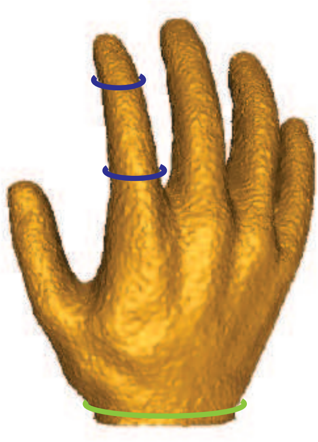
Topological measures of similarity

Erin Wolf Chambers
Saint Louis University
erin.chambers@slu.edu

Motivation: Measuring Similarity Between Curves

How can we tell when two cycles or curves are similar to each other?



Applications

Similarity measures have many potential applications:

- Analyzing GIS data
- Map analysis and simplification
- Handwriting recognition
- Computing "good" morphings between curves
- Surface parameterizations

Applications

Similarity measures have many potential applications:

- Analyzing GIS data
- Map analysis and simplification
- Handwriting recognition
- Computing "good" morphings between curves
- Surface parameterizations

There are many different ways to check similarity. Most focus on either the geometry or the topology of the curve and the ambient space.

Hausdorff distance

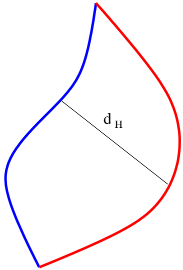
Hausdorff distance is one way to measure distance: consider any point on one curve and find the closest point on the other curve. Take the largest of all these distances.

Hausdorff distance

Hausdorff distance is one way to measure distance: consider any point on one curve and find the closest point on the other curve. Take the largest of all these distances.

More formally, given two curves $\gamma_1, \gamma_2 : [0,1] \to M$:

$$d_H(\gamma_1, \gamma_2) = \max\{\sup_{s\in[0,1]} \inf_{t\in[0,1]} d(\gamma_1(s), \gamma_2(t)), \sup_{t\in[0,1]} \inf_{s\in[0,1]} d(\gamma_1(s), \gamma_2(t))\}$$

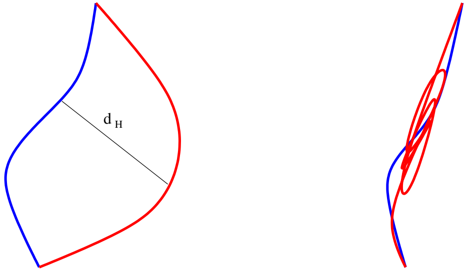


Hausdorff distance

Hausdorff distance is one way to measure distance: consider any point on one curve and find the closest point on the other curve. Take the largest of all these distances.

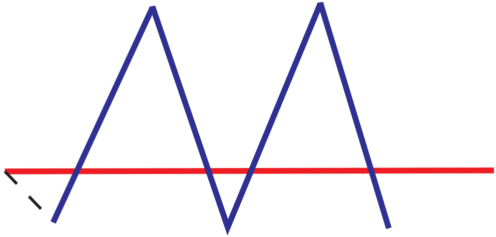
More formally, given two curves $\gamma_1, \gamma_2 : [0,1] \to M$:

$$d_H(\gamma_1,\gamma_2) = \max\{\sup_{s\in[0,1]} \inf_{t\in[0,1]} d(\gamma_1(s),\gamma_2(t)), \sup_{t\in[0,1]} \inf_{s\in[0,1]} d(\gamma_1(s),\gamma_2(t))\}$$



Fréchet Distance

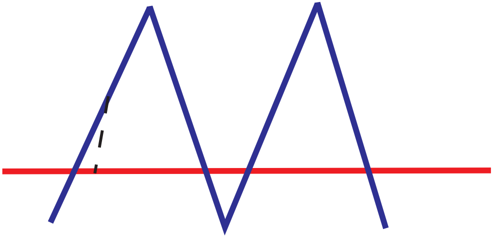
With Fréchet distance (or dog leash distance), the flow of the curve is accounted for.



Imagine a man walking along one curve and a dog along the other, with a leash always connecting them, and minimize the length of the longest leash.

Fréchet Distance

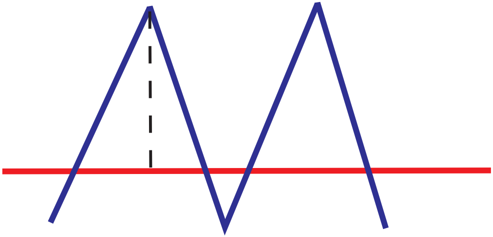
With Fréchet distance (or dog leash distance), the flow of the curve is accounted for.



Imagine a man walking along one curve and a dog along the other, with a leash always connecting them, and minimize the length of the longest leash.

Fréchet Distance

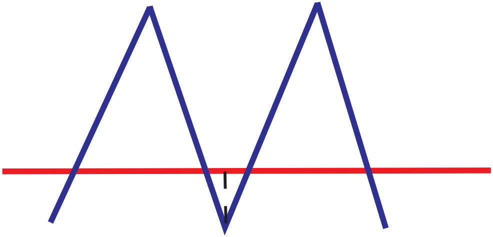
With Fréchet distance (or dog leash distance), the flow of the curve is accounted for.



Imagine a man walking along one curve and a dog along the other, with a leash always connecting them, and minimize the length of the longest leash.

Fréchet Distance

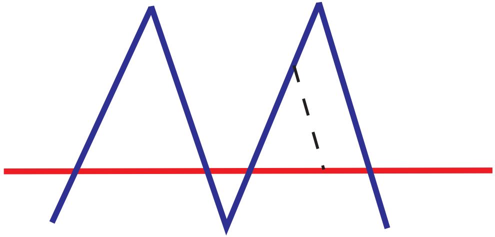
With Fréchet distance (or dog leash distance), the flow of the curve is accounted for.



Imagine a man walking along one curve and a dog along the other, with a leash always connecting them, and minimize the length of the longest leash.

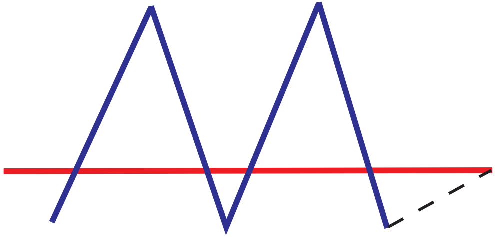
Fréchet Distance

With Fréchet distance (or dog leash distance), the flow of the curve is accounted for.



Imagine a man walking along one curve and a dog along the other, with a leash always connecting them, and minimize the length of the longest leash.

With Fréchet distance (or dog leash distance), the flow of the curve is accounted for.



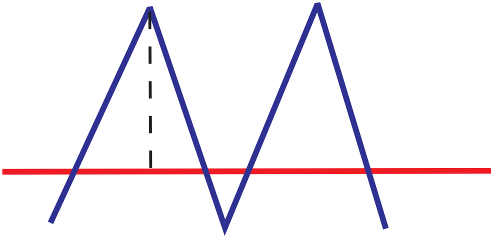
Imagine a man walking along one curve and a dog along the other, with a leash always connecting them, and minimize the length of the longest leash.

Fréchet Distance

More formally, given two curves γ_1 and γ_2, the Fréchet distance is:

$$F(A,B) = \inf_{\alpha,\beta} \max_{t\in[0,1]} \{d(\gamma_1(\alpha(t)), \gamma_2(\beta(t)))\}$$

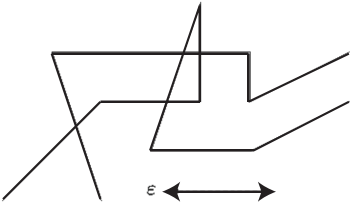
where α and β are reparameterizations of $[0,1]$.

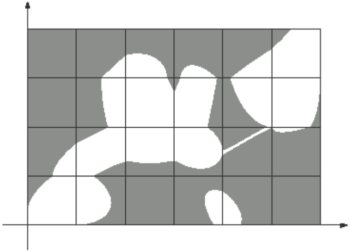


Alt and Godau gave the first algorithm to compute this for piecewise linear curves in the Eucldean space; their algorithm runs in $O(mn\log(mn))$ time.

Main tool: Free space diagram

Consider each pair of segments from the two curves, and calculate which portions are within ϵ of each other.





We build the *free space diagram* by forming the n by m grid, and determine if there is a matching that keeps the leash $\leq \epsilon$ by searching in this grid.

Fréchet distance continued

Since the initial algorithm, it has been studied extensively: applications, approximations, improved algorithms for restricted classes of curves, and lower bounds are just a few of the many results.

Fréchet distance continued

Since the initial algorithm, it has been studied extensively: applications, approximations, improved algorithms for restricted classes of curves, and lower bounds are just a few of the many results.

In addition, Fréchet distance has also been considered in higher dimensions:

- It is NP-Hard to compute the Fréchet distance between two surfaces [Godau 1998].
- Even NP-Hard to compute between terrains or polygons with holes [Buchin-Buchin-Schulz 2010].
- Still NP hard even for surfaces traced by curves [Buchin-Ophelders-Speckmann 2015].
- There is a $(1+\epsilon)$-approximation algorithm for computing Fréchet distance between genus zero surfaces, where the running time is bounded if the input surfaces are "nice" [Nayyeri Xu 2016].

Geodesic Fréchet Distance

When not in $\mathbb{R}^d$, distance calculations are more complex since shortest paths are not unique.

Geodesic Fréchet Distance

When not in $\mathbb{R}^d$, distance calculations are more complex since shortest paths are not unique.

Definition

A geodesic is a path that avoids any obstacles and cannot be locally shortened by perturbations.

Geodesic Fréchet Distance

When not in $\mathbb{R}^d$, distance calculations are more complex since shortest paths are not unique.

Definition

A geodesic is a path that avoids any obstacles and cannot be locally shortened by perturbations.

In geodesic Fréchet distance, the leash is required to be a geodesic in the ambient space.

When not in $\mathbb{R}^d$, distance calculations are more complex since shortest paths are not unique.

Definition

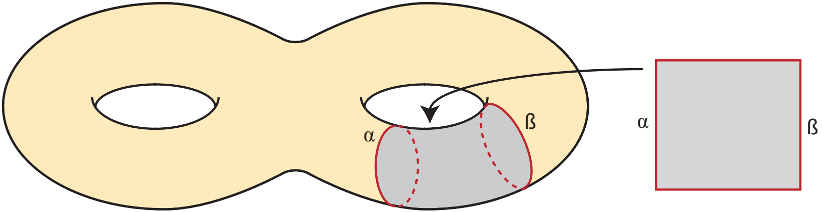
A geodesic is a path that avoids any obstacles and cannot be locally shortened by perturbations.

In geodesic Fréchet distance, the leash is required to be a geodesic in the ambient space.

Algorithms are known in some limited settings, such as convex polytopes [Maheshwari and Yi 2005] and simple polygons [Cook Wenk 2008]. However, much remains open.

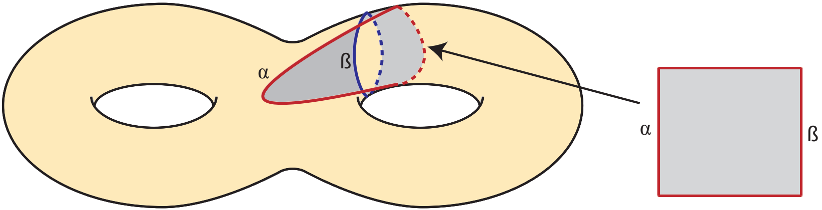
Definition

A homotopy is a continuous deformation of one path to another. More formally, a homotopy between two curves α and β on a surface M is a continuous function $H : [0,1] \times [0,1] \to M$ such that $H(\cdot, 0) = \alpha(\cdot)$ and $H(\cdot, 1) = \beta(\cdot)$.



Definition

A homotopy is a continuous deformation of one path to another. More formally, a homotopy between two curves α and β on a surface M is a continuous function $H : [0,1] \times [0,1] \to M$ such that $H(\cdot, 0) = \alpha(\cdot)$ and $H(\cdot, 1) = \beta(\cdot)$.



Testing if two curves are homotopic

Testing if two curves are homotopic has been studied.

- Cabello et al (2004) give an algorithm to test if two paths in the plane minus a set of obstacles are homotopic in $O(n^{3/2} \log n)$ time; there is also an output sensative algorithm that takes $O(\log^2 n)$ time per output vertex [Bespamyatnikh 2003].
- Given a graph cellularly embedded on a surface and two closed walks on that graph, there is an $O(n)$ time algorithm to decide if the two walks are homotopic [Dey and Guha 1999, Lazarus and Rivaud 2011, Erickson and Whittlesey 2012].

Combinatorially optimal homotopies

There is work [Chang-Erickson 2016] on finding the "best" homotopy, as well; usually, this involves minimizing number of simplifications moves to untangle a curve.

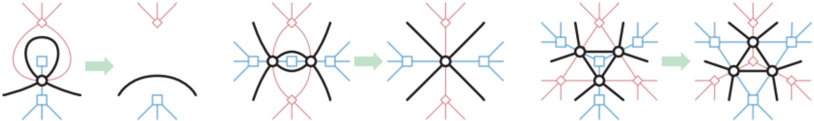
Figure 1.1. Homotopy moves 1→0, 2→0, and 3→3.

Combinatorially optimal homotopies

There is work [Chang-Erickson 2016] on finding the "best" homotopy, as well; usually, this involves minimizing number of simplifications moves to untangle a curve.

Figure 1.1. Homotopy moves 1→0, 2→0, and 3→3.

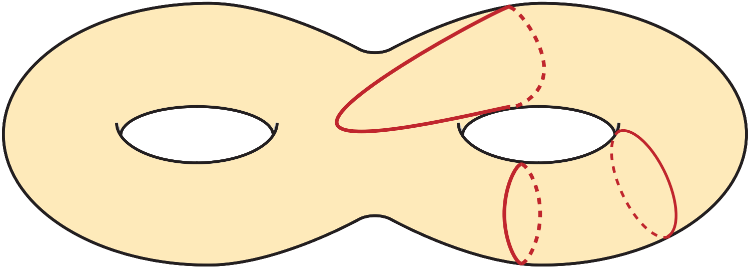
In the plane, they prove this is $\Theta(n^{3/2})$.



This connects to older results [Steinitz 1916, Francis 1969, Truemper 1989, Feo and Provan 1993, Noble and Welsh 2000], and electrical moves on the medial graph of the input planar graphs.

Beyond testing homotopy

However, in many applications we'd like to include more of a notion of the geometry of the underlying space, as well.



Homotopic Fréchet Distance

The definition of Fréchet distance or geodesic Fréchet distance will directly generalize to surfaces, but does not take homotopy into account. Essentially, either definition allows the leash to jump discontinously.

Homotopic Fréchet Distance

The definition of Fréchet distance or geodesic Fréchet distance will directly generalize to surfaces, but does not take homotopy into account. Essentially, either definition allows the leash to jump discontinously.

Homotopic Fréchet is adds a constraint that the curves must be homotopic, and the leashes must move continuously in the ambient space [C.-Colin de Verdiére-Erickson-Lazard-Lazarus-Thite 2009].

Homotopic Fréchet Distance

The definition of Fréchet distance or geodesic Fréchet distance will directly generalize to surfaces, but does not take homotopy into account. Essentially, either definition allows the leash to jump discontinously.

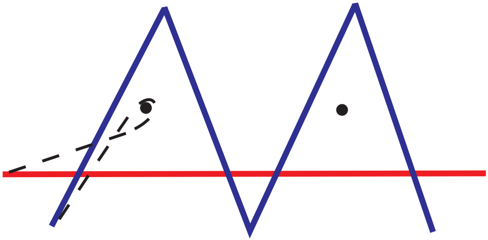
Homotopic Fréchet is adds a constraint that the curves must be homotopic, and the leashes must move continuously in the ambient space [C.-Colin de Verdiére-Erickson-Lazard-Lazarus-Thite 2009].

Intuitively, curves with small homotopic Fréchet distance will be close both geometrically and topologically.

Homotopic Fréchet Distance

The homotopic Fréchet distance is the length of the shortest leash we can can use for our homotopy. Formally,

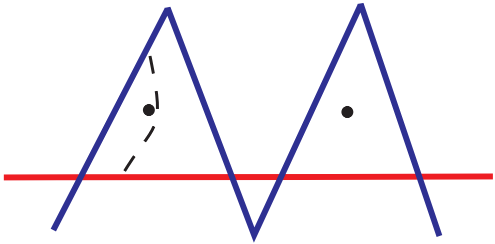
$$d_F(\gamma_1, \gamma_2) = \inf_{\text{homotopies } H} \{\sup\{|H(\cdot, t))| \mid t \in [0,1]\}\}$$



Homotopic Fréchet Distance

The homotopic Fréchet distance is the length of the shortest leash we can can use for our homotopy. Formally,

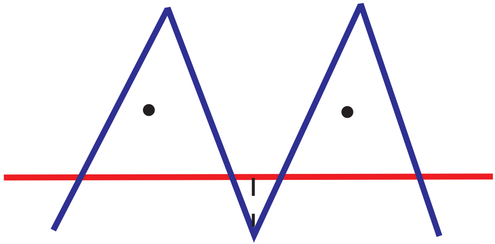
$$d_F(\gamma_1, \gamma_2) = \inf_{\text{homotopies } H} \{\sup\{|H(\cdot, t))| \mid t \in [0, 1]\}\}$$



Homotopic Fréchet Distance

The homotopic Fréchet distance is the length of the shortest leash we can can use for our homotopy. Formally,

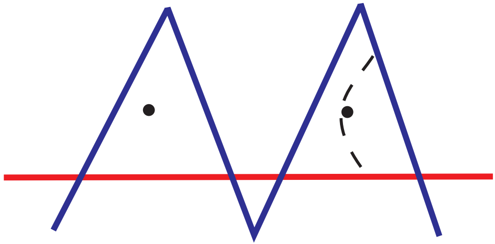
$$d_F(\gamma_1, \gamma_2) = \inf_{\text{homotopies } H} \{\sup\{|H(\cdot, t))| \mid t \in [0,1]\}\}$$



Homotopic Fréchet Distance

The homotopic Fréchet distance is the length of the shortest leash we can can use for our homotopy. Formally,

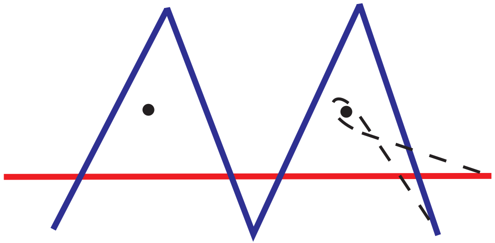
$$d_F(\gamma_1, \gamma_2) = \inf_{\text{homotopies } H} \{\sup\{|H(\cdot, t))| \mid t \in [0, 1]\}\}$$



Homotopic Fréchet Distance

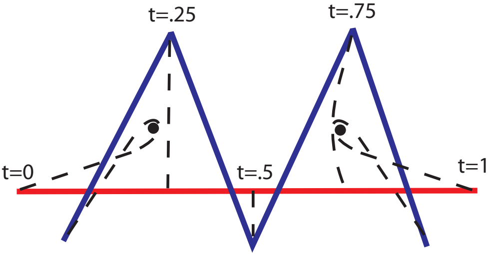
The homotopic Fréchet distance is the length of the shortest leash we can can use for our homotopy. Formally,

$$d_F(\gamma_1, \gamma_2) = \inf_{\text{homotopies } H} \{\sup\{|H(\cdot, t))| \mid t \in [0, 1]\}\}$$



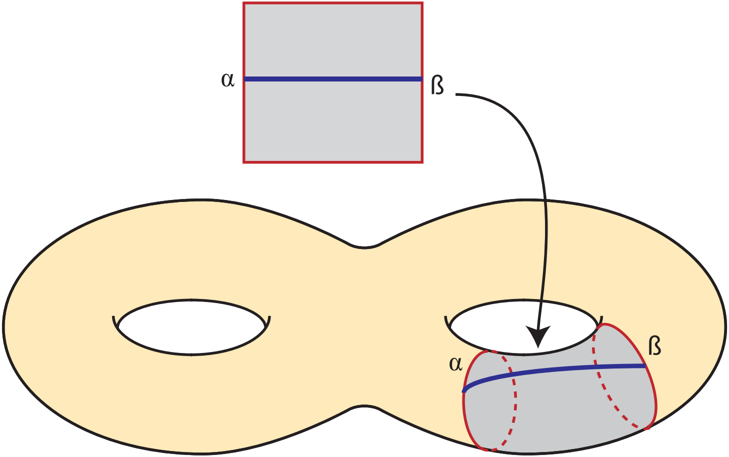
The homotopic Fréchet distance is the length of the shortest leash we can can use for our homotopy. Formally,

$$d_F(\gamma_1, \gamma_2) = \inf_{\text{homotopies } H} \{\sup\{|H(\cdot, t))| \mid t \in [0, 1]\}\}$$



Homotopic Fréchet Distance on a Surface

We could just have easily called this the *width* of the homotopy:



(Note: it is not known how to compute this on surfaces at all.)

Computing the Homotopic Fréchet Distance

There is a polynomial time algorithm algorithm to compute the homotopic Fréchet Distance between two polygonal curves in the plane minus a set of polygonal obstacles [C.-Colin de Verdiére-Erickson-Lazard-Lazarus-Thite 2009].

Computing the Homotopic Fréchet Distance

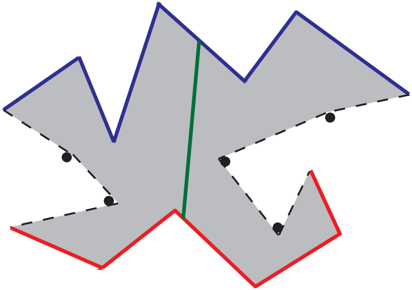
There is a polynomial time algorithm algorithm to compute the homotopic Fréchet Distance between two polygonal curves in the plane minus a set of polygonal obstacles [C.-Colin de Verdiére-Erickson-Lazard-Lazarus-Thite 2009].

The algorithm has some similarities to the work of Alt and Godau, but is considerably more complex since there are an infinite number of homotopy classes to consider.

Key lemma

Lemma

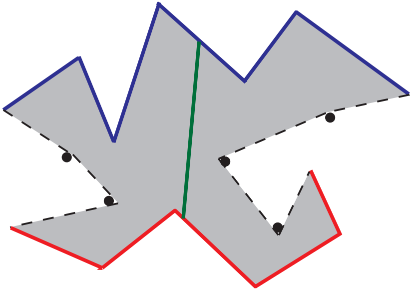
When obstacles are points, an optimal homotopy class contains a straight line segment.



Key lemma

Lemma

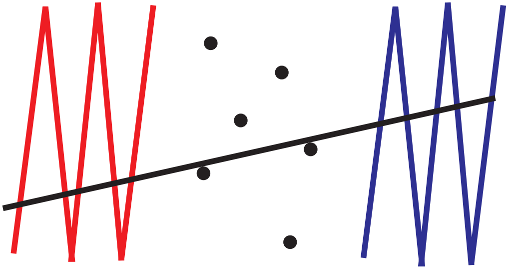
When obstacles are points, an optimal homotopy class contains a straight line segment.



This allows us to brute force a set of possible homotopy classes which could be optimal, by trying all straight line segments.

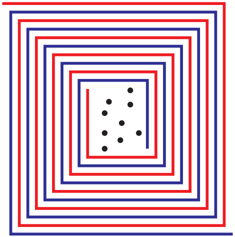
How bad could this be?

However, there are still a lot of possible straight line segments::



How bad could this be?

However, there are still a lot of possible straight line segments::



For each of these, the algorithm needs to run a free space computation (like in standard Fréchet distance calculations), so the total running time polynomial but not fast.

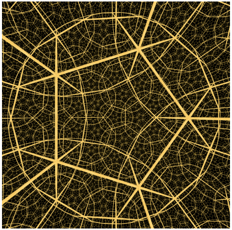
Homotopic Fréchet Distance in other settings

It is unlikely our algorithm will generalize to surfaces, since it heavily relies on nonpositive curvature.

Homotopic Fréchet Distance in other settings

It is unlikely our algorithm will generalize to surfaces, since it heavily relies on nonpositive curvature.

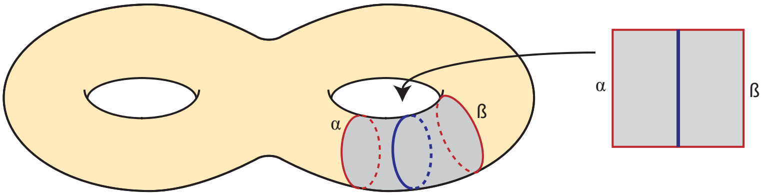
We can generalize the key lemmas to any surface of nonpositive curvature. However, the algorithmic tools in those settings are lacking.



Height of a homotopy

The height of a homotopy is an orthogonal definition to homotopic Fréchet distance:

$$d_{HH}(\gamma_1, \gamma_2) = \inf_{\text{homotopies } H} \{\sup\{|H(s, \cdot)| \mid s \in [0,1]\}\}$$



Computing homotopy height

No algorithm is known to compute the homotopy height between two curves in any setting.

We do know that the problem is in NP.

Computing homotopy height

No algorithm is known to compute the homotopy height between two curves in any setting.

We do know that the problem is in NP.

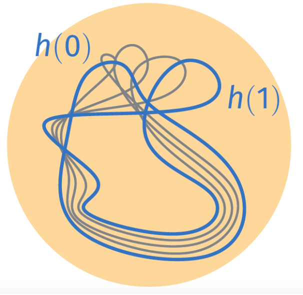
(This is all joint work with Arnaud de Mesmay and Tim Ophelders, and the first part is also joint with Gregory Chambers and Regina Rotman.)

Formalizing the notation

A homotopy through closed curves is a continuous map $h : S^1 \times [0,1] \to \Sigma$, where Σ is a triangulated surface.

We let $h(t)$ be the curve $h(\cdot, t)$, and the homotopy goes from $h(0)$ to $h(1)$; the height is then $\sup_t ||h(t)||$.

An isotopy between the two curves is a homotopy where all $h(t)$ are simple curves.



Structure of optimal homotopies

[G. Chambers and Liokumovich] prove that some optimal homotopy is actually an isotopy.

[G. Chambers and Liokumovich] prove that some optimal homotopy is actually an isotopy.

Sketch:

- Take a homotopy of height L from γ to γ'
- Decompose into a sequence of curves $\gamma = \gamma_1, \ldots \gamma_n = \gamma'$, with at most 1 Reidemeister move between each γ_i and γ_{i+1}

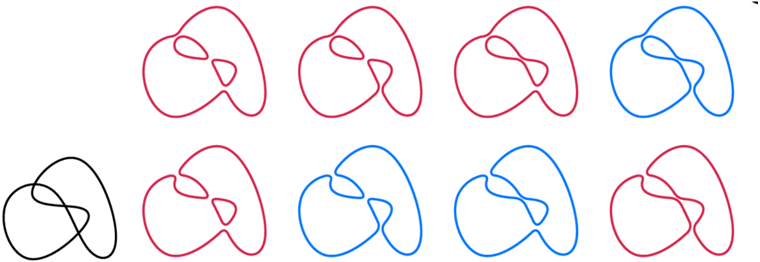
Figure 1.1. Homotopy moves 1→0, 2→0, and 3→3.

Structure of optimal homotopies (cont)

[G. Chambers and Liokumovich] prove that some optimal homotopy is actually an isotopy.

Sketch (cont):

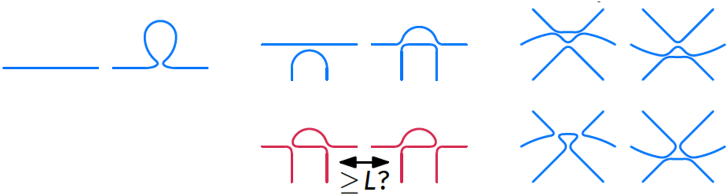
- We then consider all resolutions of the crossings that would get a single, simple curve



[G. Chambers and Liokumovich] prove that some optimal homotopy is actually an isotopy.

Sketch (cont):

- Then construct "trivial" isotopies of height at most L between resolutions that are 1 Reidemeister move apart.



- Note: not all of these have a trivial isotopy between them!

Structure of optimal homotopies (cont)

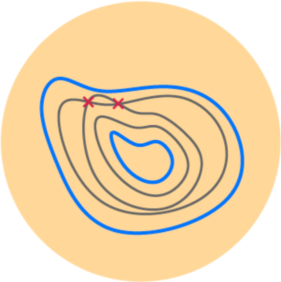
[G. Chambers and Liokumovich] prove that some optimal homotopy is actually an isotopy.

Sketch (cont):

- To fix this, they actually build a graph: vertices are the resolutions, and edges are the trivial isotopies of height $< L$.
- Most of the work is then proving that the graph contains a path from γ to γ'. (Surprisingly, this all boils down to the handshaking lemma.)

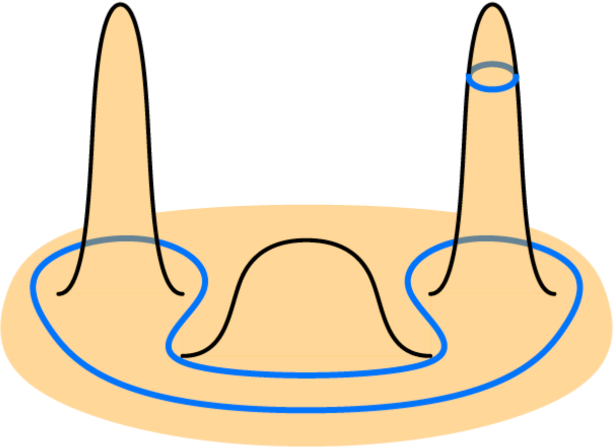
Monotone isotopies

Where we come in, and where I was stuck for over a decade: determining if you can always find a monotone isotopy, so that h_t and $h_{t'}$ are disjoint for any $t < t'$:



Monotone isotopies

Note that these don't always exist! In particular, if you do not start with the boundary of the disk, the best isotopy sometimes won't be monotone:

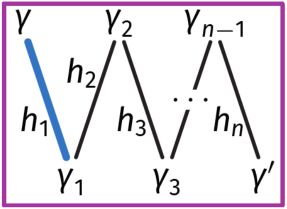


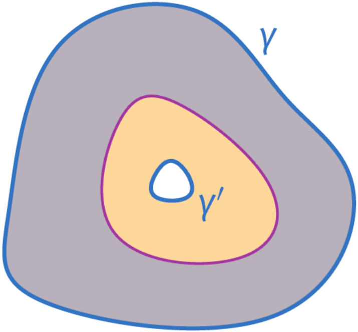
Monotone isotopies

In [Chambers, Chambers, de Mesmay, Ophelders and Rotman] we show that monotone isotopies always exist when the curves bound an annulus.

Proof sketch:

- Decompose the isotopy into monotone sub-isotopies, where h_i goes from γ_i to γ_{i+1}:

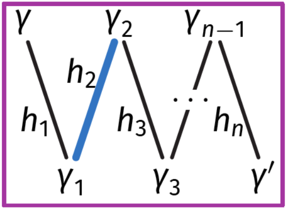


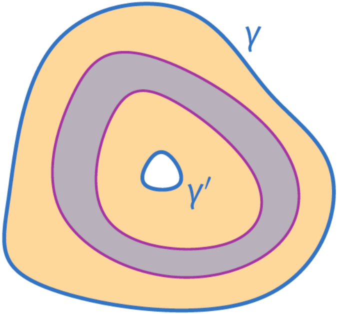


In [Chambers, Chambers, de Mesmay, Ophelders and Rotman] we show that monotone isotopies always exist when the curves bound an annulus.

Proof sketch:

- Decompose the isotopy into monotone sub-isotopies, where h_i goes from γ_i to γ_{i+1}:

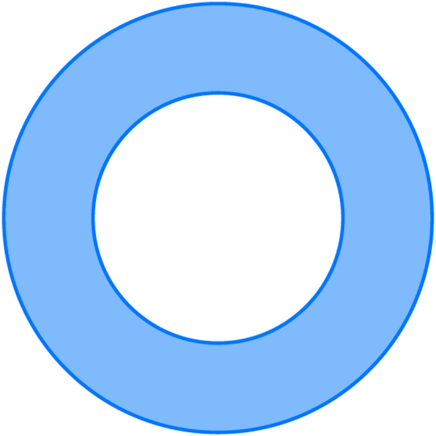




Monotonicity

High level idea:

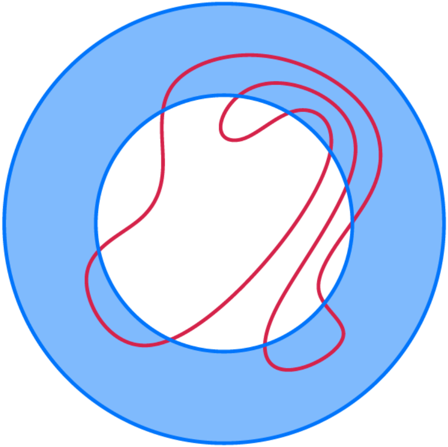
- If later parts (say h_{i+1}) of the homotopy come back inside a previously swept portions, we want to construct a retract which stays outside:



Monotonicity

High level idea:

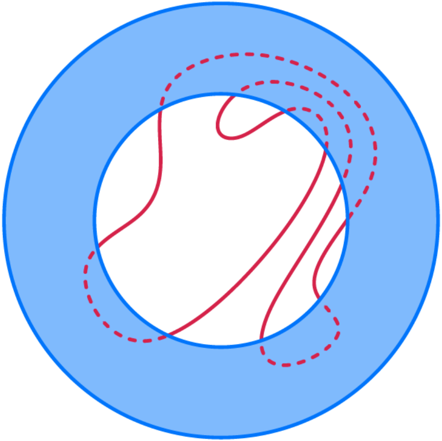
- If later parts (say h_{i+1}) of the homotopy come back inside a previously swept portions, we want to construct a retract which stays outside:



Monotonicity

High level idea:

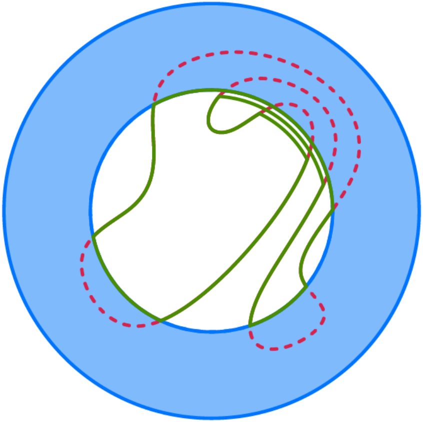
- If later parts (say h_{i+1}) of the homotopy come back inside a previously swept portions, we want to construct a retract which stays outside:



Monotonicity

High level idea:

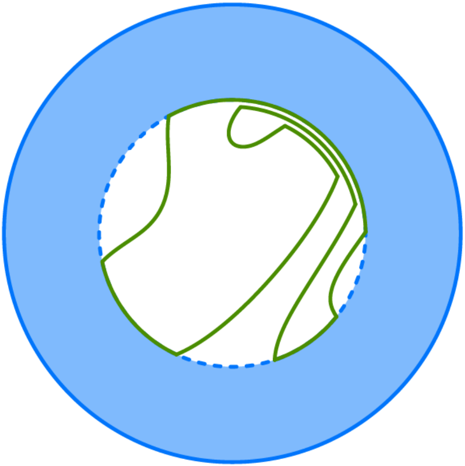
- If later parts (say h_{i+1}) of the homotopy come back inside a previously swept portions, we want to construct a retract which stays outside:



Monotonicity

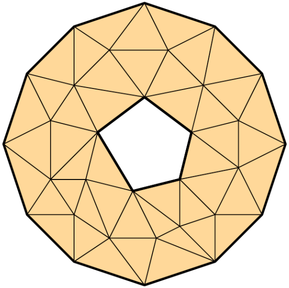
High level idea:

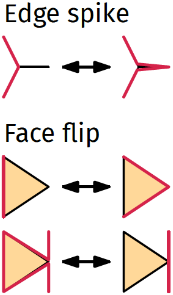
- If later parts (say h_{i+1}) of the homotopy come back inside a previously swept portions, we want to construct a retract which stays outside:



Homotopy height is in NP

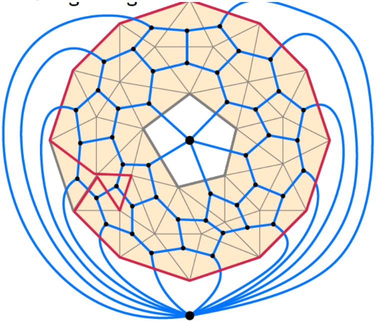
In the discrete settings, we have a triangulated annulus, and we discretize the homotopy accordingly:





Primal versus dual

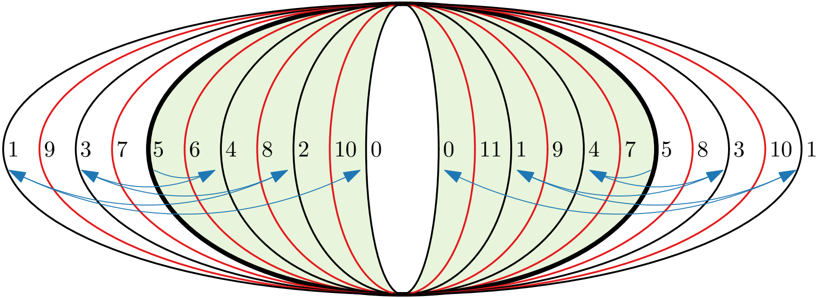
If we dualize the graph, then face moves correspond to change of crossings in the dual graph:



Monotonicity does still hold in this discretized setting if we start on the boundary of the disk, essentially since this is a very simple type of Riemannian disk.

Non-boundary case

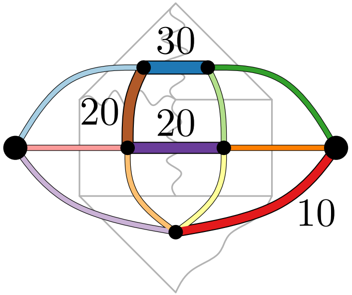
Note that we still cannot assume that the sweep is montone if we do not begin at the boundary:

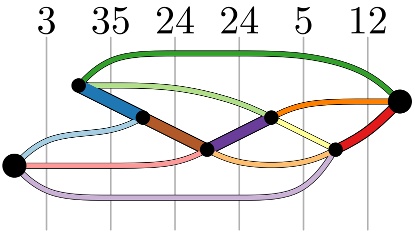


(Example courtesy of Arnaud de Mesmay)

The dual problem

This problem in the dual is very close to the cut width of a graph, where we fix a single embedding:

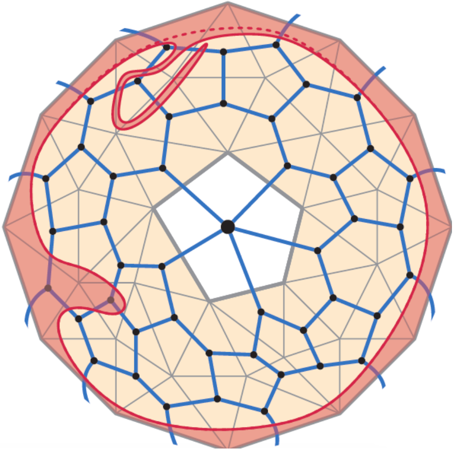




Note: this is open even with unit weights, since NP hardness reductions for cut width alter the embedding of the underlying graph.

Showing NP-Hardness

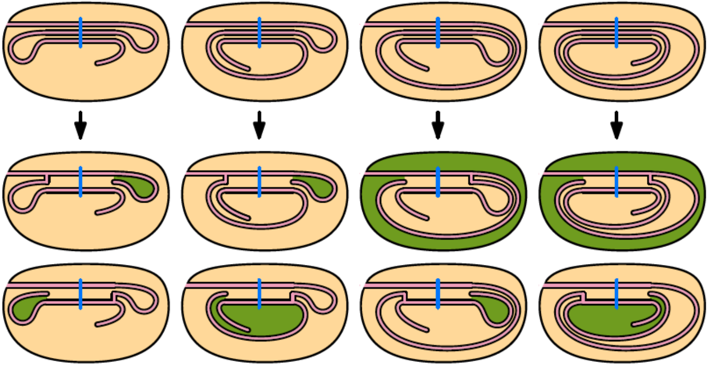
Monotonicity implies that each face flips at most once, but it does not prove the problem is in NP!



The issue is edges: those can be spiked many times from different directions.

Bounding spikes

We show that spikes can be delayed or done early, to simplify the structure. Long paths of spikes will contain spirals, which we can simplify (essentially by case analysis):



In the end, get a quadratic bound on the number of spikes on any given edge, so homotopy height is in NP.

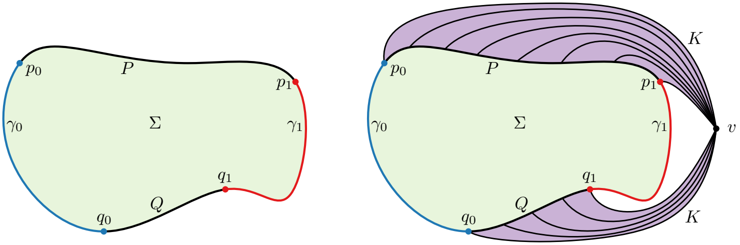
Back to homotopic Fréchet

We were also able to show that a close variant of homotopic Fréchet distance is in NP as well.

Back to homotopic Fréchet

We were also able to show that a close variant of homotopic Fréchet distance is in NP as well.

If you fix the start and end leashes of the homotopy, then you can transform an instance of the homotopic Fréchet problem into one of homotopy height:



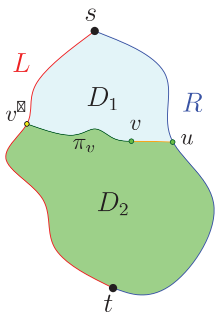
Approximation algorithms

The first (and only) algorithmic work on homotopy height [Har-Peled-Nayyeri-Salavatipour-Sidiropoulos 2012] is $O(\log n)$ approximation algorithm for computing both the homotopy height and the homotopic Fréchet distance between two curves on a PL surface.

Approximation algorithms

The first (and only) algorithmic work on homotopy height [Har-Peled-Nayyeri-Salavatipour-Sidiropoulos 2012] is $O(\log n)$ approximation algorithm for computing both the homotopy height and the homotopic Fréchet distance between two curves on a PL surface.

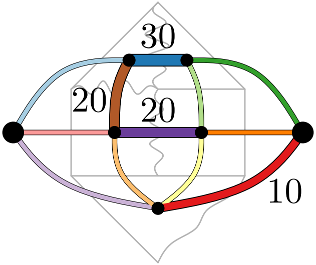
They use a clever divide and conquer algorithm based on shortest paths for homotopy height, and then use this algorithm as a subroutine to solve homotopic Fréchet distance.

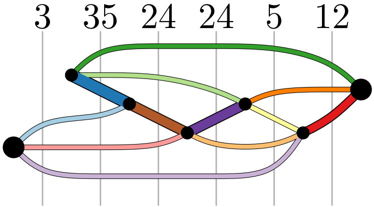


Connections to other graph parameters

As mentioned earlier, homotopy height is quite naturally related to several other parameters.

Recall: homotopy height in a graph where the curve does not spike is the same as cut width of the dual graph (where embedding stays fixed):





We will call this *simple homotopy height*.

Bar Visibility Representation

A *bar visibility representation* of a graph G is a representation where each vertex is mapped to a bar, and any two vertices are connected in G if and only if the corresponding bars have a verticle line segment that connected them and intersects no other bar.

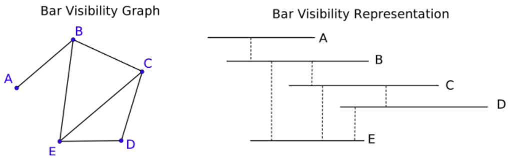


Figure 1: A bar visibility graph and its bar visibility representation.

From [Babbit 2012]

Bar visibility

If we require bars to be drawn on horizontal integer lines, then the bar visibility height is the smallest height possible.

Bar visibility

If we require bars to be drawn on horizontal integer lines, then the bar visibility height is the smallest height possible.

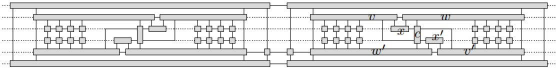
It is known that any planar graph has a bar visibility representation [Wismath 1985, Tamassia-Tollis 1986, Rosentiehl-Tarjan 1986].

Bar visibility

If we require bars to be drawn on horizontal integer lines, then the bar visibility height is the smallest height possible.

It is known that any planar graph has a bar visibility representation [Wismath 1985, Tamassia-Tollis 1986, Rosentiehl-Tarjan 1986].

Bar visibility height is always less than or equal to 2 times the straight line drawing height: the minimum height grid such that G can be embedded on integer points and drawn with straight line edges [Biedl 2014]:



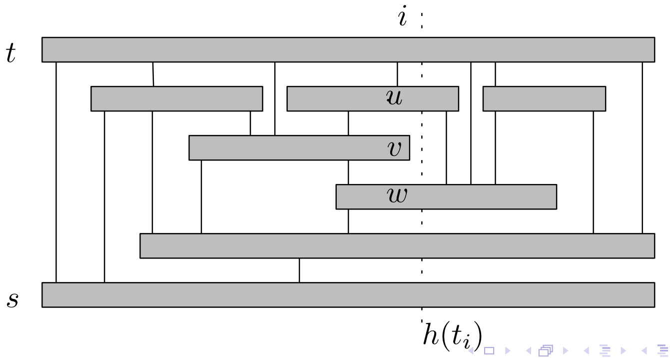
Bar visibility and simple homotopy height

We [Biedl et. al, unpublished] also consider a new variant where we fix vertices s and t on the outer face, and ask for the minimum visibility height that places s on the top and t on the bottom of the representation.

Bar visibility and simple homotopy height

We [Biedl et. al, unpublished] also consider a new variant where we fix vertices s and t on the outer face, and ask for the minimum visibility height that places s on the top and t on the bottom of the representation.

We prove that this is in fact exactly the same as simple homotopy height:



Node searching or sweeping

Searching is another graph theory parameter, modeling how long it takes to sweep through a graph. In all variants, the edges of the graph are contaminated, and the graph must be cleared by guards. If at any point a cleared edge has a path to a contaminated one with no guards on the path, then it becomes recontaminated.

Node searching or sweeping

Searching is another graph theory parameter, modeling how long it takes to sweep through a graph. In all variants, the edges of the graph are contaminated, and the graph must be cleared by guards. If at any point a cleared edge has a path to a contaminated one with no guards on the path, then it becomes recontaminated.

- Search number: At each step, you may add a new guard to any vertex, remove a guard from a vertex, or move a guard along an indicent edge of the current vertex in order to reach a new vertex. An edge is cleared when a guard moves over it.

Node searching or sweeping

Searching is another graph theory parameter, modeling how long it takes to sweep through a graph. In all variants, the edges of the graph are contaminated, and the graph must be cleared by guards. If at any point a cleared edge has a path to a contaminated one with no guards on the path, then it becomes recontaminated.

- Search number: At each step, you may add a new guard to any vertex, remove a guard from a vertex, or move a guard along an indicent edge of the current vertex in order to reach a new vertex. An edge is cleared when a guard moves over it.
- Connected search number: The same as node searching, but the set of edges cleared stays connected through the search.

Node searching or sweeping

Searching is another graph theory parameter, modeling how long it takes to sweep through a graph. In all variants, the edges of the graph are contaminated, and the graph must be cleared by guards. If at any point a cleared edge has a path to a contaminated one with no guards on the path, then it becomes recontaminated.

- Search number: At each step, you may add a new guard to any vertex, remove a guard from a vertex, or move a guard along an indicent edge of the current vertex in order to reach a new vertex. An edge is cleared when a guard moves over it.
- Connected search number: The same as node searching, but the set of edges cleared stays connected through the search.
- Monotonic search number: If the set of cleared edges only grows at every stage, then the search is monotonic.

Connection to homotopy height

Node searching has clear connections to homotopy height: homotopies are one type of search. (This is actually why we originally looked at it, since node searching is always monotonic [LaPaugh 1993, Bienstock-Seymour 1991].)

Connection to homotopy height

Node searching has clear connections to homotopy height: homotopies are one type of search. (This is actually why we originally looked at it, since node searching is always monotonic [LaPaugh 1993, Bienstock-Seymour 1991].)

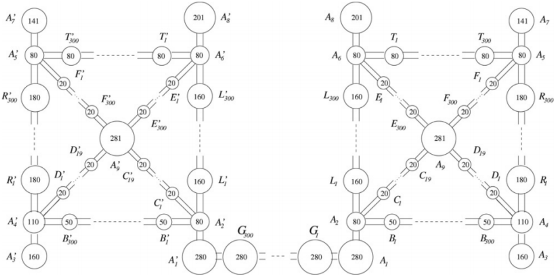
However, homotopy height is actually strictly stronger than even connected graph searching: both sides of the “cut” must say connected for it to be a homotopy.

Connection to homotopy height

Node searching has clear connections to homotopy height: homotopies are one type of search. (This is actually why we originally looked at it, since node searching is always monotonic [LaPaugh 1993, Bienstock-Seymour 1991].)

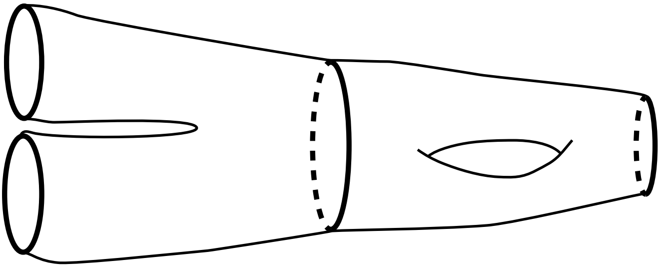
However, homotopy height is actually strictly stronger than even connected graph searching: both sides of the "cut" must say connected for it to be a homotopy.

Interestingly, it is known that connected search number is NOT monotonic [Yang, Dyer, Alspach 2009].



Homology "height" (or length, more accurately)

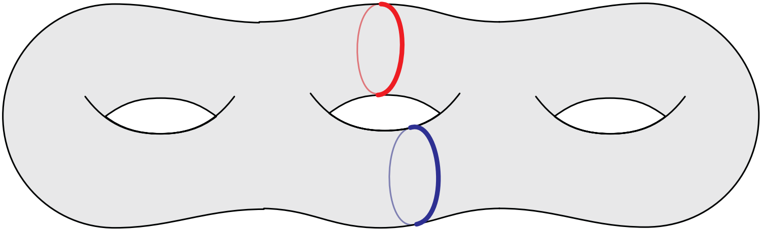
- Homology is a coarser invariant than homotopy - all homotopies produce homologies, but not all homologies come from homotopies.
- In general, much more tractable - reduces to a linear algebra problem, and software is widely available and highly optimized.



Homologous Subgraphs

Definition

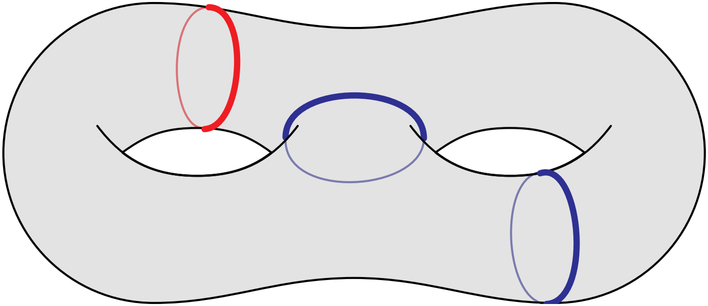
Two even subgraphs are $\mathbb{Z}_2$-homologous if their union forms a cut on the surface.



Homologous Subgraphs

Definition

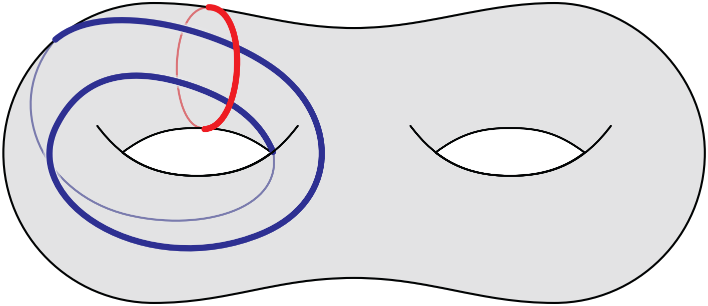
Two even subgraphs are $\mathbb{Z}_2$-homologous if their union forms a cut on the surface.



Homologous Subgraphs

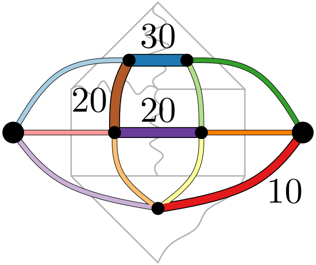
Definition

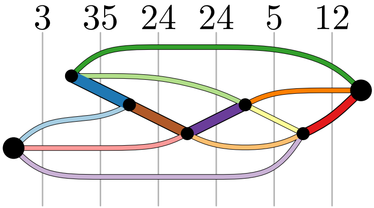
Two even subgraphs are $\mathbb{Z}_2$-homologous if their union forms a cut on the surface.



Homology height: NP-Hard

In fact, homology legnth is precisely the same as the cutwidth of the dual graph (once you adapt the monotonicity proof from graph searching [Bienstock-Seymour 1991]):

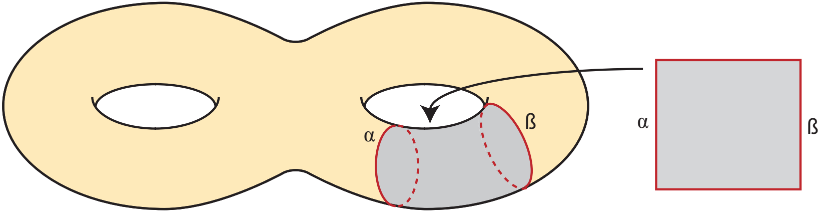




Here, you can line the vertices up even if they are not dual to adjacent faces: this corresponds to a new piece of the homology cycle appearing around the face, since all dual edges will be cut.

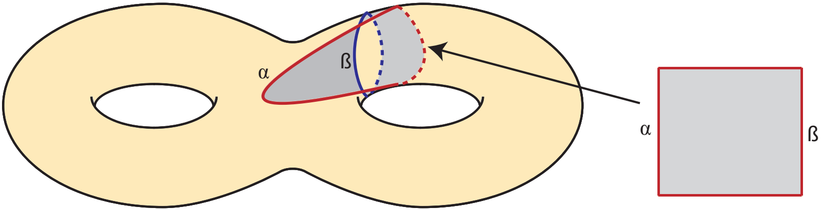
Area of a homotopy

Instead of focusing on the length or width, we can also examine the total area swept by a homotopy or homology.



Computing homotopy area

Surprisingly, this measure is much more tractable on surfaces than any other measure which takes topology into account, even for non-disjoint curves.



Definition

More formally, given a homotopy H, the area of H is defined as:

$$\text{Area}(H) = \int_{s\in[0,1]} \int_{t\in[0,1]} \left| \frac{dH}{ds} \times \frac{dH}{dt} \right| dsdt$$

Definition

More formally, given a homotopy H, the area of H is defined as:

$$\text{Area}(H) = \int_{s\in[0,1]} \int_{t\in[0,1]} \left| \frac{dH}{ds} \times \frac{dH}{dt} \right| dsdt$$

We are then interested in the smallest such value: $\inf_H \text{Area}(H)$.

Definition

More formally, given a homotopy H, the area of H is defined as:

$$\text{Area}(H) = \int_{s\in[0,1]} \int_{t\in[0,1]} \left| \frac{dH}{ds} \times \frac{dH}{dt} \right| dsdt$$

We are then interested in the smallest such value: $\inf_H \text{Area}(H)$.

Note that in generally, this is an improper integral, and the value for any H is not necessarily even finite.

Douglas and Rado's work

Douglas and Rado (1930's) were the first to consider this problem, as a variant of Plateau's problem (1847) dealing with soap bubbles and minimal surfaces.

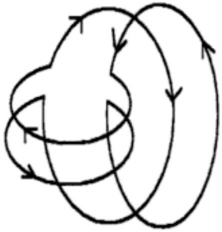


Fig. 4.1

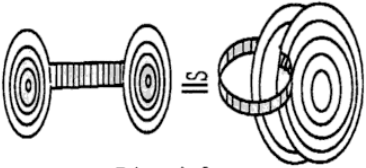


Fig. 4.2

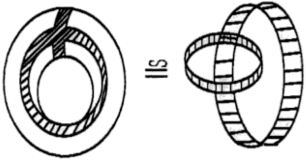
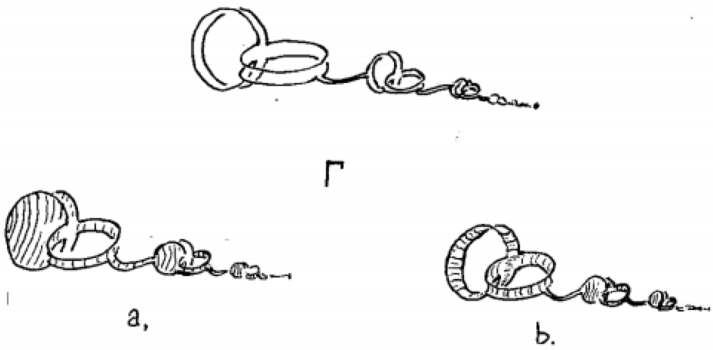


Fig. 4.3

[Minimal sub manifolds and related topics, Y. L. Xin]

Realizing the minimum area

There is an additional problem in that to find the infimum, we might have a pathological case where a sequence of good H's converge to something that is not even continuous.



[Lectures on Minimal Submanifolds, H. B. Lawson]

Douglas' theorem

They developed a restricted version using Dirichlet integrals (or energy integrals) which allow control over the parameterizations of the minimal surfaces. These integrals not only minimize area, but also ensure (almost) conformal parameterizations in the space.

Douglas' theorem

They developed a restricted version using Dirichlet integrals (or energy integrals) which allow control over the parameterizations of the minimal surfaces. These integrals not only minimize area, but also ensure (almost) conformal parameterizations in the space.

Theorem

Let γ be a finite Jordan curve in $\mathbb{R}^n$. Then there exists a continuous map $\Gamma : \{(x, y) \in \mathbb{R}^2 : x^2 + y^2 \leq 1\} \to \mathbb{R}^n$ such that:

1. Γ *maps the boundary of the disk monontically onto* γ.
2. Γ *is harmonic and almost conformal*
3. Γ *realizes the infimum of all areas*

Douglas' theorem

They developed a restricted version using Dirichlet integrals (or energy integrals) which allow control over the parameterizations of the minimal surfaces. These integrals not only minimize area, but also ensure (almost) conformal parameterizations in the space.

Theorem

Let γ be a finite Jordan curve in $\mathbb{R}^n$. Then there exists a continuous map $\Gamma : \left\{(x, y) \in \mathbb{R}^2 : x^2 + y^2 \leq 1\right\} \to \mathbb{R}^n$ such that:

1. *Γ maps the boundary of the disk monontically onto γ.*
2. *Γ is harmonic and almost conformal*
3. *Γ realizes the infimum of all areas*

(Well, I'm hiding a few details about the Dirichlet integrals here...)

Necessary assumptions

In [C-Wang 2013], we consider a much simpler setting - we are either in $\mathbb{R}^2$ or a piecewise linear surface. However, we do need some assumptions in order for the minimum area homotopy to exist.

- We must assume that H is continuous and piecewise differentiable (so it is differentiable everywhere except at a finite set of points and arcs).

Necessary assumptions

In [C-Wang 2013], we consider a much simpler setting - we are either in $\mathbb{R}^2$ or a piecewise linear surface. However, we do need some assumptions in order for the minimum area homotopy to exist.

- We must assume that H is continuous and piecewise differentiable (so it is differentiable everywhere except at a finite set of points and arcs).
- We must also assume the homotopy is monotone along the boundary of the domain and is regular on the interior (meaning intermediate curves are "kink-free").

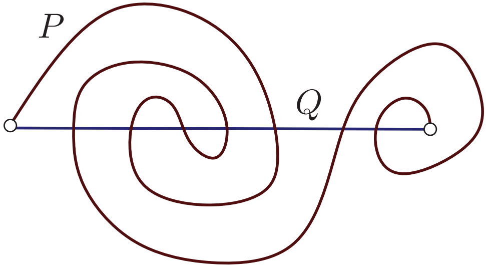
Necessary assumptions

In [C-Wang 2013], we consider a much simpler setting - we are either in $\mathbb{R}^2$ or a piecewise linear surface. However, we do need some assumptions in order for the minimum area homotopy to exist.

- We must assume that H is continuous and piecewise differentiable (so it is differentiable everywhere except at a finite set of points and arcs).
- We must also assume the homotopy is monotone along the boundary of the domain and is regular on the interior (meaning intermediate curves are "kink-free").
- Finally, we will assume our input curves (on M) are simple and have a finite number of piecewise analytic components. (In practice, they will simply be PL curves.)

Algorithm in the plane

In the plane, we consider the decomposition of the plane given by the union of the two curves.

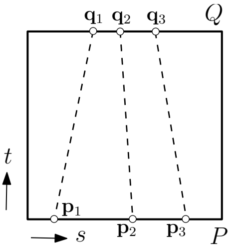


(I'm drawing continuous curves here for simplicity, but think of these as PL when we get to the running time.)

Anchor points

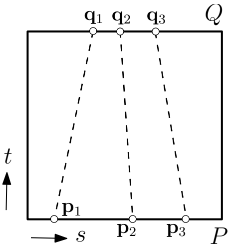
Note that any vertex of intersection could either be fixed throughout the homotopy (we call this an *anchor point*) or could be moved by the homotopy.

Note also that the ordering of the anchor points along the two curves P and Q will be the identical.



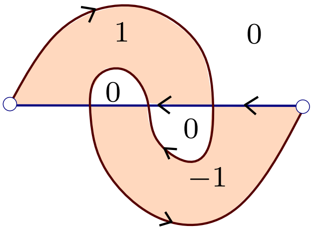
Note that any vertex of intersection could either be fixed throughout the homotopy (we call this an *anchor point*) or could be moved by the homotopy.

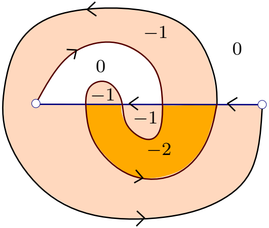
Note also that the ordering of the anchor points along the two curves P and Q will be the identical.



Also, in between anchor points, we prove that the homotopy will always move locally forward.

The *winding number* of a closed curve γ with respect to a point x, $wn(x;\gamma)$ is the number of times that curve travels counterclockwise around the point.

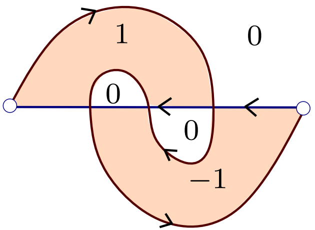


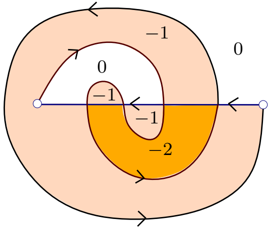


Using the winding number

Lemma

Any homotopy with no anchor points will have consistent winding numbers (all non-negative or all non-positive).



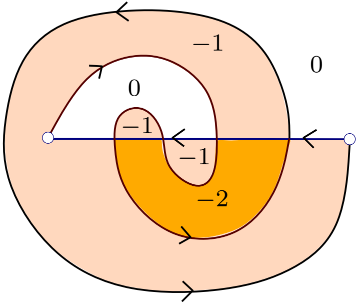


Calculating with no anchor points

Lemma

If $P \circ Q$ has consistent winding numbers, then:

$$\inf_{H} Area(H) = \int_{\mathbb{R}^2} |wn(x; P \circ Q)| dx$$



The algorithm: dynamic programming

Our algorithm now proceeds quite simply.

Our algorithm now proceeds quite simply.

- We can compute the winding number of each planar region. If all are non-positive or non-negative, then we simply sum the areas of each region with multiplicity given by the winding number.

Our algorithm now proceeds quite simply.

- We can compute the winding number of each planar region. If all are non-positive or non-negative, then we simply sum the areas of each region with multiplicity given by the winding number.
- If the numbers are not consistent, then we know there is at least one anchor point. Since the order of the anchor points along each curve is the same, we can enumerate all the possible sets of anchor points, and in between the anchor points compute the winding numbers again.

Running time in the plane

Let I be the number of intersections and n be the complexity of the input curves.

Running time in the plane

Let I be the number of intersections and n be the complexity of the input curves.

We give an algorithm that can be implemented in $O(I^2 n)$ time using dynamic programming, which simply builds up the sets of anchor points iteratively and uses previous solutions to speed up future computation.

Running time in the plane

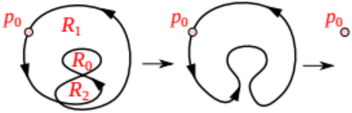
Let I be the number of intersections and n be the complexity of the input curves.

We give an algorithm that can be implemented in $O(I^2 n)$ time using dynamic programming, which simply builds up the sets of anchor points iteratively and uses previous solutions to speed up future computation.

However, this can be improved to $O(I^2 \log I)$ time with $O(I \log I + n)$ preprocessing if we are more careful about how we compute the winding numbers.

More recent algorithms for homotopy area

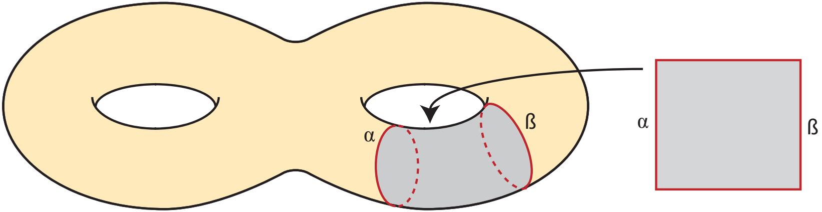
There has also been recent work to compute the best area homotopy when the input curve is not so "nice", but is an immersion of a disk into the plane.



- One result [Nie 2014] connects this problem to the weighted cancellation norm, which is a very combinatorial way to covert the best homotopy into a series of reduction moves on a word problem. The result is a polynomial time algorithm.
- Another [Fasy-Karakoc-Wenk 2016] consider a different approach which is more geometric, building up an exponential time algorithm, although perhaps faster dynamic programming techniques can speed this up.

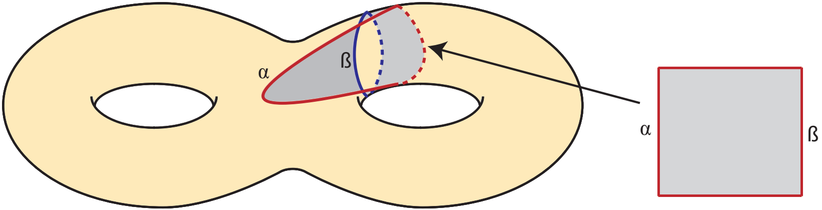
Homotopy area on a surface

Our paper [C.-Wang] also considers the algorithm for surfaces, which builds upon the planar algorithm.



Consider two homotopic curves on a triangulated surface M with positive genus.

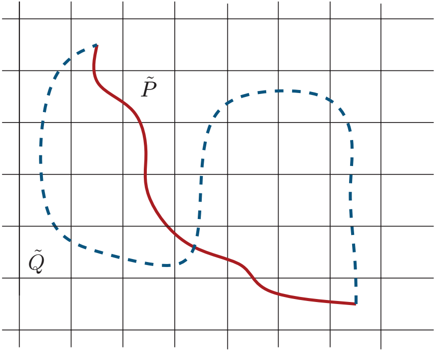
Our paper [C.-Wang] also considers the algorithm for surfaces, which builds upon the planar algorithm.



Consider two homotopic curves on a triangulated surface M with positive genus.

Lifting P and Q

If we fix a lift for the endpoints of P and Q in the universal cover $U(M)$, then $P \circ Q$ lifts to a unique closed curve in $U(M)$. Therefore, any homotopy between P and Q on M will correspond to a homotopy between their lifts in $U(M)$ with the same area.



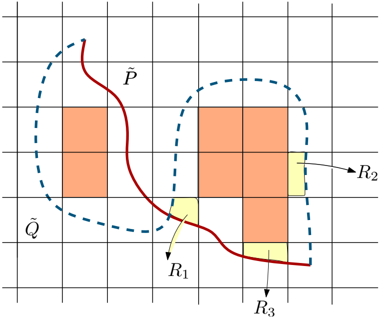
Algorithm for surfaces

We construct a portion of the universal cover which contains the lifts of P and Q as well as the regions inside their concatenation.

Algorithm for surfaces

We construct a portion of the universal cover which contains the lifts of P and Q as well as the regions inside their concatenation.

We then use our planar algorithm in $U(M)$, since similar results about the winding number will hold. Since we can simplify much of the interior of the regions in our representation, the total running time here is $O(gK \log K + I^2 \log I + In)$.



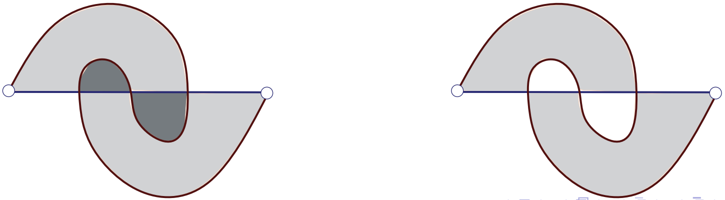
How to compute homology area

Formally (joint work with Mikael Vejdemo Johansson, and originally considered in slightly more restrcited settings in Dey, Hirani and Krishnamoorthy):

- Given cycles α and β, try to compute z such that $dz = \alpha - \beta$.
- Goal: compute z with a smallest area. Recall that d is a linear operator, and z and $\alpha - \beta$ are vectors.
- Optimization problem is then:
  arg min_z (area z), subject to $dz = \alpha - \beta$.

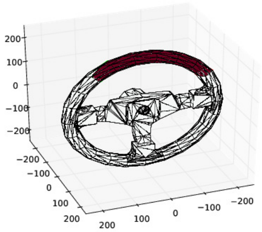
Formally (joint work with Mikael Vejdemo Johansson, and originally considered in slightly more restrcited settings in Dey, Hirani and Krishnamoorthy):

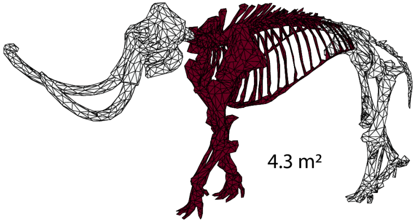
- Given cycles α and β, try to compute z such that $dz = \alpha - \beta$.
- Goal: compute z with a smallest area. Recall that d is a linear operator, and z and $\alpha - \beta$ are vectors.
- Optimization problem is then:
  arg min_z (area z), subject to $dz = \alpha - \beta$.
- Note again that this is NOT the same as homotopy area, at least for $d \leq 3$:



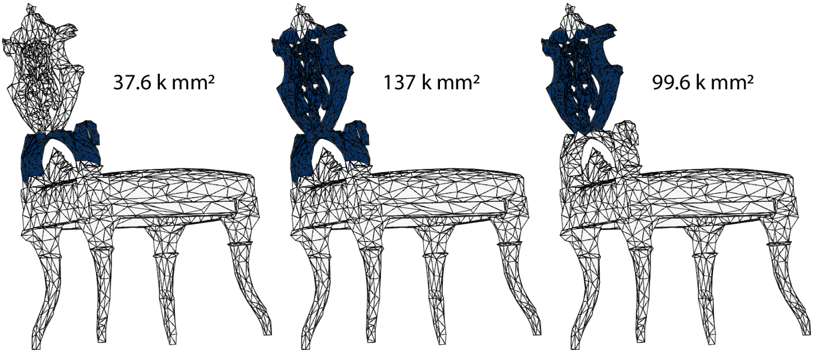
Final algorithm for homology area

In matrix multiply time, we can compute the best area homology on meshes:

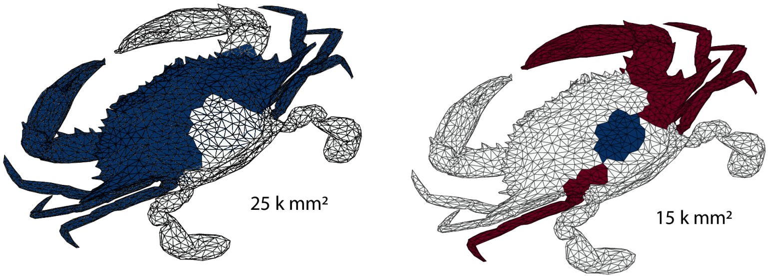




Chair model



Crab model



Open questions

- There is no algorithm to compute or approximate homotopic Fréchet distance on surfaces (or even polyhedra).
- Height of a homotopy algorithms (or hardness) are also open; all that is known is an O(log n) approximation and that it's in NP.
- These seem to be connected to all sorts of graph parameters, and even suggest some new variants to consider.
- It is unknown how to compute homotopy area between cycles on surfaces.
- Not clear if we can generalize any of these ideas (besides homology area) to surfaces instead of curves. Fréchet distance gets harder when you move to surfaces, but we don't know anything about topological variants.