

APPROXIMATING GROMOV-HAUSDORFF DISTANCE IN PLANAR GRAPHS

by

Seong Ioi Wang



APPROVED BY SUPERVISORY COMMITTEE:

---

Kyle Fox, Chair

---

Sergey Bereg

---

Benjamin Raichel

Copyright © 2020

Seong Ioi Wang

All rights reserved

APPROXIMATING GROMOV-HAUSDORFF DISTANCE IN PLANAR GRAPHS

by

SEONG IOI WANG, BS

THESIS

Presented to the Faculty of  
The University of Texas at Dallas  
in Partial Fulfillment  
of the Requirements  
for the Degree of

MASTERS OF SCIENCE IN  
COMPUTER SCIENCE

THE UNIVERSITY OF TEXAS AT DALLAS

May 2020

## ACKNOWLEDGMENTS

This paper would not exist without the help of Kyle Fox. I am sincerely grateful for his guidance throughout this journey and always being patient with me. I also want to thank Benjamin Raichel and Sergey Bereg for willing to be committee members on this thesis.

April 2020

# APPROXIMATING GROMOV-HAUSDORFF DISTANCE IN PLANAR GRAPHS

Seong Ioi Wang, MS  
The University of Texas at Dallas, 2020

Supervising Professor: Kyle Fox, Chair

Many concepts in computer science can be represented as metric graphs, and Gromov-Hausdorff distance offers a natural way to measure the additive distortion between any metric spaces. However, there are no known polynomial time algorithms to compute or approximate this distance for general graphs. Even in metric trees, it is NP-Hard to compute or even approximate it within a constant factor of 3. This thesis provides a constant factor approximation for Gromov-Hausdorff distance between planar graphs whose edges are a constant factor larger than their Gromov-Hausdorff distance. A constant factor approximation for functional distortion distance between Contour trees with long edges is also provided.

## TABLE OF CONTENTS

ACKNOWLEDGMENTS . . . . .	iv
ABSTRACT . . . . .	v
LIST OF FIGURES . . . . .	vii
CHAPTER 1 INTRODUCTION . . . . .	1
1.1 Preliminaries . . . . .	1
1.1.1 Gromov-Hausdorff Distance . . . . .	2
1.1.2 SPQR Trees . . . . .	4
1.1.3 Contour Tree . . . . .	7
1.2 Related Works . . . . .	9
CHAPTER 2 APPROXIMATING GROMOV-HAUSDORFF DISTANCE . . . . .	10
2.1 Low $d_{GH}$ for Graphs with Long Edges . . . . .	10
2.2 Approximating $d_{GH}$ for Planar Graphs with Long Edges . . . . .	12
CHAPTER 3 APPROXIMATING FUNCTIONAL DISTORTION DISTANCE . . . . .	16
3.1 Low $d_{FD}$ for Trees with Long Edges . . . . .	16
3.2 Approximating $d_{FD}$ for Trees with Long Edges . . . . .	17
CHAPTER 4 CONCLUSION . . . . .	19
REFERENCES . . . . .	20
BIOGRAPHICAL SKETCH . . . . .	22
CURRICULUM VITAE	

## LIST OF FIGURES

1.1	Example graph for decomposition into SPQR tree . . . . .	6
1.2	Biconnected component nodes . . . . .	6
1.3	Biconnected component tree with vertex connectivity . . . . .	6
1.4	ST-numbering . . . . .	7
1.5	SPQR tree . . . . .	7
1.6	Example mesh with it's depicted contour tree. These figures are taken from [5] .	8

# CHAPTER 1

## INTRODUCTION

There have been multiple tools introduced to solve the problem of shape comparison and each one offers a different methodology to measure “similarity.” Gromov-Hausdorff distance, denoted as  $d_{GH}$ , is one of these tools and it measures how “isometric” two shapes with associated metrics are. When Gromov-Hausdorff distance was introduced, it was mainly used by pure mathematicians who were interested in the topology between structures [13]. Since then it has been shown to be a useful measurement between metric spaces in computational geometry [11]. The measure is invariant to bends and rigid isometries which allows it to detect metric similarities. Facundo Memoli [12] clarified several observations regarding the lower bounds of Gromov-Hausdorff distance and showed that calculating Gromov-Hausdorff distance is NP-Hard through a reduction from the Bottleneck Quadratic Assignment Problem.

We extend the previous results for approximating Gromov-Hausdorff [1] distance by introducing a constant factor approximation algorithm for planar graphs whose edges are a constant factor larger than their Gromov-Hausdorff distance. The algorithm we present will first decompose the planar graphs into tree like structure, so we can then approximate the Gromov-Hausdorff distance from the leaves up to the root. Furthermore, we can extend the observations made in Gromov-Hausdorff distance to a related metric for contour trees called functional distortion distance due to their similarity. We then describe a constant factor approximation for functional distortion distance between contour trees.

### 1.1 Preliminaries

A *metric space* is defined as a pair  $(X, d_X)$  where  $X$  is a set,  $d_X : X \times X \rightarrow \mathbb{R}$  is a distance function, and the following conditions are satisfied for all  $x, y \in X$ :

1.  $d_X(x, y) = 0 \Leftrightarrow x = y$  (Identity)



2.  $d_X(x, y) = d_X(y, x)$  (Symmetry)
3.  $d_X(x, z) \leq d_X(x, y) + d_X(y, z)$  (Triangle Inequality)

The two dimensional Euclidean space is an example of a metric space where  $X = \mathbb{R}^2$  and the distance function is  $d((x_1, y_1), (x_2, y_2)) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}$ . Metric spaces also includes finite spaces such as graphs  $G = (V, E)$  paired with their shortest path distance. In this paper, all graphs are assumed to be metric graphs; the metric is over all points in a geometric realization of the graph, and one thing to note is that a vertex of degree two in a metric graph is indistinguishable from the interior of an edge as far as the metric is concerned.

A graph  $X$  is *planar* if it is isomorphic to a graph  $H$  such that the vertices and edges of  $H$  are contained in the plane and such that at most one vertex or edge passes through any point of the plane. An *articulation point* is a vertex on a graph that disconnects it when removed. A *biconnected* graph is a graph with no articulation points. A *separation pair* is a pair of vertices that disconnects the graph when removed. A *triconnected graph* is a graph with no separation pair. An *embedding* of a planar graph is an ordering of edges around every vertex in a graph which allows the graph to be drawn on a plane without any edges crossing [9].

### 1.1.1 Gromov-Hausdorff Distance

Gromov-Hausdorff distance is often defined as an extension of Hausdorff distance. Let  $X$  and  $Y$  be subspaces of a metric space  $Z$ . Hausdorff distance is formally defined by

$$d_H(X, Y) = \max\{\sup_{y \in Y} \inf_{x \in X} d_Z(x, y), \sup_{x \in X} \inf_{y \in Y} d_Z(x, y)\}.$$

A more intuitive definition is that Hausdorff distance between two subspaces is the furthest point from one subset to the nearest point in the other subset.

An important difference between Gromov-Hausdorff distance and Hausdorff distance is that Hausdorff distance only measures the distance between subsets in the same metric space and not the distance between abstract metric spaces. *Gromov-Hausdorff* distance extends this definition by creating an isometric embedding for both  $X$  and  $Y$  to some arbitrary metric space  $Z$  (denoted as  $f_{X \rightarrow Z}$  and  $g_{Y \rightarrow Z}$ ) and taking the Hausdorff distance between them:

$$d_{GH}(X, Y) = \inf_{f, g} d_H(f_{X \rightarrow Z}(X), g_{Y \rightarrow Z}(Y))$$

Observe that  $d_{GH}(X, Y) \leq d_H(X, Y)$  if  $X$  and  $Y$  are subsets of a common metric space. The Gromov-Hausdorff distance between isometric spaces is 0.

A direct application of Gromov-Hausdorff distance requires the construction of a new metric space  $Z$ . This requires a complex construction even for simple cases. A more convenient approach is to directly compare the two spaces. To accomplish that, we first describe the notion of correspondence  $\mathfrak{R} \subset X \times Y$ . A *correspondence* satisfies the conditions that for every  $x \in X$  there exist a  $y \in Y$ , and for every  $y \in Y$  there exists a  $x \in X$  such that  $(x, y) \in \mathfrak{R}$ .

The *distortion* of  $\mathfrak{R}$  is defined as:

$$dis \mathfrak{R} = \sup\{|d_X(x, x') - d_Y(y, y')| : (x, y), (x', y') \in \mathfrak{R}\}.$$

Now, we can define Gromov-Hausdorff distance as

$$d_{GH}(X, Y) = \frac{1}{2} \inf_{\mathfrak{R}} dis(\mathfrak{R}).$$

In other words, the Gromov-Hausdorff between  $X$  and  $Y$  is also half the minimum distortion between any correspondence.

### 1.1.2 SPQR Trees

The *SPQR tree*, introduced by Di Battista and Tamassia [2], is a graph structure that maintains a planar embedding of a biconnected planar graph while isolating its triconnected components. It is a useful tool for graph drawing, planarity testing, representing all embeddings, etc. Further results also show that it is possible to create a unique SPQR tree if the root is known a priori [9]. This data structure was initially introduced to solve the incremental planarity testing problem [6].

To construct an SPQR tree of a biconnected graph  $G$ , an *st-graph* is needed. The *st-graph* of  $G$  will orientate the edges such that it creates a directed acyclic graph with a unique vertex  $s$  with no incoming edges, and a unique vertex  $t$  with no outgoing edges. The *st-graph* algorithm described by Even and Tarjan will produce a planar *st-graph* in linear time given an undirected biconnected planar graph [7]. First, identify one source vertex  $s$  and one sink vertex  $t$  in  $G$ , which admits a planar embedding such that  $s$  and  $t$  are on the boundary of the same face. Then, run a modified depth-first search algorithm to check that the graph is biconnected and if so, construct an topological ordering of the of all the vertices. Further details can be found in Tarjan's *st-numbering* algorithm [15]. We will use the following terms to discuss SPQR trees [6]: A *split pair* is either a separation-pair or a pair of adjacent vertices. A *split component* of a split pair is either an edge or a maximal subgraph  $C$  of  $G$  such that  $\{u, v\}$  is not a split pair of  $C$  (removing  $\{u, v\}$  from  $C$  does not disconnect  $C \setminus \{u, v\}$ ). A *maximal split pair* is one that will have all other split pairs in its split component.

The decomposition of a biconnected planar graph into an SPQR tree is done recursively with respect to its split pairs. The tree contains four types of nodes: Serial or S-nodes, Parallel or P-nodes, Trivial or Q-node, and Rigid or R-nodes. Each node will contain an associated planar *st-graph* called a *skeleton*. Every node that is not the root is also associated

with an edge in skeleton of the parent node called the *virtual edge*. The tree is recursively defined as follows.

Trivial Case: If  $G$  is a single edge from  $s$  to  $t$ , the SPQR tree consist of a Q-node whose skeleton is  $G$  itself.

Parallel Case: If the split pair  $\{s, t\}$  has at least three split components  $G_1, \dots, G_k$  for  $k \geq 2$ , the root is a P-node, whose skeleton consist of  $k$  parallel edges from  $s$  to  $t$ .

Serial Case: If  $G \setminus \{s, t\}$  has an articulation point, let  $c_1, \dots, c_{k-1}$  for  $k \geq 2$  be the articulation points of  $G$ , which will partition  $G$  into its blocks  $G_1, \dots, G_k$ , in this order from  $s$  to  $t$ . The root is an S-node. Moreover, since the graph is biconnected, the skeleton of this node is the chain of  $k$  edges in a cycle.

Rigid Case: If none of the cases apply, let  $\{s_1, t_1\}, \dots, \{s_k, t_k\}$  for  $k \geq 1$  be the maximal split pairs of  $G$  which will be either virtual or real edges. For  $i = 1, \dots, k$ , let  $G_i$  be the union of all the split components of  $\{s_i, t_i\}$ . The root is an R-node. The skeleton of this node is obtained from  $G$  by replacing each subgraph  $G_i$  with virtual edge  $e_i$  from  $s_i$  to  $t_i$ .

In a later work, Di Battista and Tamassia showed some properties regarding the SPQR nodes [9].

- The skeleton of an R-node is a triconnected graph.
- The skeleton of an S-node is a cycle.
- The skeleton of a P node is a triconnected multigraph consisting of multiple edges.

Note a planar graph can be decomposed into multiple SPQR trees by first decomposing it into a biconnected component tree. Consider the example graph shown in Figure 1.1. First the graph  $G$  is decomposed into its biconnected component, shown in Figure 1.2 where each

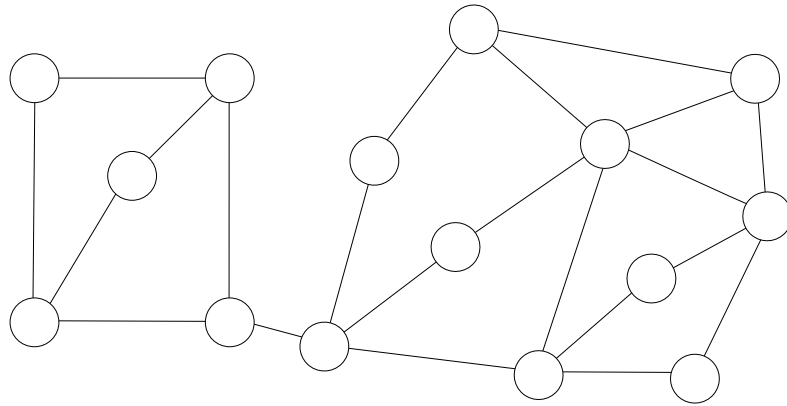


Figure 1.1. Example graph for decomposition into SPQR tree

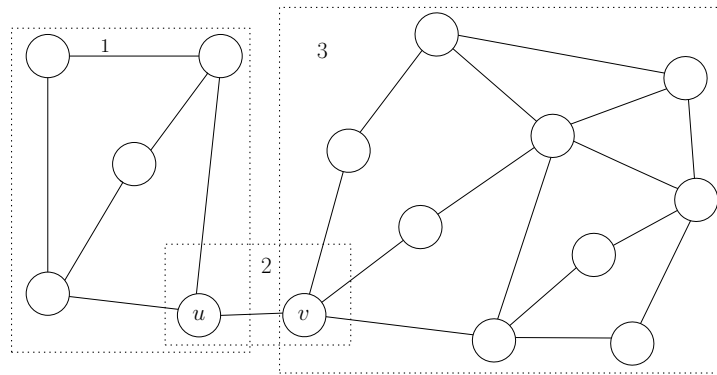


Figure 1.2. Biconnected component nodes

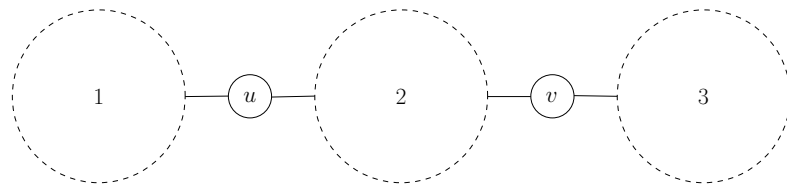


Figure 1.3. Biconnected component tree with vertex connectivity

rectangle denotes a component. The tree consisting of biconnected components and with the articulation points is shown in 1.3. Then the SPQR tree of each biconnected component node is created by first finding an st-numbering as shown in Figure 1.4 and decomposing it into a tree as shown in Figure 1.5 where dashed boxes denote nodes and dashed lines denote virtual edges.

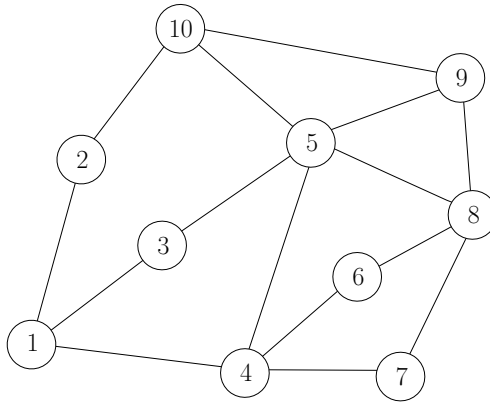


Figure 1.4. ST-numbering

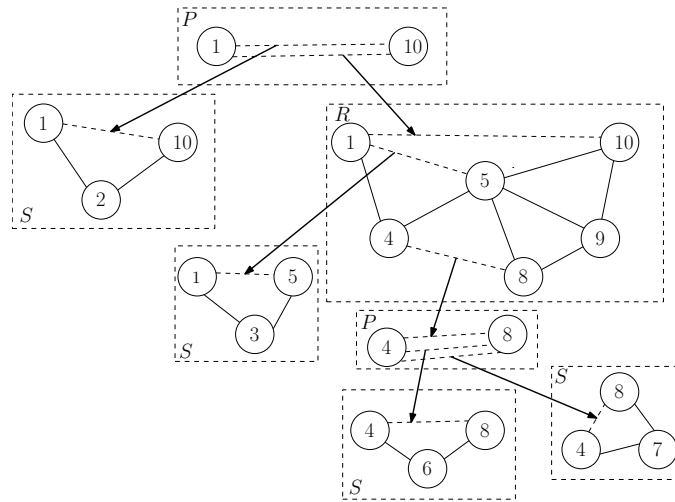


Figure 1.5. SPQR tree

### 1.1.3 Contour Tree

A *contour tree* is a tree structure that stores how the topology of level sets of a space are changing with respect to some continuously increasing height value. In this tree, the leaf vertices represent the creation or deletion of any components in their level set. In Figure 1.6, vertices 9, 10, 8, and 7 denote the creation of contours and vertices 1 and 2 shows the contours disappearing. An interior vertex represents components merging into a single contour or splitting into multiple contours. A popular use for contour trees is to capture the isolines on terrain maps in GIS. The contour tree will capture the elevation points where the contour

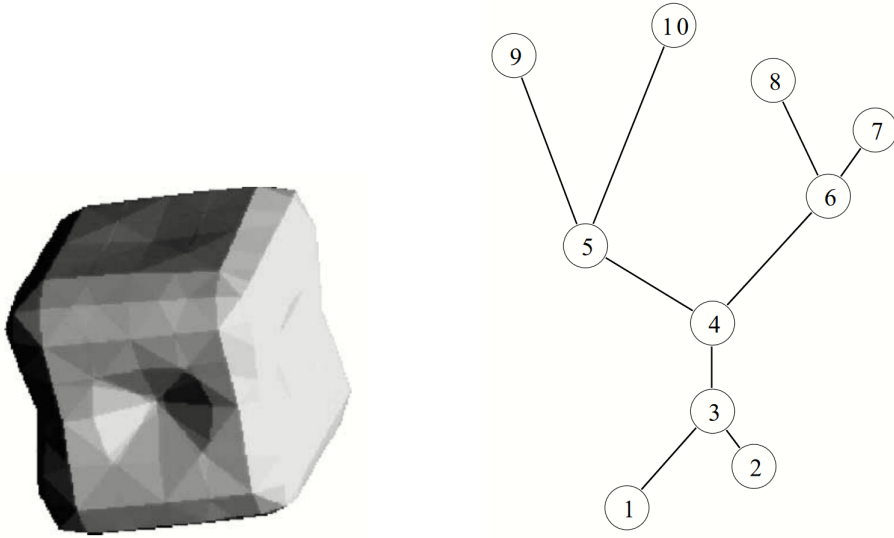


Figure 1.6. Example mesh with its depicted contour tree. These figures are taken from [5] lines split, join, appear, and disappear [5]. Bauer, et. al. [3] introduced *functional distortion distance* as a modification of the Gromov-Hausdorff distance to measure differences in Reeb Graphs, a generalization of the contour tree.

Let  $C$  be a contour tree. To only capture the differences in height values in  $C$ , let  $\pi$  be the continuous path between two points  $u$  and  $v$ . The range of this path is defined as  $range(\pi) := [\min_{x \in \pi} f(x), \max_{x \in \pi} f(x)]$ , and its height is  $height(\pi) = \max_{x \in \pi} f(x) - \min_{x \in \pi} f(x)$ . The distance between  $u$  and  $v$  is defined as

$$d_f(u, v) = height(\pi)$$

where  $\pi$  is the path from  $u$  to  $v$  in  $C$ . To defined the distance between contour trees  $C_1$  and  $C_2$  with height functions  $f : C_1 \rightarrow \mathbb{R}$  and  $g : C_2 \rightarrow \mathbb{R}$ , let  $\Phi : C_1 \rightarrow C_2$  and  $\Psi : C_2 \rightarrow C_1$  be continuous maps. Now define

$$G(\Phi, \Psi) = \{(x, \Phi(x)) : x \in C_1\} \cup \{(\Psi(y), y) : y \in C_2\}$$

$$D(\Phi, \Psi) = \sup_{(x, y), (x', y') \in G(\Phi, \Psi)} |d_{C_1}(x, x') - d_{C_2}(y, y')|$$

where  $G(\Phi, \Psi)$  is the set of correspondences between  $C_1$  and  $C_2$  induced by  $\Phi$  and  $\Psi$ . The functional distortion distance is defined as:

$$d_{FD}(C_1, C_2) = \inf_{\Phi, \Psi} \max\left\{\frac{1}{2}D(\Phi, \Psi), \|f - g \circ \Phi\|_\infty, \|f \circ \Psi - g\|_\infty\right\}$$

where  $\Phi$  and  $\Psi$  range over all continuous maps between  $C_1$  and  $C_2$ .

## 1.2 Related Works

A paper by Agarwal, et. al. [1] proved that approximating the Gromov-Hausdorff distance for metric trees to a factor better than 3 is NP-Hard. They also provided an  $O(n^{9/2} \log n)$  time algorithm that can compute an  $O(\min\{n, \sqrt{rn}\})$  factor approximation where  $r$  is the ratio of the longest and shortest edge length in both trees. This was accomplished by first showing that a related tree metric called interleaving distance  $d_I$ , is within a constant factor of the Gromov-Hausdorff distance  $\frac{1}{2}d_{GH}(T_1, T_2) \leq d_I(T_1, T_2) \leq 14d_{GH}(T_1, T_2)$ . Therefore, any approximation algorithm for interleaving distance also applies for Gromov-Hausdorff distance within constant bounds [10, 1]. Further results from Touli and Wang [16] provided a fixed-parameter tractable algorithm to compute the exact interleaving distance and thus, a 14-approximation for Gromov-Hausdorff distance.

Bauer, et. al. [4] showed that there is a strong equivalence between functional distortion distance and interleaving distance,  $d_I$ , where  $d_I(X, Y) \leq d_{FD}(X, Y) \leq 7d_I(X, Y)$ . As stated, interleaving distance is within constant factor of Gromov-Hausdorff distance. Thus, functional distortion distance and Gromov-Hausdorff distance are closely related as there is a strong equivalence between the two.



## CHAPTER 2

### APPROXIMATING GROMOV-HAUSDORFF DISTANCE BETWEEN PLANAR GRAPHS

Let planar metric graphs  $X = ([V_X, E_X], d_X)$  and  $Y = ([V_Y, E_Y], d_Y)$  be the input with  $n$  and  $m$  vertices respectively. We assume that all vertices have degree 1 or at least 3.

Suppose  $X$  and  $Y$  are isomorphic graphs with a known matching between the vertices and edges. We will say  $X$  and  $Y$  are  $\epsilon$ -compatible if the lengths of the matched edges differ no more than  $\epsilon$  and the all pair-wise differences between their all pair shortest path (APSP) matrices induced by the matching is no greater than  $\epsilon$ .

#### 2.1 Low $d_{GH}$ for Graphs with Long Edges

**Lemma 1.** Let  $\epsilon > 0$  and  $X$  and  $Y$  be graphs such that all edges  $e$  have length strictly greater than  $10\epsilon$  (i.e.  $X$  and  $Y$  only have long edges). If  $d_{GH}(X, Y) \leq \epsilon$ ,  $X$  and  $Y$  are isomorphic and  $8\epsilon$ -compatible.

*Proof.* First, let us show that every vertex in  $V_X$  must correspond to a point at most  $5\epsilon$  distance away from some vertex in  $V_Y$  and vice versa. Suppose  $a \in V_X$  corresponds to  $a' \in Y$  where  $a'$  is over  $5\epsilon$  away from any vertices in  $V_Y$ . Suppose  $a$  has degree at least 3, let  $c_i, c_j$  and  $c_k$  be distinct points  $3\epsilon$  distance away from  $a$  correspond to points  $c'_i, c'_j$  and  $c'_k$  respectively. Given  $d_{GH}(X, Y) \leq \epsilon$ , the distance between points  $c'_i, c'_j$  or  $c'_k$  must be within  $[4\epsilon, 8\epsilon]$  and the distance from  $a'$  to  $c'_i, c'_j$  or  $c'_k$  must be within  $[\epsilon, 5\epsilon]$ . Point  $a'$  is interior of an edge. Suppose  $c'_i$  and  $c'_j$  are on the same side of  $a'$  on the edge with  $c'_i$  closer to  $a'$  and  $c'_k$  is on the other side. These correspondences cause  $d_Y(a', c'_i) \geq \epsilon$ ,  $d_Y(a', c'_j) \leq 5\epsilon$ , and  $d_Y(a', c'_k) \geq \epsilon$ . However, this also leads to a contradiction as this correspondence will cause  $d_Y(c'_k, c'_i) < 4\epsilon$  or  $d_Y(c'_k, c'_j) > 8\epsilon$ . A similar contradiction exists if  $c'_i, c'_j$  and  $c'_k$  are on the same side of the edge  $a'$  is on. In the case where  $a$  has degree 1,  $a'$  should also be  $5\epsilon$  distance

away from a vertex with degree 1. In the case where  $a'$  is less than  $5\epsilon$  away from a vertex with degree greater than 1, let  $c'_1$  with  $d_Y(c'_1, a') = 4\epsilon$  be a point that corresponds to  $c_1 \in X$  and let  $c'_2$  with  $d_Y(c'_2, a') = 4\epsilon$  be a point the other side of  $a'$ . In  $Y$ ,  $d_Y(c'_1, c'_2) = 8\epsilon$ . This implies that  $c_1$  and  $c_2$  follows  $d_X(a, c_1), d_X(a, c_2) \in [2\epsilon, 6\epsilon]$  while maintaining  $d_X(c_1, c_2) \geq 6\epsilon$ , a contradiction. If  $a'$  does not map within  $5\epsilon$  of any vertices, a similar contradiction applies.

Now, consider vertices  $a, b \in V_X$  corresponding within a distance of  $5\epsilon$  of  $u, v \in V_Y$ . If the edge  $(a, b)$  exists, the edge  $(u, v)$  also exists. Let  $k$  be the midpoint between  $a$  and  $b$  and let  $a', b'$  and  $k'$  be the corresponding points in  $Y$ . The distortions  $d_Y(k', a') \leq d_X(a, k) + 2\epsilon$  and  $d_Y(k', b') \leq d_X(b, k) + 2\epsilon$  must be maintained while  $k$  must be  $3\epsilon$  away from every point in  $Y$  which is not possible unless the edge  $(u, v)$  exists. Furthermore, if there is an edge incident to  $a$  and  $b$ , the edge incident to  $u$  and  $v$  should have no more than  $8\epsilon$  difference between their edge lengths. Let us subdivide edge  $(a, b)$  in half into edges  $(a, k)$  and  $(k, b)$  with segments  $d_X(a, k) = d_X(k, b)$  and similarly subdivide  $(u, v)$  into  $(u, k')$  and  $(k', v)$  where  $k \in X$  corresponds with  $k' \in Y$ . Consider points  $c_a$  and  $c_b$  on edge  $(a, b)$  that are  $3\epsilon$  away from  $a$  and  $b$  that correspond to points  $c'_a$  and  $c'_b$  on  $(u, v)$ . The distortion on the edge from  $(a, b)$  with  $(u, v)$  can not be greater than  $8\epsilon$  because  $|d_Y(c'_b, k') - d_X(c_b, k)| \leq 2\epsilon$ ,  $|d_Y(c'_a, k') - d_X(c_a, k)| \leq 2\epsilon$ ,  $c_a$  has a max of  $2\epsilon$  distortion with  $a$ , and  $c_b$  has a max of  $2\epsilon$  distortion with  $b$ . Additionally, matched vertices  $a, b \in V_X$  corresponding within a distance of  $5\epsilon$  of  $u, v \in V_Y$ , have at most  $2\epsilon$  distortion (i.e.  $|d_X(a, b) - d_Y(u, v)| \leq 7\epsilon$ ). We know, by definition, that  $a$  and  $b$  corresponding points  $a', b' \in Y$  have  $|d_X(a, b) - d_Y(a', b')| \leq 2\epsilon$ . We can asserted that the maximum distortion from  $a'$  and  $b'$  to their closest vertices  $u$  and  $v$  is  $|d_Y(a', b') - d_Y(u, v)| \leq 5\epsilon$  since  $a'$  and  $b'$  must be within  $5\epsilon$  of  $u$  and  $v$ . Adding these distortion will yield a maximum distortion of  $7\epsilon$  between  $a, b \in X$  and  $u, v \in Y$ .

All together, all vertices should map near vertices with the same degree, edges should only map onto edges of similar length, and the all pair shortest path between the matched graphs should have distortion less than  $2\epsilon$ . Therefore,  $X$  and  $Y$  are  $8\epsilon$ -compatible.  $\blacksquare$

**Lemma 2.** Let  $\epsilon > 0$  and  $X$  and  $Y$  be graphs such that all edges have lengths strictly greater than  $10\epsilon$  (i.e.  $X$  and  $Y$  only have long edges). If  $X$  and  $Y$  are isomorphic and  $2\epsilon$ -compatible, then  $d_{GH}(X, Y) \leq 3\epsilon$ .

*Proof.* Let us construct a mapping between  $X$  and  $Y$  such that  $d_{GH}(X, Y) \leq 3\epsilon$ . We define our correspondence by matching every vertex and edge between  $X$  and  $Y$  according to their valid  $2\epsilon$ -compatibility. If two corresponding edges have different length, we will map each point linearly according to their relative distance between the incident vertices. We can show that the maximum distortion in this matching is  $6\epsilon$ .

Consider any pair of corresponding points  $a, b \in X$  and  $a', b' \in Y$ . We know that  $a$  and  $a'$  are on corresponding edges by construction and the same with  $b$  and  $b'$ . Therefore, we know that the maximum increase in length at the endpoints of these paths is  $2\epsilon + 2\epsilon$ . Furthermore, we know that the increase between vertices is at most  $2\epsilon$ . Together, the increase in length of corresponding paths crossing an edge, traveling vertex to vertex, and across another edge has at most  $6\epsilon$  difference. (i.e.  $d_Y(a', b') \leq d_X(a, b) + 6\epsilon$  and a similar argument shows  $d_Y(a', b') \geq d_X(a, b) - 6\epsilon$ ). Thus,  $d_{GH}(X, Y) \leq 3\epsilon$ . ■

An interesting consequence of  $X$  and  $Y$  being  $2\epsilon$ -compatible is that we can construct a continuous bijective map between the graphs.

## 2.2 Approximating $d_{GH}$ for Planar Graphs with Long Edges

Let  $X$  and  $Y$  be planar graphs with long edges. We first decompose the graphs into biconnected component trees where vertex ( $X_i$  and  $Y_i$ ) is a biconnected component or an articulation point. We will pick one articulation point in  $X$  and try to pair it with every other articulation point in  $Y$  as roots in the biconnected component trees to approximate  $d_{GH}(X, Y)$ . We will do something similar to find the  $d_{GH}(X_i, Y_i)$  of each biconnected component by picking an edge  $(a, b) \in X_i$  and try pairing it to an arbitrary edge  $(c, d) \in Y_i$ .

We decompose  $X_i$  and  $Y_i$  into SPQR trees  $T^X$  and  $T^Y$  by using the split pair  $r = (a, b)$  and  $r' = (c, d)$  as root nodes, respectively. Starting from the leaf nodes, we will calculate the minimum distortion achievable between the skeletons of all matched nodes in their subtree. At nodes  $\alpha \in T^X$  and  $\beta \in T^Y$  in the SPQR trees, let  $T_\alpha^X$  and  $T_\beta^Y$  denote the sub SPQR trees rooted at node  $\alpha$  and  $\beta$ , and let  $G(T_\alpha^X)$  and  $G(T_\beta^Y)$  denote the subgraph of the biconnected components represented by  $T_\alpha^X$  and  $T_\beta^Y$ .

We now define the *skeleton compatibility* of nodes  $\alpha$  and  $\beta$  to be the minimum skeleton  $\delta$ -compatibility where length of virtual edges  $\{u, v\}$  is defined by the shortest distance between  $u$  and  $v$  in the skeleton where  $\{u, v\}$  is the split pair. If  $\alpha$  and  $\beta$  are different type of SPQR nodes, we declare the compared skeleton to be  $\infty$ -compatible. If they are the same type of nodes with isomorphic skeletons, we compute the skeleton compatibility based on the type of node. We define the *children compatibility* to be the maximum value needed to match children nodes of  $\alpha$  and  $\beta$  such that they are skeleton compatible. In the recursive step for node  $\alpha$  and  $\beta$ , we will also find the *skeleton distortion*  $\sigma(\alpha, \beta)$  of  $\alpha$  and  $\beta$  which is the maximum skeleton compatibility, children compatibility, or *ancestor compatibility* between the path  $d_X(a, st) - d_Y(a', st')$  where  $a \in G(T_\alpha^X)$  and  $a' \in G(T_\beta^Y)$  are vertices and  $st \in X$  and  $st' \in Y$  are matching ancestor split vertices. We can find  $\sigma(\alpha, \beta)$  base on the type of nodes.

Trivial Nodes: Trivial nodes consist of an edge. Therefore, the  $\delta$ -compatibility is just the difference in edge lengths and there are no child nodes to check.

Serial Nodes: There are 2 possible matchings between  $\alpha$  and  $\beta$ . We select the matching that minimizes the maximum skeleton compatibility or maximum children compatibility.

Parallel Nodes: We build a bipartite graph where the vertices on one virtual side represents the edges in one of the parallel nodes and similarly for the other side. We

set the edge weights in this bipartite graph to the skeleton compatibilities of their subtrees. After the construction of this bipartite graph, a bottleneck matching is calculated in polynomial time as described by Punnen and Nair [14]. The skeleton distortion for  $\alpha$  and  $\beta$  is the maximum edge weight of the bottleneck matching.

**Rigid Nodes:** Triconnected planar graphs can only have 2 embeddings [17]. Thus, we can directly check both embeddings and select the one with the minimum compatibility. This compatibility is the skeleton distortion.

The following lemma is immediate.

**Lemma 3.** Suppose biconnected components  $X_i$  and  $Y_i$  are  $\epsilon$ -compatible. Our procedure for their SPQR trees finds a  $\delta$  and  $\sigma$  for their rooted nodes that guarantees a  $\delta + 2\sigma$ -compatibility between  $X_i$  and  $Y_i$ .

*Proof.* Consider the isomorphism corresponding to our computation of  $\delta$ . Matched edges differ by at most  $\delta$  in length. Let  $a$  and  $b$  in  $X_i$  correspond to  $a'$  and  $b'$  in  $Y_i$ . Let  $\alpha$  and  $\beta$  be the lowest nodes such that  $a, b \in G(T_\alpha^X)$  and  $a', b' \in G(T_\beta^Y)$  and let  $(i, j)$  correspond to  $(i', j')$  and  $(k, l)$  and  $(k', l')$  be the ancestor split pairs of  $a$  and  $b$  in the skeleton in skeleton of  $\alpha$ . The distortion from the path from both  $\alpha$  and  $\beta$  to their split pair will have a maximum of  $\delta$  distortion. The shortest path from  $a$  to  $b$  must intersect  $(i, j)$  and  $(k, l)$  in that order. The total distortion of the path is therefore  $\delta + 2\sigma$ . ■

Let us now approximate the compatibility  $\gamma$  between  $X$  and  $Y$ . Repeatably test an arbitrary pair of articulation points between the biconnected component trees as roots. In each level of the rooted biconnected component tree, we build a bipartite graph where the vertices on each side represent the biconnected components. The edge weight is the  $\delta + 2\sigma$  for the two components. Now we can run the bottleneck matching to find the minimum

$\delta + 2\sigma$  needed to find a matching. For every matching split pair  $st \in X$  and  $s't' \in Y$  in the biconnected components, we will call distortion between  $st$  to the root articulation point in  $X$  and  $s't'$  to the root articulation point in  $Y$  the *up distortion* of  $st$  and  $s't'$ . Let  $\gamma$  be the maximum possible up distortion of all paired split pairs between  $X$  and  $Y$ .

**Lemma 4.** This procedure will produce a  $\delta + 2\sigma + 2\gamma$  guarantee for the compatibility between  $X$  and  $Y$ .

*Proof.* Consider the isomorphism corresponding to our computation of  $\gamma$ . We know the  $\delta + 2\sigma$ -compatibility between biconnected components  $X_i$  and  $Y_i$ . Let points  $a, b \in X$  correspond to  $a', b' \in Y$ . Let articulation points  $p$  and  $p'$  be lowest roots such that  $a$  and  $b$  are in the subtree rooted at  $p$  and vice versa for  $p'$ . The shortest path from  $a$  to  $b$  and  $a'$  to  $b'$  will traverse through articulation points and biconnected components. Let  $\delta + 2\sigma$  be the maximum compatibility of all the biconnected components in the path from  $a$  to  $b$  and  $a'$  to  $b'$ . The total distortion of the path is  $\delta + 2\sigma + 2\gamma$ . ■

Combining Lemma 2, 3 and 4, we can conclude with the following theorem.

**Theorem 5.** Let  $\epsilon > 0$  and  $X$  and  $Y$  are planar graphs such that all edges have length strictly greater than  $10\epsilon$  (i.e.  $X$  and  $Y$  only have long edges). There exists a polynomial time  $\frac{15}{2}$ -approximation algorithm for the Gromov-Hausdorff distance of  $X$  and  $Y$ .

## CHAPTER 3

### APPROXIMATING FUNCTIONAL DISTORTION DISTANCE BETWEEN CONTOUR TREES

Let  $C_1$  and  $C_2$  be contour trees with height functions  $f$  and  $g$ , respectively.

Suppose  $C_1$  and  $C_2$  are isomorphic contour trees with a known matching between the vertices and edges. We will say  $C_1$  and  $C_2$  are  $\epsilon$ -compatible if the length between all the matched edges differ greater than  $\epsilon$ , the all pair-wise differences between their matched all pair shortest path (APSP) matrices is no greater than  $\epsilon$ , and the height value between matched vertices differ by at most  $\epsilon$ .

#### 3.1 Low $d_{FD}$ for Trees with Long Edges

**Lemma 6.** Let  $\epsilon > 0$  and  $C_1$  and  $C_2$  be contour trees such that all edges have length strictly greater than  $6\epsilon$  (i.e.  $C_1$  and  $C_2$  only have long edges). If  $d_{FD}(C_1, C_2) \leq \epsilon$ , there is an isomorphism between  $C_1$  and  $C_2$  they are  $6\epsilon$ -compatible.

*Proof.* There should be a matching between the nodes on both trees. Let vertices  $a, b \in C_1$  map close to the same vertex  $u \in C_2$ .  $d_x(a, b) > 6\epsilon$  and  $d_y(\Phi(a), \Phi(b)) > 4\epsilon$  since all edges are long. However, if they both map within  $2\epsilon$  of  $u$ , then  $d_y(\Phi(a), \Phi(b)) \leq 4\epsilon$ . Therefore, there is a one to one mapping between the vertices.

Any edge that exist between vertices  $a$  and  $b$  that corresponds to vertices  $u$  and  $v$  means  $u$  and  $v$  should also have an edge between them. Since this is a tree, a correspondence can not exist if the edge doesn't exist. Furthermore, corresponding edges should have at most  $2\epsilon$  distortion because every path on the tree is unique.

Let any arbitrary path in  $C_1$  corresponding to path  $C_2$ . If the paths starts and ends on the middle on edges, the distortion from the middle of edges to the closest vertices on the

path will be  $4\epsilon$  total. The distortion between any two vertices is  $2\epsilon$ . Therefore, the maximum distortion is  $6\epsilon$  between any two points.

Similar to planar graphs with low Gromov-Hausdorff distance, vertices should map near vertices of the same degree, edges should map onto edges of similar length, and the APSP between the matched contour trees should have distortion less than  $2\epsilon$ . Therefore,  $C_1$  and  $C_2$  they are  $6\epsilon$ -compatible. ■

**Lemma 7.** Let  $\epsilon > 0$  and  $C_1$  and  $C_2$  be contour trees such that all edges have length strictly greater than  $6\epsilon$  (i.e.  $C_1$  and  $C_2$  only have long edges). If  $X$  and  $Y$  are isomorphic and  $2\epsilon$ -compatible,  $d_{FD}(C_1, C_2) \leq 2\epsilon$ .

*Proof.* We define our correspondence by matching every vertex and edge between  $C_1$  and  $C_2$  according to their valid  $\epsilon$ -compatibility. If two corresponding edges have different length, we will map each point linearly according to their relative distance between the incident vertices.

Suppose that  $d_{FD}(C_1, C_2) > 2\epsilon$ . There is a corresponding path  $(a, b) \in C_1$  paired with  $(u, v) \in C_2$  such that  $|d_f(a, b) - d_g(u, v)| > 2\epsilon$ . Any maximum distortion starting and ending on middle on edges can be shifted because the maps are continuous. So no matter where  $a, b, u$ , or  $v$  starts, it is possible to shift these points in a manner that one of the endpoints are indecent to vertices. Let the distances  $d_f(a, b)$  and  $d_g(u, v)$  start on vertices  $a$  and  $u$ , they will have the distortion  $2\epsilon$  to the vertex before where  $b$  and  $v$  are on. Similarly, the edge  $b$  and  $v$  are on also have at most  $2\epsilon$  distortion. Therefore, the function distortion distance is less than  $2\epsilon$ . ■

### 3.2 Approximating $d_{FD}$ for Trees with Long Edges

Similar to calculating  $d_{GH}$ , let  $C_1$  and  $C_2$  be contour trees and test all pairs of leaf vertices as roots of our contour trees. Given a pair of leaf nodes as roots find if the tree is  $\epsilon$ -compatible.



If these trees are  $\epsilon$ -compatible,  $d_{FD}(C_1, C_2) \leq \epsilon$ . Starting from the leaves, find the minimum valid  $\epsilon$  in a recursive manner.

Given the subtrees  $C_1^\alpha$  and  $C_2^\beta$  rooted at node  $\alpha$  and  $\beta$ , we can find the minimum  $\epsilon$ -compatibility by picking the maximum of either the difference between the heights of  $\alpha$  and  $\beta$  or the distortion between the children of  $\alpha$  and  $\beta$ . The minimum distortion of the children can be found by constructing a bipartite graph. Let  $a_1, \dots, a_n$  be the children nodes of  $\alpha$  and  $b_1, \dots, b_m$  be the children nodes of  $\beta$ . The vertices in each side of the bipartite graph will represent the children of  $\alpha$  or  $\beta$ . In the bipartite graph, we set the weight of the edge  $a_i$  to  $b_i$  to the  $\delta$ -compatibility between subtrees  $C_1^{a_i}$  and  $C_2^{b_i}$ . We also keep track of the *up distortion*  $\sigma(\alpha, \beta)$  which is the difference between the longest distance between distinct children of  $\alpha$  and between distinct children of  $\beta$ . In this case,  $\sigma(\alpha, \beta)$  represents the worse distortion between paths that crosses the roots  $\alpha$  and  $\beta$ .

Suppose the worse path on this tree will start in two matching subtrees rooted at  $\alpha$  and  $\beta$ , go up to their lowest matching ancestor, and down to the other matching subtree. We have calculated the minimum  $\delta$ -compatibility which is also the between these trees and we know the largest distortion between paths going through  $\alpha$  and  $\beta$  is the twice the up distortion  $2\sigma(\alpha, \beta)$ . Therefore, the total distortion from this construction is  $\delta + 2\sigma(\alpha, \beta)$ .

**Theorem 8.** Let  $\epsilon > 0$  and  $C_1$  and  $C_2$  be contour trees such that all edges have length strictly greater than  $6\epsilon$  (i.e.  $C_1$  and  $C_2$  only have long edges). There exist a polynomial time 3-approximation algorithm for finding the functional distortion distance between  $C_1$  and  $C_2$ .

## CHAPTER 4

### CONCLUSION

We have shown that graphs with long edges will have a bipartite matching between edges and vertices, induced by an optimal Gromov-Hausdorff distance correspondence. We presented constant factor approximation algorithms for Gromov-Hausdorff distance and functional distortion distance in metric graphs/trees with long edges. Possibility for future work includes a possible  $O(n)$  approximation algorithm for Gromov-Hausdorff distance between planar graphs as the naive approach of contracting any edges that are not long edges appears to yield an  $O(n^2)$  approximation based on previous work from Agarwal et. al. [1]. Similarly, the approach presented by Agarwal et. al showed that it may be possible to find a  $O(n)$  approximation algorithm for functional distortion distance by computing the candidate values and turning the optimization function in Section 3.2 into a decision procedure.

## REFERENCES

- [1] Agarwal, P. K., K. Fox, A. Nath, A. Sidiropoulos, and Y. Wang (2018). Computing the gromov-hausdorff distance for metric trees. *ACM Transactions on Algorithms (TALG)* 14(2), 1–20.
- [2] Battista, G. D. and R. Tamassia (1996). On-line maintenance of triconnected components with spqr-trees. *Algorithmica* 15, 302–318.
- [3] Bauer, U., X. Ge, and Y. Wang (2014). Measuring distance between reeb graphs. In *Proceedings of the thirtieth annual symposium on Computational geometry*, pp. 464–473.
- [4] Bauer, U., E. Munch, and Y. Wang (2014). Strong equivalence of the interleaving and functional distortion metrics for reeb graphs. *arXiv preprint arXiv:1412.6646*, 1–15.
- [5] Carr, H., J. Snoeyink, and U. Axen (2003). Computing contour trees in all dimensions. *Computational Geometry* 24(2), 75–94.
- [6] Di Battista, G. and R. Tamassia (1989). Incremental planarity testing. In *30th Annual Symposium on Foundations of Computer Science*, pp. 436–441.
- [7] Even, S. and R. E. Tarjan (1976). Computing an st-numbering. *Theoretical Computer Science* 2(3), 339 – 344.
- [8] Hopcroft, J. E. and J. K. Wong (1974). Linear time algorithm for isomorphism of planar graphs (preliminary report). In *Proceedings of the Sixth Annual ACM Symposium on Theory of Computing, STOC 74*, New York, NY, USA, pp. 172–184. Association for Computing Machinery.
- [9] Kuklук, J. P., L. B. Holder, and D. J. Cook (2004). Algorithm and experiments in testing planar graphs for isomorphism. *J. Graph Algorithms Appl.* 8, 313–356.
- [10] Lesnick, M. (2011). The optimality of the interleaving distance on multidimensional persistence modules. *CoRR abs/1106.5305*.
- [11] Mémoli, F. (2007, 01). On the use of gromov-hausdorff distances for shape comparison. *Proceedings Point Based Graphics*, 81–90.
- [12] Mémoli, F. (2008). Gromov-hausdorff distances in euclidean spaces. In *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1–8. IEEE.
- [13] Mémoli, F. (2012). Some properties of gromov-hausdorff distances. *Discrete & Computational Geometry* 48(2), 416–440.

- [14] Punnen, A. P. and K. Nair (1994). Improved complexity bound for the maximum cardinality bottleneck bipartite matching problem. *Discrete Applied Mathematics* 55(1), 91 – 93.
- [15] Tarjan, R. (1972). Depth-first search and linear graph algorithms. *SIAM Journal on Computing* 1(2), 146–160.
- [16] Touli, E. F. and Y. Wang (2018). Fpt-algorithms for computing gromov-hausdorff and interleaving distances between trees. *arXiv preprint arXiv:1811.02425*, 1–35.
- [17] Whitney, H. (1933). A set of topological invariants for graphs. *American Journal of Mathematics* 55(1), 231–235.

## **BIOGRAPHICAL SKETCH**

Seong Ioi Wang was born in Macau in June of 1998. He started studying Computer Science at The Woodlands High School where he joined the Computer Science UiL team. After high school, Seong went to The University of Texas at Dallas and received his bachelor's in Computer Science with a minor in Mathematics in 2019. He is now pursuing his master's in Computer Science with a concentration in data science. After his master's, Seong will work as a software engineer at Applied Research Associates, a private scientific research and engineering company.

## CURRICULUM VITAE

# Seong Ioi Wang

April 6, 2020

### Contact Information:

Email: [sxw160530@utdallas.edu](mailto:sxw160530@utdallas.edu)

### Educational History:

BS, Computer Science, The University of Texas at Dallas, 2019

MS, Computer Science, The University of Texas at Dallas, 2020

*Approximating Gromov-Hausdorff Distance in Planar Graphs*

M.S. Thesis

Computer Science Department, The University of Texas at Dallas

Advisor: Dr. Kyle Fox

### Employment History:

Software Engineer Intern, CoreLogic, June 2019 – August 2019

Software Developer Intern, OnPoynt, August 2018 – December 2018