An FPTAS for an Elastic Shape Matching Problem with Cyclic Neighborhoods

Christian Knauer, Luise Sommer, and Fabian Stehn

Institut für Informatik, Universität Bayreuth, Bayreuth, Germany

Abstract

The elastic geometric shape matching (EGSM) problem class is a generalisation of the well-known geometric shape matching problem class: Given two geometric shapes, the 'pattern' and the 'model', find a *single* transformation from a given transformation class that, if applied to the pattern, minimizes the distance between the transformed pattern and the model with respect to a suitable distance measure.

In EGSM, the pattern is divided into subshapes that are transformed by a 'transformation ensemble', i.e., a set of transformations. The goal is to minimize the distance between the union of the transformed subpatterns and the model in object space as well as the distance between specific transformations of the ensemble. The 'neighborhood graph' encodes which translations should be similar.

We present an FPTAS for EGSM instances for point sequences under translations with fixed correspondence where the neighborhood graph is a simple cycle.

1 Introduction

In classical geometric shape matching (GSM) problems, one is given a pattern P and a model Q, both from a class $\mathcal{S}$ of geometric shapes, along with a distance measure $d : \mathcal{S} \times \mathcal{S} \mapsto \mathbb{R}_0^+$. The task is to find a single transformation t from a given transformation class $\mathcal{T}$ acting on $\mathcal{S}$, so that $d(t(P), Q)$ is minimized.

Matching geometric shapes is a problem that occurs in many applications such as character recognition, logo detection, human-computer-interaction, etc., and in a variety of different scientific fields, e.g., robotics, computer aided medicine, drug design, etc., and thus has already received a considerable amount of attention. We refer to the survey papers by Alt et al. [1] and Veltkamp et al. [4] for an extensive overview.

Many *geometric registration problems* (where the task is to align two shapes in different coordinate systems), e.g., between the coordinate system of an operation theatre and the coordinate system of a $3D$-model of a patient acquired during a pre-operative MRI scan, can be modelled as a GSM instance by appropriately choosing $\mathcal{S}$ and d. There, the transformation that minimizes the distance between both geometric shapes is then used as the mapping from the pattern space into the model space.

In many applications, where local distortions and complex deformations may occur, such as soft tissue registrations, GSM problems are too restrictive because a single transformation is chosen to match the entire pattern to the model. This issue is addressed by the *elastic geometric shape matching* (EGSM) framework, a generalisation of GSM. Here, the pattern is partitioned into subshapes and instead of one single transformation, a so-called transformation ensemble is computed. Each subshape of the pattern is transformed by an individual transformation of the ensemble in order to minimize the distance between the transformed pattern and the model. Also, the 'consistency' of the ensemble is guaranteed by forcing the transformations acting on some neighboring subshapes of the pattern to be similar with respect to a suitable similarity measure for the class of transformations at hand. The dependencies between the transformations of an ensemble are encoded in a so-called neighborhood graph.

34th European Workshop on Computational Geometry, Berlin, Germany, March 21–23, 2018.
This is an extended abstract of a presentation given at EuroCG'18. It has been made public for the benefit of the community and should be considered a preprint rather than a formally reviewed paper. Thus, this work is expected to appear eventually in more final form at a conference with formal proceedings and/or in a journal.

In [3], the authors considered several variants of this problem for different distance measures and graph families, including an algorithm that solves a variant of the problem for trees where only translations in a fixed direction are allowed in $O(n^2 \log n)$ time. In this paper, we focus on EGSM for point sequences under translations with fixed correspondence where the neighborhood graph is a simple cycle.

Problem Statement. In the following, everything is stated in $\mathbb{R}^2$, $\|\cdot\|$ denotes the Euclidean norm and translations are represented by translation vectors. All index arithmetic is modulo n.

▶ **Problem 1.1.** *Given two sequences of points $P = (p_0, \dots, p_{n-1})$ (the pattern) and $Q = (q_0, \dots, q_{n-1})$ (the model), find a sequence of translations $T = (t_0, \dots, t_{n-1})$, so that the function $\gamma(T, P, Q) \coloneqq \max\{\max_{0 \le i < n} \|q_i - (p_i + t_i)\|, \max_{0 \le i < n} \|t_i - t_{i+1}\|\}$ is minimized.*

Measuring the distance of the points $(t_i + p_i)$ and q_i in model space is the same as measuring the distance of the points t_i and $q_i - p_i$ in translation space. This is why Problem 1.1 can be studied in translation space entirely: Let $c_i \coloneqq q_i - p_i$ for $0 \le i < n$ and $C \coloneqq (c_0, \dots, c_{n-1})$. The function $\gamma(T, P, Q)$ can be rewritten as $\gamma(T, C) \coloneqq \max\{\max_{0 \le i < n} \|c_i - t_i\|, \max_{0 \le i < n} \|t_i - t_{i+1}\|\}$. We refer to points in translation space (i.e. translations) simply as points.

Basic Definitions. Let $c, u, v \in \mathbb{R}^2$ and $r > 0$. $D_r(c)$ denotes the disk with radius r centered in c and $\partial D_r(c)$ denotes its boundary. We define $I_r(c, u, v) \coloneqq D_r(c) \cap D_r(u) \cap D_r(v)$.
For a given sequence $C = (c_0, \dots, c_{n-1})$, we define $\delta^* \coloneqq \min_T \gamma(T, C)$. We call a sequence of points $T = (t_0, \dots, t_{n-1})$ *δ-admissible (for C)*, iff $\gamma(T, C) \le \delta$. A sequence, that is δ^*-admissible is called an *optimal* sequence. We will use the symbol T^* to denote an optimal sequence. A point t is called *(δ, i)-admissible (for C)*, iff there is a δ-admissible sequence T so that $T = (t_0, \dots, t_i = t, \dots, t_{n-1})$. Strictly speaking, δ^* and the concepts of admissibility depend on C, but since C is part of the input, we refrain from including C in the notation.

Previous Work and our Contribution. In [3] and in [2], the authors discussed several variants of EGSM problems. However, there are no results regarding problem instances, where the neighborhood graph is a simple cycle. In particular, there is no literature, that deals with efficient exact or approximation algorithms for Problem 1.1 and we do not know, if the problem is NP-hard.
In this paper, we provide an FPTAS for Problem 1.1 and prove that it computes a $(1+\epsilon)$-approximation to δ^* in $O\left(\epsilon^{-1/2} \left(\log \epsilon^{-1}\right)^2 n^3 \log n\right)$ time or in $O\left(\left(\log \epsilon^{-1}\right) \epsilon^{-2} n^2 \log n\right)$ time for some $\epsilon > 0$.

2 Our Results

Due to space limitations, the proofs of all lemmata and all figures have been omitted.
We define $\delta^{(3)} \coloneqq \gamma(C, C)$.

▶ **Lemma 2.1.** *C gives a 3-approximation to δ^*, i.e., $\delta^{(3)} \le 3\delta^*$.*

Lemma 2.1 is the basis for the construction of our FPTAS, since it implies, that every $(\delta^*, 0)$-admissible point lies within the disk $D_{\delta^{(3)}}(c_0)$. A simple way to get a $(1+\epsilon)$-approximation to δ^* for some $\epsilon > 0$ is to sample t_0 from a dense enough ϵ_{grid}-grid (a grid where the distance between samples is at most ϵ_{grid}) that covers $D_{\delta^{(3)}}(c_0)$. We call the points of this grid *translation-samples*. Here, $\epsilon_{\text{grid}} = \Theta\left(\epsilon\delta^{(3)}\right)$. We also sample the value δ of the objective function on the interval $[\frac{1}{3}\delta^{(3)}, \delta^{(3)}]$ with sample-distance $\epsilon_{\text{obj}} = \Theta\left(\epsilon\delta^{(3)}\right)$. We call the samples on $[\frac{1}{3}\delta^{(3)}, \delta^{(3)}]$ *radius-samples*. We then test for every radius-sample δ, whether there exists a solution T' so that $\gamma(T', C) \leq \delta$ and a translation-sample is the ith component of T'. This test is a variant of a problem that has already been studied in [3], where the authors give an algorithm that solves this problem for paths in $O(n^2 \log n)$. Consequently, this simple FPTAS runs in $O\left(\epsilon^{-3} n^2 \log n\right)$ time.

This result can be improved in several ways. The first obvious improvement is to perform a binary search on $[\frac{1}{3}\delta^{(3)}, \delta^{(3)}]$, which improves the run-time to $O\left(\left(\log \epsilon^{-1}\right) \epsilon^{-2} n^2 \log n\right)$.

The second idea is based on Lemma 3.1 below, which says, that for every $\delta \geq \delta^*$, there is a δ-admissible sequence T containing a point t_i that lies on $\partial D_\delta(c_i)$ for some i. Consequently, we do not have to sample the whole disk $D_{\delta^{(3)}}(c_i)$ for the current radius-sample δ, but to only sample $\partial D_\delta(c_i)$. Unfortunately, there is no way to identify the disks (the c_i) with this property, hence it is no longer possible to pick an arbitrary disk and sample it, but we have to sample the boundary of all disks. This changes the run-time to $O\left(\left(\log \epsilon^{-1}\right) \epsilon^{-1} n^3 \log n\right)$. Of course, this is only an improvement if $\epsilon^{-1} \gg n$. On the other hand, this strategy enables us to apply another modification: We can approximate each $\partial D_\delta(c_i)$ by a regular polygon with $O\left(\epsilon^{-1/2}\right)$ vertices. Due to the convexity of the problem, we can then perform a binary search on the edges of this polygon and get an FPTAS that runs in $O\left(\epsilon^{-1/2}\left(\log \epsilon^{-1}\right)^2 n^3 \log n\right)$ time. This gives us the following tradeoff between precision and input size:

▶ **Theorem 2.2.** *We can compute a $(1+\epsilon)$-approximation to δ^* in $O\left(\left(\log \epsilon^{-1}\right) \epsilon^{-2} n^2 \log n\right)$ time or in $O\left(\epsilon^{-1/2}\left(\log \epsilon^{-1}\right)^2 n^3 \log n\right)$ time.*

Since it is very clear how to implement the approximation when sampling the interior of $D_{\delta^{(3)}}(c_0)$, we elaborate on the improvements of the second strategy.

3 A Detailed Description

On (δ, i)-Admissible Points. The reason why it suffices to sample the boundaries of all disks rather than sampling the interior of one disk with a grid is, that any optimal solution T^* contains a key-point: A (δ^*, i)-admissible point t_i^* of $T^* = (t_0^*, \ldots, t_{n-1}^*)$ is called a *key-point*, iff $I_{\delta^*}(c_i, t_{i-1}^*, t_{i+1}^*) = \{t_i^*\}$ and $t_i^* \in \partial D_{\delta^*}(c_i)$.

▶ **Lemma 3.1.** *For every optimal sequence $T^* = (t_0^*, \ldots, t_{n-1}^*)$, there is an index $0 \leq i < n$ so that t_i^* is a key-point.*

On Computing δ^*. There is at least one index $0 \leq i < n$ for every $T^* = (t_0^*, \ldots, t_{n-1}^*)$, so that t_i^* is a key-point, which implies, that $t_i^* \in \partial D_{\delta^*}(c_i)$. Since we have no way of determining the index i, so that t_i^* is a key-point, the boundaries of all disks have to be sampled in order to find a suitable approximation to t_i^*. Since we do not know the optimal radius δ^* either, we have to sample the boundary of all disks for dense enough radius-samples in $[\frac{1}{3}\delta^{(3)}, \delta^{(3)}]$. For every index i, let δ_i^* be the smallest (not necessarily a sample-radius) value, so that there is a (δ_i^*, i)-admissible point $t_i \in \partial D_{\delta_i^*}(c_i)$. In order to compute δ^* from the values $\delta_0^*, \ldots, \delta_{n-1}^*$, we need the following observation:

▶ **Lemma 3.2.** $\delta^* = \min_{0 \leq i < n} \delta_i^*$.

Consequently, in order to find δ^*, it suffices to compute δ_i^* for all $0 \leq i < n$.

Let $T^{(\epsilon)}$ be a solution so that $\delta^{(\epsilon)} := \gamma(T^{(\epsilon)}, C) \leq (1+\epsilon)\delta^*$. Let $t_i^{(\epsilon)}$ denote a $(1+\epsilon)$-approximation to a (δ_i^*, i)-admissible point t_i with $t_i \in \partial D_{\delta_i^*}(c_i)$. Let $\delta_i^{(\epsilon)}$ be the radius-sample of $t_i^{(\epsilon)}$. In order to prove that a binary search for δ_i^* on $[\frac{1}{3}\delta^{(3)}, \delta^{(3)}]$ works for every index i, we need one more characteristic of (δ, i)-admissible points.

▶ **Lemma 3.3.** *Let $\bar{\delta} \geq 0$ be so that there is a $(\bar{\delta}, i)$-admissible point $t_i \in \partial D_{\bar{\delta}}(c_i)$. Then there is at least one (δ, i)-admissible point on $\partial D_\delta(c_i)$ for all $\delta \geq \bar{\delta}$.*

Consequently, a binary search for δ_i^* on $[\frac{1}{3}\delta^{(3)}, \delta^{(3)}]$ can be carried out for every index $0 \leq i < n$ and it remains to determine a suitable sample-distance ϵ_{obj}: Since we aim for a $(1+\epsilon)$-approximation, we have to guarantee, that $\delta^{(\epsilon)} \leq (1+\epsilon)\delta^*$. Hence, it suffices to find some $\delta_i^{(\epsilon)} \in [\delta_i^*, \delta_i^* + \epsilon\delta_i^*]$ for each $0 \leq i < n$ and we have to choose ϵ_{obj} (the density of the radius-samples) subject to ϵ and $\delta^{(3)}$. The analysis on how ϵ_{obj} has to be chosen exactly will be carried out in Lemma 3.6, since it also depends on our final improvement, in particular on the polygons that will be used to approximate the boundaries of all disks.
In order to describe the final improvement in more detail, we need to briefly explain the 'propagation along the path' decision algorithm $A1$ of [3], that, given a radius-sample δ, a translation-sample t_i and an index i, decides, whether t_i is (δ, i)-admissible.

An Algorithm for Paths. A point $t_i \in D_\delta(c_i)$ is (δ, i)-admissible iff there are points $t_{i+1}, \ldots, t_{n-1}$, $t_0, \ldots, t_{i-1}$ so that all constraints encoded in $\gamma(T, C)$ are met. This chain of constraints can be interpreted as a path that starts and ends at t_i. In [3], the authors introduced an algorithm, that solves this problem for the case that only translations in a fixed direction are allowed. This algorithm can also be applied in our setting and then runs in $O(n^2 \log n)$ time and space. Starting at one end of the path, the basic idea is to propagate 'admissible regions', i.e., sets of translations that satisfy the current prefix of constraints, along the path. This is done by inflating them (i.e., computing the Minkowski sum of the region at hand and D_δ) and intersecting the result with the admissible region encoded in the subsequent node of the path. This strategy can be applied iteratively until either t_i is met again (in which case the algorithm returns YES) or the intersection of two regions is empty at some point. In this case the algorithm returns NO along with the tuple $(k(t_i), \mu(t_i))$, where $k(t_i)$ is the index of the first node that was not reached, and $\mu(t_i)$ is the Euclidean distance between the inflated version of the last non-empty admissible region and its succeeding admissible region.

Approximating the Boundary of a Disk with a Polygon. The simplest approach that tests, if there is a $(1+\epsilon)$-approximation to a key-point on $\partial D_\delta(c_i)$ is to pick $k = \Theta\left(\epsilon^{-1}\delta^{(3)}\right)$ suitably distributed translation-samples on $\partial D_\delta(c_i)$ and propagate all of them according to algorithm $A1$. In that way, $O(k)$ propagations (i.e., calls to algorithm $A1$) have to be carried out. This number can be reduced to $O(k^{1/2} \log k)$ by exploiting the convex structure of the admissible regions that occur during the propagation process: The main idea is to approximate $\partial D_\delta(c_i)$ by a regular polygon with $O(k^{1/2})$ vertices and to perform a binary search on each of its edges with a sample-distance that depends on ϵ and $\delta^{(3)}$.
Let $P_{\delta,p}(c_i)$ (or $P_\delta(c_i)$ in short) denote the inscribing regular polygon of $\partial D_\delta(c_i)$ with p vertices. By a slight abuse of notation, we identify $P_\delta(c_i)$ with its boundary, since we solely operate on the boundary of the polygons at hand. Also, let all such polygons be concentric.

▶ **Lemma 3.4.** *Let $p := \lceil 3^{1/4}\pi\epsilon^{-1/2} \rceil$ and let the edges of $P_\delta(c_i)$ be sampled with sample-distance ϵ_{edge} so that $\epsilon_{edge} \leq \frac{1}{3}\delta^{(3)}\epsilon$. Then, there is a translation-sample $t \in P_\delta(c_i)$ for every point $u \in \partial D_\delta(c_i)$ so that $\|t - u\| \leq \frac{1}{3}\delta^{(3)}\epsilon$.*

Strictly speaking, p depends on ϵ, but we refrain from including ϵ in the notation.
In the remainder, we will show, that the binary search among the samples on one edge of $P_\delta(c_i)$ can be carried out in $O\left(\log\epsilon^{-1}n^2\log n\right)$ time. Here $O\left(n^2\log n\right)$ is the time that is needed to carry out the propagation process for a single translation-sample t by algorithm A1. Since this approach builds on several properties of the tuple $(k(t),\mu(t))$ returned by algorithm A1, we have to introduce some of them first: The following lemma describes the dependency of the tuple on the translation-samples of one edge of $P_\delta(c_i)$.

▶ **Lemma 3.5.** *Let s and s' be two* NO*-instances of algorithm A1 for a given radius-sample δ, i.e., $A1(s,\delta,i)=(\text{NO},(k(s),\mu(s)))$ and $A1(s',\delta,i)=(\text{NO},(k(s'),\mu(s')))$, and let $t\in\overline{ss'}$. Then, either $A1(t,\delta,i)=\text{YES}$, or $A1(t,\delta,i)=(\text{NO},(k(t),\mu(t)))$. In the latter case the tuple $(k(t),\mu(t))$ has the following properties:*

1. $k(t)\geq\min\{k(s),k(s')\}$,
2. *if $k(t)=k(s)=k(s')$, then $\mu(t)\leq\max\{\mu(s),\mu(s')\}$.*

Moreover, if $k(t)=k(s)=k(s')$ for all points $t\in\overline{ss'}$, the function $f\to[0,1]$, $x\mapsto\mu((1-x)s+xs')$ is strictly convex.

As a consequence of the convexity of function f, in order to test if there is a (δ,i)-admissible point on the line segment $\overline{ss'}$ (which means, that there is a point on $t\in\overline{ss'}$ so that the propagation of t with radius-sample δ is successful) a binary search can be carried out among the samples along the line segment $\overline{ss'}$.
The runtime depends on the number of translation-samples that have to be evaluated, which is $O\left(\log{\epsilon_{\text{edge}}}^{-1}\right)$ for sample-distance ϵ_{edge}. Since every propagation takes $O(n^2\log n)$ time, the procedure runs in $O\left(\log{\epsilon_{\text{edge}}}^{-1}n^2\log n\right)$ time.
Since all edges of $P_\delta(c_i)$ have to be considered, evaluating the edges of one polygon takes $O\left(p\log{\epsilon_{\text{edge}}}^{-1}n^2\log n\right)$ time.
We already know from Lemma 3.4, that the length of an edge of $P_\delta(c_i)$ is at most $2\delta\pi p^{-1}$. With $\epsilon_{\text{edge}}\leq\frac{1}{3}\delta^{(3)}\epsilon$ and $p=\lceil 3^{1/4}\pi\epsilon^{-1/2}\rceil$, the number of translation-samples, that have to be propagated, is

$$\log\left(\frac{2\delta\pi}{\epsilon_{\text{edge}}p}\right)\leq\log\left(\frac{6\delta\pi\sqrt{\epsilon}}{\epsilon\delta^{(3)}\pi\sqrt[4]{3}}\right)\leq\log\left(\frac{1}{\sqrt{\epsilon}}\left(\frac{6}{\sqrt[4]{3}}\right)\right)\in O\left(\log\frac{1}{\sqrt{\epsilon}}\right),\tag{1}$$

which leads to a runtime of $O\left(\epsilon^{-1/2}\log\epsilon^{-1}n^2\log n\right)$ in total for the evaluation of one polygon and a fixed radius-sample.
For technical reasons, we also need to state the following insight:

▶ **Lemma 3.6.** *Let $\bar{\delta}:=\delta^*+\epsilon_{obj}$ and let s and s' be the endpoints of the circular arc of $(\bar{\delta},i)$-admissible points on $\partial D_{\bar{\delta}}(c_i)$, then $\|s-s'\|\geq\epsilon_{obj}$.*
Let the sample-distance of the points on the edges of $P_{\bar{\delta}}(c_i)$ be $\epsilon_{edge}:=\frac{\sqrt{3}}{12}\epsilon\delta^{(3)}$ and let $\epsilon_{obj}:=\frac{\sqrt{3}}{12}\epsilon\delta^{(3)}$. Then there is a translation-sample on $P_{\bar{\delta}}(c_i)$ that is a $(1+\epsilon)$-approximation to t_i^*.

Description and Analysis of the Algorithm. We first describe the algorithm: At the start, the value of a 3-approximation to δ^* is computed in $O(n)$ time. Except for basic arithmetic operations, the algorithm consists of four nested loops: The first loop iterates over all of the n input points of the sequence C. For every such point a binary search for $\delta\in[\frac{1}{3}\delta^{(3)},\delta^{(3)}]$ up to accuracy $\epsilon_{\text{obj}}=\frac{\sqrt{3}}{12}\delta^{(3)}\epsilon$ is carried out; this takes $O(\log\epsilon^{-1})$ steps. In each step of this binary search all $p=\lceil 3^{1/4}\pi\epsilon^{-1/2}\rceil\in O\left(\epsilon^{-1/2}\right)$ edges of $P_\delta(c_i)$ are inspected, and on each of them a binary search among $\frac{2\delta\pi}{\epsilon_{\text{edge}}} \in O\left(\frac{1}{\epsilon}\right)$ translation-samples is performed. Each translation-sample is propagated with algorithm A1 for paths, which takes $O(n^2 \log n)$ time per call. This gives a total runtime of $O\left(\epsilon^{-1/2}\left(\log \epsilon^{-1}\right)^2 n^3 \log n\right)$.

If $\delta^{(\epsilon)} < \delta^{(3)}$ is returned, it is valid since there was a translation-sample that has been propagated successfully and therefore is part of a $\delta^{(\epsilon)}$-admissible sequence T. This also means that the very translation-sample that establishes $\delta^{(\epsilon)}$ is propagated and together with the intermediate steps of the propagation gives a T, which then serves as a witness. If there was no successful propagation, $\delta^{(\epsilon)} = \delta^{(3)}$ is returned and we know from Lemma 2.1 that there is always a $\delta^{(3)}$-admissible sequence.

Now we analyse the precision of the algorithm: The precision of the binary search on δ is $\epsilon_{\text{obj}} < \frac{1}{3}\delta^{(3)}\epsilon$; also, all polygons are concentric by construction. If $P_\delta(c_i)$ and $P_{\delta+\epsilon_{\text{obj}}}(c_i)$ are two polygons with circumradii that differ by ϵ_{obj}, the distance between any point on $P_\delta(c_i)$ and the polygon $P_{\delta+\epsilon_{\text{obj}}}(c_i)$ is at most ϵ_{obj} and vice versa. Every edge of these two polygons is sampled with points of distance ϵ_{edge}, and with Thales' theorem it follows that for every translation-sample on $P_\delta(c_i)$ there is a translation-sample on $P_{\delta+\epsilon_{\text{obj}}}(c_i)$ with distance $\frac{1}{3}\delta^{(3)}\epsilon$ or less and vice versa. Combined with Lemma 3.4, we have that for every δ there is a translation-sample in $D_\epsilon(t_i)$ for every (δ, i)-admissible point $t_i \in \partial D_\delta(c_i)$. According to Lemma 3.6, one of the following two cases holds: Either there is at least one (δ, i)-admissible translation-sample on $P_\delta(c_i)$ for every $\delta \geq \delta^* + \frac{1}{3}\delta^{(3)}\epsilon$ so that the line segment of all (δ, i)-admissible points on one of the edges of this polygon is at least $\frac{1}{3}\delta^{(3)}\epsilon$ long, or one vertex of the polygon is a (δ, i)-admissible point and since all polygons are concentric, this vertex is (δ, i)-admissible for every $\partial D_\delta(c_i)$ with $\delta \geq \delta^*$. We consider the radius-samples $\bar{\delta}$, $\bar{\delta} + \epsilon_{\text{obj}}$ and $\bar{\delta} + 2\epsilon_{\text{obj}}$, where $\bar{\delta} := \delta^* + \zeta - \epsilon_{\text{obj}}$ for some $0 < \zeta < \epsilon_{\text{obj}}$. Since $\bar{\delta} < \delta^*$, none of the propagations for this δ are successful. A short analysis leads to $\bar{\delta} + \epsilon_{\text{obj}} < \delta^* + \epsilon_{\text{obj}} < \bar{\delta} + 2\epsilon_{\text{obj}} < \delta^* + \frac{1}{3}\delta^{(3)}\epsilon$. Due to Lemma 3.6, this means, that for radius-sample $\bar{\delta} + 2\epsilon_{\text{obj}}$ the two endpoints of the circular arc of $(\bar{\delta} + 2\epsilon_{\text{obj}}, i)$-admissible points in $D_{\bar{\delta}+2\epsilon_{\text{obj}}}(c_i)$ have a distance of at least ϵ_{obj}, which is why there is at least one translation-sample on the inscribing polygon of this disk, that is propagated successfully and the algorithm returns $\delta^{(\epsilon)} = \bar{\delta} + 2\epsilon_{\text{obj}} < \delta^* + \frac{1}{3}\delta^{(3)}\epsilon < (1+\epsilon)\delta^*$ as the approximation to δ^*. Hence the algorithm computes a $(1+\epsilon)$-approximation to δ^* for Problem 1.1.

It also returns a $(\delta^{(\epsilon)}, i)$-admissible point $t^{(\epsilon)}$ from which an $\delta^{(\epsilon)}$-admissible sequence $T^{(\epsilon)}$ can be computed in $O(n^2 \log n)$ time.

References

1. Helmut Alt and Leonidas Guibas. Discrete Geometric Shapes: Matching, Interpolation, and Approximation. In *Handbook of Computational Geometry*, pages 121–153. Elsevier B.V., 2000.
2. Christian Knauer, Luise Sommer, and Fabian Stehn. Elastic Geometric Shape Matching for Translations under the Manhattan Norm. In *Computational Geometry: Theory and Applications*, 2018.
3. Christian Knauer and Fabian Stehn. Elastic Geometric Shape Matching for Point Sets under Translations. In *Proceedings of Algorithms and Data Structures - 14th International Symposium (WADS)*, pages 578–592, 2015.
4. Remco C. Veltkamp and Michiel Hagedoorn. State of the art in shape matching. In *Principles of Visual Information Retrieval. Advances in Pattern Recognition*, pages 87–119. Springer, London, 2001.