BAB 1

PENDAHULUAN

1.1 Latar Belakang

KIRI (http://kiri.travel) adalah sebuah situs yang dikelola oleh PT. Kirana Sistem Transportasi, yang menyediakan layanan navigasi dari satu titik ke titik lain memanfaatkan transportasi publik. Layanan ini diberikan secara gratis kepada pengunjung situs / pengguna aplikasi bergerak mereka. Untuk mendukung layanan tersebut, tim KIRI melakukan kurasi rute angkot secara mandiri di Bandung dengan mencatat perjalanan setiap trayek menggunakan peralatan GPS (*Global Positioning System*). Pada perkembangannya, KIRI melakukan ekspansi ke beberapa kota lainnya termasuk Jakarta. Hanya saja, karena keterbatasan sumberdaya data di Jakarta dimasukkan berbekal data yang tersedia di internet, tanpa validasi di lapangan. Di sisi lain, ada beberapa pihak yang tertarik akan layanan KIRI di Bandung dan berniat untuk mendapatkan layanan serupa di kota mereka.

Situs https://angkot.web.id (selanjutnya disebut angkot.web.id saja) merupakan sebuah situs yang dikembangkan oleh Fajran Iman Rusadi yang berbasis di Belanda. Situs ini memungkinkan pengguna publik untuk melihat, memasukkan, atau memperbaiki data rute angkot di Indonesia (dengan kata lain, *crowdsourcing*). Layanan ini juga diberikan secara gratis.

Sampai saat ini, KIRI serta angkot.web.id merupakan dua buah situs yang terpisah dan bekerja secara independen.

1.2 Rumusan Masalah

Dari latar belakang yang sudah dijelaskan, peneliti bermaksud untuk mengintegrasikan data yang dimiliki kedua situs web tersebut. Integrasi tersebut dirumuskan ke dalam masalah-masalah berikut:

- Bagaimana mekanisme penarikan data oleh KIRI terhadap angkot.web.id secara otomatis?
- Bagaimana memisahkan data yang dimiliki oleh KIRI sendiri dengan data yang ditarik dari angkot.web.id?
- Bagaimana mengoptimasi protokol yang digunakan, sehingga kebutuhan *bandwidth* dapat dihemat?
- Bagaimana respon pengguna KIRI terhadap fitur yang diimplementasikan?

1.3 Tujuan

Berdasarkan rumusan masalah yang sudah dijabarkan, maka didefinisikan tujuan-tujuan berikut:

- Mengimplementasikan mekanisme penarikan data otomatis oleh KIRI terhadap angkot.web.id.

- Mengimplementasikan pemisahan data antara rute milik KIRI dengan data yang ditarik dari angkot.web.id.
- Mengoptimasi protokol yang digunakan, sehingga kebutuhan *bandwidth* dapat dihemat.
- Mempelajari respon pengguna KIRI terhadap fitur yang diimplementasikan.

1.4 Batasan Masalah

Penelitian ini memiliki batasan-batasan seperti berikut:

- Penelitian dilakukan untuk rute angkot kota Bandung saja.
- Integrasi otomatis akan dilakukan secara berkala (tidak *realtime*).
- Dengan alasan kerahasiaan, mesin navigasi KIRI dan angkot.web.id hanya dijelaskan pada bagian-bagian yang terkait dengan penelitian ini saja.

1.5 Metode Penelitian

Dalam penelitian ini, akan dilakukan langkah-langkah berikut:

- Melakukan studi terhadap mesin navigasi KIRI, protokol angkot.web.id, serta teori-teori lain yang mendukung kedua hal tersebut.
- Melakukan analisis untuk menemukan hal yang dapat dilakukan untuk mengintegrasikan data kedua situs tersebut.
- Melakukan perancangan untuk implementasi integrasi kedua sistem tersebut.
- Melakukan implementasi dari rancangan yang sudah dilakukan.
- Melakukan publikasi terhadap pengguna KIRI sehingga mereka dapat menguji hasil implementasi tersebut.
- Menarik kesimpulan atas hasil yang telah dilaksanakan.

1.6 Sistematika Penulisan

Berikut adalah sistematika penulisan dari dokumen ini:

- Bab 1 membahas latar belakang, rumusan masalah, tujuan penulisan, batasan-batasan, serta metode yang digunakan pada penelitian ini.
- Bab 2 membahas teori-teori yang digunakan dalam penelitian ini.
- Bab 3 membahas analisis yang dilakukan terhadap teori yang telah dijabarkan pada bab 2.
- Bab 4 membahas perancangan yang dilakukan sebelum mengimplementasikan integrasi yang dimaksud.
- Bab 5 membahas implementasi serta pengujian dari integrasi yang telah dilakukan.
- Bab 6 membahas kesimpulan dari keseluruhan penelitian ini, serta saran-saran yang dapat diberikan untuk penelitian berikutnya.

BAB 2

LANDASAN TEORI

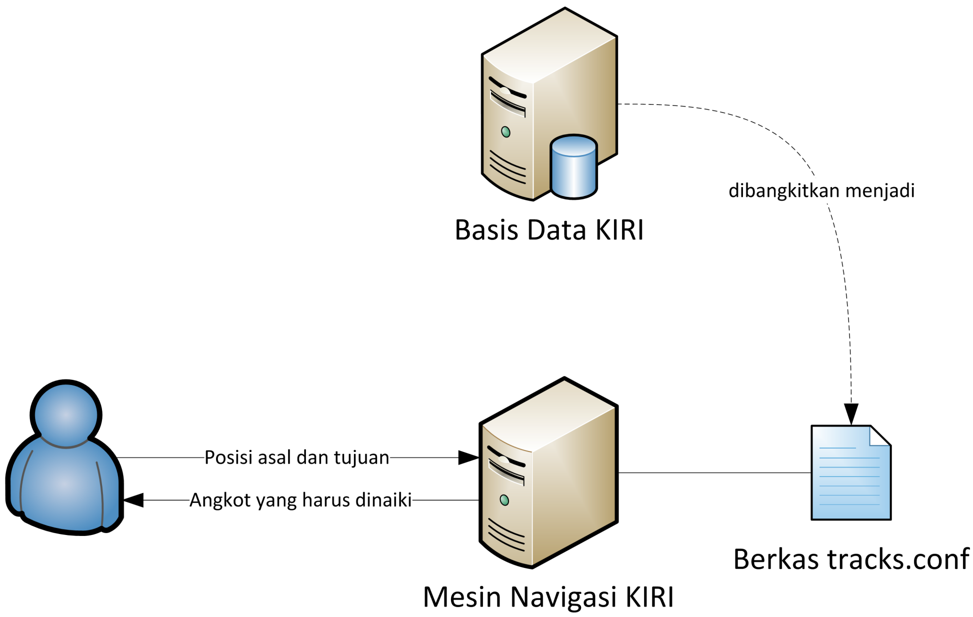
2.1 Mesin Navigasi KIRI

KIRI memiliki mesin navigasi yang dibangun pada bahasa Java. Mesin ini bertugas untuk menerima masukan berupa koordinat titik asal dan tujuan, kemudian menemukan angkot-angkot yang harus dinaiki untuk menuju titik tujuan dari titik asal. Karena alasan kerahasiaan, pembahasan mengenai mesin navigasi KIRI tidak mengacu pada referensi publik, melainkan dari survei terhadap kode sumber internal KIRI.

Seperti dapat dilihat pada gambar **??**, elemen arsitektur yang mendukung navigasi KIRI dibagi menjadi tiga, yaitu:

- **Basis Data KIRI** menyimpan informasi rute 33 trayek angkot, yang mencakup identifikasi trayek (`trackId` dan `trackTypeId`), nama trayek (`trackName`), daftar koordinat yang dilewati (`geodata`), informasi pulang-pergi (`pathloop`), catatan internal (`internalInfo`), prioritas untuk dipilih (`penalty`), informasi naik/turun penumpang (`transferNodes`), dan parameter ekstra untuk pembelian tiket (`extraParameters`).
- **Berkas tracks.conf** adalah hasil ekstraksi dari basis data kiri, yang menyimpan informasi penting saja yang dibutuhkan oleh algoritma mesin navigasi KIRI, yakni: (`trackId`, `trackTypeId`, `penalty`, `geodata`, `pathloop`, dan `transferNodes`.
- **Mesin Navigasi KIRI** adalah program yang bertugas mengolah data yang ada pada berkas tracks.conf, sehingga dapat menjawab pertanyaan navigasi dari titik asal ke titik tujuan. Karena alasan historis, mesin navigasi tidak membaca data langsung dari basis data, melainkan dari berkas tracks.conf.

2.2 Protokol angkot.web.id



Gambar 2.1: Arsitektur Saat Ini

BAB 3

INTRODUCTION

3.1 Motivation

A trajectory is the motion path of a moving object. Various moving objects such as animals (probably in a wildlife area), hurricanes, or customers in a shopping area have trajectories that can provide valuable information. For example, trajectory data can be used to predict the movement of the same type of object in similar situation in the future.

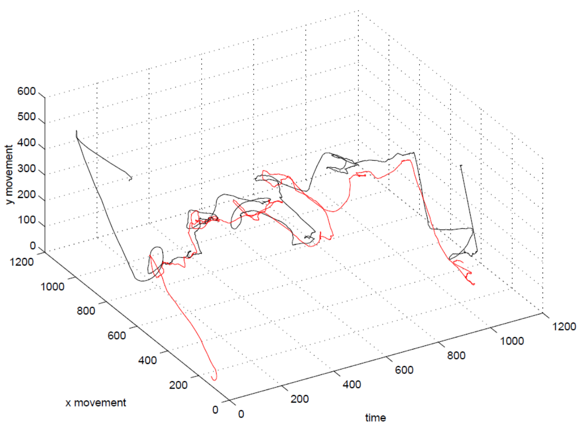
During a tracking procedure, the location of these moving objects can be obtained using various location detection devices (e.g: RFID, GPS devices and mobile phones). Later, this information will be sent to a database using any communication network (usually a wireless network). Because typical trajectory data is obtained during a specific interval, then trajectory data also has a temporal component, besides its spatial component.

The trajectory of a moving object is typically modeled as a sequence of consecutive locations in a multi-dimensional (generally two or three dimensional) Euclidean space [?]. Figure **??** shows an example of two trajectories from two objects which are moving in a 2D plane. With their temporal component, we can see that these trajectories are represented as polylines in a 3D space.

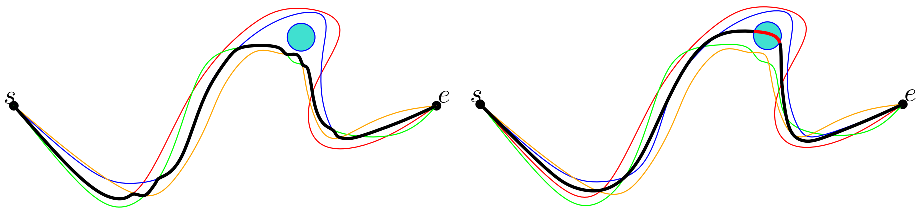
Nowadays, with the rapid development of technologies in mobile computing and wireless communication, many devices with location acquisition capabilities make it possible to obtain huge volumes of trajectory data from various moving objects. Furthermore, analysis of trajectory data is an important task for many applications that contain processing and managing moving objects, such as animal movements [?, ?, ?, ?], traffic and transport analysis [?], defense and surveillance areas [?], oceanographic observations[1], weather and natural phenomena [?], people behavior [?] and sports [?, ?].

Previous work on trajectory data analysis shows that there are several ways to analyze

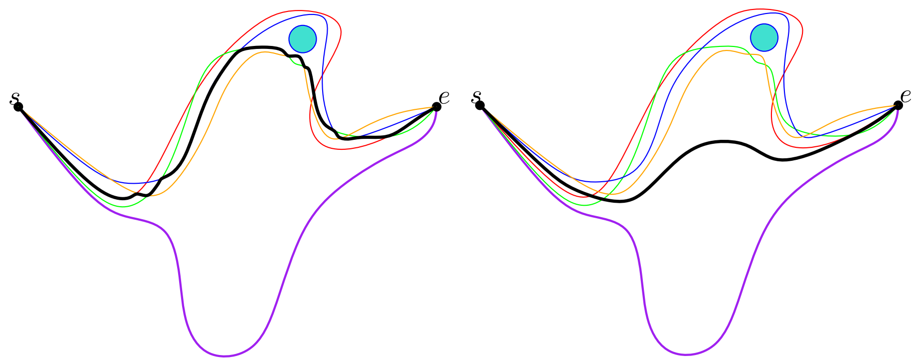
[1]W.S. Kessler, "Argo work in the coral sea."http://faculty.washington.edu/kessler/noumea/gliders/argo_coral_sea.html, March 2010.



Gambar 3.1: Example of 2D trajectories with time component, from [?]



Gambar 3.2: Example of the median (left) and the mean (right) trajectory [?]



Gambar 3.3: Robustness of the median trajectory

sets of trajectories. For example, similarity between trajectories can be determined [?, ?, ?]. Trajectories can also be clustered into groups with similar characteristics [?, ?, ?, ?]. Other examples are common data mining tasks such as classification [?, ?] and outlier detection [?]. Furthermore, interesting movement patterns such as flocking can also be computed from a set of trajectories [?, ?, ?, ?].

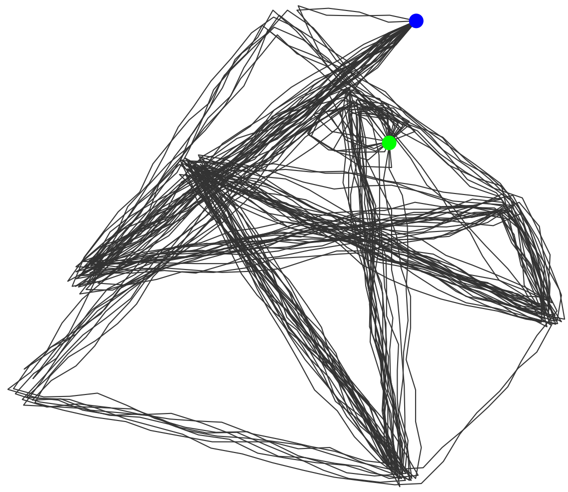
Even though analysis and research on trajectories has expanded in recent years, several basic concepts still need to be studied further. Some of them are the median and the mean trajectory for a collection/set of trajectories. The median and the mean trajectory share some common properties: they should be similar to other trajectories in the set and all parts of them should be located roughly in the middle of the set. However, there are several important differences between them: Firstly, the median trajectory must use only parts of trajectories in the set. It uses only parts of one trajectory or combines parts from many different trajectories. This property might be a disadvantage for the median because some parts of it can be located not in the middle of the set, but in several situations this might be useful.

Figure **??** shows a set of four trajectories which avoid the light-blue obstacle. The possible mean trajectory (the black trajectory in the right-hand side of the figure) will pass through the obstacle because the mean must lie in the middle of all trajectories in the set. In this case, it is clear that the mean trajectory is not suitable for a path of a moving object.

The median trajectory (black trajectory in the left-hand side of the figure) gives a more suitable path because it always uses parts of other trajectories. In parts near the obstacle, the median is not really in the middle of other trajectories.

Secondly, the median trajectory is more robust against outliers than the mean trajectory. Figure **??** shows this situation: we add one trajectory (with purple color) which can be categorized as an outlier compared to other trajectories. While the median trajectory only needs to be modified a little bit, the mean trajectory has to be changed a lot (comparing to the mean trajectory in Figure **??**), to keep it in the middle of other trajectories.

In this thesis, we will not cover the mean trajectory and only discuss the median trajectory and algorithms to compute it. We also ignore the temporal component of the trajectory because it is not clear yet how to take it into account when computing the median trajectory. However, some research on motion and kinetic data structures contains a temporal component and are related to the median/mean trajectory [?, ?].



Gambar 3.4: The set of 30 trajectories,starting at the blue point & ending at the green point

For other types of data, a median has a clear definition. The median from a population (or a sample) of integer numbers is the number that separates the population into two halves, where at most half of the population have a smaller value and the other half of the population have a larger value than the median.

For geometric data types, the concept of median also exist. A *center point* of a set P of n points in the plane is a point such that any closed half-plane whose bounding line contains the center point, contains at least $n/3$ points of P [**?**]. If we force the center point to be one of the points from P, then we obtain a 2-dimensional version of the median, although the "quality" of this median can be bad.

The median trajectory does not have any formal definition yet. Based on several properties that we mention earlier, such as its similarity with other trajectories and lying approximately in the middle of the set, we can determine a possible median, which can useful in several ways:

3.1.1 Determine the most typical trajectory

The property of the median trajectory makes it suitable to analyze the movement behavior or movement pattern from a group of same objects because the median somehow represents the whole trajectories in the set/collection. The median trajectory properties, such as the length, the direction or the average speed (if we include the temporal component), could give valuable information.

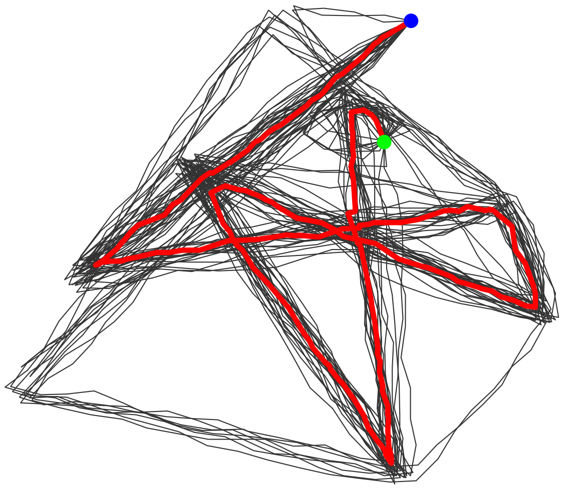
Example applications include the detection of outliers, which can be done by analyzing the length and the similarity of the shape of the median with other trajectories. Analyzing the average speed together with the shape of the median trajectory might be useful to understand the behavior and the movement pattern of people walking around in an area which has several interesting places to be visited (e.g., a zoo or an amusement park).

3.1.2 Better visualization for a set of trajectories

Visualization of the median trajectory, together with its set of trajectories, might give the viewer a better interpretation and information about the set of trajectories.

We give an example in Figure **??**, where a set has 30 trajectories, which is paths of 30 objects moving from the blue point to the green point. From this figure, we can hardly tell anything about the general behavior or the direction of these trajectories. However, we know that several trajectories are different than others and probably can predict what the majority does, but it is still difficult to visualize what the majority of these trajectories does.

In the following figure (Figure **??**), we present a possible median trajectory as the red and thick trajectory. From this visualization, it is clear what the majority of trajectories



Gambar 3.5: A set of 30 trajectories with its possible median trajectory

does. Moreover, we can identify what parts of some trajectories are completely different from others.

The visualization of the median trajectory could be useful in some real-life applications: The median trajectory from trajectories of visitors in a national park can be used to see the most common path taken by visitors, which is probably the path that is preferred by future visitors. This information might be useful if we want to create a map that can help those visitors by providing valuable information about a path and direction on that national park in the map, so that he/she can decide which path that he/she will take.

3.1.3 k-medoid clustering

Another application that could use the median trajectory is the k-medoid algorithm, which is used in cluster analysis. The k-medoid clustering algorithm is related to the k-means algorithm, a method to partition/group a set of objects into k different clusters containing similar objects.

In general, each cluster in both algorithms has one object act as a *central object* and other members of the cluster should be similar or having a small distance to this object. The similarity or the distance between objects can be measured using different distance functions (e.g. Euclidean distance, Minkowski distance, etc), depending on the type of objects and the purpose of the clustering.

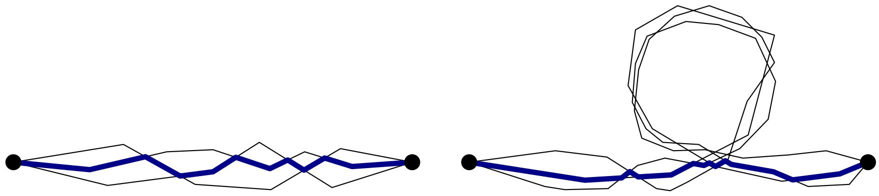
The main difference between the two algorithms is on the selection of the central object for each cluster. While the k-means simply uses the mean of objects, the k-medoid must use the medoid (an object which has the smallest average of dissimilarity/distance to all other objects in the set, but it must be a member of the set). This implies that k-means could create a new object to be the central object whereas the k-medoid must use one of the objects from the set. Thus, the k-medoid algorithm is more suitable for spatial clustering purposes and less sensitive against noise and outliers.

Partitioning Around Medoids (PAM) [?] is a basic k-medoid clustering algorithm. It works as follow:

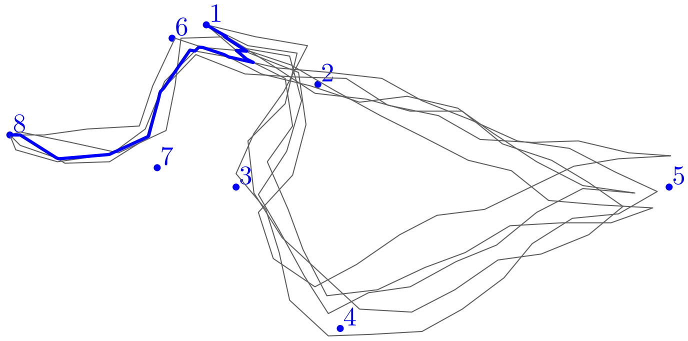
1. Define a value k and choose k objects as a set of medoids.

2. Assign every object to its closest/similar medoid and after that, compute the cost for the whole configuration.

3. Find another configuration by selecting a pair of medoid and non-medoid objects which have the smallest distance cost and swapping them temporarily. Then, we assign all other objects to this temporary set of medoids and obtain a new configuration.



Gambar 3.6: Illustration of the simple idea using switching



Gambar 3.7: The median trajectory makes a shortcut [?]

4. If the new configuration has smaller cost than the last configuration, then we change the set of medoids and return to step 3

5. Otherwise, stop and we find the set of medoids with their non-overlapping set of clusters.

In case we want to cluster a set of trajectories, we can use the median trajectory as a medoid in this algorithm. However, some changes probably should be made. For example, finding another configuration is not done by simply swapping the median with other trajectory, instead we can choose to swap part of them (with the requirement that both trajectories intersect one another).

3.2 Basic Idea of the Research

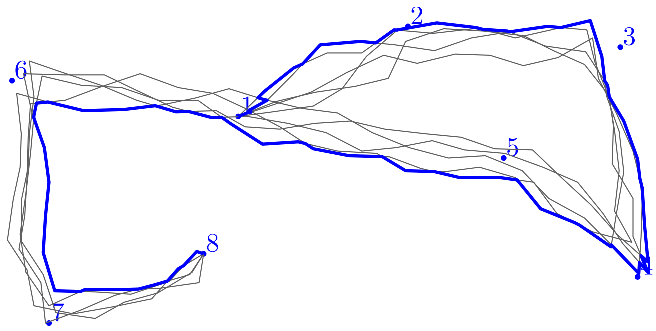
Consider a set T of m trajectories. We want to to compute the median trajectory of T. In this set, all trajectories have the same start and end points. The median trajectory of T must be built using parts of trajectories in T and somehow must follow what other trajectories in T do, while staying in the "middle" of other trajectories.

3.2.1 The Simple Switching Method

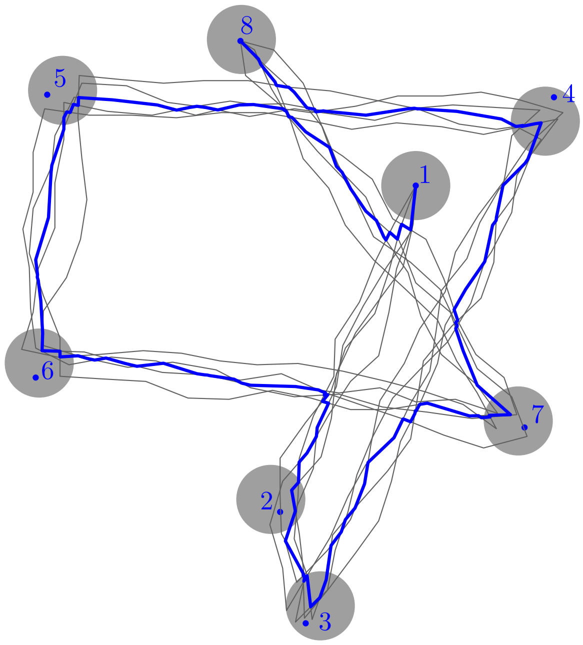
A simple idea to obtain a median trajectory from T is to start from the "middle" trajectory, which is the $(m+1)/2$-level of arrangement formed by all trajectories in T (we assume m is odd). At every intersection point, the median trajectory will switch to another trajectory and keep $(m+1)/2$ trajectories above and below the median [?].

Figure **??** shows the result (the median trajectory is the thick-blue trajectory) of this approach for two different types of set of trajectories, one of them contains trajectories with self intersection. From the right-hand side of the figure in Figure **??**, we can see that this method cannot produce suitable median trajectory because the median does not follow the loop created by the three trajectories.

In general, this method will not give a suitable median if a set of trajectories contains self-intersecting trajectories. More examples from [?] show several "incorrect" median trajectories obtained by using this simple switching method. The blue median trajectory in Figure **??** makes a shortcut path to the end point.



Gambar 3.8: The median trajectory does not stay in the middle [?]



Gambar 3.9: The median trajectory does not follow the correct sequence of regions [?]

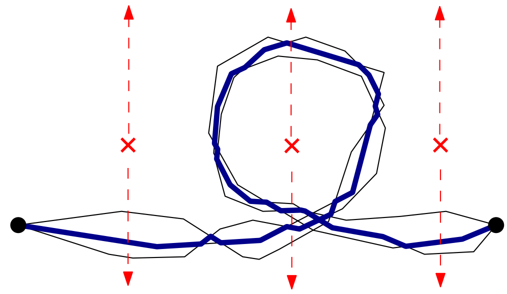
The median trajectory in Figure **??** does not stay in the "middle" of other trajectories. Finally, in Figure **??**, the median trajectory does not follow the sequence of regions as the other trajectories. The correct sequence of regions is $1-2-3-4-5-6-7-8$.

3.2.2 The Algorithm Using the Concept of Homotopy

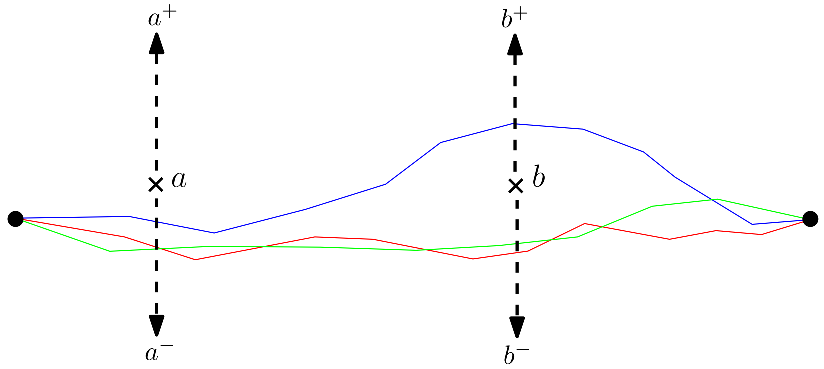
Another algorithm to compute the median trajectory uses the concept of homotopy (along with the modified simple switching method) [?]. This algorithm works by placing cross in a relatively large face bounded by segments from a set of trajectories. Figure **??** shows an example where cross is placed in the relatively large bounded face and two crosses are placed in the outer face.

Based on the location of these crosses, each trajectory in T will be assigned a *signature*. Figure **??** shows three trajectories and two crosses a and b. From these two crosses, four half-lines are created: a^+ and a^- are half-lines above and below a, while b^+ and b^- are half-lines above and below b, respectively.

For all trajectories in T, we give them a signature based on how they intersect with the half-line(s) from the crosses. Note that each trajectory might have a different signature, because it depends on the position of the trajectory with respect to all crosses in the plane. In Figure **??**, the blue trajectory intersect with a^- and b^+, thus its signature will be a^-b^+ (the order is following the direction of the trajectory). In the same way, the signature of the red trajectory will be the same as the green trajectory: a^-b^-.



Gambar 3.10: Illustration of the algorithm using homotopy concept



Gambar 3.11: Trajectories and crosses

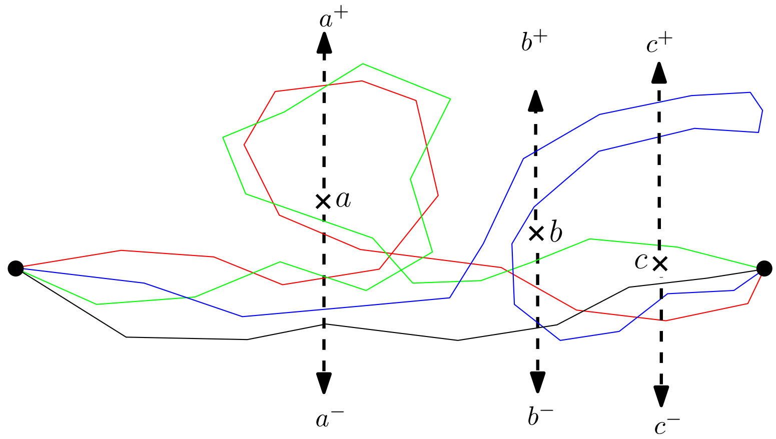
Two different trajectories are homotopic if one trajectory can be deformed continuously into the other one without passing through any crosses, while the start and end point are not moved. Naturally, two trajectories are homotopically equivalent if their signatures are exactly the same. However, two homotopic trajectories do not always have the same signatures. We shown an example in Figure **??** where the blue and the black trajectory are homotopic, but their signatures are different.

To determine whether two trajectories with different signature are homotopic or not, we perform a *reduce* operation. This *reduce* operation works by eliminating two exact same signs, if their position is next to each other in the signature. In Figure **??**, the signature of the blue trajectory is $a^-b^+c^+c^+b^+b^-c^-$. Notice that it has two c^+ that we can eliminate. This will change the signature of the blue trajectory into $a^-b^+b^+b^-c^-$. Once again, we can identify that two b^+ are positioned directly to each other. Performing the reduce operation again, we will get the final signature of the blue trajectory: $a^-b^-c^-$. At this point, we cannot apply the reduce operation again to this signature, and we say that the signature has been *maximally reduced*. Finally,we conclude that if two trajectories have the same maximally reduced signature, then the two trajectories are homotopically equivalent.

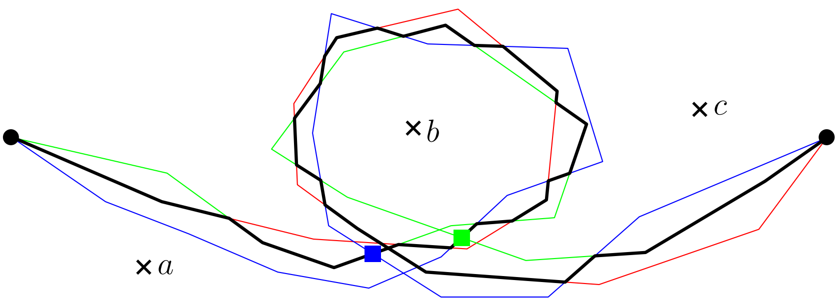
The next step of the algorithm is to create a subset T' of T, and then find the median trajectory by using only parts of trajectories from T'. Creating T' is straightforward, we only need to compute maximally reduced signature for all trajectories and choose a subset with the largest number of trajectories which have the same signature.

To create the median trajectory from T', we use the modified version of the switching method. This method start at the first segment of the“middle” trajectory. We find such a segment by determine the outer face of the set of trajectories. The first segment of the “middle” trajectory is the segment where there are $(n-1)/2$ first segments from other trajectories (assume n is odd) between the segment and the outer face (on each side of the segment).

Then, at every intersection, the median will switch to another trajectory if the continuation along this trajectory (without ever switching again) gives the same signature with the signature of one trajectory from T'. Figure **??** shows an example of this algorithm. After



Gambar 3.12: The blue and black trajectory are homotopically equivalent [?]



Gambar 3.13: Modified switching method [?]

starting with the red trajectory and switching to the green trajectory at the first intersection, the next intersection is with the blue trajectory (at the small blue square) and so far, the signature for the median is a^+. Although the blue trajectory is going forward, the signature after this intersection while following the blue trajectory (until the end point) is b^-c^-.

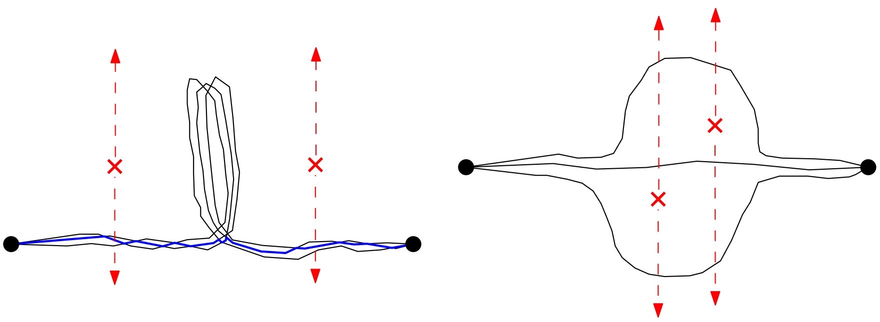
If the median switches at this intersection, the final signature will be $a^+b^-c^-$, which is not the signature of this set $(a^+b^-b^+b^-c^-)$. At this point, the median does not switch to another trajectory. Instead, it continues to move along the green trajectory. The same situation also occurs when the median (now following the blue trajectory) intersects with the green trajectory (at the small green square).

Although this algorithm can produce a more suitable median trajectory for the situation where the switching method fails, the quality of the homotopic median trajectory depends heavily on the following factors:

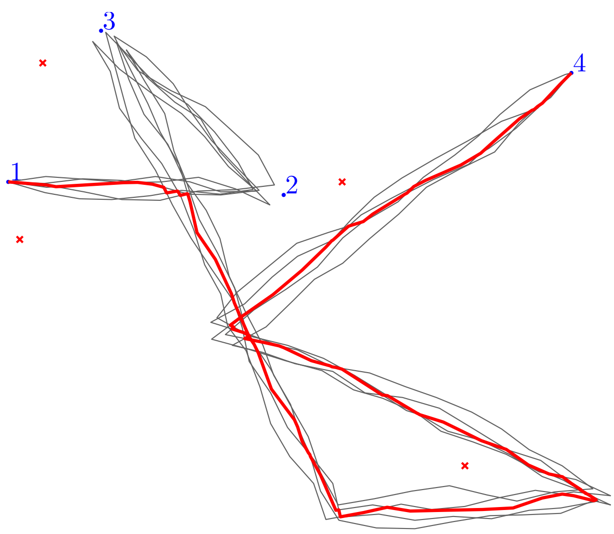
- placement of the crosses
- the number of trajectories which have the same signature

Therefore, in several cases the algorithm with the homotopy concept cannot produce suitable median trajectories. We give two examples in Figure **??**: in the left-hand side of figure, the final median trajectory (blue) does not follow other trajectories to the area with a narrow space. This problem arises because that narrow space is not large enough for a cross to be placed.

In the right-hand side of the figure, there are no two trajectories homotopically equivalent to each other. Nevertheless, by looking at their position, the median trajectory should be the one in the middle (between the two crosses). However, the algorithm with the homotopy concept does not guarantee that a suitable median trajectory will be found because there is no subset that contains the majority of trajectories in T. Figure **??** shows an example from [?], where the median trajectory does not completely follow what other trajectories do.



Gambar 3.14: Example cases where algorithm using homotopy will fail



Gambar 3.15: The median trajectory does not pass through part of trajectories in the upper left area

3.2.3 The Proposed Solutions

To solve the problems we mention in the previous section, we propose two algorithms to compute the median trajectory from a set of m trajectories (where each trajectory has n segments).

The first algorithm is an $O(1.2108^m + m^5n^5)$ worst-case time algorithm. This algorithm uses the Fréchet Distance [?] and works similar to the algorithm using the homotopy concept because both of them have to create the largest subset of similar trajectories and then compute the median trajectory by using parts of trajectories in that subset. By using the Fréchet Distance, we avoid the requirement to find proper places to put crosses, but still can produce suitable median trajectory in the situation where the homotopic algorithm fails (e.g. the example with a narrow space).

The second algorithm uses the combination of the buffer concept and Dijkstra's Shortest Path algorithm. Unlike all previous algorithms, this algorithm does not need to find the largest subset consisting similar trajectories. Using this algorithm, we can compute the median trajectory in $O(h^2 \log h)$ time in the worst-case, where h is the number of all segments in T ($h = O(mn)$).

We implemented the second algorithm in Java programming language and experiments have been done to determine the quality of the resulting median trajectory produced by this algorithm. To provide the test data (set of trajectories), we use a trajectories generator instead of using real-world data. This allow us to test much larger sets of trajectories

3.3 Outline of the Thesis

Chapter 2 describes the properties for a set of trajectories and also some properties the median trajectory should have.

The next two chapters explain in detail the two algorithms:

- Chapter 3 starts with a brief introduction of the Fréchet Distance and after that, we will explain how to use it to compute the median trajectory.

- Chapter 4 introduces the method to compute the median trajectory using the combination of the buffer concept and Dijkstra's shortest path algorithm.

We will give an explanation about our implementation, particularly on the implementation of the trajectories generator, in Chapter 5. In Chapter 6, we present the measures used to evaluate the quality of the median trajectory, the experiments set-up and the results from the experiments. This thesis will be concluded in Chapter 7 and 8, where we draw conclusions and discuss some issues and possible directions for further research.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus

mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Donec odio elit, dictum in, hendrerit sit amet, egestas sed, leo. Praesent feugiat sapien aliquet odio. Integer vitae justo. Aliquam vestibulum fringilla lorem. Sed neque lectus, consectetuer at, consectetuer sed, eleifend ac, lectus. Nulla facilisi. Pellentesque eget lectus. Proin eu metus. Sed porttitor. In hac habitasse platea dictumst. Suspendisse eu lectus. Ut mi mi, lacinia sit amet, placerat et, mollis vitae, dui. Sed ante tellus, tristique ut, iaculis eu, malesuada ac, dui. Mauris nibh leo, facilisis non, adipiscing quis, ultrices a, dui.

Morbi luctus, wisi viverra faucibus pretium, nibh est placerat odio, nec commodo wisi enim eget quam. Quisque libero justo, consectetuer a, feugiat vitae, porttitor eu, libero. Suspendisse sed mauris vitae elit sollicitudin malesuada. Maecenas ultricies eros sit amet ante. Ut venenatis velit. Maecenas sed mi eget dui varius euismod. Phasellus aliquet volutpat odio. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Pellentesque sit amet pede ac sem eleifend consectetuer. Nullam elementum, urna

vel imperdiet sodales, elit ipsum pharetra ligula, ac pretium ante justo a nulla. Curabitur tristique arcu eu metus. Vestibulum lectus. Proin mauris. Proin eu nunc eu urna hendrerit faucibus. Aliquam auctor, pede consequat laoreet varius, eros tellus scelerisque quam, pellentesque hendrerit ipsum dolor sed augue. Nulla nec lacus.

Suspendisse vitae elit. Aliquam arcu neque, ornare in, ullamcorper quis, commodo eu, libero. Fusce sagittis erat at erat tristique mollis. Maecenas sapien libero, molestie et, lobortis in, sodales eget, dui. Morbi ultrices rutrum lorem. Nam elementum ullamcorper leo. Morbi dui. Aliquam sagittis. Nunc placerat. Pellentesque tristique sodales est. Maecenas imperdiet lacinia velit. Cras non urna. Morbi eros pede, suscipit ac, varius vel, egestas non, eros. Praesent malesuada, diam id pretium elementum, eros sem dictum tortor, vel consectetuer odio sem sed wisi.

Sed feugiat. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Ut pellentesque augue sed urna. Vestibulum diam eros, fringilla et, consectetuer eu, nonummy id, sapien. Nullam at lectus. In sagittis ultrices mauris. Curabitur malesuada erat sit amet massa. Fusce blandit. Aliquam erat volutpat. Aliquam euismod. Aenean vel lectus. Nunc imperdiet justo nec dolor.

Etiam euismod. Fusce facilisis lacinia dui. Suspendisse potenti. In mi erat, cursus id, nonummy sed, ullamcorper eget, sapien. Praesent pretium, magna in eleifend egestas, pede pede pretium lorem, quis consectetuer tortor sapien facilisis magna. Mauris quis magna varius nulla scelerisque imperdiet. Aliquam non quam. Aliquam porttitor quam a lacus. Praesent vel arcu ut tortor cursus volutpat. In vitae pede quis diam bibendum placerat. Fusce elementum convallis neque. Sed dolor orci, scelerisque ac, dapibus nec, ultricies ut, mi. Duis nec dui quis leo sagittis commodo.

Aliquam lectus. Vivamus leo. Quisque ornare tellus ullamcorper nulla. Mauris porttitor pharetra tortor. Sed fringilla justo sed mauris. Mauris tellus. Sed non leo. Nullam elementum, magna in cursus sodales, augue est scelerisque sapien, venenatis congue nulla arcu et pede. Ut suscipit enim vel sapien. Donec congue. Maecenas urna mi, suscipit in, placerat ut, vestibulum ut, massa. Fusce ultrices nulla et nisl.

Etiam ac leo a risus tristique nonummy. Donec dignissim tincidunt nulla. Vestibulum rhoncus molestie odio. Sed lobortis, justo et pretium lobortis, mauris turpis condimentum augue, nec ultricies nibh arcu pretium enim. Nunc purus neque, placerat id, imperdiet sed, pellentesque nec, nisl. Vestibulum imperdiet neque non sem accumsan laoreet. In hac habitasse platea dictumst. Etiam condimentum facilisis libero. Suspendisse in elit quis nisl aliquam dapibus. Pellentesque auctor sapien. Sed egestas sapien nec lectus. Pellentesque vel dui vel neque bibendum viverra. Aliquam porttitor nisl nec pede. Proin mattis libero vel turpis. Donec rutrum mauris et libero. Proin euismod porta felis. Nam lobortis, metus quis elementum commodo, nunc lectus elementum mauris, eget vulputate ligula tellus eu neque. Vivamus eu dolor.

BAB 4

INTRODUCTION

4.1 Motivation

A trajectory is the motion path of a moving object. Various moving objects such as animals (probably in a wildlife area), hurricanes, or customers in a shopping area have trajectories that can provide valuable information. For example, trajectory data can be used to predict the movement of the same type of object in similar situation in the future.

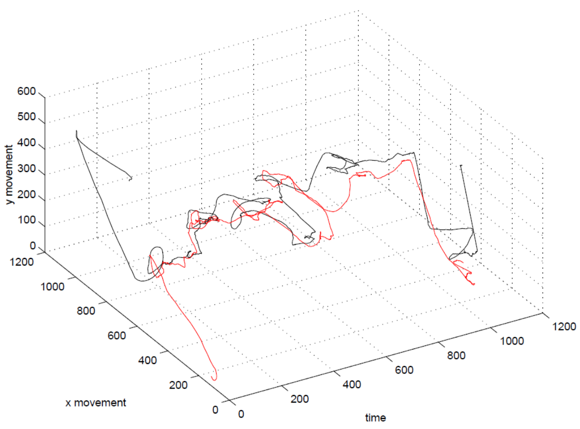
During a tracking procedure, the location of these moving objects can be obtained using various location detection devices (e.g: RFID, GPS devices and mobile phones). Later, this information will be sent to a database using any communication network (usually a wireless network). Because typical trajectory data is obtained during a specific interval, then trajectory data also has a temporal component, besides its spatial component.

The trajectory of a moving object is typically modeled as a sequence of consecutive locations in a multi-dimensional (generally two or three dimensional) Euclidean space [?]. Figure **??** shows an example of two trajectories from two objects which are moving in a 2D plane. With their temporal component, we can see that these trajectories are represented as polylines in a 3D space.

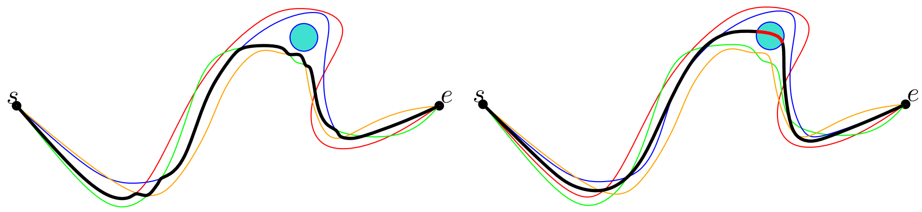
Nowadays, with the rapid development of technologies in mobile computing and wireless communication, many devices with location acquisition capabilities make it possible to obtain huge volumes of trajectory data from various moving objects. Furthermore, analysis of trajectory data is an important task for many applications that contain processing and managing moving objects, such as animal movements [?, ?, ?, ?], traffic and transport analysis [?], defense and surveillance areas [?], oceanographic observations[1], weather and natural phenomena [?], people behavior [?] and sports [?, ?].

Previous work on trajectory data analysis shows that there are several ways to analyze

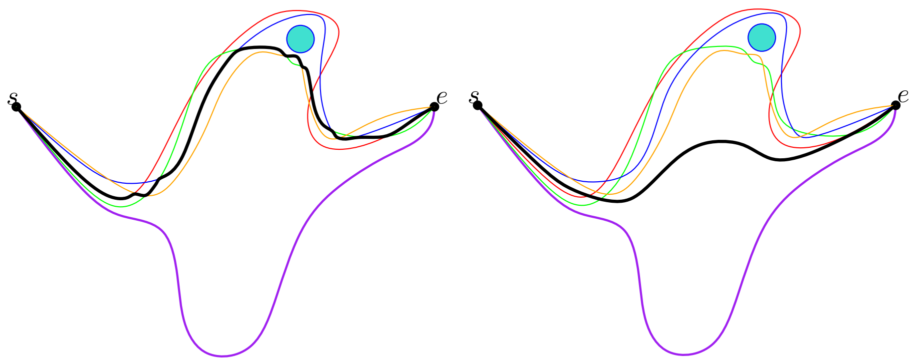
[1]W.S. Kessler, "Argo work in the coral sea."http://faculty.washington.edu/kessler/noumea/gliders/argo_coral_sea.html, March 2010.



Gambar 4.1: Example of 2D trajectories with time component, from [?]



Gambar 4.2: Example of the median (left) and the mean (right) trajectory [?]



Gambar 4.3: Robustness of the median trajectory

sets of trajectories. For example, similarity between trajectories can be determined [?, ?, ?]. Trajectories can also be clustered into groups with similar characteristics [?, ?, ?, ?]. Other examples are common data mining tasks such as classification [?, ?] and outlier detection [?]. Furthermore, interesting movement patterns such as flocking can also be computed from a set of trajectories [?, ?, ?, ?].

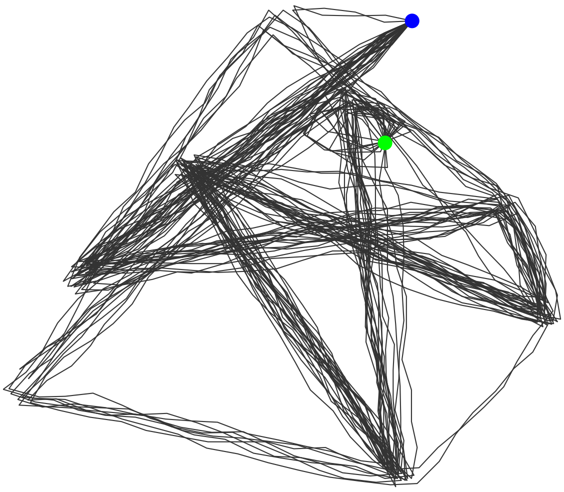
Even though analysis and research on trajectories has expanded in recent years, several basic concepts still need to be studied further. Some of them are the median and the mean trajectory for a collection/set of trajectories. The median and the mean trajectory share some common properties: they should be similar to other trajectories in the set and all parts of them should be located roughly in the middle of the set. However, there are several important differences between them: Firstly, the median trajectory must use only parts of trajectories in the set. It uses only parts of one trajectory or combines parts from many different trajectories. This property might be a disadvantage for the median because some parts of it can be located not in the middle of the set, but in several situations this might be useful.

Figure **??** shows a set of four trajectories which avoid the light-blue obstacle. The possible mean trajectory (the black trajectory in the right-hand side of the figure) will pass through the obstacle because the mean must lie in the middle of all trajectories in the set. In this case, it is clear that the mean trajectory is not suitable for a path of a moving object.

The median trajectory (black trajectory in the left-hand side of the figure) gives a more suitable path because it always uses parts of other trajectories. In parts near the obstacle, the median is not really in the middle of other trajectories.

Secondly, the median trajectory is more robust against outliers than the mean trajectory. Figure **??** shows this situation: we add one trajectory (with purple color) which can be categorized as an outlier compared to other trajectories. While the median trajectory only needs to be modified a little bit, the mean trajectory has to be changed a lot (comparing to the mean trajectory in Figure **??**), to keep it in the middle of other trajectories.

In this thesis, we will not cover the mean trajectory and only discuss the median trajectory and algorithms to compute it. We also ignore the temporal component of the trajectory because it is not clear yet how to take it into account when computing the median trajectory. However, some research on motion and kinetic data structures contains a temporal component and are related to the median/mean trajectory [?, ?].



Gambar 4.4: The set of 30 trajectories,starting at the blue point & ending at the green point

For other types of data, a median has a clear definition. The median from a population (or a sample) of integer numbers is the number that separates the population into two halves, where at most half of the population have a smaller value and the other half of the population have a larger value than the median.

For geometric data types, the concept of median also exist. A *center point* of a set P of n points in the plane is a point such that any closed half-plane whose bounding line contains the center point, contains at least $n/3$ points of P [**?**]. If we force the center point to be one of the points from P, then we obtain a 2-dimensional version of the median, although the "quality" of this median can be bad.

The median trajectory does not have any formal definition yet. Based on several properties that we mention earlier, such as its similarity with other trajectories and lying approximately in the middle of the set, we can determine a possible median, which can useful in several ways:

4.1.1 Determine the most typical trajectory

The property of the median trajectory makes it suitable to analyze the movement behavior or movement pattern from a group of same objects because the median somehow represents the whole trajectories in the set/collection. The median trajectory properties, such as the length, the direction or the average speed (if we include the temporal component), could give valuable information.

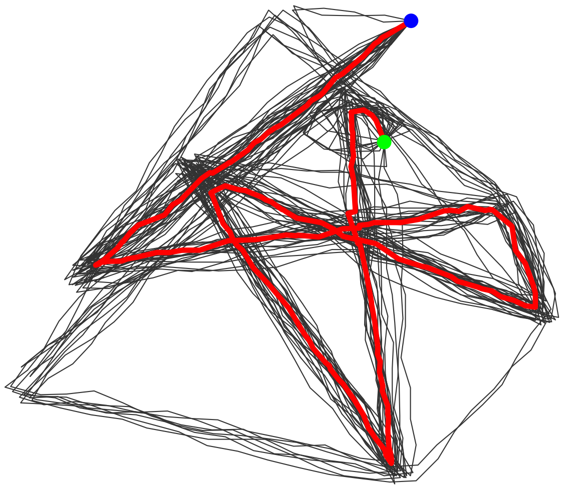
Example applications include the detection of outliers, which can be done by analyzing the length and the similarity of the shape of the median with other trajectories. Analyzing the average speed together with the shape of the median trajectory might be useful to understand the behavior and the movement pattern of people walking around in an area which has several interesting places to be visited (e.g., a zoo or an amusement park).

4.1.2 Better visualization for a set of trajectories

Visualization of the median trajectory, together with its set of trajectories, might give the viewer a better interpretation and information about the set of trajectories.

We give an example in Figure **??**, where a set has 30 trajectories, which is paths of 30 objects moving from the blue point to the green point. From this figure, we can hardly tell anything about the general behavior or the direction of these trajectories. However, we know that several trajectories are different than others and probably can predict what the majority does, but it is still difficult to visualize what the majority of these trajectories does.

In the following figure (Figure **??**), we present a possible median trajectory as the red and thick trajectory. From this visualization, it is clear what the majority of trajectories



Gambar 4.5: A set of 30 trajectories with its possible median trajectory

does. Moreover, we can identify what parts of some trajectories are completely different from others.

The visualization of the median trajectory could be useful in some real-life applications: The median trajectory from trajectories of visitors in a national park can be used to see the most common path taken by visitors, which is probably the path that is preferred by future visitors. This information might be useful if we want to create a map that can help those visitors by providing valuable information about a path and direction on that national park in the map, so that he/she can decide which path that he/she will take.

4.1.3 k-medoid clustering

Another application that could use the median trajectory is the k-medoid algorithm, which is used in cluster analysis. The k-medoid clustering algorithm is related to the k-means algorithm, a method to partition/group a set of objects into k different clusters containing similar objects.

In general, each cluster in both algorithms has one object act as a *central object* and other members of the cluster should be similar or having a small distance to this object. The similarity or the distance between objects can be measured using different distance functions (e.g. Euclidean distance, Minkowski distance, etc), depending on the type of objects and the purpose of the clustering.

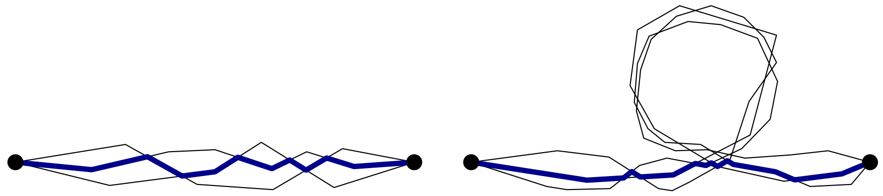
The main difference between the two algorithms is on the selection of the central object for each cluster. While the k-means simply uses the mean of objects, the k-medoid must use the medoid (an object which has the smallest average of dissimilarity/distance to all other objects in the set, but it must be a member of the set). This implies that k-means could create a new object to be the central object whereas the k-medoid must use one of the objects from the set. Thus, the k-medoid algorithm is more suitable for spatial clustering purposes and less sensitive against noise and outliers.

Partitioning Around Medoids (PAM) [?] is a basic k-medoid clustering algorithm. It works as follow:

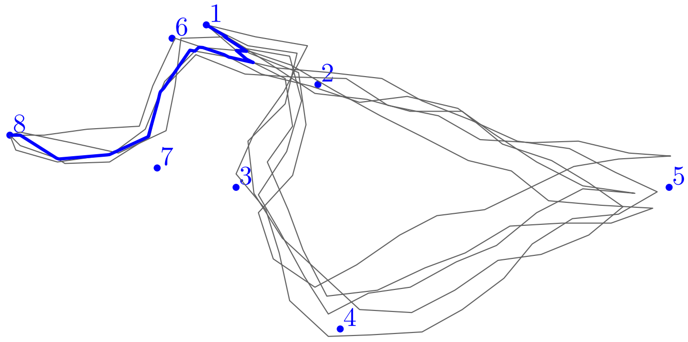
1. Define a value k and choose k objects as a set of medoids.

2. Assign every object to its closest/similar medoid and after that, compute the cost for the whole configuration.

3. Find another configuration by selecting a pair of medoid and non-medoid objects which have the smallest distance cost and swapping them temporarily. Then, we assign all other objects to this temporary set of medoids and obtain a new configuration.



Gambar 4.6: Illustration of the simple idea using switching



Gambar 4.7: The median trajectory makes a shortcut [?]

4. If the new configuration has smaller cost than the last configuration, then we change the set of medoids and return to step 3
5. Otherwise, stop and we find the set of medoids with their non-overlapping set of clusters.

In case we want to cluster a set of trajectories, we can use the median trajectory as a medoid in this algorithm. However, some changes probably should be made. For example, finding another configuration is not done by simply swapping the median with other trajectory, instead we can choose to swap part of them (with the requirement that both trajectories intersect one another).

4.2 Basic Idea of the Research

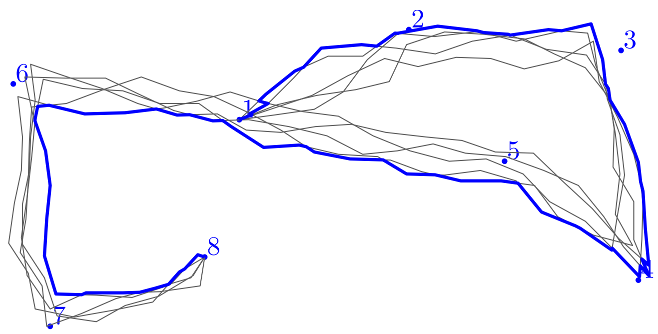
Consider a set T of m trajectories. We want to to compute the median trajectory of T. In this set, all trajectories have the same start and end points. The median trajectory of T must be built using parts of trajectories in T and somehow must follow what other trajectories in T do, while staying in the "middle" of other trajectories.

4.2.1 The Simple Switching Method

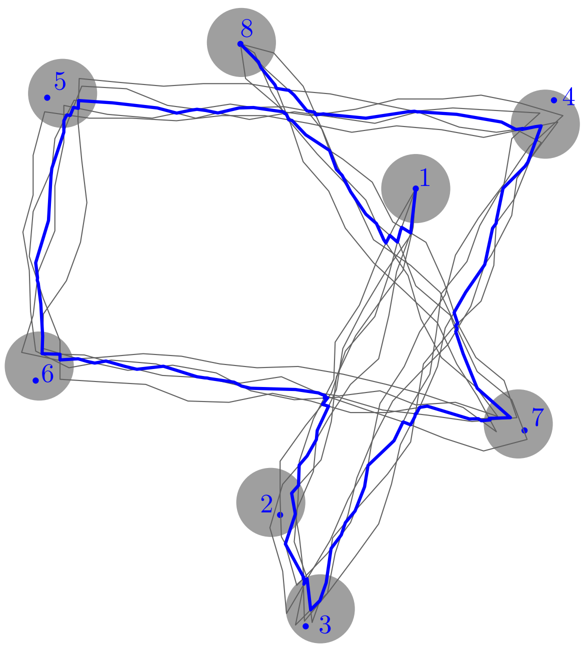
A simple idea to obtain a median trajectory from T is to start from the "middle" trajectory, which is the $(m+1)/2$-level of arrangement formed by all trajectories in T (we assume m is odd). At every intersection point, the median trajectory will switch to another trajectory and keep $(m+1)/2$ trajectories above and below the median [?].

Figure **??** shows the result (the median trajectory is the thick-blue trajectory) of this approach for two different types of set of trajectories, one of them contains trajectories with self intersection. From the right-hand side of the figure in Figure **??**, we can see that this method cannot produce suitable median trajectory because the median does not follow the loop created by the three trajectories.

In general, this method will not give a suitable median if a set of trajectories contains self-intersecting trajectories. More examples from [?] show several "incorrect" median trajectories obtained by using this simple switching method. The blue median trajectory in Figure **??** makes a shortcut path to the end point.



Gambar 4.8: The median trajectory does not stay in the middle [?]



Gambar 4.9: The median trajectory does not follow the correct sequence of regions [?]

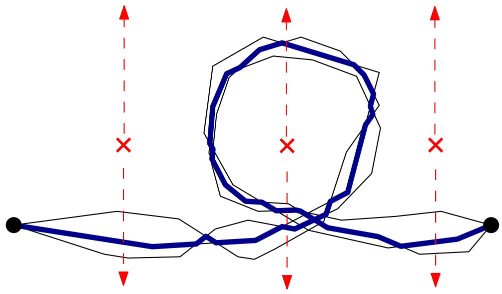
The median trajectory in Figure **??** does not stay in the "middle" of other trajectories. Finally, in Figure **??**, the median trajectory does not follow the sequence of regions as the other trajectories. The correct sequence of regions is $1-2-3-4-5-6-7-8$.

4.2.2 The Algorithm Using the Concept of Homotopy

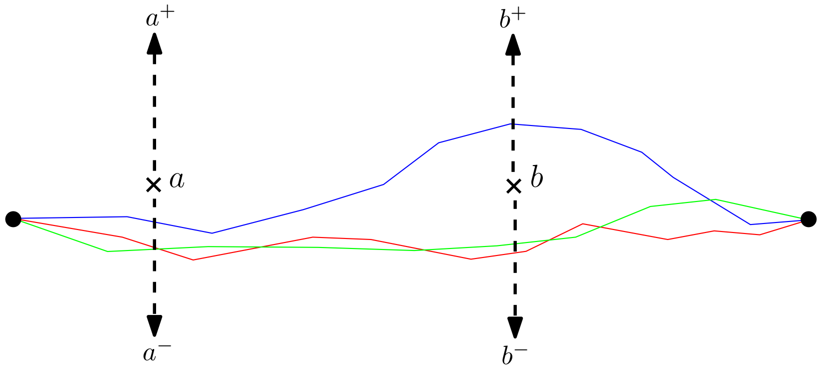
Another algorithm to compute the median trajectory uses the concept of homotopy (along with the modified simple switching method) [?]. This algorithm works by placing cross in a relatively large face bounded by segments from a set of trajectories. Figure **??** shows an example where cross is placed in the relatively large bounded face and two crosses are placed in the outer face.

Based on the location of these crosses, each trajectory in T will be assigned a *signature*. Figure **??** shows three trajectories and two crosses a and b. From these two crosses, four half-lines are created: a^+ and a^- are half-lines above and below a, while b^+ and b^- are half-lines above and below b, respectively.

For all trajectories in T, we give them a signature based on how they intersect with the half-line(s) from the crosses. Note that each trajectory might have a different signature, because it depends on the position of the trajectory with respect to all crosses in the plane. In Figure **??**, the blue trajectory intersect with a^- and b^+, thus its signature will be a^-b^+ (the order is following the direction of the trajectory). In the same way, the signature of the red trajectory will be the same as the green trajectory: a^-b^-.



Gambar 4.10: Illustration of the algorithm using homotopy concept



Gambar 4.11: Trajectories and crosses

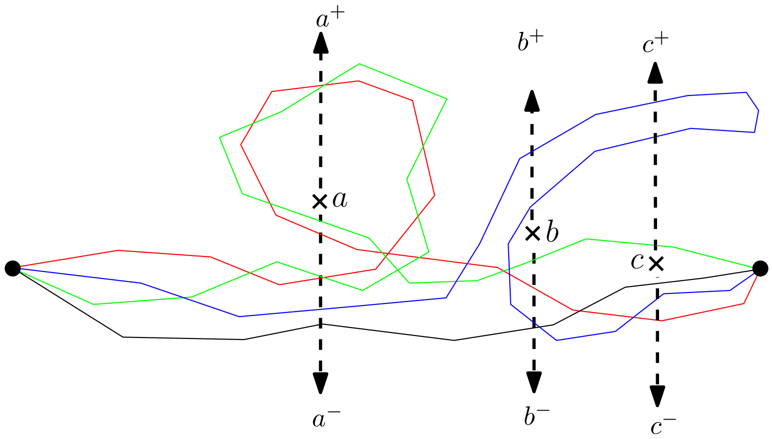
Two different trajectories are homotopic if one trajectory can be deformed continuously into the other one without passing through any crosses, while the start and end point are not moved. Naturally, two trajectories are homotopically equivalent if their signatures are exactly the same. However, two homotopic trajectories do not always have the same signatures. We shown an example in Figure **??** where the blue and the black trajectory are homotopic, but their signatures are different.

To determine whether two trajectories with different signature are homotopic or not, we perform a *reduce* operation. This *reduce* operation works by eliminating two exact same signs, if their position is next to each other in the signature. In Figure **??**, the signature of the blue trajectory is $a^-b^+c^+c^+b^+b^-c^-$. Notice that it has two c^+ that we can eliminate. This will change the signature of the blue trajectory into $a^-b^+b^+b^-c^-$. Once again, we can identify that two b^+ are positioned directly to each other. Performing the reduce operation again, we will get the final signature of the blue trajectory: $a^-b^-c^-$. At this point, we cannot apply the reduce operation again to this signature, and we say that the signature has been *maximally reduced*. Finally,we conclude that if two trajectories have the same maximally reduced signature, then the two trajectories are homotopically equivalent.

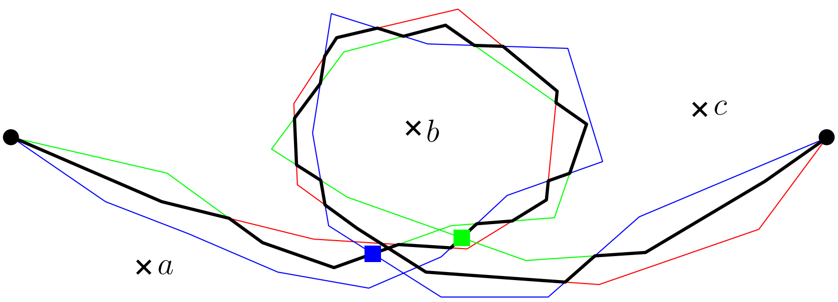
The next step of the algorithm is to create a subset T' of T, and then find the median trajectory by using only parts of trajectories from T'. Creating T' is straightforward, we only need to compute maximally reduced signature for all trajectories and choose a subset with the largest number of trajectories which have the same signature.

To create the median trajectory from T', we use the modified version of the switching method. This method start at the first segment of the“middle” trajectory. We find such a segment by determine the outer face of the set of trajectories. The first segment of the "middle" trajectory is the segment where there are $(n-1)/2$ first segments from other trajectories (assume n is odd) between the segment and the outer face (on each side of the segment).

Then, at every intersection, the median will switch to another trajectory if the continuation along this trajectory (without ever switching again) gives the same signature with the signature of one trajectory from T'. Figure **??** shows an example of this algorithm. After



Gambar 4.12: The blue and black trajectory are homotopically equivalent [?]



Gambar 4.13: Modified switching method [?]

starting with the red trajectory and switching to the green trajectory at the first intersection, the next intersection is with the blue trajectory (at the small blue square) and so far, the signature for the median is a^+. Although the blue trajectory is going forward, the signature after this intersection while following the blue trajectory (until the end point) is b^-c^-.

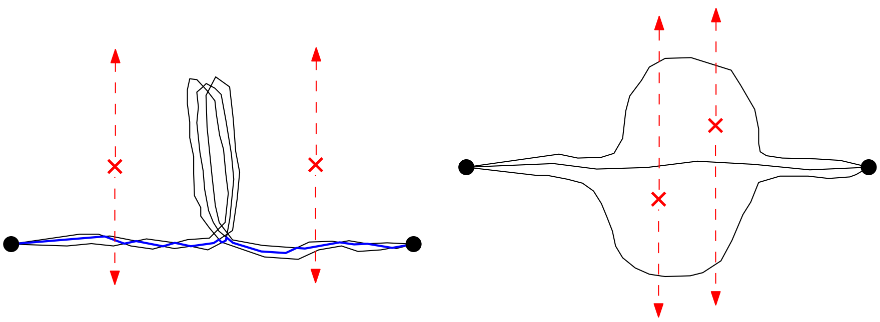
If the median switches at this intersection, the final signature will be $a^+b^-c^-$, which is not the signature of this set ($a^+b^-b^+b^-c^-$). At this point, the median does not switch to another trajectory. Instead, it continues to move along the green trajectory. The same situation also occurs when the median (now following the blue trajectory) intersects with the green trajectory (at the small green square).

Although this algorithm can produce a more suitable median trajectory for the situation where the switching method fails, the quality of the homotopic median trajectory depends heavily on the following factors:

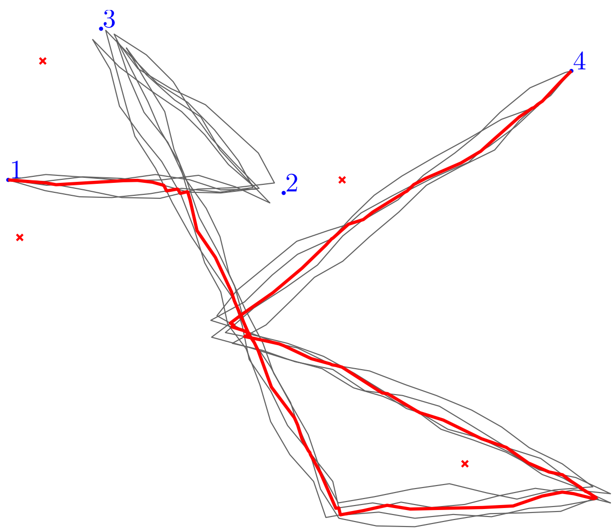
- placement of the crosses
- the number of trajectories which have the same signature

Therefore, in several cases the algorithm with the homotopy concept cannot produce suitable median trajectories. We give two examples in Figure **??**: in the left-hand side of figure, the final median trajectory (blue) does not follow other trajectories to the area with a narrow space. This problem arises because that narrow space is not large enough for a cross to be placed.

In the right-hand side of the figure, there are no two trajectories homotopically equivalent to each other. Nevertheless, by looking at their position, the median trajectory should be the one in the middle (between the two crosses). However, the algorithm with the homotopy concept does not guarantee that a suitable median trajectory will be found because there is no subset that contains the majority of trajectories in T. Figure **??** shows an example from [?], where the median trajectory does not completely follow what other trajectories do.



Gambar 4.14: Example cases where algorithm using homotopy will fail



Gambar 4.15: The median trajectory does not pass through part of trajectories in the upper left area

4.2.3 The Proposed Solutions

To solve the problems we mention in the previous section, we propose two algorithms to compute the median trajectory from a set of m trajectories (where each trajectory has n segments).

The first algorithm is an $O(1.2108^m + m^5n^5)$ worst-case time algorithm. This algorithm uses the Fréchet Distance [?] and works similar to the algorithm using the homotopy concept because both of them have to create the largest subset of similar trajectories and then compute the median trajectory by using parts of trajectories in that subset. By using the Fréchet Distance, we avoid the requirement to find proper places to put crosses, but still can produce suitable median trajectory in the situation where the homotopic algorithm fails (e.g. the example with a narrow space).

The second algorithm uses the combination of the buffer concept and Dijkstra's Shortest Path algorithm. Unlike all previous algorithms, this algorithm does not need to find the largest subset consisting similar trajectories. Using this algorithm, we can compute the median trajectory in $O(h^2 \log h)$ time in the worst-case, where h is the number of all segments in T $(h = O(mn))$.

We implemented the second algorithm in Java programming language and experiments have been done to determine the quality of the resulting median trajectory produced by this algorithm. To provide the test data (set of trajectories), we use a trajectories generator instead of using real-world data. This allow us to test much larger sets of trajectories

4.3 Outline of the Thesis

Chapter 2 describes the properties for a set of trajectories and also some properties the median trajectory should have.

The next two chapters explain in detail the two algorithms:

- Chapter 3 starts with a brief introduction of the Fréchet Distance and after that, we will explain how to use it to compute the median trajectory.

- Chapter 4 introduces the method to compute the median trajectory using the combination of the buffer concept and Dijkstra's shortest path algorithm.

We will give an explanation about our implementation, particularly on the implementation of the trajectories generator, in Chapter 5. In Chapter 6, we present the measures used to evaluate the quality of the median trajectory, the experiments set-up and the results from the experiments. This thesis will be concluded in Chapter 7 and 8, where we draw conclusions and discuss some issues and possible directions for further research.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus

mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Donec odio elit, dictum in, hendrerit sit amet, egestas sed, leo. Praesent feugiat sapien aliquet odio. Integer vitae justo. Aliquam vestibulum fringilla lorem. Sed neque lectus, consectetuer at, consectetuer sed, eleifend ac, lectus. Nulla facilisi. Pellentesque eget lectus. Proin eu metus. Sed porttitor. In hac habitasse platea dictumst. Suspendisse eu lectus. Ut mi mi, lacinia sit amet, placerat et, mollis vitae, dui. Sed ante tellus, tristique ut, iaculis eu, malesuada ac, dui. Mauris nibh leo, facilisis non, adipiscing quis, ultrices a, dui.

Morbi luctus, wisi viverra faucibus pretium, nibh est placerat odio, nec commodo wisi enim eget quam. Quisque libero justo, consectetuer a, feugiat vitae, porttitor eu, libero. Suspendisse sed mauris vitae elit sollicitudin malesuada. Maecenas ultricies eros sit amet ante. Ut venenatis velit. Maecenas sed mi eget dui varius euismod. Phasellus aliquet volutpat odio. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Pellentesque sit amet pede ac sem eleifend consectetuer. Nullam elementum, urna

vel imperdiet sodales, elit ipsum pharetra ligula, ac pretium ante justo a nulla. Curabitur tristique arcu eu metus. Vestibulum lectus. Proin mauris. Proin eu nunc eu urna hendrerit faucibus. Aliquam auctor, pede consequat laoreet varius, eros tellus scelerisque quam, pellentesque hendrerit ipsum dolor sed augue. Nulla nec lacus.

Suspendisse vitae elit. Aliquam arcu neque, ornare in, ullamcorper quis, commodo eu, libero. Fusce sagittis erat at erat tristique mollis. Maecenas sapien libero, molestie et, lobortis in, sodales eget, dui. Morbi ultrices rutrum lorem. Nam elementum ullamcorper leo. Morbi dui. Aliquam sagittis. Nunc placerat. Pellentesque tristique sodales est. Maecenas imperdiet lacinia velit. Cras non urna. Morbi eros pede, suscipit ac, varius vel, egestas non, eros. Praesent malesuada, diam id pretium elementum, eros sem dictum tortor, vel consectetuer odio sem sed wisi.

Sed feugiat. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Ut pellentesque augue sed urna. Vestibulum diam eros, fringilla et, consectetuer eu, nonummy id, sapien. Nullam at lectus. In sagittis ultrices mauris. Curabitur malesuada erat sit amet massa. Fusce blandit. Aliquam erat volutpat. Aliquam euismod. Aenean vel lectus. Nunc imperdiet justo nec dolor.

Etiam euismod. Fusce facilisis lacinia dui. Suspendisse potenti. In mi erat, cursus id, nonummy sed, ullamcorper eget, sapien. Praesent pretium, magna in eleifend egestas, pede pede pretium lorem, quis consectetuer tortor sapien facilisis magna. Mauris quis magna varius nulla scelerisque imperdiet. Aliquam non quam. Aliquam porttitor quam a lacus. Praesent vel arcu ut tortor cursus volutpat. In vitae pede quis diam bibendum placerat. Fusce elementum convallis neque. Sed dolor orci, scelerisque ac, dapibus nec, ultricies ut, mi. Duis nec dui quis leo sagittis commodo.

Aliquam lectus. Vivamus leo. Quisque ornare tellus ullamcorper nulla. Mauris porttitor pharetra tortor. Sed fringilla justo sed mauris. Mauris tellus. Sed non leo. Nullam elementum, magna in cursus sodales, augue est scelerisque sapien, venenatis congue nulla arcu et pede. Ut suscipit enim vel sapien. Donec congue. Maecenas urna mi, suscipit in, placerat ut, vestibulum ut, massa. Fusce ultrices nulla et nisl.

Etiam ac leo a risus tristique nonummy. Donec dignissim tincidunt nulla. Vestibulum rhoncus molestie odio. Sed lobortis, justo et pretium lobortis, mauris turpis condimentum augue, nec ultricies nibh arcu pretium enim. Nunc purus neque, placerat id, imperdiet sed, pellentesque nec, nisl. Vestibulum imperdiet neque non sem accumsan laoreet. In hac habitasse platea dictumst. Etiam condimentum facilisis libero. Suspendisse in elit quis nisl aliquam dapibus. Pellentesque auctor sapien. Sed egestas sapien nec lectus. Pellentesque vel dui vel neque bibendum viverra. Aliquam porttitor nisl nec pede. Proin mattis libero vel turpis. Donec rutrum mauris et libero. Proin euismod porta felis. Nam lobortis, metus quis elementum commodo, nunc lectus elementum mauris, eget vulputate ligula tellus eu neque. Vivamus eu dolor.

BAB 5

INTRODUCTION

5.1 Motivation

A trajectory is the motion path of a moving object. Various moving objects such as animals (probably in a wildlife area), hurricanes, or customers in a shopping area have trajectories that can provide valuable information. For example, trajectory data can be used to predict the movement of the same type of object in similar situation in the future.

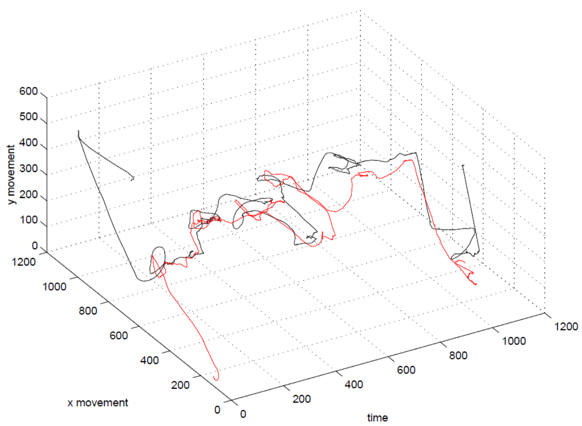
During a tracking procedure, the location of these moving objects can be obtained using various location detection devices (e.g: RFID, GPS devices and mobile phones). Later, this information will be sent to a database using any communication network (usually a wireless network). Because typical trajectory data is obtained during a specific interval, then trajectory data also has a temporal component, besides its spatial component.

The trajectory of a moving object is typically modeled as a sequence of consecutive locations in a multi-dimensional (generally two or three dimensional) Euclidean space [**?**]. Figure **??** shows an example of two trajectories from two objects which are moving in a 2D plane. With their temporal component, we can see that these trajectories are represented as polylines in a 3D space.

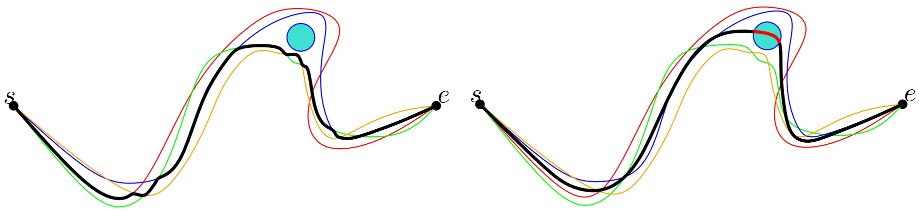
Nowadays, with the rapid development of technologies in mobile computing and wireless communication, many devices with location acquisition capabilities make it possible to obtain huge volumes of trajectory data from various moving objects. Furthermore, analysis of trajectory data is an important task for many applications that contain processing and managing moving objects, such as animal movements [**?**, **?**, **?**, **?**], traffic and transport analysis [**?**], defense and surveillance areas [**?**], oceanographic observations[1], weather and natural phenomena [**?**], people behavior [**?**] and sports [**?**, **?**].

Previous work on trajectory data analysis shows that there are several ways to analyze

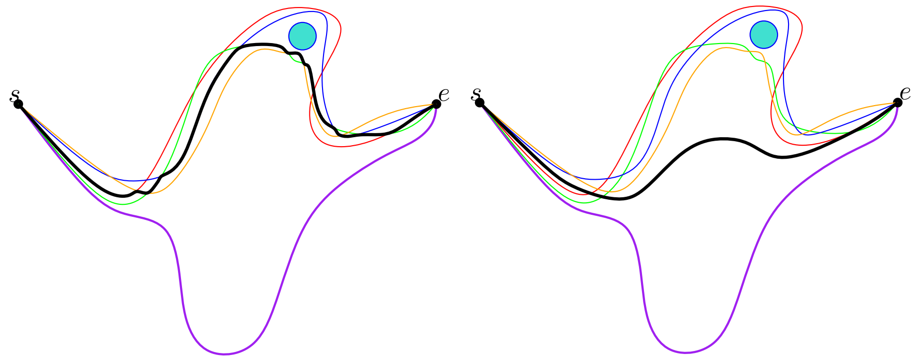
[1] W.S. Kessler, "Argo work in the coral sea."http://faculty.washington.edu/kessler/noumea/gliders/argo_coral_sea.html, March 2010.



Gambar 5.1: Example of 2D trajectories with time component, from [**?**]



Gambar 5.2: Example of the median (left) and the mean (right) trajectory [?]



Gambar 5.3: Robustness of the median trajectory

sets of trajectories. For example, similarity between trajectories can be determined [?, ?, ?]. Trajectories can also be clustered into groups with similar characteristics [?, ?, ?, ?]. Other examples are common data mining tasks such as classification [?, ?] and outlier detection [?]. Furthermore, interesting movement patterns such as flocking can also be computed from a set of trajectories [?, ?, ?, ?].

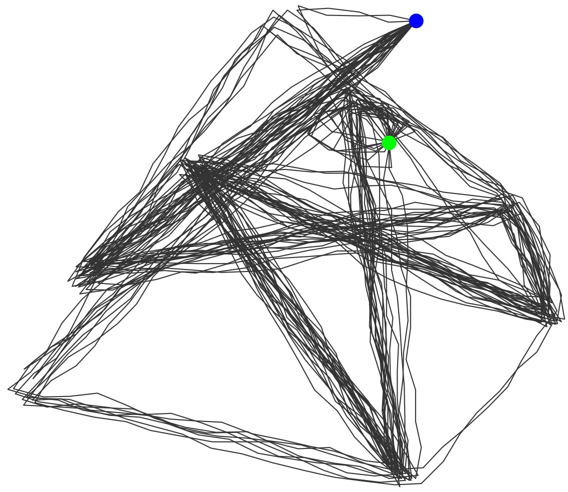
Even though analysis and research on trajectories has expanded in recent years, several basic concepts still need to be studied further. Some of them are the median and the mean trajectory for a collection/set of trajectories. The median and the mean trajectory share some common properties: they should be similar to other trajectories in the set and all parts of them should be located roughly in the middle of the set. However, there are several important differences between them: Firstly, the median trajectory must use only parts of trajectories in the set. It uses only parts of one trajectory or combines parts from many different trajectories. This property might be a disadvantage for the median because some parts of it can be located not in the middle of the set, but in several situations this might be useful.

Figure **??** shows a set of four trajectories which avoid the light-blue obstacle. The possible mean trajectory (the black trajectory in the right-hand side of the figure) will pass through the obstacle because the mean must lie in the middle of all trajectories in the set. In this case, it is clear that the mean trajectory is not suitable for a path of a moving object.

The median trajectory (black trajectory in the left-hand side of the figure) gives a more suitable path because it always uses parts of other trajectories. In parts near the obstacle, the median is not really in the middle of other trajectories.

Secondly, the median trajectory is more robust against outliers than the mean trajectory. Figure **??** shows this situation: we add one trajectory (with purple color) which can be categorized as an outlier compared to other trajectories. While the median trajectory only needs to be modified a little bit, the mean trajectory has to be changed a lot (comparing to the mean trajectory in Figure **??**), to keep it in the middle of other trajectories.

In this thesis, we will not cover the mean trajectory and only discuss the median trajectory and algorithms to compute it. We also ignore the temporal component of the trajectory because it is not clear yet how to take it into account when computing the median trajectory. However, some research on motion and kinetic data structures contains a temporal component and are related to the median/mean trajectory [?, ?].



Gambar 5.4: The set of 30 trajectories,starting at the blue point & ending at the green point

For other types of data, a median has a clear definition. The median from a population (or a sample) of integer numbers is the number that separates the population into two halves, where at most half of the population have a smaller value and the other half of the population have a larger value than the median.

For geometric data types, the concept of median also exist. A *center point* of a set P of n points in the plane is a point such that any closed half-plane whose bounding line contains the center point, contains at least $n/3$ points of P [**?**]. If we force the center point to be one of the points from P, then we obtain a 2-dimensional version of the median, although the "quality" of this median can be bad.

The median trajectory does not have any formal definition yet. Based on several properties that we mention earlier, such as its similarity with other trajectories and lying approximately in the middle of the set, we can determine a possible median, which can useful in several ways:

5.1.1 Determine the most typical trajectory

The property of the median trajectory makes it suitable to analyze the movement behavior or movement pattern from a group of same objects because the median somehow represents the whole trajectories in the set/collection. The median trajectory properties, such as the length, the direction or the average speed (if we include the temporal component), could give valuable information.

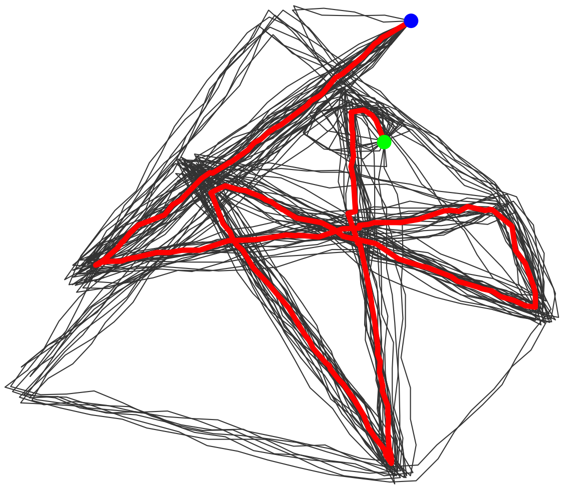
Example applications include the detection of outliers, which can be done by analyzing the length and the similarity of the shape of the median with other trajectories. Analyzing the average speed together with the shape of the median trajectory might be useful to understand the behavior and the movement pattern of people walking around in an area which has several interesting places to be visited (e.g., a zoo or an amusement park).

5.1.2 Better visualization for a set of trajectories

Visualization of the median trajectory, together with its set of trajectories, might give the viewer a better interpretation and information about the set of trajectories.

We give an example in Figure **??**, where a set has 30 trajectories, which is paths of 30 objects moving from the blue point to the green point. From this figure, we can hardly tell anything about the general behavior or the direction of these trajectories. However, we know that several trajectories are different than others and probably can predict what the majority does, but it is still difficult to visualize what the majority of these trajectories does.

In the following figure (Figure **??**), we present a possible median trajectory as the red and thick trajectory. From this visualization, it is clear what the majority of trajectories



Gambar 5.5: A set of 30 trajectories with its possible median trajectory

does. Moreover, we can identify what parts of some trajectories are completely different from others.

The visualization of the median trajectory could be useful in some real-life applications: The median trajectory from trajectories of visitors in a national park can be used to see the most common path taken by visitors, which is probably the path that is preferred by future visitors. This information might be useful if we want to create a map that can help those visitors by providing valuable information about a path and direction on that national park in the map, so that he/she can decide which path that he/she will take.

5.1.3 k-medoid clustering

Another application that could use the median trajectory is the k-medoid algorithm, which is used in cluster analysis. The k-medoid clustering algorithm is related to the k-means algorithm, a method to partition/group a set of objects into k different clusters containing similar objects.

In general, each cluster in both algorithms has one object act as a *central object* and other members of the cluster should be similar or having a small distance to this object. The similarity or the distance between objects can be measured using different distance functions (e.g. Euclidean distance, Minkowski distance, etc), depending on the type of objects and the purpose of the clustering.

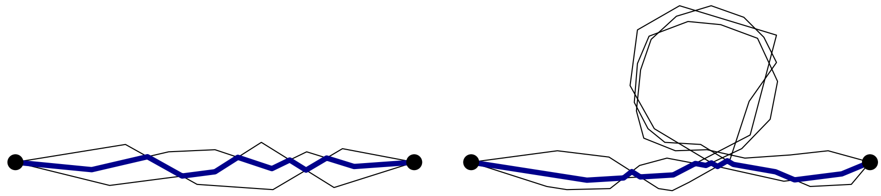
The main difference between the two algorithms is on the selection of the central object for each cluster. While the k-means simply uses the mean of objects, the k-medoid must use the medoid (an object which has the smallest average of dissimilarity/distance to all other objects in the set, but it must be a member of the set). This implies that k-means could create a new object to be the central object whereas the k-medoid must use one of the objects from the set. Thus, the k-medoid algorithm is more suitable for spatial clustering purposes and less sensitive against noise and outliers.

Partitioning Around Medoids (PAM) [?] is a basic k-medoid clustering algorithm. It works as follow:

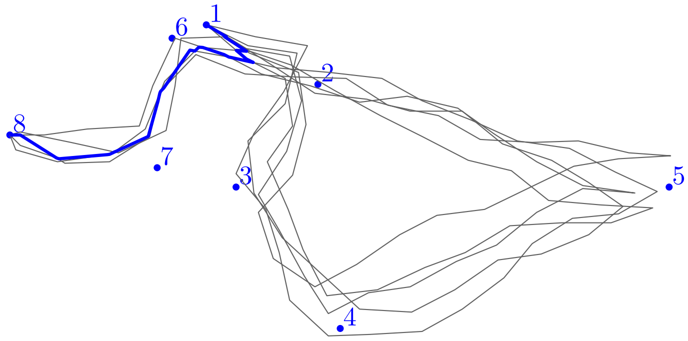
1. Define a value k and choose k objects as a set of medoids.

2. Assign every object to its closest/similar medoid and after that, compute the cost for the whole configuration.

3. Find another configuration by selecting a pair of medoid and non-medoid objects which have the smallest distance cost and swapping them temporarily. Then, we assign all other objects to this temporary set of medoids and obtain a new configuration.



Gambar 5.6: Illustration of the simple idea using switching



Gambar 5.7: The median trajectory makes a shortcut [?]

4. If the new configuration has smaller cost than the last configuration, then we change the set of medoids and return to step 3

5. Otherwise, stop and we find the set of medoids with their non-overlapping set of clusters.

In case we want to cluster a set of trajectories, we can use the median trajectory as a medoid in this algorithm. However, some changes probably should be made. For example, finding another configuration is not done by simply swapping the median with other trajectory, instead we can choose to swap part of them (with the requirement that both trajectories intersect one another).

5.2 Basic Idea of the Research

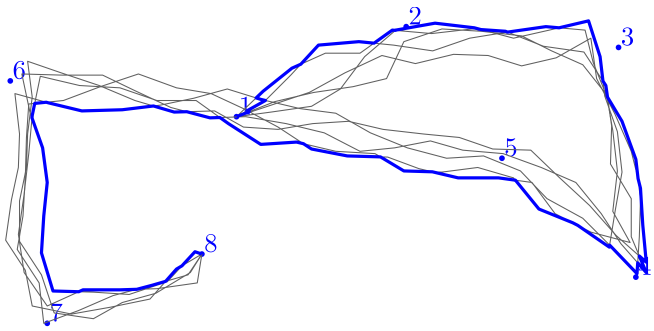
Consider a set T of m trajectories. We want to to compute the median trajectory of T. In this set, all trajectories have the same start and end points. The median trajectory of T must be built using parts of trajectories in T and somehow must follow what other trajectories in T do, while staying in the "middle" of other trajectories.

5.2.1 The Simple Switching Method

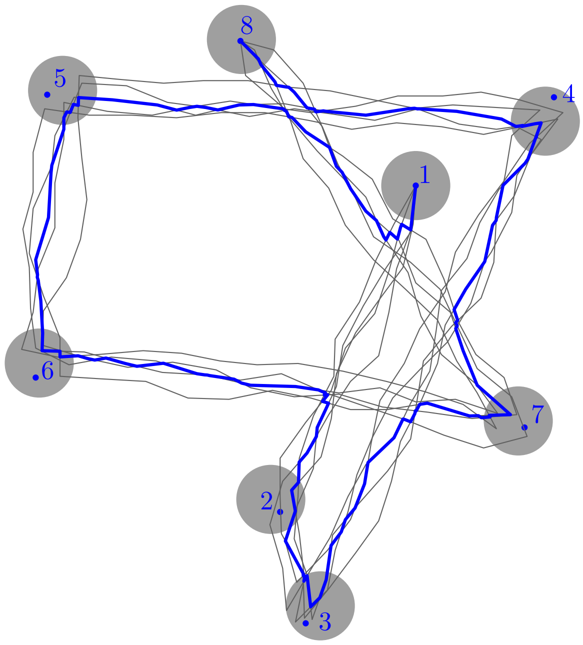
A simple idea to obtain a median trajectory from T is to start from the "middle" trajectory, which is the $(m+1)/2$-level of arrangement formed by all trajectories in T (we assume m is odd). At every intersection point, the median trajectory will switch to another trajectory and keep $(m+1)/2$ trajectories above and below the median [?].

Figure **??** shows the result (the median trajectory is the thick-blue trajectory) of this approach for two different types of set of trajectories, one of them contains trajectories with self intersection. From the right-hand side of the figure in Figure **??**, we can see that this method cannot produce suitable median trajectory because the median does not follow the loop created by the three trajectories.

In general, this method will not give a suitable median if a set of trajectories contains self-intersecting trajectories. More examples from [?] show several "incorrect" median trajectories obtained by using this simple switching method. The blue median trajectory in Figure **??** makes a shortcut path to the end point.



Gambar 5.8: The median trajectory does not stay in the middle [?]



Gambar 5.9: The median trajectory does not follow the correct sequence of regions [?]

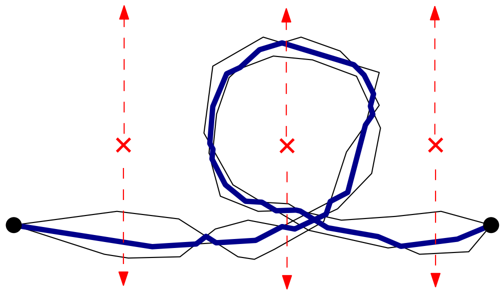
The median trajectory in Figure **??** does not stay in the "middle" of other trajectories. Finally, in Figure **??**, the median trajectory does not follow the sequence of regions as the other trajectories. The correct sequence of regions is $1-2-3-4-5-6-7-8$.

5.2.2 The Algorithm Using the Concept of Homotopy

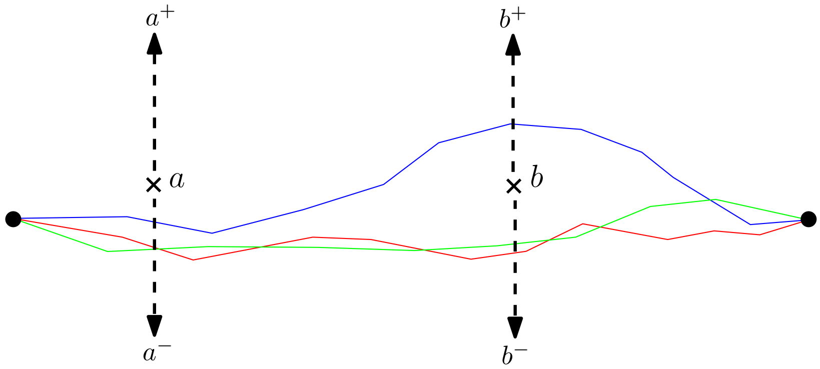
Another algorithm to compute the median trajectory uses the concept of homotopy (along with the modified simple switching method) [?]. This algorithm works by placing cross in a relatively large face bounded by segments from a set of trajectories. Figure **??** shows an example where cross is placed in the relatively large bounded face and two crosses are placed in the outer face.

Based on the location of these crosses, each trajectory in T will be assigned a *signature*. Figure **??** shows three trajectories and two crosses a and b. From these two crosses, four half-lines are created: a^+ and a^- are half-lines above and below a, while b^+ and b^- are half-lines above and below b, respectively.

For all trajectories in T, we give them a signature based on how they intersect with the half-line(s) from the crosses. Note that each trajectory might have a different signature, because it depends on the position of the trajectory with respect to all crosses in the plane. In Figure **??**, the blue trajectory intersect with a^- and b^+, thus its signature will be a^-b^+ (the order is following the direction of the trajectory). In the same way, the signature of the red trajectory will be the same as the green trajectory: a^-b^-.



Gambar 5.10: Illustration of the algorithm using homotopy concept



Gambar 5.11: Trajectories and crosses

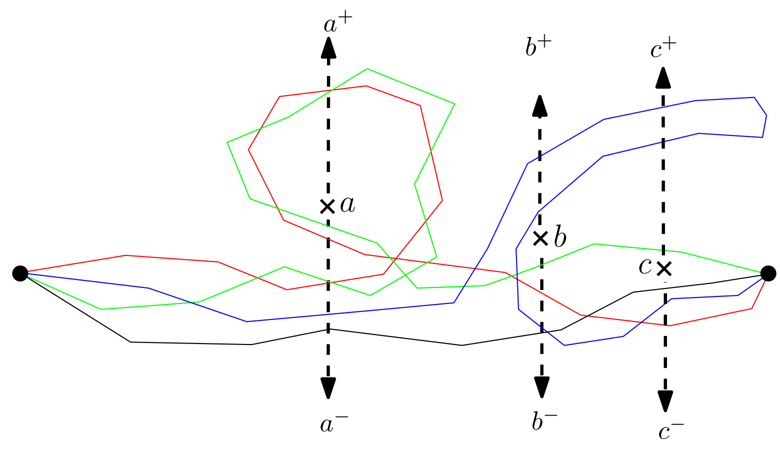
Two different trajectories are homotopic if one trajectory can be deformed continuously into the other one without passing through any crosses, while the start and end point are not moved. Naturally, two trajectories are homotopically equivalent if their signatures are exactly the same. However, two homotopic trajectories do not always have the same signatures. We shown an example in Figure **??** where the blue and the black trajectory are homotopic, but their signatures are different.

To determine whether two trajectories with different signature are homotopic or not, we perform a *reduce* operation. This *reduce* operation works by eliminating two exact same signs, if their position is next to each other in the signature. In Figure **??**, the signature of the blue trajectory is $a^-b^+c^+c^+b^+b^-c^-$. Notice that it has two c^+ that we can eliminate. This will change the signature of the blue trajectory into $a^-b^+b^+b^-c^-$. Once again, we can identify that two b^+ are positioned directly to each other. Performing the reduce operation again, we will get the final signature of the blue trajectory: $a^-b^-c^-$. At this point, we cannot apply the reduce operation again to this signature, and we say that the signature has been *maximally reduced*. Finally,we conclude that if two trajectories have the same maximally reduced signature, then the two trajectories are homotopically equivalent.

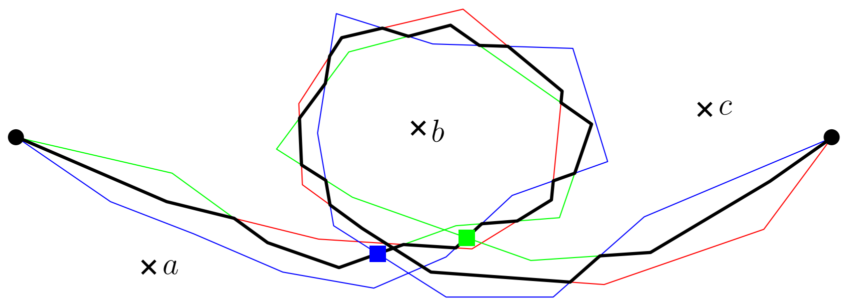
The next step of the algorithm is to create a subset T' of T, and then find the median trajectory by using only parts of trajectories from T'. Creating T' is straightforward, we only need to compute maximally reduced signature for all trajectories and choose a subset with the largest number of trajectories which have the same signature.

To create the median trajectory from T', we use the modified version of the switching method. This method start at the first segment of the"middle" trajectory. We find such a segment by determine the outer face of the set of trajectories. The first segment of the "middle" trajectory is the segment where there are $(n-1)/2$ first segments from other trajectories (assume n is odd) between the segment and the outer face (on each side of the segment).

Then, at every intersection, the median will switch to another trajectory if the continuation along this trajectory (without ever switching again) gives the same signature with the signature of one trajectory from T'. Figure **??** shows an example of this algorithm. After



Gambar 5.12: The blue and black trajectory are homotopically equivalent [?]



Gambar 5.13: Modified switching method [?]

starting with the red trajectory and switching to the green trajectory at the first intersection, the next intersection is with the blue trajectory (at the small blue square) and so far, the signature for the median is a^+. Although the blue trajectory is going forward, the signature after this intersection while following the blue trajectory (until the end point) is b^-c^-.

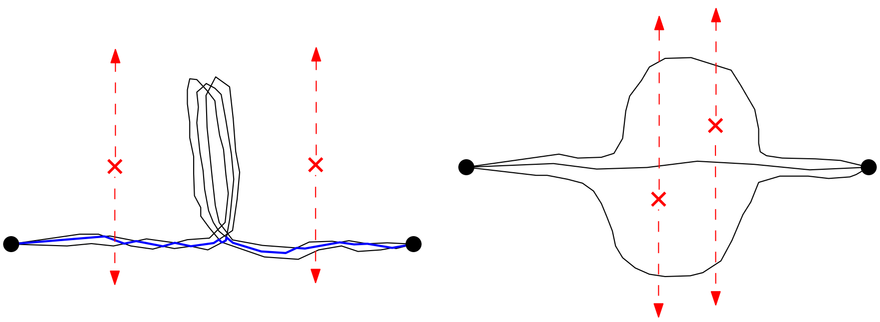
If the median switches at this intersection, the final signature will be $a^+b^-c^-$, which is not the signature of this set $(a^+b^-b^+b^-c^-)$. At this point, the median does not switch to another trajectory. Instead, it continues to move along the green trajectory. The same situation also occurs when the median (now following the blue trajectory) intersects with the green trajectory (at the small green square).

Although this algorithm can produce a more suitable median trajectory for the situation where the switching method fails, the quality of the homotopic median trajectory depends heavily on the following factors:

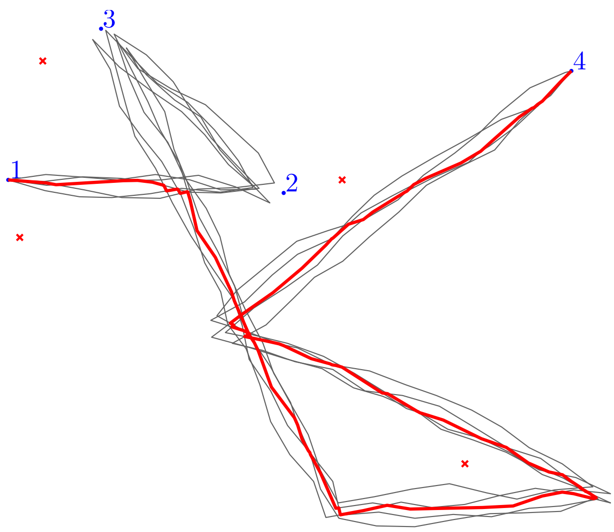
- placement of the crosses
- the number of trajectories which have the same signature

Therefore, in several cases the algorithm with the homotopy concept cannot produce suitable median trajectories. We give two examples in Figure **??**: in the left-hand side of figure, the final median trajectory (blue) does not follow other trajectories to the area with a narrow space. This problem arises because that narrow space is not large enough for a cross to be placed.

In the right-hand side of the figure, there are no two trajectories homotopically equivalent to each other. Nevertheless, by looking at their position, the median trajectory should be the one in the middle (between the two crosses). However, the algorithm with the homotopy concept does not guarantee that a suitable median trajectory will be found because there is no subset that contains the majority of trajectories in T. Figure **??** shows an example from [?], where the median trajectory does not completely follow what other trajectories do.



Gambar 5.14: Example cases where algorithm using homotopy will fail



Gambar 5.15: The median trajectory does not pass through part of trajectories in the upper left area

5.2.3 The Proposed Solutions

To solve the problems we mention in the previous section, we propose two algorithms to compute the median trajectory from a set of m trajectories (where each trajectory has n segments).

The first algorithm is an $O(1.2108^m + m^5 n^5)$ worst-case time algorithm. This algorithm uses the Fréchet Distance [?] and works similar to the algorithm using the homotopy concept because both of them have to create the largest subset of similar trajectories and then compute the median trajectory by using parts of trajectories in that subset. By using the Fréchet Distance, we avoid the requirement to find proper places to put crosses, but still can produce suitable median trajectory in the situation where the homotopic algorithm fails (e.g. the example with a narrow space).

The second algorithm uses the combination of the buffer concept and Dijkstra's Shortest Path algorithm. Unlike all previous algorithms, this algorithm does not need to find the largest subset consisting similar trajectories. Using this algorithm, we can compute the median trajectory in $O(h^2 \log h)$ time in the worst-case, where h is the number of all segments in T ($h = O(mn)$).

We implemented the second algorithm in Java programming language and experiments have been done to determine the quality of the resulting median trajectory produced by this algorithm. To provide the test data (set of trajectories), we use a trajectories generator instead of using real-world data. This allow us to test much larger sets of trajectories

5.3 Outline of the Thesis

Chapter 2 describes the properties for a set of trajectories and also some properties the median trajectory should have.

The next two chapters explain in detail the two algorithms:

- Chapter 3 starts with a brief introduction of the Fréchet Distance and after that, we will explain how to use it to compute the median trajectory.

- Chapter 4 introduces the method to compute the median trajectory using the combination of the buffer concept and Dijkstra's shortest path algorithm.

We will give an explanation about our implementation, particularly on the implementation of the trajectories generator, in Chapter 5. In Chapter 6, we present the measures used to evaluate the quality of the median trajectory, the experiments set-up and the results from the experiments. This thesis will be concluded in Chapter 7 and 8, where we draw conclusions and discuss some issues and possible directions for further research.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus

mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Donec odio elit, dictum in, hendrerit sit amet, egestas sed, leo. Praesent feugiat sapien aliquet odio. Integer vitae justo. Aliquam vestibulum fringilla lorem. Sed neque lectus, consectetuer at, consectetuer sed, eleifend ac, lectus. Nulla facilisi. Pellentesque eget lectus. Proin eu metus. Sed porttitor. In hac habitasse platea dictumst. Suspendisse eu lectus. Ut mi mi, lacinia sit amet, placerat et, mollis vitae, dui. Sed ante tellus, tristique ut, iaculis eu, malesuada ac, dui. Mauris nibh leo, facilisis non, adipiscing quis, ultrices a, dui.

Morbi luctus, wisi viverra faucibus pretium, nibh est placerat odio, nec commodo wisi enim eget quam. Quisque libero justo, consectetuer a, feugiat vitae, porttitor eu, libero. Suspendisse sed mauris vitae elit sollicitudin malesuada. Maecenas ultricies eros sit amet ante. Ut venenatis velit. Maecenas sed mi eget dui varius euismod. Phasellus aliquet volutpat odio. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Pellentesque sit amet pede ac sem eleifend consectetuer. Nullam elementum, urna

vel imperdiet sodales, elit ipsum pharetra ligula, ac pretium ante justo a nulla. Curabitur tristique arcu eu metus. Vestibulum lectus. Proin mauris. Proin eu nunc eu urna hendrerit faucibus. Aliquam auctor, pede consequat laoreet varius, eros tellus scelerisque quam, pellentesque hendrerit ipsum dolor sed augue. Nulla nec lacus.

Suspendisse vitae elit. Aliquam arcu neque, ornare in, ullamcorper quis, commodo eu, libero. Fusce sagittis erat at erat tristique mollis. Maecenas sapien libero, molestie et, lobortis in, sodales eget, dui. Morbi ultrices rutrum lorem. Nam elementum ullamcorper leo. Morbi dui. Aliquam sagittis. Nunc placerat. Pellentesque tristique sodales est. Maecenas imperdiet lacinia velit. Cras non urna. Morbi eros pede, suscipit ac, varius vel, egestas non, eros. Praesent malesuada, diam id pretium elementum, eros sem dictum tortor, vel consectetuer odio sem sed wisi.

Sed feugiat. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Ut pellentesque augue sed urna. Vestibulum diam eros, fringilla et, consectetuer eu, nonummy id, sapien. Nullam at lectus. In sagittis ultrices mauris. Curabitur malesuada erat sit amet massa. Fusce blandit. Aliquam erat volutpat. Aliquam euismod. Aenean vel lectus. Nunc imperdiet justo nec dolor.

Etiam euismod. Fusce facilisis lacinia dui. Suspendisse potenti. In mi erat, cursus id, nonummy sed, ullamcorper eget, sapien. Praesent pretium, magna in eleifend egestas, pede pede pretium lorem, quis consectetuer tortor sapien facilisis magna. Mauris quis magna varius nulla scelerisque imperdiet. Aliquam non quam. Aliquam porttitor quam a lacus. Praesent vel arcu ut tortor cursus volutpat. In vitae pede quis diam bibendum placerat. Fusce elementum convallis neque. Sed dolor orci, scelerisque ac, dapibus nec, ultricies ut, mi. Duis nec dui quis leo sagittis commodo.

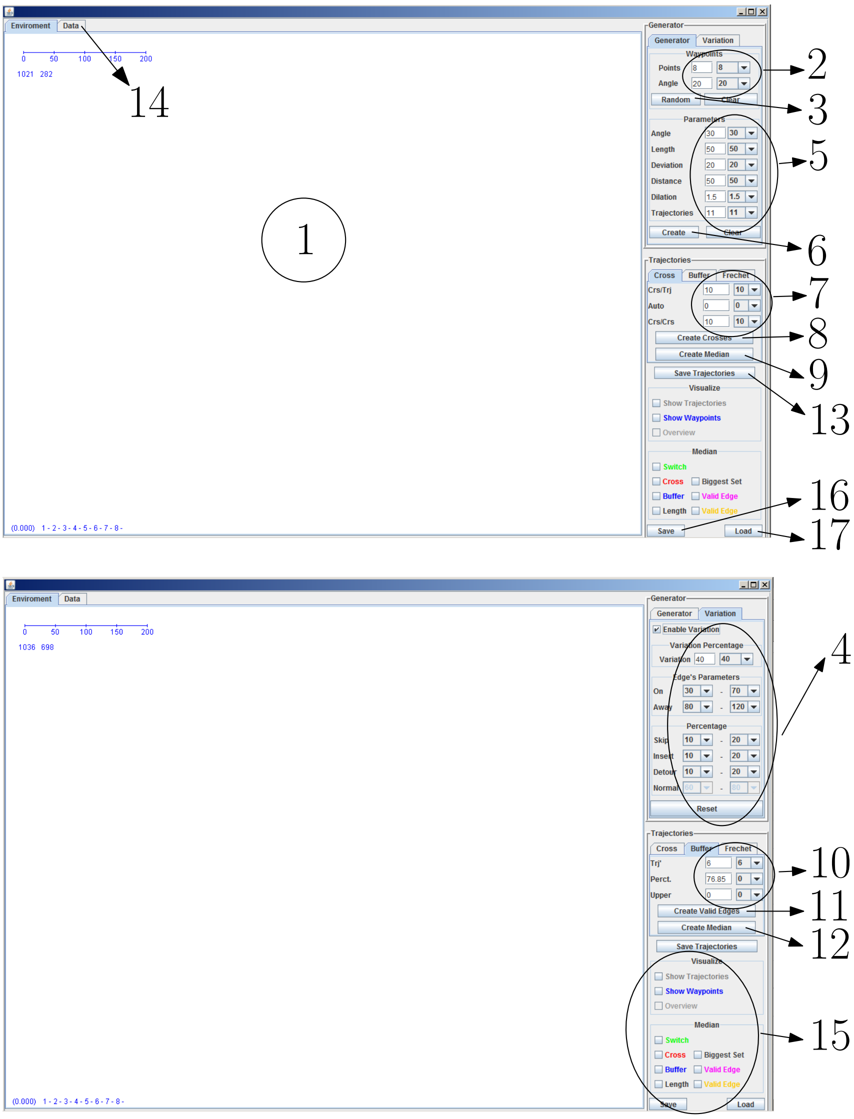
Aliquam lectus. Vivamus leo. Quisque ornare tellus ullamcorper nulla. Mauris porttitor pharetra tortor. Sed fringilla justo sed mauris. Mauris tellus. Sed non leo. Nullam elementum, magna in cursus sodales, augue est scelerisque sapien, venenatis congue nulla arcu et pede. Ut suscipit enim vel sapien. Donec congue. Maecenas urna mi, suscipit in, placerat ut, vestibulum ut, massa. Fusce ultrices nulla et nisl.

Etiam ac leo a risus tristique nonummy. Donec dignissim tincidunt nulla. Vestibulum rhoncus molestie odio. Sed lobortis, justo et pretium lobortis, mauris turpis condimentum augue, nec ultricies nibh arcu pretium enim. Nunc purus neque, placerat id, imperdiet sed, pellentesque nec, nisl. Vestibulum imperdiet neque non sem accumsan laoreet. In hac habitasse platea dictumst. Etiam condimentum facilisis libero. Suspendisse in elit quis nisl aliquam dapibus. Pellentesque auctor sapien. Sed egestas sapien nec lectus. Pellentesque vel dui vel neque bibendum viverra. Aliquam porttitor nisl nec pede. Proin mattis libero vel turpis. Donec rutrum mauris et libero. Proin euismod porta felis. Nam lobortis, metus quis elementum commodo, nunc lectus elementum mauris, eget vulputate ligula tellus eu neque. Vivamus eu dolor.

LAMPIRAN A

THE PROGRAM

The interface of the program is shown in Figure **??**:



Gambar A.1: Interface of the program

Step by step to compute the median trajectory using the program:

1. Create several waypoints. Click anywhere in the "Environment" area(1) or create them automatically by setting the parameters for waypoint(2) or clicking the button "Random"(3).

2. The "Variation" tab could be used to create variations by providing values needed to make them(4).

3. Create a set of trajectories by setting all parameters(5) and clicking the button "Create"(6).

4. Compute the median using the homotopic algorithm:

   - Define all parameters needed for the homotopic algorithm(7).
   - Create crosses by clicking the "Create Crosses" button(8).
   - Compute the median by clicking the "Compute Median" button(9).

5. Compute the median using the switching method and the buffer algorithm:

   - Define all parameters needed for the buffer algorithm(10).
   - Create valid edges by clicking the "Create Valid Edges"button(11).
   - Compute the median by clicking the "Compute Median"button(12).

6. Save the resulting median by clicking the "Save Trajectories" button(13). The result is saved in the computer memory and can be seen in "Data" tab(14)

7. The set of trajectories and its median trajectories will appear in the "Environment" area(1) and the user can change what to display by selecting various choices in "Visualize" and "Median" area(15).

8. To save all data to the disk, click the "Save"(16) button. A file dialog menu will appear.

9. To load data from the disk, click the "Load"(17) button.

LAMPIRAN B

THE SOURCE CODE

Listing B.1: MyFurSet.java

```java

import java.util.ArrayList;
import java.util.Collections;
import java.util.HashSet;

/**
 *
 * @author Lionov
 */

//class for set of vertices close to furthest edge
public class MyFurSet {
    protected int id;                                       //id of the set
    protected MyEdge FurthestEdge;                          //the furthest edge
    protected HashSet<MyVertex> set;                        //set of vertices close to furthest edge
    protected ArrayList<ArrayList<Integer>> ordered;        //list of all vertices in the set for each
         trajectory
    protected ArrayList<Integer> closeID;                   //store the ID of all vertices
    protected ArrayList<Double> closeDist;                  //store the distance of all vertices
    protected int totaltrj;                                 //total trajectories in the set

    /**
     * Constructor
     * @param id : id of the set
     * @param totaltrj : total number of trajectories in the set
     * @param FurthestEdge : the furthest edge
     */
    public MyFurSet(int id,int totaltrj,MyEdge FurthestEdge) {
        this.id = id;
        this.totaltrj = totaltrj;
        this.FurthestEdge = FurthestEdge;
        set = new HashSet<MyVertex>();
        ordered = new ArrayList<ArrayList<Integer>>();
        for (int i=0;i<totaltrj;i++) ordered.add(new ArrayList<Integer>());
        closeID = new ArrayList<Integer>(totaltrj);
        closeDist = new ArrayList<Double>(totaltrj);
        for (int i = 0;i <totaltrj;i++) {
            closeID.add(-1);
            closeDist.add(Double.MAX_VALUE);
        }
    }

    /**
     * set a vertex into the set
     * @param v : vertex to be added to the set
     */
    public void add(MyVertex v) {
        set.add(v);
    }

    /**
     * check whether vertex v is a member of the set
     * @param v : vertex to be checked
     * @return true if v is a member of the set, false otherwise
     */
    public boolean contains(MyVertex v) {
        return this.set.contains(v);
    }
}
```

LAMPIRAN C

THE SOURCE CODE

Listing C.1: MyFurSet.java

```java

import java.util.ArrayList;
import java.util.Collections;
import java.util.HashSet;

/**
 *
 * @author Lionov
 */

//class for set of vertices close to furthest edge
public class MyFurSet {
    protected int id;                                    //id of the set
    protected MyEdge FurthestEdge;                       //the furthest edge
    protected HashSet<MyVertex> set;                     //set of vertices close to furthest edge
    protected ArrayList<ArrayList<Integer>> ordered;     //list of all vertices in the set for each trajectory
    protected ArrayList<Integer> closeID;                //store the ID of all vertices
    protected ArrayList<Double> closeDist;               //store the distance of all vertices
    protected int totaltrj;                              //total trajectories in the set

    /**
     * Constructor
     * @param id : id of the set
     * @param totaltrj : total number of trajectories in the set
     * @param FurthestEdge : the furthest edge
     */
    public MyFurSet(int id,int totaltrj,MyEdge FurthestEdge) {
        this.id = id;
        this.totaltrj = totaltrj;
        this.FurthestEdge = FurthestEdge;
        set = new HashSet<MyVertex>();
        ordered = new ArrayList<ArrayList<Integer>>();
        for (int i=0;i<totaltrj;i++) ordered.add(new ArrayList<Integer>());
        closeID = new ArrayList<Integer>(totaltrj);
        closeDist = new ArrayList<Double>(totaltrj);
        for (int i = 0;i <totaltrj;i++) {
            closeID.add(-1);
            closeDist.add(Double.MAX_VALUE);
        }
    }

    /**
     * set a vertex into the set
     * @param v : vertex to be added to the set
     */
    public void add(MyVertex v) {
        set.add(v);
    }

    /**
     * check whether vertex v is a member of the set
     * @param v : vertex to be checked
     * @return true if v is a member of the set, false otherwise
     */
    public boolean contains(MyVertex v) {
        return this.set.contains(v);
    }

    /**
     *  create a column for table Gamma, sorted for each row
     */
    public void createColumn() {
        for (MyVertex v : set) {
            for (Integer key : v.vertexnum.keySet()) {
                for (Integer values : v.vertexnum.get(key)) {
                    ordered.get(key).add(values);
                }
            }
        }
        for (ArrayList<Integer> al : ordered) Collections.sort(al);
    }


}
```