniques apply to similarity queries as well and yield similar results. Our main result is the following

Theorem 1 *Suppose we are given a set $\mathcal{D}$ that consists of n patterns from a semialgebraic pattern space Π of dimension $d = O(1)$. Then we can build in $O(n^2)$ time a data structure of size $O(n^2)$ that answers ϵ-similarity queries in $O(n^{1-1/(2d-3)} + k)$ time, where k is the size of the answer.*

PROOF. The construction works in two steps. We first describe a data structure that can be built in $O(n^{2d-2})$ time (and requires the same amount of space) and can answer a query in $O(\log n + k)$ time, where k is the size of the answer. Then we use a simple partitioning approach to yield the desired result (in fact we prove a somewhat stronger tradeoff that implies the Theorem).

Since $\Pi = (\Omega, \delta)$ is a semialgebraic pattern space of dimension $d = O(1)$, there is a boolean formula B in $s = O(1)$ boolean variables $z_1, \ldots, z_s$, and there are s polynomials $g_1, \ldots, g_s$ in $(2d+1)$ real variables $\boldsymbol{x}_P, \boldsymbol{x}_I, x_\epsilon$, such that

$$\begin{aligned}\delta(Q, I) \leq \epsilon \iff & \\ & B(z_1 \leftarrow [g_1(\boldsymbol{x}_Q \leftarrow Q, \boldsymbol{x}_I \leftarrow I, x_\epsilon \leftarrow \epsilon) \geq 0], \\ & \qquad \cdots \\ & z_t \leftarrow [g_t(\boldsymbol{x}_Q \leftarrow Q, \boldsymbol{x}_I \leftarrow I, x_\epsilon \leftarrow \epsilon) \geq 0]).\end{aligned}$$

For $j = 1, \ldots, s$ and for all $I \in \mathcal{D}$, we compute the $(d+1)$-variate polynomials

$$g_{j,I}(\boldsymbol{x}_Q, x_\epsilon) := g_j(\boldsymbol{x}_Q, \boldsymbol{x}_I \leftarrow I, x_\epsilon).$$

This is done by substituting I for $\boldsymbol{x}_I$ in g_j and therefore takes time $O(1)$. The total time to compute all these polynomials is $O(n)$.

For the $g_{j,I}$ we compute a subdivision Ξ of $\mathbb{R}^{d+1}$ with the property that the sign of each $g_{j,I}$ remains constant on each cell of the subdivision. Such a subdivsion of size $O(n^{2d-3})$ can be computed in $O(n^{2d-3})$ time, along with a point-location data-structure $\mathcal{L}(\Xi)$ for the subdivision with $O(\log n)$ query-time, c.f., [8]; the computation also yields for each cell $\chi \in \Xi$ a point $(Q_\chi, \epsilon_\chi) \in \chi$.

In a next step we process each cell $\chi \in \Xi$ in turn and compute a set $\mathcal{D}_\chi \subset \mathcal{D}$, which is initially empty. For all $I \in \mathcal{D}$ we do the following: First compute for $j = 1, \ldots, s$ the numbers

$$\gamma_{i,I,\chi} := g_{j,I}(\boldsymbol{x}_Q \leftarrow Q_\chi, x_\epsilon \leftarrow \epsilon_\chi).$$

Next, we compute the truth-value

$$B_{I,\chi} := B(z_1 \leftarrow [\gamma_{1,I,\chi} \geq 0], \ldots, z_s \leftarrow [\gamma_{s,I,\chi} \geq 0]).$$

If $B_{I,\chi}$ is true, we have that $\delta(Q_\chi, I) \leq \epsilon_\chi$ and we add I to $\mathcal{D}_\chi$. We augment the data-structure $\mathcal{L}(\Xi)$ by storing the set $\mathcal{D}_\chi$ for each cell $\chi \in \Xi$. The total time needed to compute it is $O(n^{2d-2})$, and it needs $O(n^{2d-2})$ space.

To answer a query $(Q, \epsilon) \in \Omega \times \mathbb{R}$, we proceed as follows: Using $\mathcal{L}(\Xi)$ we locate the cell $\chi \in \Xi$ with $(Q, \epsilon) \in \chi$ in $O(\log n)$ time. Since the sign of the $g_{j,I}$'s is constant on each cell of Ξ, we have that $\delta(Q_\chi, I) \leq \epsilon_\chi$ iff $\delta(Q, I) \leq \epsilon$, so we can report $\mathcal{D}_\chi$ as the answer to the query. The total time required is $O(\log n + k)$, where $k = |\mathcal{D}_\chi|$ is the size of the answer.

Now we use a simple partitioning approach to yield the desired result: We split $\mathcal{D}$ into $g = \Theta(n/m)$ groups $\mathcal{D}_1, \ldots, \mathcal{D}_g$, each of size $\Theta(m)$, where $1 \leq m \leq n$ is a suitable parameter (see below). Then we build the aforementioned data-structure for each $\mathcal{D}_i$ separately. To answer a query $(Q, \epsilon) \in \Omega \times \mathbb{R}$, we query each data-structure separately and combine the individual answers. The total time needed to compute all the structures is $O(gm^{2d-2}) = O(nm^{2d-3})$ (this is also the total space requirement), and the query time is $O(g \log n) = O((n/m) \log n)$. Setting $m = n^{1/(2d-3)}$ proves the claimed result. □

3. More semialgebraic pattern spaces

In the following we show that the notion of a semialgebraic pattern space captures many combinations of geometric patterns (e.g., point sets, polygons, polygonal curves) and geometric distance measures (e.g., Hausdorff-distance, area of overlap, Fréchet-distance) together with their quotients under various transformation classes (e.g., translations, rigid motions).

Since the set accepted by an algebraic decision tree is semialgebraic we get the following

Lemma 2 *Let Π be a pattern space. If $\mathcal{F}_\Pi$ can be decided by an algorithm in the algebraic decision tree model, then Π is semialgebraic.*

It is straightforward to verify that many of the algorithms for deciding the most prominent distance measures can be implemented in the algebraic decision tree model, c.f., [1]. This shows for example that polygonal curves on k vertices in $\mathbb{R}^d$

wrt. the Fréchet-distance constitute a semialgebraic pattern space.

Let $\Pi = (\mathbb{R}^d, \delta)$ be a semialgebraic pattern space, and let $t : \mathbb{R}^d \times \mathbb{R}^d \times \mathbb{R}^f \to \mathbb{R}^d \times \mathbb{R}^d$ be a function. This function induces a new distance measure and thus a new pattern space $\Pi_t := (\mathbb{R}^d, \delta_t)$ as follows:

$$\delta_t(P, I) := \min_{v \in \mathbb{R}^f} \delta(t(P, I, v)).$$

We call δ_t the *quotient of* δ *under* t and f the *number of degrees of freedom of* t.

As an example consider again the pattern space of all point sets in the plane with at most r points each and the directed Hausdorff-distance $h(,)$ as a distance measure. The function $t(P, I, v) := (P + v, I)$ has two degrees of freedom and we have that $h_t(P, I) = \min_{v \in \mathbb{R}^2} h(P + v, I)$ is the smallest directed Hausdorff distance that a *translate* of P has to I.

Theorem 3 *If t is rational, then Π_t is semialgebraic. In that case a semialgebraic description of $\mathcal{F}_{\Pi_t}$ can effectilvey be computed from such a description of $\mathcal{F}_\Pi$.*

PROOF. First, observe that

$$\delta_t(Q, I) \le \epsilon \iff \exists v \in \mathbb{R}^f : \delta(t(Q, I, v)) \le \epsilon.$$

Since $\Pi = (\Omega, \delta)$ is a semialgebraic pattern space and t is rational, there is a boolean formula B in s boolean variables $z_1, \ldots, z_s$, and there are s polynomials $g_1, \ldots, g_s$ in $(2d + 1)$ real variables $\boldsymbol{x}_P, \boldsymbol{x}_I, x_\epsilon$, such that

$$\begin{aligned}
&\delta_t(Q, I) \le \epsilon \iff \\
&\quad \exists v B(z_1 \leftarrow [g_1(t(\boldsymbol{x}_Q \leftarrow Q, \boldsymbol{x}_I \leftarrow I, \boldsymbol{x}_v \leftarrow v), \\
&\qquad\qquad x_\epsilon \leftarrow \epsilon) \ge 0], \\
&\qquad\qquad \ldots \\
&\qquad z_s \leftarrow [g_s(t(\boldsymbol{x}_Q \leftarrow Q, \boldsymbol{x}_I \leftarrow I, \boldsymbol{x}_v \leftarrow v), \\
&\qquad\qquad x_\epsilon \leftarrow \epsilon) \ge 0]
\end{aligned}$$

($\boldsymbol{x}_v$ is a sequence of f new variables).

In general $g_i(t(),)$ is not a polynomial. However, since t is rational, the conditions '$g_i(t(),) \ge 0$' can be rewritten as equivalent *polynomial* inequalities. This shows that $\mathcal{F}_{\Pi_t}$ is a Tarski-set, and therefore semialgebraic. Using standard quantifier elimination techniques [4], a semialgebraic description of $\mathcal{F}_{\Pi_t}$ can effectilvey be computed. □

If the dimension d of Π is $O(1)$ (relative to $n = |\mathcal{D}|$), then the dimension d' of Π_t is also $O(1)$ (unfortunately, in general, d' is doubly exponential in d) and a semialgebraic description of $\mathcal{F}_{\Pi_t}$ can be computed in $O(1)$ time.

This shows for example that polygonal curves on k vertices in $\mathbb{R}^d$ wrt. the smallest Fréchet-distance that can be attained under rigid motions constitute a semialgebraic pattern space of dimension $O(1)$ if $k, d = O(1)$.

Acknowledgments. The author would like to thank Peter Braß for fruitful discussion on the subject.

References

[1] H. Alt and L. J. Guibas. Discrete geometric shapes: Matching, interpolation, and approximation. In J.-R. Sack and J. Urrutia, editors, *Handbook of Computational Geometry*, pages 121–153. Elsevier Science Publishers B.V. North-Holland, Amsterdam, 2000.

[2] S. Brin. Near neighbor search in large metric spaces. In *The VLDB Journal*, pages 574–584, 1995.

[3] K. L. Clarkson. Nearest neighbor queries in metric spaces. In *Proc. 29th Annu. ACM Sympos. Theory Comput.*, pages 609–617, 1997.

[4] G. E. Collins. Quantifier elimination for real closed fields by cylindrical algebraic decomposition. In *Proc. 2nd GI Conference on Automata Theory and Formal Languages*, volume 33 of *Lecture Notes Comput. Sci.*, pages 134–183. Springer-Verlag, 1975.

[5] E.Chávez, G. Navarro, R.Baeza-Yates, and J.Marroquín. Searching in metric spaces. Technical Report TR/DCC-99-3, Dept. of Computer Science, Univ. of Chile, 1999.

[6] P. Indyk. Approximate nearest neighbor algorithms for frechet distance via product metrics. In *Proceedings of the eighteenth annual symposium on Computational geometry*, pages 102–106. ACM Press, 2002.

[7] P. Indyk and M. Farach-Colton. Approximate nearest neighbor algorithms for Hausdorff metrics via embeddings. In *Proc. 40th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 171–180, 1999.

[8] V. Koltun. Almost tight upper bounds for vertical decompositions in four dimensions. In *Proc. 42nd Annu. IEEE Sympos. Found. Comput. Sci.*, 2001.

[9] S. A. Nene and S. K. Nayar. A simple algorithm for nearest neighbor search in high dimensions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:989–1003, 1997.

[10] P. N. Yianilos. Excluded middle vantage point forests for nearest neighbor search. Technical report, NEC Research Institute, 1999.