Towards Dynamic River Networks

Beurskens, Thijs P.J.

Award date:
2022

Department of Mathematics and Computer Science

Algorithms, Geometry & Applications

Towards Dynamic River Networks

Master's Thesis

T. P. J. Beurskens

19-10-2022

Supervision:

prof. dr. B. Speckmann

dr. O.T.C. Tse

dr. T.A.E. Ophelders

Assessment committee:

prof. dr. B. Speckmann

dr. O.T.C. Tse

dr. T.A.E. Ophelders

Credits: 45

This is a public Master's Thesis.

This Master's Thesis has been carried out in accordance with the rules of the TU/e Code of Scientific Conduct.

Abstract

Rivers are one of the main contributors to the shape of the landscape, and they are often unpredictable and can cause floods and landslides. To understand how a river evolves, we want to find methods to analyse the complex behaviour of rivers, but before we can find such methods, we need a way to represent a time-varying river. As of now, there are no satisfying automated ways to extract an abstract time-varying representation of a time-varying river, based on DEMs of the input terrain. Therefore, we propose two models for dynamic river networks, which capture the changing channel structure in evolving rivers, by matching features over subsequent time steps. The first model is based on the use of a similarity measure, comparing channels, and the second model is based on the use of a displacement field, describing the evolution of the terrain. To find adequate models for such measures and displacement fields, we use various existing techniques, such as a volume-based similarity measure, and the theory of optimal transport, to experimentally find desirable channel matchings. The two proposed models for a dynamic river network provide two promising building blocks that can lead to a desirable representation of a time-varying river. Moreover, through the process of modelling such networks, we identify and pose some of the key challenges in pursuing algorithms for the construction of a dynamic river network.

Contents

1 Introduction **1**
- 1.1 Time-varying river networks . . . 2
- 1.2 Similarity measures . . . 6
- 1.3 Displacement fields . . . 8
- 1.4 Implementation . . . 10

2 Modelling dynamic river networks **11**
- 2.1 Channel matchings . . . 12
- 2.2 A suitable objective . . . 14
  - 2.2.1 Maximising similarity . . . 15
  - 2.2.2 Minimising displacement . . . 16
  - 2.2.3 Alternatives and extensions . . . 17
- 2.3 Discussion. . . . 18

3 Comparing channels **19**
- 3.1 Sand function . . . 21
- 3.2 Dealing with intersections . . . 23
- 3.3 Dealing with endpoints . . . 26
- 3.4 Dealing with an evolving terrain . . . 27
- 3.5 Discussion . . . 28

4 Transforming the terrain **30**
- 4.1 Optimal transport map . . . 30
- 4.2 Transporting terrains . . . 34
- 4.3 Discussion . . . 36

5 Implementation and experiments **38**
- 5.1 Implementation . . . 42
- 5.2 Experimenting with a similarity measure . . . 44
- 5.3 Experimenting with a displacement field . . . 46
- 5.4 Discussion . . . 52

6 Concluding remarks **54**

Bibliography **56**

Chapter 1

Introduction

Rivers play an important role in the history of humankind. As they provide a source of drinking water, a natural defence barrier and an easy way to transport goods, rivers have often been chosen as locations for settlements. However, rivers can also be dangerous to human life. Rivers can change their course, due to processes such as erosion, which can lead to natural disasters, like floods and landslides. Over time, humans found ways to influence the course and shape of a river. Dams can be build to prevent floods, and rivers can be dredged to allow for larger ships. As a consequence, there are both natural and artificial factors contributing to the course of a river, making them act unpredictable and hard to study.

There are in general two complementary approaches to studying rivers: mathematical modelling and data analysis. Modelling rivers is one of the primal focuses of *geomorphology* - the study of processes near the Earth's surface. Geomorphologists try to find laws governing the behaviour of landscapes, rivers and their interaction. These laws lead to a variety of models of changing landscapes, which can be used to run simulations. On the other hand, algorithms to analyse data of simulated and real-world rivers can help answer questions about the dynamics of rivers. With advancements in numerical simulations and remote sensing techniques, there is an extensive growth in the amount of available data. Due to the geometric nature of the data, *computational geometry* plays an important role in the design of algorithms.

Multi-channel rivers. The focus of this thesis lies on developing geometric algorithms for the analysis of multi-channel rivers, like *braided rivers* and *estuaries*. A braided river [31] consists of many, often shallow, *channels* that are separated by small islands called *bars* (see Figure 1.1). Channels split at *bifurcations*, stay separated along the length of a bar and join again at a *confluence*. An estuary is the part of a river that streams into a sea, and thus exhibits both marine as well as fluvial processes. Estuaries are influenced by tidal processes, meaning that the water level fluctuates significantly

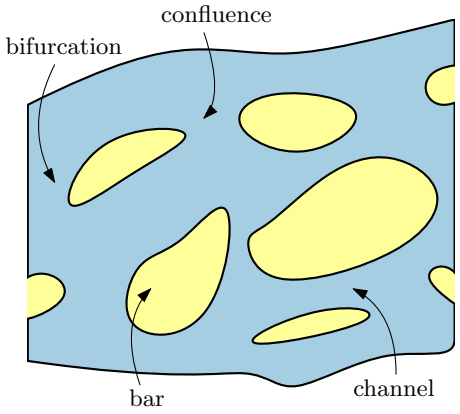


Figure 1.1: Illustrative drawing of a braided river.

and frequently. Typically, estuaries are larger than braided systems, but their channels display similar properties.

All rivers evolve over time, due to natural process such as riverbed erosion and meandering. Braided rivers are among the most dynamic rivers, which manifests in a rapidly changing channel structure [39]. Existing research on braided rivers has shown that these rivers develop and maintain their multi-channel structure due to a combination of natural processes [23, 35]. Wheaton et al. [57] present a classification of 10 different morphodynamic mechanisms, and they show the contribution of each of these to the dynamics of braided rivers. In a similar fashion, we are interested in the effects of disturbances, for instance due to dredging, to the network structure of multi-channel rivers.

1.1 Time-varying river networks

To predict how perturbations in a multi-channel river propagate through the channel structure, it is necessary to understand how these rivers evolve over time. To our knowledge, there is currently no automated method to easily track components of the river, such as channels or bars, over subsequent time steps. A possibility to study the evolution of rivers is to do the analysis by hand. However, that is a time consuming and error-prone task, as there are various aspects in a river landscape that are difficult to manually detect. Therefore, the aim of this thesis is to study the problem of automated feature tracking in multi-channel rivers.

Problem statement 1. *Can we find algorithms for the automated tracking of features in multi-channel rivers?*

When studying the behaviour of complex, real-world objects, such as rivers, the first step

Figure 1.2: Orthoimagery of a the Western Scheldt, an estuary in the Netherlands (image from image archive of Rijkswaterstaat: https://beeldbank.rws.nl/).

is to properly define a problem statement and a desired output. To do that, there needs to be an understanding of the context in which the output will be used. For instance, if the goal is to study whether dredging causes downstream channels to silt up, the output should identify channels, store the depth of channels and maintain a spatial structure of the channel network. However, if the goal is to study the lifespan of individual channels, there is no need to maintain a spatial structure or store depths.

Given a formal definition of the output, the next step is to find a rigorous model of the problem, in terms of the input. For example, to study the lifespan of channels, there needs to be a rigorous model of the identity of a channel and a clear understanding of how to detect channels from the input. An important part in the modelling step, is to understand the natural processes and morphodynamical laws guiding the evolution of channels. The last step, is to use these models to design efficient algorithms to compute the desired output.

These steps—finding a problem statement, model and an algorithm—form an iterative modelling framework. To gain a better understanding of the context, and to identify sub-problems, a first iteration typically starts with a simplified problem statement. In our case, we start by studying the problem of *dynamic river networks*. A dynamic river network is an abstract representation of a time-varying *river network*. A river network is an embedded graph which captures the channel structure of a multi-channel river. Edges in a river network represent the channels in the river, and nodes represent bifurcations and confluences. An example of a manually drawn river network is shown in Figure 1.3. Existing algorithms for the automated construction of river networks depend on the available input data.

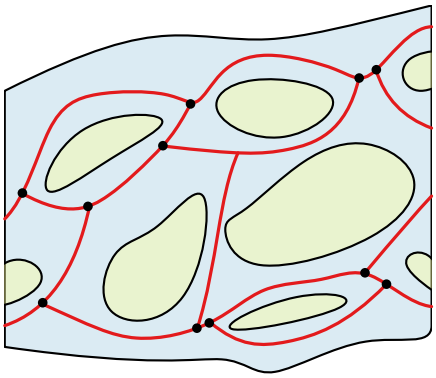


Figure 1.3: An example of a manually drawn river network.

There exist a number of methods to gather data on rivers. A relatively cheap way to gather data is to deploy an aerial survey like an unmanned aerial vehicle (UAV) or a satellite. This yields *orthoimagery*: coloured aerial photos of the terrain (see figure 1.2). Alternatively, bathymetry can be used to extract a *digital elevation model* (DEM) of the terrain. A DEM contains a height map of the terrain and can be rendered as a grey-scale image, where lighter areas indicate higher elevations.

Algorithms for computing river networks. If the input consists of orthoimagery, a common way to construct river networks is by converting the coloured photos into binary images indicating 'wet' and 'dry' surface [13, 14, 37, 59], which we refer to as the wet-and-dry approach. In terms of precision, the wet-and-dry approach has a few drawbacks. Firstly, identifying wet areas relies heavily on tuning parameters dealing with the quality of the photo and colours of the terrain. Secondly, for the analysis of rivers on the timescale of years, the dependence on the water level is a major disadvantage. If the photo was taken during a very dry period, the network may contain only a few 'wet' surfaces, that do not represent the actual river network well. Finally, most algorithms based on the wet-and-dry approach do not account for the shape of the terrain. This means that there is little to no information on the submerged terrain encoded in the resulting networks.

Extracting networks from DEMs is also a well-studied topic. Early algorithms were designed to compute *drainage networks* or *flows* [1, 5, 21, 48, 60]. These networks are typically used to analyse how a uniformly distributed input in the terrain, like rain, accumulates to local minima. Most existing algorithms rely on the assumption that flowing water follows the direction of steepest descent. As gravity attracts droplets of water in the direction of steepest descent, this seems like a reasonable assumption. However, there are several reasons why a channel does not necessarily follow the direction

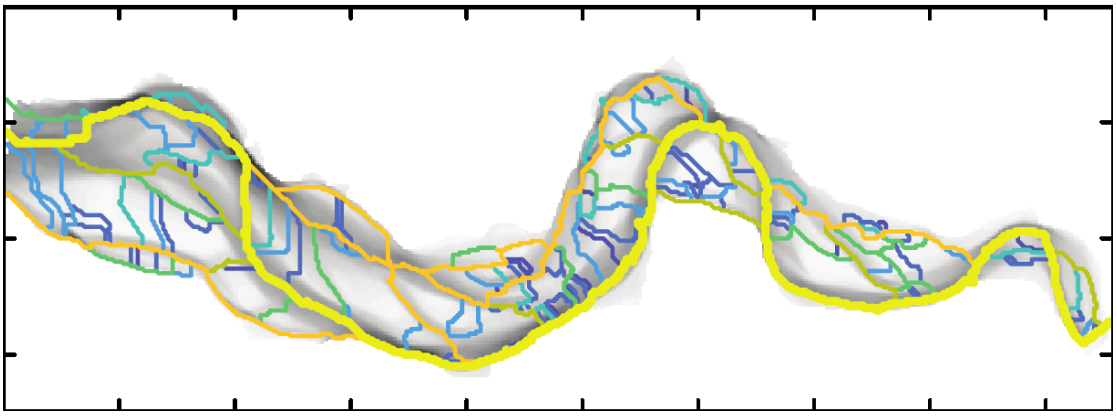


Figure 1.4: Example of a river network of the Western Scheldt based on the lowest paths in the terrain (image from [30]).

of steepest descent, and might even ascend. For instance, a river acts as a body of moving water, and with that motion comes inertia that might push water in a different direction. Moreover, local minima in the terrain may be filled with water, meaning that channels ascend locally.

Lowest paths. Recently, a new approach to developing river networks from DEMs was proposed, resulting in two new algorithms [36, 42]. The authors explicitly dropped the assumption that flowing water follows the direction of steepest descent. Their approach is based on assembling *lowest paths* in the terrain. Informally, a lowest path connects two points in such a way that its highest point is as low as possible. They show that these lowest paths lie on the *piecewise linear Morse-Smale complex* [22]. By computing a discrete Morse-Smale complex, they obtain a set of lowest paths. Among these lowest paths are many insignificant, small channels. To reduce the size of the resulting network, they only allow two channels to be represented in the same network if there is enough sediment in between the two. This approach resulted in a first algorithm [36]. A disadvantage of the resulting network is that it is rather unstable: small changes in the DEM could lead to completely different networks.

Because of the instability of the first method, they developed a second algorithm for computing river networks from DEMs, again using the concept of lowest paths [42]. The model starts by pruning features of the terrain based on *volume-persistence*. By cutting off parts of the terrain that have little volume, the resulting Morse-Smale complex consists of only significant channels. This second approach is more stable, as the decision to include channels in the network is taken at a local level. Figure 1.4 shows an example of a river network computed by this second algorithm.

The same authors also developed a *kinetic data structure* (KDS) [7] for maintaining an *area-persistent* terrain in one dimension. Unfortunately, extending this approach to maintaining volume-persistence in two dimensions seemed infeasible due to a high computational cost.

A more concrete problem statement. There are numerous advantages of using DEMs instead of orthoimagery, so in this thesis we assume the input data is given as DEMs. Among the mentioned algorithms for computing a river network from a DEM, the lowest path algorithms are most interesting to us, since they do not assume that rivers follow the direction of steepest descent. Therefore, we model channels as lowest paths in the terrain, and a river network as a graph. We tackle the problem of constructing a dynamic river network, by finding a suitable *channel matching*. A channel matching assigns to every channel in the initial network, a channel in the second network. We are now ready to state a more concrete problem statement, which will be the focus of this thesis:

Problem statement 2. *Given two DEMs of an evolving terrain, and two corresponding river networks, can we find algorithms for the automated construction of a dynamic river network?*

Contribution. In Chapter 2, we discuss the notion of a channel matching in more detail, and we propose two models for a dynamic river network. The two models tackle the problem of matching channels from a different perspective. In the first model, we see the channels as the primary object of interest, whereas in the second model, we focus more on the terrains. In the remainder of this chapter, we briefly introduce these two perspectives.

1.2 Similarity measures

A first way to look at the problem is from the channel's perspective. From a morphological point of view this makes sense, as the channels are the main acting party on the landscape. So, in this approach we focus on the movement of the channels, and trace them through time. To do that, we first compute river network for the two subsequent DEMs of the river. Next, we find a coherent matching of channels, based on how similar they are with respect to some *similarity measure*. The performance of such a method depends greatly on the choice of similarity measure.

From an abstract point of view, the channels can be seen as paths or curves on a surface. Similarity measures for paths and curves are an important topic of study in various fields. In computational geometry there exist a plethora of distance metrics [53], of which the Fréchet distance [3], the Wasserstein distance (also known as Earth Mover's distance) [45] and the Hausdorff distance [2] are the most studied examples. Another interesting

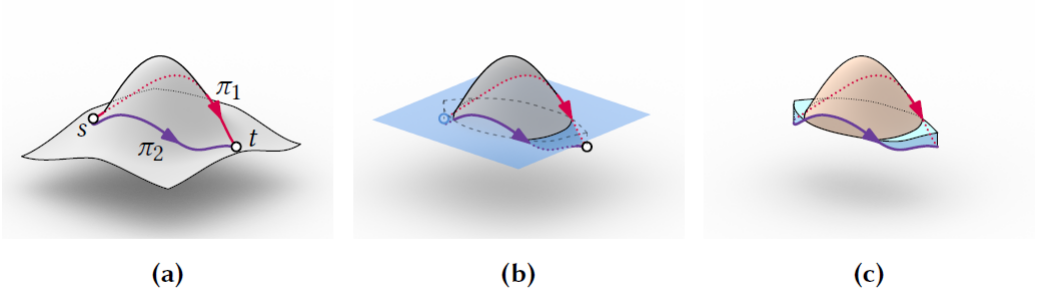


Figure 1.5: Execution of a volume-based similarity measure on two paths on a terrain (image from [51]).

class of measures uses the topology of the underlying space where the curves reside. Among these measures are the homotopic and isotopic Fréchet distances, by Chambers et al. [15, 16]. The same authors also constructed an algorithm that minimises the *homotopy area* [17], which models the area swept over when continuously deforming one curve into the other. These measures are all rather general in the sense that they are unaware of any context.

Context-aware similarity measures. A similarity measure that is designed for a specific context is usually not widely applicable. However, such measures often outperform the classical, general methods in their specific domain. Context-aware similarity measures have been developed for all kinds of problems. There exists a measure for comparing GPS trajectories while being aware of buildings [38], one for modelling lane changes while also taking into account the other traffic [29] and there is a group of context-aware similarity measures based on geographic context [12]. The latter one uses a subdivision of the surface to distinguish between different types of terrain (e.g. soil and water).

More interesting for our context is a set of six *volume-based similarity measures* [51], which take a 3-dimensional terrain into account. These measures construct a base surface—each one is different and defines a specific measure—which slices the terrain between the two paths. The part of the terrain above the base surface is considered the earth above the two paths, and the volume of that earth is taken as output. See Figure 1.5 for an example of the application of such a measure.

Contributions. In Chapter 3, we discuss similarity measures in more detail. In particular, we focus on extending the volume-based similarity measures. The six original proposed methods work only for paths that do not intersect, but do share endpoints. However, we want to apply a measure to more general paths, that may intersect and do

not necessarily share endpoints.

1.3 Displacement fields

An alternative to tracking channels is to focus on the transformation of the terrain. As a first step, we want to construct a morph of the initial terrain into the subsequent terrain. With such a morph, it is possible to track the sand on the terrain. We then match channels using the transformation. It might seem more intuitive to track the channels directly, since that is how nature acts as well. However, by considering the terrain first, we are exploiting all of the available data.

The more general problem of aligning images is the primal focus of *image registration* [27, 40], which is an active field of research in image analysis. Besides images, the research in this area also focuses on other sorts of data, such as DEMs or *digital terrain models* (DTMs) [44, 54]. Algorithms for data registration typically follow a similar framework. They first establish a similarity measure between two sets of data, to quantify how well they align after applying a transformation. Next, they consider a class of possible transformations. From this class, the transformation is chosen that optimises the similarity measure. Finally, the chosen transformation is used to morph the initial data set.

Registration algorithms are widely used in various fields, such as remote sensing and medical imaging [61]. In geographical context, image registration algorithms have been used to monitor the locations of agricultural terraces [56] or to track the evolution of active landslides [54]. Image registration techniques have also been utilised in river analysis, for instance to study the evolution of a single-channel river [52].

Displacement field. Due to the diversity of the problems, data registration algorithms can be classified by numerous characteristics. If, for instance, the data is obtained at distinct moments in time, the problem of aligning data is called *multi-temporal* data registration [20]. To represent the transformation, multi-temporal registration methods often use a *displacement field*, consisting of *displacement vectors*. A displacement vector assigns a direction and magnitude to each data point of the input, capturing the motion of that data point. A common technique to derive such a displacement field is *image correlation*. Image correlation is used to measure a correspondence of sub-images by moving these sub-images over the target image. This technique has for instance been used to derive a displacement field for evolving landslides [24].

Recently, Chadwick et al. used image correlation to track the motion of channels in braided rivers [14]. Inspired by pixel-based tracking algorithms as given in [59], they first converted their data into binary wet-and-dry images. Then, they used an image correlation technique called *particle image velocimetry* (PIV) to derive a displacement

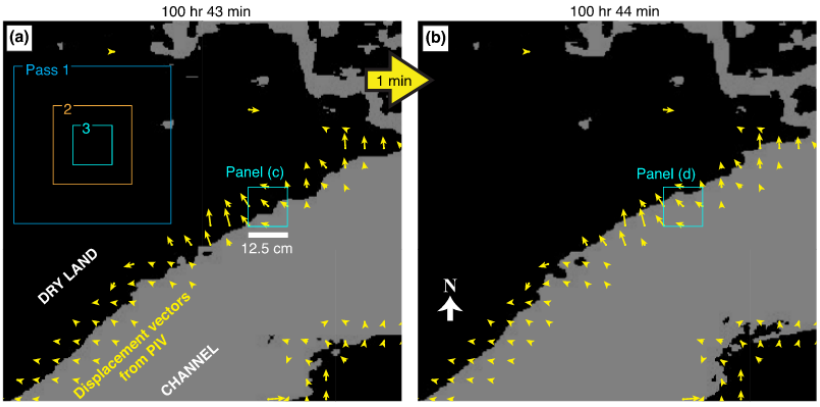


Figure 1.6: Displacement field derived by the PIV-based approach (image from [14]).

field (see Figure 1.6). They were able to use their algorithm to show that channel motion occurs sideways as well as bank-perpendicular, refuting a common assumption in braided river modelling.

Optimal transport. In our approach, we also compute a displacement field to model the transformation of the terrain. The PIV-based method is restrictive in the sense that they only use wet-and-dry pixels. To exploit all data, we consider a different method to compute a displacement field: *optimal transport* (OT). Optimal transport [46, 50, 55] is a branch of mathematics that originates from the 18th century and was first introduced by Gaspard Monge. The general problem that optimal transport tries to tackle is to find an *optimal transport map*: a cost-efficient way to redistribute mass from a given source to a given target, while preserving the total mass. Since the last century, the problem has been extensively studied, refined and reformulated, but the essence has stayed the same.

The problem of optimal transport is very versatile, as there are numerous reformulations of the problem, leading to the same optimal map. As a result, there are many interpretations of the problem, each suggesting the use of a different numerical algorithm. Finding fast algorithms to compute an optimal transport map is the main concern of *computational optimal transport* [43]. Algorithms to find an optimal transport map have been used in both image registration [25, 41] as well as in river analysis [10]. Moreover, OT-based algorithms have shown to be effective tools to warp images [28].

Contributions. In Chapter 4 we discuss the construction of a displacement field in more detail. Specifically, we look at the (basic) theory of optimal transport, and how to apply that to our case. Moreover, we explore possibilities for a suitable cost function, and discuss how to deal with the constraint of mass preservation.

1.4 Implementation

In the previous two sections, we introduced two different tools - similarity measures and displacement fields - to help in finding a realistic matching. In Chapter 5, we describe how to obtain channel matchings from these tools. Moreover, we describe the implementation of our methods, and show the performance on both simulated and artificial test cases. Finally, in Chapter 6, we conclude our findings and pose some interesting open problems for future research.

Chapter 2

Modelling dynamic river networks

The goal of this thesis is to compute a *dynamic river network*. Recall, in Section 1.1, we introduced the concept of a river network. A river network is an abstract representation of the channel structure in a river. We model dynamic river network as a *channel matching* that is optimal in some sense. Informally, a channel matching assigns to every channel in the initial river network, a channel in the second network. A possible objective for a dynamic river network could be to maximise the number of *correctly matched channels*. We say a channel is correctly matched, if it in reality evolved into its matched channel. Unfortunately, we do not know how to automate the process of checking whether a channel is correctly matched. Therefore, we look for a relaxed objective.

The problem of computing dynamic river networks is two-fold. Firstly, the notion of a channel matching needs to be formalised. Secondly, we want to find a mathematically rigorous objective, that is optimised to obtain a dynamic river network. In the remainder of this chapter, we discuss these problems in more detail. In the following paragraphs, we first discuss how to model terrains, channels, and river networks. Then in Section 2.1 we consider possible definitions for a channel matching. Lastly, in Section 2.2 we propose two models for a dynamic river network.

Modelling terrains. Given two DEMs of an evolving terrain, we refer to the terrain at the first time step as the *source terrain*, and to the terrain at the second time step as the *target terrain*. Similarly, objects on the source (target) terrain are referred to as source (target) objects. Essentially, a DEM stores the elevation of the terrain at a finite number of points. We assume that a DEM is represented as a *triangular irregular network* (TIN). A TIN is a triangulation T of a compact surface $\Sigma \subset \mathbb{R}^2$, that stores for each point a height value. By linear interpolation of these height values over the edges and triangles of T, we obtain a *height function* $h\colon \Sigma \to \mathbb{R}$.

Following the definitions of Kleinhans et al. [36], we model channels as *lowest paths*, and a river network as an embedded graph on the surface Σ. A path $\pi\colon [0,1] \to \Sigma$ is a continuous function connecting a start point s to an endpoint t. On a TIN, a path is stored as an ordered set of points $\{s, v_1, \ldots, v_n, t\} \subset \Sigma$. For two points s and t, let ℓ be the lowest possible height, such that any path from s to t reaches at least the height ℓ with respect to h. A path π from s to t is then called a lowest path, if its highest point is at most at height ℓ, and it minimises the length of the sub-path that lies at that height.

A river network is an *embedded graph* on the surface Σ. A graph embedding of a graph $G = (V, E)$ on a surface Σ, associates with each edge in G a path on Σ. Given a height function h, a river network is then a graph G, such that it associates with each edge, a lowest path on Σ with respect to h. Formally, this means that there exists a projection $P\colon E \to \mathcal{P}(\Sigma)$, which maps an edge in the graph to a path on the surface. Here, $\mathcal{P}(\Sigma)$ is the set of paths on Σ. For simplicity, in the remainder of this thesis we refer to edges in E as channels on Σ, but whenever we do, we implicitly use the projection P.

2.1 Channel matchings

The first problem in defining dynamic river networks, is to formalise the notion of a channel matching. Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be two river networks, extracted from the source and target DEM respectively. A natural choice for a channel matching is to consider functions $M\colon E_1 \to E_2$. However, there are a couple of limitations to this definition. For instance, channels can bifurcate, and as a result, a single channel in the source network, might correspond to two different channels in the target network. Moreover, a channel might silt up, meaning it disappears entirely from the river network. By choosing M as a function from E_1 to E_2, we can not capture these transformations.

As a solution, there are other ways to define a channel matching. For instance, to deal with bifurcating channels, one could consider functions $M\colon E_1 \to 2^{E_2}$, assigning to each channel in the source network, a set of channels in the target network. Similarly, to account for perishing channels, one could extend the range of the channel matching with a sink channel τ, and consider functions $M\colon E_1 \to E_2 \cup \{\tau\}$. If a channel is matched to the sink channel, this means that it no longer exists in the target network. Note that it is not necessary to include a *source* channel to account for appearing channels in the target network. An appearing channel should naturally not be matched to any of the channels in the source network.

Question 1. *What is a suitable range for a channel matching, capturing all possible transformations of a channel?*

Figure 2.1: Two types of motion of a river channel. Channel migration is depicted in green, and channel avulsion in red.

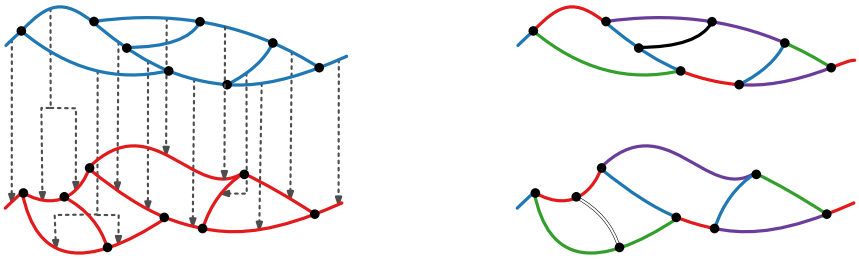
Channel operations. A possible next extension, would be to include more information on what exactly happened during the transformation. Suppose we consider a class of functions $M\colon E_1 \to \mathcal{R}$, where $\mathcal{R}$ is a range of choice. There are various different morphodynamical changes that can happen to a channel in a multi-channel river. One could construct a class $\mathcal{O}$ of possible *operations*, capturing the type of morphodynamical change. A matching $M\colon E_1 \to \mathcal{R} \times \mathcal{O}$ then maps each channel to one or more channels in the target network and a corresponding operation. A simple example of such a class is $\mathcal{O} = \{\text{persist}, \text{perish}\}$.

A more interesting class of operations would be to consider the 10 different mechanisms presented by Wheaton et al. [57]. As mentioned in Chapter 1, these mechanisms describe the possible transformations of channels in a braided river. Four of these mechanisms—central bar development, chute cutoff of point bars, transverse bar conversion and lobe dissections—stem from earlier work by Ashmore [6] and Ferguson [23]. To detect these changes, Wheaton et al. developed *Geomorphic Change Detection* [58] software.

A last example of a class of operations is to distinguish between *channel migration* and *channel avulsion*, following the definitions in [14] (see Figure 2.1). Channel migration describes a gradual and slow type of motion of channels, which can for instance be caused by riverbed erosion. Channel avulsion on the other hand, describes a more abrupt type of motion, often causing a sudden change in the network structure. Since channel avulsion makes it harder to track channels, a good pre-processing step could be to first detect areas where channel avulsion took place.

Question 2. *What would be a good set of channel operations, and how do we detect them?*

Visualising channel matchings. To inspect the output of automated methods, a visual representation of a channel matching is useful. There are various ways to visualise a channel matching. Perhaps the most intuitive choice, is to draw arrows between matched channels (see Figure 2.2a). However, in large networks, this might cause a lot of clutter, making it hard to carefully inspect the output.



(a) (b)

Figure 2.2: Two possible visualisations for a channel matching.

Another option is to consider a colour-based visualisation, where matched channels share the same colour (see Figure 2.2b). To indicate channels that perish or emerge, one could reserve two distinct colours, like black and white. A disadvantage of this method, is that there need to be a lot of colours if there are many channels. One could consider to limit the number of different colours, by deploying a four-colour algorithm. However, this might cause issues when channels split or merge.

Alternatively, one could also consider representing a channel matching as a series of interpolated matchings. To bring the channels closer to each other, it is also possible to overlay the networks. A final option is to combine some of these methods, to use the best of multiple worlds. For instance, colouring the networks in two distinct colours, then overlaying them and drawing arrows between matched channels might provide a nice visualisation .

Question 3. *How do we visualise a channel matching as clearly as possible.*

2.2 A suitable objective

Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be two river networks, and let $M \colon E_1 \to \mathcal{R}$ be a channel matching. For now, we assume that the channel matching maps each source channel to a single target channel, i.e. $\mathcal{R} = E_2$. Recall, a dynamic river network is a channel matching that optimises some objective. The goal of this section, is to find a suitable objective, which is both mathematically rigorous and reasonable from a morphological point of view. Ideally, a dynamic river network maximises the number of correctly matched channels, but unfortunately we do not know how to check automatically whether a channel is matched correctly. It is however, desirable to have a measure on the quality of the resulting dynamic river network.

Question 4. *Can we find an automated way to determine the quality of a generated dynamic river network, in an objective manner.*

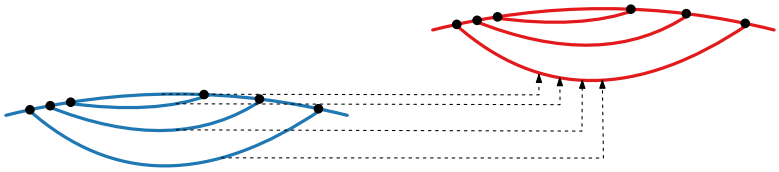


Figure 2.3: For a similarity measure capturing the area between the paths, all source channels will be matched to the same target channel.

2.2.1 Maximising similarity

A natural way to relax the notion of a correctly matched channel, is to consider *optimally matched channels*. Given a *similarity measure* $d\colon \mathcal{P}(\Sigma) \times \mathcal{P}(\Sigma) \to [0, \infty)$ and a channel $\pi \in E_1$, we define a dynamic river network:

$$M(\pi) := \rho^* \in E_2, \text{ such that } d(\pi, \rho^*) = \min_{\rho \in E_2} d(\pi, \rho).$$

We assume that d is surjective, so that there is precisely one channel $\rho \in E_2$ that minimises the similarity measure. If d is not surjective, one could consider using a selection criteria to choose one of the optimal target channels. In this model, every source channel is matched to the best possible target channel, with respect to a chosen similarity measure. Therefore, the similarity measure plays an important role in this model.

Problem statement 3. *How do we define a suitable similarity measure, to compare channels from an evolving terrain?*

This problem statement is the central focus of Chapter 3. Note that in this construction, we consider each channel in the source network separately. As a consequence, the resulting channel matching might not be globally coherent. For instance, all source channels might be matched to a single target channel (see Figure 2.3). A possible solution for this is to impose constraints on the channel matching. For instance, to prevent that all source channels are matched to a single target channel, we could enforce that at most two source channels are matched to the same target channel. An added advantage of adding constraints, is that a constraint can implement morphologically relevant behaviour.

Question 5. *What is a good set of constraints to impose on a channel matching?*

A simple way to extend this model in the case that the range of M is chosen differently, is to consider *thresholds*. For instance, if the range includes a sink channel, i.e. $\mathcal{R} = E_1 \cup \{\tau\}$, one could define a threshold d_τ. If then, for a given source channel π, d_τ is smaller then the distance of π to the closest target channel, π is matched to τ. Similarly, if a single source channel is mapped to a set of target channels, one could consider a

threshold d_M and match π to all channels that are within a distance d_M. Choosing right values for these threshold depends on the choice of similarity measure.

Question 6. *Given a similarity measure, how do we determine the right values for thresholds?*

2.2.2 Minimising displacement

Before we describe a second model for dynamic river networks, we first introduce some notation. Recall, we define a channel to be a path $\pi\colon [0,1] \to \Sigma$ on a surface Σ, and we denote the set of paths on Σ, as $\mathcal{P}(\Sigma)$. Now, we say a time-varying path η, is a continuous function, that maps a time $t \in [0,1]$ to a path in $\mathcal{P}(\Sigma)$. In other words, η is a path, in the space of paths: $\eta \in \mathcal{P}(\mathcal{P}(\Sigma))$. We say that the source terrain is obtained at time $t = 0$, and the target terrain is obtained at time $t = 1$. If we assume that every channel π in E_1 has a unique, corresponding time-varying channel η^π, then we know that the correct channel matching M^* is given by

$$M^*(\pi) = \eta^\pi(1).$$

In our second model, we take this assumption as a starting point for defining an optimal channel matching. We extend the concept of a channel, to that of a time-varying channel, and model such time-varying channels as time-varying paths. A time-varying channel captures the, possibly non-continuous, motion of a channel over time, as it changes its course. A natural way to model a time-varying channel, is by extending the notion of a lowest path to a time-varying variant, which we will call a *persistent lowest path*. Without explicitly defining these paths, let $\Gamma(\Sigma)$ be the set of persistent lowest path on Σ.

To define a persistent lowest path, we need to choose which properties such paths should exhibit. For instance, an intuitive definition would be to consider time-varying paths η, such that at every time t, $\eta(t)$ is a lowest path with respect to the terrain at time t. Now, let $E\colon \mathcal{P}(\Sigma) \to \Gamma(\Sigma)$ denote the function that maps a lowest path π to its corresponding persistent lowest path η^π, such that $\eta^\pi(0) = \pi$. We assume that such a function E is well-defined, but note that that is not necessarily the case. Depending on the definition for a persistent lowest path, there might not be a persistent lowest path that satisfies $\eta^\pi(0) = \pi$. If there is a corresponding persistent lowest path, then it is not guaranteed that it is unique.

Question 7. *How do we (rigorously) define a persistent lowest path?*

Since the definition of a lowest path depends on the height of the terrain, we assume that the same holds for a persistent lowest path. We know only the height of the terrain at times $t = 0$ and $t = 1$, so instead of considering the height, we consider a *displacement*

function $D\colon [0,\infty)\times\Sigma \to \Sigma$. A displacement function captures the transformation of the terrain, and is related to the time-varying height of the terrain. Therefore, we assume that we can express E in terms of D:

$$E(\pi, t) = L(D(t), \pi),$$

for some function L.

Question 8. *What is the relation between a persistent lowest path and the displacement function D, i.e. how do we define the function L?*

Finding the displacement function at time $t = 1$ will be the central focus of Chapter 4.

Problem statement 4. *How do we obtain the displacement field $D(1)$?*

We now have a way to map a path π on the source terrain, to another path ρ on the target terrain. Specifically, we have $\rho := E(\pi, 1) = L(D(1), \pi)$. Since the involved functions—L and D—model very complex processes, there is little hope of finding exact solutions. Therefore, the path ρ is very likely not a channel in the target river network. To make sure that we do match source channels to target channels, we need to snap ρ to the target network. In particular, we introduce a function $r\colon \mathcal{P}(\Sigma) \to E_2$, that assigns a channel in E_2 to the obtained path ρ.

Question 9. *How do we snap paths to the target network, i.e. how do we define r?*

At last, we say that the dynamic river network is given by the channel matching

$$M(\pi) := r(L(D(1), \pi))$$

Finding a suitable model for L and r is the main focus of Chapter 5.

2.2.3 Alternatives and extensions

Recall that the river network is stored as a graph, with edges representing channels in the network. An alternative to the two models proposed in this chapter, is to consider a model based on the structure of the graph. That is, the edges and nodes in a graph naturally provide a spatial relation between channels in the river network. A possible way to define a channel matching, is such that it maintains these spatial relations. Besides an alternative model, one could also look to extend our current definitions, by including more properties of the river network.

Widths and depths. In braided rivers, the depth of a channel plays an important role in the dynamics of the river. For instance, the process of sediment transport is dependent on the flow velocity of the water. In places where the velocity is low, sediment settles

at the bottom. As a result, the channel becomes less deep, causing the velocity to drop, and amplifying the settling of sediment. Therefore, tracking the widths and depths of channels might provide more insight in the morphodynamical behaviour, making it easier to define a suitable matching.

Question 10. *How do we define and track properties, like width and depth, of a channel?*

The time step. So far, we have not yet mentioned a crucial component of the initial problem statement: the time step. Note that time is not necessarily a relevant parameter to the problem, as some rivers evolve slowly, while others evolve more rapidly. Therefore, a better parameter would be to consider a morphological difference of the input terrain. Alternatively, one could also consider a topological difference in the river networks extracted from the input terrains.

Question 11. *How do we measure the similarity of the input terrains?*

The models proposed in this chapter are independent of any similarity of the terrains. However, we do expect that the more alike two networks are, the better our methods work. This is a natural consequence, and has been observed in related problems as well. For instance, in [14], the authors note that their methods to track channel migration work best, if channels migrate approximately 25% of their channel widths.

Question 12. *Can we quantify how well our methods perform in terms of the similarity of the input terrains?*

2.3 Discussion.

This chapter poses a lot of questions, of which many are inherently subjective. For instance, finding a good model for a channel matching relies heavily on the—almost philosophical—question of what a time-varying channel is. In modelling, it is always important to keep the application in mind. So, when it comes to modelling complex rivers, it is wise to keep a geomorphological point of view.

There is usually a trade-off between adding complexity to a model and obtaining more accurate results. In the remainder of this thesis, we simplify the problem rather bluntly, by adding a lot of assumptions. As a consequence, the results of our models might not be very accurate. However, they might provide more insight to the problem's context, and show some promising direction, which may lead to better results in a next iteration of the modelling proces.

Chapter 3

Comparing channels

In this chapter we explore the potential of a volume-based similarity measure, that compares paths obtained at different moments on an evolving terrain. Tools to analyse geometric data often use a similarity measure to compare paths, but there is not a single, best way to do so. In our case, we use paths to represent river channels, but they can also be used to represent various other geographic objects. For instance, paths can be used to delineate fields growing crops or to represent trajectories of migrating birds. In both examples, the paths are dynamic and can change over time. Crop fields may merge and birds might fly a different route next year. The same goes for rivers: channels change their course as they evolve over time.

Similarity measures. To quantify the similarity of two paths, we need a substantial notion of similarity. If the problem is set in a specific context, this context needs to be taken into account. As a result, numerous similarity measures have been designed, tailored to a variety of contexts. For example, when comparing trajectories of migrating birds it is important to take the air's temperature or even wind patterns into account. Likewise, comparing river channels requires an understanding of the processes that cause a river to change its course and an analysis of how an evolving river affects the terrain.

Most existing similarity measures are some variant on measuring what 'lies between' two paths. As we have seen in Section 1.2, some of the more general measures such as the Fréchet distance and the Hausdorff distance do not consider any context at all. Other measures compute the area between two paths and some measures are specifically designed for a geometric use case. In our 3-dimensional setting, we are most interested in the volume-based similarity measures proposed by Sonke et al. [51]. These measures first construct a suitable base surface between the two paths, which may both slice through and hover above the terrain. The volume of earth above the base surface and below the terrain is considered the volume of earth between the two input paths. An example of such a measure, with as base surface a simple horizontal plane, can be seen in

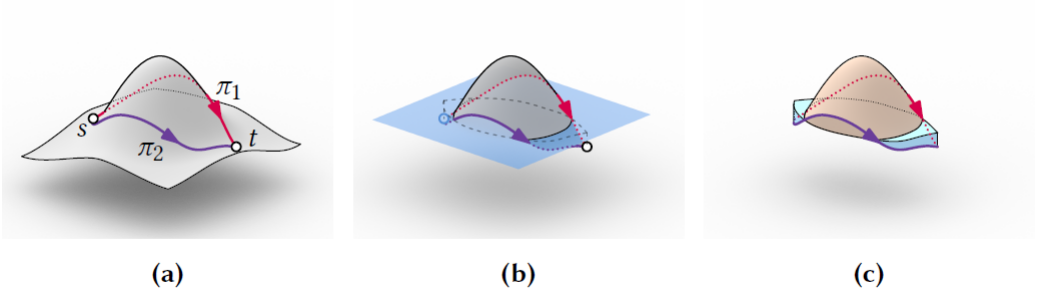


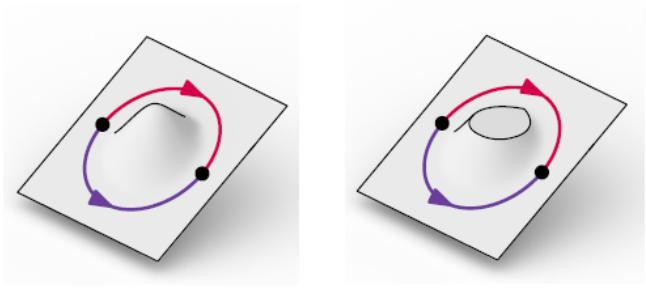
Figure 3.1: Execution of a volume-based similarity measure on two paths on a terrain (image from [51]).

Figure 3.1. Sonke et al. propose six different ways to construct the base surface, leading to six different measures with their own applications.

Using the volume of sediment between two channels is a reasonable way to capture the effort needed for a channel to move across the terrain. We want to exploit this idea in pursuit of a similarity measure for our use case. Specifically, we wish to compare paths obtained at different moments in time on an evolving terrain. The volume-based similarity measures assume that the input paths share endpoints and are disjoint elsewhere. Moreover, the measures use a single height map, arising from a single moment in time, in the construction of base surface and computation of the volume. To apply such a measure, we thus need to pre-process our input paths and find a single height map to use.

Assumptions and organisation. We assume that our input consists of a triangulation T of a surface $\Sigma \subset \mathbb{R}^2$, where each vertex is assigned two height values. Let h_1 and h_2 be the height functions obtained by linearly interpolating these height values over the edges and triangles of T. Moreover, we are given two path π, ρ on Σ. We assume that both paths are *simple*, i.e. that they do not intersect themselves. This is a safe assumption, since by construction of a river network, our paths are separated when they split or join. Our goal is to compute a similarity between π and ρ, capturing the effort needed for the path π to change its course to path ρ, with respect to the terrain. For simplicity, we consider only channel migration from π to ρ and do not account for channel avulsion.

In the remainder of this chapter we discuss the possibilities of a volume-based similarity measure between π and ρ. In Section 3.1 we first discuss the existing volume-based similarity measures in more detail, and we choose a suitable base surface for our application. Then, in sections 3.2 and 3.3 we discuss two problems that arise when dealing with more general paths: intersections and disjoint endpoints. Next, in Section 3.4 we discuss the



(a) **(b)**

Figure 3.2: Illustration of the working of the WFS (images from [51]). **(a)** a sloped hill with a bump. **(b)** The WFS slices off the part above the saddle point.

time-varying component of our problem. Finally, in Section 3.5 we discuss the challenges in designing a suitable similarity measure.

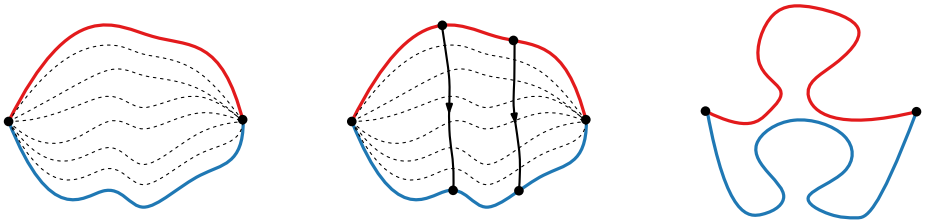
3.1 Sand function

In this section, we discuss the volume-based similarity measures proposed by Sonke et al. [51]. Formally, let T be a triangulation of a surface Σ, where each vertex has a height value. Let $h\colon \mathbb{R}^2 \to \mathbb{R}$ be a height function obtained when linearly interpolating the height values over the edges and triangles of T. We are given two paths π and ρ on Σ and we assume that the paths intersect exactly twice, namely at their endpoints. The two paths enclose a single region $D \subset \Sigma$. A volume-based similarity measure is defined by its base surface, which acts as a function $\mathcal{B}\colon \mathbb{R}^2 \to \mathbb{R}$. The measure induced by a specific choice of base surface, is then given by

$$d_{\text{earth}}(\pi, \rho) = \int_D \max\{0, h(x, y) - \mathcal{B}(x, y)\}\, \mathrm{d}x\mathrm{d}y. \tag{3.1}$$

This measure captures the volume of earth above the base surface $\mathcal{B}$. A high value means that there is a large volume 'in between' the two paths, and a low value means that there is little volume in between.

Water flow surface. Out of the six existing base surfaces, the *water flow surface* (WFS) is the most suitable choice for our application. The WFS models the minimum amount of earth that needs to be removed for π to move its course to ρ, with respect to the height function h. From now on, we refer to the volume-based similarity measure using the WFS as its base surface, as the *sand function*. Note that the sand function is asymmetric, as it is harder for a channel to move uphill than it is to move downhill. Informally, the WFS is a surface on or below h, on which there exists a smooth morph from π to ρ in D



(a) **(b)** **(c)**

Figure 3.3: **(a)** A monotone isotopy between two paths, depicted as a sequence of intermediate curves. **(b)** Two matching curves of the monotone isotopy. **(c)** An isotopy would have to squeeze through the bottleneck.

that never moves uphill (see Figure 3.2). Since there may be many such surfaces, the WFS is the one that measures the smallest volume of earth.

To formally define the WFS, we first introduce the necessary terminology. Let π' and ρ' be two non-intersecting paths in a plane that share endpoints and let D' denote the region enclosed by the two paths. An *isotopy* between π' and ρ' is a smooth transition from π' to ρ' with only simple intermediate paths. An isotopy is called *monotone* if it only moves forward and stays inside D' (see Figure 3.3a). Such an isotopy can be seen as a matching of the points on π' to the points of ρ'. If we fix a point on π', we can trace it along the intermediate curves to its matched point on ρ'. The sequence of points on the intermediate curves is called a *matching curve* (see Figure 3.3b).

We are now ready to define the WFS. A monotone isotopy between π and ρ is smooth, and forward only, so it defines a function over D. By choosing the elevations such that the function is continuous over D, the image of the monotone isotopy is a surface. Since we do not want to add sand, we restrict the elevations to be at most as high as h. Moreover, we want the matching curves of the monotone isotopy to be monotonically decreasing in elevation. This yields a set of valid surfaces, and the WFS is the surface from this set that minimises Equation (3.1).

The WFS models the cheapest way to morph one path into another, where cheapest refers to encountering the least volume of earth. It assumes that this morph stays inside the enclosed region D, but it is not immediately clear whether that assumption is reasonable. When only considering the area swept by the morph, it is clear that leaving the enclosed region yields a larger area. However, it is not trivial to assume that this also means that a larger amount of volume needs to be sliced off. Is it for instance possible that part of π finds a 'cheaper' path towards its matched part of ρ by leaving D?

(a) **(b)**

Figure 3.4: Two pairs of paths with different orders of intersection.

Question 13. *Suppose $\mathcal{B}$ is the surface obtained by taking the surface on or below h, for which a monotone homotopy exists with monotonically decreasing matching curves and which measured the smallest volume of earth. Does there then also exists a monotone isotopy whose matching curves are monotonically decreasing corresponding to $\mathcal{B}$?*

Following up on this, we should ask ourselves whether the WFS is a realistic model for the transition of a channel. For most moving paths, it sounds reasonable to assume that a channel finds the cheapest way to smoothly morph into another channel. However, for some paths, it might make more sense if it takes a detour. In particular, if two channels share a similar bend, it might be more likely that the bend was carried over, thereby leaving the enclosed region (see Figure 3.3c). One possible way to take these cases into account is by considering a new base surface, by either removing constraints (allowing for homotopies instead of isotopies) or by adding constraints (restraining paths from disentangling).

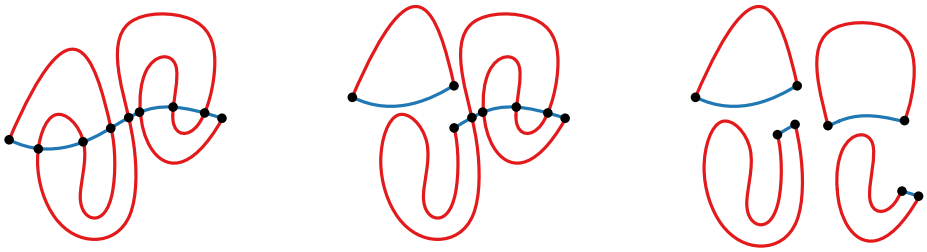
Question 14. *When do we consider two paths to be close to each other? Is there a more realistic notion than the one proposed in the construction of the water flow surface? And if so, can we construct a base surface capturing this new notion of similarity?*

3.2 Dealing with intersections

In this section we discuss how to deal with intersecting paths. The most intuitive way to deal with intersections is to split an invalid path into smaller valid sub-paths. Specifically, we want to partition both input paths π and ρ into n sub-paths $\pi_1, \ldots, \pi_n$ and $\rho_1, \ldots, \rho_n$, such that each pair (π_i, ρ_i) is non intersecting. Given these sub-paths, we can then evaluate the sand function for each pair, and sum the resulting values:

$$d'_{\text{earth}}(\pi, \rho) = \sum_{i=1}^{n} d_{\text{earth}}(\pi_i, \rho_i).$$

Naturally, this means we are computing the volume of two piles of sand as the sum of the volumes of the piles separately.



(a) (b) (c)

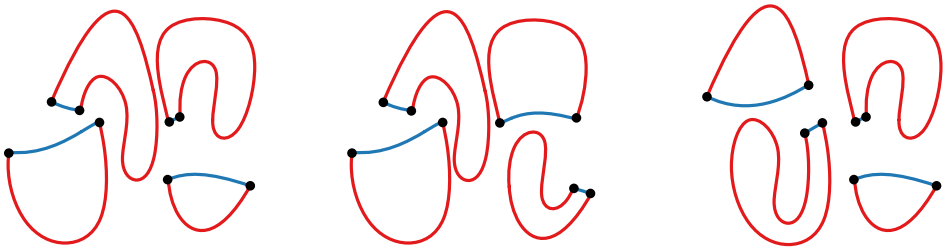
Figure 3.5: Execution of intersection partitioning.

The most straightforward way to partition the paths, is to split both paths at their intersections. Yet, for some pairs of paths that does not necessarily yield well-defined pairs of sub-paths. If the *order* of intersections is the same for both paths, there is no problem in partitioning at the intersections. We call such a pair of paths an *orderly* pair of paths (see Figure 3.4a). If the order is not the same for both paths, partitioning the paths along their intersections might give sub-paths that do no share their endpoints. We call such a pair *disorderly* (see Figure 3.4b).

Before we discuss how to deal with disorderly pairs, we introduce some useful notation and definitions. We are given two simple paths π and ρ on a plane. For now, we assume that the paths share their endpoints, but in the next section we discuss how to deal with paths that do not share endpoints. We denote by s and t the (shared) start and endpoint of π and ρ. A *sub-path* of a path π is a path contained in π. *Partitioning* a path π at a point x gives two sub-paths: one starting at s and ending at x, and one starting at x and ending at t. Furthermore, a pair of paths is *valid* if the paths share endpoints and do not intersect. Let $I = \{x_1, \ldots, x_n\}$ be the set of intersection points between π and ρ, sorted by their order along π. Note that $n \geq 2$, since the paths share endpoints. We now discuss how to obtain valid pairs of paths consisting of sub-paths of π and ρ.

Partition along intersections. The simplest idea is to partition two paths along their intersection points. For orderly pairs of paths it is easy to see how to create valid pairs on which we can compute the sand function. For disorderly pairs however, there is no straightforward way to obtain valid pairs. Therefore, we need to alter this method slightly to also work for disorderly pairs.

Assume that $n > 2$, since otherwise the two paths already form a valid pair. Formally, we partition π at x_2, to get two sub-paths π_1 and π_2. Note that ρ does not intersect π_1, except at the endpoints of π_1. Therefore, we also split ρ at x_2 to get two sub-paths ρ_1 and ρ_2. The sub-paths π_1 and ρ_1 do not intersect, but they do share endpoints. Hence, they form a first valid pair. The remaining two sub-paths π_2 and ρ_2 also share endpoints,



(a) (b) (c)

Figure 3.6: Three different sets of valid pairs, obtained by intersection partitioning.

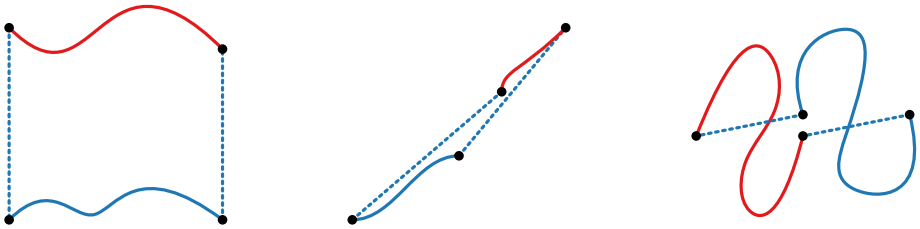
so we can repeat this process until we end at t. Figure 3.5 shows the execution of this method on a disorderly pair of paths.

To illustrate this method, consider the resulting pairs in Figure 3.5c. By computing the sand function on those pairs separately, we match sub-paths of π and ρ and prescribe how a morph from π to ρ would take place. This method of partitioning is a natural extension to disorderly pairs, for a number of reasons. Firstly, note that for an orderly pair of paths, this method comes down to simply partitioning at all intersections. Secondly, every part of π is matched to some part of ρ, meaning that we model the morph of the entire path. Lastly, start points in the sub-paths of π are matched to start point in the sub-paths of ρ. As a result, the direction of the flow in a channel is preserved.

If we assume that a path is not allowed to cross itself, the resulting pairs induce an order in which the sub-paths should transform. For instance, the bottom right red sub-path should have moved at least a bit before the red sub-path of the top right pair is able to finish its transition to its corresponding blue sub-path. Otherwise, the initial red path intersects itself. This could be taken into account when computing the sand function for these sub-paths: instead of using the same terrain for each pair, one could use a slightly different terrain, based on the transformations that already took place.

Question 15. *Can we update the terrain in between computing the sand function for sub-paths?*

Note, that the partitions change if we sort the intersections along ρ instead of π. In fact, at every iteration of the procedure, one could choose to partition at the next intersection along π, or along ρ. If these intersections are not the same, choosing a different one leads to a different set of sub-paths (see Figure 3.6). An automated method, that is independent of sorting along π or ρ, is to recursively partition at the *earliest* intersection. For the next intersection x along π, there is a parameter $t \in [0, 1]$ such that $\pi(t) = x$. Similarly, for the next intersection y along ρ, there is a parameter $t' \in [0, 1]$ such that $\rho(t) = y$. If $t < t'$, the earliest intersection is x, and otherwise it is y. Another,



(a) (b) (c)

Figure 3.7: Three examples of connecting endpoints by a shortest path.

independent, way to partition, is to greedily choose the intersection that yields the lowest sand function. A final method, would be to consider choosing the partitioning that minimises the total sand function.

Question 16. *How do we choose an order along which to partition?*

Alternatives. There are other ways to construct valid pairs of sub-paths. For instance, one could consider partitioning at all intersections. This yields smaller enclosed regions, and thus also cheaper transformations. However, this does not necessarily give a realistic matching of sub-paths. The sub-paths are no longer connected continuously, nor is the direction of flow preserved. Alternatively, one could consider using the outline of paths. However, this means we construct new paths, consisting of sub-paths from both π and ρ. If we are dealing with a single, closed path, this might be useful, but in our setting that does not make sense.

3.3 Dealing with endpoints

In this section, we briefly discuss the difficulties that arise when generalising a similarity measure to paths that do not share endpoints. If two paths π and ρ on a plane are not connected, there is no enclosed region, which is a crucial component of computing the water flow surface. The simplest way to obtain an enclosed region is by extending π such that its endpoints coincide with the endpoints of ρ. We denote the extended path by π'. Since π' and ρ do share endpoints, we are able to apply the sand function and compute a volume.

There are infinitely many ways to extend π, but the most natural choice is by drawing shortest paths between the endpoints of π and the endpoints of ρ. For some input paths, this might be a reasonable choice. If, for instance, the paths are more or less parallel, drawing shortest paths makes sense (see Figure 3.7a). For other input paths, it is not immediately clear whether using shortest paths gives a reasonable result.

For instance, in some cases, the enclosed region is thin and small (see Figure 3.7b). The sand function then might consider these paths to be relatively close, while in reality such a transformation is unexpected. Since the WFS chooses the morph between paths that requires removing the least volume of earth, it does not necessarily use realistic morphs. In fact, these cases resemble the issue of channels with bends from Figure 3.3c. One way to tackle this problem, is by considering a different base surface.

In other cases, the extended path might be intersecting itself (see Figure 3.7c). Since we assumed that our input paths are simple, this leads to a new kind of problem, that is outside the scope of this thesis. This problem does not stand on its own, and is due to an inherent defect in connecting two paths. By extending π, we introduce a new 'piece' of channel. As a result, we obtain a different path and thus a new problem, namely that of morphing the extended path π' to ρ. Therefore, extending paths might not be the right solution.

An alternative would be to reconsider the choice of base function. If morphs are allowed to move outside the enclosed region, we might not even need to connect the paths. Moreover, it is wise to add some restrictions on which paths we wish to compare. If paths are for instance very different in size, are oriented differently, or are separated by other paths, it is not likely that they should be matched. A potential option to deal with paths that do not share endpoints, would therefore be to combine a filtering step with an improved base surface.

3.4 Dealing with an evolving terrain

In this section, we discuss paths that are obtained at different moments on an evolving terrain. We are now given a triangulation T equipped with two height values per vertex. Let h_1 and h_2 be the two piece-wise linear height functions, obtained by interpolating the height values. The sand function computes a volume between two paths, and for the concept of volume to make sense, we need to have one well-defined height map. The simplest option is to discard one of the input height maps, but one should be careful when ignoring data. Therefore, we explore the possibilities for, and interpretations of choosing a single height map h.

Finding a height map. The input to our problem contains two height maps h_1 and h_2, describing the elevation of the terrain at different moments. The simplest way to obtain one height map is to discard one of the input height maps and use the other. Recall that the sand function models the minimum amount of earth that needs to be removed for a channel to move from π to ρ. By choosing $h = h_1$, we thus measure an expected effort of moving π to ρ with respect to the source terrain. Alternatively, we can also choose $h = h_2$. Instead of an expected effort, the sand function now gives an indication on how

likely it is that π indeed moved to ρ.

Both options give a reasonable estimate for our desired similarity measure, but they do have some shortcomings. One shortcoming is that they use the information of the terrain at a single moment in time, whereas there is more information available. For instance, if the sand function is high on the source terrain, but low on the target terrain, this means that the area between the two paths is affected over time, possibly due to the motion of π to ρ. Therefore, a better approach would be to compute the sand function for both $h = h_1$ as well as $h = h_2$, and compare the resulting volumes. A large decrease in the volumes could indicate that something—possibly the motion of a channel—reshaped the terrain. Similarly, a small difference in volume could indicate that the terrain remained rather intact, not experiencing any major changes.

Although informative, a difference in volume does not give a definite answer to what actually happened. A large difference could also be due to another disruption in the terrain between the two paths. Likewise, a small difference does not mean that the terrain did not change. For instance, sediment could have shifted, therefore not causing a change in height between two paths. In other words, we do not know anything about where the terrain was changed. One could consider other height maps, such as the average height of the two terrains: $h = (h_1 + h_2)/2$. Another option, is to consider an absolute difference of the terrains: $h = |h_2 - h_1|$. Still, these height maps exhibit a similar shortcoming: they do not give any information on the direction of motion.

Recovering transformation. We are given two height maps, and we know for a fact that there exists a transformation between the two. If we are able to recover this transformation, we can reconstruct the evolution of the terrain and infer how channels moved. Finding such a displacement is largely independent of finding a suitable similarity measure. Therefore, we introduced a second model for a dynamic river network in Section 2.2.2, depending on a *displacement field*. Finding a method to compute a reasonable displacement field is the topic of Chapter 4.

3.5 Discussion

In this chapter, we explored the possibilities for a volume-based similarity measure for comparing general paths on an evolving terrain. Our focus was on extending the sand function, which is a volume-based similarity measure for non-intersecting, connected paths on a static terrain, introduced by Sonke et al. [51]. Although we did not find a suitable measure yet, this chapter does contain some useful building blocks to extend the sand function, and poses some interesting questions for further research. The sand function, and the more general framework of volume-based similarity measures, provide a good starting point in pursuit of a suitable measure for comparing paths on different

terrains. To gain a better understanding of the resulting values for non-intuitive paths, it is crucial to define how paths are allowed to move, taking the setting of the problem into account. Depending on the context, the different ways to partition intersecting paths or connect disjoint paths considered in this chapter, provide ways to deal with more general paths.

Chapter 4

Transforming the terrain

In this chapter, we discuss an algorithm to reconstruct the transformation of a terrain, based on the principles of optimal transport. Specifically, we propose a method to compute a *displacement field*, which describes an elastic transformation of the source terrain into the target terrain.

Problem statement. Formally, we are given a triangulation T_Σ of a surface Σ, where each vertex is assigned two height values. Let $h_1, h_2 \colon M \to \mathbb{R}$ be the height functions obtained by linearly interpolating these height values over the edges and triangles of T_Σ. The goal is to obtain a reasonable reconstruction of the transformation of h_1 into h_2.

4.1 Optimal transport map

Given a pile of sand, what is the most efficient way to transport the sand to form a desired *target* shape? In 1781, Gaspard Monge studied the problem of efficiently redistributing mass from a set of sources into a set of targets. His original problem is an optimisation problem, where the objective is to minimise a cost of transportation, constrained to conservation of mass. In the last century, the problem sparked more interest again. This led to a lot of results in the field of optimal transport, and a wide variety of applications in a diversity of fields, like machine learning, image warping and economics.

Before we can present the formulation of the Monge-problem, we need to introduce some definitions. Let $X, Y \subset \mathbb{R}^d$ be two compact sets and let f be a *density* on X. A density is a non-negative function f such that it has total mass equal to 1, i.e.

$$\int_X f(x)\mathrm{d}x = 1.$$

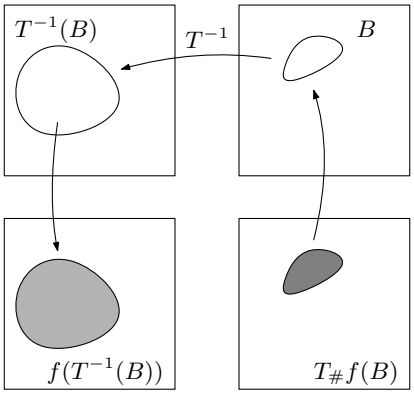


Figure 4.1: The push-forward of a measure.

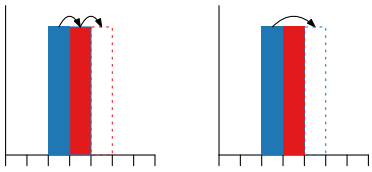


Figure 4.2: Two ways to transport books.

Next, let $T\colon X \to Y$ be a map and take a subset $B \subset X$. The *pre-image* $T^{-1}(B)$ of B under T is the set of points in X that T maps to B. Furthermore, the *push-forward* $T_{\#}f$ of f under T is a density on Y, locally satisfying conservation of mass:

$$\int_B T_{\#}f(y)\mathrm{d}y = \int_{T^{-1}(B)} f(x)\mathrm{d}x, \ \forall B \subset Y.$$

In words, this means that the mass (with respect to $T_{\#}f$) of every set B in Y is equal to the mass (with respect to f) of its pre-image, $T^{-1}(B)$, in X (see Figure 4.1). If $T_{\#}f = g$, for a density g on Y, we say T is a *transport map* which transports f into g.

Monge problem. We are now ready to state Monge's formulation of the optimal transport problem. Let $X, Y \subset \mathbb{R}^d$ be two compact subsets, let f, g be two densities on X and Y respectively, and let $c\colon X \times Y \to [0, \infty)$ be a non-negative, continuous, *cost function*. The cost function describes the cost of transporting mass from a point $x \in X$ to a point $y \in Y$. The Monge-problem is then to find a solution to the following minimisation problem:

$$\inf\left\{\int_X c(x, T(x))f(x)\mathrm{d}x \mid T\colon X \to Y \text{ s.t. } T_{\#}f = g\right\}.$$

If it exists, the transport map T that minimises the above expression, is called the *optimal transport map*.

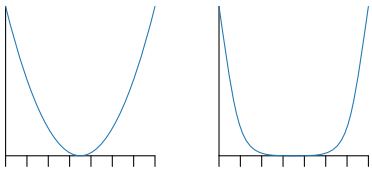


Figure 4.3: An example of a strictly convex (left) and a convex (right) function.

The existence and uniqueness of an optimal transport map depend on the choice of cost function c. A common analogy to understand why the choice of cost function is important, is the problem of moving books (see Figure 4.2). Suppose you have two books: one at position 1, and one at position 2. This setup defines your source distribution, and you want to move the books to a target distribution, where one book is at position 2 and one book is at position 3. There are two obvious transport maps to transform the source setting into the target setting. The first option is to slide both books one position to the right, and the second option is to take the book at position 1 and place it at position 3. The optimal transport map is given by the map that minimises the cost, and it turns out that for some cost there is no unique solution.

If the cost function is a simple Euclidean distance, $c(x, y) = |x - y|$, both transport maps have a cost of 2. Although there does exist an optimal transport map[1], it is not unique. On the other hand, if the cost function is a squared Euclidean distance $c(x, y) = |x - y|^2/2$, there is a unique optimal map. The first option gives a cost of 1, whereas the second option costs 2. In fact, it was shown by Brenier [11], that if $X = Y = \mathbb{R}^d$ and the densities f and g are zero outside a compact set, the general Monge problem admits a unique minimiser under a (scaled) quadratic Euclidean cost $c = \alpha|x - y|^2$.

This result was later generalised by Gangbo et al. [26]. They showed that for cost functions $c(x, y) = h(y - x)$, where h is a *strictly convex* function, the Monge problem admits a unique minimiser. A function $h : \mathbb{R}^d \to \mathbb{R}$ is convex if for all $x, y \in \mathbb{R}^d$ and for all $\lambda \in [0, 1]$ it holds that

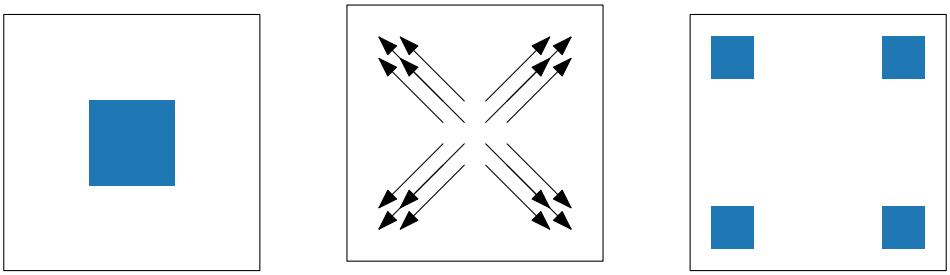
$$h(\lambda x + (1 - \lambda)y) \leq \lambda h(x) + (1 - \lambda)h(y).$$

If the inequality is replaced by a strict inequality, h is called strictly convex. Informally, a strictly convex function is such that any straight line connecting two points on the graph of the function lies above the graph, except for the endpoints (see Figure 4.3).

Displacement field. A transport map T assigns to every point in X a new location in Y. We define our displacement field as follows:

$$D(x) := T(x) - x. \tag{4.1}$$

[1] Under some regularity conditions on the densities, the cost function $c(x, y) = |x - y|$ always admits at least one optimal transport map [4]



(a) **(b)** **(c)**

Figure 4.4: Transforming the **(a)** source terrain into the **(c)** target terrain yields a **(b)** displacement field, describing the transport of mass.

To understand what a displacement field encodes, consider the displacement field in Figure 4.4b. The optimal transport map matches points in the source distribution to points in the target distribution. The displacement field then shows how the points moved.

Reformulations and computations. There exist numerous reformulations of the original problem by Monge, of which the *Monge-Kantorovich problem* [33] is the most studied one. The Monge-Kantorovich problem relaxes the original problem, by looking for *optimal transport plan* rather than *optimal transport maps*. An optimal transport plan is a coupling, assigning to each pair $x, y \in X \times Y$ a value indicating how much mass moves from x to y. If there exists an optimal transport map T, it can easily be expressed as an optimal transport plan γ, namely by setting $\gamma(x, y) = 1$, if $T(x) = y$, and $\gamma(x, y) = 0$ otherwise. The Monge-Kantorovich problem is a linear optimisation program under convex constraints, meaning that the problem also admits a dual formulation.

Yet another reformulation of the problem, is by considering a different point of view, and is inspired by methods from *computational fluid dynamics*. The formulations considered so far, are from a *Lagrangian* perspective: using a transport map T, we track points trough space, over time. Alternatively, we can also view the problem from a *Eulerian* perspective, where we do not track individual particles, but focus on fixed locations in the underlying space. Instead of finding an optimal map, we then want to find a function $\rho_t(x)$, that gives the density for a non-moving point x at time t. A Eulerian formulation of the problem, often called *dynamic optimal transport*, was first introduced by Beckmann [8]. The relation between dynamic and regular optimal transport was then later established by Benamou and Brenier [9]. An important component of the *Benamou-Brenier formulation*, is the *continuity equation*, which represents the conservation of mass constraint.

In theory, all of the reformulations solve to the same value. In practice however, it is in general not feasible to obtain an exact solution. Therefore, most existing algorithms only look for approximate solutions. When discretising, the various reformulations lead to distinct problems, for which different algorithms can be used to solve them (see for instance an overview paper on optimal transport on discrete grids [50]). The various approaches try to optimise the approximate solution in terms of several criteria, such as accuracy, efficiency and stability. In addition, it can also be desirable to preserve certain structural properties, such as the triangle inequality.

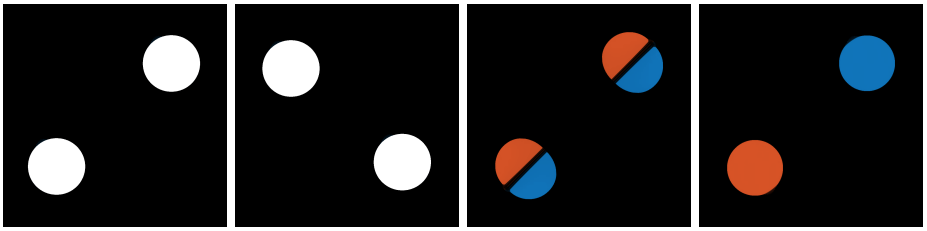
On a discrete grid, the Monge-Kantorovich formulation can be written as a linear program, which means it is solvable by classic algorithms such as the Simplex algorithm. It is typically not necessary to find an exact solution for a discretised problem, so a trade-off between efficiency and accuracy can be made. A commonly used algorithm is the Sinkhorn algorithm [19], which solves a simpler version of the Kantorovich problem, by *entropically regularising* the objective. Regularising an optimisation problem, means that the objective of the optimisation problem is modified, to make it easier to compute. In the case of entropically-reguralised optimal transport, a small multiple of the *entropy* of the source distribution is added, leading to a modified, but simpler, optimisation problem. The entropy of a distribution is a formal way to define how much information is encoded in the distribution. Finally, algorithms to solve the dynamical optimal transport problem typically use an iterative, gradient-descent based approach, to find a minimum-energy flow [9].

4.2 Transporting terrains

In this section, we discuss how to model a terrain such that we can apply the theory of optimal transport to obtain a displacement field. This means that we need to find an underlying space in which the transport happens, define two densities that represent the mass that we are transporting, and choose a suitable cost function capturing the effort of transporting mass. Recall that as input, we have a flat surface $\Sigma \subset \mathbb{R}^2$, and two height map $h_1, h_2 \colon \Sigma \to \mathbb{R}$, describing the elevation of a terrain on Σ. We assume that the soil in the terrain is homogeneous, and we refer to the soil as *sand*. Moreover, we assume that the heights in the input TINs are absolute, and that the *volume* of sand in both terrains is equal:

$$V(h_1) = V(h_2), \quad \text{where } V(h) = \int_\sigma h(x)\mathrm{d}x.$$

In fact, one can easily extend our methods to terrains that do not contain the same volume of sand, by normalising the height functions. However, as we will discuss in the next section, there are possibly better alternatives than normalising. Our goal is now to



(a) **(b)** **(c)** **(d)**

Figure 4.5: Optimal transport map from **(a)** source to **(b)** target distribution, for cost (4.2) with **(c)** $p_1 = p_2 = 2$ and **(d)** $p_1 = 1.1, p_2 = 3$. The transport is depicted by colour: orange moves to the left top, while blue moves to the right bottom (images from [32]).

find a displacement field, describing the transport of sand from the source, to the target terrain, constrained to conservation of volume.

Since sand is transported from a location on Σ to another location on Σ, we set $X = Y = \Sigma$. Informally, one can see this as a large pile of sand resting on a flat surface. For the densities, we take the normalised height functions:

$$f = \frac{h_1}{V(h_1)}, \qquad g = \frac{h_2}{V(h_2)}.$$

Intuitively, these densities capture the distribution of sand. To enforce that no sand is created or lost in the process, we consider only transport maps $T\colon \Sigma \to \Sigma$, such that $T_{\#}f = g$.

Lastly, we need to choose a suitable cost function $c\colon \Sigma \times \Sigma \to \mathbb{R}$, describing the cost of moving sand from a point $x \in \Sigma$ to a point $y \in \Sigma$. The most natural choice for the transport of sand is to consider Euclidean distance $c(x, y) = |x - y|$, for which we know that there exists an optimal map. To also guarantee uniqueness of the optimal map, one could consider a cost function of the form $c(x, y) = h_p(y - x)$. Here, the function $h_p(x) = |x|^p/p$ is strictly convex for $p > 1$. We have already seen the special case $c(x, y) = h_2(y - x)$, which prefers moving shorter distances as compared to the $p = 1$ case.

In [32], the authors consider a class of cost function that put an emphasis on transporting in a chosen direction. For $p_1 > 1$ and $p_2 > 1$, they consider the cost function

$$c(x, y) = \frac{|y_1 - x_1|^{p_1}}{p_1} + \frac{|y_2 - x_2|^{p_2}}{p_2}, \tag{4.2}$$

which is strictly convex and thus admits a unique optimal transport map. They illustrate that if $p_1 < p_2$, the resulting map penalises moving horizontally (see Figure 4.5). By detecting a direction of flow, one could tweak p_1 and p_2 such that the preferred direction

of transport is in the direction of the flow. Especially in estuaries, where due to tides the flow of a river is downstream as well as upstream, this could be an interesting consideration for the cost function.

An interesting study would be to see if there are other, more realistic cost functions that could be considered. For instance, the cost functions we discussed do not take the shape of terrain into account. As a consequence, the resulting map does not care whether sand moves up or down. A solution to that, would be to consider a volume-based cost function. However, if we were to consider a volume-based cost function, we do not have any certainty about the feasibility of the problem, i.e. whether there exists an optimal map.

Question 17. *What is a suitable cost function describing the cost of transporting soil? And if we can find one, can we prove that it admits an optimal map?*

An alternative model. An alternative to the model described above, is to consider the problem in a 3-dimensional setting. Then, we take $X = Y = \mathbb{R}^3$, and let f and g be the uniform densities on the subset $S_i = \{(x, y, z) \mid (x, y) \in \Sigma, 0 \leq z \leq h_i(x, y)\}$ for $i = 1, 2$ respectively. In this setting, one could consider a cost function that penalises moving up, and prefers moving down. A disadvantage of this approach is that there is no control on how sand moves relative to other sand particles. For instance, sand buried deep below a mountain could move, if that is what an optimal map prescribes. Still, given a reasonable cost function this might be an interesting idea worth exploring.

4.3 Discussion

In the previous section, we considered the use of Monge's problem to find a displacement field. To fit the precise problem formulation, we had to make some assumptions on our input. For instance, we assumed that the terrain is made of a homogeneous soil and that no sand enters or leaves the terrain over time. To obtain a more sophisticated model, we could consider some extensions to the original optimal transport problem.

Due to the process of sediment transport, sand is carried through rivers. Sand may leave the area of interest towards the sea, or it may enter the area of interest via an incoming channel. Therefore, it is not a reasonable assumption to say that the total volume of sand is the same in subsequent terrains. As mentioned, one could normalise the height function to get densities. However, this also means that we are no longer using *absolute* heights, which may lead to unexpected behaviour of an optimal transport map. One solution is add a *waste* point to one of the densities. The cost of transporting from a point on the terrain, to the waste point is set as high as necessary, so that only excess sand is transported there. A more general framework to deal with un-normalised densities is proposed in a paper on *unbalanced optimal transport* [18].

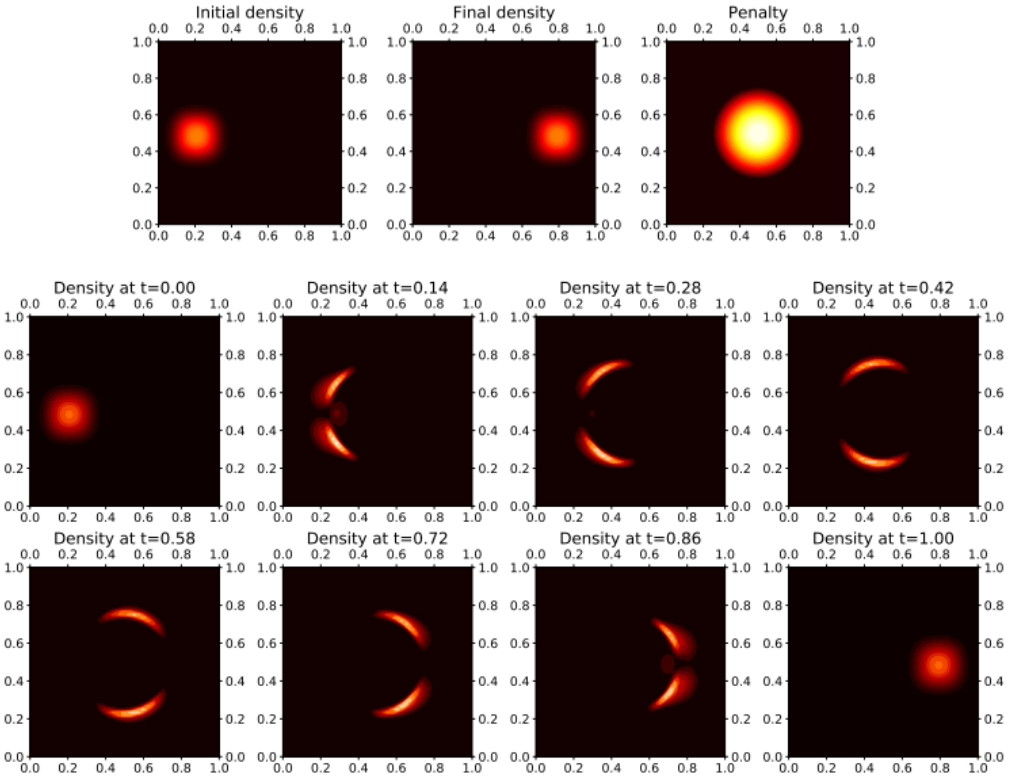


Figure 4.6: The top row shows the source and target distributions, and a mass penalty. The bottom shows the resulting interpolation for the computed mass constrained optimal transport map (image from [34]).

Another consequence of sediment transport is that the transport of sand often happens through rivers. In the previous section, we already proposed a cost function that takes this into account. Alternatively, one could also consider the aforementioned dynamic optimal transport problem, where the goal is to find a density for every point on the terrain. Additionally, the dynamical formulation of the problem allows for adding a *mass constraint* [34]. For instance, one could add a penalty for moving through a specified region of the terrain (see Figure 4.6). This type of problem, called *constrained mass optimal transport* could potentially also be used to model different types of soil.

Besides optimal transport, there are other ways to define a displacement field. As we mentioned in Section 1.3, a recent paper [14] proposed the use of image correlation to obtain a displacement field. The same authors compared their method to a Monte Carlo approach, called the Demon's algorithm. An interesting study would be to see how our optimal transport approach compares to these two algorithms.

Chapter 5

Implementation and experiments

In this thesis, we study the problem of modelling, and computing dynamic river networks. Recall that in Chapter 2, we defined a dynamic river network to be a channel matching that is optimal in some sense. To formalise the concept of a dynamic river network, we thus need to define the notion of an optimal channel matching. We proposed two models for this. Specifically, in a first model, a dynamic river network is a channel matching such that the similarity of matched channels is maximised. In Chapter 3, we discussed the notion of similarity in more detail, and we explored the possibility of using a volume-based similarity measure. In the second model, a dynamic river network is a channel matching, where each source channel is matched to a target channel by evolving in accordance to the transformation of the terrain. To reconstruct the evolution of a terrain, we considered the concept of a displacement field in Chapter 4.

In this chapter, we study how well these models work in practice. We primarily focus on the second model, based on the displacement field. A significant part of this thesis is dedicated to an experimental investigation of our models. To gain a better understanding of the problem, we consider available data and make some useful observations. Moreover, to validate or refute some of the ideas that arose in previous chapters, we implement and test some of the proposed methods.

Data sets. To get a feel for working with braided rivers, we study the simulated braided river model by Schuurman et al. [47]. The dataset consists of 662 subsequent DEMs, and we primarily use time step 500 as our source DEM, to ensure the model reached a stable period. For ease of testing and visualisation, we only consider a small part of the input DEMs. In particular, we work with a square region, see Figure 5.1. To evaluate how well the methods perform when the time step is increased, we consider three time steps to act as target DEM: time steps 501, 510 and 550. The difference of 50 time steps is equivalent to about 3 months of morphological development.

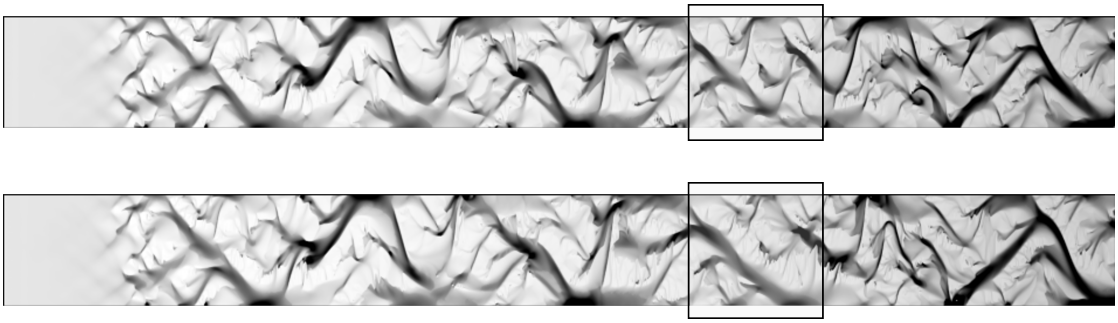


Figure 5.1: DEMs obtained from a simulated braided river model [47] shown as grey-scale images. The flow is from left to right. Darker areas indicate a lower elevation. Shown are time steps 500 (top) and 550 (bottom). We focus on the part of the river enclosed by the square.

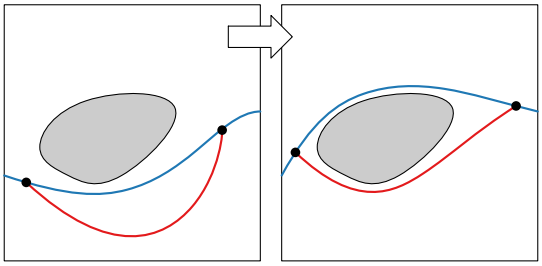
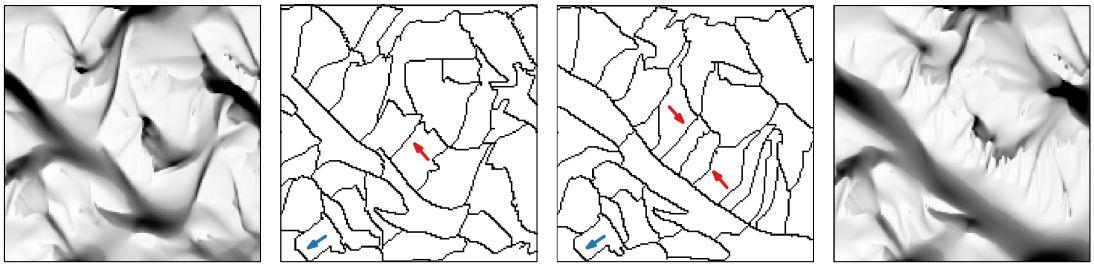


Figure 5.2: An example of an unrealistic synthetic test case. The grey area indicates a hill.

Besides simulated data, we also use a set of synthetic test cases. Such test cases are very simple, and provide insight in certain behaviour of proposed methods. In designing synthetic test cases, some care needs to be taken when choosing the terrain. Consider for instance the test case and corresponding 'correct channel matching' in Figure 5.2. This might seem to be a natural choice for a channel matching, but it completely ignores the fact that there is a hill blocking the presumed motion of the channel. Since there is not really another option in this case, it is unclear what a correct matching should be. Therefore, it is not necessarily an insightful test case. To prevent unrealistic terrain changes, we mostly use flat terrains in our test cases. These test cases give some idea of how our methods work, but do not capture the full range of what we wish to know.

Question 18. *How do we design useful synthetic test cases, that still portray realistic behaviour of the terrain?*

Input and output. We assume that the input DEMs are given as rectangular grids, covering a rectangular region of the terrain, and that for both DEMs we have computed



(a) **(b)** **(c)** **(d)**

Figure 5.3: Two river networks obtained from time steps **(a)-(b)** 500 and **(c)-(d)** 550.

a river network. Let $\Sigma \subset \mathbb{R}^2$ be a compact surface, covering the input region, and let T be a triangulation of Σ. Each vertex of T is equipped with two height values, describing the elevation of the source and target terrains. Let $h_1, h_2 \colon \Sigma \to \mathbb{R}$ be the two height functions, obtained as a linear interpolation of the height values over the edges and triangles of T. Moreover, let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be two river networks extracted from the input DEMs.

For simplicity, we restrict to finding channel matchings $M \colon E_1 \to E_2$, assigning to each source channel a single target channel. To visualise a channel matching, we use the channel matching based on colours (see Figure 2.2b).

Constructing dynamic river networks by hand. A first step towards a good model, is to better understand the input. Consider two DEMs from the simulated data set, obtained at time steps 500 and 550. By purely looking at the two terrains, it is not that hard to identify the 'main' channel, and to see how it evolves onto the target terrain. Moreover, some of the smaller channels in the bottom and at the top of the terrain are also still good detectable. However, for the smallest channels, it is already quite hard to identify them in the DEM, let alone match them over time. The same holds for comparing the two river networks extracted from these input DEMs. Some channels, like the ones indicated by the blue arrows, are relatively easy to match. The source channel indicated by the red arrow, is more difficult to match. There are three options, which all cross the same bar and follow a similar course. Moreover, we do not get much wiser by consulting the input DEMs, as none of the channels are clearly recognisable on the terrains.

Assuming that the networks resemble each other at least a bit, there are various factors that help when manually matching channels. Given a source channel, often the first step in finding a corresponding target channel is to select a few candidates, that have approximately the same *location* as the source channel. Typically, this yields only a few options. A next criteria is then to compare the *course* of the channels. The two

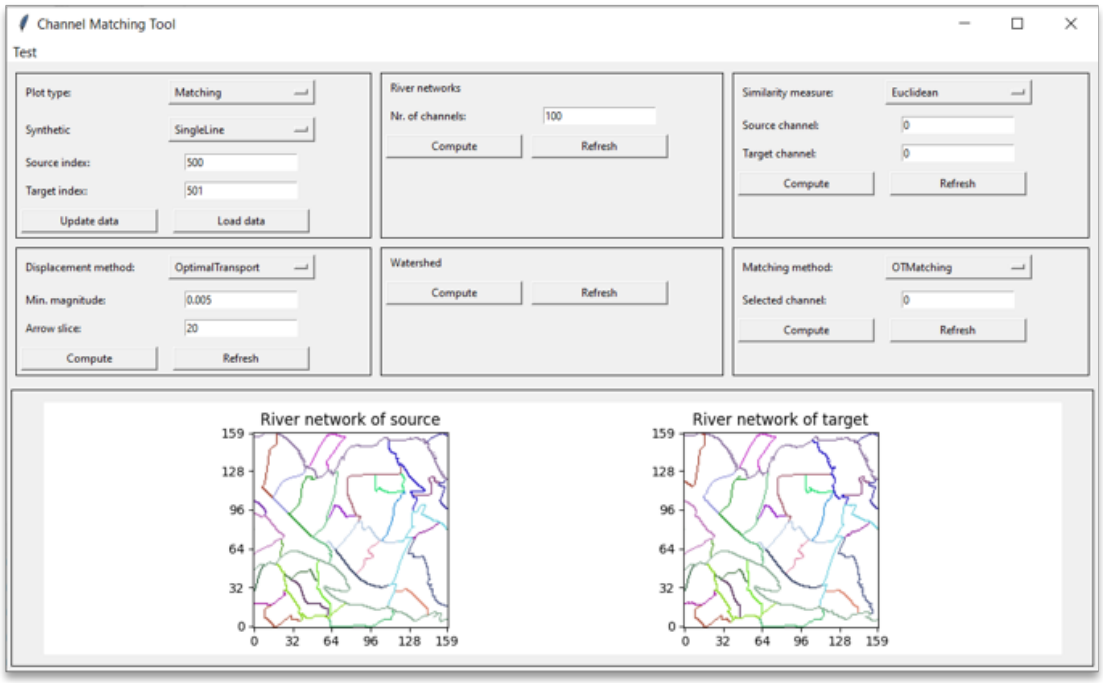


Figure 5.4: Screenshot of the tool developed for modelling dynamic river networks.

matched channels in the example are very likely the same, since they follow the same course in both terrains. If there are still too many options, one could consider to use the *shape of the terrain*, to deduce whether it is likely that a channel traversed in a certain direction.

Finding a dynamic river network manually is a tedious task, so we seek an automated method. Unfortunately, there is currently no way to validate automatically whether a resulting matching is reasonable. Some matchings might look good, but we do not know the ground truth. To design and test automated methods, it is therefore desirable to have an objective means of measuring the quality of resulting channel networks.

Overview. In the remainder of this chapter, we shift our attention to the automated computation of dynamic river networks. In Section 5.1 we first describe the implementation of a tool in which we can test our methods and inspect the resulting channel matchings. Next, in Section 5.2, we briefly discuss our first model of a dynamic river network, based on the use of a similarity measure. Lastly, in Section 5.3, we explore the possibilities of using a displacement field to construct a channel matching.

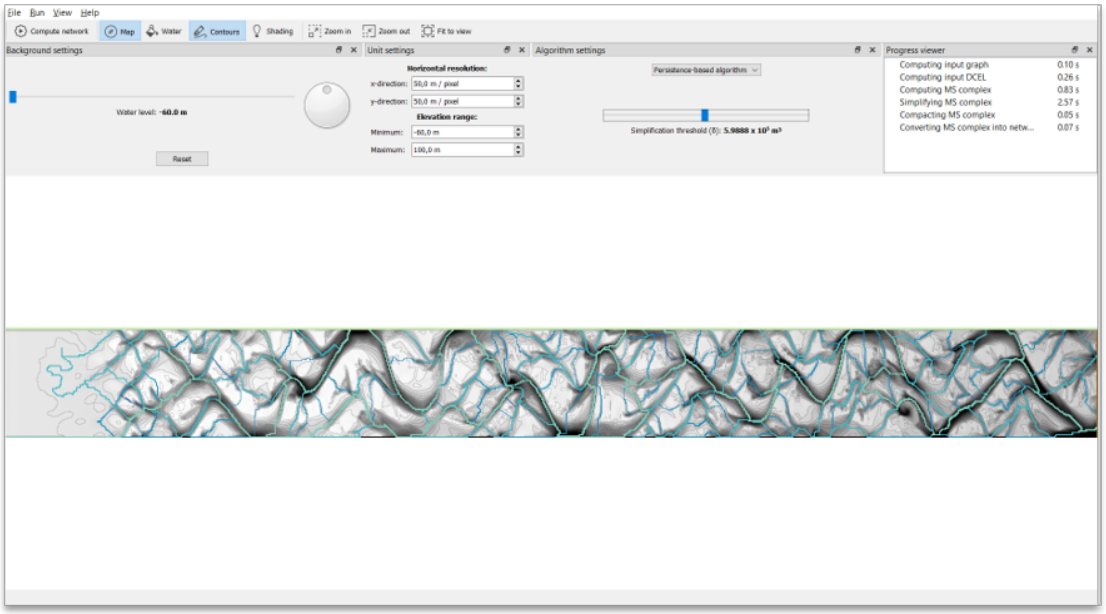


Figure 5.5: Screenshot of the graphical user interface of Topological Tools for Geomorphological Analysis. The river shown in the screenshot is time-step 500 of the simulated braided river.

5.1 Implementation

In this section, we describe the implementation of *channel matching tool* (CMT), a simple tool to test our methods. The tool uses various existing software packages to compute river networks and optimal transport maps. To inspect the output of the various numerical experiments, we implemented a graphical user interface (GUI, see Figure 5.4). The GUI allows the user to select input source and target DEMs, and displays them as grey-scale images. It furthermore allows the user to select the preferred method for computing a matching and set a few parameters. To visualise a channel matching, we use the colour-based approach (see Figure 2.2b).

Topological tools for Geomorphological Analysis. To compute a river network from a DEM, we use the software package called *Topological Tools for Geomorphological Analysis*[1] (TTGA). TTGA implements the lowest path algorithms [36, 42], discussed in Section 1.1. It consists of two separate interfaces. In the graphical user interface (see Figure 5.5), one can compute and view a construction of a river network. The second interface is a command-line interface, which is useful for integrating the construction of river networks into automated processes.

[1] https://github.com/tue-alga/ttga

TTGA assumes that the input consists of a rectangular grid T, representing the DEM of the terrain. After reading the input data, TTGA converts the input grids into triangular grids. TTGA allows two input formats: text-based and image-based. The text-based input format stores the height values of the vertices of T, in row-major order, and the image-based input format encodes height values in each pixel. An advantage of the image-based format, is that it enables an easy construction of synthetic test cases.

The produced river network is stored[2] as an ordered list of *links*, which are paths on T, representing a channel. For each link, the output contains a sequence of vertices on T, and an associated *significance value*. The significance value of a channel captures how similar it is to other channels in the network, in terms of the volume between other channels. One can specify a threshold parameter δ, and the resulting network then only contains channels that have a significance value higher than δ.

We are interested in the volume-persistence based algorithm [42], which yields a more stable river network. CMT integrates TTGA by using the command-line interface to extract a text-file containing a river network. We do not specify a significance threshold beforehand, so the resulting network contains all channels. When matching channels, we mainly consider the significant channels in the source network, and we expect that these are matched to significant channels in the target network. When analysing the evolution of a river network, the significance of a channel might be a good indication of its persistence over time.

Question 19. *What can we derive from a (lack of) change in significance of a channel?*

Sand function To compute the sand function, we use an unpublished version of TTGA, which includes a new package called the *generalised sand function* (GSF). GSF implements the volume-based similarity measures [51], as discussed in Section 1.2. The package extends TTGA's GUI, by allowing the user to manually draw two paths. Moreover, the tool allows the user to upload a text-file containing two paths. The input paths should be stored as sequences of vertices of T, and are assumed to not share any vertices, except for the endpoints.

To integrate the GSF tool into CMT, we implemented a command-line interface, that allows the same text-based input as the GSF GUI. As discussed in Section 3.1, we are interested in the volume-based similarity measure that uses the water flow surface as its base surface. The algorithm computes the volume of the enclosed domain above the base surface. To find the interior of the enclosed domain, we implemented a point-in-polygon algorithm [49]. The point-in-polygon algorithm determines whether a point is inside a polygon, by considering a ray, starting from the point and going in any direction. If the number of intersections with the boundary of the polygon is even, the point is outside

[2]TTGA also stores river networks as graphs, as we will discuss in Section 5.3.

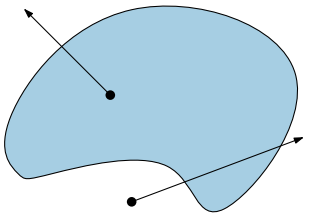


Figure 5.6: Illustration of the point-in-polygon algorithm.

the polygon, and otherwise it is inside (see Figure 5.6).

In CMT, the user can select a channel from the source network, and a channel from the target network. Before computing the sand function, the paths are pre-processed. First, the paths are connected, by adding shortest paths between the endpoints of both paths projected on a horizontal plane. Next, the paths are partitioned into non-intersecting pairs, following the approach proposed in Section 3.2. Lastly, the sand function is computed for every pair, and the resulting values are summed to obtain a total volume.

Optimal transport As discussed in 4.1, there are numerous algorithms to compute an optimal transport map, based on the formulation of the problem. Computational optimal transport is still a very active field, which means that the process of finding more efficient algorithms is very ongoing. Since we do not need an exact solution for our discretised problem, we are more interested in fast solvers for large grids. Therefore, we use a fairly new method that solves the dual problem of the Monge-Kantorovich problem, called the *back-and-forth method* [32]. In CMT, we use an open source software package[3] that implements the back-and-forth method.

5.2 Experimenting with a similarity measure

In this section, we briefly discuss the first model for a dynamic river network, based on the use of a similarity measure. Recall, in Section 5.2, we defined a dynamic river network as follows:

$$M(\pi) := \rho^* \in E_2, \text{ such that } d(\pi, \rho^*) = \min_{\rho \in E_2} d(\pi, \rho).$$

We consider two different similarity measures for comparing channels. A first way to compare channels, is to measure the distance between their endpoints. By comparing the endpoints, we capture the most important criteria in matching channels by hand:

[3] https://github.com/Math-Jacobs/bfm

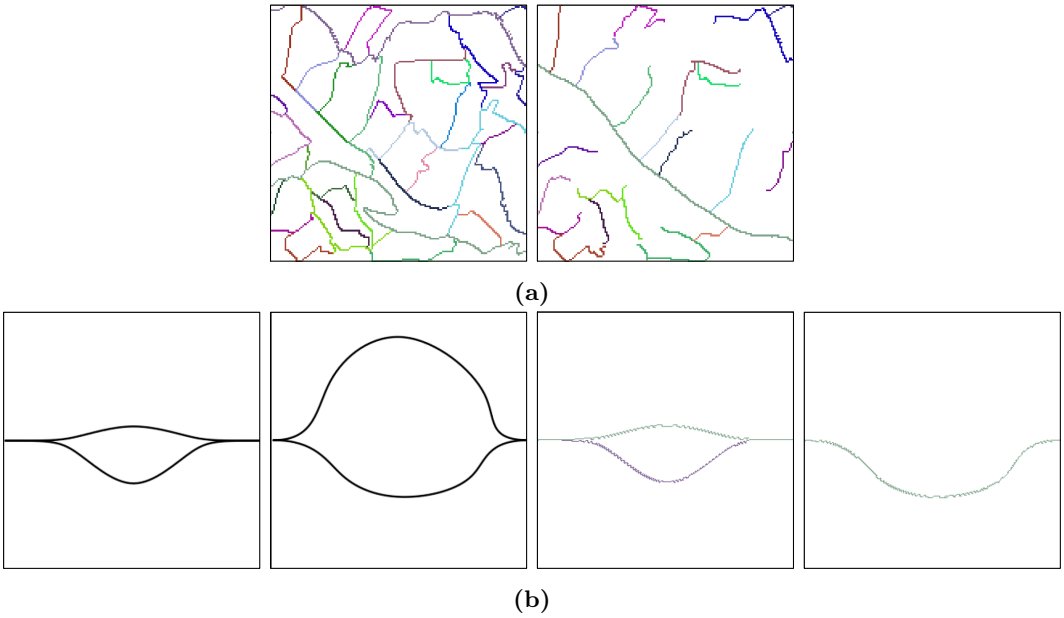


Figure 5.7: Two channel matchings computed using the similarity measure d_e.

the location of the channels. Formally, given a channel $\pi \in E_1$ and a channel $\rho \in E_2$, we define

$$d_e(\pi, \rho) := \|\pi(0) - \rho(0)\|_2 + \|\pi(1) - \rho(1)\|_2, \text{ where } \|x\|_2 := \sqrt{x_1^2 + x_2^2} \text{ for } x \in \mathbb{R}^2.$$

A clear disadvantage of this measure, is that it does not use the shape of the terrain in any way, and only considers the endpoints of the terrain. Therefore, we also consider the sand function d_{earth}, which we discussed in Chapter 3. We expect that this measure gives better results, as it does take the shape of the terrain into account. However, for large networks, it is not very efficient to compute the sand function for every pair of channels. In that case, we can make a trade-off between speed and accuracy, by first computing a set of candidates, using a more efficient, but less accurate, similarity measure.

Question 20. *How do we efficiently compute a channel matching based on the use of a similarity measure.*

For now, we only compute a channel matching using the similarity measure d_e. Specifically, Figure 5.7a shows a channel matching between the river networks extracted from the simulated data, and Figure 5.7b shows a channel matching between two artificial terrains. For the simulated data, we use the terrain at time step 500 as the source terrain, and the terrain at time step 550 as the target terrain. The resulting networks does not seem to be very good, as there are only a few target channels that are being matched. Of course, by only using the distance between endpoints, we do not use a lot

of information. This becomes even more apparent in the synthetic test case, where a seemingly easy pair of channels is mismatched.

5.3 Experimenting with a displacement field

In this section, we look at our second model for a dynamic river network. Recall, that in Section 2.2.2, we construct a channel matching as follows:

$$M(\pi) = r(L(D(1), \pi, 1)).$$

Here, $D(1)$ describes the displacement of the sand in the terrain, from the source to the target terrain. In Chapter 4, we discussed a method to obtain a displacement field, using the theory of optimal transport. Given the displacement, we want to track channels as they evolve over the terrain. To express the motion of channels in term of the displacement field, we introduced a function L. Applying L to a channel in the source network, gives a candidate path on the surface. That path does not necessarily correspond to a channel in the target network. The last step, is thus to snap the candidate path to the target river network, by mapping it to one of the target channels via a function r.

In this section, we experimentally explore how we can choose the functions L and r. We use both the simulated data set as well as a set of synthetic data. To find a suitable relation between the motion of channels and the displacement of the sand in the terrain, we first need to understand what a displacement field between two terrains encodes. Recall, a displacement field gives for every point x in the surface, a displacement vector $D(x)$. Such a displacement vector then specifies the new location of that point, relative to its old location. We store the input surface as a triangulated grid of points, and each grid cell represents a small region of the terrain. By computing a displacement field, every grid cell is assigned a displacement vector, describing the motion of the sand in that region.

Simulated data. In Figure 5.8, we show the displacement fields for four source-target combinations of the simulated data set. In all four combinations, the source terrain is the terrain at time step 500. To see how well our methods perform over an increasing amount of time, we consider four different target terrains, namely the terrains at time step 500, 501, 510, and 550. DEMs of these terrains are depicted in the first row of the figure, from left to right.

Since we use an algorithm that approximates the optimal transport map, its solution may contain some noise. Therefore, we remove this noise by ignoring very small displacements. The second row shows the displacement fields, where each displacement vector is plotted as an arrow. To see the directions of the arrows more clearly, we also plotted a

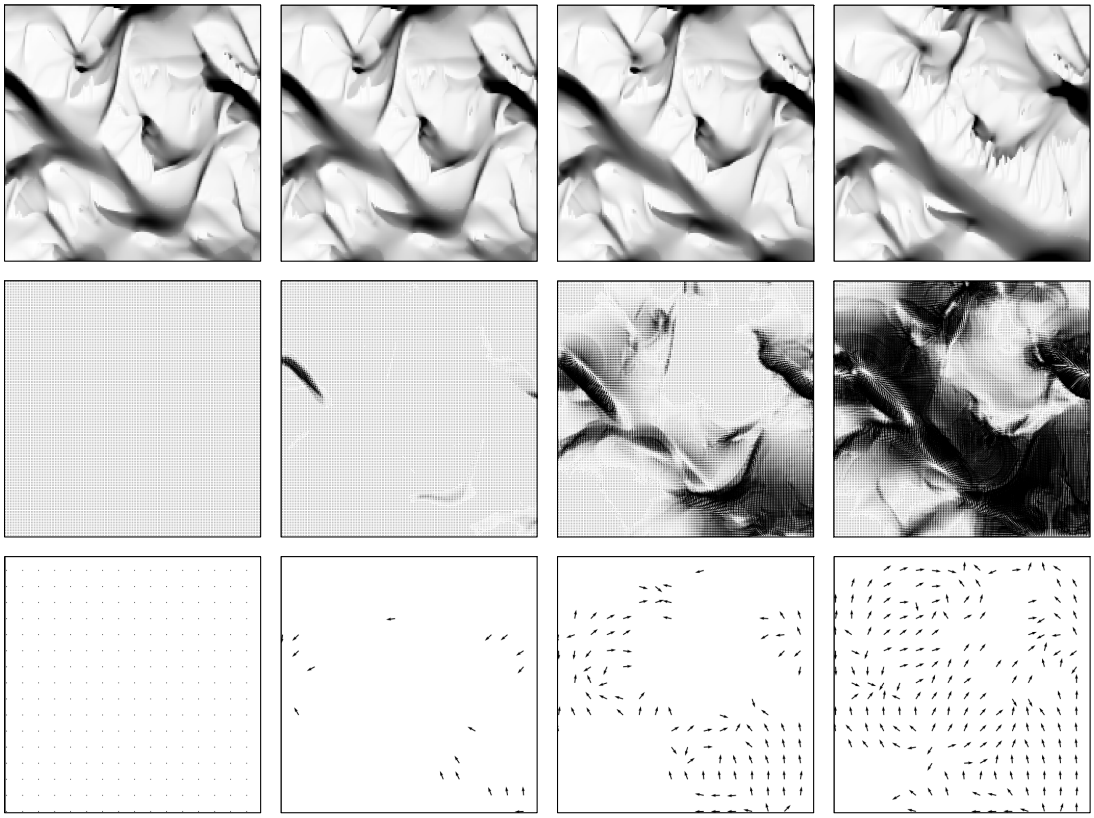


Figure 5.8: Displacement fields with as source terrain, the terrain at time step 500, and as target terrains, the terrains at time step 500, 501, 510, and 550 (left to right).

simplified displacement field, shown in the third row. To get the simplified displacement field, we increase the threshold for including displacement vectors. Moreover, we scale all the vectors to have the same length, and plot only a subset of the rows and columns of the displacement field.

Inspecting the middle row, containing the complete displacement fields, we see that the plots contain a lot of dots. A dot represents a zero displacement vector, meaning that the displacement at that point was smaller than our noise threshold. The first column shows the displacement of sand from a terrain to itself, and is used as a simple validation of the method. Naturally, no displacement is detected. The other plots contain lighter and darker regions. A lighter region in a plot is the result of relatively small displacement vectors, indicating that there is a noticeable transport of sand in these regions. A darker region on the other hand, is the result of many, relatively large, arrows, indicating that there is more transport of sand in such a region. In other words, we expect that the darker regions in the displacement field correspond to areas in the terrain that have undergone the most change.

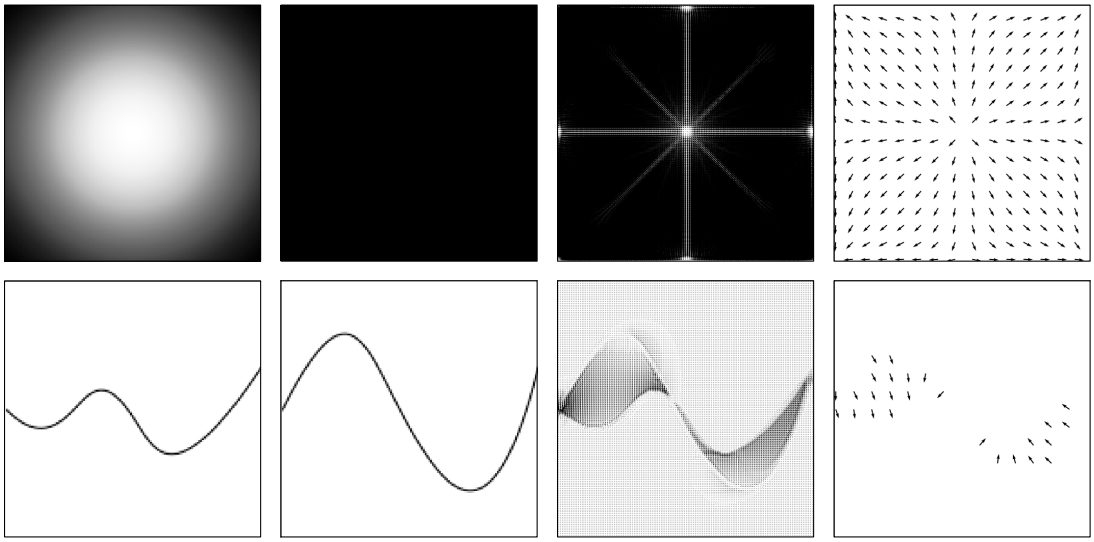


Figure 5.9: Two synthetic data sets and their corresponding (simplified) displacement fields. The top row shows a flattening hill, the bottom row shows a meandering river.

If we compare the dark regions in the displacement field, with their corresponding regions in the DEMs, it indeed seems to be the case that dark regions correspond to changing areas in the terrain. For instance, the bottom right corner of the DEM of time step 500 shows a relatively small channel. At time step 550 however, the channel is much larger, meaning that the terrain changed significantly. And indeed, the bottom right corner of the displacement field at time step 550 shows a dark region.

A displacement field gives an indication of areas in the terrain where a lot of transport happens. However, it is hard to tell in which direction the transport happens. Therefore, we also consider the simplified displacement field, containing fewer arrows (see the bottom row in the figure). Since the simplified displacement vectors have a normalised length, the plot does no longer contain information on the magnitude of transport of sand, but we do get a clearer picture of the direction of transport. For instance, if we again look at the bottom right corner, we see that the displacement vectors point north. Therefore, we expect that sand in that area of the terrain is transported north. Unfortunately, due to the complex nature of the morphodynamical changes in a terrain, it is difficult to visually confirm that it is indeed the case.

Interpreting displacement fields. To gain a better understanding of what kind of movement the displacement fields encode, we turn to more predictable terrains. Consider for instance the two synthetic data sets in Figure 5.9. The first data set contains a flattening hill. The source terrain, a hill, is modelled by a Gaussian distribution, and the target terrain is a flat plane, modelled by a uniform distribution. The source and target

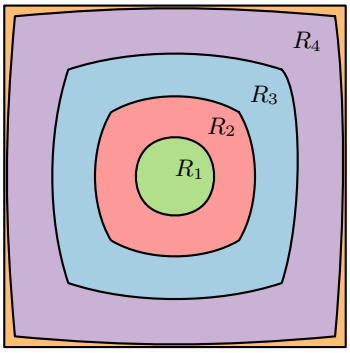


Figure 5.10: Illustration of how sand spreads out according to the displacement field.

terrains of the second data set both contain a single channel, represented as a low path, on an otherwise horizontal plane.

If we consider the hill example, we see that the transport of sand happens on the entire terrain. At the top of the hill, the sand diffuses in every direction, and near the boundaries, sand is pushed towards the corners. Note that the moving sand acts as a wave: it ripples through the terrain. In a similar fashion as the example of moving books (see Figure 4.2), the optimal transport map associated with this displacement field prefers moving shorter distances. To illustrate this, consider Figure 5.10. The sand in region R_1 in the source terrain spreads out to fill a larger region $R_1 \cup R_2$ in the target terrain. Likewise, the sand in region $R_1 \cup R_2$ diffuses into $R_1 \cup R_2 \cup R_3$, and so on.

The way sand is transported is mostly due to the choice of cost function in computing the optimal transport map. As discussed in Section 4.1, a quadratic cost penalises moving larger distances more. A disadvantage of a quadratic cost, is that processes such as sediment transport are not accurately captured. On the other hand, if a channel sweeps through the terrain, this is a reasonable model for the transport of sand. For instance, consider the second row in Figure 5.9. A single, meandering river moves through the terrain. We expect that this motion is smooth, and that the sand is moved in small steps. The displacement field seems to capture this well. In fact, the arrows seem to connect the river in the target terrain, to the river in the source terrain.

Tracing gradients. The example of the meandering river provides a promising model for the relation between a time-varying channel and the displacement field, as the displacement vectors seem to connect two corresponding channels. To be able to trace displacement vectors, we convert a displacement field D into a directed graph, which we call the *displacement graph*. For every vertex in T, we add a node to the graph. If the displacement vector at a point $x \in \Sigma$ is pruned by our noise-threshold, we do not add an outgoing edge from the corresponding vertex in the grid. For all other points x, we

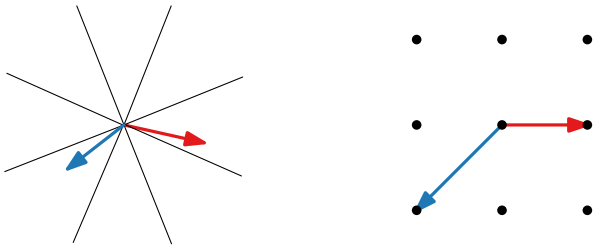


Figure 5.11: Converting a displacement vector into a directed edge.

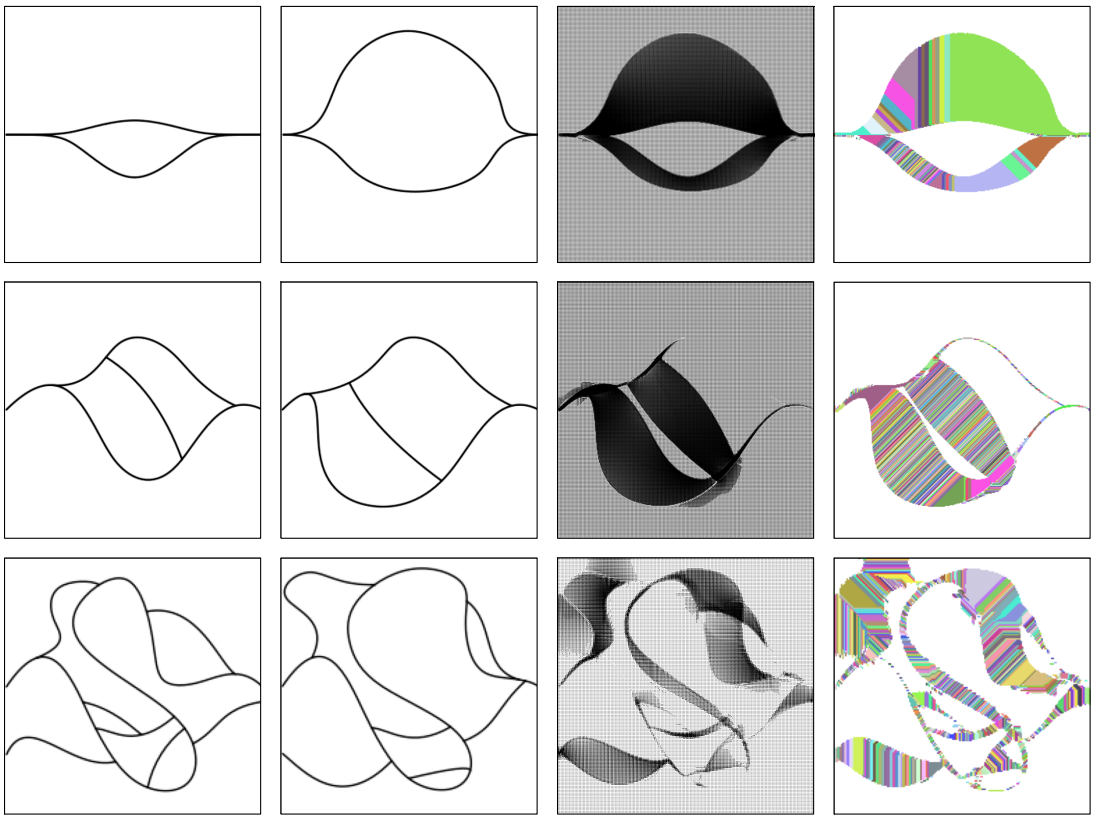


Figure 5.12: Three synthetic data sets and their corresponding displacement fields and traced points.

determine to which neighbouring grid cell y its displacement vector points, and add a directed edge from x to y. To determine to which neighbouring grid cell a displacement vector points, we check in which octant it lies (see Figure 5.11).

To see what we can learn from a displacement graph, we consider three new synthetic data sets (see Figure 5.12). These data sets contain simple, multi-channel rivers, on an otherwise flat terrain. The first and second column of the figure show the source and target terrains, the third column shows the displacement field again, and the fourth column shows the connected components, of at least size two, of the displacement graph. Each connected component is plotted in a distinct colour. An interesting observation, is that the connected components seem to provide a point-to-point matching of two channels.

Naturally, a channel does not move in the same direction as the sand in the terrain. In fact, one could argue that sand that erodes into a channel, opens up a new lower region of the terrain. The displacement fields, and thus also the displacement graph, exhibit a similar property. The channels seem to move in the opposite direction. Informally, this means that we want to model L as some sort of pre-image of the path. Note, that the pre-image is typically not enough, as it will only shift the channel a single grid cell. Moreover, taking the pre-image of a channel might give a non-continuous set of points.

For now, we use a very simple solution. We trace every point on the source channel over the edges in the displacement graph, until we either hit a target channel or end at a node with no more outgoing edges. We then match the source channel to the target channel that is hit by most points. This final step can be seen as a model for the snap function r.

Figure 5.13 shows the resulting channel matchings for the three synthetic data sets (on the left), and for three source-target pairs of the simulated data (on the right). In particular, for all three pairs, the source is again the terrain at time step 500, and the targets are 501, 510, and 550 (from top to bottom). If we consider the synthetic data sets, we see that the channel matchings seem to be quite reasonable. As for the simulated test cases, it is a bit harder to assess the resulting channel matchings. However, the channel matchings do look promising. In particular, the smaller channels in the source network seem to be matched to corresponding channels in the two earliest target networks.

Remark. In our current implementation, we used TTGA to extract links, representing channels. By construction, these links represent channels from sink to source, leading to possibly very long channels in the river network. There is, however, a second way to extract river networks from TTGA: as a graph, where each edge corresponds to a channel (from bifurcation to confluence). That is also how we model a river network,

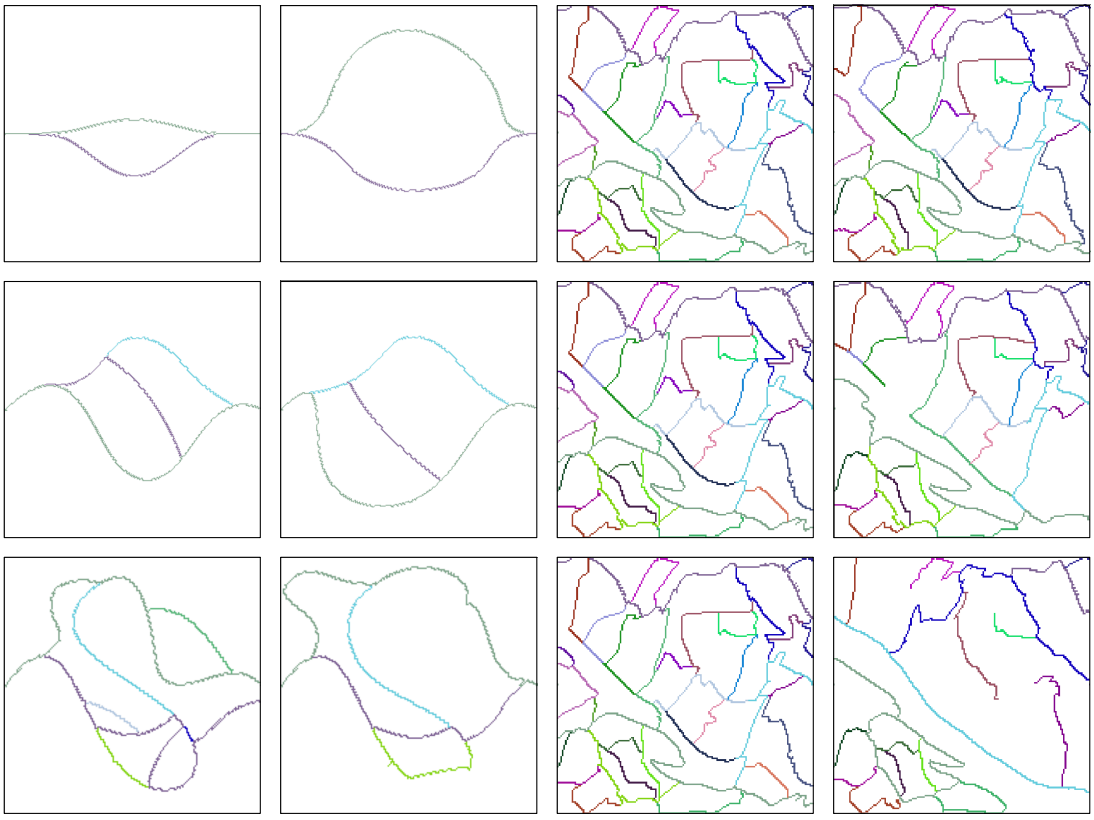


Figure 5.13: Channel matchings for three synthetic source-target pairs, and three simulated source-target pairs.

and it would thus be a better alternative to store the river networks. As a result, the channels in the river networks will be shorter, and that automatically leads to better matchings.

5.4 Discussion

In this chapter, we investigated the two models for a dynamic river networks experimentally. We only briefly discussed the first model, based on the use of a similarity measure. In particular, we saw that using a Euclidean distance measure between the endpoints of two channels is not a reasonable choice. To see whether the sand function is a better alternative, we need to improve the current implementation of the sand function. As of now, it is computationally inefficient to compute the sand function for every pair of paths in the river network. Moreover, there is no integrated way to test artificial data yet. A possible solution for the first issue, is to first construct a set of candidate target paths only, by using a more efficient, but less accurate similarity measure.

The second model, based on a displacement field, seems promising. Although the results do not always seem to be reasonable, this model does use all of the available information. Moreover, the current model is very flexible and can be extended in various ways. For instance, the displacement field is computed using a quadratic cost function and without taking the loss of sand into account. However, as we saw in Section 4.3, there are numerous ways to extend the construction of an optimal map. The same goes for the construction of the functions L and r. Currently, we use a very crude model for both functions, and we expect that a more rigorous definition could lead to improved channel matchings.

Chapter 6

Concluding remarks

In this thesis, we studied the problem of tracking channels over subsequent river networks. Specifically, we modelled the concept of a dynamic river network, which is a representation for a time-varying, multi-channel river. In Chapter 2, we saw that the problem of modelling a dynamic river network, can be decomposed into two sub-problems. Firstly, we need a reasonable, and most of all workable, definition for the concept of a channel matching. Secondly, we are not interested in just any channel matching, but rather one that resembles real-world changes in the channel structure of a river. We then proposed two models for a dynamic river network, both capturing a different notion of an optimal channel matching.

In Chapter 3, we discussed the first model, based on the use of a similarity matching. In particular, we considered extending a volume-based similarity measure, that captures the amount of earth between two channels on a terrain. The resulting measure is potentially a good tool to compare channels on an evolving terrain. Next, in Chapter 4, we considered a second model, based on a displacement field, that captures the transport of sand. We saw that the theory of optimal transport provides a method to compute a reasonable displacement field. There is a lot of freedom in the construction of a displacement field, as we can choose different cost functions. Moreover, we saw that there are various extensions and reformulations of the problem, that could potentially lead to more accurate displacement fields.

Finally, in Chapter 5, we implemented and evaluated our methods experimentally. The main focus of the test lies on using the displacement field to obtain a reasonable channel matching. We saw that, even under very rough assumptions, the resulting channel matchings are promising. In particular, they seem to capture the motion of the multi-channel river as a whole quite well. Currently, there is no objective way to measure the quality of resulting channels however. To validate our methods, it is desirable to have a means of objectively measuring the quality of a dynamic river network.

Future work. Throughout this thesis, we posed and stated a multitude of questions on defining, modelling and constructing dynamic river networks. These questions offer starting points for future research. Moreover, these questions have shown, that designing algorithms for context-specific problems requires a careful consideration of the problem statement. For instance, to construct a dynamic river network, we first need to know what exactly we want to capture with a dynamic river network. To answer such questions, it is important to also consider the context in which the algorithms will be used.

Given a clear problem statement, one can draw inspiration from the methods proposed in this thesis. Moreover, some of the key problems in defining, improving and visualising algorithms for the construction of algorithms have already been identified. Finally, our implementation CMT, provides a useful means test future methods.

Bibliography

[1] P. Agarwal, M. de Berg, P. Bose, K. Dobrint, M. Van Kreveld, M. Overmars, M. de Groot, T. Roos, J. Snoeyink, and S. Yu. The complexity of rivers in triangulated terrains. In *Proc. 8th Canadian Conference on Computational Geometry (CCCG)*, pages 325–330, 1996.

[2] H. Alt, P. Braß, M. Godau, C. Knauer, and C. Wenk. Computing the Hausdorff distance of geometric patterns and shapes. In *Discrete and Computational Geometry*, pages 65–76. Springer, 2003. doi:10.1007/978-3-642-55566-4_4.

[3] H. Alt and M. Godau. Computing the Fréchet distance between two polygonal curves. *International Journal of Computational Geometry & Applications*, 5(01n02):75–91, 1995. doi:10.1142/S0218195995000064.

[4] L. Ambrosio. Lecture notes on optimal transport problems. In *Mathematical Aspects of Evolving Interfaces*, pages 1–52. Springer, 2003. doi:10.1007/978-3-540-39189-0_1.

[5] L. Arge, J.S. Chase, P. Halpin, L. Toma, J.S. Vitter, D. Urban, and R. Wickremesinghe. Efficient flow computation on massive grid terrain datasets. *GeoInformatica*, 7(4):283–313, 2003. doi:10.1023/A:1025526421410.

[6] P. Ashmore. Channel morphology and bed load pulses in braided, gravel-bed streams. *Geografiska Annaler: Series A, Physical Geography*, 73(1):37–52, 1991. doi:10.1080/04353676.1991.11880331.

[7] J. Basch, L.J. Guibas, and J. Hershberger. Data structures for mobile data. *Journal of Algorithms*, 31(1):1–28, 1999. doi:10.1006/jagm.1998.0988.

[8] M. Beckmann. A continuous model of transportation. *Econometrica*, 20(4):643–660, 1952. doi:10.2307/1907646.

[9] J.D. Benamou and Y. Brenier. A computational fluid mechanics solution to the Monge-Kantorovich mass transfer problem. *Numerische Mathematik*, 84(3):375–393, 2000. doi:10.1007/s002110050002.

[10] B. Birnir and J. Rowlett. Mathematical models for erosion and the optimal transportation of sediment. *International Journal of Nonlinear Sciences and Numerical Simulation*, 14(6):323–337, 2013. doi:10.1515/ijnsns-2013-0048.

[11] Y. Brenier. Polar factorization and monotone rearrangement of vector-valued functions. *Communications on Pure and Applied Mathematics*, 44(4):375–417, 1991. doi:10.1002/cpa.3160440402.

[12] M. Buchin, S. Dodge, and B. Speckmann. Similarity of trajectories taking into account geographic context. *Journal of Spatial Information Science*, 9:101–124, 2014. doi:10.5311/JOSIS.2014.9.179.

[13] D. Cazanacli, C. Paola, and G. Parker. Experimental steep, braided flow: application to flooding risk on fans. *Journal of Hydraulic Engineering*, 128(3):322–330, 2002. doi:10.1061/(ASCE)0733-9429(2002)128:3(322).

[14] A.J. Chadwick, E. Steel, R.A. Williams-Schaetzel, P. Passalacqua, and C. Paola. Channel migration in experimental river networks mapped by Particle Image Velocimetry. *Journal of Geophysical Research: Earth Surface*, 127(1):e2021JF006300, 2021. doi:10.1029/2021JF006300.

[15] E.W. Chambers, E.C. de Verdiere, J. Erickson, S. Lazard, F. Lazarus, and S. Thite. Homotopic Fréchet Distance between curves or, walking your dog in the woods in polynomial time. *Computational Geometry: Theory and Applications*, 43(3):295–311, 2010. doi:10.1016/j.comgeo.2009.02.008.

[16] E.W. Chambers, D. Letscher, T. Ju, and L. Liu. Isotopic Fréchet Distance. In *Proc. 23th Canadian Conference on Computational Geometry (CCCG)*, 2011.

[17] E.W. Chambers and Y. Wang. Measuring similarity between curves on 2-manifolds via homotopy area. In *Proc. 29th Symposium on Computational Geometry (SoCG)*, pages 425–434, 2013. doi:10.1145/2462356.2462375.

[18] L. Chizat, G. Peyré, B. Schmitzer, and F.X. Vialard. Unbalanced Optimal Transport: Dynamic and Kantorovich formulations. *Journal of Functional Analysis*, 274(11):3090–3123, 2018. doi:10.1016/j.jfa.2018.03.008.

[19] M. Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Proc. 26th International Conference on Neural Information Processing Systems*, volume 2, pages 2292—-2300, 2013. doi:10.5555/2999792.2999868.

[20] X. Dai and S. Khorram. Development of a feature-based approach to automated image registration for multitemporal and multisensor remotely sensed imagery. In *Proc. International Geoscience and Remote Sensing Symposium*, pages 243–245, 1997. doi:10.1109/IGARSS.1997.615851.

[21] M. de Berg, O. Cheong, H. Haverkort, J. Lim, and L. Toma. The complexity of flow on fat terrains and its I/O-efficient computation. *Computational Geometry: Theory and Applications*, 43(4):331–356, 2010. doi:10.1016/j.comgeo.2008.12.008.

[22] H. Edelsbrunner, J. Harer, and A. Zomorodian. Hierarchical Morse complexes for piecewise linear 2-manifolds. *Discrete & Computational Geometry*, 30:87–107, 2003. doi:10.1007/s00454-003-2926-5.

[23] R.I. Ferguson. Understanding braiding processes in gravel-bed rivers: Progress and unsolved problems. *Geological Society, London, Special Publications*, 75(1):73–87, 1993. doi:10.1144\%2FGSL.SP.1993.075.01.03.

[24] C. Fey, M. Rutzinger, V. Wichmann, C. Prager, M. Bremer, and C. Zangerl. Deriving 3D displacement vectors from multi-temporal airborne laser scanning data for landslide activity analyses. *GIScience & Remote Sensing*, 52(4):437–461, 2015. doi:10.1080/15481603.2015.1045278.

[25] J. Feydy, B. Charlier, F.X. Vialard, and G. Peyré. Optimal transport for diffeomorphic registration. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 291–299. Springer, 2017. doi:10.1007/978-3-319-66182-7_34.

[26] W. Gangbo and R.J. McCann. The geometry of optimal transportation. *Acta Mathematica*, 177(2):113–161, 1996. doi:10.1007/BF02392620.

[27] A.A. Goshtasby. *2-D and 3-D image registration:For medical, remote sensing, and industrial applications*. John Wiley & Sons, 2005.

[28] S. Haker, L. Zhu, A. Tannenbaum, and S. Angenent. Optimal mass transport for registration and warping. *International Journal of Computer Vision*, 60(3):225–240, 2004. doi:10.1023/B:VISI.0000036836.66311.97.

[29] H. Hamedi and R. Shad. Context-aware similarity measurement of lane-changing trajectories. *Expert Systems with Applications*, 209:118289, 2022. doi:10.1016/j.eswa.2022.118289.

[30] M. Hiatt, W. Sonke, E.A. Addink, W.M. van Dijk, M. van Kreveld, T. Ophelders, K. Verbeek, J. Vlaming, B. Speckmann, and M.G. Kleinhans. Geometry and topology of estuary and braided river channel networks automatically extracted from topographic data. *Journal of Geophysical Research: Earth Surface*, 125(1):e2019JF005206, 2020. doi:10.1029/2019JF005206.

[31] A.D. Howard, M.E. Keetch, and C.L. Vincent. Topological and geometrical properties of braided streams. *Water Resources Research*, 6(6):1674–1688, 1970. doi:10.1029/WR006i006p01674.

[32] M. Jacobs and F. Léger. A fast approach to optimal transport: The Back-and-Forth method. *Numerische Mathematik*, 146(3):513–544, 2020. doi:10.1007/s00211-020-01154-8.

[33] L.V. Kantorovich. On the translocation of masses. *Journal of Mathematical Sciences*, 133(4):1381–1382, 2006. doi:10.1007/s10958-006-0049-2.

[34] S. Kerrache and Y. Nakauchi. Constrained Mass Optimal Transport. *arXiv preprint arXiv:2206.13352*, 2022. doi:10.48550/arXiv.2206.13352.

[35] M.G. Kleinhans. Sorting out river channel patterns. *Progress in Physical Geography*, 34(3):287–326, 2010. doi:10.1177/0309133310365300.

[36] M.G. Kleinhans, M. van Kreveld, T. Ophelders, W. Sonke, B. Speckmann, and K. Verbeek. Computing representative networks for braided rivers. *Journal of Computational Geometry*, 10(1):423–443, 2019. doi:10.20382/jocg.v10i1a14.

[37] Y. Liu, P.A. Carling, Y. Wang, E. Jiang, and P.M. Atkinson. An automatic graph-based method for characterizing multichannel networks. *Computers & Geosciences*, 166:105180, 2022. doi:10.1016/j.cageo.2022.105180.

[38] R. Mariescu-Istodor and P. Fränti. Context-aware similarity of GPS trajectories. *Journal of Location Based Services*, 14(4):231–251, 2020. doi:10.1080/17489725.2020.1842923.

[39] W.A. Marra, M.G. Kleinhans, and E.A. Addink. Network concepts to describe channel importance and change in multichannel systems: Test results for the Jamuna River, Bangladesh. *Earth Surface Processes and Landforms*, 39(6):766–778, 2014. doi:10.1002/esp.3482.

[40] I. Misra, M.K. Rohil, M.S. Manthira, and D. Dhar. Feature based remote sensing image registration techniques: A comprehensive and comparative review. *International Journal of Remote Sensing*, 43(12):4477–4516, 2022. doi:10.1080/01431161.2022.2114112.

[41] D. Motta, W. Casaca, and A. Paiva. Vessel optimal transport for automated alignment of retinal fundus images. *IEEE Transactions on Image Processing*, 28(12):6154–6168, 2019. doi:10.1109/TIP.2019.2925287.

[42] T. Ophelders, W. Sonke, B. Speckmann, and K. Verbeek. A KDS for discrete Morse-Smale complexes. In *Abstr. Computational Geometry: Young Researchers Forum (CG:YRF)*, pages 3:1–3:2, 2018.

[43] G. Peyré and Marco Cuturi. Computational optimal transport. *Center for Research in Economics and Statistics*, 2017.

[44] M. Ravanbakhsh and C.S. Fraser. A comparative study of DEM registration approaches. *Journal of Spatial Science*, 58(1):79–89, 2013. doi:10.1080/14498596.2012.759091.

[45] Y. Rubner, C. Tomasi, and L.J. Guibas. The Earth Mover's Distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121, 2000. doi:10.1023/A:1026543900054.

[46] F. Santambrogio. *Optimal transport for applied mathematicians*. Birkhäuser, 2015. doi:10.1007/978-3-319-20828-2.

[47] F. Schuurman, W.A. Marra, and M.G. Kleinhans. Physics-based modeling of large braided sand-bed rivers: Bar pattern formation, dynamics, and sensitivity. *Journal of Geophysical Research: Earth Surface*, 118(4):2509–2527, 2013. doi:10.1002/2013JF002896.

[48] U.M.S. Shamsi. Arc hydro: a framework for integrating GIS and hydrology. *Journal of Water Management Modeling*, 16:165–182, 2008. doi:10.14796/JWMM.R228-11.

[49] M. Shimrat. Algorithm 112: Position of point relative to polygon. *Communications of the ACM*, 5(8):434, 1962. doi:10.1145/368637.368653.

[50] J. Solomon. Optimal transport on discrete domains. *AMS Short Course on Discrete Differential Geometry*, 2018.

[51] W. Sonke, M. van Kreveld, T. Ophelders, B. Speckmann, and K. Verbeek. Volume-based similarity of linear features on terrains. In *Proc. 26th ACM International Conference on Advances in Geographic Information Systems (SIGSPATIAL)*, pages 444–447, 2018. doi:10.1145/3274895.3274937.

[52] D. Spada, P. Molinari, W. Bertoldi, A. Vitti, and G. Zolezzi. Multi-temporal image analysis for fluvial morphological characterization with application to Albanian rivers. *International Journal of Geo-Information*, 7(8):314, 2018. doi:10.3390/ijgi7080314.

[53] R.C. Veltkamp. Shape matching: Similarity measures and algorithms. In *Proc. International Conference on Shape Modeling and Applications*, pages 188–197, 2001. doi:10.1109/SMA.2001.923389.

[54] G. Ventura, G. Vilardo, C. Terranova, and E.B. Sessa. Tracking and evolution of complex active landslides by multi-temporal airborne LiDAR data: The Montaguto landslide (Southern Italy). *Remote Sensing of Environment*, 115(12):3237–3248, 2011. doi:10.1016/j.rse.2011.07.007.

[55] C. Villani. *Optimal transport: Old and new*. Springer, 2009. doi:10.1007/978-3-540-71050-9.

[56] Z. Wei, Y. Han, M. Li, K. Yang, Y. Yang, Y. Luo, and S.H. Ong. A small UAV based multi-temporal image registration for dynamic agricultural terrace monitoring. *Remote Sensing*, 9(9):904, 2017. doi:10.3390/rs9090904.

[57] J.M. Wheaton, J. Brasington, S.E. Darby, A. Kasprak, D Sear, and D. Vericat. Morphodynamic signatures of braiding mechanisms as expressed through change in sediment storage in a gravel-bed river. *Journal of Geophysical Research: Earth Surface*, 118(2):759–779, 2013. doi:10.1002/jgrf.20060.

[58] J.M. Wheaton, J. Brasington, S.E. Darby, and D.A. Sear. Accounting for uncertainty in DEMs from repeat topographic surveys: Improved sediment budgets. *Earth Surface Processes and Landforms*, 35(2):136–156, 2010. doi:10.1002/esp.1886.

[59] A.D. Wickert, J.M. Martin, M. Tal, W. Kim, B. Sheets, and C. Paola. River channel lateral mobility: Metrics, time scales, and controls. *Journal of Geophysical Research: Earth Surface*, 118(2):396–412, 2013. doi:10.1029/2012JF002386.

[60] S. Yu, M. van Kreveld, and J. Snoeyink. Drainage queries in TINs: From local to global and back again. In *Proc. 7th International Symposium on Spatial Data Handling*, pages 829–842, 1997.

[61] B. Zitova and J. Flusser. Image registration methods: A survey. *Image and vision computing*, 21(11):977–1000, 2003. doi:10.1016/S0262-8856(03)00137-9.