Planimetric simplification and lexicographic optimal chains for 3D urban scene reconstruction

Julien Vuillamy

UNIVERSITÉ CÔTE D'AZUR

ÉCOLE DOCTORALE
SCIENCES ET TECHNOLOGIES DE L'INFORMATION ET DE LA COMMUNICATION

THÈSE DE DOCTORAT

Simplification planimétrique et chaînes lexicographiques pour la reconstruction 3D de scènes urbaines

Julien VUILLAMY

Dassault Systèmes Provence – INRIA Sophia-Antipolis Méditerranée

Présentée en vue de l'obtention du grade de docteur en Informatique d'Université Côte d'Azur
Dirigée par : Pierre Alliez
Co-encadrée par :
David Cohen-Steiner et Florent Lafarge
Soutenue le : 17 septembre 2021

Devant le jury, composé de :

Dominique Attali	Directeur de recherche	GIPSA-lab Grenoble
Yusu Wang	Professeur	UC San Diego
Jean-Daniel Boissonnat	Directeur de recherche	INRIA Sophia-Antipolis
Tamal Krishna Dey	Professeur	Purdue University
Pierre Alliez	Directeur de recherche	INRIA Sophia-Antipolis
Florent Lafarge	Directeur de recherche	INRIA Sophia-Antipolis
David Cohen-Steiner	Chargé de recherche	INRIA Sophia-Antipolis
Gaël Guennebaud	Chargé de recherche	INRIA Bordeaux

DASSAULT SYSTEMES

ANRT ASSOCIATION NATIONALE RECHERCHE TECHNOLOGIE

Inria

Simplification planimétrique et chaînes lexicographiques pour la reconstruction 3D de scènes urbaines

Planimetric simplification and lexicographic optimal chains for 3D urban scene reconstruction

Jury :

Rapporteurs
Dominique Attali, Directeur de recherche, GIPSA-lab Grenoble
Yusu Wang, Professeur, University of California, San Diego

Examinateurs
Jean-Daniel Boissonnat, Directeur de recherche, INRIA Sophia-Antipolis
Tamal Krishna Dey, Professeur, Purdue University
Pierre Alliez, Directeur de recherche, INRIA Sophia-Antipolis
Florent Lafarge, Directeur de recherche, INRIA Sophia-Antipolis
David Cohen-Steiner, Chargé de recherche, INRIA Sophia-Antipolis
Gaël Guennebaud, Chargé de recherche, INRIA Bordeaux

Invités
André Lieutier, Ingénieur R&D, Dassault Systèmes Provence
Guillaume Randon, Ingénieur R&D, Dassault Systèmes

Abstract

Creating mesh representations for urban scenes is a requirement for numerous modern applications of urban planning ranging from visualization, inspection, to simulation. Adding to the diversity of possible input data – photography, laser-based acquisitions and existing geographical information system (GIS) data, the variety of urban scenes as well as the large-scale nature of the problem makes for a challenging line of research. Working towards an automatic approach to this problem suggests that a one-fits-all method is hardly realistic. Two independent approaches of reconstruction from point clouds have thus been investigated in this work, with radically different points of view intended to cover a large number of use cases.

In the spirit of the GIS community, the first approach makes strong assumptions on the reconstructed scenes and creates a 2.5D piecewise-planar representation of buildings using an intermediate 2D cell decomposition. Constructing these decompositions from noisy or incomplete data often leads to overly complex representations, which lack the simplicity or regularity expected in this context of reconstruction. Loosely inspired by clustering problems such as mean-shift, the focus is put on simplifying such partitions by formulating an optimization process based on a tradeoff between attachment to the original partition and objectives striving to simplify and regularize the arrangement. This method involves working with point-line duality, defining local metrics for line movements and optimizing using Riemannian gradient descent.

The second approach is intended to be used in contexts where the strong assumptions on the representation of the first approach do not hold. We strive here to be as general as possible and investigate the problem of point cloud meshing in the context of noisy or incomplete data. By considering a specific minimization, corresponding to lexicographic orderings on simplicial chains, polynomial-time algorithms finding lexicographic optimal chains, homologous to a given chain or bounded by a given chain, are derived from algorithms for the computation of simplicial persistent homology. For pseudomanifold complexes in codimension 1, leveraging duality and an augmented version of the disjoint-set data structure improves the complexity of these problem instances to quasi-linear time algorithms. By combining its uses with a sharp feature detector in the point cloud, we illustrate different use cases in the context of urban reconstruction.

Keywords: geometry, optimization, simplicial homology

Résumé

La représentation par maillages surfaciques de scènes urbaines est devenue un prérequis à de nombreuses applications de planification urbaine, comme la visualisation, l'inspection ou encore la simulation. La diversité des données souvent massives, sous forme de photographies aériennes, d'acquisitions laser ou issues de systèmes d'information géographique (SIG) ainsi que la variété des scènes urbaines en font une ligne de recherche ambitieuse. Aussi, une méthode unique et automatique ne pourra réalistiquement pas satisfaire l'intégralité des domaines d'application. Dans ce contexte, deux méthodes indépendantes sont étudiées dans ce travail, avec des points de vues radicalement différents afin de couvrir un large spectre de cas d'utilisation.

Proche des méthodes traditionnelles en SIG, la première approche adopte des hypothèses fortes sur les scènes à reconstruire, en représentant les bâtiments par extrusion d'une partition cellulaire 2-dimensionnelle. La construction de ces partitions, à partir de données bruitées ou incomplètes, conduit à des représentations souvent trop complexes comparées aux attentes de simplicité et régularité pour ce genre de maillage. Inspirée des méthodes de partitionnement comme le *mean shift*, l'attention des travaux est portée sur la simplification de ces partitions grâce à une méthode d'optimisation formulée comme un compromis entre la fidélité à la partition d'origine et des objectifs encourageant la simplicité et régularité de l'arrangement. Cette méthode emploie la dualité point-ligne, définit des métriques locales associées aux lignes de l'arrangement, et décrit une optimisation basée sur une descente de gradient dans un cadre de géométrie riemannienne.

La seconde approche est adaptée aux contextes où les hypothèses de régularité de la première ne s'appliquent pas. L'accent est alors porté sur la conception d'une méthode la plus générale possible, afin de pouvoir obtenir un maillage à partir de données bruitées ou incomplètes. En considérant une optimisation particulière, correspondant à un ordre lexicographique sur les chaines simpliciales, des algorithmes de complexité en temps polynomial, proches de ceux développés pour le calcul de l'homologie simpliciale persistante, permettent d'obtenir des chaînes minimales au sens lexicographique. Dans le cas de pseudo-variétés, la dualité ainsi que des structures de données efficaces permettent d'obtenir des versions en temps quasi-linéaires de ces algorithmes. Leur utilisation est enfin illustrée par plusieurs applications dans le contexte de la reconstruction urbaine.

Mots-clés : géométrie, optimisation, homologie simpliciale

Contents

Chapter 1

Introduction

Contents

1.1 Context

1.1.1 Inception of GIS

Cities are faced with a number of challenges that require to modernize the organization of spaces. Air pollution, cleaner transportation or efficient housing are all examples of complex and large-scale problems that need to be measured and analyzed in order to take informed decisions.

Geographic Information Systems (GIS) were born on this belief that combining multiple sources of information could improve the decisions related to urban spaces. In the 1960s, the Canadian government started storing and processing terrain information in an attempt to get a better understanding of their land-use. These systems, capable of processing spatially-localized semantic information allow to query complex phenomena at play in modern cities. For instance, provided the right information is given to the system, the following questions can be answered:

— What percentage of the city's area is dedicated to natural spaces?

— What areas are at risk in case of a natural disaster?

— How are hospital or schools distributed according to the population density?

— Where can a network of antennas be placed to maximize the overall connectivity?

The scope of use for GIS has extended beyond its traditional use as an analysis tool. As an example, the city of Paris has adopted legislation requiring that new constructions adhere to a minimal coefficient of vegetation, calculated differently whether it concerns ground level cover or landscaped building surfaces [Cit19]. These coefficients can be efficiently enforced with GIS.

Most data of these systems can be stored as 2-dimensional information. However, as we will see in Section 1.2, 3-dimensional data are needed for numerous applications related to urban planning.

1.1.2 Representations of geometry for buildings

The first representations of geometry in Geographic Information Systems were generated for terrain environments. A distinction was made between Digital Elevation Models (DEM), representing the bare-Earth surface, and Digital Surface Models (DSM), containing both the natural and artificial elements of the environment. These models are represented either by rasters or using Triangular Irregular Networks (TINs) allowing to interpolate the measurements. These representations are useful for terrain analysis but are not well suited to built scenes, which present regularities and sharp height discontinuities.

The representation of buildings in GIS has been largely influenced by its traditional 2-dimensional past. A decomposition into Level Of Details (LOD) is frequently used. For instance, as illustrated in Figure 1.1, the CityGML standard [Sta12] characterizes the following 5 levels:

— The coarsest level (LOD0) represents buildings as their footprint or roof edge polygons. These footprints are often represented as closed polylines.

— LOD1 adds a height information to form prismatic blocks for each building. This can be seen as a piecewise-constant function on the 2-dimensional horizontal space.

— By allowing differentiated roof structures, the next finer level (LOD2) represents buildings as piecewise-linear functions on a 2D space.

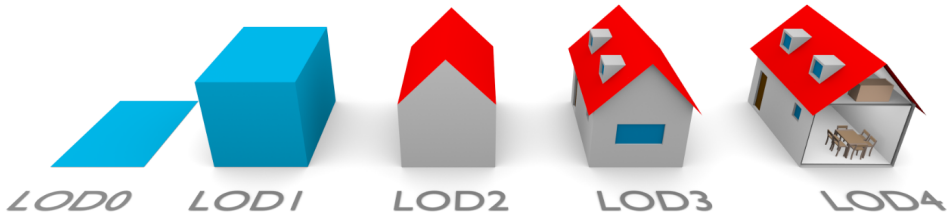


Figure 1.1: The five levels of details described in the CityGML standard [Sta12]. Image from [BSL+15].

— LOD3 introduces openings (windows and doors) and finer geometric elements (chimneys, balconies). This is the first level that introduces proper semantic for buildings.

— The finest level (LOD4) is used to detail interiors of buildings.

LOD models offer a good trade-off between the complexity of the representation, which needs to be as small as possible in GIS databases to efficiently perform spatial queries, and its ability to represent most urban architectures. The levels of details are also perfectly suited to visualization of large environments. Finally, their wide adoption by software and practitioners has made them the *de facto* standard representations for buildings in GIS.

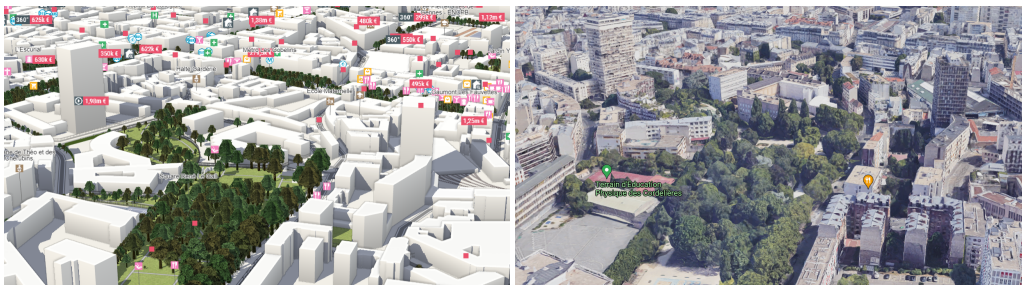
In the past, the creation of these GIS databases was done manually from ground measurements or imagery, with software offering interactive tools to simplify the process. This manual creation is costly and time consuming. Automatic reconstruction approaches promise to make this process more efficient and allow for more frequent updates of these databases.

The definition and applicability of these LOD models are however topics of discussion [BLS16]. When describing a few applications of urban planning requiring 3-dimensional information (Section 1.2), we will compare these LOD *parsimonious* descriptions to *dense* meshes, that is less compact triangular meshes reconstructed without the strong regularity priors and structure of LOD representations.

1.1.3 Towards Building Information Modeling (BIM)

Instead of creating a single snapshot of the urban environment at the time of acquisition, the idea of Building Information Modeling (BIM) revolves around following the whole lifecycle of urban assets, from their design, cost estimation, construction, certification to their destruction. Contrary to its name, this applies to more than simply buildings and encompasses all urban infrastructure such as roads, gas and water pipelines or street furniture. This single model is used to coordinate multiple actors working on a same construction project. Its main advantage is that this approach can be time-saving and less error-prone in comparison with each actor having to maintain their own digital representations. The requirements on the models are of course a lot more detailed than previously described LODs: this model contains finer geometry (electrical and heating systems, pipes) as well as more semantic labels (material types, construction information, certifications).

BIM has seen its adoption grow dramatically in recent years and has become a requirement for public construction projects in multiple countries [ULW19]. BIM representations are often restricted to new construction projects, as their creation for existing buildings is an arduous task [THA+10, XAAH13, HY18]. The gap between GIS and BIM



(a) Visualization based on LOD1 (b) Visualization based on dense meshes

Figure 1.2: Illustration of visualization tools for (a) the real estate service Bien'ici [Bie21] using GIS at level of details 1, and (b) the online mapping service Google Maps [Goo21] using dense meshes.

applications is however becoming smaller with time, many systems already offering interoperability between the different data formats.

1.2 The need for meshes in applications

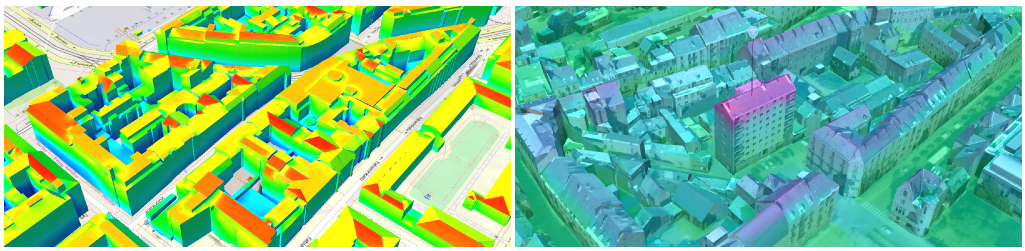
A growing number of applications, both in GIS and BIM industries, require 3-dimensional representations of buildings. We focus in this section on understanding the requirements imposed to the representations by three major domains of use: visualization, inspection and simulation. An extensive description of other use cases can be found in [BSL+15].

1.2.1 Visualization

The most obvious and publicly accessible use of 3D building representations is for visualization purposes. Numerous online services such as Google Maps propose a 3D view of cities, often enriched with semantic information. For real estate projects, a visual contextualization, combining architectural sketches of a project to an existing environment, is a great medium of presentation to the public. Although GIS practitioners might prefer working in 2D reference views for positioning tasks in 3D space, 3-dimensional representations are more intuitive for a non-expert audience and are also better suited to shape understanding tasks than multiple 2D views [SJCSO01].

Although rarely used, point clouds can be a reasonable visualization medium, when sufficiently dense and combined with color and normal information as well as eventually processed by shaders improving the depth understanding (as in Figure 1.4). However, this representation requires massive amounts of data to correctly render the illusion of depth continuity. Also, any defects in the data such as noise or occlusions will degrade immensely this visualization.

Deciding between dense meshes and LOD representations for visualization is more debatable. As illustrated in Figure 1.2, the dense mesh will of course contain a lot more details of the scenes and will be more accurate than LOD models. However, the simplified representation provides enough visual cues to understand the scene and gives a less cluttered visualization, especially when urban furniture such as tree or cars can be filtered out. Also, the hierarchical nature of LOD representations allows for efficient rendering, while dense meshes require additional processing such as mesh compression and streaming to offer a smooth visualization experience.



(a) Solar exposition simulation (b) Electromagnetic simulation

Figure 1.3: Examples of simulation based on LOD2 representations. (a) Visualization of a solar exposition on the city of Helsinki [Cit15]. (b) Visualization of an electromagnetic simulation performed on the city of Rennes [RS19].

1.2.2 Change tracking and inspection

Another common use for 3D models of urban scenes concerns change tracking and inspection. Keeping a digital representation of the whole life cycle of a built asset in BIM means confronting this virtual model to its real counterpart. This can be used to update the model after new data acquisitions, to enforce rules such as verifying construction permits [SZBH15] or to inspect structures at smaller scales [Eur20].

The type of geometric representation depends mainly on the scale of the task. For the verification of construction permits, LOD2 models provide enough information to correctly estimate the position, floor-plan area and construction heights. Inspection will often demand *dense* meshes, with a higher degree of precision than typically described in GIS. Indeed, the tasks will require to precisely measure the deviation between a virtual twin and the captured real representation of the scene. This is often performed in domains such as civil engineering, to verify structures with difficult human access, or for cultural heritage preservation.

1.2.3 Simulation

Finally, the ability to perform simulation in urban environments has the potential to drastically improve the efficiency of constructions and the well being of residents. Simulation will probably take an even larger part in the future, especially with the advent of BIM: structural calculations, energy consumption or durability reports are all examples of certifications that can be done through simulation. Each of these types of simulations however requires precise semantic information and has its own specific requirements on the geometry.

However, some simulations can be performed on LOD representations, as illustrated in Figure 1.3. Electromagnetic and solar exposition can be simulated using models with differentiated roof structures (LOD2). Other calculation, such as wind, noise or pollutions are other examples of simulations where LOD1 or LOD2 models are acceptable representations.

Newer simulation applications require more detailed models. As an example, the deployment of 5G technologies in dense urban contexts has proven to be challenging: the use of millimeter wave frequency, which allows to increase the bandwidth speed, reduces the range of transmission and is highly affected by occlusions. The placement of antennas therefore requires to be carefully thought out in order to create a stable coverage. Simulating this placement requires line-of-sight computation on dense meshes.

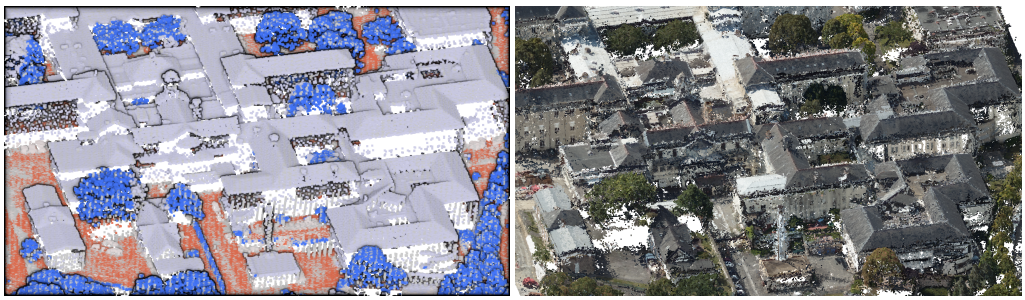


Figure 1.4: Visual comparison of LiDAR (left) and photogrammetry-based (right) point clouds of the same scene. The LiDAR scan is shown with its semantic classification (red: ground, dark blue: vegetation, light blue: buildings). While LiDAR point clouds offers an almost noise-free sampling of the scene, all verticalities are undersampled. Point clouds obtained by a photogrammetry process offer a more uniform but noisier sampling. The visualization of the LiDAR set of points uses a shader for better understanding of depth [Bou09].

1.3 Data acquisition

Large-scale surveys of urban scenes are done either by photography or using laser scanning techniques. We describe succinctly both process of acquisition and highlight the characteristics of the measurements provided by each type of acquisition. A visual comparison of the different point clouds on the same scene is given in Figure 1.4.

1.3.1 Photography-based acquisitions

Using photography for the reconstruction of 3D geometry is one of the most studied problem of Computer Vision and has been an active line of research for the past 30 years [FH15]. The most successful approaches are based on computing stereo correspondence. In a process called *Structure-from-Motion*, local image descriptors [Low99, RRKB11] are paired into correspondences in overlapping images. The camera intrinsic parameters and image viewpoints can then be robustly computed (up to a global similarity transformation) by an optimization known as *Bundle adjustment* [TMHF00]. In a second phase, the geometry needs to be estimated from the correctly positioned images. We mention in particular, patch-match approaches, which use a probabilistic framework based on photometric [ZDJF14] and geometric consistency [SF16] priors to estimate the depth and normal maps of each image. These depth maps are then combined to form a set of points representing the captured scene.

Although this process has been applied to challenging inputs such as mobile imagery for street views or even databases of public images scraped on the Internet [ASS+09], most professional acquisitions use aerial or drone imagery, where the trajectory is controlled to ensure optimal coverage of the region of interest and good overlaps between images. This controlled acquisition helps to reduce defects in the output set of points, which can be noisy or incomplete when the image coverage of the scene is not sufficient. The error of current photogrammetry methods in airborne settings is less than the meter, and measured in centimeters for drone acquisitions [LCZ+19].

1.3.2 Laser scanning acquisitions

Commonly grouped under the LiDAR acronym for Light Detection And Ranging, laser scanning denotes all remote sensing techniques using the response of a laser beam to

measure the distance of an object to the device. These techniques come in different flavors: they can be either fixed (Terrestrial Laser Scanning; TLS), embarked on a small aircraft (Airbone Laser Scanning; ALS), on drones (Unmanned Aerial Vehicule LiDAR; UAV LiDAR) or on cars (Mobile Laser Scanning; MLS). Each of these methods has its own scope of use: ALS is better suited to aquisitions of large scenes, while TLS offers the best precision. UAV LiDAR and MLS are used on smaller regions, for instance in the survey of a construction project.

A lot of context-specific choices are made when deciding the LiDAR acquisition strategy. For instance, LiDAR systems can either record the full waveform response or record discrete returns (up to 4 returns). This distinction is particularly relevant for forest surveys, where the full waveform response corresponds to the profile of the tree canopy, and discrete returns simply correspond to the tree top and ground. In urban context, the massive amount of data of the full waveform is not useful, and the discrete returns strategy is preferred. Coupling the laser acquisitions with a global positioning system (GPS) and inertial measurement units (IMU), the discrete returns are used to compute a set of regularly-spaced 3D points along scan lines. The precision of these systems depends on many criteria (flight height, laser beam width, positioning error) but the measurement accuracy in height of airborne systems is around 10 centimeters. In aerial settings, the angle between the scanner and the nadir direction is often small and therefore the visilibity of vertical structures is reduced. This means that facades are generally sparsely sampled compared to other parts of the building. The method is also known to generate a lot of outliers, corresponding to incoherent measures far from the observed surface, which need to be removed in post-processing.

The resulting point cloud often contains additional attributes: intensity, color if the system is coupled with a photographic system, land use classification, scan angle or number of returns.

1.4 Scope of Interest

From the context set by application needs and acquisition capabilities described in previous sections, we can give a precise scope of interest for this work.

Inputs Although many reconstruction methods are based on aerial imagery or Digital Terrain models, this works focuses on a common denominator for both photogrammetry and LiDAR methods: point clouds. We will assume that no additional information is associated to the set of points. However, we want eventually to exploit optional given information, such as normals, colors, or classification.

The challenge of considering point clouds resides mostly in the unstructured nature of this data source. Considering two different kinds of acquisition means dealing with the defects appearing in both photogrammetry and LiDAR point clouds: noise, occlusion and the presence of outliers. Finally, these point clouds are often massive in size and need to be efficiently processed.

Geometry representation We are interested in creating parsimonious geometry with differentiated roof structures at large scales (i.e. city blocks). Integrating these models in GIS software requires conforming to LOD representations. The goal is therefore set on LOD2 geometry, as the robust treatment of further details such as windows or doors, would require additional data such as ground-level imagery. The challenge for this type of reconstruction is to obtain geometry that is as simple and regular as manually created GIS databases.

For applications at smaller-scales (i.e. single building), reconstructions with finer geometry are required, ones that do not fit with the strong assumptions imposed by LOD representations. This type of reconstruction, using dense meshes, should therefore focus on being as general as possible.

Semi-automatic The variability of chosen inputs and the variety of urban scenes to reconstruct make a fully-automated process hard to conceive. Instead, we position our scope of interest on semi-automatic processes, where the interaction are reduced to the choice of a few parameters, as well as user verification at different steps of the reconstruction process. The goal is that, once these parameters have been tuned correctly on a smaller region, the reconstruction can be applied at larger scales automatically.

Chapter 2

State of the art and contributions

Contents

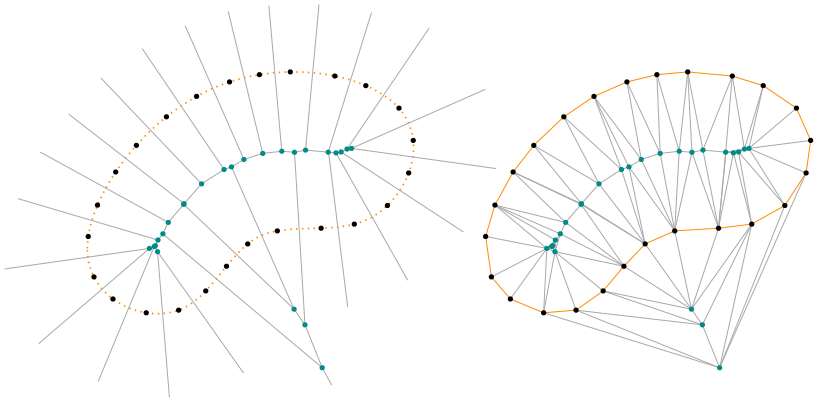


Figure 2.1: Illustration of the crust algorithm in dimension 2. Left: From a set of points sampling the surface (orange dotted curve), the Voronoi diagram is used to define poles (cyan points). Right: after constructing the Delaunay triangulation of the points combined with the poles, the crust reconstruction is given by the set of edges with vertices in the original set of points.

The scope of interest described in Section 1.4 encompasses both general reconstruction methods from points clouds in order to create dense meshes, as well as urban-specific techniques interested in parsimonious representations.

2.1 Generic reconstruction methods

The problem of surface reconstruction has been extensively studied and the survey by Berger et al. [BTS+14] classifies a large number of methods according to the assumptions and information used in addition to geometry. We focus mainly on the methods that have been used frequently for terrain and urban environment modelling, in particular interpolatory techniques based on a Delaunay triangulation and methods using an implicit function.

2.1.1 Delaunay-based interpolatory reconstructions

A number of reconstruction algorithms are based on extracting a subcomplex of the Delaunay tetrahedralization of the input point cloud. A summary of these methods is given in [CG04]. These types of approaches can be divided into two categories, either surface-based or volume-based reconstructions.

Surface-based approaches select a subset of triangles in the Delaunay tetrahedralization. A prime example of this type of approaches is the crust algorithm by Amenta et al. [ABK98, ACDL00], which is based on the observation that the shape of Voronoi cells for a set of points densely sampling a surface is elongated along the normal direction of the surface. This allows to define a subset of vertices of the Voronoi diagram named *poles* which approximate the medial axis, as illustrated in Figure 2.1. Although methods theoretically guaranteeing the quality of reconstruction in 2 dimensions previously existed [BB97, Att98], the crust algorithm was the first to give guarantees in dimension 3. The algorithm then computes the Delaunay triangulation of the original points combined with these additional poles and selects all triangles whose vertices are solely among the original points. Under strict sampling conditions, this set of triangles contains a triangulation of the initial surface. Closely related to the crust algorithm, the cocone family of reconstructions [ACDL00] also make use of these poles of the Voronoi

diagram. However, it avoids the construction of a second Delaunay triangulation containing the poles, by selecting candidate triangles around each sample point inside the complement of a double cone in the Voronoi cell. Of great interest for their theoretical guarantees, the two previously described approaches underperform when the theoretical sampling conditions are not met for real-world acquisitions. Extensions of both algorithms (Power crust [ACK01], Tight Cocone [DG03], Robust Cocone [DG06]) assume as additional information that the reconstructed surface is closed and fall in the category of volume-based approaches.

Several other proposed surface-based approaches incrementally construct a surface from an initial seed by taking local decisions on triangles neighboring the current surface. One well-known reconstruction algorithm of this type is the ball-pivoting algorithm [BMR+99]. More advanced methods might consider different rules for the selection of neighboring triangles. For instance, in the greedy algorithm [CD04], the neighboring triangles should not create topological singularities and should form a small dihedral angle in order to garantee a certain smoothness of the output surface. Each iteration chooses the triangle that has the smallest radius for its smallest empty circumscribed ball. The locality of the decision allows for efficient processing of large input sets, but does not provide theoretical guarantees on the reconstruction quality.

Volume-based methods are usually more robust by reconstructing a surface as the boundary of a subset of tetrahedra of the Delaunay triangulation. However, an immediate consequence of this formulation is that this approach is not applicable to the reconstruction of open surfaces. One of the earliest contribution to the problem of surface reconstruction, referred to as the sculpting algorithm [Boi84], uses a priority queue to eliminate tetrahedra starting from the convex hull of the set of points. The order on the priority queue is based on a geometric criteria as well as an additional combinatorial rule enforcing that the surface has genus 0. Another algorithm guaranteeing to reconstruct a triangulation under strict sampling conditions is based on natural neighbors [BC00]. The reconstruction in this method is based on the zero set of a function, defined for each point as the average of the distances to the tangent planes of its natural neighbors. Many other approaches [Ede03, Cha03] have been proposed with no theoretical guarantees but performing well on real-world data inputs.

Additionally, numerous reconstruction approaches have been formulated as graph-cut problems. For instance, the spectral surface reconstruction [KSO04] attempts to classify Voronoi cells into interior and exterior in order to extract the surface at the frontier. The spectral clustering used is closely related to normalized cut problems introduced in [JM00]. Graph-cuts have also been used either to reconstruct a surface in a banded region around an isosurface [HK06], or as a global min-cut problem [LB07]. Labatut et al. use a graph-cut based reconstruction in the Delaunay triangulation of the set of points with an optimization based on photo-consistency [LPK07] for Structure-From-Motion point clouds or visibility information [LPK09] for laser scans. These approaches use additional information to avoid the common problem of obtaining an empty surface as solution for a global graph-cut optimization.

In general, Delaunay interpolatory methods have been extensively studied as they can provide theoretical guarantees on the reconstruction of a surface, given conditions on the sampling and the sampled surface (curvature, reach or local feature size). However, these theoretically-guaranteed methods generally behave poorly on real-world data acquisitions, where these strict sampling conditions do not apply. Robust methods are often formulated as volume-based approaches, where the surface is obtained as the boundary of a set of tetrahedra. A few methods have therefore been formulated as the graph-cut on the Delaunay triangulation. The main disadvantage of interpolatory methods is often that the resulting surface contains many triangles. For noisy sets of points, surfaces can also be bumpy and overly complex.

2.1.2 Implicit functions

A second family of surface reconstruction methods defines a function over 3-dimensional space from the input set of points and extract its zero set in the form of a triangle surface mesh, for instance by using marching cubes [LC87].

One of the first approaches contributed by Hoppe et al. [HDD+92] defines a function that associates to any point in space, the signed distance to its projection on the tangent plane of its closest input point. This method requires precise normal estimation – in particular the coherence of normal *orientations* impacts greatly the signed distance function. Consequently, other methods have been proposed either for more specific contexts [CL96] or to render the method more robust. For instance, to circumvent the difficult problem of propagating normal directions on an input set of points, the "signing the unsigned" approach [MDGD+10] chooses to transform a robust unsigned distance function into a signed distance function, by constructing an interior/exterior labeling.

Radial basis functions are commonly used for interpolation problems. For surface reconstruction [CBC+01], they can be efficiently solved and when working with globally-supported basis functions, the reconstruction is globally smooth, creating watertight surfaces even when dealing with non-uniform and incomplete data. They require however off-surface constraints, i.e. points on either side of the surface, which can be difficult to automatically estimate for noisy configurations. Techniques such as cone carving [SSZC10] can help discovering off-surface constraints.

Methods can also estimate an indicator function, whose gradient best approximates the normal field associated to the set of points. This minimization can be restated as a Poisson equation, and solved by a locally supported radial basis function in an adaptive octree [KBH06]. The method has been later improved in terms of computation efficiency [BKBH07, MPS08] and to incorporate the points as interpolation constraints [KH13].

Many recent methods have also proposed to learn these implicit functions [HTM17, GCV+19, MON+19, MPJ+19, PFS+19]. These data-driven approaches are interesting as they could derive better reconstruction priors in specific contexts. However, the main difficulty of these approaches lies in finding a concise representation for 3-dimensional space compatible with deep learning methods.

Overall, implicit functions, and especially screened Poisson reconstruction [KH13], have become popular approaches for reconstruction from points obtained via photogrammetry. They offer a good compromise between the fidelity to input data and robustness to noise and outliers. Contrary to interpolatory approaches, the number of output triangles is not correlated to the number of input points and can therefore produce lighter meshes. One major inconvenient, however, lies in the smoothness of the produced result, which is not well suited to the significant presence of sharp features in urban scenes.

2.1.3 Mesh simplification

We now detail approaches that consider simplifying an initial dense mesh in order to deduce a concise version useful in the context of urban scenes.

Often motivated by visualization applications, improvements in the field of simplification have focused mainly on efficiency and scalability [Lin00, LGL+09, CA15, LTB19] or visual coherence between the simplified versions and their original counterparts [GBK03, GMH+20].

Most efficient approaches consists in creating a sequence of edge collapses until reaching either a target mesh complexity or a maximum error deviation. The main difficulty of mesh simplification is propagating local information in the simplified meshes. Garland et al. [GH97] propose to associate to each vertex a Quadric Error Metric (QEM), characterizing its local error. When performing a collapse, the optimal contraction position is computed from the metrics and errors are propagated by adding the quadrics

associated with the edge collapse. These simplification schemes are based on a sequence of local decisions. However, the large reduction factor required to obtain meshes of similar complexity to GIS expectations suggests more global approaches. This can mean preserving piecewise-planar structures of objects by guiding edge contractions using predetected planar shapes [SLA15]. When the whole mesh can be represented by planar shapes and the adjacency graph of these primitives is correctly extracted, primitives can be assembled into a concise mesh [CAD04].

Among specialized approaches for urban scenes, Kada proposes a method for decomposing a building into structural parts before replacing them by 3D idealized primitives [Kad07]. Bredif et al. utilize a kinetic model to make facets of an existing model more consistent with the data [BBPM08]. Verdie et al. [VLA15] assemble planar shapes that are filtered to produce output meshes with different levels of detail. More recently, Li et al. [LN21] propose to first relocate the mesh vertices using a bilateral filtering of the normals, making the primitive detection more robust and allowing to apply edge contractions inside each primitive. These approaches yield good results when the original meshes are geometrically and topologically accurate but fail to correctly simplify meshes that were originally constructed from defect-laden data.

Finally, a few methods propose to construct hybrid meshes, which mix sparse and dense representations in the same model. By combining a mesh segmentation into primitives and sampling steps, Lafarge et al. [LKBV13] propose an iterative refinement procedure which creates a representation combining regular regions represented by primitives and detailed regions represented by a dense meshes. Another approach by Holzmann [HMFB18], inspired by the graph-cut approach by Labatut [LPK07], inserts detected planar primitives into the 3-dimensional Delaunay triangulation to construct a hybrid model. These methods are interesting to simplify meshes obtained from Structure-from-Motion, but their hybrid representation does not fit well with usual representations of geometry in GIS.

2.2 Urban-specific data-driven approaches

Data-driven or *bottom-up* approaches often rely on the detection of primitives (segments or planes) and the consolidation of these primitives into a concise model. In contrast to previously described methods, no dense mesh has been computed and therefore discovering the adjacency of primitives from the point cloud is more challenging. The particularity of urban scenes and the requirements of LOD2 modeling imply that most of these methods use a space decomposition, either of the 2-dimensional horizontal plane or of the 3-dimensional space. The challenge in both types of decomposition is finding a good trade-off between the complexity of the decomposition and the capability of the partition to complete missing data.

2.2.1 2D decomposition

2D space partitioning data structures have been used routinely for urban reconstruction [MWA+13]. Commonly constructed from unorganized measurement data such as point clouds or images, the 3-dimensional representation is obtained by lifting these 2-dimensional partitions to represent urban objects such as buildings or facades with a simple disk-like topology.

The 2D decomposition can again be obtained using a dense or sparse representation. Delaunay triangulations of the point cloud projected in the 2D plane offer dense partitions that need to be simplified in order to produce concise output meshes. Reducing the size of the triangulation is achieved, for instance, by contracting edges with an optimal transport approach [dCAD11], splitting and merging triangles [GS97], or inserting

and removing vertices within a spatial point process framework [FLBA20]. However, such operations alter the alignment of the triangulation with the input data and make the subsequent reconstruction step unable to produce meshes that are both concise and geometrically accurate [BRVG15]. Voronoi diagrams or adaptive grids also exhibit this weakness. The former, used in [DL16], does not allow complex buildings to be accurately reconstructed by a few Voronoi cells lifted in 3D. The latter can only reconstruct buildings accurately with dense grids [ZN10].

2D partitions of polygons offer a better tradeoff between data fidelity and conciseness on urban objects. For instance, the roof section of a building can be ideally abstracted by a single polygon. Such partitions can be created by kinetic simulations that propagate line-segments aligned with the data [Gui04, BL18] or by the vectorization of region maps [AS17]. However, their use to reconstruct buildings [ZBKB08, BL19] or facade objects [RBDD18] is effective only when the partitions are simple and preserve the geometric regularities inside objects and scenes, such as parallelism, orthogonality or symmetry. The partition can be refined by merging and splitting polygons [LLM20]. Most approaches add regularity priors by either globally snapping along the axis [ZN10, HK12], clustering 2-dimensional segments sharing the same directions [ZBKB08, ZN08] or optimizing a fidelity/regularity trade-off between spatially-close segments [BL19].

2.2.2 3D decomposition

Urban models can also be obtained by detecting planes from the initial set of points and creating a polyhedral arrangement of these planes in 3D. When all intersections are considered, which grows as a cubic to the number of planes, a lot of care needs to be taken on the number of initial primitives [BdM14, MMP16] and these approach have mostly found applications in relatively small space such as indoor scenes. Approaches for urban scenes require to make these intersections more local: the intersections of a plane primitive can for instance be limited to the super-pixels of a coarse volumetric grid containing the primitive [CLP10]. This cuts down the $\mathcal{O}(n^3)$ arrangement's complexity but depends on the order of insertion of the primitives. Again, these methods can also integrate regularity priors at the level of primitives. For instance, Monszpart et al. [MMBM15] propose an energy minimization framework with priors that encourage regularities for the extraction of planar primitives from point clouds.

2.3 Urban-specific model-driven approaches

We finally describe model-driven or *top-down* approaches, that consists in constructing a parameterized model that best fits the input observations. One early attempt by Maas et al. [MV99] uses invariant moments to estimate the parameters describing a symmetric gable roof. Defining a library of different building models leads to optimizing non-convex functions. Given enough time, Markov Chain Monte Carlo (MCMC) techniques [HBS11, VL12, HGSP13] can find a good configuration of parameters. A deep learning approach [ZWF18] was also proposed to jointly classify the building type and estimate the model parameters. Model-driven methods are however often limited by the expressiveness of the library of shapes they rely upon and cannot cope with inputs that are too different from this library.

Attempts have been made of fitting models using Roof Topology Graph (RTG): graph vertices represent detected planar primitives, and an edge between two vertices is added to the graph when the corresponding primitives have been detected to be adjacent. From a small library of known graphs for simple shapes, a complex RTG can be decomposed into simple shapes [VKH06]. This requires sub-graph matching, which is a NP-complete problem and limits the size of the graphs that can be dealt with.

Nevertheless, this method has the advantage of being more expressive than previously described model-driven approaches, as graph parts that are not detected as matches of a simpler shape can still be reconstructed. This has been further developed with a more complex library of shapes [EV09]. The whole method relies however on the correct detection of primitives and their relations: an error in the RTG can result in a badly matched subgraph. The graph-edit dictionary [XOEV14] attempts to solve this limitation, by recovering missing primitives or edges in the graph from a dictionary of common graph errors.

2.4 Contributions

Two independent approaches for reconstruction from point clouds have been investigated in this work, with radically different points of view intended to cover a large number of use cases.

2.4.1 LOD2 reconstruction using simplified 2D partitions

We make the same assumptions as the LOD2 model and create a piecewise-planar reconstruction of the buildings.

Our approach is data-driven and uses an intermediate 2-dimensional decomposition. We give a few reasons to justify this choice. First, we believe data-driven methods, compare to model-based methods, have the best chance of adapting to different urban environments. Also, formulating the problem in 2D helps to reduce the complexity of the problem and allows to apply the method at larger scales. Finally, this intermediate representation in 2D can also be enriched from pre-existing GIS information (i.e. cadastral or semantic information). We found however that these 2-dimensional partitions are often too complex, especially when dealing with point clouds from Structure-From-Motion. The resulting 3-dimensional representations are then slightly too complex for GIS expectations. Our main contribution is a simplification scheme for 2D partitions, formulated as an optimization trade-off between fidelity to the original decomposition and objectives striving for simplicity and regularity.

Compared to other simplification methods, the objectives are defined locally thanks to the connectivity of the partition, whereas axis-aligned snapping [ZN10,HK12] or angle clustering methods [ZBKB08, ZN08] are too global and can often produce large errors. The proposed approach can also simplify small cells of the partition in a continuous optimization, which was previously only considered using discrete operations [LLM20].

2.4.2 Dense mesh from optimal simplicial chains

The second contribution presented in this work is focused on obtaining dense meshes for urban applications. It studies optimal simplicial chains for its application to point cloud triangulation, where optimality is defined as a lexicographic order on chains. Algorithms for the computation of lexicographic optimal chains in arbitrary dimensions are first studied to highlight connections with matrix reduction algorithms used for the computation of persistent homology. Using duality and efficient data structures, meshing schemes with quasi-linear time complexity are described for the reconstruction of closed and open 2-surfaces in 3-dimensional space.

This reconstruction approach falls in the category of Delaunay-based interpolatory reconstructions. This means that the resulting mesh contains most of the raw data points and is therefore far from being a parsimonious representation compared to implicit functions. However, we argue that having an interpolatory reconstruction can be appreciated for applications such as inspection, where a great deal of care is generally

taken to control the noise in the input data. Defects in the acquisition therefore need to be captured, as well as very small details that might be relevant to the task of inspection. Although not explored in this work, this precise mesh could also be considered as an initialization for further simplification or regularity detection methods, as described in Section 2.1.3.

The different algorithms developed in this contribution are formulated as global optimization according to a lexicographic order. The quantities involved in the computation of this lexicographic order – radius of the smallest enclosing and circumscribing balls of triangles – are natural quantities for measuring triangles and bear resemblances to the criteria used either in the carving algorithm [Boi84] or in the greedy algorithm [CD04]. The global lexicographic optimization along these quantities will be strongly justified by the connections between lexicographic optimal chains and Delaunay triangulations. Similarly to the way we derive a surface reconstruction method from a characterization of Delaunay triangulations, the tangential Delaunay [BF04, CDE+00] generalizes Delaunay complexes to arbitrary metric spaces and is then used as a reconstruction method for submanifolds [BG14].

The presented reconstruction methods can be characterized as volume-based approaches. We noted previously that volume-based approaches are often more robust to noise or incomplete data, as this formulation essentially adds a topological constraint on the reconstruction compared to surface-based methods. We also note that previously described graph-cut approaches required additional information (photo-consistency or visibility) to avoid finding the empty surface as an optimal solution: in our case, this additional input will be explicitly given either as an interior tetrahedron for closed surface reconstruction or as the expected boundary for open surface reconstruction. Finally, previous volume-based methods were only used to reconstruct watertight closed surfaces. Quite surprisingly, the proposed open surface reconstruction, where an input 1-boundary needs to be provided, can still be seen as a volume-based method, where the volume bounds the difference between the resulting chain and a transient constructed chain.

2.4.3 Publications & Patents

This thesis is supported by the following publications and patents:

— David Cohen-Steiner, André Lieutier, and Julien Vuillamy. Lexicographic Optimal Homologous Chains and Applications to Point Cloud Triangulations. Presented at the *36th International Symposium on Computational Geometry*, 2020.

— David Cohen-Steiner, André Lieutier, and Julien Vuillamy. Regular triangulations as lexicographic optimal chains. Submitted to the *Discrete & Computational Geometry* journal, 2020.

— Two patents, for closed and open surface reconstruction processes, have been filed, 2019 & 2020.

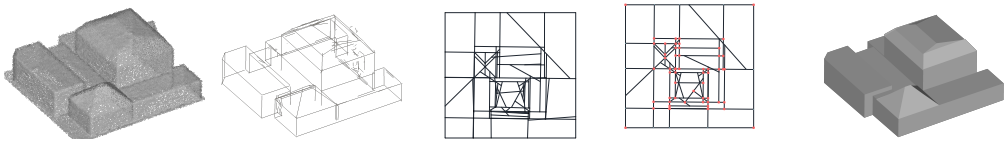
— Julien Vuillamy, André Lieutier, Florent Lafarge, and Pierre Alliez. Simplification of 2D Polygonal Partitions via Point-Line Projective Duality, and Application to Urban Reconstruction. Accepted for fast-track publication in the *Computer Graphics Forum* journal, 2021.

Chapter 3

Parsimonious representations from 2D partitions

Contents

3.1 Overview



(a) (b) (c) (d) (e)

Figure 3.1: Overview of our method. From an input point cloud (a), by extracting 3D line-segments (b) and projecting them onto the horizontal plane to form a polygonal partition (c), our method simplifies the partition while capturing some geometric regularities highlighted in red (d). The resulting partition is then lifted in 3D to create a concise polygon surface mesh (e).

The proposed method generates models compatible with Geographic Information Systems (GIS) from point clouds typically generated from photogrammetry or laser scanning. It uses an intermediate 2-dimensional representation and creates a reconstruction as a 2-dimensional piecewise-linear function over this 2D representation. This constraints the type of buildings that this method is able to correctly model. In particular, overhangs cannot be represented. However, this output representation fits exactly with the type of reconstruction expected for LOD2 representations of buildings in GIS.

Figure 3.1 depicts the main steps of our method. Section 3.2 starts by describing the reconstruction process without simplification (steps (a), (b), (c) and (e) in Figure 3.1). We will also motivate the need for simplification of the intermediate 2-dimensional arrangements. Section 3.3 proposes a simplification optimization performed on 2-dimensional partitions. Finally, Section 3.4 gives examples of reconstruction of urban scenes and discusses the usefulness and limitations of the proposed simplification process.

3.2 Reconstruction pipeline

We start by detailing a reconstruction pipeline that combines many ideas already explored for the reconstruction of urban scenes.

3.2.1 Extraction of 3D line-segments

Planar primitives and their adjacency relationships are first detected in the original point cloud using a k-nearest neighbors algorithm and a region-growing approach [FTK14, HB13, RvdHV06]. For all adjacent planes, which can be recovered from a k-nearest neighbors, intersection lines are computed and 3D segments are generated from the points located near each intersection line. The segments are then projected in the 2-dimensional horizontal plane. The problem is from now on considered in 2D. However, we keep some 3D information about the segments. In particular, we identify all segments that are at the intersection of a vertical plane with a non-vertical plane, as to locate the possible building facades.

3.2.2 Kinetic cell arrangement

In order to obtain a parsimonious 2-dimensional polygonal partition from projected segments, we now describe a method called kinetic framework [Gui04, BL18]. By extending

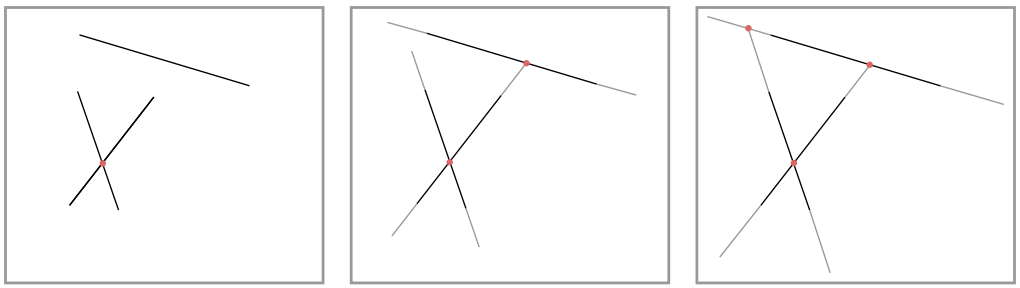


Figure 3.2: Illustration of the kinetic rules. From left to right: initial segment intersections are created at time $t = 0$. A prolonged segment intersecting an initial segment is stopped. When two prolonged segments intersect, the last to arrive on the intersection is stopped.

all line segments, intersection events are used to decide whether to stop one of the two line-segments forming the intersection. More specifically, for a segment S, any point I on its supporting line can be assigned a *Time of Arrival*, which equates to the distance to the segment for a constant unit speed movement:

$$T_S(I) = d(I, S)$$

The kinetic framework can be seen as a set of rules dictating the segments' progression. In its simplest form, the progression of a segment in one direction is stopped at an intersection event when it arrives last on the intersection. For instance, a segment S_1 will be stopped at its intersection I with S_2 if $T_{S_1}(I) > T_{S_2}(I)$. All intersection events I' concerning S_1 in the same direction as I and verifying $T_{S_1}(I') > T_{S_1}(I)$ can then be discarded, as segment S_1 will not cross segment S_2. These rules are illustrated in Figure 3.2.

Additional rules can be designed to improve the connectivity of the arrangement, such as setting a custom speed for each segment or allowing a fixed number of crossings. We found particularly useful to allow crossings when the time of arrival to the next intersection is smaller than a threshold value. This makes the connectivity of the partition less dependent on small perturbations in the detected segments.

Arrangements resulting from the kinetic framework exhibits substantially fewer intersections than the complete line arrangement containing $\mathcal{O}(n^2)$ intersections. It also better preserves details, as small segments are kept in the arrangement but are likely stopped by larger features and therefore will only have a local impact on the arrangement. Finally, an important property of the resulting partition is that initial segments spanning large distances are often decomposed into multiple smaller segments in the arrangement, while still sharing the same supporting line.

3.2.3 Lifted model

Given a 2D polygonal partition and a point cloud $\mathbf{P}$, a lifted model can be created by assigning to each cell of the partition the plane that best fits the points projecting in the cell. However, this approach is not resilient to imprecise partitions, missing data or noise in the input points. Instead, using the set of detected planes Π in Section 3.2.1, a discrete optimization is performed to obtain a coherent lifted model.

Denote by $\mathcal{C}$ the set of cells of the partition. We call labeling a map $\mathcal{L} : \mathcal{C} \mapsto \Pi$ that associates to each cell of the arrangement a plane in Π. The global energy objective is

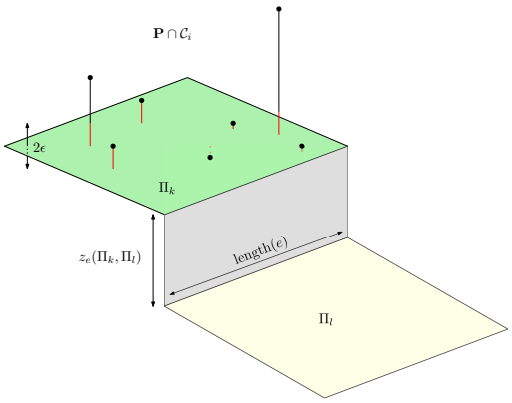


Figure 3.3: Illustration of the quantities involved in $E_{\mathcal{C}_i}(\Pi_k)$ and $V_e(\Pi_k, \Pi_l)$. For each point of $\mathbf{P} \cap \mathcal{C}_i$, the height in red corresponds to the quantity appearing in the sum of $E_{\mathcal{C}_i}(\Pi_k)$ for a fixed threshold ϵ.

expressed as follows:

$$E(\mathcal{L}) = \sum_{\mathcal{C}_i \in \mathcal{C}} E_{\mathcal{C}_i}(\mathcal{L}(\mathcal{C}_i)) + \mu \sum_{e = C_i \cap C_j} V_e(\mathcal{L}(\mathcal{C}_i), \mathcal{L}(\mathcal{C}_j))$$

The quantities involved in both terms $E_{\mathcal{C}_i}$ and V_e are illustrated in Figure 3.3.

For each cell $\mathcal{C}_i \in \mathcal{C}$, the function $E_{\mathcal{C}_i}$ will measure the fidelity between a plane of Π and the subset of points of $\mathbf{P}$ whose projection in the horizontal plane lies inside $\mathcal{C}_i$, which we denote by $\mathbf{P} \cap \mathcal{C}_i$. More formally, given a plane equation $z = \Pi_k(x, y)$,

$$E_{\mathcal{C}_i}(\Pi_k) = \frac{\mathcal{A}_{\mathcal{C}_i}}{\text{Card}(\mathbf{P} \cap \mathcal{C}_i)} \sum_{p \in \mathbf{P} \cap \mathcal{C}_i} \min(\epsilon, |p_z - \Pi_k(p_x, p_y)|) \tag{3.1}$$

where $\mathcal{A}_{\mathcal{C}_i}$ denotes the cell area of $\mathcal{C}_i$. A threshold ϵ is used on the z-distance to the plane to improve the robustness of the plane fitting in the presence of noise in the point cloud or slight errors in the cell decomposition.

The regularity term V_e between two cells $(\mathcal{C}_i, \mathcal{C}_j)$ intersecting on an edge $e = \mathcal{C}_i \cap \mathcal{C}_j$ needs to penalize the height difference along e. Given planes $\Pi_k, \Pi_l \in \Pi$ assigned to the cells on each side of the edge e, a regularity term measures a mean vertical area along the edge e:

$$V_e(\Pi_k, \Pi_l) = p_e \,\text{length}(e)\, \text{z}_\text{e}(\Pi_k, \Pi_l) \tag{3.2}$$

where z_e denotes the average vertical height along an edge $e = ((x_1, y_1), (x_2, y_2))$:

$$\text{z}_\text{e}(\Pi_k, \Pi_l) = \frac{1}{2} \sum_{i=1,2} |\Pi_k(x_i, y_i) - \Pi_l(x_i, y_i)|$$

For each edge, the parameter p_e in Equation (3.2) denotes a *prior* probability that the edge is along a building facade and can for instance be found by identifying segments at the intersection of a vertical and non-vertical plane.

The regularity parameter μ of the global optimization, homogeneous to a length, should be understood as the smallest "feature" size we want to distinguish. This objective function, decomposed as a unary fidelity term and binary regularity interactions on labels, belongs to a family of functions where local minima can be reached efficiently using graph cut optimization [BVZ01].

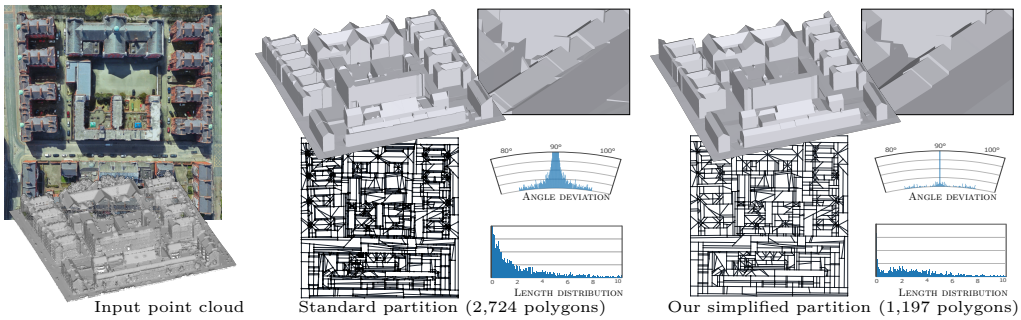


Figure 3.4: Decomposing scenes into planimetric partitions of 2D polygons is a popular approach in urban reconstruction. Traditional partitioning schemes however produce overly complex and unregularized partitions. Our approach simplifies such partitions while preserving their fidelity to data and enforcing some geometric regularities contained in the scene (see histograms depicting the angle deviation around 90° and the distribution of edge lengths in the partitions). The output model, reconstructed by lifting our partition in 3D, is both more concise and regular (see closeups).

3.2.4 Complexity of resulting models

The proposed pipeline enjoys the benefits of data-driven methods, by allowing to recover the complexity of urban scenes, without being restricted to the fixed complexity of model-driven approaches. The projection into a 2-dimensional space reduces the complexity of the problem. Using a kinetic process then creates a relatively sparse connectivity for the detected segments in a regular structure containing multiple collinear segments. The lift optimization finally confronts the 2-dimensional partition with the input point cloud in a trade-off that tends to avoid small discontinuities in the 2-dimensional piecewise-planar lift function. This successfully ignores small building features that are not relevant for a LOD2 type of reconstruction as well as noise in the input data.

The created models tend however to be slightly too complex compared to the expected representations of GIS. In Figure 3.4, we show an input point cloud on a small building complex extracted from a 2015 LiDAR survey of Dublin [LAA+17]. Although the general structure of the buildings is simple, many small features are present in the point cloud. Using the previously described pipeline, the segment detection and kinetic framework create a complex 2-dimensional partition, which result in a few errors in the lift model (middle of Figure 3.4). By observing that the 2-dimensional partition contains many small segments as well as many angles close to orthogonality, we are interest in simplifying 2-dimensional partitions by essentially removing small cells and collapsing small segments while enforcing orthogonality and preserving collinearities present by construction in the partition.

3.3 Simplification of 2D partitions

We now detail the simplification process used on 2D partitions in order to address the over-complexity presented in Section 3.2.4.

3.3.1 Global versus local minima and convex versus non-convex objectives

We start by mentioning two problems in the context of point cloud simplification in 2D.

For two non-empty set of points X, Y, the Hausdorff distance $d_H(X, Y)$ between the two subsets is defined as:

$$d_H(X,Y) \underset{\text{def.}}{=} \max\left\{ \max_{x\in X} d(x,Y),\ \max_{y\in Y} d(y,X) \right\}$$

where $d(x, Y)$ (resp. $d(y, X)$) denotes the minimal distance between the point x (resp. y) and any points of Y (resp. X).

The following two problems were shown to be NP-hard [MBH+95].

Problem 1 (Geometric unit disk cover)**.** *Given a set of points* $\mathbf{P}$ *in* $\mathbb{R}^2$ *and a distance* $\epsilon > 0$*, find a cardinal-minimal set of points* $X \in \mathbb{R}^2$ *such that* $d_H(\mathbf{P}, X) \leq \epsilon$*.*

Problem 2 (Minimum dominating set on unit disk graphs)**.** *Given a set of points* $\mathbf{P}$ *in* $\mathbb{R}^2$ *and a distance* $\epsilon > 0$*, find a cardinal-minimal subset of points* $X \subset \mathbf{P}$ *such that* $d_H(\mathbf{P}, X) \leq \epsilon$*.*

These clustering problems can be seen as "point cloud simplification" problems and, as such, share some similarities with our simplification problem. However, while simpler, finding a global optimal for these problems has been proven to be NP-hard. For this reason, in practice, efficient point clustering algorithms only search for local minima: for instance, mean-shift or k-means iterations can be interpreted as steps of a gradient descent toward a local minimum. As it is already the case for these simpler problems, we do not expect the existence of a global minimum formulation for our simplification problem.

The following ambiguous situation also motivates us to formulate the problem solution as a local minimum of a non-convex objective. Consider the artificial and symmetric configuration where 360 segments of similar length form a perfect regular 360-gon. We wish to simplify this configuration by merging adjacent segments making an angle below 5 degrees. A practical solution must break the symmetry and hence make some arbitrary choices, as would mean-shift for points regularly spaced on a circle. Such a desired behavior requires a non-convex objective, since, by symmetry, several well separated local minima are equivalent.

Finally, the choice of a continuous optimization seems more promising than a sequence of discrete operations. Indeed, as mentioned in Section 3.2.2, the partition created by a kinetic process contains multiple collinear segments (as T-junctions). Keeping these regularities, as well as enforcing orthogonality or parallelism, fits better with a continuous optimization scheme.

3.3.2 Projective duality: line movement vs. point movement

We use the point-line projective duality to describe a partition. Each element is described as a 3D vector representing:

— The coefficients (a, b, c) of an oriented line $ax + by + c = 0$;

— The homogeneous coordinates (x, y, w) for a 2D point $(\frac{x}{w}, \frac{y}{w})$.

This duality is especially visible in the symmetry of the roles played by points and lines in the line equation written as an inner product:

$$\begin{pmatrix} a & b & c \end{pmatrix} \begin{pmatrix} x \\ y \\ w \end{pmatrix} = 0$$

A comprehensive description of projective duality can be found in [Ber94].

We summarize two possible representations of 2D partitions, dual to one another in projective geometry:

— *point representation*: points of the partition are given explicit coordinates and lines are implicitly defined from two of these points. This representation allows to encode in the combinatorial structure the situation where multiple lines share a same point, but not the situation where more than two points are supported by a single line.

— *line representation*: lines of the partition are given explicit coordinates and points are implicitly defined as the intersection of two of these lines. This dual representation, symmetrically, encodes in the combinatorial structure the situation where a line carries multiple points but not the dual configuration for which more than 2 lines intersect at the same point.

In terms of optimization, dealing with point coordinates is simpler than line coordinates, as the Euclidean metric associated to the distance between 2-dimensional points is natural. Optimizing via line coordinates is more involved as no natural metric describes a distance between lines which is invariant under arbitrary Euclidean isometries. Appealed by its apparent simplicity, we have tried to use the point representation in a first attempt. However, two arguments justify the choice of optimizing lines instead of points in our specific context and were confirmed by experiments on the point representation.

Firstly, the initial data for our problem consists of a set of detected line segments, while points are only secondary data constructed from the initial segments by the kinetic framework. Measuring data fidelity with respect to the initial segments seems therefore more natural to our problem than data fidelity with respect to point coordinates.

Secondly, by construction, many segments of the partition exhibit collinearity relationships, that ought to be preserved for the sake of keeping the partition as simple as possible. In a point-based representation, this requires optimizing under collinearity constraints. However, as depicted in the inset, configurations close to edge collapses, which were our objectives in a point-based representation, are also configurations where collinearity constraints are unstable: constraining three points to be aligned when two of them are indistinguishable from one another is not a well-posed problem. In our first experiments, these frequent configurations resulted in numerical instabilities whose robust treatment was problematic. In contrast, in the line representation model, these collinearities are naturally preserved in the structure.

The homogeneous line coordinates are initialized via Euclidean normalization, also referred to as its normal form: $a^2 + b^2 = 1$. Although the *spherical* normalization ($a^2 + b^2 + c^2 = 1$) is used for its ability to represent all projective lines, our problem does not require to represent the line at infinity and we hence adhere to Euclidean normalization. We will further justify this choice when giving a geometrical meaning to the regularity term of our optimization energy. Figure 3.5 depicts the unit cylinder structure of the line coordinates with Euclidean normalization.

3.3.3 Energy formulation

We denote by $\mathcal{L} = (L_1, \ldots, L_n) \in \mathbb{R}^{3n}$ the set of lines, represented in their normal form. Our simplification problem is formulated as a trade-off between a fidelity term $E_{fidelity}$, describing the attachment of the partition to the initial configuration, and complexity terms $E_{concurrent}$ and $E_{orthogonality}$, measuring respectively edge collapses and orthogonality objectives on the lines of the partition:

$$E(\mathcal{L}) = E_{fidelity}(\mathcal{L}) + \lambda_1 E_{concurrent}(\mathcal{L}) + \lambda_2 E_{orthogonality}(\mathcal{L}) \tag{3.3}$$

We now provide details on each term of this objective function.

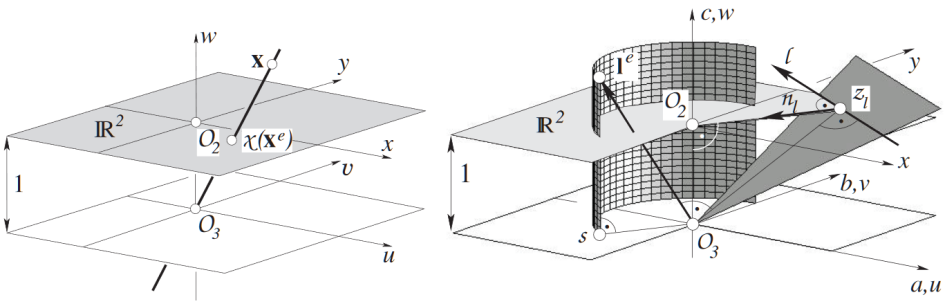


Figure 3.5: Illustration of Euclidean normalizations. Left: the Euclidean normalized coordinates of a projective point $\mathbf{X}$ correspond to the intersection of the $w = 1$ plane and the line passing through O_3 and $\mathbf{X}$. Right: the Euclidean normalized coordinates l^e of the projective line l lies on the cylinder verifying $a^2 + b^2 = 1$. Image taken from [FW16].

Fidelity term. As mentioned previously, no metric defines a distance between lines which is invariant under Euclidean isometries. Indeed, the only discrepancy measures on lines that are preserved by Euclidean isometries are the angle between two lines and the distance between two parallel lines.

In our problem, we have the additional information that each line L is associated to an initial detected segment $S = ((x_0, y_0), (x_1, y_1))$. This can be used to define a distance from the initial configuration. Consider a point $P = (x, y)$ on a line L parametrized by (a, b, c).

$$ax + by + c = 0 \tag{3.4}$$

Under line movement $\delta L = (\delta a, \delta b, \delta c)$, the squared distance of the point P to the line $L + \delta L$ can be written as:

$$\begin{aligned} d(P, L + \delta L)^2 &= (\delta a \cdot x + \delta b \cdot y + \delta c)^2 \\ &= \delta L^t \begin{pmatrix} x^2 & xy & x \\ yx & y^2 & y \\ x & y & 1 \end{pmatrix} \delta L \end{aligned}$$

This quantity measures the square of a first order approximation of the distance between point P and line $L+\delta L$, which is exact only when $L+\delta L$ satisfies the Euclidean normalization $(a + \delta a)^2 + (b + \delta b)^2 = 1$.

For each line L associated to a segment $S = ((x_0, y_0), (x_1, y_1))$, we consider the sum of these approximated squared distances between the line $L + \delta L$ with the two initial segment endpoints:

$$\frac{1}{2} \begin{bmatrix} x_0^2 + x_1^2 & x_0 y_0 + x_1 y_1 & x_0 + x_1 \\ x_0 y_0 + x_1 y_1 & y_0^2 + y_1^2 & y_0 + y_1 \\ x_0 + x_1 & y_0 + y_1 & 2 \end{bmatrix}$$

Note that by definition, the kernel of this matrix contains all homogeneous coordinates of the line passing through the points (x_0, y_0) and (x_1, y_1).

The quadratic form associated with this positive semi-definite matrix offers a good balance between the translation and rotation movements depending on the segment vertex locations: the rotation of a line which is associated to a large segment will be more penalized compared to the rotation of the same line associated to a shorter segment. However, in the extreme case where the segment used for the definition of the quadratic

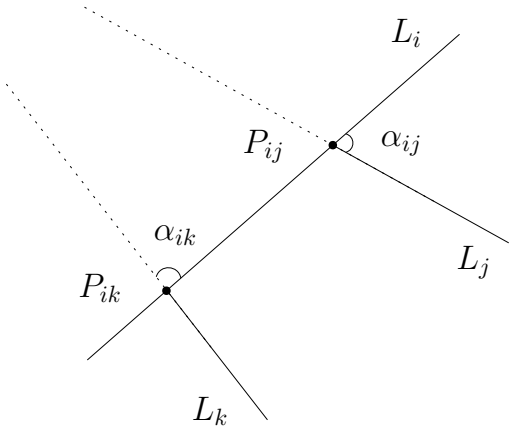


Figure 3.6: Illustration of a detected regularization configuration: three lines L_i, L_j, L_k form two intersections in the partition.

form has zero length, the kernel of the matrix contains all homogeneous coordinates of lines that pass through this unique point $(x_0, y_0) = (x_1, y_1)$. In the optimization, this means lines which are associated to very small segments can freely rotate, which can cause numerical instabilities. We stabilize our quadratic form by adding a small penalty l^2_{min} to the rotation part of all matrices and finally define the matrix M_i for each line L_i in $\mathcal{L}$:

$$M_i \underset{\text{def.}}{=} \frac{1}{2} \begin{bmatrix} x_0^2 + x_1^2 + l^2_{\text{min}} & x_0 y_0 + x_1 y_1 & x_0 + x_1 \\ x_0 y_0 + x_1 y_1 & y_0^2 + y_1^2 + l^2_{\text{min}} & y_0 + y_1 \\ x_0 + x_1 & y_0 + y_1 & 2 \end{bmatrix} \tag{3.5}$$

where $((x_0, y_0), (x_1, y_1))$ denotes the segment associated to the line L_i, and l_{min} is a minimal length at which we consider segments to be relevant.

The fidelity term on all lines L_i of the arrangement is then defined as follows:

$$E_{fidelity}(\mathcal{L}) \underset{\text{def.}}{=} \frac{1}{2} \sum_{i=1}^{n} L_i^t M_i L_i \tag{3.6}$$

Observation 3.1. *Fidelity terms are usually written as a quadratic form Q applied to the difference between the current configuration X and the initial configuration X_0:*

$$(X - X_0)^t Q (X - X_0)$$

However, in our case, the information of the initial configuration of each line L_i^0 is contained in the matrix M_i. Indeed, for a line L_i^0 passing through the segment used to define M_i, we have:

$$M_i L_i^0 = 0$$

Regularization terms. We consider triplets of lines (L_i, L_j, L_k) where L_i, L_j and L_i, L_k form intersection points in the partition respectively denoted by P_{ij} and P_{ik}. We define the following quantity D_{ijk}:

$$D_{ijk} \underset{\text{def.}}{=} |\det(L_i, L_j, L_k)| = |(L_i \times L_j) \cdot L_k|$$

We illustrate such a configuration in Figure 3.6 and provide a geometrical meaning to this quantity, when L_i, L_j, L_k are (Euclidean) normalized line coordinates. The point-line projective duality gives that the cross product $L_i \times L_j$ is a homogeneous vector

representing the intersection point $P_{ij} = (x_{ij}, y_{ij})$. Therefore,

$$L_i \times L_j = w_{ij} \begin{pmatrix} x_{ij} \\ y_{ij} \\ 1 \end{pmatrix}$$

with, thanks to the choice of an Euclidean normalization for lines,

$$w_{ij} = \begin{vmatrix} a_i & a_j \\ b_i & b_j \end{vmatrix} = \sin \alpha_{ij}$$

where α_{ij} denotes the angle between L_i and L_j.

The Euclidean normalization gives also that the inner product of $L_i \times L_j$ with the vector L_k yields the (2-dimensional) distance from the point P_{ij} to the line L_k, multiplied by w_{ij}:

$$D_{ijk} = w_{ij} \begin{pmatrix} x_{ij} \\ y_{ij} \\ 1 \end{pmatrix} \cdot L_k = w_{ij}\, d(P_{ij}, L_k).$$

The distance from P_{ij} to line L_k can be expressed with the angle α_{ik} between the lines L_i and L_k:

$$d(P_{ij}, L_k) = ||P_{ij} - P_{ik}|| \sin \alpha_{ik}.$$

We get finally:

$$D_{ijk} = ||P_{ij} - P_{ik}|| \sin \alpha_{ij} \ \sin \alpha_{ik} \tag{3.7}$$

where α_{ij} and α_{ik} respectively denote the angles $\angle(L_i, L_j)$ and $\angle(L_i, L_k)$.

The invariance under isometries follows from Equation (3.7) while the invariance under permutation is inherited from the determinant expression, so that:

Lemma 3.2. *Under Euclidean normalization of the line coordinates, D_{ijk} is invariant under Euclidean isometries and permutations.*

In the specific case of two parallel lines intersected perpendicularly by a third line, this determinant is exactly the distance separating the two parallel lines. We can therefore define a largest allowed distance ϵ to consider edge collapses. By denoting $\mathcal{T}$ the set of triplets (i, j, k) corresponding to lines (L_i, L_j, L_k) forming at least two intersections in the partition, we define the following regularity objective:

$$E_{concurrent} = \sum_{(i,j,k)\in\mathcal{T}} \min\left(\epsilon, |\det(L_i, L_j, L_k)|\right) \tag{3.8}$$

We also favor orthogonality by adding an objective on the set $\mathcal{P}$ of pairs (i, j) corresponding to intersecting lines (L_i, L_j) in the arrangement:

$$E_{orthogonality} = \sum_{(i,j)\in\mathcal{P}} \min(\sin \alpha_{\max}, |\,\mathrm{dot2d}(L_i, L_j)|) \tag{3.9}$$

where dot2d denotes the inner product on the two first component of the vectors L_i and L_j, and $\alpha_{\max}$ is a tolerance angle below which the two lines are considered as orthogonal.

The choice of a L^2-like term for the fidelity objective of Equation (3.6) and L^1-like terms, that is with a gradient discontinuity at 0, for the simplification objectives of Equation (3.8) and Equation (3.9) is natural, since the fidelity must be maximized while we are looking for exact line concurrency or orthogonality when this is possible. Our simplification process thus behaves as an avatar of the celebrated sparsity of L^1 minima.

Note that taking the minimum with some threshold in Equation (3.8) and Equation (3.9) makes these objectives active in the minimization only when the configuration is close to the corresponding exact constraint satisfaction. In a more conceptual perspective, taking the min with a threshold also contributes to the desired non-convexity of the objectives motivated in Section 3.3.1.

3.3.4 Gradient descent in a Riemannian manifold

We solve our optimization problem via a gradient descent algorithm. Indeed, as discussed in Section 3.3.1, we are looking for local minima and, despite its apparent simplicity, the gradient method has proven to be a reliable descent method in particular for non-smooth and non-convex objective minimizations.

While the gradient is merely the transpose of the objective's first derivative in an Euclidean context, working on the space of lines and using homogeneous coordinates lead to several changes in the standard expression of the gradient.

We recall here a few notions of Riemannian geometry, more details being provided by Lee [Lee13]. Consider a differentiable real-valued function $f : \mathcal{M} \to \mathbb{R}$, where $(\mathcal{M}, g)$ is a Riemannian manifold with metric g. Denote by $\mathcal{T}_p\mathcal{M}$ the tangent space of $\mathcal{M}$ at a point $p \in \mathcal{M}$. The directional derivative df and the gradient $\operatorname{grad} f$ of f at a point $p \in \mathcal{M}$ verify, for $v \in \mathcal{T}_p\mathcal{M}$:

$$df(v) = \langle \operatorname{grad} f, v \rangle_g \tag{3.10}$$

For G , D, and ∇f the respective matrix representations of g, df, and $\operatorname{grad} f$, this translates to

$$\nabla f = G^{-1} D^t \tag{3.11}$$

We denote by $\mathcal{M}$ the manifold of (oriented) lines, seen as the submanifold of the space $\mathbb{R}^3$ of homogeneous coordinates with Euclidean normalization. As illustrated in Figure 3.5, it corresponds to the cylinder with implicit equation $a^2+b^2 = 1$. The gradient descent is performed in the n-fold Cartesian product manifold $\mathcal{M} \times \cdots \times \mathcal{M}$, where n is the number of lines in $\mathcal{L}$. Each of these copies of $\mathcal{M}$ is associated to a line of $\mathcal{L}$ and a positive semi-definite matrix as defined in Equation (3.5). Because of this Cartesian product, the gradient of each line can be computed independently, and we consider the copy of $\mathcal{M}$ associated to the i^{th} line. $L_i = (a, b, c)$ and M_i denote respectively the current position in normalized homogeneous coordinates and the matrix associated to this line. We also denote by D_i the derivative of the objective function E given by Equation (3.3) with respect to the homogeneous coordinates of line L_i:

$$D_i = \left(\frac{dE}{da}, \frac{dE}{db}, \frac{dE}{dc} \right)$$

The tangent space at L_i corresponds to a 2-dimensional vector space written as:

$$\mathcal{T}_{L_i}\mathcal{M} = \{ v \in \mathbb{R}^3, v \cdot (a, b, 0) = 0 \} = U_i \mathbb{R}^2 \tag{3.12}$$

with

$$U_i = \begin{pmatrix} -b & 0 \\ a & 0 \\ 0 & 1 \end{pmatrix}$$

The quadratic form associated to the matrix M_i, restricted to this 2-dimensional vector space $\mathcal{T}_{L_i}\mathcal{M}$, is non-degenerate – it can be seen as an actual Riemannian metric for the space $\mathcal{M}$ associated to the i^{th} line. We give the expression of the derivative D_i' and the matrix M_i' associated to the quadratic form in $\mathcal{T}_{L_i}\mathcal{M}$:

$$\begin{aligned} D_i' &= D_i U_i \\ M_i' &= U_i^t M_i U_i \end{aligned}$$

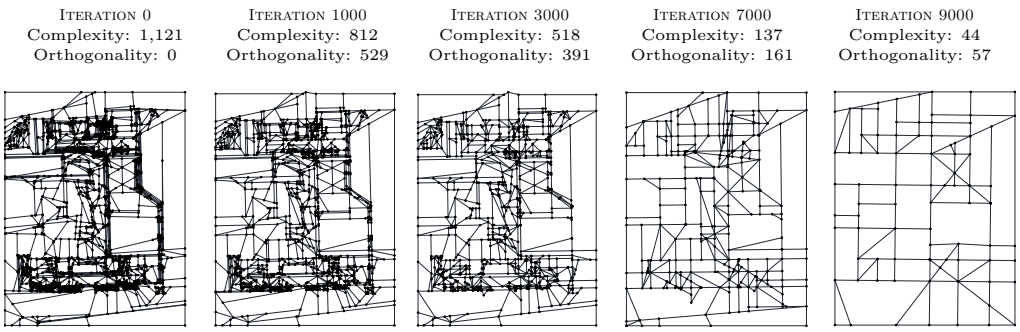


Figure 3.7: Evolution of partitions during optimization. The global simplification algorithm progressively reduces the number of lines while increasing the orthogonality between them ("complexity" and "orthogonality" refer to the number of lines and number of orthogonal pairs of lines in the partition).

Using Equation (3.11), the expression of the gradient with respect to line L_i in the tangent space $\mathcal{T}_{L_i}\mathcal{M}$ is:

$$\Big(\nabla E\Big)_{\mathcal{T}_{L_i}\mathcal{M}} = (M_i')^{-1} {D_i'}^t$$

In order to update the coordinates of the line L_i, one could in theory apply the exponential map from $\mathcal{T}_{L_i}\mathcal{M}$ to $\mathcal{M}$. We use instead a simpler procedure which is equivalent at first order. The expression of the gradient $\nabla_i E$ in homogeneous coordinates is obtained as the product of U_i mapping $\mathcal{T}_{L_i}\mathcal{M}$ to the space $\mathbb{R}^3$ of homogeneous coordinates with $\Big(\nabla E\Big)_{\mathcal{T}_{L_i}\mathcal{M}}$:

$$\nabla_i E = U_i (M_i')^{-1} {D_i'}^t \tag{3.13}$$

For a given step α, the homogeneous coordinates are updated by adding $-\alpha\nabla_i E$ and normalized, which can be seen as an orthogonal projection onto $\mathcal{M}$.

Denoting $normalize\big((a,b,c)^t\big)$ the Euclidean normalization of vector $(a,b,c)^t$, a gradient descent step consists in:

$$\forall i, \quad L_i^{(k+1)} = normalize\left(L_i^{(k)} - \alpha\nabla_i E\right) \tag{3.14}$$

3.3.5 Global Algorithm

The global simplification process is shown on Algorithm 1 where $\mathcal{C}$ and $\mathcal{L} = (L_1, \ldots, L_n) \in \mathbb{R}^{3n}$ represent respectively the combinatorial and numerical components of the line arrangement. Figure 3.7 illustrates how a polygonal partition typically evolves during the simplification process.

The step $\mathcal{L} \leftarrow$ **GradientStep**$(\mathcal{C}, \mathcal{L}, \alpha)$ consists in one Riemannian gradient step described in Equation (3.14) for each line L_i in which only the numerical data $\mathcal{L}$ is updated.

The instruction $\alpha \leftarrow$ **Update**(α) governs the choice of the step α. The analysis of the rate of convergence of gradient methods for non-smooth, non-convex objectives [KW18] is a difficult question that has not been studied in this work. Many techniques, for either choosing the step such as line search methods [NW99] or improving the convergence speed such as momentum [Qia99, Nes83], are taking on renewed importance with the popularity of deep learning. Experimentally, although more elaborate schemes could improve the speed of convergence, a geometric sequence $\alpha_i = \alpha_0 r^i$, with a ratio r

smaller but close to 1 was enough to successfully converge to local minima. We want the stopping criterion $\alpha_{\min}$ to be small enough as to bring some regularization terms below a certain threshold η.

The step $NbReductions, \mathcal{C}, \mathcal{L} \leftarrow$ **Reductions**$(\mathcal{C}, \mathcal{L})$ in Algorithm 1, applies a combinatorial reduction step, that updates both the combinatorial structure $\mathcal{C}$, by decreasing the number n of lines and updating the line intersection relations as well as the list of coordinates $\mathcal{L}$ of new lines resulting from merge decisions. It returns the integer variable "$NumberOfReductions$", which denotes the number of achieved merges and is utilized in the termination test.

Algorithm 1: Global simplification algorithm

Inputs : $\mathcal{C}$: combinatorial structure,
$\mathcal{L}$: list of lines coordinates,
α: step.
Output: $\mathcal{C}$, $\mathcal{L}$.
do
  do
    $\mathcal{L} \leftarrow$ **GradientStep**$(\mathcal{C}, \mathcal{L}, \alpha)$
    $\alpha \leftarrow$ **Update**(α)
  while $\alpha > \alpha_{\min}$
  $NumberOfReductions, \mathcal{C}, \mathcal{L} \leftarrow$ **Reductions**$(\mathcal{C}, \mathcal{L})$
while $NumberOfReductions > 0$

Robustness of combinatorial reductions. It is well known that optimization with L^1 regularization leads to sparser solutions than its L^2 counterpart, which in our case translates into edge collapses, orthogonalities and line fusions. Although the line representation does not enable collapsing edges – as this would require multiple lines to pass through the same point – it is useful to interlace, as shown in Algorithm 1, optimization iterations with combinatorial line merges and thus continue optimizing in a simplified partition. Such an entanglement of numerical and combinatorial processes may lead to what is known in computational geometry as robustness problems: combinatorial decisions require a perfect consistency whereas numerical computations produce only approximations.

If the numerical gradient descent was able to perfectly cancel a determinant formed by 3 lines $|\det(L_i, L_j, L_k)| = 0$, then at least one of the two following projective situations would occur: (a) two lines are equal or (b) the 3 lines differ but meet at a same single point. In the language of affine geometry, situation (b) splits into the case of 3 lines that meet at a single affine point (b1) and 3 lines that meet at infinity, in other words 3 parallel lines (b2). However, what becomes this observation when replacing $|\det(L_i, L_j, L_k)| = 0$ by $|\det(L_i, L_j, L_k)| < \eta$ for a small threshold $\eta > 0$?

In order to take the right combinatorial decisions when merging lines, while using the finite accuracy from numerical computations, we need a carefully quantified version of the previous "exact" implication of one of situations (a), (b1) and (b2) so that, when lines (L_i, L_j, L_k) form a regularization term $|\det(L_i, L_j, L_k)|$ smaller than a threshold value η, we can guarantee that one of the following situations occur:

— two (or more) lines of the triplet are indistinguishable up to some quantified accuracy d_L, corresponding to exact situation (a)

— two (or more) points at the line intersections are indistinguishable up to some quantified accuracy d_P corresponding to exact situation (b1)

— the intersection points are outside a disk of radius R corresponding to exact situation (b2)

We relate this threshold η in the continuous optimization to values used in discrete combinatorial operations: a minimal distance d_P at which points are considered identical, a minimal quantity between line vectors d_L at which they are considered identical and the radius R of a disk centered at 0 containing the partition:

$$\eta = \frac{d_P \, d_L^2}{5(1+R^2)} \tag{3.15}$$

Proof. By translating the arrangement such that the origin is at its center, we make the assumptions that all lines as well as all points of the arrangement are at a distance from the origin bounded by R, where R represents the half-size of the arrangement. We want to determine a value η such that when a regularity criteria D_{ijk} on a triplet of lines (L_i, L_j, L_k) is smaller than η, we can guarantee that either two lines L, L' of the triplet verify $||L - L'|| < d_L$ or that the two points at the intersection of (L_i, L_j) and (L_i, L_k), denoted respectively P_{ij} and P_{ik}, have a distance smaller than a chosen length d_P.

To that effect, we assume that $D_{ijk} < \eta$ and all pair of lines L, L' in the triplet of lines verify $||L - L'|| > d_L$ and we expect that $||P_{ij} - P_{ik}|| < d_P$.

Consider two lines L, L' represented respectively by homogeneous coordinates (a, b, c) and (a', b', c'). Recall that their intersection point P has homogeneous coordinates:

$$P = L \times L' = \left(\begin{vmatrix} b & b' \\ c & c' \end{vmatrix}, \begin{vmatrix} a & a' \\ c & c' \end{vmatrix}, \sin\alpha \right)$$

where α denotes the angle between L and L'. Therefore, using the normalization $a^2 + b^2 = a'^2 + b'^2 = 1$:

$$\begin{aligned} OP^2 \sin^2\alpha &= (bc' - b'c)^2 + (ac' - a'c)^2 \\ &= (a^2 + b^2)c'^2 + (a'^2 + b'^2)c^2 - 2cc'(aa' + bb') \\ &= (c' - c)^2 + 2cc'(1 - \cos\alpha) \end{aligned}$$

And finally,

$$(c' - c)^2 = OP^2 \sin^2\alpha - 4cc' \sin^2\frac{\alpha}{2} \tag{3.16}$$

The assumption that the point P as well as the lines L and L' have a distance to the origin upper bounded by R can be translated into:

$$\begin{cases} OP \leq R \\ |c| \leq R \\ |c'| \leq R \end{cases}$$

Combined with Equation (3.16), we get:

$$(c' - c)^2 \leq R^2 \sin^2\alpha - 4cc' \sin^2\frac{\alpha}{2} \leq 5R^2 \sin^2\alpha$$

From the assumption that $d_L < ||L - L'||$,

$$\begin{aligned} d_L^2 &< (a' - a)^2 + (b' - b)^2 + (c' - c)^2 \\ &< 4\sin^2\frac{\alpha}{2} + 5R^2 \sin^2\alpha \\ &< 5(1 + R^2)\sin^2\alpha \end{aligned}$$

The last inequality holds for $\alpha = \alpha_{ij}$ and for $\alpha = \alpha_{ik}$ so that:

$$\sin \alpha_{ij} \sin \alpha_{ik} > \frac{d_L^2}{5(1+R^2)} \tag{3.17}$$

which is used to upper bound the distance between two points from the regularity criteria. Indeed, if:

$$D_{ijk} = ||P_{ij} - P_{ik}|| \sin \alpha_{ij} \sin \alpha_{ik} < \eta,$$

then (3.17) gives:

$$||P_{ij} - P_{ik}|| < \frac{5\eta(1+R^2)}{d_L^2}$$

From the assumptions, we want $||P_{ij} - P_{ik}|| < d_P$ and therefore set:

$$\eta = \frac{d_P \, d_L^2}{5(1+R^2)}$$

□

When dealing with 64-bit arithmetic, a typical numerical value in Equation (3.15) would be $\eta = 10^{-16}$, for $R = 1$ and $d_P = d_L = 10^{-5}$. A discrete step in our optimization consists then in deciding for each regularization term in $E_{concurrent}$ below η the nature of the simplification: point fusion (b1), line fusion (a), or removal of an intersection point outside the area of interest (b2).

Propagation of reductions. The function **Reductions**$(\mathcal{C}, \mathcal{L})$ must propagate the simplifications in the partition, by verifying the following geometric invariant of our structure: two lines cannot have more than one common point and two points cannot have more than one common line.

In a first step, for each triplet (i, j, k) for which $|\det(L_i, L_j, L_k)| < \eta$ one of the mentioned 3 alternatives (lines merge, points merge, or point deletion) is applied. When lines L_i and L_j merge into line L_{ij}, if there is a point indexed by (L_j, L_k), it is inherited by the pair (L_{ij}, L_k). Along successive merges, the aforementioned geometric invariant may be violated, breaking the arrangement validity. Each time such a configuration is encountered, the indexes of the faulty pair is pushed to a stack. Once the first step is achieved, one applies either a line merge or a point merge operation for each pair popped from the stack. These merges may again push new pairs on the stack. Since each pop operation induces a reduction of the structure, the combinatorial simplification eventually ends.

Finally, when merging lines L_i and L_j into L_f, we combine metrics associated to both lines into a resulting fused metric:

$$M_f = M_i + M_j$$

This ensures that the fidelity to original lines of the arrangement is propagated throughout discrete operations. Note that this bears some resemblance with the discrete decimation algorithm using Quadric Error Metric (QEM) [GH97].

3.4 Experiments

3.4.1 Tuning the method

Tradeoff between fidelity, complexity and regularity. The two main parameters of our method are λ_1 and λ_2 that balance the three energy terms in the simplification of the partition. As illustrated in Figure 3.8, these two parameters allow an intuitive

control on the accuracy, complexity and the level of orthogonality of the partition. Note that, when increasing λ_2, the number of lines usually stays stable and the number of points decreases slightly: this decrease is explained by the situation where two lines become orthogonal to a third line and their intersection point is sent to infinity. The combinatorial reduction will then remove this intersection from the partition.

In Figure 3.8, as the error is concentrated in small cells, reasonable choices for concurrent lines and orthogonality parameters do not introduce large distortions and therefore the increase in 3D fidelity error is controlled. Further simplifications with larger regularity parameters degrade the partition gracefully: for instance, building contours can be identified in the most simplified example (top right). These large distortions lead however to large errors in the 3D fidelity measure.

Flexibility and robustness. We test our reconstruction method on different scales of urban scenes, from individual houses (Figure 3.11), building blocks (Figure 3.9), and larger architectural structures (Figure 3.4 and Figure 3.10). Our method produces concise and accurate models as long as observed buildings can be represented by a piecewise planar and disk-topology geometry. We also evalute our algorithm on input point clouds generated by different acquisition systems. As illustrated in Figure 3.11, our method returns 3D models of similar quality on both a Laser point cloud with missing data and occlusions and a multiview stereo point cloud with noise and outliers. The importance of the partition simplification step, our main contribution, is visible in this figure: without simplification, output results are less accurate and concise.

Performance. Our method is implemented in C++ (single-thread) and uses the Eigen library [GJ+10] for its ease of use when dealing with homogeneous representations as well as solving small linear algebra systems appearing in the computation of the Riemannian gradient. Section 3.4.1 shows the processing times of the different steps of our method on input data of various sizes. The creation and simplification of 2D partitions are fast and scalable, typically a few seconds for large urban scenes. By implementing the main geometric operations in the horizontal 2D plane, we avoid the time- and memory-consuming issues arisen from the creation and manipulation of 3D polyhedral arrangements. Adding this simplification method greatly improves the performance of the extrusion step by reducing the number of cells in the partition.

Metric propagation along combinatorial simplification. A prime example of the usefulness of propagating a metric information is given by Figure 3.7. In this example, the segment detection led to an original partition containing a lot of small segments clustered along the building's facades. During simplification, lines carrying these segments are merged in the first few iterations and their metric information is thus accumulated by lines representing large parts of the facades. When further simplifying, the fidelity term on these lines leads to very little deviation from their original position and are therefore stabilized by the information of previous simplifications. In the final partition, made of 44 lines – a 96% reduction from its original counterpart – the outline of the building's facades is still visible and lines on this outline can carry the information of up to 100 initial segments.

3.4.2 Comparison

We compare our method with the piecewise-planar reconstruction method [CLP10] and three mesh simplification pipelines in which input points are first converted into a dense surface mesh by the screened Poisson algorithm [KH13] before being simplified either by the popular Quadric Error Metrics edge contraction algorithm [GH97], by structure-aware edge decimation [SLA15] or by variational shape approximation [CAD04]. As illustrated in Figure 3.11, edge contraction based methods [GH97,SLA15] cannot produce very low complexity models (i.e. with one or two dozens of facets for a standard house)

	BarnMVS (Figure 3.11)	BuildingBlock (Figure 3.9)	Dublin (Figure 3.4)
#input points	619,472	1,000,000	6,305,813
Scene area	400 m^2	13,000 m^2	19,000 m^2
Line-segment extraction	6.7s	8.9s	37s
#line-segments	213	813	1,681
Kinetic partitioning	3ms	4ms	238ms
#polygons	330	1,422	2,724
Simplification	0.4s	1.5s	5.3s
#polygons	61	288	1,197
Extrusion	0.9s	3.1s	87s
#output faces	24	139	637

Table 3.1: Performances of our algorithm on different input data. The processing time for creating and simplifying 2D partitions are negligible compared to those of 3D operations, i.e. line-segment extraction and output mesh extrusion.

without strongly degrading geometric accuracy to input data. The mesh simplification pipeline [CAD04] and the reconstruction method [CLP10] can produce more accurate results but with a higher complexity. Our method offers the best compromise between accuracy and complexity as well as a stable results for different acquisition systems (laser and photogrammetry). Moreover, only our method can deliver meshes in which orthogonalities are preserved. Contrary to two previous algorithms [GH97, SLA15], our method does not offer a direct control on the output mesh complexity. This being said, our method has been designed to produce very concise meshes, departing from other algorithms [CAD04, CLP10].

3.4.3 Limitations

We acknowledge a few limitations of our method. First, the assumption that observed objects can be represented by a 2.5D disk-topology geometry, while true for many buildings, does not hold for other urban objects. The geometric regularities taken into account by our method are also limited to orthogonalities and, by construction, to collinearities. Considering other regularities such as symmetries would probably improve the quality of our results: it is conceivable to extend our optimization formulation to take them into account. Finally, the simplification of partitions can in some cases degrade the accuracy of the output model. As illustrated in Figure 3.12, this typically happens on curved structures, where small segments in 2D play a large role in the quality of the extruded facade, and the resulting models can appear overly simplified.

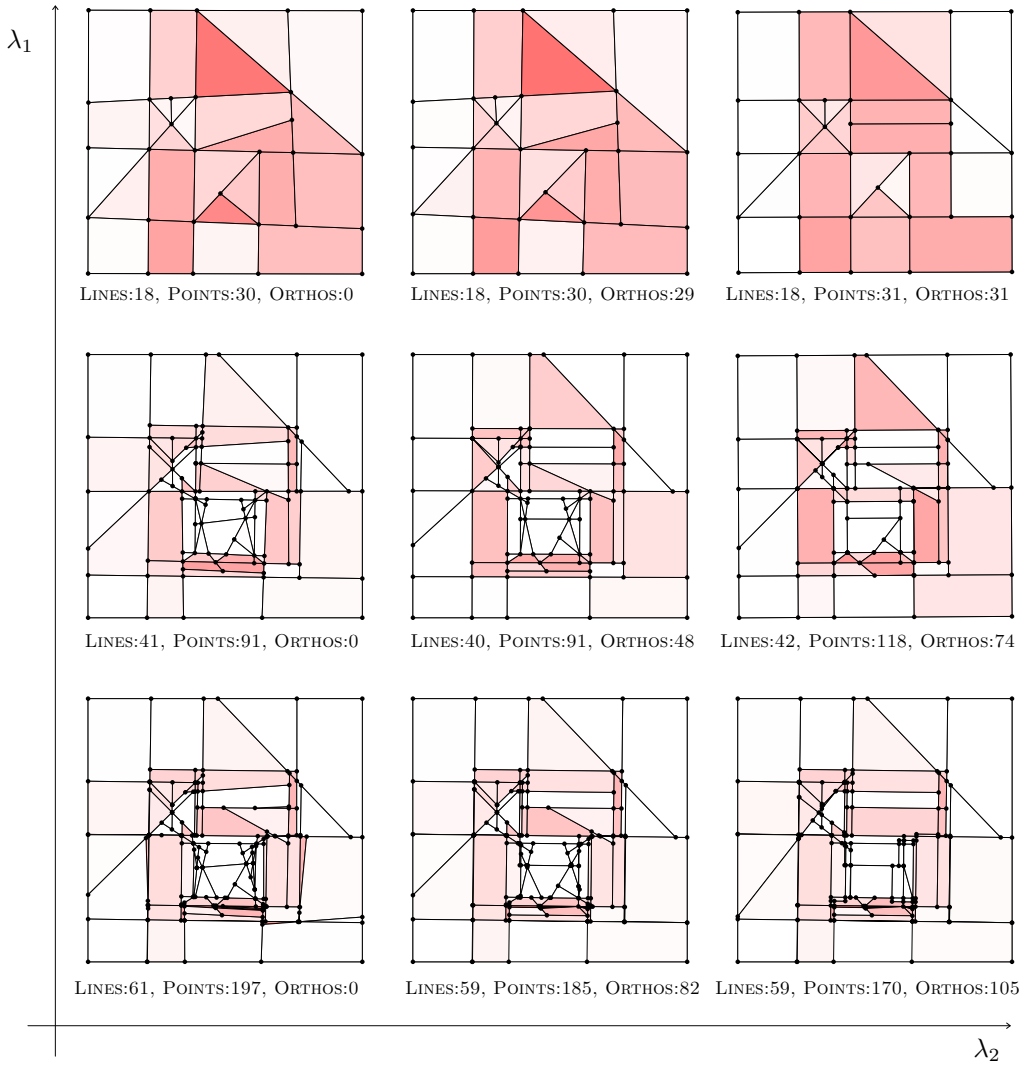


Figure 3.8: Impact of parameters λ_1 and λ_2. Increasing λ_1 reduces the complexity of the partition, while increasing λ_2 encourages the presence of orthogonal lines (Orthos refers to the number of orthogonalities counted when the angle difference with $\frac{\pi}{2}$ is under 10^{-4} degrees). Increasing λ_1 and λ_2 simultaneously reduce the accuracy of the partition as illustrated by the increasing presence of red polygons towards the top-right diagonal (the root mean squared error of the best-fit plane of the 3D input points projected in each polygon is displayed from white (zero error) to red (high error)).

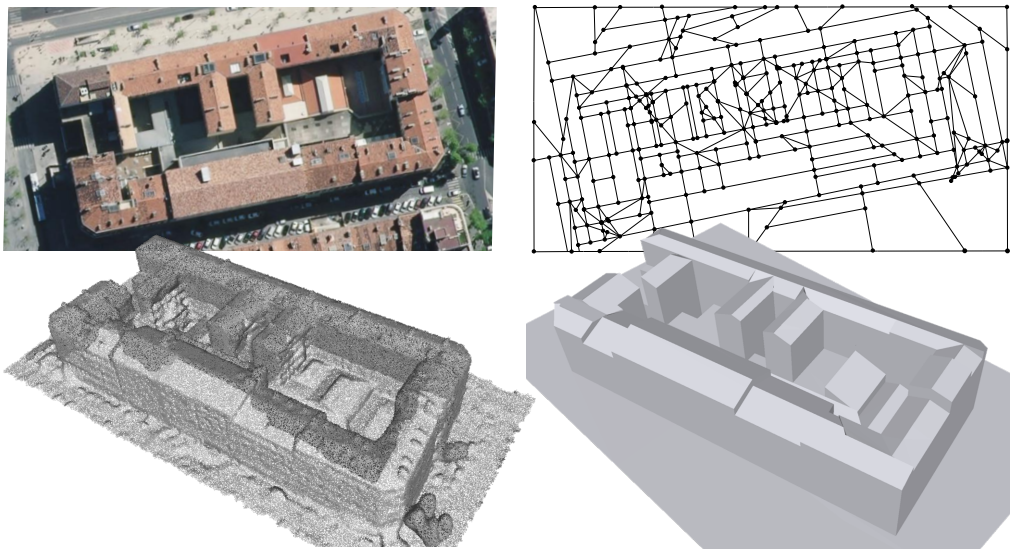


Figure 3.9: Result on a building block. The output 3D model (bottom right) obtained from a multiview stereo point cloud (bottom left) preserves the main roof components while ignoring small elements such as chimneys and dormer-windows.

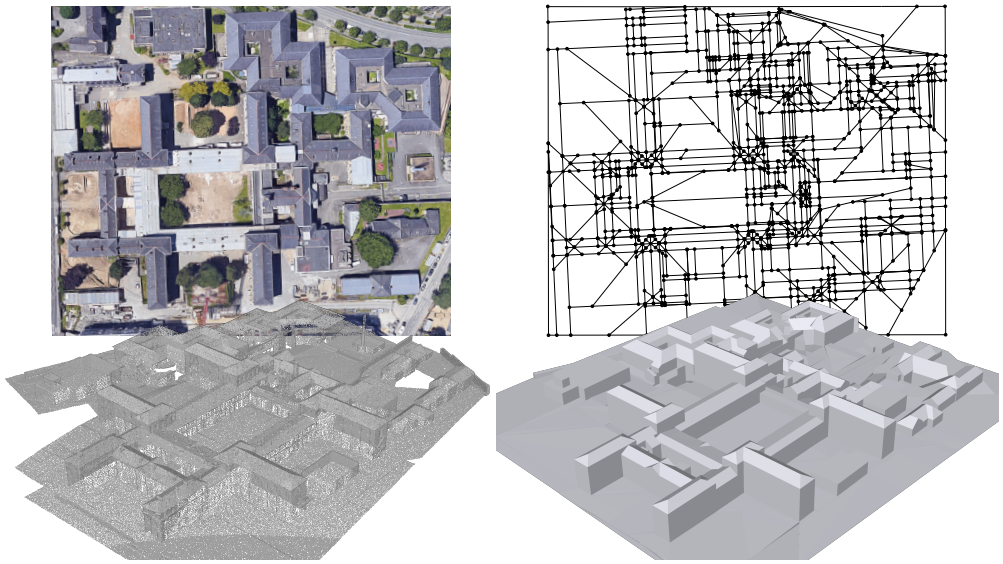


Figure 3.10: Result on large regular building complex. Our 3D output model (bottom right) preseves the orthogonality existing between facade and rooftop components. Note how the planimetric partition (top right) aligns well with the aerial image (top left, not used).

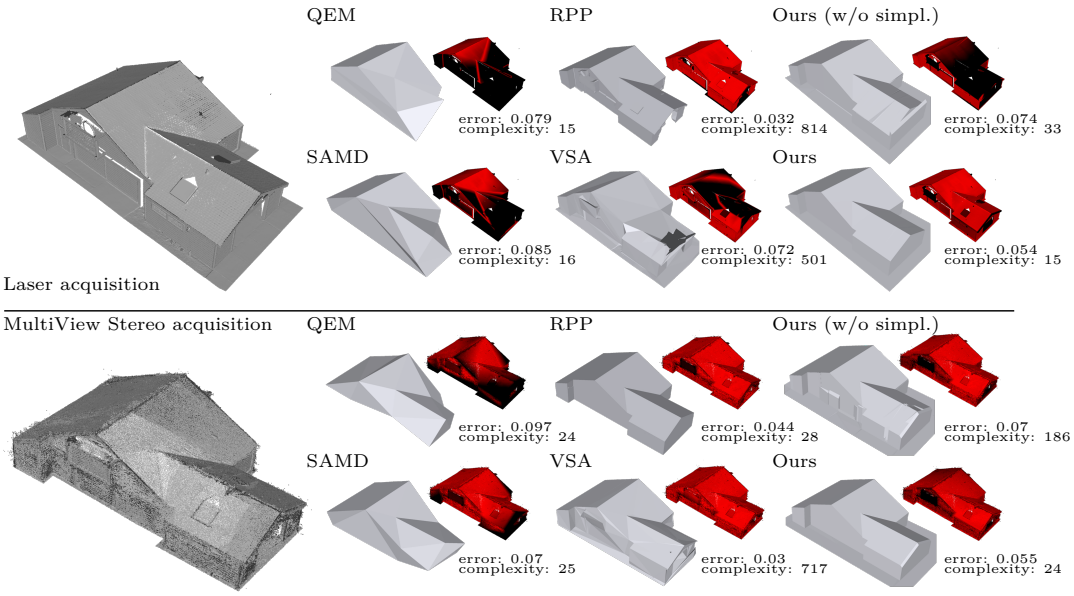


Figure 3.11: Comparisons with existing methods. Mesh simplification algorithms QEM [GH97] and SAMD [SLA15] applied from Screened Poisson dense meshes [KH13] cannot produce low complexity models without altering the geometric accuracy ("complexity" and "error" refer to the number of facets in the output models and to the root mean square Hausdorff distance from input point clouds (left) to the output models, the Hausdorff distance ranging from 0 meter (red) to 0.2 meter (black)). VSA [CAD04] and the piecewise-planar reconstruction method RPP [CLP10] deliver more accurate results but with a higher complexity. Our method yields the best trade-off between accuracy and complexity, independently of the acquisition system used to generate the input point cloud. In addition, it also preserves the orthogonality between the walls of the building. Note how our results are degraded when the simplification of the 2D partition is not activated. Input data from [KPZK17].

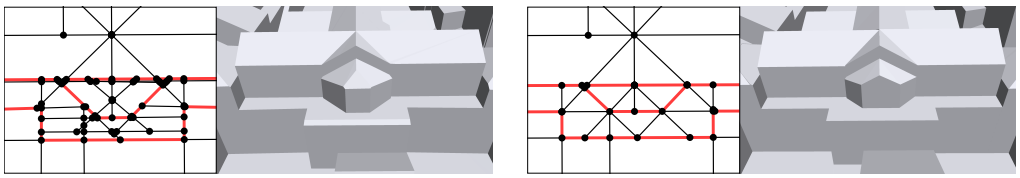


Figure 3.12: Failure case. Our simplification process may degrade the accuracy of the output model on curved structures. The left (respectively right) model is obtained without (resp. with) simplification of the 2D partition. Edges highlighted in red, corresponding to facades in the 3D model, show the discrepancy between 2D regularity and 3D fidelity.

Chapter 4

Dense representations from lexicographic optimal chains

Contents

List of Symbols

$\mathbf{P}$	Set of points
$\mathbb{F}$	Arbitrary coefficient field
$\mathbb{Z}$	Integer ring
$\mathbb{Z}_2$ or $\mathbb{Z}/2\mathbb{Z}$	Integer modulo 2 field
K	Simplicial complex
$K^{(k)}$	Set of k-simplices of K
B	Subcomplex of K
$K_{\mathbf{P}}$	Full complex of the set of points $\mathbf{P}$
$\mathcal{D}el(\mathbf{P})$	Delaunay complex of the set of points $\mathbf{P}$
$\mathcal{CH}(\mathbf{P})$	Convex hull of the set of points $\mathbf{P}$
d	Dimension of the complex K
k	Arbitrary dimension (verifying $k \leq d$)
$\mathbf{C}_k(K)$	Vector space of absolute k-chains on K
$\mathbf{Z}_k(K)$	Vector space of absolute k-cycles on K
$\mathbf{B}_k(K)$	Vector space of absolute k-boundaries on K
$\mathbf{C}_k(K, B)$	Vector space of relative k-chains on (K, B)
$\mathbf{Z}_k(K, B)$	Vector space of relative k-cycles on (K, B)
$\mathbf{B}_k(K, B)$	Vector space of relative k-boundaries on (K, B)
$\lvert\sigma\rvert$	Support of the simplex σ
$\lvert\Gamma\rvert$	Support of the chain Γ
$\mathbf{SEB}(\sigma)$	Smallest enclosing ball of σ
$\mathrm{R_B}(\sigma)$	Radius of the smallest enclosing ball of σ
$\mathrm{R_C}(\sigma)$	Radius of the smallest circumscribed sphere of σ
$\sigma_1 \leq \sigma_2$	Total order on simplices σ_1, σ_2
$\Gamma_1 \sqsubseteq_{lex} \Gamma_2$	Lexicographic preorder on chains Γ_1, Γ_2

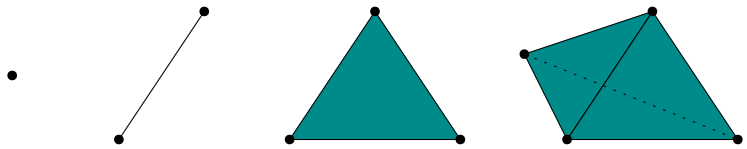


Figure 4.1: Simplices in 3D.

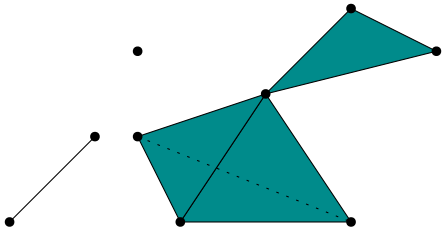


Figure 4.2: Example of a simplicial 3-complex.

4.1 Definitions and notations

This section presents succinctly the concepts of simplicial homology required for the rest of this chapter. Introductions to simplicial homology can be found in [Mun18,EH10]. We also give definitions and usual characterizations of Delaunay and regular triangulations. More details can be found in [BCY18].

4.1.1 Simplicial complexes and chains

An **abstract simplicial complex** K is a collection of finite non-empty sets that is closed under taking subsets, i.e. every subset of an element of K is also in K. A ***k*-simplex** of K is an element of K of size $k+1$. 0-, 1-, 2- and 3-simplices of K are referred to as *vertices*, *edges*, *triangles* and *tetrahedrons* of K and are illustrated in Figure 4.1.

Figure 4.2 gives an example of an abstract simplicial complex. The dimension of a finite abstract simplicial complex is the largest dimension of any simplex of the complex. We denote by $K^{(k)}$ the set of k-simplices in K. We will only consider finite complexes in this work.

A simplex τ subset of σ is called a **face** of σ, and is denoted as $\tau \preceq \sigma$. Reversely, we say σ a **coface** of τ.

A **subcomplex** B of an abstract simplicial complex K is an abstract simplicial complex whose simplices are included in K. In particular, the ***k*-skeleton** of K, containing all simplices in K of dimension at most k, is a subcomplex of K.

We call the **full *d*-complex** $K_{\mathbf{P}}$ over vertices $\mathbf{P}$ the collection of all possible simplices over $\mathbf{P}$ of dimension at most d.

For any k-simplex σ, its support $|\sigma|$ is defined as the convex hull of $k+1$ points in $\mathbb{R}^{k+1}$. We can associate to the simplicial complex K its geometric realization $|K|$, obtained by gluing supports of adjacent simplices on their common faces. For finite complexes, this topological space can be always be embedded, for instance in $\mathbb{R}^N$, where N denotes the number of vertices of the complex.

In all that follows, the set $\mathbf{P}$ will denote d-dimensional points with coordinates in $\mathbb{R}^d$. This defines therefore an map from vertices of K to $\mathbb{R}^d$, and by extension a unique piecewise-linear map, linear on each simplex, from $|K|$ to $\mathbb{R}^d$. When this application is injective, for instance for the Delaunay triangulation, we say it is an embedding of the abstract simplicial complex. When the application is non injective, which can happen

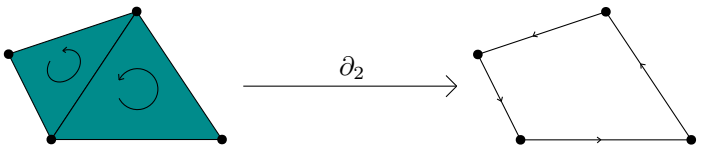


Figure 4.3: The boundary operator on a 2-chain. The coefficient of the 2-chain and 1-chain is understood to be 1 on each oriented simplex (indicated by arrows).

for the full d-complex over points $\mathbf{P}$, we say the simplicial complex is not embedded in $\mathbb{R}^d$.

Nevertheless, the support $|\sigma|$ of a k-simplex σ ($k \leq d$) will denote the convex hull of the $k+1$ points of $\mathbf{P}$ associated with this k-simplex.

Let K be a simplicial complex of dimension at least k. While the notions of chains and homology can be defined with coefficients in any ring, we consider here coefficients in a field $\mathbb{F}$, where we have in mind in particular the cases of the field of integers modulo 2 ($\mathbb{F} = \mathbb{Z}_2$) and the field of rationals ($\mathbb{F} = \mathbb{Q}$). A ***k*-chain** A with coefficients in $\mathbb{F}$ is a formal sum of k-simplices:

$$A = \sum_i x_i \sigma_i, \text{ with } x_i \in \mathbb{F} \text{ and } \sigma_i \in K^{(k)}$$

We denote by $\mathbf{C}_k(K)$, or $\mathbf{C}_k(K;\mathbb{F})$ when we want to emphasize the chain coefficient, the vector space of k-chains in the complex K.

For a chain $\Gamma \in \mathbf{C}_k(K)$, its **support**, denoted $|\Gamma|$, is the set of simplices for which the coefficient in Γ is not zero:

$$|\Gamma| \underset{\text{def.}}{=} \left\{\sigma \in K^{(k)}, \Gamma(\sigma) \neq 0\right\}$$

When $\mathbb{F} = \mathbb{Z}_2$, the chain coefficient in front of any simplex can be interpreted as indicating the existence of this simplex in the chain. We can view k-chains as sets of k-simplices: for a k-simplex σ and a k-chain A, we then write that $\sigma \in A$ if the coefficient for σ in A is 1. With this convention, the sum (or difference) of two chains corresponds to the symmetric difference on their sets. In what follows, a k-simplex σ can also be interpreted as the k-chain containing only the k-simplex σ.

A d-dimensional simplicial complex is said **pure** if it is of dimension d and any simplex has at least one coface of dimension d. A ***d*-pseudomanifold** is a pure d-dimensional simplicial complex for which each $(d-1)$-face has exactly two d-dimensional cofaces.

The **dual graph** of a d-pseudomanifold $\mathcal{M}$ is the graph whose vertices are in one-to-one correspondence with the d-simplices of $\mathcal{M}$ and whose edges are in one-to-one correspondence with $(d-1)$-simplices of $\mathcal{M}$: an edge e connects two vertices v_1 and v_2 of the graph if and only if e corresponds to the $(d-1)$-face with cofaces corresponding to v_1 and v_2.

A **strongly connected** d-pseudomanifold is a d-pseudomanifold whose dual graph is connected.

4.1.2 Boundary operator and homology

The **boundary operator** $\partial_k : \mathbf{C}_k(K) \to \mathbf{C}_{k-1}(K)$ is the linear map defined for any k-simplex $\sigma = [a_0, \ldots, a_k]$ as:

$$\partial_k \sigma \underset{\text{def.}}{=} \sum_{i=0}^{k} (-1)^i [a_0, \ldots, \widehat{a_i}, \ldots, a_k]$$

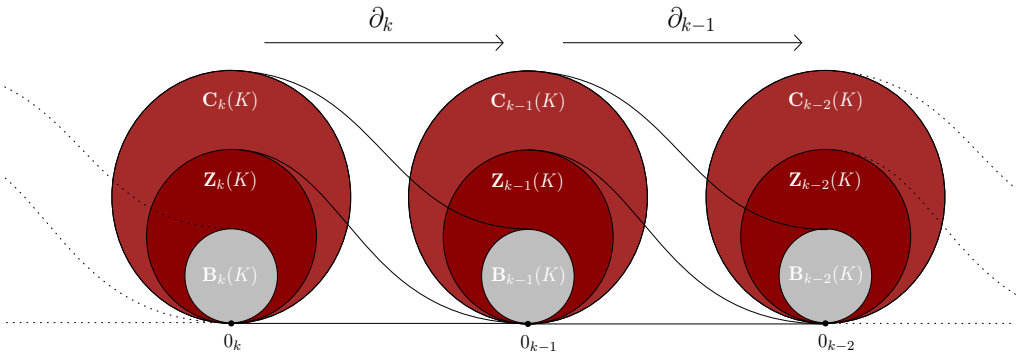


Figure 4.4: Illustration of the structure induced by the boundary operator. Chains are mapped to boundaries, cycles are mapped to the null chain of inferior dimension and boundaries form a subgroup of cycles.

where the symbol $\widehat{a_i}$ means the vertex a_i is deleted from the array. Figure 4.3 illustrates this natural operator.

The kernels and images of the boundary operator form respectively the vector space of **cycles** and **boundaries**:

$$\mathbf{Z}_k(K) \underset{\text{def.}}{=} \operatorname{Ker} \partial_k = \left\{ \Gamma \in \mathbf{C}_k(K), \partial_k \Gamma = 0 \right\}$$
$$\mathbf{B}_k(K) \underset{\text{def.}}{=} \operatorname{Im} \partial_{k+1} = \left\{ \Gamma \in \mathbf{C}_k(K), \exists A \in \mathbf{C}_{k+1}(K) \mid \Gamma = \partial_{k+1} A \right\}$$

The following fundamental property of the boundary operator

$$\partial_k \partial_{k+1} = 0$$

implies that

$$\operatorname{Im} \partial_{k+1} \subset \operatorname{Ker} \partial_k$$

or equivalently

$$\mathbf{B}_k(K) \subset \mathbf{Z}_k(K)$$

and induces a special structure across dimensions illustrated in Figure 4.4.

Homology groups – forming vector spaces over the field $\mathbb{F}$ – are defined as quotient spaces of cycles over boundaries:

$$\mathcal{H}_k(K) \underset{\text{def.}}{=} \frac{\mathbf{Z}_k(K)}{\mathbf{B}_k(K)}$$

Two k-cycles $A, A' \in \mathbf{Z}_k(K)$ are then said to be **homologous** if they belong to the same homology class in $\mathcal{H}_k(K)$, or equivalently if $A - A' = \partial_{k+1} B$ for some $(k+1)$-chain B. Figure 4.5 illustrates the notion of homologous cycles. By extension, we also say two k-chains $\Gamma, \Gamma' \in \mathbf{C}_k(K)$ are homologous if their difference is a boundary, i.e. there exists a $(k+1)$-chain B such that $\Gamma - \Gamma' = \partial B$.

4.1.3 Relative homology

Roughly speaking, relative homology of a simplicial pair (K, B) is constructed similarly to absolute homology but "ignores" all the part of K inside of B. Given a simplicial subcomplex B of K, relative k-chains are defined as classes in the following quotient space:

$$\mathbf{C}_k(K, B) \underset{\text{def.}}{=} \frac{\mathbf{C}_k(K)}{\mathbf{C}_k(B)}$$

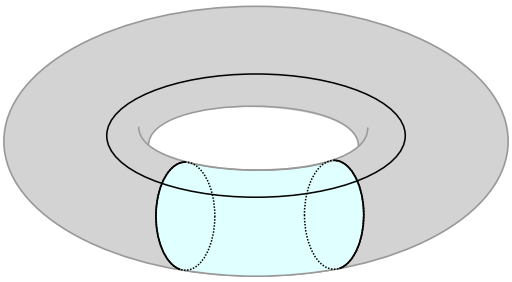


Figure 4.5: Illustration of homologous cycles on a torus. The two cycles on the minor radius of the torus are homologous: their difference is a boundary highlighted in blue. The cycle on the major radius of the torus is however in a different homology class than the two other cycles.

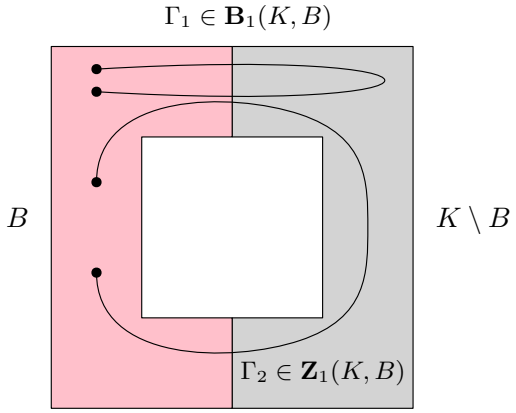


Figure 4.6: Illustration of relative 1-boundary and 1-cycle. Γ_1 and Γ_2 are respectively elements of a relative boundary and relative cycle class.

Denote by $\partial_k^{(r)}$ the relative k-boundary operator in (K, B) and by $\partial_k^{(a)}$ the absolute k-boundary operator in K. For any absolute k-chain $\Gamma \in \mathbf{C}_k(K)$,

$$\partial_k^{(r)}(\Gamma + \mathbf{C}_k(B)) \underset{\text{def.}}{=} \partial_k^{(a)}(\Gamma) + \mathbf{C}_{k-1}(B)$$

This is well-defined as $\partial_k^{(a)}(\mathbf{C}_k(B)) \subseteq \mathbf{C}_{k-1}(B)$. Relative cycles and boundaries, are still defined as respective kernel and image of the relative boundary operator:

$$\mathbf{Z}_k(K, B) \underset{\text{def.}}{=} \operatorname{Ker} \partial_k^{(r)}$$
$$\mathbf{B}_k(K, B) \underset{\text{def.}}{=} \operatorname{Im} \partial_{k+1}^{(r)}$$

Translated in term of absolute chains, for $\Gamma \in \mathbf{C}_k(K)$,

$$\Gamma + \mathbf{C}_k(B) \in \mathbf{Z}_k(K, B) \iff |\partial_k^{(a)}\Gamma| \subset B$$
$$\Gamma + \mathbf{C}_k(B) \in \mathbf{B}_k(K, B) \iff \exists A \in \mathbf{C}_{k+1}(K), |\Gamma - \partial_{k+1}^{(a)} A| \subset B$$

The notion of relative cycles and boundaries is illustrated on Figure 4.6.

Relative homology is again defined as the quotient space of relative cycles over relative boundaries:

$$\mathcal{H}_k(K, B) \underset{\text{def.}}{=} \frac{\mathbf{Z}_k(K, B)}{\mathbf{B}_k(K, B)}$$

While, strictly speaking, a relative k-chain $\Gamma \in \mathbf{C}_k(K, B)$ is a class, we allow ourselves to write, for simplicity,

$$\Gamma = \sum_{\sigma_i \in K \setminus B} x_i \sigma_i$$

where the simplex $\sigma_i \in K \setminus B$ stands in this case for $\sigma_i + \mathbf{C}_k(B)$. For all that follows, the distinction between absolute and relative boundary operators will be omitted and the notation ∂_k will denote either the absolute and relative boundary operator depending on the context.

4.1.4 Delaunay and regular complexes

For its ubiquity throughout this work, from theoretical justification of lexicographic orders to its use in practice as a 3-dimensional complex, we define and give a few insights on Delaunay complexes and its generalization for weighted points, called regular triangulations.

In all that follows, $\mathbf{SEB}(\sigma)$ denotes the smallest enclosing ball of a simplex σ. $\mathrm{R_B}(\sigma)$ and $\mathrm{R_C}(\sigma)$ respectively denote the radius of the smallest enclosing ball and the radius of the smallest circumscribed sphere associated to a simplex σ.

Delaunay triangulations Let $\mathbf{P} = (P_1, \ldots, P_N)$ denote a set of points in $\mathbb{R}^d$, with $N \geq d+1$. A condition of genericity is added to the set of points to ensure the existence of circumscribing spheres.

Condition 1. *We say that a set of point $\mathbf{P} = \{P_1, \ldots, P_N\} \subset \mathbb{R}^d$ satisfies the first generic condition if no set of $(d+1)$ points lies on a same $(d-1)$-dimensional affine space.*

The Delaunay triangulation knows a few different characterizations. We recall first its definition from the empty sphere property, illustrated in Figure 4.7.

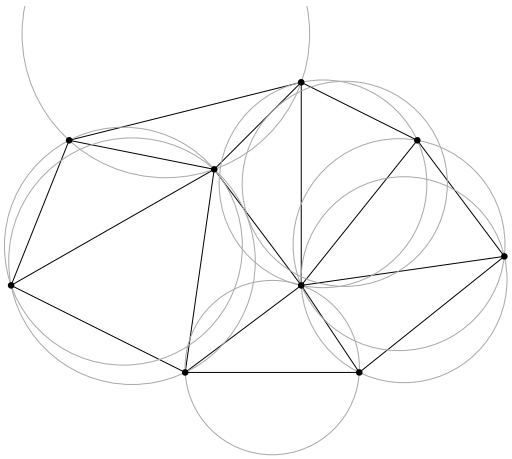


Figure 4.7: The empty sphere property of Delaunay triangulations.

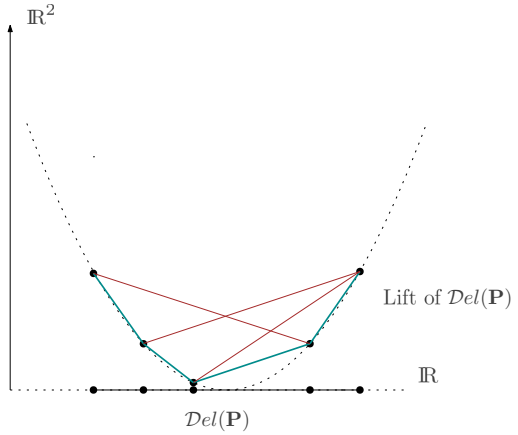


Figure 4.8: Illustration of a 1-dimensional Delaunay triangulation (in black) and its lift (in cyan). In contrast with other simplices shown in red, the lifted simplices of the Delaunay triangulation are on the lower hull of the lifted set of points.

Definition 4.1 (Empty-sphere property)**.** *A d-dimensional Delaunay triangulation $\mathcal{D}el(\mathbf{P})$ of $\mathbf{P} \subset \mathbb{R}^d$ is a triangulation of the convex hull of $\mathbf{P}$ such that no point in $\mathbf{P}$ lies inside the circumscribed sphere of any of its d-simplices.*

Particularly relevant to our work, we recall the lift characterization [Bro79] of Delaunay triangulations. Given a point $P \in \mathbb{R}^d$, its lift, denoted $\text{lift}(P)$, is a point in $\mathbb{R}^d \times \mathbb{R}$ given as:

$$\text{lift}(P) \underset{\text{def.}}{=} (P, \|P\|^2)$$

The boundary of the convex hull of the set of lifted points $\text{lift}(\mathbf{P})$ can be partitioned into two: the lower (resp. upper) hull correponds to simplices visible (resp. not visible) from the weighted point $(0, -\infty)$. The lift characterization of the Delaunay triangulation states that:

Proposition 1. *A simplex σ is in the Delaunay triangulation if and only if its lifted simplex $\text{lift}(\sigma)$ is on the lower hull of $\text{lift}(\mathbf{P})$.*

Figure 4.8 illustrates Proposition 1 in dimension 1 and shows how simplices that are

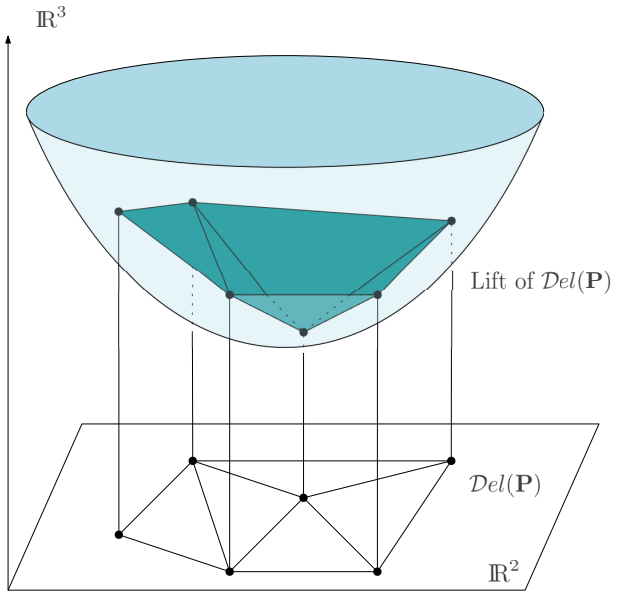


Figure 4.9: Illustration of a 2-dimensional Delaunay triangulation and its lift.

not in the Delaunay triangulation do not appear on the lower hull of the lifted points. The lifted Delaunay triangulation is also illustrated in dimension 2 in Figure 4.9.

Regular triangulations We briefly recall the notions associated to weighted points and distances required to introduce the generalization of Delaunay triangulations, called regular or weighted Delaunay triangulations.

We consider a set of weighted points $\mathbf{P} = \{(P_1, \mu_1), \ldots, (P_N, \mu_N)\} \subset \mathbb{R}^d \times \mathbb{R}$ with $N \geq n + 1$. For positive weights, a weighted point (P, μ) can be seen as a usual point associated with a sphere of radius $r = \sqrt{\mu}$.

Definition 4.2. *Given two weighted points $(P_1, \mu_1), (P_2, \mu_2) \in \mathbb{R}^d \times \mathbb{R}$ their weighted distance is defined as:*

$$\mathrm{D}\left((P_1, \mu_1), (P_2, \mu_2)\right) \underset{def.}{=} (P_1 - P_2)^2 - \mu_1 - \mu_2$$

we say that (P_1, μ_1) and (P_2, μ_2) are orthogonal *if $\mathrm{D}\left((P_1, \mu_1), (P_2, \mu_2)\right) = 0$.*

The orthogonality denomination is justified by Figure 4.10 when considering the spheres associated with positively-weighted points. From this weighted distance, we can define generalizations of circumscribed spheres and smallest enclosing balls for weighted points. Condition 1 is again required to properly define the generalization of circumscribed spheres.

Definition 4.3. *Let $\mathbf{P}$ be a set of weighted points verifying Condition 1. Given a k-simplex $\sigma \subset \mathbf{P}$ with $0 \leq k \leq d$ the generalized circumsphere and smallest enclosing ball of σ are the weighted points $(P_C, \mu_C)(\sigma)$ and $(P_B, \mu_B)(\sigma)$ respectively defined as:*

$$\mu_C\left(\sigma\right) \underset{def.}{=} \min\left\{\mu \in \mathbb{R}, \exists P \in \mathbb{R}^d, \forall (P_i, \mu_i) \in \sigma,\ \mathrm{D}\left((P, \mu), (P_i, \mu_i)\right) = 0\right\} \tag{4.1}$$

$$\mu_B\left(\sigma\right) \underset{def.}{=} \min\left\{\mu \in \mathbb{R}, \exists P \in \mathbb{R}^d, \forall (P_i, \mu_i) \in \sigma,\ \mathrm{D}\left((P, \mu), (P_i, \mu_i)\right) \leq 0\right\} \tag{4.2}$$

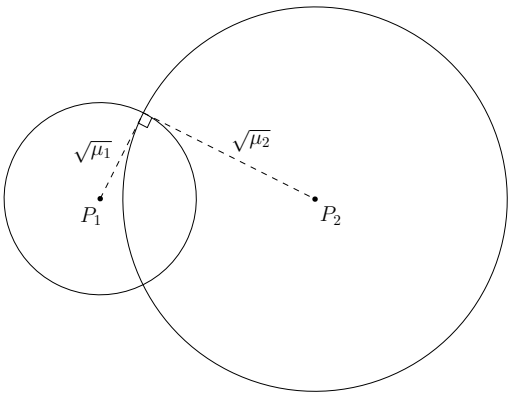


Figure 4.10: Two orthogonal points $(P_1, \mu_1), (P_2, \mu_2)$ with positive weights in dimension 2.

$P_C(\sigma)$ and $P_B(\sigma)$ are the unique points P that respectively realize the minimum in Equations (4.1) *and* (4.2)*. The weights $\mu_C(\sigma)$ and $\mu_B(\sigma)$ are called respectively "circumweight" and bounding weight of σ.*

Observation 4.4. *In the particular case of zero weights:*

- *The weighted distance is simply the squared Euclidean distance. For two points P_1 and P_2:*
$$D((P_1, 0), (P_2, 0)) = (P_1 - P_2)^2$$
- *$P_C(\sigma)$ and $P_B(\sigma)$ are respectively the center of the smallest circumscribed sphere and the center of the smallest enclosing ball of the simplex σ.*
- *$\mu_C(\sigma)$ and $\mu_B(\sigma)$ correspond to the squared radius of the smallest circumscribed sphere and the squared radius of the smallest enclosing ball of the simplex σ:*
$$\mu_C(\sigma) = \mathrm{R_C}(\sigma)^2 \quad \textit{and} \quad \mu_B(\sigma) = \mathrm{R_B}(\sigma)^2$$

Similarly to Delaunay triangulations, regular triangulations can be defined with a generalization of the empty sphere property.

Definition 4.5. *A regular triangulation $\mathcal{T}$ of the set of weighted points $\mathbf{P} = \{(P_1, \mu_1), \ldots, (P_N, \mu_N)\} \subset \mathbb{R}^d \times \mathbb{R}$, $N \geq d+1$, is a triangulation of the convex hull of $\{P_1, \ldots, P_N\}$ taking its vertices in $\{P_1, \ldots, P_N\}$ such that for any d-simplex $\sigma \in \mathcal{T}$*

$$(P_i, \mu_i) \in \mathbf{P} \setminus \sigma \Rightarrow \mathrm{D}\left((P_C(\sigma), \mu_C(\sigma)), (P_i, \mu_i)\right) \geq 0$$

The lift characterization also applies for Delaunay triangulations, with the following definition of lift for weighted points:

$$\begin{aligned} \text{lift} : \mathbb{R}^d \times \mathbb{R} &\to \mathbb{R}^d \times \mathbb{R} \\ (P, \mu) &\mapsto (P, \|P\|^2 - \mu) \end{aligned}$$

4.2 Lexicographic order and total orders on simplices

In this section, a generic preorder on k-chains induced by a total order on k-simplices is defined and qualified as *lexicographic*. At first sight, minimizing along this lexicographic

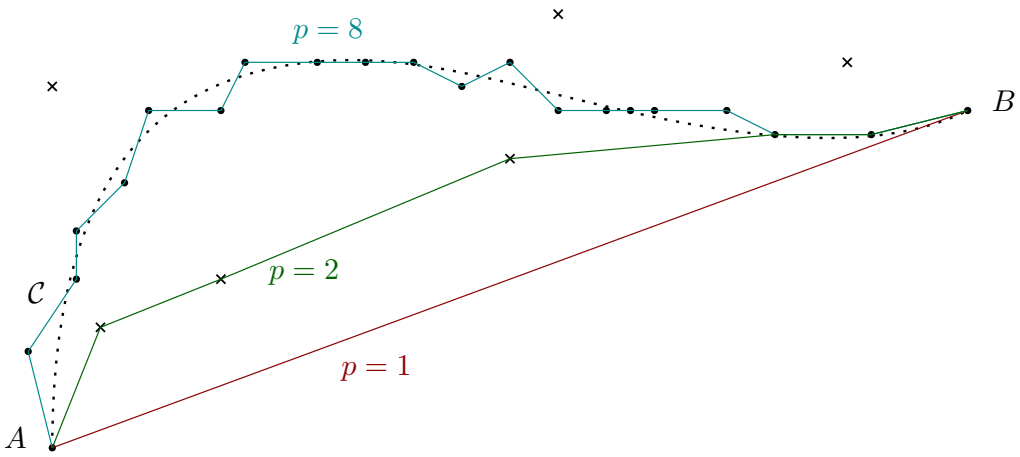


Figure 4.11: Illustration of weighted shortest paths for different exponents, as defined in Equation (4.4). The set of points is composed of inliers (represented as dots) sampling the curve $\mathcal{C}$ and outliers (represented as crosses). The graph is not drawn but is supposed to be the complete graph. Increasing the exponent on the edge lengths favors smaller edges, and the shortest path goes through dense regions of the set of points.

preorder might seem peculiar and therefore an informal example of curve reconstruction in 2D is given. When considering generalizing this approach to shape reconstruction in higher dimensions, the main question that arises is the definition of the total order on k-simplices required by this lexicographic order. A total order on 2-simplices, which will be useful to reconstruction applications of 2-surfaces, will be justified by its connection to 2-dimensional Delaunay triangulations. Finally, we will mention that a total order with similar connections to regular triangulations, can be defined in any dimension. As the justification of these total orders extends beyond the scope of this work, some results are given without proof but with references to the corresponding result in [CLV19a].

4.2.1 An intuition of lexicographic optimality

We first give an informal example of lexicographic optimality to build an understanding of its usefulness for shape reconstruction.

Depicted in Figure 4.11, consider a set of points $\mathbf{P}$ sampling an open curve $\mathcal{C}$ in 2D, with the usual defects found in acquisitions:

- noisy and non-uniformly sampled inliers located close to $\mathcal{C}$;
- outliers far from $\mathcal{C}$, in relatively few number compared to inliers.

We make the additional assumption that the boundary of the curve $\mathcal{C}$, denoted by points A and B, is part of $\mathbf{P}$. For simplicity, we also construct the complete graph $G = (\mathbf{P}, \mathcal{E})$, having an edge between any two points of $\mathbf{P}$. We assume all edges have distinct lengths and therefore for $e_1, e_2 \in \mathcal{E}$:

$$||e_1|| = ||e_2|| \implies e_1 = e_2 \tag{4.3}$$

where $\|e\|$ denotes to the length of the edge e.

Given this boundary information for $\mathcal{C}$, one might consider looking at shortest paths from A to B for the reconstruction of the shape $\mathcal{C}$. However, in the complete graph, the shortest path between any two points for the usual Euclidean distance will always be the edge between A and B. Instead, we are interested in a family of weights associating

to each edge e its p-powered length for $p \in [1, \infty)$. Denoting by $\{e_i\}_{i \in I}$ a set of edges in G forming a path from A to B, its associated cost c_p is then:

$$c_p(\{e_i\}_{i \in I}) \underset{\text{def.}}{=} \sum_{i \in I} \|e_i\|^p \tag{4.4}$$

In Figure 4.11, three paths minimal for this cost c_p are drawn, for the values $p = 1$ (Euclidean distance), $p = 2$, and $p = 8$. Whereas the first two shortest paths are far from the expected shape, go through outliers and very few inlier points, the latter avoids outliers and goes through the densest parts of the point cloud, as a higher exponent on the edge lengths favors smaller edges.

This leads to considering the limit behaviour of weights when $p \to \infty$. In this case, the cost c_p of a path defined by Equation (4.4) is dominated by the weight $\|e_{\max}\|^p$ of its longest edge $e_{\max}$. This means comparing two paths asymptotically involves comparing the length of their longest edge, and in case of equality (which implies both paths go through the same largest edge from Equation (4.3)), the length of their second longest edge and so on... The comparison of two paths when $p \to \infty$ uses therefore a lexicographic comparison on the edges of both paths, sorted in decreasing order of their lengths.

The lexicographic order will be formally defined in the next section for simplicial chains. Indeed, although paths in graphs can be somewhat more intuitive, the vector space structure of chains is the right formalism to generalize lexicographic minimization problems. Instead of minimizing in the set of paths connecting A to B, this will be formulated on the set of chains Γ verifying $\partial\Gamma = B - A$. This is a weaker condition, as any path Γ from A to B verifies $\partial\Gamma = B - A$ but not the converse. One can however observe that a lexicographic optimal chain $\Gamma_{\min}$ verifying $\Gamma_{\min} = B - A$ will in fact be a path between A and B.

4.2.2 Lexicographic order on chains

Whenever considering lexicographic orders on k-chains, we assume a total order on k-simplices, which will simply be denoted by $\leq$.

Definition 4.6 (Lexicographic total preorder)**.** *Given* $\Gamma_1, \Gamma_2 \in \mathbf{C}_k(K; \mathbb{F})$*,*

$$\Gamma_1 \sqsubseteq_{lex} \Gamma_2 \underset{def.}{\iff} \begin{cases} |\Gamma_1| = |\Gamma_2| \\ \text{or} \\ \max\left\{\sigma \in |\Gamma_1| \triangle |\Gamma_2|\right\} \in |\Gamma_2| \end{cases}$$

where $\triangle$ *denotes the set symmetric difference.*

A corresponding strict order $\sqsubset_{lex}$ is defined as:

$$\Gamma_1 \sqsubset_{lex} \Gamma_2 \underset{\text{def.}}{\iff} \Big(\Gamma_1 \sqsubseteq_{lex} \Gamma_2\Big) \quad \text{and} \quad \neg\Big(\Gamma_2 \sqsubseteq_{lex} \Gamma_1\Big)$$

From the definition of $\sqsubseteq_{lex}$, we make the following observation:

Observation 4.7. *Consider* $\Gamma_1, \Gamma_2 \in \mathbf{C}_k(K; \mathbb{F})$*.*

$$\big(\Gamma_1 \sqsubseteq_{lex} \Gamma_2 \quad \textit{and} \quad \Gamma_2 \sqsubseteq_{lex} \Gamma_1\big) \iff |\Gamma_1| = |\Gamma_2|$$

Observation 4.8. *When* $\mathbb{F} = \mathbb{Z}_2$*, the preorder* $\sqsubseteq_{lex}$ *is a total order and, by considering chains on* $\mathbb{Z}_2$ *as sets, the order can be written as follows.*

Given $\Gamma_1, \Gamma_2 \in \mathbf{C}_k(K; \mathbb{Z}_2)$,

$$\Gamma_1 \sqsubseteq_{lex} \Gamma_2 \iff \begin{cases} \Gamma_1 = \Gamma_2 \\ \text{or} \\ \max\left\{\sigma \in \Gamma_1 + \Gamma_2\right\} \in \Gamma_2 \end{cases}$$

Observation 4.9. *Qualifying this order as lexicographic is natural from the fact the comparison depends on the first place where the two chain supports differ. However, our lexicographic order is different from the order used, for instance, in dictionaries. By analogy, our order assumes all letters (i.e. simplices) of each word (i.e. chain support) are first ordered from largest to smallest before comparison.*

We first characterize the existence and unicity of lexicographic optimal chains. These properties of lexicographic minima can be best understood in the abstract context of quotient vector spaces. Consider a pair $A \subset E$ of finite dimensional vector spaces, with a given ordered basis:

$$e_1 < \ldots < e_n$$

of E inducing a lexicographic preorder $\sqsubseteq_{lex}$ on E.

Definition 4.10 (Application M_{lex}: ***lex***icographic ***M***inimal representative)**.** *The map* $M^A_{lex} : E \to E$ *is defined as:*

$$M^A_{lex}(x) \underset{def.}{=} \min_{\sqsubseteq_{lex}} x + A \tag{4.5}$$

The exponent A *is omitted when there is no ambiguity.*

The map M_{lex} is well defined since one has:

Lemma 4.11. *Given* $x \in E$*, the minimum* $M^A_{lex}(x)$ *under the total preorder* $\sqsubseteq_{lex}$ *in Equation* (4.5) *exists and is unique.*

Proof. We call the support $|x| \subset \{e_1, \ldots, e_n\}$ of $x \in E$ the set of basis elements for which the corresponding coordinates in x are not zero. Since $x \in x + A$, the set $x + A$ is not empty. Since there is a finite number of supports on the space $x + A$, there exists a minimal support, and therefore at least one minimal vector. Assume, for a contradiction, that there exists $x_1, x_2 \in x + A$, $x_1 \neq x_2$, minimum under the preorder $\sqsubseteq_{lex}$. Consider a basis element $\sigma \in |x_2 - x_1|$. From Observation 4.7, we have that $|x_1| = |x_2|$, therefore $|x_2 - x_1| \subset |x_1|$ and σ is in $|x_1|$. We now consider the following vector:

$$x_3 \underset{\text{def.}}{=} x_1 + \frac{x_1(\sigma)}{(x_2 - x_1)(\sigma)}(x_2 - x_1)$$

As $x_1 \in x + A$ and $x_2 - x_1 \in A$, we have that x_3 belongs to $x + A$. By construction, it is also smaller than x_1 for the preorder $\sqsubseteq_{lex}$, as $|x_3| \subset |x_1| \setminus \{\sigma\}$. This is a clear contradiction with the definition of x_1 as a minimum of $x + A$. □

A major insight for solving lexicographic optimality problems lies in the linearity of the application M_{lex}:

Lemma 4.12. *The application* M^A_{lex} *is linear. In other words, for* $x_1, x_2 \in E$ *and* $\lambda \in \mathbb{F}$*,*

$$M^A_{lex}(x_1 + \lambda x_2) = M^A_{lex}(x_1) + \lambda M^A_{lex}(x_2)$$

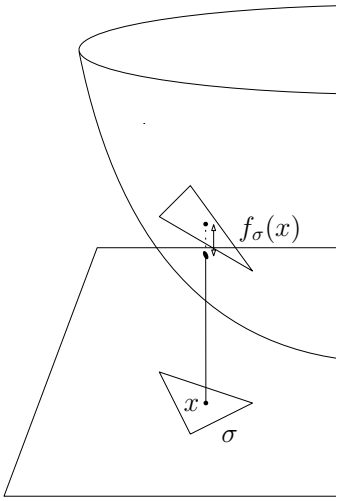


Figure 4.12: The height difference function f_σ associated with a 2-simplex σ.

Proof. Recall that $\forall x \in E, M^A_{lex}(x) - x \in A$, so that one can define $a \in A$ as:

$$a \underset{\text{def.}}{=} M_{lex}(x_1 + \lambda x_2) - M_{lex}(x_1) - \lambda M_{lex}(x_2) \tag{4.6}$$

Assume for a contradiction that $a \neq 0$ and take $\sigma_{\max} = \max |a|$. The basis element $\sigma_{\max}$ belongs to the support of (at least) one of the three terms of Equation (4.6). We assume first that $\sigma_{\max} \in |M_{lex}(x_1)|$. We now have the following chain:

$$x \underset{\text{def.}}{=} M_{lex}(x_1) - \frac{M_{lex}(x_1)(\sigma_{\max})}{a(\sigma_{\max})} a$$

that verifies by design $\sigma_{\max} \notin |x|$, so that $x \sqsubset_{lex} M_{lex}(x_1)$. Since $x \in x_1 + A$, this is a clear contradiction with the definition of $M_{lex}(x_1)$: we conclude that $\sigma_{\max} \notin |M_{lex}(x_1)|$. A similar construction assuming $\sigma_{\max} \in |M_{lex}(x_1 + \lambda x_2)|$ or $\sigma_{\max} \in |M_{lex}(x_2)|$ leads to the same contradiction. We conclude that $a = 0$ and M_{lex} is linear. □

Consequences of these properties of uniqueness and linearity will be further explored in Section 4.7.2.

4.2.3 A total order on 2-simplices

The main result of this section is deriving a total order on 2-simplices (Lemma 4.14) and giving an insight on the proof showing that lexicographic optimal chains induced by this total order characterize 2-dimensional Delaunay triangulations.

Variational approach to the Delaunay triangulations

The lift characterization (Proposition 1) inspired authors [Mus03, CX04, ACYD05] to consider a variational approach to Delaunay triangulations. We consider a set of points $\mathbf{P} \subset \mathbb{R}^d$ and denote by $K_{\mathbf{P}}$ the d-dimensional full complex over $\mathbf{P}$, made of all simplices over vertices of $\mathbf{P}$ of dimension at most d.

For each k-simplex $\sigma = \{P_0, \ldots, P_k\}$, define a function:

$$\begin{aligned} f_\sigma : |\sigma| &\to \mathbb{R} \\ x &\mapsto f_\sigma(x) \underset{\text{def.}}{=} \sum_i \lambda_i(x) \|P_i\|^2 - \|x\|^2 \end{aligned} \tag{4.7}$$

where $\lambda_i(x)$ are the barycentric coordinates of x verifying

$$x = \sum_i \lambda_i(x) P_i \text{ with } \sum_i \lambda_i(x) = 1 \text{ and } \lambda_i(x) \geq 0 \text{ for all } i = 0, \ldots, k$$

As illustrated on Figure 4.12 for 2-dimensional simplices, this function measures, for any point $x \in |\sigma|$, the height difference along the line $((x,0),(x,+\infty))$ between the lifted simplex $\text{lift}(\sigma)$ and the paraboloid defined by the function $P \mapsto \|P\|^2$.

Defining a weight w_p for each k-simplex σ

$$w_p(\sigma) \underset{\text{def.}}{=} \|f_\sigma\|_p = \left(\int_{|\sigma|} f_\sigma(x)^p dx \right)^{\frac{1}{p}} \tag{4.8}$$

the variational characterization of the Delaunay triangulation in [CX04] states that, for any $p \in [1, \infty)$:

$$\mathcal{D}el(\mathbf{P}) = \underset{\mathcal{T} \in \mathcal{V}_\mathbf{P}}{\arg\min} \left(\sum_{\sigma \in \mathcal{T}} w_p(\sigma)^p \right)^{\frac{1}{p}}$$

where $\mathcal{V}_\mathbf{P}$ corresponds to the set of all triangulations of the convex hull $\mathcal{CH}(\mathbf{P})$ of $\mathbf{P}$. This characterization can in fact be extended to simplicial chains. Define the following L^1 norms on d-chains:

$$\begin{aligned} \|\Gamma\|_{(p)} : \mathbf{C}_d(K_\mathbf{P}) &\to \mathbb{R} \\ \Gamma &\mapsto \sum_{\sigma \in K^{(d)}} |\Gamma(\sigma)| w_p(\sigma)^p \end{aligned} \tag{4.9}$$

The following proposition extends the variational formulation to d-chains.

Proposition 2 (Proposition 6.13 in [CLV19a])**.** *Denote by $\beta_\mathbf{P} \in \mathbf{B}_{d-1}(K_\mathbf{P})$ the $(d-1)$-boundary made of the simplices belonging to the boundary of $\mathcal{CH}(\mathbf{P})$. For any $p \in [1, \infty)$, define the chain*

$$\Gamma_{\min} \underset{def.}{=} \underset{\substack{\Gamma \in \mathbf{C}_d(K_\mathbf{P}) \\ \partial\Gamma = \beta_\mathbf{P}}}{\arg\min} \|\Gamma\|_{(p)}$$

The support $|\Gamma_{\min}|$ of $\Gamma_{\min}$ corresponds to all d-simplices of the Delaunay triangulation of $\mathbf{P}$.

Asymptotical behaviour of w_p

We are now interested in the asymptotical behaviour of weights $w_p(\sigma)$ when $p \to \infty$ in dimension $d = 2$. Assume the set of points $\mathbf{P} \subset \mathbb{R}^2$ verifies the following generic condition, which enforces that no two edges have the same length and four points cannot be co-cyclic.

Condition 2. *The set of points $\mathbf{P}$ verifies Condition 1 (no degenerate triangles) and for any two edges or triangles σ_1 and $\sigma_2 \in K_\mathbf{P}$, one has:*

$$\mathrm{R_C}(\sigma_1) = \mathrm{R_C}(\sigma_2) \implies \sigma_1 = \sigma_2$$

We give the following asymptotical behaviour of the weight $w_p(\sigma)$ according to the nature of the triangle σ. The proof of this result is given in Appendix A.1. Recall the asymptotic equivalence notation:

$$f(x) \underset{x \to \infty}{\sim} g(x) \underset{\text{def.}}{\iff} \lim_{x \to \infty} \frac{f(x)}{g(x)} = 1$$

Lemma 4.13 (Proof in Appendix A.1)**.** *Given a set of points* $\mathbf{P}$ *verifying Condition 2, if a triangle* $\sigma \in K_{\mathbf{P}}$ *is:*

– *strictly acute,*

$$w_p(\sigma)^p \underset{p\to\infty}{\sim} \pi \frac{1}{p} \mathrm{R_B}(\sigma)^{2+2p}$$

– *right,*

$$w_p(\sigma)^p \underset{p\to\infty}{\sim} \frac{\pi}{2} \frac{1}{p} \mathrm{R_B}(\sigma)^{2+2p}$$

– *strictly obtuse,*

$$w_p(\sigma)^p \underset{p\to\infty}{\sim} \sqrt{\frac{\pi}{4h^2} \frac{1}{p^3}} \mathrm{R_B}(\sigma)^{2+2p}$$

with $h^2 = \frac{\mathrm{R_C}(\sigma)^2 - \mathrm{R_B}(\sigma)^2}{\mathrm{R_B}(\sigma)^2}$.

Define the following comparison between 2-simplices $\sigma_1, \sigma_2 \in K^{(2)}$

$$\sigma_1 \leq \sigma_2 \underset{\text{def.}}{\iff} \exists p \in \mathbb{N} \mid \forall p' > p, \quad w_{p'}(\sigma_1) \leq w_{p'}(\sigma_2)$$

as well as a strict order $<$ based on the strict comparison of the weights.

is shown to be equivalent to the following comparison thanks to the asymptotical behaviours of Lemma 4.13.

Lemma 4.14. *For* $\sigma_1, \sigma_2 \in K^{(2)}$,

$$\sigma_1 \leq \sigma_2 \iff \begin{cases} \mathrm{R_B}(\sigma_1) < \mathrm{R_B}(\sigma_2) \\ \text{or} \\ \mathrm{R_B}(\sigma_1) = \mathrm{R_B}(\sigma_2) \quad \text{and } \mathrm{R_C}(\sigma_1) \geq \mathrm{R_C}(\sigma_2) \end{cases}$$

Proof. Consider two distinct 2-simplices $\sigma_1 \neq \sigma_2$. We make the following two distinctions:

Case 1: $\mathrm{R_B}(\sigma_1) \neq \mathrm{R_B}(\sigma_2)$
The asymptotical behaviours of Lemma 4.13 give immediately that, independently of the nature of the triangle:

$$\sigma_1 < \sigma_2 \iff \mathrm{R_B}(\sigma_1) < \mathrm{R_B}(\sigma_2)$$

Case 2: $\mathrm{R_B}(\sigma_1) = \mathrm{R_B}(\sigma_2)$
The contrapositive of Condition 2 implies that $\mathrm{R_C}(\sigma_1) \neq \mathrm{R_C}(\sigma_2)$. As acute and right triangles have same radius of smallest enclosing ball and circumscribed sphere, one of the two triangles has to be obtuse. The comparison of the asymptotical behaviours between an obtuse triangle σ_1 and an acute or right triangle σ_2 leads to $\sigma_1 < \sigma_2$, which coincides with the comparison $\mathrm{R_C}(\sigma_1) > \mathrm{R_C}(\sigma_2)$, as $\mathrm{R_C}(\sigma_1) > \mathrm{R_B}(\sigma_1)$ and $\mathrm{R_C}(\sigma_2) = \mathrm{R_B}(\sigma_2) = \mathrm{R_B}(\sigma_1)$. For two stricly obtuse triangles, the comparison of the asymptotical behaviour also give that $\sigma_1 < \sigma_2$ is equivalent to $\mathrm{R_C}(\sigma_1) > \mathrm{R_C}(\sigma_2)$. □

It is clear that, under Condition 2, the order $\leq$ is antisymmetric and therefore defines a total order on 2-simplices.

Delaunay triangulations as optimal lexicographic chains

To characterize the 2-dimensional Delaunay triangulation using the lexicographic order $\sqsubseteq_{lex}$ induced by this total order, we now need to compare it with the preorder induced by the asymptotical behaviour of norms defined in Equation (4.9):

Definition 4.15 ($\sqsubseteq_\infty$ order on 2-chains)**.** *For $\Gamma_1, \Gamma_2 \in \mathbf{C}_2(K)$:*

$$\Gamma_1 \sqsubseteq_\infty \Gamma_2 \underset{def.}{\iff} \exists p \in [1,\infty), \forall p' \in [p,\infty), \|\Gamma_1\|_{(p')} \leq \|\Gamma_2\|_{(p')}$$

We can again define a strict order $\sqsubset_\infty$ as:

$$\Gamma_1 \sqsubset_\infty \Gamma_2 \underset{\text{def.}}{\iff} \Big(\Gamma_1 \sqsubseteq_\infty \Gamma_2\Big) \quad \text{and} \quad \neg\Big(\Gamma_2 \sqsubseteq_\infty \Gamma_1\Big)$$

If the orders $\sqsubseteq_{lex}$ and $\sqsubseteq_\infty$ were equivalent, Proposition 2 would then give immediately a characterization of the Delaunay triangulation as a lexicographic optimal chain. However, a counter-example can be given, showing orders $\sqsubseteq_{lex}$ and $\sqsubseteq_\infty$ are not equivalent.

Observation 4.16 ($\sqsubseteq_{lex}$ and $\sqsubseteq_\infty$ differ)**.** *Consider three obtuse triangles $\sigma_1, \sigma_2, \sigma_3$ sharing a same longest edge – this can occur generically – and therefore sharing the same smallest enclosing ball. Denote, for $i = 1, 2, 3$, the quantities from Lemma 4.13 of each simplex:*

$$h_i^2 = \frac{\mathrm{R_C}(\sigma_i)^2 - \mathrm{R_B}(\sigma_i)^2}{\mathrm{R_B}(\sigma_i)^2}$$

It is possible for these values to verify simultaneously:

$$\begin{cases} h_1 < h_2 \\ h_1 < h_3 \\ \frac{1}{h_1} < \frac{1}{h_2} + \frac{1}{h_3} \end{cases}$$

The first two inequalities, with the fact they share the same smallest enclosing ball, translate to $\sigma_2 < \sigma_1$ and $\sigma_3 < \sigma_1$ resulting in:

$$\sigma_2 + \sigma_3 \sqsubset_{lex} \sigma_1$$

However, from the last inequality, the asymptotical behaviour for obtuse triangles of Lemma 4.13 gives us:

$$\sigma_1 \sqsubset_\infty \sigma_2 + \sigma_3$$

Even though the orders $\sqsubseteq_{lex}$ and $\sqsubseteq_\infty$ are not equivalent, we can show that this counter-example cannot appear in the minima of either $\sqsubseteq_{lex}$ and $\sqsubseteq_\infty$.

Lemma 4.17. *Let $\mathbf{P} \subset \mathbb{R}^d$ be a finite set of points satisfying Condition 2 and let $K_\mathbf{P}$ be the full complex over $\mathbf{P}$. For some 1-boundary $\beta \in \mathbf{B}_1(K_\mathbf{P})$, define the set $\mathcal{D}$ of 2-chains bounded by β:*

$$\mathcal{D} \underset{def.}{=} \Big\{\Gamma \in \mathbf{C}_2(K_\mathbf{P}) \mid \partial\Gamma = \beta\Big\}$$

If $\Gamma \in \mathbf{C}_2(K_\mathbf{P})$ is a minimum in $\mathcal{D}$ for one of the orders $\sqsubseteq_\infty$ or $\sqsubseteq_{lex}$, then Γ cannot contain two obtuse triangles sharing the same longest edge.

Proof. Consider two obtuse triangles $acb, abd \in {K_\mathbf{P}}^{(2)}$ with same longest edge ab. The triangles adc and bcd as well as the tetrahedron $abcd$ also belong to $K_\mathbf{P}$. It follows that $acb + abd$ is homologous to $adc + bcd$ in $K_\mathbf{P}$. Therefore, if some chain Γ contains acb and abd, then the chain $\Gamma' = \Gamma - acb - abd + adc + bcd$ is homologous to Γ. One checks easily that, while $\Gamma' \neq \Gamma$, one has $\Gamma' \sqsubseteq_\infty \Gamma$ and $\Gamma' \sqsubseteq_{lex} \Gamma$. Γ can therefore not be minimum in $\mathcal{D}$ for either order $\sqsubseteq_\infty$ or $\sqsubseteq_{lex}$. □

The following lemma states that minima taken under either $\sqsubseteq_{lex}$ or $\sqsubseteq_\infty$ are identical.

Lemma 4.18 (Lemma 7.8 in [CLV19a])**.** *Let $\mathbf{P} \subset \mathbb{R}^d$ be a finite set of points satisfying Condition 2 and let $K_{\mathbf{P}}$ be the full complex over $\mathbf{P}$. For a 1-boundary $\beta \in \mathbf{B}_1(K_{\mathbf{P}})$, define the set $\mathcal{D}$ of 2-chains bounded by β:*

$$\mathcal{D} \underset{def.}{=} \left\{\Gamma \in \mathbf{C}_2(K) \mid \partial\Gamma = \beta\right\}$$

The minima on $\mathcal{D}$ under either $\sqsubseteq_\infty$ or $\sqsubseteq_{lex}$ are identical:

$$\min_{\sqsubseteq_\infty} \mathcal{D} = \min_{\sqsubseteq_{lex}} \mathcal{D}$$

Finally, we can state the following characterization of Delaunay triangulation as optimal lexicographic chain induced by the total order on 2-simplices given in Lemma 4.14.

Proposition 3. *Let $\mathbf{P} = \{P_1, \dots, P_N\} \subset \mathbb{R}^2$ with $N \geq 3$ be in general position and let K be any 2-dimensional complex containing the Delaunay triangulation of $\mathbf{P}$. Denote by $\beta_{\mathbf{P}} \in \mathbf{B}_1(K)$ the 1-boundary made of edges belonging to the boundary of convex hull $\mathcal{CH}(\mathbf{P})$. Define the chain*

$$\Gamma_{\min} \underset{def.}{=} \min_{\sqsubseteq_{lex}} \{\Gamma \in \mathbf{C}_2(K), \partial\Gamma = \beta_{\mathbf{P}}\}$$

The support $|\Gamma_{\min}|$ of $\Gamma_{\min}$ corresponds to the 2-simplices of the Delaunay triangulation of $\mathbf{P}$.

Proof. Let $\Gamma_{\mathcal{D}el(\mathbf{P})}$ be the 2-chain with coefficients 1 on each Delaunay 2-simplex. We will show that $\Gamma_{\mathcal{D}el(\mathbf{P})}$ is the lexicographic optimal chain over all 2-chains bounding $\beta_{\mathbf{P}}$ in the full complex $K_{\mathbf{P}}$. The result will be *a fortiori* true for any subcomplex K containing $\Gamma_{\mathcal{D}el(\mathbf{P})}$.

We know that $\partial\Gamma_{\mathcal{D}el(\mathbf{P})} = \beta_{\mathbf{P}}$, and from Proposition 2, for any $p \geq 1$:

$$\Gamma_{\mathcal{D}el(\mathbf{P})} = \arg\min_{\partial\Gamma=\beta_{\mathbf{P}}} \|\Gamma\|_{(p)}$$

It follows that, for any $\Gamma \in \mathbf{C}_2(K_{\mathbf{P}})$,

$$\partial\Gamma = \beta_{\mathbf{P}} \implies \forall p \geq 1, \left\|\Gamma_{\mathcal{D}el(\mathbf{P})}\right\|_{(p)} \leq \left\|\Gamma\right\|_{(p)} \implies \Gamma_{\mathcal{D}el(\mathbf{P})} \sqsubseteq_\infty \Gamma$$

Therefore, the chain $\Gamma_{\mathcal{D}el(\mathbf{P})}$, whose support corresponds to the Delaunay triangulation, verifies:

$$\Gamma_{\mathcal{D}el(\mathbf{P})} = \min_{\sqsubseteq_\infty} \{\Gamma \in \mathbf{C}_2(K_{\mathbf{P}}), \partial\Gamma = \beta_{\mathbf{P}}\}$$

Finally, the equality of minima under $\sqsubseteq_\infty$ and $\sqsubseteq_{lex}$ (Lemma 4.18) completes the proof. □

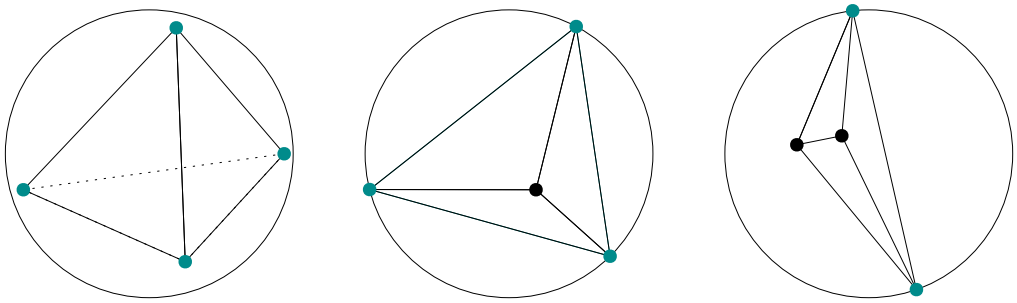
4.2.4 Characterizing regular triangulations as lexicographic optimal chains

In this section, we simply state a generalization of the result of the previous section for regular triangulations of arbitrary dimension.

We consider a set of weighted points $\mathbf{P}$ that verifies the following generic condition.

Condition 3. *We say that $\mathbf{P} = \left\{(P_1, \mu_1), \dots, (P_N, \mu_N)\right\} \subset \mathbb{R}^d \times \mathbb{R}$ is in general position if it satisfies Condition 1 and for a k-simplex σ and a k'-simplex σ' in $K_{\mathbf{P}}$ with $1 \leq k, k' \leq n$, one has:*

$$\mu_C(\sigma) = \mu_C(\sigma') \Rightarrow \sigma = \sigma' \tag{4.10}$$



$\dim \Theta(\sigma) = 3$ $\dim \Theta(\sigma) = 2$ $\dim \Theta(\sigma) = 1$

Figure 4.13: For zero weights, three generic cases can define different $\Theta(\sigma)$ for a tetrahedron. In each case, the simplex $\Theta(\sigma)$ is spanned by the highlighted vertices which contribute to the smallest enclosing ball. Note however that, for arbitrary weights, a fourth situation is possible, where $\dim \Theta(\sigma) = 0$.

From Definition 4.3 of generalized circumspheres and smallest enclosing balls, we can show that any d-simplex σ admits a unique inclusion minimal face $\Theta(\sigma)$ having the same bounding weight as the simplex σ:

$$(P_B, \mu_B)(\sigma) = (P_B, \mu_B)(\Theta(\sigma))$$

Figure 4.13 illustrates the different possibilities for $\Theta(\sigma)$ in the case $d = 3$ with zero weights.

We can then construct for each d-simplex σ a sequence of inclusion-increasing faces $\Theta_k(\sigma)$ defined incrementally, for $0 \leq k \leq \dim \sigma - \dim \Theta(\sigma)$, by:

$$\begin{aligned} \Theta_0(\sigma) &\underset{\text{def.}}{=} \Theta(\sigma) \\ \Theta_k(\sigma) &\underset{\text{def.}}{=} \operatorname*{arg\,min}_{\substack{\Theta_{k-1}(\sigma) \preceq \tau \preceq \sigma \\ \dim \tau = \dim \Theta_{k-1}(\sigma)+1}} \mu_C(\tau) \end{aligned}$$

The order on d-simplices will consider the circumweights of this sequence $(\Theta_k(\sigma))_k$, denoted:

$$\mu_k(\sigma) \underset{\text{def.}}{=} \mu_C(\Theta_k(\sigma))$$

Note that in particular,

$$\mu_0(\sigma) = \mu_C(\Theta(\sigma)) = \mu_B(\sigma) \tag{4.11}$$

The following relation defines a generalized total order on d-simplices over weighted points:

$$\sigma_1 \leq \sigma_2 \underset{\text{def.}}{\iff} \sigma_1 = \sigma_2 \quad \text{or} \quad \begin{cases} \mu_0(\sigma_1) < \mu_0(\sigma_2) \\ \quad \text{or} \\ \exists k \geq 1,\ \mu_k(\sigma_1) > \mu_k(\sigma_2) \\ \text{and } \forall j,\ 0 \leq j < k,\ \mu_j(\sigma_1) = \mu_j(\sigma_2) \end{cases} \tag{4.12}$$

Note first that for two d-simplices σ_1, σ_2, thanks to Condition 3 and Equation (4.11):

$$\mu_B(\sigma_1) = \mu_B(\sigma_2) \Rightarrow \Theta(\sigma_1) = \Theta(\sigma_2)$$

which implies that, when $\mu_B(\sigma_1) = \mu_B(\sigma_2)$, the range of values of $(\mu_k(\sigma_1))_k$ and $(\mu_k(\sigma_2))_k$ are the same and the order is well-defined.

We can verify, for zero weights in dimension 2, that this order is equivalent to the one presented in Lemma 4.14, the only difference being the comparison uses squared radii. Indeed, as mentioned in Observation 4.4, for zero weights, we have:

$$\mu_0(\sigma) = \mu_B(\sigma) = \mathrm{R_B}^2(\sigma)$$

For two distinct triangles σ_1, σ_2, the order first compares the squared radii of the smallest enclosing circle of each triangle. For obtuse triangles, these radii can generically be equal when they share their longest edge, which corresponds to $\Theta(\sigma_1) = \Theta(\sigma_2)$. In this case, the tie is broken by comparing in reverse order the squared circumradii of the triangles.

The previous characterization result for 2-dimensional Delaunay triangulation can then be extended to regular triangulations of any dimension.

Theorem 4.19 (Theorem 3.1 in [CLV19b])**.** *Let $\mathbf{P} = \{(P_1, \mu_1), \ldots, (P_N, \mu_N)\} \subset \mathbb{R}^d \times \mathbb{R}$, with $N \geq d+1$ be in general position and let $K_{\mathbf{P}}$ be the d-dimensional full complex over $\mathbf{P}$. Denote by $\beta_{\mathbf{P}} \in \mathbf{B}_{d-1}(K_{\mathbf{P}})$ the $(d-1)$-boundary made of simplices belonging to the boundary of $\mathcal{CH}(\mathbf{P})$. Define the chain*

$$\Gamma_{\min} \underset{def.}{=} \min_{\sqsubseteq_{lex}} \{\Gamma \in \mathbf{C}_d(K_{\mathbf{P}}), \partial\Gamma = \beta_{\mathbf{P}}\}$$

The support $|\Gamma_{\min}|$ of $\Gamma_{\min}$ corresponds to the d-simplices of the regular triangulation of $\mathbf{P}$.

4.3 Problem statement and solutions in arbitrary dimension

In this section, after stating two lexicographic optimal chain problems over coefficients in $\mathbb{Z}_2$, polynomial-time solutions are derived from algorithms akin to Gauss elimination and very similar to the computation of persistent homology.

We assume a total order on k-simplices is given, inducing a lexicographic order $\sqsubseteq_{lex}$ on k-chains as defined in Definition 4.6. The choice of coefficients in $\mathbb{Z}_2$ simplifies the presentation of matrix reduction algorithms, however these results could be extended to any coefficient field $\mathbb{F}$ with the same algorithmic complexity by considering matrices in *Smith normal forms* [EH10]. These extensions to different coefficient fields are especially useful to carry an information of orientation on the simplices.

In $\mathbb{Z}_2$ arithmetic, the addition and subtraction operations are of course equivalent. However, both symbols appear in this section: this is justified by the fact that one operation might be more natural than the other if the problem was considered in an arbitrary coefficient field.

4.3.1 Problem statement

The Optimal Homologous Chain Problem (OHCP), introduced in [DHK11], can be stated as follows.

Problem 3 (OHCP)**.** *Given a k-chain Γ_0 in a simplicial complex K and a set of weights given by a diagonal matrix W of appropriate dimension, find the L^1-norm minimal chain $\Gamma_{\min}$ homologous to Γ_0:*

$$\Gamma_{\min} = \arg\min_{\Gamma} ||W \cdot \Gamma||_1 \text{ such that } \Gamma \text{ is homologous to } \Gamma_0$$

It has been shown that this problem, in an arbitrary simplicial complex with chain coefficients in $\mathbb{Z}_2$, is NP-hard [CF11, CEN09].

The first problem studied in this section, denoted Lexicographic Optimal Homologous Chain Problem (Lex-OHCP), can be seen as a particular instance of OHCP.

Problem 4 (Lex-OHCP)**.** *Given a simplicial complex K and a k-chain $\Gamma_0 \in \mathbf{C}_k(K)$, find the unique chain $\Gamma_{\min}$ defined by:*

$$\Gamma_{\min} = \min_{\sqsubseteq_{lex}} \Big\{ \Gamma \in \mathbf{C}_k(K) \mid \exists A \in \mathbf{C}_{k+1}(K), \Gamma - \Gamma_0 = \partial_{k+1} A \Big\}$$

Observation 4.20. *By verifying that $\Gamma_{\min} = M_{lex}^{\mathbf{B}_k(K)}(\Gamma_0)$, the existence and unicity of Problem 4 is immediate from Lemma 4.11.*

Observation 4.21. *Formulating Lex-OHCP as OHCP with coefficients in $\mathbb{Z}_2$ requires ordering simplices along their total order and taking a weight matrix W with generic term $W_{ii} = 2^i$, in which case the L^1-norm minimization is equivalent to a minimization along the lexicographic order.*

The second problem of interest in this section is finding a lexicographic optimal chain under imposed boundary.

Problem 5. *Given a simplicial complex K with a total order on the k-simplices and a $(k-1)$-cycle A, check if A is a boundary:*

$$B_A \underset{def.}{=} \{\Gamma \in \mathbf{C}_k(K) \mid \partial_k \Gamma = A\} \neq \emptyset$$

If it is the case, find the minimal k-chain Γ bounded by A:

$$\Gamma_{\min} = \min_{\sqsubseteq_{lex}} B_A$$

4.3.2 Boundary matrix reduction

The following terminology will be useful when discussing the links with matrix reduction algorithms.

Definition 4.22. *A k-chain $\Gamma_0 \in \mathbf{C}_k(K)$ is said **reducible** if there is a k-chain $\Gamma \in \mathbf{C}_k(K)$ (called **reduction**) and a $(k+1)$-chain $A \in \mathbf{C}_{k+1}(K)$ such that:*

$$\Gamma \sqsubset_{lex} \Gamma_0 \quad \text{and} \quad \Gamma - \Gamma_0 = \partial_{k+1} A$$

*If this property cannot be verified, the k-chain Γ_0 is said **irreducible**. If Γ_0 is reducible, we call **total reduction** of Γ_0 the unique irreducible reduction of Γ_0. If Γ_0 is irreducible, Γ_0 is said to be its own total reduction.*

Note that Problem 4 can be reformulated as finding the total reduction of Γ_0.

With $m = \dim \mathbf{C}_k(K)$ and $n = \dim \mathbf{C}_{k+1}(K)$, we consider the m-by-n boundary matrix ∂_{k+1} with entries in $\mathbb{Z}_2$. We enforce that rows of the matrix are ordered according to the given strict total order on k-simplices $\sigma_1 < \cdots < \sigma_m$, where the k-simplex σ_i is the basis element corresponding to the i^{th} row of the boundary matrix. The order of columns, corresponding to an order on $(k+1)$-simplices, is not relevant for this section and can be chosen arbitrarily.

For a matrix R, the index of the lowest non-zero coefficient of a column R_j is denoted by $\mathrm{low}(j)$, or sometimes $\mathrm{low}(R_j)$ when we want to make explicit which matrix is considered. This index is not defined for a column whose coefficients are all zero.

Algorithm 2 is a slightly modified version of the boundary reduction algorithm presented in [EH10]. Indeed, for our purpose, we do not need the boundary matrix storing all the simplices of all dimensions, as does [EH10], and apply the algorithm to the submatrix $\partial_{k+1} : \mathbf{C}_{k+1}(K) \to \mathbf{C}_k(K)$. Note that this algorithm can be seen as a matrix factorization $R = \partial_{k+1} V$, where V is not explicitly constructed. One checks easily that Algorithm 2 has $\mathcal{O}(mn^2)$ time complexity.

Algorithm 2: Reduction algorithm for the ∂_{k+1} matrix

Input : Boundary matrix ∂_{k+1}
Output: Reduced matrix R
R = ∂_{k+1}
for $j \leftarrow 1$ **to** n **do**
  while $R_j \neq 0$ **and** $\exists j_0 < j$ *with* $\text{low}(j_0) = low(j)$ **do**
    $R_j \leftarrow R_j + R_{j_0}$
  end
end

We now introduce a few lemmas useful for solving Problem 4. We allow ourselves to consider each column R_j of the matrix R, formally an element of $\mathbb{Z}_2^m$, as the corresponding k-chain in the basis $(\sigma_1, \ldots, \sigma_m)$.

Lemma 4.23. *A k-chain Γ is reducible if and only if at least one of its k-simplices is reducible.*

Proof. If there is a reducible k-simplex $\sigma \in \Gamma$, Γ is reducible by the k-chain $\Gamma' = \Gamma - \sigma + Red(\sigma)$, where $Red(\sigma)$ is a reduction for σ.
We now suppose Γ to be reducible. Let $\Gamma' \sqsubset_{lex} \Gamma$ be a reduction for Γ and A the $(k+1)$-chain such that $\Gamma - \Gamma' = \partial A$. We denote $\sigma_{\max} = \max\{\sigma \in \Gamma - \Gamma'\}$. Note that $\sigma_{\max}$ is homologous to $\Gamma' - \Gamma + \sigma_{\max}$. The chain $\Gamma' - \Gamma + \sigma_{\max}$ only contains simplices smaller than $\sigma_{\max}$, by definition of the lexicographic order (Definition 4.6). We have thus shown that if Γ is reducible, it contains at least one simplex that is reducible. □

Lemma 4.24. *After matrix reduction (Algorithm 2), a non-zero column $R_j \neq 0$ can be described as*

$$R_j = \sigma_{\text{low}(j)} + \Gamma, \text{ where } \Gamma \text{ is a reduction for } \sigma_{\text{low}(j)}.$$

Proof. As all matrix operations performed on R by the reduction algorithm are linear, each non-zero column R_j of R is in the image of ∂_{k+1}. Therefore, there exist a $(k+1)$-chain A such that $R_j = \sigma_{\text{low}(j)} + \Gamma = \partial_{k+1} A$, which, is equivalent in $\mathbb{Z}_2$ to $\Gamma - \sigma_{\text{low}(j)} = \partial_{k+1} A$. By definition of the low of a column, we also have immediately: $\Gamma \sqsubset_{lex} \sigma_{\text{low}(j)}$. For each non-zero column, the largest simplex is therefore reducible by the other k-simplices of the column. □

Lemma 4.25. *After matrix reduction (Algorithm 2), there is a one-to-one correspondence between the reducible k-simplices and non-zero columns of R:*

$$\sigma_i \in \mathbf{C}_k(K) \text{ is reducible } \iff \exists j, 1 \leq j \leq n, R_j \neq 0 \text{ and } \text{low}(j) = i$$

Proof. Lemma 4.24 shows immediately that the simplex corresponding to the lowest index of a non-zero column is reducible.

Suppose now that a k-simplex $\widetilde{\sigma}$ is reducible and let $\widetilde{\Gamma}$ be a reduction of it: $\widetilde{\sigma} - \widetilde{\Gamma} = \partial_{k+1} A$ for some $A \in \mathbf{C}_{k+1}(K)$ and $\widetilde{\Gamma} \sqsubset_{lex} \widetilde{\sigma}$. Algorithm 2 realizes the matrix factorization $R = \partial_{k+1} \cdot V$, where matrix V is invertible [EH10]. It follows that $\text{Im}\, R = \text{Im}\, \partial_{k+1}$. Therefore, non-zero columns of R span $\text{Im}\, \partial_{k+1}$ and since $\widetilde{\sigma} - \widetilde{\Gamma} = \partial_{k+1} B \in \text{Im}\, \partial_{k+1}$, there is a family $(R_j)_{j \in J} = (\sigma_{\text{low}(j)}, \Gamma_j)_{j \in J}$ of columns of R such that :

$$\widetilde{\sigma} - \tilde{\Gamma} = \sum_{j \in J} \sigma_{\text{low}(j)} + \Gamma_j$$

Every $\sigma_{\text{low}(j)}$ represents the largest simplex of a column, and Γ_j a reduction chain for $\sigma_{\text{low}(j)}$. As observed in [EH10, Section VII.1], one can check that the low indices in R

are unique after the reduction algorithm. Therefore, there is a $j_{\max} \in J$ such that for all j in $J \setminus \{j_{\max}\}$, $\text{low}(j) < \text{low}(j_{\max})$, which implies:

$$\sigma_{j_{\max}} = \max\{\sigma \in \sum_{j \in J} \sigma_{\text{low}(j)} + \Gamma_j\} = \max\left\{\sigma \in \widetilde{\sigma} - \widetilde{\Gamma}\right\} = \widetilde{\sigma}$$

We have shown that for the reducible simplex $\widetilde{\sigma}$, there is a non-zero column $R_{j_{\max}}$ with $\widetilde{\sigma} = \sigma_{\text{low}(j_{\max})}$ as its largest simplex. □

4.3.3 Total reduction algorithm

Combining the three previous lemmas give the intuition on how to construct the total reduction solving Problem 4: Lemma 4.23 allows to consider each simplex individually, Lemma 4.25 characterizes the reducible nature of a simplex using the reduced boundary matrix and Lemma 4.24 describes a column of the reduction boundary matrix as a simplex and its reduction. We now present Algorithm 3, referred to as the **total reduction algorithm**. For a k-chain Γ, $\Gamma[i] \in \mathbb{Z}_2$ denotes the coefficient of the i^{th} simplex in the chain Γ.

Algorithm 3: Total reduction algorithm

```
Inputs: A k-chain Γ, the reduced boundary matrix R
for i ← m to 1 do
    if Γ[i] ≠ 0 and ∃j ∈ [1, n] with low(j) = i in R then
        Γ ← Γ + R_j
    end
end
```

Proposition 4. *Algorithm 3 finds the total reduction of a given k-chain in $\mathcal{O}(m^2)$ time complexity.*

Proof. In Algorithm 3, let Γ_{i-1} be the value of the variable Γ after iteration i. Since the iteration counter i decreases from m to 1, the input and output of the algorithm are respectively Γ_m and Γ_0. At each iteration, Γ_{i-1} is either equal to Γ_i or $\Gamma_i + R_j$. Since $R_j \in \text{Im}\,\partial_{k+1}$, Γ_{i-1} is in both cases homologous to Γ_i. Therefore, Γ_0 is homologous to Γ_m. We are left to show that Γ_0 is irreducible. From Lemma 4.23, it is enough to check that it does not contain any reducible simplex.
Let σ_i be a reducible simplex and let us show that $\sigma_i \notin \Gamma_0$. Two possibilities may occur:

- if $\sigma_i \in \Gamma_i$, then $\Gamma_{i-1} = \Gamma_i + R_j$. Since $\text{low}(j) = i$, we have $\sigma_i \in R_j$ and therefore $\sigma_i \notin \Gamma_{i-1}$.
- if $\sigma_i \notin \Gamma_i$, then $\Gamma_{i-1} = \Gamma_i$ and again $\sigma_i \notin \Gamma_{i-1}$.

Furthermore, from iterations $i-1$ to 1, the added columns R_j contain only simplices smaller than σ_i and therefore $\sigma_i \notin \Gamma_{i-1} \Rightarrow \sigma_i \notin \Gamma_0$.

Observe that using an auxiliary array allows to determine in constant time, if it exists, the column index j corresponding to a lowest element $\text{low}(j)$. The column addition nested inside the loop lead to a $\mathcal{O}(m^2)$ time complexity for Algorithm 3. □

It follows that Problem 4 can be solved in $\mathcal{O}(mn^2)$ time complexity, by applying successively Algorithm 2 and Algorithm 3, or in $\mathcal{O}(N^3)$ complexity if N is the size of the simplicial complex.

4.3.4 A basis of cycles from matrix reduction

In order to derive an algorithmic solution to Problem 5, we revisit the boundary matrix reduction algorithm presented in Section 4.3.2. Unlike Problem 4, which uses the ∂_{k+1} boundary operator, we now consider ∂_k, meaning the given total order on k-simplices applies to the columns of the matrix. An arbitrary order can be taken for the $(k-1)$-simplices to build the matrix ∂_k. We mentioned that the matrix reduction algorithm could be seen as a matrix factorization problem $R = \partial_k \cdot V$. As the order on $(k-1)$-simplices is chosen arbitrarily, the properties of the columns of the reduced matrix presented in Section 4.3.2 are not useful for the minimization of Problem 5. However, the rows of the transformation matrix V do follow the given simplicial ordering. Very similar to Algorithm 2, Algorithm 4 shows how to compute the transformation matrix V.

Algorithm 4: Boundary factorization algorithm

Input : Boundary matrix ∂_k
Outputs: Reduced matrix R, Transformation matrix V
$R = \partial_k$
$V =$ Identity
for $j \leftarrow 1$ **to** n **do**
  while $R_j \neq 0$ **and** $\exists j_0 < j$ *with* $\text{low}(j_0) = low(j)$ **do**
    $R_j \leftarrow R_j + R_{j_0}$
    $V_j \leftarrow V_j + V_{j_0}$
  end
end

After calling Algorithm 4, the matrices R and V are used to construct a basis of cycles. Indeed, the number of zero columns of R is the dimension of $\mathbf{Z}_k = \operatorname{Ker} \partial_k$, which we denote by n^{Ker}. By selecting all columns in V corresponding to zero columns in R, we obtain the matrix V^{Ker}, whose columns $V_1^{\text{Ker}}, \ldots, V_{n^{\text{Ker}}}^{\text{Ker}}$ form a basis of $\mathbf{Z}_k$.

Similarly to R, we show the lows of the columns of V^{Ker} are unique.

Lemma 4.26. *Indices* $\left\{ \text{low}(V_i^{\text{Ker}}) \right\}_{i \in [1, n^{\text{Ker}}]}$ *are unique:*

$$i \neq j \Rightarrow \text{low}(V_i^{\text{Ker}}) \neq \text{low}(V_j^{\text{Ker}})$$

If $A \in \operatorname{Ker} \partial_k \setminus \{0\}$*, there exists a unique column* $V_{\text{max}}^{\text{Ker}}$ *of* V^{Ker} *with* $\text{low}(V_{\text{max}}^{\text{Ker}}) = \text{low}(A)$.

Proof. Before the boundary matrix reduction algorithm, the initial matrix V is the identity: the low indices are therefore unique. During iterations of the algorithm, the matrix V is right-multiplied by an column-adding elementary matrix $L_{j_0,j}$, adding column j_0 to j with $j_0 < j$.

$$L_{j_0,j} = \begin{array}{c} \\ \\ \\ \\ \\ \\ \end{array} \begin{array}{c} \begin{array}{ccccccc} & & & & j & & \\ \end{array} \\ \begin{bmatrix} 1 & & & & & & \\ & 1 & & & 1 & & \\ & & \ddots & & & & \\ & & & & 1 & & \\ & & & & & \ddots & \\ & & & & & & 1 \end{bmatrix} \end{array} \begin{array}{l} \\ \\ \\ \\ \\ \\ j_0 \end{array}$$

Therefore, the indices $\{\text{low}(V_i), V_i \in V\}$ stay on the diagonal during the reduction algorithm and are therefore unique. The restriction of V to V^{Ker} does not change this property.

If $A \in \operatorname{Ker} \partial_k \setminus \{0\}$, A can be written as a non-zero linear combination of columns $(V_i^{\text{Ker}})_{i \in I}$ of V^{Ker}. Call $i_{\max} = \text{low}(A)$ the index of the largest element $\sigma_{i_{\max}}$ in A. Suppose, for a contradiction, that no column of $(V_i^{\text{Ker}})_{i \in I}$ has $i_{\max}$ as its low index. By existence of $\sigma_{i_{\max}}$ in A, there is an odd number of columns $V_j^{\text{Ker}} \in (V_i^{\text{Ker}})_{i \in I}$ satisfying $\sigma_{i_{\max}} \in V_j^{\text{Ker}}$ with $\text{low}(V_j^{\text{Ker}}) > i_{\max}$. We have shown however that the lows of V^{Ker} are unique, which implies the lows of columns V_j^{Ker} would appear in A: this contradicts the definition of $i_{\max}$ as the low of A. □

4.3.5 Lexicographic-minimal chain under imposed boundary

This section studies a variant of Lex-OHCP (Problem 4) in order to solve the subsequent problem of finding a minimal k-chain bounding a given $(k-1)$-cycle (Problem 5).

Problem 6. *Given a simplicial complex K with a total order on the k-simplices and a k-chain $\Gamma \in \mathbf{C}_k(K)$, find:*

$$\Gamma_{\min} = \min_{\sqsubseteq_{lex}} \{\Gamma' \in \mathbf{C}_k(K) \mid \partial_k \Gamma' = \partial_k \Gamma\}$$

Indeed, in Problem 5, once a representative Γ in the set $B_A \neq \varnothing$ such that $\partial_k \Gamma = A$ has been found, we are then taken back to Problem 6 to find the minimal k-chain $\Gamma_{\min}$ such that $\partial_k \Gamma_{\min} = \partial_k \Gamma = A$.

We apply a similar total reduction algorithm as previously introduced but using the matrix V^{Ker} defined in Section 4.3.4. In the following algorithm, $m = \dim \mathbf{C}_k(K)$.

Algorithm 5: Total reduction variant

```
Inputs : A k-chain Γ and V^Ker
for i ← m to 1 do
    if Γ[i] ≠ 0 and ∃j ∈ [1, n^Ker] with low(j) = i in V^Ker then
        Γ ← Γ + V_j^Ker
    end
end
```

Proposition 5. *Algorithm 5 computes the solution for Problem 6 in $\mathcal{O}(m^2)$ time complexity.*

Proof. The proof is similar to the one of Proposition 4.
In Algorithm 5, we denote by Γ_{i-1} the value of variable Γ after iteration i. Since iteration counter i is decreasing from m to 1, the input and output of the algorithm are respectively Γ_m and Γ_0. Since $V_j^{\text{Ker}} \in \operatorname{Ker} \partial_k$, at each iteration $\partial \Gamma_{i-1} = \partial \Gamma_i$ therefore $\partial \Gamma_0 = \partial \Gamma_m$. We are left to show the algorithm's result is the minimal solution.

Suppose there is $\Gamma^\star$ such that $\partial_k \Gamma^\star = \partial_k \Gamma$ and $\Gamma^\star \sqsubset_{lex} \Gamma_0$. Let's consider the difference $\Gamma_0 - \Gamma^\star$, and its largest element index $\text{low}(\Gamma_0 - \Gamma^\star) = i$, with $\sigma_i \in \Gamma_0$ and $\sigma_i \notin \Gamma^\star$ by Definition 4.6 of the lexicographic order. As $\Gamma_0 - \Gamma^\star \in \operatorname{Ker} \partial_k \setminus \{0\}$, there has to be a column V_j^{Ker} in V^{Ker} where $\text{low}\left(V_j^{\text{Ker}}\right) = i$, from Lemma 4.26. Two possibilities may occur at iteration i:

- if $\sigma_i \in \Gamma_i$, then $\Gamma_{i-1} = \Gamma_i + V_j^{\text{Ker}}$. Since $i = \text{low}(j)$, we have $\sigma_i \in V_j^{\text{Ker}}$ and therefore $\sigma_i \notin \Gamma_{i-1}$.

– if $\sigma_i \notin \Gamma_i$, then $\Gamma_{i-1} = \Gamma_i$ and again $\sigma_i \notin \Gamma_{i-1}$.

However, from iterations $i-1$ to 1, the added columns V_j^{Ker} contains only simplices with indices smaller than i and therefore we obtain $\sigma_i \notin \Gamma_{i-1} \Rightarrow \sigma_i \notin \Gamma_0$, a contradiction to the definition of σ_i as the largest element of $\Gamma^\star - \Gamma_0$. □

4.3.6 Finding a representative chain given a cycle

As previously mentioned, solving Problem 5 requires deciding if the set

$$B_A = \{\Gamma \in \mathbf{C}_k(K) \mid \partial_k \Gamma = A\}$$

is empty and in case it is not empty, finding an element of the set B_A. Algorithm 5 can then be used to minimize this element under imposed boundary. In the following algorithm, $m = \dim \mathbf{C}_{k-1}(K)$ and $n = \dim \mathbf{C}_k(K)$.

Algorithm 6: Finding a representative

```
Inputs : A (k − 1)-chain A, a boundary matrix R reduced by V
Γ_0 ← ∅
A_0 ← A
for i ← m to 1 do
    if A_0[i] ≠ 0 then
        if ∃j ∈ [1, n] with low(j) = i in R then
            A_0 ← A_0 − R_j
            Γ_0 ← Γ_0 + V_j
        else
            The set B_A is empty.
        end
    end
end
```

Proposition 6. *Algorithm 6 decides if the set B_A is non-empty, and in that case, finds a representative Γ_0 such that $\partial_k \Gamma_0 = A$ in $\mathcal{O}(m^2)$ time complexity.*

Proof. We start by two trivial observations from the definition of a reduction. First, A is a boundary if and only if its total reduction is the null chain. Second, if a non-null chain is a boundary, then its greatest simplex is reducible.

If, at iteration i, $A_0[i] \neq 0$, then σ_i is the greatest simplex in A_0. In the case R has no column R_j such that $low(j) = i$, σ_i is not reducible by Lemma 4.25 and therefore A_0 is not a boundary. Since A and A_0 differ by a boundary (added columns of R), A is not a boundary either. This means the set B_A is empty.
The main difference with the previous chain reduction is we keep track of the column operations in Γ_0. If the total reduction of A is null, we have found a linear combination $A = \sum_{j \in J} R_j$. We have also computed Γ_0 as the sum of the corresponding columns in V: $\Gamma_0 = \sum_{j \in J} V_j$. As $R = \partial_k \cdot V$, we can verify that

$$\partial_k \Gamma_0 = \partial_k \left(\sum_{j \in J} V_j \right) = \sum_{j \in J} R_j = A$$

□

4.4 Lexicographic optimal homologous cycle in codimension 1

We've seen in the previous section that the Lexicographic Optimal Homologous Chain Problem (Lex-OHCP) can be solved for any arbitrary simplicial complex in time cubic in the size of the complex. However, having in mind applications of reconstruction of 2-surfaces in 3-dimensional space, we look at ways of further improving this complexity. In this section, we study the problem of finding a lexicographic optimal homologous cycle in codimension 1 of a pseudomanifold complex K, allowing to formulate the problem as a particular mincut problem on the dual graph of the complex, which we call *lexicographic mincut* and will be solved in $\mathcal{O}(n \log n)$ time complexity, where n stands for the size of the complex K. We assume the coefficient field is $\mathbb{Z}_2$, allowing to consider chains as sets of simplices.

The idea of considering the minimal cut problem on the dual graph has been previously studied [Sul90]. In particular, Chambers et al. [CEN09] have considered graph duality to derive complexity results for the computation of L^1-optimal homologous cycles on 2-manifolds. Chen et al. [CF11] also use a reduction to a minimum cut problem on a dual graph to compute minimal non-null homologous cycles on d-complexes embedded in $\mathbb{R}^d$. Their polynomial algorithm (Theorem 5.2.3 in [CF11]) for computing a homology class representative of minimal radius is reminiscent of our algorithm for computing lexicographic minimal representatives (Algorithm 6). In a recent work [DHM20], Dey et al. consider the dual graph of pseudomanifolds in order to obtain polynomial time algorithms for computing minimal persistent cycles. Since they consider arbitrary weights, they obtain the $\mathcal{O}(n^2)$ complexity of best known minimum cut/maximum flow algorithms [Orl13].

4.4.1 Dual formulation as lexicographic min-cut

Given a strongly connected d-pseudomanifold $\mathcal{M}$ (see definition in Section 4.1.1) and $\tau_1 \neq \tau_2$ two d-simplices of $\mathcal{M}$, we consider a special case of Problem 4 where $K = \mathcal{M} \setminus \{\tau_1, \tau_2\}$ and $A = \partial\tau_1$:

Problem 7. *Given a strongly connected d-pseudomanifold $\mathcal{M}$ with a total order on the $(d-1)$-simplices and two distinct d-simplices (τ_1, τ_2) of $\mathcal{M}$, find:*

$$\Gamma_{\min} = \min_{\sqsubseteq_{lex}} \Big\{ \Gamma \in \mathbf{C}_{d-1}(\mathcal{M}) \mid \exists B \in \mathbf{C}_d(\mathcal{M} \setminus \{\tau_1, \tau_2\}), \Gamma - \partial\tau_1 = \partial B \Big\}$$

Definition 4.27. *Seeing a graph G as a 1-dimensional simplicial complex, we define the **coboundary operator** $\partial^0 : \mathbf{C}_0(G) \to \mathbf{C}_1(G)$ as the linear operator defined by the transpose of the matrix of the boundary operator $\partial_1 : \mathbf{C}_1(G) \to \mathbf{C}_0(G)$ in the canonical basis of simplices.*[1]

For a given graph $G = (\mathcal{V}, \mathcal{E})$, $\mathcal{V}$ and $\mathcal{E}$ respectively denote its vertex and edge sets. For a $(d-1)$-chain $\alpha \in \mathbf{C}_{d-1}(\mathcal{M})$ and a d-chain $\beta \in \mathbf{C}_d(\mathcal{M})$, $\widetilde{\alpha}$ and $\widetilde{\beta}$ denote their respective dual 1-chain and dual 0-chain in the dual graph $G(\mathcal{M})$ of $\mathcal{M}$ (see definition in Section 4.1.1). We easily see that:

Observation 4.28. *For a set of vertices $\mathcal{V}_0 \subset \mathcal{V}$, $\partial^0\mathcal{V}_0$ is exactly the set of edges in $G = (\mathcal{V}, \mathcal{E})$ that connect vertices in $\mathcal{V}_0$ with vertices in $\mathcal{V} \setminus \mathcal{V}_0$.*

[1] In order to avoid to introduce non essential formal definitions, the coboundary operator is defined over chains since, for finite simplicial complexes, the canonical inner product defines a natural bijection between chains and cochains.

Observation 4.29. *Let $\mathcal{M}$ be a d-pseudomanifold. If $\alpha \in \mathbf{C}_{d-1}(\mathcal{M})$ and $\beta \in \mathbf{C}_d(\mathcal{M})$, then $\widetilde{\alpha} = \partial^0 \widetilde{\beta} \iff \alpha = \partial_d \beta$.*

The order on $(d-1)$-simplices of a d-pseudomanifold $\mathcal{M}$ naturally defines a corresponding order on the edges of the dual graph: $\tau_1 < \tau_2 \iff \widetilde{\tau}_1 < \widetilde{\tau}_2$. This order extends similarly to a lexicographic order $\sqsubseteq_{lex}$ on sets of edges (or, equivalently, 1-chains over $\mathbb{Z}_2$ coefficients) in the graph.

In what follows, we say a set of edges $\widetilde{\Gamma}$ **disconnects** $\widetilde{\tau}_1$ and $\widetilde{\tau}_2$ in the graph $(\mathcal{V}, \mathcal{E})$ if $\widetilde{\tau}_1$ and $\widetilde{\tau}_2$ are not in the same connected component of the graph $(\mathcal{V}, \mathcal{E} \setminus \widetilde{\Gamma})$.

Given a graph with weighted edges and two vertices, the min-cut/max-flow problem [EK72, Orl13] consists in finding the minimum cut (i.e. set of edges) disconnecting the two vertices, where minimum is meant as minimal sum of weights of cut edges. We consider a similar problem where the minimum is meant in term of a lexicographic order: the Lexicographic Min-Cut (LMC).

Problem 8 (LMC)**.** *Given a connected graph $G = (\mathcal{V}, \mathcal{E})$ with a total order on $\mathcal{E}$ and two vertices $\widetilde{\tau}_1, \widetilde{\tau}_2 \in \mathcal{V}$, find the set $\widetilde{\Gamma}_{\text{LMC}} \subset \mathcal{E}$ minimal for the lexicographic order $\sqsubseteq_{lex}$, that disconnects $\widetilde{\tau}_1$ and $\widetilde{\tau}_2$ in G.*

As illustrated by Figure 4.14, we show the primal formulation of Problem 7 is equivalent to the dual formulation of Problem 8.

Proposition 7. *$\Gamma_{\min}$ is solution of Problem 7 if and if only its dual 1-chain $\widetilde{\Gamma}_{\min}$ is solution of Problem 8 on the dual graph $G(\mathcal{M})$ of $\mathcal{M}$ where $\widetilde{\tau}_1$ and $\widetilde{\tau}_2$ are respective dual vertices of τ_1 and τ_2.*

Proof. Problem 7 can be equivalently formulated as:

$$\Gamma_{\min} = \min_{\sqsubseteq_{lex}} \Big\{ \partial_d(\tau_1 + B) \mid B \in \mathbf{C}_d(\mathcal{M} \setminus \{\tau_1, \tau_2\}) \Big\} \tag{4.13}$$

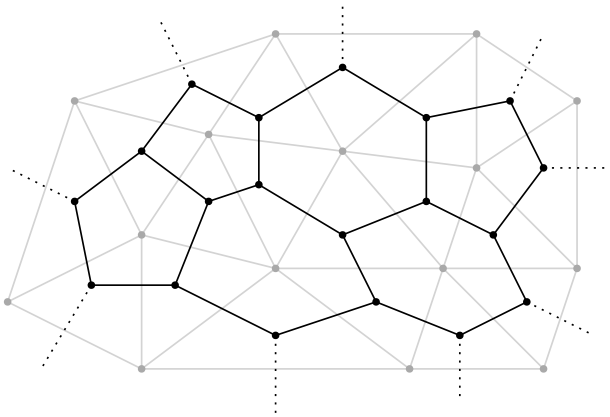
Using Observation 4.29, we see that $\Gamma_{\min}$ is the minimum in Equation (4.13) if and only if its dual 1-chain $\widetilde{\Gamma}_{\min}$ satisfies:

$$\widetilde{\Gamma}_{\min} = \min_{\sqsubseteq_{lex}} \Big\{ \partial^0(\widetilde{\tau}_1 + \widetilde{B}) \mid \widetilde{B} \subset \mathcal{V} \setminus \{\widetilde{\tau}_1, \widetilde{\tau}_2\} \Big\} \tag{4.14}$$

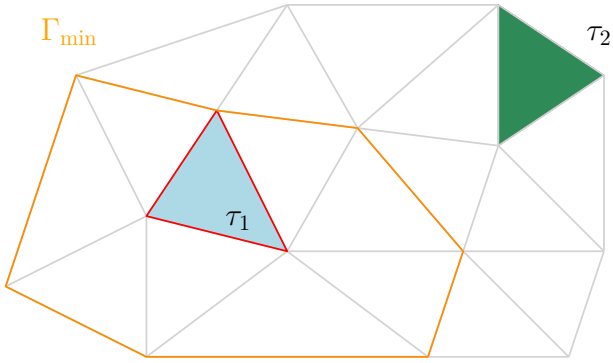
Denoting $\widetilde{\Gamma}_{\text{LMC}}$ the minimum of Problem 8, we need to show that $\widetilde{\Gamma}_{\text{LMC}} = \widetilde{\Gamma}_{\min}$.
As $\widetilde{\Gamma}_{\text{LMC}}$ disconnects $\widetilde{\tau}_1$ and $\widetilde{\tau}_2$ in $G = (\mathcal{V}, \mathcal{E})$, $\widetilde{\tau}_2$ is not in the connected component of $\widetilde{\tau}_1$ in $(\mathcal{V}, \mathcal{E} \setminus \widetilde{\Gamma}_{\text{LMC}})$. We define $\widetilde{B}$ as the connected component of $\widetilde{\tau}_1$ in $(\mathcal{V}, \mathcal{E} \setminus \widetilde{\Gamma}_{\text{LMC}})$ minus $\widetilde{\tau}_1$. We have that $\widetilde{B} \subset \mathcal{V} \setminus \{\widetilde{\tau}_1, \widetilde{\tau}_2\}$. Consider an edge $e \in \partial^0(\widetilde{\tau}_1 + \widetilde{B})$. From Observation 4.28, e connects a vertex $v_a \in \{\widetilde{\tau}_1\} \cup \widetilde{B}$ with a vertex $v_b \notin \{\widetilde{\tau}_1\} \cup \widetilde{B}$. If the edge e does not belong to Γ_{LMC}, this means vertices v_a and v_b are in the same connected component in $(\mathcal{V}, \mathcal{E} \setminus \Gamma_{\text{LMC}})$, which contradicts the statement that $v_b \notin \{\widetilde{\tau}_1\} \cup \widetilde{B}$. We have therefore shown that $\partial^0(\widetilde{\tau}_1 + \widetilde{B}) \subset \widetilde{\Gamma}_{\text{LMC}}$. Using Equation (4.14), we get:

$$\widetilde{\Gamma}_{\min} \sqsubseteq_{lex} \partial^0(\widetilde{\tau}_1 + \widetilde{B}) \sqsubseteq_{lex} \widetilde{\Gamma}_{\text{LMC}} \tag{4.15}$$

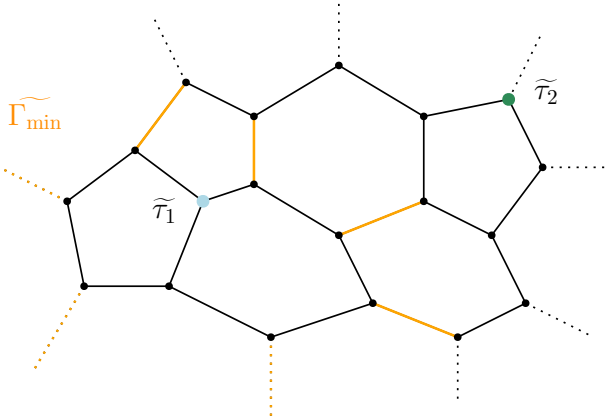
Now we claim that if there is a $\widetilde{C} \subset \mathcal{V} \setminus \{\widetilde{\tau}_1, \widetilde{\tau}_2\}$ with $\widetilde{\Gamma} = \partial^0(\widetilde{\tau}_1 + \widetilde{C})$, then $\widetilde{\Gamma}$ disconnects $\widetilde{\tau}_1$ and $\widetilde{\tau}_2$ in $(\mathcal{V}, \mathcal{E})$. Consider a path in G from $\widetilde{\tau}_1$ to $\widetilde{\tau}_2$. Let v_a be the last vertex of the path that belongs to $\{\widetilde{\tau}_1\} \cup \widetilde{C}$ and v_b the next vertex on the path (which exists since $\widetilde{\tau}_2$ is not in $\{\widetilde{\tau}_1\} \cup \widetilde{C}$). From Observation 4.28, we see that the edge (v_a, v_b) must belong to $\widetilde{\Gamma} = \partial^0(\widetilde{\tau}_1 + \widetilde{C})$. We have shown that any path in G connecting $\widetilde{\tau}_1$ and $\widetilde{\tau}_2$ has to contain an edge in $\widetilde{\Gamma}$ and the claim is proven.
In particular, the minimum $\widetilde{\Gamma}_{\min}$ disconnects $\widetilde{\tau}_1$ and $\widetilde{\tau}_2$ in $(\mathcal{V}, \mathcal{E})$. As $\widetilde{\Gamma}_{\text{LMC}}$ denotes the minimum of Problem 8, $\widetilde{\Gamma}_{\text{LMC}} \sqsubseteq_{lex} \widetilde{\Gamma}_{\min}$ which, together with Equation (4.15) and Observation 4.8, gives us $\widetilde{\Gamma}_{\text{LMC}} = \widetilde{\Gamma}_{\min}$. We have therefore shown the minimum defined by Equation (4.14) coincides with the minimum defined in Problem 8. □



(a) Duality



(b) Primal problem



(c) Dual problem

Figure 4.14: (a): A primal 2-pseudomanifold $\mathcal{M}$ in dimension 2 and its dual graph G. The dotted edges are all connected to an infinite dual vertex. (b): The primal problem finds a chain $\Gamma_{\min}$ lexicographic optimal 1-chain homologous to $\partial\tau_1$ (in red) in $\mathcal{M} \setminus \{\tau_1, \tau_2\}$. (c): The corresponding min-cut problem on the dual graph finds a lexicographic minimal set $\widetilde{\Gamma_{\min}}$ that disconnects $\widetilde{\tau_1}$ and $\widetilde{\tau_2}$ in the dual graph.

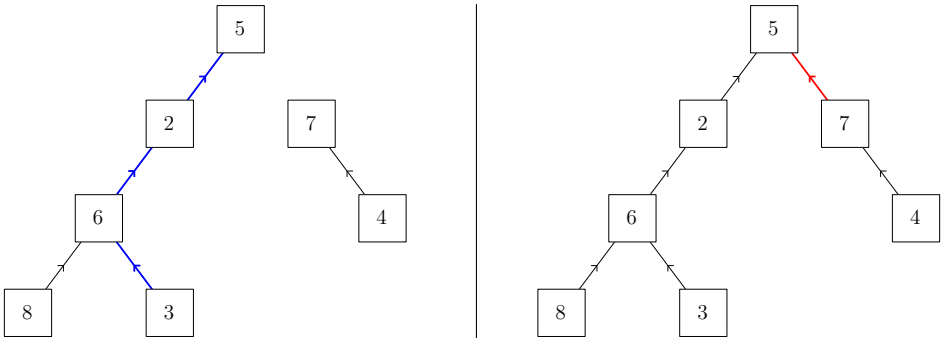


Figure 4.15: Disjoint sets represented as parent-pointer trees. On the left, the blue path shows the computation of the FindSet operation on node 3. On the right, the red edge is the result of the LinkSet operation between set representatives 5 and 7.

4.4.2 Disjoint-sets data structure

The dual formulation involves verifying connectivity on the dual graph. We recall a classic data structure called the disjoint-sets structure. Introduced for finding connected components [GF64] in a graph, it is capable of representing connected components by associating with each set a different representative value. Three operations are defined on the structure:

— MakeSet(x): given an element x, creates a set containing this element x.

— FindSet(x): given an element x, returns the representative of the set. The FindSet operation will return the same representative for any two elements in the same set and different representatives for elements in different sets.

— LinkSet(x, y): merges two sets represented by x and y. After this operation, elements of both sets have the same representative.

In practice, the structure can be implemented using parent-pointer trees. Figure 4.15 represents the structure and shows the action of FindSet and LinkSet operations. Each set is represented as a tree with the root as its representative. Each element of the set is a node of the tree containing a pointer either to an ancestor element for non-root nodes or itself for the root. The FindSet(x) operation consists of retrieving the root by iteratively asking for the ancestor starting at the node x. The LinkSet(x, y) simply updates the parent pointer of x to y. Algorithm 7 gives for reference the implementation of the different operations.

For sake of completeness, we quickly summarize the two strategies needed to make the structure asymptotically optimal [FS89]: path compression and union by rank. Path compression consists of shortening the path of ancestors at each call of the FindSet operation, for instance by setting the parent pointer of each node on the path to the found root node. Union by rank consists in keeping a value for each set representing its depth and, when using the LinkSet operation, updates the parent pointer of the shallowest tree to the deepest one so as to keep the tree heights as small as possible. Using both strategies brings the amortized complexity for FindSet and LinkSet operations down to $\mathcal{O}(\boldsymbol{\alpha}(n))$ [Tv84], where n is the number of elements in the structure and $\boldsymbol{\alpha}$ is the inverse Ackermann function.

Algorithm 7: Disjoint-sets operations

Function `MakeSet(`*Element* x`):`
| $x.parent = x$

Function `FindSet(`*Element* x`):`
| **while** $x \neq x.parent$ **do**
| | $x = x.parent$
| **end**
| return x;

Function `LinkSet(`*Element* r_1, *Element* r_2`):`
| // r_1, r_2 are set representatives
| $r_1.parent = r_2$

4.4.3 Algorithm for lexicographic min-cut

In light of the new problem equivalency, we construct an algorithmic solution to Problem 8. As we will only consider the dual graph for this section, we leave behind the dual chain notation: vertices $\widetilde{\tau}_1$ and $\widetilde{\tau}_2$ are replaced by α_1 and α_2, and the solution to the problem is simply noted Γ_{LMC}. The following lemma exposes a constructive property of the solution on subgraphs.

Lemma 4.30. *Consider Γ_{LMC} solution of Problem 8 for the graph $G = (\mathcal{V}, \mathcal{E})$ and $\alpha_1, \alpha_2 \in \mathcal{V}$. Let e_0 be an edge in $\mathcal{V} \times \mathcal{V}$ such that $e_0 < \min\{e \in \mathcal{E}\}$. Then:*

(a) The solution for $(\mathcal{V}, \mathcal{E} \cup \{e_0\})$ is either Γ_{LMC} or $\Gamma_{\text{LMC}} \cup \{e_0\}$.

(b) $\Gamma_{\text{LMC}} \cup \{e_0\}$ is solution for $(\mathcal{V}, \mathcal{E} \cup \{e_0\})$ if and only if α_1 and α_2 are connected in $(\mathcal{V}, \mathcal{E} \cup \{e_0\} \setminus \Gamma_{\text{LMC}})$.

Proof. Let's call Γ'_{LMC} the solution for $(\mathcal{V}, \mathcal{E} \cup \{e_0\})$. Since $\Gamma'_{\text{LMC}} \cap \mathcal{E}$ disconnects α_1 and α_2 in $(\mathcal{V}, \mathcal{E})$, one has

$$\Gamma_{\text{LMC}} \sqsubseteq_{lex} \Gamma'_{\text{LMC}}$$

Since $\Gamma_{\text{LMC}} \cup \{e_0\}$ disconnects α_1 and α_2 in $(\mathcal{V}, \mathcal{E} \cup \{e_0\})$, we also have

$$\Gamma'_{\text{LMC}} \sqsubseteq_{lex} \Gamma_{\text{LMC}} \cup \{e_0\}$$

Therefore,

$$\Gamma_{\text{LMC}} \sqsubseteq_{lex} \Gamma'_{\text{LMC}} \sqsubseteq_{lex} \Gamma_{\text{LMC}} \cup \{e_0\}$$

As $e_0 < \min\{e \in \mathcal{E}\}$, there is no set in $\mathcal{E} \cup \{e_0\}$ strictly between Γ_{LMC} and $\Gamma_{\text{LMC}} \cup \{e_0\}$ for the lexicographic order. It follows that Γ'_{LMC} is either equal to Γ_{LMC} or $\Gamma_{\text{LMC}} \cup \{e_0\}$.

The choice for Γ'_{LMC} is therefore ruled by the property that it should disconnect α_1 and α_2: if α_1 and α_2 are connected in $(\mathcal{V}, \mathcal{E} \cup \{e_0\} \setminus \Gamma_{\text{LMC}})$, Γ_{LMC} does not disconnect α_1 and α_2 in $(\mathcal{V}, \mathcal{E} \cup \{e_0\})$ and $\Gamma_{\text{LMC}} \cup \{e_0\}$ has to be the solution for $(\mathcal{V}, \mathcal{E} \cup \{e_0\})$. On the other hand, if α_1 and α_2 are not connected in $(\mathcal{V}, \mathcal{E} \cup \{e_0\} \setminus \Gamma_{\text{LMC}})$, then both Γ_{LMC} and $\Gamma_{\text{LMC}} \cup \{e_0\}$ disconnect α_1 and α_2 in $(\mathcal{V}, \mathcal{E} \cup \{e_0\})$, but as $\Gamma_{\text{LMC}} \sqsubset_{lex} \Gamma_{\text{LMC}} \cup \{e_0\}$, $\Gamma_{\text{LMC}} \cup \{e_0\}$ is not the solution for $(\mathcal{V}, \mathcal{E} \cup \{e_0\})$. □

We now describe Algorithm 8. The algorithm expects a set of edges sorted in decreasing order according to the lexicographic order.

Algorithm 8: Lexicographic min-cut

Inputs : $G = (\mathcal{V}, \mathcal{E})$ and $\alpha_1, \alpha_2 \in \mathcal{V}$, with $\mathcal{E} = \{e_i, i = 1, \dots, n\}$ in decreasing order
Output: Γ_{LMC}
$\Gamma_{\text{LMC}} \leftarrow \varnothing$
for $v \in \mathcal{V}$ **do**
  MakeSet(v)
end
for $e \in \mathcal{E}$ *in decreasing order* **do**
  $e = (v_1, v_2) \in \mathcal{V} \times \mathcal{V}$
  $r_1 \leftarrow$ **FindSet**(v_1), $r_2 \leftarrow$ **FindSet**(v_2)
  $c_1 \leftarrow$ **FindSet**(α_1), $c_2 \leftarrow$ **FindSet**(α_2)
  if $\{r_1, r_2\} = \{c_1, c_2\}$ **then**
    $\Gamma_{\text{LMC}} \leftarrow \Gamma_{\text{LMC}} \cup e$
  else
    UnionSet(r_1, r_2)
  end
end

Proposition 8. *Algorithm 8 computes the solution of Problem 8 for a given graph $(\mathcal{V}, \mathcal{E})$ and two vertices $\alpha_1, \alpha_2 \in \mathcal{V}$. Assuming the input set of edges $\mathcal{E}$ are sorted, the algorithm has $\mathcal{O}(n\boldsymbol{\alpha}(n))$ time complexity, where n is the cardinal of $\mathcal{E}$ and $\boldsymbol{\alpha}$ the inverse Ackermann function.*

Proof. We denote by e_i the i^{th} edge along the decreasing order and Γ^i_{LMC} the value of the variable Γ_{LMC} of the algorithm after iteration i. The algorithm works by incrementally adding edges in decreasing order and tracking the growing connected components of the set associated with α_1 and α_2 in $(\mathcal{V}, \{e \in \mathcal{E}, e \geq e_i\} \setminus \Gamma^i_{\text{LMC}})$, for $i = 1, \dots, n$.

At the beginning, no edges are inserted, and $\Gamma^0_{\text{LMC}} = \varnothing$ is indeed solution for $(\mathcal{V}, \varnothing)$. With Lemma 4.30, we are guaranteed at each iteration i to find the solution for $(\mathcal{V}, \{e \in \mathcal{E}, e \geq e_i\})$ by only adding to $\Gamma^{i-1}_{\text{LMC}}$ the current edge e_i if α_1 and α_2 are connected in $\{e \in \mathcal{E}, e \geq e_i\} \setminus \Gamma^{i-1}_{\text{LMC}}$, which is done in the if-statement. If the edge is not added, we update the connectivity of the graph $(\mathcal{V}, \{e \in \mathcal{E}, e \geq e_i\} \setminus \Gamma^i_{\text{LMC}})$ by merging the two sets represented by r_1 and r_2. After each iteration, Γ^i_{LMC} is solution for $(\mathcal{V}, \{e \in \mathcal{E}, e \geq e_i\})$ and when all edges are processed, Γ^n_{LMC} is solution for $(\mathcal{V}, \mathcal{E})$.

The complexity of the **MakeSet**, **FindSet** and **UnionSet** operations have been shown to be respectively $\mathcal{O}(1)$, $\mathcal{O}(\boldsymbol{\alpha}(v))$ and $\mathcal{O}(\boldsymbol{\alpha}(v))$, where $\boldsymbol{\alpha}(v)$ is the inverse Ackermann function on the cardinal of the vertex set [Tv84]. Assuming sorted edges as input of the algorithm – which is performed in $\mathcal{O}(n \ln(n))$, the algorithm runs in $\mathcal{O}(n\boldsymbol{\alpha}(n))$ time complexity. □

4.5 Lexicographic optimal homologous relative chain in codimension 1

This section and the next are motivated by constructing an efficient version in codimension 1 for the lexicographic optimal chain under imposed boundary problem (Problem 5). However, we will require stronger assumptions on the complex. Recall that solving Problem 5 in an arbitrary simplicial complex requires two steps:

1. finding a representative chain bounded by the given boundary (Algorithm 6),

2. finding the optimal chain having the same boundary (Algorithm 5).

Similarly, both steps are required in codimension 1. This section will solve the problem of finding the optimal relative chain homologous to a given chain $\Gamma_0 \in \mathbf{C}_{d-1}(K, B)$.

Problem 9 (Lex-OHRCP)**.** *Given a simplicial pair (K, B), and a $(d-1)$-chain $\Gamma_0 \in \mathbf{C}_{d-1}(K, B)$, find the lexicographic optimal chain $\Gamma_{\min}$ homologous to Γ_0:*

$$\Gamma_{\min} \underset{def.}{=} \min_{\sqsubseteq_{lex}} \Big\{ \Gamma_0 + b,\ b \in \mathbf{B}_{d-1}(K, B) \Big\}$$

This problem is similar to Problem 4, but formulated for relative chains in codimension 1 over a general coefficient field $\mathbb{F}$. Note that the constraints on the minimization of Problem 9 are stronger than those of Problem 6, as being homologous implies having the same boundary but the converse is not always true. These problems are however equivalent when the complex has trivial $(d-1)$-homology. The presentation of this section will be similar to Section 4.4 but requires to expose a few properties related to Poincaré duality (for absolute homology) or Lefschetz duality (for relative homology). The use of relative chains will be discussed in applications presented in Section 4.8.2 and Section 4.8.3.

4.5.1 Lefschetz duality and intersection product

For this section, we consider K a simplicial complex triangulating the d-sphere $\mathbb{S}^d$. After specifying the duality with a cell complex $\widetilde{K}$, we define an intersection product between relative k-chains in (K, B) and the $(d-k)$-chains of the dual $\widetilde{K \setminus B}$ and expose a few property of the induced Lefschetz duality. The chain coefficient group is supposed to be a general field $\mathbb{F}$ in this section.

Duality in elementary linear algebra

We start by giving a simple duality result of linear algebra. Consider E, F finite dimensional vector spaces and a linear map $L : E \to F$.
Denote by E^*, F^* the respective duals of E and F, and, for $a \in E, b \in E^*$ and the image $b(a)$ of a by b by:

$$b(a) = \langle b,\ a \rangle \in \mathbb{F}.$$

The orthogonal complement [2] of a subspace A of E, denoted $A^\perp$, can be defined as the subspace of E^* verifying:

$$A^\perp \underset{\text{def.}}{=} \Big\{ b \in E^* \mid \forall a \in A, \langle b,\ a \rangle = 0 \Big\}$$

The adjoint of L, is the linear map $L^* : F^* \to E^*$ defined by:

$$\forall a \in E, \forall b \in F^*,\ \langle L^* b,\ a \rangle \underset{\text{def.}}{=} \langle b,\ La \rangle$$

The following equivalences

$$\begin{aligned} a \in \operatorname{Ker} L &\iff La = 0 \\ &\iff \forall b \in F^*,\ \langle b,\ La \rangle = 0 \\ &\iff \forall b \in F^*,\ \langle L^* b,\ a \rangle = 0 \\ &\iff \forall c \in \operatorname{Im} L^*,\ \langle c,\ a \rangle = 0 \iff a \in \Big(\operatorname{Im} L^* \Big)^\perp \end{aligned}$$

give the following lemma:

[2] This is actually the annihilator, which, for finite vector space, is the dual of the orthogonal complement.

Lemma 4.31. *If E, F are finite dimensional vector spaces and $L : E \to F$ a linear map, then:*

$$\operatorname{Ker} L = \Big(\operatorname{Im} L^*\Big)^\perp$$
$$\Big(\operatorname{Ker} L\Big)^\perp = \operatorname{Im} L^*$$

Relative duality between (K, B) and $\widetilde{K \setminus B}$

Any triangulated manifold K admits a dual polyhedral decomposition. We denote $\widetilde{K}$ the cell complex dual to K, where k-cells are in bijective correspondence with the $(d-k)$-simplices of K. A prime example of this duality is Voronoi diagrams, dual to Delaunay triangulations.

For $0 \leq k \leq d$, denote by $K^{(k)}$ and $\widetilde{K}^{(k)}$ the set of k-simplices of K and the set of k-cells of $\widetilde{K}$. For $\sigma, \tau \in K$ with $\dim(\sigma) = \dim(\tau) - 1$ one write $\sigma \preceq_s \tau$, where $s \in \{-1, 1\}$, when σ is a face of τ and $s \in \{-1, 1\}$ is the sign of σ as it appears in $\partial\tau$ (the sign s can be ignored if $\mathbb{F} = \mathbb{Z}_2$). The duality between K and $\widetilde{K}$ means precisely that:

— for $0 \leq k \leq d$, there is a bijection between $K^{(k)}$ and $\widetilde{K}^{(d-k)}$, where the cell dual of the k-simplex $\sigma \in K^{(k)}$ is denoted $\tilde{\sigma} \in \widetilde{K}^{(d-k)}$,

— for $\sigma, \tau \in K$ such that $\dim(\sigma) = \dim(\tau) - 1$ then:

$$\sigma \preceq_s \tau \iff \tilde{\tau} \preceq_s \tilde{\sigma} \tag{4.16}$$

Equation (4.16) can be equivalently formulated as saying the matrices of the respective boundary operators $\partial_{\mathbf{C}_{k+1}(K)} : \mathbf{C}_{k+1}(K) \to \mathbf{C}_k(K)$ and $\partial_{\mathbf{C}_{d-k}(\widetilde{K})} : \mathbf{C}_{d-k}(\widetilde{K}) \to \mathbf{C}_{d-k-1}(\widetilde{K})$, expressed in the simplices, or cells, bases, are transpose of each other:

$$\partial_{\mathbf{C}_{d-k}(\widetilde{K})} = \big(\partial_{\mathbf{C}_{k+1}(K)}\big)^t \tag{4.17}$$

Intersection product

Denote by *intersection product* the bilinear form defined by:

$$\begin{aligned} \otimes : \mathbf{C}_k(K) \times \mathbf{C}_{d-k}(\widetilde{K}) &\to \mathbb{F} \\ (\Gamma, \gamma) &\mapsto \Gamma \otimes \gamma \underset{\text{def.}}{=} \sum_{\sigma \in K^{(k)}} \Gamma(\sigma)\, \gamma(\widetilde{\sigma}) \end{aligned} \tag{4.18}$$

Observation 4.32. *The intersection product vanishes when Γ and γ have disjoint support. Extending the $\widetilde{\cdot}$ notation to sets of simplices:*

$$\forall \Gamma \in \mathbf{C}_k(K), \forall \gamma \in \mathbf{C}_{d-k}(\widetilde{K}), \quad |\Gamma| \cap \widetilde{|\gamma|} = \emptyset \implies \Gamma \otimes \gamma = 0$$

If we consider $\Gamma \in \mathbf{C}_k(K)$ and $\gamma \in \mathbf{C}_{d-k}(K)$ as vectors expressed in the simplex basis, the intersection product can be written $\Gamma \otimes \gamma = \Gamma^t \gamma$, where Γ^t is the transpose of vector Γ. Using Equation (4.17) we get that:

$$\partial\Gamma \otimes \gamma = \big(\partial_{\mathbf{C}_{k+1}(K)}\Gamma\big)^t \gamma = \Gamma^t \big(\partial_{\mathbf{C}_{k+1}(K)}\big)^t \gamma = \Gamma^t \partial_{\mathbf{C}_{d-k}(\widetilde{K})} \gamma$$

So that:

$$\partial\Gamma \otimes \gamma = \Gamma \otimes \partial\gamma \tag{4.19}$$

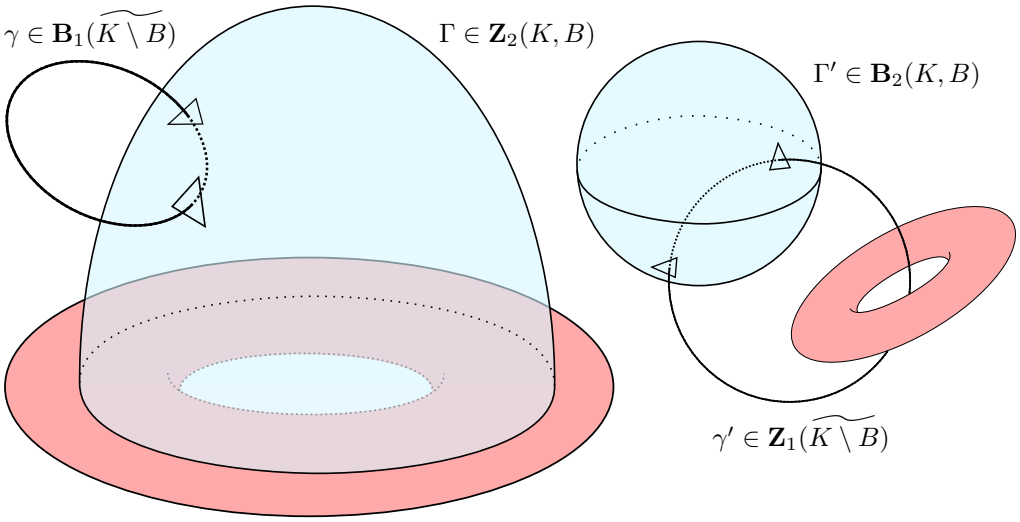


Figure 4.16: Illustration of the intersection product and Lemma 4.34 in dimension 3. The pink region corresponds to subcomplex B of the ambient complex K. The intersection product between $\Gamma \in \mathbf{Z}_2(K, B)$ and any $\gamma \in B_1(\widetilde{K \setminus B})$ verifies $\Gamma \otimes \gamma = 0$. The intersection product between $\Gamma' \in \mathbf{B}_2(K, B)$ and any $\gamma' \in \mathbf{Z}_1(\widetilde{K \setminus B})$ also verifies $\Gamma' \otimes \gamma' = 0$.

While $K \setminus B$ is not a simplicial complex in general, because some simplices in $K \setminus B$ may have faces in B, by contrast, the set of its dual cells, $\widetilde{K \setminus B}$, is a cellular complex. Observes that the set of k-simplices in $K \setminus B$, in bijection with their dual $(d-k)$-cells in $\widetilde{K \setminus B}$, defines a canonical basis for both $\mathbf{C}_k(K, B)$ and $\mathbf{C}_{d-k}(\widetilde{K \setminus B})$ in which the intersection product $\otimes$ corresponds to the canonical dot product between respective coordinates in $\mathbb{F}^n$, where n is the cardinal of $(\widetilde{K \setminus B})^{(d-k)}$. Identifying a chain $\Gamma \in \mathbf{C}_{d-1}(K, B)$ with the dual element $\Gamma^* : \gamma \mapsto \Gamma \otimes \gamma$, the notation $\cdot^\perp$ used in Lemma 4.31 becomes:

Definition 4.33. *For a vector subspace V of $C_1(\widetilde{K \setminus B})$,*

$$V^\perp \underset{def.}{=} \left\{ \Gamma \in \mathbf{C}_{d-1}(K, B) \mid \forall \gamma \in V,\ \Gamma \otimes \gamma = 0 \right\} \tag{4.20}$$

As illustrated in dimension 3 in Figure 4.16, applying Lemma 4.31 in the context of Equation (4.19) gives the following properties:

Lemma 4.34. *Let $K \simeq \mathbb{S}^d$ and B a subcomplex of K.*

$$\mathbf{Z}_{d-1}(K, B) = \mathbf{B}_1(\widetilde{K \setminus B})^\perp \tag{4.21}$$

$$\mathbf{B}_{d-1}(K, B) = \mathbf{Z}_1(\widetilde{K \setminus B})^\perp \tag{4.22}$$

In other words, for any $\Gamma \in \mathbf{C}_{d-1}(K, B)$, one has:

$$\Gamma \in \mathbf{Z}_{d-1}(K, B) \iff \left(\forall \gamma \in \mathbf{B}_1(\widetilde{K \setminus B}), \quad \Gamma \otimes \gamma = 0 \right) \tag{4.23}$$

$$\Gamma \in \mathbf{B}_{d-1}(K, B) \iff \left(\forall \gamma \in \mathbf{Z}_1(\widetilde{K \setminus B}), \quad \Gamma \otimes \gamma = 0 \right) \tag{4.24}$$

Lefschetz duality

The intersection product of previous section extends to homology classes. For two homologous relative $(d-1)$-cycles $\Gamma, \Gamma' \in \mathbf{Z}_{d-1}(K, B)$, the fact that $\Gamma' - \Gamma$ belongs to $\mathbf{B}_{d-1}(K, B)$ implies, thanks to Equation (4.24):

$$\forall \gamma \in \mathbf{Z}_1(\widetilde{K \setminus B}), \Gamma' \otimes \gamma = \Gamma \otimes \gamma$$

Similarly, for two homologous 1-cycles $\gamma, \gamma' \in \mathbf{Z}_1(\widetilde{K \setminus B})$, thanks to Equation (4.23):

$$\forall \Gamma \in \mathbf{Z}_{d-1}(K, B)\, \Gamma \otimes \gamma' = \Gamma \otimes \gamma$$

The intersection product is therefore independent of the chosen representative of the homology class and extends to a bilinear form on homology groups:

$$\otimes : \mathcal{H}_{d-1}(K, B) \times \mathcal{H}_1(\widetilde{K \setminus B}) \to \mathbb{F}$$

Observation 4.35. *Thanks to Lemma 4.34, the pairing*

$$\otimes : \mathcal{H}_{d-1}(K, B) \times \mathcal{H}_1(\widetilde{K \setminus B}) \to \mathbb{F}$$

is perfect, which means it induces an isomorphism between $\mathcal{H}_1(\widetilde{K \setminus B})$ and the dual of $\mathcal{H}_{d-1}(K, B)$. This can be seen as a particular case of Lefschetz duality.

4.5.2 Dual problem formulation

We now give a dual formulation for Problem 9. We consider $G_{K\setminus B} = (\mathcal{V}_{K\setminus B}, \mathcal{E}_{K\setminus B})$ the 1-skeleton of the complex $\widetilde{K \setminus B}$, where $\mathcal{V}_{K\setminus B}$ and $\mathcal{E}_{K\setminus B}$ are respectively the set of vertices and edges of the graph. We consider the following graph problem:

Problem 10. *Given a subgraph G of $G_{\widetilde{K\setminus B}}$, with a total order on its edges, and $\Gamma_0 \in \mathbf{C}_{d-1}(K, B)$, consider the set:*

$$\triangle_G \underset{def.}{=} \Big\{ \Gamma \in \mathbf{C}_{d-1}(K, B) \;\mid\; \forall \gamma \in \mathbf{Z}_1(G),\ (\Gamma - \Gamma_0) \otimes \gamma = 0 \Big\}$$

and find the chain defined by:

$$\Gamma_{\min} \underset{def.}{=} \min_{\sqsubseteq_{lex}} \triangle_G \tag{4.25}$$

From Lemma 4.34, Equation (4.24), we have the following equivalence:

Lemma 4.36. *Consider $\Gamma_0 \in \mathbf{C}_{d-1}(K, B)$ and $(K \simeq \mathbb{S}^d, B)$ a simplicial pair. Problem 9 for the simplicial pair (K, B) and Γ_0 is equivalent to Problem 10 for the graph $G_{K\setminus B}$ and Γ_0.*

Denoting by n the number of edges of the graph $G_{K\setminus B}$, we enumerate the edges of $\mathcal{E}_{K\setminus B}$ in a decreasing order induced by the total order on the primal $(d-1)$-simplices of $K \setminus B$:

$$e_1 > e_2 > \cdots > e_n \tag{4.26}$$

We consider the increasing sequence of graphs:

$$G_i \underset{\text{def.}}{=} \Big(\mathcal{V}_{K\setminus B},\, \mathcal{E}_i \underset{\text{def.}}{=} \{ e_j \in \mathcal{E}_{K\setminus B} \mid j \leq i \} \Big) \tag{4.27}$$

In particular, $G_0 = (\mathcal{V}_{K\setminus B}, \varnothing)$ and $G_n = G_{K\setminus B}$.

The following observation gives an important property on the support of the solution of Problem 10 for this increasing sequence of graphs.

Observation 4.37. *Denote by $\triangle_{G_i}$ the set of Problem 10 associated to some graph G_i defined in Equation* (4.27)*. Recall from Observation 4.32 that, for $\Gamma \in \mathbf{C}_{d-1}(K,B)$ and $\gamma \in \mathbf{Z}_1(G_i)$,*

$$|\Gamma - \Gamma_0| \cap \widetilde{|\gamma|} = \emptyset \implies (\Gamma - \Gamma_0) \otimes \gamma = 0$$

The constraints defining the set $\triangle_{G_i}$ rely only on the value of Γ on the $(d-1)$-simplices dual to 1-simplices in $\mathcal{E}_i$. The optimal chain on $\triangle_{G_i}$ then verifies

$$\left| \min_{\sqsubseteq_{lex}} \triangle_{G_i} \right| \subset \widetilde{\mathcal{E}_i}$$

We show that the minimum of Problem 10 for $G_{K\setminus B}$ can be constructed incrementally by considering this increasing sequence of graphs. To this end, we borrow terminology from persistent homology [EH10] and qualify edges as *positive* if their addition creates new cycles in the graph and as *negative* if their addition merges two connected components.

Lemma 4.38. *Consider $\Gamma_0 \in \mathbf{C}_{d-1}(K,B)$. Denote by $\Gamma^{(i)}$ and $\Gamma^{(i+1)}$ the respective solutions of Problem 10 for the graphs G_i and G_{i+1} and the chain Γ_0. Denote by e_{i+1} the edge added to G_i to form G_{i+1}.*

- *If the edge e_{i+1} is negative, i.e. kills a connected component in G_{i+1} and* $\dim \mathbf{Z}_1(G_{i+1}) = \dim \mathbf{Z}_1(G_i)$,
$$\Gamma^{(i+1)} = \Gamma^{(i)}$$
- *If the edge e_{i+1} is positive, i.e. forms a new cycle in G_{i+1} and* $\dim \mathbf{Z}_1(G_{i+1}) = \dim \mathbf{Z}_1(G_i) + 1$*, for $\gamma_{i+1} \in \mathbf{Z}_1(G_{i+1}) \setminus \mathbf{Z}_1(G_i)$, with $\gamma_{i+1}(e_{i+1}) = 1$,*
$$\Gamma^{(i+1)} = \Gamma^{(i)} + \alpha\, \widetilde{e_{i+1}}$$
with $\alpha = -(\Gamma^{(i)} - \Gamma_0) \otimes \gamma_{i+1}$.

Proof. A negative edge will kill a connected component but no new cycles will be formed by adding this edge and $\mathbf{Z}_1(G_{i+1}) = \mathbf{Z}_1(G_i)$. We have immediately that $\Gamma^{(i)}$ is solution for G_{i+1}.

For a positive edge e_{i+1}, a new graph cycle is formed in G_{i+1}. In term of chains, this means $\dim \mathbf{Z}_1(G_{i+1}) = \dim \mathbf{Z}_1(G_i) + 1$ and we can consider a 1-chain γ_{i+1} in $\mathbf{Z}_1(G_{i+1}) \setminus \mathbf{Z}_1(G_i)$ verifying:

$$\gamma_{i+1}(e_{i+1}) = 1 \tag{4.28}$$

Let's verify that $\Gamma^{(i)} + \alpha\widetilde{e_{i+1}} \in \triangle_{G_{i+1}}$, with $\alpha = -(\Gamma^{(i)} - \Gamma_0) \otimes \gamma_{i+1}$. Consider any cycle γ in $\mathbf{Z}_1(G_{i+1})$ and $\mu = \gamma(e_{i+1})$ its coefficient for edge e_{i+1}. The cycle γ can be written as $\gamma' + \mu\gamma_{i+1}$, where $\gamma' = \gamma - \mu\gamma_{i+1}$ is in $\mathbf{Z}_1(G_i)$. The intersection product along this cycle for $\Gamma^{(i)} + \alpha\widetilde{e_{i+1}}$ and Γ_0 can now be decomposed as:

$$(\Gamma^{(i)} + \alpha\widetilde{e_{i+1}} - \Gamma_0) \otimes \gamma = (\Gamma^{(i)} - \Gamma_0) \otimes \gamma' + \alpha\widetilde{e_{i+1}} \otimes \gamma' + \mu(\Gamma^{(i)} + \alpha\widetilde{e_{i+1}} - \Gamma_0) \otimes \gamma_{i+1}$$

By definition of $\Gamma^{(i)}$ as solution of Problem 10 for G_i, $(\Gamma^{(i)} - \Gamma_0) \otimes \gamma' = 0$. As $e_{i+1} \notin |\gamma'|$, we also have $\widetilde{e_{i+1}} \otimes \gamma' = 0$. Finally, from the definition of α and Equation (4.28), the last term of the sum is also zero:

$$(\Gamma^{(i)} + \alpha\widetilde{e_{i+1}} - \Gamma_0) \otimes \gamma_{i+1} = (\Gamma^{(i)} - \Gamma_0) \otimes \gamma_{i+1} + \alpha\, \widetilde{e_{i+1}} \otimes \gamma_{i+1} = 0$$

and we have shown that $\Gamma^{(i)} + \alpha\widetilde{e_{i+1}} \in \triangle_{G_{i+1}}$. As $\Gamma^{(i+1)} = \min_{\sqsubseteq_{lex}} \triangle_{G_{i+1}}$, $\Gamma^{(i+1)} \sqsubseteq_{lex} \Gamma^{(i)} + \alpha\widetilde{e_{i+1}}$. As $\triangle_{G_{i+1}} \subseteq \triangle_{G_i}$, we also have that $\Gamma^{(i)} \sqsubseteq_{lex} \Gamma^{(i+1)}$. Hence the following bounds for $\Gamma^{(i+1)}$:

$$\Gamma^{(i)} \sqsubseteq_{lex} \Gamma^{(i+1)} \sqsubseteq_{lex} \Gamma^{(i)} + \alpha\widetilde{e_{i+1}} \tag{4.29}$$

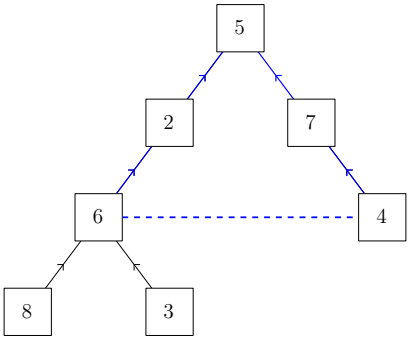


Figure 4.17: A set of oriented edges obtained by performing FindSet operations for the two nodes 6 and 4 of the same set. The dotted edge, which is not in the disjoint-sets, illustrates how a cycle can be constructed from the structure.

Recall that Observation 4.37 showed

$$\left|\Gamma^{(i+1)}\right| \subset \widetilde{\mathcal{E}_{i+1}}$$

which means, together with Equation (4.29), that the supports of $\Gamma^{(i)}$ and $\Gamma^{(i+1)}$ can only differ on $\widetilde{e_{i+1}}$. When $\alpha = -(\Gamma^{(i)} - \Gamma_0) \otimes \gamma_{i+1}$ is not zero, $\Gamma^{(i)} \notin \triangle_{G_{i+1}}$. Therefore, the support of $\Gamma^{(i+1)}$ needs to contain $\widetilde{e_{i+1}}$ and the unicity of the solution from Lemma 4.11 implies $\Gamma^{(i+1)} = \Gamma^{(i)} + \alpha\widetilde{e_{i+1}}$. When $\alpha = -(\Gamma^{(i)} - \Gamma_0) \otimes \gamma_{i+1}$ is zero, we can of course also write $\Gamma^{(i+1)} = \Gamma^{(i)} + \alpha\widetilde{e_{i+1}}$. □

As illustrated in Lemma 4.38, two key elements are required to incrementally construct the solution for Problem 10: tracking connected components in the sequence of graphs $(G_i)_{i=0\dots n}$ to distinguish between positive and negative edges and computing the coefficient α for any new cycle going through a positive edge. The former requires again the disjoint-sets data structure to keep track of incremental connected components. In the next section, we will show how a small augmentation of this data structure can help to efficiently compute the latter.

4.5.3 Augmented disjoint-sets data structure

As described in Lemma 4.38, the incremental construction of the lexicographic optimal homologous relative chain requires, for a *positive* edge e_{i+1}, to compute the coefficient of the intersection product $(\Gamma^{(i)} - \Gamma_0) \otimes \gamma_{i+1}$ of the previous solution $\Gamma^{(i)}$ minus the initial given chain Γ_0 with a cycle γ_{i+1} passing through this new edge. As illustrated in Figure 4.17, the tree structure of the disjoint-set defines such a cycle, going through the representative element of the set.

We now describe an augmented version of the disjoint-set structure enabling the computation of intersection products and give the corresponding MakeSet*, FindSet* and LinkSet* operations in Algorithm 9. A tree node now consists of a parent pointer to an ancestor and a value in $\mathbb{F}$, representing the intersection product along a directed path in the disjoint set from this node to the parent (respectively denoted by *parent* and *value* in functions of Algorithm 9).

The reader can verify that the two strategies making the standard disjoint-sets asymptotically optimal, described in Section 4.4.2, can still be used in this augmented version. During path compression, the *value* associated to a node needs to be updated by summing all coefficients on the compressed path starting from this node. During union by rank, the given value in the LinkSet* operation has to be set to its opposite if

Algorithm 9: Augmented disjoint-sets operations

Function `MakeSet*`(*Element* x)**:**
  $x.parent = x$
  $x.value = 0_{\mathbb{F}}$

Function `FindSet*`(*Element* x)**:**
  $g = 0_{\mathbb{F}}$
  while $x \neq x.parent$ **do**
    $g = g + x.value$
    $x = x.parent$
  end
  return (x, g);

Function `LinkSet*`(*Element* r_1, *Element* r_2, $\mathbb{F}$ δ)**:**
  // r_1, r_2 are set representatives
  $r_1.parent = r_2$
  $r_1.value = \delta$

the largest set corresponds to the first given representative. The structure makes exactly the same number of addition as the tree height when calling the FindSet* operation. The complexity of its augmented version is therefore $c\boldsymbol{\alpha}(n)$, where n is the number of sets, $\boldsymbol{\alpha}$ is the inverse Ackermann function and c denotes the cost of an addition in $\mathbb{F}$.

While, in this context, stored values are in a field $\mathbb{F}$, the augmentation can be generalized to values in a non-commutative group: it is conceivable that this data-structure might have other applications.

4.5.4 Lexicographic optimal homologous chain

In this section, from the dual formulation presented in Section 4.5.2 and with the help of the augmented disjoint-set structure described in Section 4.5.3, we present an algorithmic solution for Problem 9.

Lemma 4.39. *Taking as inputs $\Gamma_0 \in \mathbf{C}_{d-1}(K, B)$ and the graph $G_{K\setminus B}$ with edges sorted in decreasing order along the total order defined on $(d-1)$-simplices, Algorithm 10 solves Problem 9 in $\mathcal{O}(cn\boldsymbol{\alpha}(n))$ time complexity, where n is the number of edges in the graph $G_{K\setminus B}$, $\boldsymbol{\alpha}$ the inverse Ackermann function and c the cost of addition in $\mathbb{F}$.*

Proof. Note first that all edges added to the disjoint-sets structure are negative and therefore, from Lemma 4.38, $\Gamma_{\min}(\widetilde{e}) = 0$ for any edge e in the disjoint-sets structure. More generally,

$$\Gamma_{\min} \otimes \gamma = 0 \tag{4.30}$$

for any path γ on edges in the disjoint-sets structure.

Algorithm 10 is an iterative application of Lemma 4.38. Every edge of the graph is considered in decreasing order along the total order on 2-simplices, which corresponds to the graph filtration $(G_i)_{i=0,\dots,n}$. We will show by induction that, at each iteration i,

- $\Gamma_{\min}$ is the solution of Problem 10 for G_i,
- the augmentation part of the disjoint-sets structure verifies, for any negative edge $e = (v_1, v_2) \in \mathcal{E}_i$:

$$\Gamma_0 \otimes e = g_2 - g_1$$

Algorithm 10: Lexicographic optimal homologous relative cycle

Inputs : $G = (\mathcal{V}_{K\setminus B}, \mathcal{E}_{K\setminus B})$ with $\mathcal{E}_{K\setminus B} = \{e_i, i = 1, \dots, n\}$ in decreasing order, $\Gamma_0 \in \mathbf{C}_{d-1}(K, B)$.

Output: $\Gamma_{\min} \in \mathbf{C}_{d-1}(K, B)$, lexicographic optimal relative chain homologous to Γ_0.

```
Γ_min ← 0
for v ∈ V_{K\B} do
 | MakeSet*(v)
end
for e ∈ E_{K\B} in decreasing order do
 | e = (v_1, v_2) ∈ V_{K\B} × V_{K\B}
 | (r_1, g_1) ← FindSet*(v_1)
 | (r_2, g_2) ← FindSet*(v_2)
 | α = Γ_0(ẽ) + g_2 − g_1
 | if r_1 = r_2 then
 |  | Γ_min ← Γ_min + α · ẽ
 | else
 |  | LinkSet*(r_1, r_2, α)
end
```

where g_1, g_2 are the respective values returned by the FindSet* operations on v_1 and v_2.

Sets are initially first made inside the disjoint set structure for each vertex in $\mathcal{V}_{K\setminus B}$ using the MakeSet* operation. We start with $\Gamma_{\min} = 0$ as it is the solution of Problem 10 for G_0. As $\mathcal{E}_0 = \varnothing$, the invariant on the augmentation part of the disjoint-sets structure is also verified.

At an iteration i, we consider the edge e_{i+1} and suppose $\Gamma_{\min}$ is the solution of Problem 10 for graph G_i. We also assume the invariant on the augmented disjoint-sets to be verified. Following Lemma 4.38, we first determine whether the edge e_{i+1} is positive or negative by testing the set representatives of both edge vertices in G_i. When the edge is negative, i.e. $r_1 \neq r_2$, Lemma 4.38 shows $\Gamma_{\min}$ is still the solution of Problem 10 for graph G_{i+1}. The LinkSet* operation is used to update the graph connectivity when adding the edge e_{i+1}. It also updates the augmented part so as to verify the invariant. The data structure after update is illustrated in Figure 4.18. Indeed, denote by g_1 and g_1' (resp. g_2 and g_2') the values obtained by calling the FindSet* operation on v_1 (resp. v_2) before and after the update of the data structure (i.e. before and after having set α as coefficient for r_1). We now have:

$$g_2' - g_1' = g_2 - (g_1 + \alpha) = g_2 - g_1 - (\Gamma_0(\widetilde{e_{i+1}}) + g_2 - g_1) = \Gamma_0 \otimes e_{i+1}$$

If the edge is positive, i.e. $r_1 = r_2$, the value α of Lemma 4.38 needs to be computed thanks to the path $\gamma_{i+1} = e_{i+1} + p_2 - p_1$, where p_1 and p_2 are respectively paths from v_1 to r_1 and v_2 to $r_2 = r_1$ inside the disjoint-sets structure. Note that the condition $\gamma_{i+1}(e_{i+1}) = 1$ is verified.

$$\alpha = -(\Gamma_{\min} - \Gamma_0) \otimes \gamma_{i+1} \tag{4.31}$$

From Equation (4.30) and as $\widetilde{e_{i+1}} \notin |\Gamma_{\min}|$:

$$\alpha = \Gamma_0 \otimes \gamma_{i+1} = \Gamma_0(\widetilde{e_{i+1}}) + \Gamma_0 \otimes p_2 - \Gamma_0 \otimes p_1$$

From the invariant on the augmentation part of disjoint-sets structure at iteration i means the values g_1 and g_2 obtained by calling FindSet* on v_1 and v_2 verify:

$$g_1 = \Gamma_0 \otimes p_1 \text{ and } g_2 = \Gamma_0 \otimes p_2$$

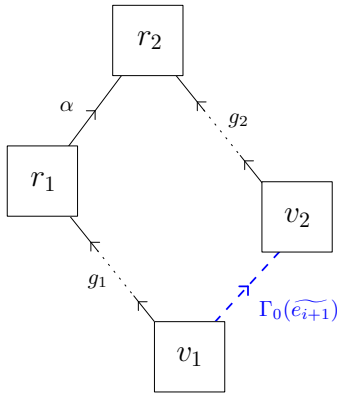


Figure 4.18: Illustration of the disjoint-sets augmented data structure after the addition of a positive edge e_{i+1} between v_1 and v_2. The value g_1 (resp. g_2) corresponds to the sum of the values along the path from the node v_1 (resp. v_2) to the node r_1 (resp. r_2). The blue dashed edge is not present in the disjoint-sets structure but illustrates the value that needs to be stored in the disjoint-sets structure.

and therefore $\alpha = \Gamma_0(\widetilde{e_{i+1}}) + g_2 - g_1$. Chain $\Gamma_{\min}$ is updated in consequence to become the minimum of Problem 10 for graph G_{i+1}. The output of the algorithm is therefore solution of Problem 10 for the whole graph G_n.

Iterating over all edges of the graph while using disjoint-sets operations leads to a $\mathcal{O}(cn\boldsymbol{\alpha}(n))$ complexity. □

4.6 Representative chain in the Delaunay 3-complex

As detailed in the introduction of Section 4.5, computing a lexicographic optimal 2-chain bounded by a given 1-boundary requires first to compute a "representative" 2-chain bounded by the given boundary. In this section, we describe Algorithm 11 that does exactly that in the Delaunay 3-complex. For simplicity, the algorithm assumes the chain coefficient is $\mathbb{F} = \mathbb{Z}_2$, but the approach extends similarly to an arbitrary coefficient field. We again consider chains as sets of simplices.

Observation 4.40. *Contrary to Algorithm 6, the* 1*-homology of the Delaunay* 3*-complex is trivial, therefore all cycles are boundaries, which implies the existence of a representative chain bounding any cycle in the Delaunay complex.*

4.6.1 Lower link of a vertex in the 3D Delaunay complex

In all that follows, $\mathbf{P}$ denotes a set of points in $\mathbb{R}^3$ verifying the following generic condition. In practice, non-generic configurations can be solved by SoS (Simulation of Simplicity) [EM90].

Condition 4. *No pair of vertices in* $\mathbf{P}$ *have same* z*-coordinate.*

We recall the notion of link and lower link.

Definition 4.41 (Link of a simplex)**.** *The link* $\mathrm{lk}_K(\tau)$ *of a simplex* τ *in a simplicial complex* K *is the simplicial complex made of all simplices* $\sigma \in K$ *such that* $\tau \cup \sigma \in K$ *and* $\tau \cap \sigma = \emptyset$*.*

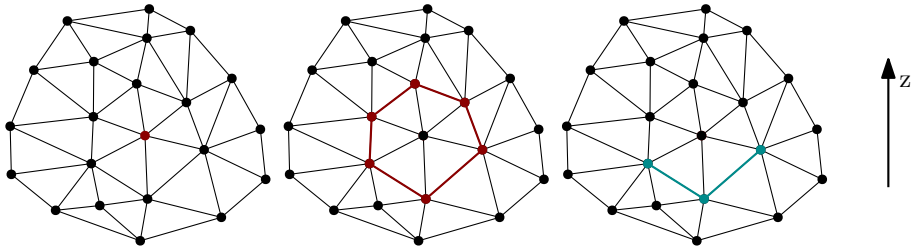


Figure 4.19: A vertex in a two-dimensional Delaunay complex (left), its link (center) and lower link (right).

Definition 4.42 (Lower link of a vertex)**.** *The lower link* $\mathrm{llk}_K(a)$ *of a vertex* $a \in \mathbf{P}$ *is the simplicial complex made of all simplices in the link* $\mathrm{lk}_K(a)$ *of* a *whose vertices have all their* z*-coordinates smaller than the* z*-coordinate of* a*.*

Figure 4.19 illustrates the definitions of link and lower link in Delaunay 2-complex. The lower link in the Delaunay 3-complex is also illustrated in Figure 4.20.

Denote by $\mathcal{D}el(\mathbf{P})$ the Delaunay 3-complex of the set of points $\mathbf{P}$. The link $\mathrm{lk}_{\mathcal{D}el(\mathbf{P})}(a)$ and lower link $\mathrm{llk}_{\mathcal{D}el(\mathbf{P})}(a)$ of vertex a are two-dimensional simplicial complexes: each tetrahedron of $\mathcal{D}el(\mathbf{P})$ containing a gives rise to a triangle in $\mathrm{lk}_{\mathcal{D}el(\mathbf{P})}(a)$, each triangle of $\mathcal{D}el(\mathbf{P})$ containing a gives rise to an edge in $\mathrm{lk}_{\mathcal{D}el(\mathbf{P})}(a)$ and each edge of $\mathcal{D}el(\mathbf{P})$ containing a gives rise to a vertex in $\mathrm{lk}_{\mathcal{D}el(\mathbf{P})}(a)$. These triangles or edges belong to $\mathrm{llk}_{\mathcal{D}el(\mathbf{P})}(a)$ if and only if all their vertices have z-coordinates smaller than the z-coordinate a_z of a (see Figure 4.20, left).

Recall that a topological space and in particular a simplicial complex is said **contractible** if it has the homotopy type of a point. In particular, a contractible simplicial complex K is connected, which means that any pair of vertices in K can be connected by a path of edges in K.

Algorithm 11 relies on the fact that, in a Delaunay complex, the lower link of a vertex is either empty or contractible.

Lemma 4.43. *Let* $\mathcal{D}el(\mathbf{P})$ *be the Delaunay complex of a set of vertices* $\mathbf{P}$ *in* $\mathbb{R}^3$ *verifying Condition 4 and* $a \in \mathbf{P}$ *a vertex of the complex.*

- *the lower link* $\mathrm{llk}_{\mathcal{D}el(\mathbf{P})}(a)$ *is empty if and only if vertex* a *has minimal* z*-coordinate in* $\mathbf{P}$*,*
- *if the* z*-coordinate of* a *is not minimal in* $\mathbf{P}$*,* $\mathrm{llk}_{\mathcal{D}el(\mathbf{P})}(a)$ *is contractible.*

Proof. Denote by a_z the z-coordinate of a. By definition of the Delaunay triangulation and of $\mathrm{lk}_{\mathcal{D}el(\mathbf{P})}(a)$, there is a one-to-one correspondence between the vertices in $\mathrm{lk}_{\mathcal{D}el(\mathbf{P})}(a)$ and the set of (possibly unbounded) facets contributing to the boundary of the Voronoi cell of a. By definition, a vertex v with z-coordinate v_z is in the lower link of a if and only if its Voronoi cell has a common boundary with the Voronoi cell of a and $v_z < a_z$. It follows that the Voronoi cell of a is contained in the half space containing a and bounded by the plane bisector of a and v. Such a vertex exists if and only if the vertical half-line starting at a and pointing toward negative z is not contained in the Voronoi cell of a. One can see that the vertical half-line starting at a and pointing toward negative z is contained in the Voronoi cell of a if and only if a has minimal z in V and the first statement is proven.

The lower envelope of the Voronoi cell of a is the union of the facets dual to edges connecting a and a vertex in the lower link of a. When this lower envelope is not empty, its projection on the horizontal plane is an homeomorphism with a convex two-dimensional polytope. This lower envelope is therefore contractible. Since the lower link

of a is the nerve of the set of facets in the lower envelope of the Voronoi cell, the second statement follows from the nerve theorem. □

4.6.2 Algorithmic description

Algorithm 11: Finding a set of triangles for a given boundary

Inputs : $\mathcal{D}el(\mathbf{P})$ a Delaunay complex and a 1-cycle $A_0 \in \mathbf{Z}_1(\mathcal{D}el(\mathbf{P}); \mathbb{Z}_2)$
Output: A 2-chain $\Gamma_0 \in \mathbf{C}_2(\mathcal{D}el(\mathbf{P}); \mathbb{Z}_2)$ verifying $\partial\Gamma_0 = A_0$
$\Gamma_0 \leftarrow 0$
$A \leftarrow A_0$
while $A \neq 0$ **do**
  $a \leftarrow$ `GetHighestVertex(`A`)`
  $V_a \leftarrow$ `GetAdjacentVertices(`a`,` A`)`
  $LL \leftarrow$ `GetLowerLink(`a`,` $\mathcal{D}el(\mathbf{P})$`)`
  $E \leftarrow$ `GetEdgesConnecting(`V_a`,`LL`)`
  for $v \in V_a$ **do**
    $A \leftarrow A - a \vee v$
  end
  for $e \in E$ **do**
    $A \leftarrow A + e$
    $\Gamma_0 \leftarrow \Gamma_0 + a \vee e$
  end
end

The discussion of Algorithm 11 is divided in two: the first lemma is given to prove the correctness of the algorithm and describes succinctly each subroutine. A better description of these subroutines is then given in order to analyze the algorithm's complexity. We denote by $\vee$ the join operator on simplices, corresponding to the union for disjoint abstract simplices. In particular, for two 0-simplices a, v, $a \vee v$ denotes the 1-simplex $[a, v]$. For a 0-simplex a and a 1-simplex $e = [e_1, e_2]$, $a \vee e$ denotes the 2-simplex $[a, e_1, e_2]$.

Lemma 4.44. *Given a Delaunay complex $K = \mathcal{D}el(\mathbf{P})$ of a set of points $\mathbf{P} \subset \mathbb{R}^3$ verifying Condition 4 and an 1-cycle $A_0 \in \mathbf{Z}_1(K; \mathbb{Z}_2)$, Algorithm 11 computes a 2-chain $\Gamma_0 \in \mathbf{C}_2(K; \mathbb{Z}_2)$ such that $\partial\Gamma_0 = A_0$.*

Proof. We verify immediately that with the algorithm's initialization of Γ_0, if A_0 is empty, the algorithm returns and the property $\partial\Gamma_0 = A_0$ is verified.

The following invariants are shown at each iteration of the while loop:

— $\partial\Gamma_0 = A_0 + A$

— A is a cycle and the z-coordinate of its highest vertex decreases,

The first invariant will imply that, if the algorithm terminates with $A = 0$, we have that $\partial\Gamma_0 = A_0$. The second invariant implies that the algorithm must terminate.

We now describe the operations performed inside the while loop. Figure 4.20 illustrates one iteration of this loop. The procedure **GetHighestVertex**(A) returns a, the vertex in A with the maximal z-coordinate. Since A is a cycle, an even number of edges of A connects a with a set V_a of vertices in the link $\mathrm{lk}_{\mathcal{D}el(\mathbf{P})}(a)$ of a. This set is returned by the procedure **GetAdjacentVertices**(a, A). The fact a is the highest vertex in A implies the set V_a is a subset of the vertices of lower link $\mathrm{llk}_{\mathcal{D}el(\mathbf{P})}(a)$. This set is not empty and has again an even cardinality. As A is not zero, it must contain at

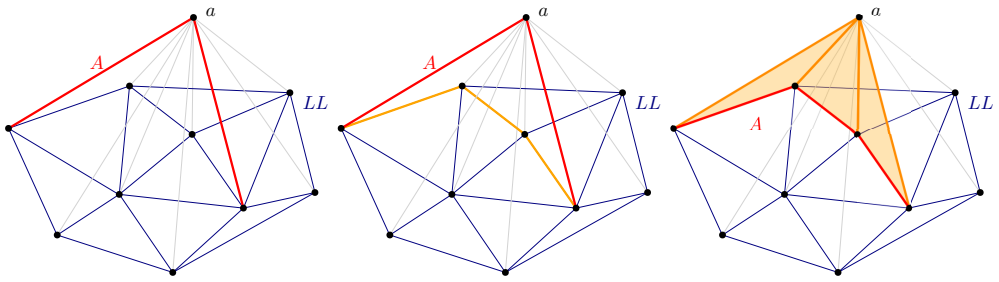


Figure 4.20: Left: the lower link LL of a (in blue) with the two adjacent edges of a in A (in red). Center: a path in LL between the two adjacent vertex of a in A (in orange). Right: created triangles (in transparent orange) join of a. The path in LL between the two adjacent vertices replaces the edges adjacent to a in A (in red).

least 2 distinct vertices and therefore, its highest vertex a cannot be the lowest vertex in $\mathcal{D}el(\mathbf{P})$. Lemma 4.43 asserts then that the lower link of a is non empty and contractible, proving as a consequence the existence of a 1-chain E bounded by the 0-chain formed by the even set of vertices in V_a. The procedure **GetLowerLink**$(a, \mathcal{D}el(\mathbf{P}))$ returns the 1-skeleton LL of the lower link of a in $\mathcal{D}el(\mathbf{P})$. **GetEdgesConnecting**$(V_a, LL)$ uses this 1-skeleton LL to construct a 1-chain $E \in \mathbf{C}_1(LL; \mathbb{Z}_2)$ that verifies $\partial E = V_a$.

The main step of the algorithm consists in replacing the edges connecting a in A by the edges on this chain E and adding, for each edge e of E, the corresponding triangle $a \vee e$ in Γ_0. The following 2-chain is added to Γ_0:

$$\sum_{e \in E} a \vee e \tag{4.32}$$

and its boundary can be evaluated as:

$$\partial\Big(\sum_{e \in E} a \vee e\Big) = \sum_{e \in E} (e - a \vee \partial e) = E - a \vee \partial E = E - a \vee V_a \tag{4.33}$$

From Equation (4.33), the boundary of the 2-chain added to Γ_0 corresponds to the 1-chain added to A. This means if the invariant $\partial\Gamma_0 = A_0 + A$ was verified at previous iteration, it still remain true after additions to Γ_0 and A: the first invariant is therefore shown. As A is initialized to A_0 and is updated by adding a boundary, A remains a cycle at each step. Also, all edges connecting a are removed from A and the added edges are in the lower link $\mathrm{llk}_{\mathcal{D}el(\mathbf{P})}(a)$ of a. This shows the second invariant. □

We now give more details on each subroutine used in Algorithm 11 to derive its complexity.

Lemma 4.45. *Under Condition 4, Algorithm 11 in the Delaunay complex $\mathcal{D}el(\mathbf{P})$ can be implemented in $\mathcal{O}(n \log n)$ time complexity, where n corresponds to the size of the complex $\mathcal{D}el(\mathbf{P})$.*

Proof. We first describe a preprocessing step to query the Delaunay triangulation, especially for the **GetLowerLink** operation. Under Condition 4, by writing each edge in $\mathcal{D}el(\mathbf{P})$ as a pair of vertices (v_1, v_2) such that $z(v_1) > z(v_2)$, one can define the following total order on the edges of $\mathcal{D}el(\mathbf{P})$:

$$\begin{aligned}(v_1, v_2) \leq (v_1', v_2') \quad &\underset{\text{def.}}{\Longleftrightarrow} \quad z(v_1) > z(v_1') \\ &\text{or} \quad v_1 = v_1' \text{ and } z(v_2) \geq z(v_2')\end{aligned}$$

The set of vertices of the lower link $\mathrm{llk}_{\mathcal{D}el(\mathbf{P})}(a)$ of a vertex a is in one-to-one correspondence with the set of ordered pairs whose first vertex is a. These pairs in the form $(a, \cdot)$ are contiguous in the set of all edges sorted according to this total order.

Similarly, each triangle in $\mathcal{D}el(\mathbf{P})$ can be represented by the ordered triple of vertex (v_1, v_2, v_3) such that $z(v_1) > z(v_2) > z(v_3)$ and one can define the following total order on the triangles of $\mathcal{D}el(\mathbf{P})$:

$$(v_1, v_2, v_3) \leq (v'_1, v'_2, v'_3) \quad \underset{\text{def.}}{\Longleftrightarrow} \quad \begin{cases} z(v_1) > z(v'_1) \\ \text{or} \quad v_1 = v'_1 \text{ and } z(v_2) > z(v'_2) \\ \text{or} \quad v_1 = v'_1 \text{ and } v_2 = v'_2 \text{ and } z(v_3) \geq z(v'_3) \end{cases}$$

As previously, the set of edges of the lower link $\mathrm{llk}_{\mathcal{D}el(\mathbf{P})}(a)$ of a vertex a is in one-to-one correspondence with the set of ordered triples whose first vertex is a. These triples in the form $(a, \cdot, \cdot)$ are again contiguous in the set of all triples ordered according to this total order.

Creating these representations (sorted edges and triangles) for the whole Delaunay complex $\mathcal{D}el(\mathbf{P})$ costs $O(n \log n)$, where n is the size of $\mathcal{D}el(\mathbf{P})$.

Procedure **GetLowerLink**$(a, \mathcal{D}el(\mathbf{P}))$, which returns the 1-skeleton LL of the lower link of a in $\mathcal{D}el(\mathbf{P})$, now has a $\log(n)$ time complexity to find the contiguous entries in the ordered sets of edges and triangles of the respective forms $(a, \cdot)$ and $(a, \cdot, \cdot)$.

The cycle A of the algorithm is also represented as an ordered set along the described total order on edges. Getting the largest element of A in the subroutine **GetHighestVertex**(A) and the edges associated with this vertex with the subroutine **GetAdjacentVertices**(a, A) can be performed in a $\mathcal{O}(1)$ time complexity in this ordered representation. Finally, each update of the cycle A (insertion or deletion) can be computed in $\mathcal{O}(\log n)$ time complexity.

The subroutine **GetEdgesConnecting**(V_a, LL) requires a bit more attention to derive the correct complexity. Denote by m the size of the lower link LL of a vertex a. The most obvious way of implementing this subroutine is by creating a spanning tree T of the lower link graph (using a breadth-first search for instance), which has a $\mathcal{O}(m)$ time complexity. From this spanning tree, we could partition the even set of vertices V_a into pairs and construct a path in T for each of these pairs. However, if V_a contains $2k = \mathcal{O}(m)$ vertices, this leads to computing k paths in the spanning tree, which has a $\mathcal{O}(m^2)$ time complexity, followed by adding up to km edges in A, which has an $\mathcal{O}(m^2 \log n)$ time complexity. Instead, we describe Algorithm 12. When creating this spanning tree T of the 1-skeleton LL of the lower link of a vertex a, we also assign an integer rank to each vertex such that the root has rank 0 and each non root vertex has a rank higher than its parent (this is a linear-time operation and can be seen as a topological sort of the spanning tree).

Note the resemblance of Algorithm 12 to Algorithm 11, but with one dimension less: given a 0-cycle V_a, Algorithm 12 finds a 1-chain E such that $\partial E = V_a$. In fact, Lemma 4.43 still applies, where the height z is replaced by the vertex rank. Indeed, each vertex which is not the root has as lower link in T: a single vertex, representing its unique parent node in T. This lower link is then contractible. The lower link of the root is empty.

The procedure **GetHighestVertex**(V) of Algorithm 11 is replaced by an iteration on all vertices of T, from the vertex of highest rank $m - 1$ to the vertex with rank 1, and the test verifying that v belongs to V. This iteration ends at rank 1, because, since at each step V is a 0-boundary and therefore contains an even number of vertices, the highest (minimal rank) vertex on V cannot be **Vertex**(0). If an array of size m contains all vertices indexed by their rank, each call to the procedure **Vertex**$(rank)$ costs $O(1)$. If the evolving set of vertex V is represented by an array of booleans of size m, where entry

Algorithm 12: GetEdgesConnecting

Inputs : An even set of vertices V_a in the lower link of a vertex a and a spanning tree T of the 1-skeleton LL of this lower link
Output: A set of edges E in T such that $\partial E = V_a$

```
E ← 0
V ← V_a
for rank ← m' − 1, 1 do
    v ← Vertex(rank)
    if v ∈ V then
        p ← GetParentInTree(v, T)
        V ← V − v + p
        E ← E + v ∨ p
    end
end
```

k indicate the membership to V of the vertex with rank k, the membership predicate $v \in V$ costs $O(1)$. When V is updated, updating this membership array can be done also in time $O(1)$. It follows that each line in the algorithm costs $O(1)$. For this reason, the cost of Algorithm 12 is $O(m)$. The proof of correctness is similar to the proof of correctness of Algorithm 11 and is based on preserving the following property along the algorithm:

$$\partial E = V_a + V \tag{4.34}$$

We now summarize the discussion of complexity. After an initial $\mathcal{O}(n \log n)$ time complexity one-time preprocessing of edges and triangles of the Delaunay triangulation, the complexity of each operation **GetLowerLink**, **GetHighestVertex** and **GetAdjacentVertices** can be upper bounded by $\mathcal{O}(\log n)$. With m the size of the lower link of a vertex, the subroutine **GetEdgesConnecting** can be performed in $\mathcal{O}(m)$ time complexity and outputs a set of edges E of size $\mathcal{O}(m)$. Keeping the ordered representation of A at each iteration (i.e. insertion and deletion into an ordered map) leads to a $\mathcal{O}(m \log n)$ time complexity. Finally, each vertex of $\mathbf{P}$ is visited at most once as the highest vertex of A. Note also that the set of edges of all lower links are in 1-to-1 correspondence with the triangles of $\mathcal{D}el(\mathbf{P})$, which means the sum of the sizes of all 1-skeletons of lower links is upper bounded by the size n of the complex. We can therefore conclude the global complexity of Algorithm 11 is $\mathcal{O}(n \log n)$. □

4.7 Critical basis of cycles

This section highlights deeper connections between lexicographic optimal chains and homology groups. From there, we define a basis of optimal cycles which we call *critical basis* and show how to construct it. A possible application of these critical basis will be given in Section 4.8.3. The chain coefficient group is assumed to be a general field $\mathbb{F}$.

4.7.1 Optimal homology representatives

From the linearity of M_{lex} (Lemma 4.12), the following vector subspace $\mathbf{Z}_k^{\min}(K, B)$ of $\mathbf{Z}_k(K, B)$, made of chains that are minimal among relative homologous chains, can be defined:

$$\mathbf{Z}_k^{\min}(K, B) \underset{\text{def.}}{=} M_{lex}(\mathbf{Z}_k(K, B))$$

Note first that M_{lex} can be seen an idempotent application from $\mathbf{Z}_k(K,B)$ to $\mathbf{Z}_k(K,B)$, and therefore a projection map onto $\mathbf{Z}_k^{\min}(K)$. Moreover, the following diagram commutes and implies an isomorphism from $\mathbf{Z}_k^{\min}(K,B)$ to $\mathcal{H}_k(K,B)$:

$$\begin{array}{ccccc} \mathbf{Z}_k(K,B) & \xrightarrow{M_{lex}} & \mathbf{Z}_k^{\min}(K,B) & \xrightarrow{\subset} & \mathbf{Z}_k(K,B) \\ & \searrow & \downarrow \textbf{iso.} & & \\ & & \mathcal{H}_k(K,B) & & \end{array}$$

The same construction applies identically to absolute homology, by taking an empty subcomplex B.

We see therefore that vector spaces of lexicographic minimal chains are isomorphic to corresponding homology groups. This mimics the property of harmonic forms. Indeed, in Hodge theory, harmonic forms, which are most often defined as being in the kernels of Laplacian operators, can be defined, equivalently, as L^2 minima of chains in their homology classes.

Also, the map that associates to a chain Γ its L^2 minimum in its homology class, i.e. the unique homologous harmonic form, is linear as is linear any map that associate to a vector V the minimum of a positive definite quadratic form on the affine space $V+B$, for a linear subspace B (in the case of homology, B is the space of boundaries). Our map M_{lex} enjoys the same linearity, as shown in Lemma 4.12.

The computation of L^1 minimal chains in their homology class has been intensively studied [EW05, CF08, CF10, DSW10, DHK11, CF11, BCC+12, CV15, WCW+17, DLW18, CVKP18, DHM20, BMN20, Att21], in particular because, unlike L^2 minima, the L^1 minima are sparse and as such are visually meaningful geometric representations of corresponding homology classes.

Our lexicographic minimal chains benefit from both properties: the isomorphism with homology of L^2 minima in Hodge theory, as well as the sparsity of support of L^1 minima.

4.7.2 Critical basis of cycles

The linear space $\mathbf{Z}_k^{\min}(K)$, made of chains that are lexicographic minimal inside their homology classes, admits a canonical basis that we call *critical basis* since each basis vector is associated to a critical simplex. The definition is given in this section explicitly only for the space $\mathbf{Z}_k^{\min}(K)$, since the exact same construction applies in the context of relative chains, and critical bases are defined similarly for the space $\mathbf{Z}_k^{\min}(K,B)$.

Definition 4.46 (Critical simplex)**.** *For $\Gamma \in \mathbf{Z}_k^{\min}(K)$, the critical simplex*

$$\operatorname{crit}(\Gamma) \underset{def.}{=} \max|\Gamma|$$

is the maximal k-simplex in the support $|\Gamma|$ of Γ.

Definition 4.47 (Critical basis)**.** *We say that $\mathbf{b} = (\mathbf{b}_1, \dots, \mathbf{b}_m) \in \big(\mathbf{Z}_k^{\min}(K)\big)^m$ is a critical basis of $\mathbf{Z}_k^{\min}(K)$ if and only if, for $0 \le i \le m-1$:*

$$\mathbf{b}_{i+1} = \min_{\sqsubseteq_{lex}} \big\{ \Gamma \in \mathbf{Z}_k^{\min}(K) \;\mid\; \Gamma \notin \operatorname{span}(\mathbf{b}_1, \dots, \mathbf{b}_i) \quad \text{and} \quad \Gamma\big(\operatorname{crit}(\Gamma)\big) = 1 \big\}$$

with the convention $\operatorname{span}(\emptyset) = \{0\}$. When $\mathbb{F} = \mathbb{Z}_2$, the condition $\Gamma\big(\operatorname{crit}(\Gamma)\big) = 1$ is unnecessary.

Lemma 4.48. *The critical basis is canonical, in other words there is a unique ordered sequence $\mathbf{b} = (\mathbf{b}_1, \dots, \mathbf{b}_m)$ satisfying Definition 4.47.*

Proof. Assume, for $i \leq m$, the first $i-1$ critical basis elements have been constructed, and denote $S_{i-1} = \text{span}(\mathbf{b}_1, , \ldots, \mathbf{b}_{i-1})$, with $S_0 = \{0\}$. Since the dimension of $\mathbf{Z}_k^{\min} = \mathbf{Z}_k^{\min}(K)$ is $m > i-1$, there is at least one chain in $\mathbf{Z}_k^{\min} \setminus S_{i-1}$, and the number of possible supports being finite, a minimum among them exists. From Observation 4.7, two lexicographic minima of $\mathbf{Z}_k^{\min} \setminus S_{i-1}$ must have same support. Consider, for a contradiction, two such minima $\Gamma_1 \neq \Gamma_2$ with $\Gamma_1\big(\text{crit}(\Gamma_1)\big) = \Gamma_2\big(\text{crit}(\Gamma_2)\big) = 1$.

One has $|\Gamma_1| = |\Gamma_2|$ so that $\text{crit}(\Gamma_1) = \text{crit}(\Gamma_2)$. Take $\sigma \in |\Gamma_1|$ such that $\Gamma_1(\sigma) \neq \Gamma_2(\sigma)$ and construct the k-cycle

$$\Gamma \underset{\text{def.}}{=} \frac{1}{\Gamma_2(\sigma) - \Gamma_1(\sigma)} \Big(\Gamma_2(\sigma)\, \Gamma_1 - \Gamma_1(\sigma)\, \Gamma_2 \Big)$$

Observe that Γ verifies $\Gamma(\text{crit}(\Gamma)) = \Gamma(\text{crit}(\Gamma_1)) = 1$ and $\Gamma(\sigma) = 0$. As $|\Gamma| \subset |\Gamma_1| \setminus \{\sigma\}$, Γ is strictly smaller for the lexicographic order $\sqsubseteq_{lex}$ than Γ_1. If $\Gamma \notin S_{i-1}$, this is an immediate contradiction with the optimality of Γ_1. Otherwise, if $\Gamma \in S_{i-1}$, we can again consider an element $\sigma' \in |\Gamma| \subset |\Gamma_1|$ and construct

$$\Gamma' \underset{\text{def.}}{=} \Gamma_1 - \frac{\Gamma_1(\sigma')}{\Gamma(\sigma')} \Gamma$$

which is smaller than Γ_1 since $|\Gamma'| \subset |\Gamma_1| \setminus \{\sigma'\}$ and $\Gamma' \notin S_{i-1}$, being the sum of $\Gamma_1 \notin S_{i-1}$ and $-\frac{\Gamma_1(\sigma')}{\Gamma(\sigma')}\Gamma \in S_{i-1}$. We have again a contradiction. □

We give two essential properties of critical bases. Since the critical basis elements are defined as minima over an inclusion decreasing sequence of sets, we have:

Observation 4.49. *For any $i, j = 1, \ldots, m$*

$$i \leq j \implies b_i \sqsubseteq_{lex} b_j$$

Lemma 4.50. *For any $i, j = 1, \ldots, m$*

$$\mathbf{b}_j(\text{crit}(\mathbf{b}_i)) = \delta_{ij}$$

where δ_{ij} is the Kronecker delta.

Proof. One has immediately that $\mathbf{b}_i(\text{crit}(\mathbf{b}_i)) = 1$. The following property

$$j < i \implies \mathbf{b}_j(\text{crit}(\mathbf{b}_i)) = 0,$$

is verified as, if this was not the case, then $\mathbf{b}_j(\text{crit}(\mathbf{b}_i))\, \mathbf{b}_i - \mathbf{b}_j$ would be strictly smaller than $\mathbf{b}_i$ and not in $\text{span}(\mathbf{b}_1, \ldots, \mathbf{b}_{i-1})$, a contradiction with the minimality of $\mathbf{b}_i$. Symmetrically, a similar argument holds to show that

$$j > i \implies \mathbf{b}_j(\text{crit}(\mathbf{b}_i)) = 0, \tag{4.35}$$

as, if this was not the case, then $\mathbf{b}_j - \mathbf{b}_j(\text{crit}(\mathbf{b}_i))\, \mathbf{b}_i$ would be strictly smaller than $\mathbf{b}_j$ and not in $\text{span}(\mathbf{b}_1, \ldots, \mathbf{b}_{j-1})$, contradicting the minimality of $\mathbf{b}_j$. □

It follows that, for $i = 1, \ldots, m$, $\Gamma \in \text{span}(\mathbf{b}_1, \ldots, \mathbf{b}_{i-1}) \implies \Gamma(\text{crit}(\mathbf{b}_i)) = 0$ and therefore:

$$\Big\{ \Gamma \in \mathbf{Z}_k^{\min}(K),\ \Gamma(\text{crit}(\mathbf{b}_i)) = 1 \Big\} \subset \mathbf{Z}_k^{\min}(K) \setminus \text{span}(\mathbf{b}_1, \ldots, \mathbf{b}_{i-1})$$

Since $\mathbf{b}_i$ is in the first set and, by definition, minimum of the second set, one has:

Observation 4.51. *For $i = 1, \ldots, m$:*

$$\mathbf{b}_i = \min_{\sqsubseteq_{lex}} \Big\{ \Gamma \in \mathbf{Z}_k^{\min}(K),\ \Gamma(\text{crit}(\mathbf{b}_i)) = 1 \Big\}$$

4.7.3 Computation of a critical basis from optimal homology representatives

We now bridge the gap between optimal representatives in $\mathbf{Z}_k^{\min}(K)$ and the critical basis of $\mathbf{Z}_k^{\min}(K)$, allowing to construct the critical basis from any basis of $\mathbf{Z}_k^{\min}(K)$.

Lemma 4.52. *Consider a basis $(\Gamma_i)_{i=1,\dots,m}$ of $\mathbf{Z}_k^{\min}(K)$ that verifies for all $i, j = 1, \dots, m$*

$$\begin{cases} \Gamma_i(\mathrm{crit}(\Gamma_i)) = 1 \\ i \leq j \implies \Gamma_i \sqsubseteq_{lex} \Gamma_j \\ i \neq j \implies \mathrm{crit}(\Gamma_i) \neq \mathrm{crit}(\Gamma_j) \end{cases} \tag{4.36}$$

The critical basis of $\mathbf{Z}_k^{\min}(K)$ can be constructed, for all $i = 1, \dots, m$, as

$$\mathbf{b}_i = \min_{\sqsubseteq_{lex}} \Gamma_i + \mathrm{span}(\Gamma_1, \dots, \Gamma_{i-1}) \tag{4.37}$$

Proof. By writing an element $\mathbf{b}_i$ in the basis $(\Gamma_j)_{j=1,\dots,m}$, the third property of Equation (4.36) requires that the critical simplex of $\mathbf{b}_i$ corresponds to a critical simplex of one element of the basis $(\Gamma_j)_{j=1,\dots,m}$. Therefore, the critical simplices of $(\mathbf{b}_i)_{i=1,\dots,m}$ form a subset of the critical simplices of $(\Gamma_i)_{i=1,\dots,m}$, with same cardinality. This implies that the set of critical simplices is the same between $(\mathbf{b}_i)_{i=1,\dots,m}$ and $(\Gamma_i)_{i=1,\dots,m}$, and in the same order thanks to the second property of Equation (4.36) and Observation 4.49. For $i = 1, \dots, m$, we have:

$$\mathrm{crit}(\Gamma_i) = \mathrm{crit}(\mathbf{b}_i)$$

We now show Equation (4.37). Denote by

$$\Gamma_i^{\min} \underset{\text{def.}}{=} \min_{\sqsubseteq_{lex}} \Gamma_i + \mathrm{span}(\Gamma_1, \dots, \Gamma_{i-1})$$

Any element Γ of the set $\Gamma_i + \mathrm{span}(\Gamma_1, \dots, \Gamma_{i-1})$ verifies $\Gamma(\mathrm{crit}(\mathbf{b}_i)) = \Gamma(\mathrm{crit}(\Gamma_i)) = 1$ as $\Gamma_i(\mathrm{crit}(\Gamma_i)) = 1$ (first condition of Equation (4.36)) and, from the two last conditions of Equation (4.36), for any $j < i$, $\mathrm{crit}(\Gamma_j) < \mathrm{crit}(\Gamma_i)$ therefore $\Gamma_j(\mathrm{crit}(\Gamma_i)) = 0$. We have thus that:

$$\Gamma_i + \mathrm{span}(\Gamma_1, \dots, \Gamma_{i-1}) \subset \left\{\Gamma \in \mathbf{Z}_k^{\min}(K),\ \Gamma(\mathrm{crit}(\mathbf{b}_i)) = 1\right\}$$

and from Observation 4.51,

$$\mathbf{b}_i \sqsubseteq_{lex} \Gamma_i^{\min}$$

Next, we write $\mathbf{b}_i$ in the basis $(\Gamma_j)_{j=1,\dots,m}$ of $\mathbf{Z}_k^{\min}(K)$:

$$\mathbf{b}_i = \sum_{j=1}^{m} \lambda_j \Gamma_j$$

From the third condition of Equation (4.36), we have that:

$$\mathrm{crit}(\mathbf{b}_i) = \max\left\{\mathrm{crit}(\Gamma_j), \forall 1 \leq j \leq m \mid\ \lambda_j \neq 0\right\}$$

As $\mathrm{crit}(\mathbf{b}_i) = \mathrm{crit}(\Gamma_i)$, this implies that $\lambda_j = 0$ for any $j > i$ and thus $\mathbf{b}_i \in \Gamma_i + \mathrm{span}(\Gamma_1, \dots, \Gamma_{i-1})$ resulting in:

$$\Gamma_i^{\min} \sqsubseteq_{lex} \mathbf{b}_i$$

We have therefore that $|\Gamma_i^{\min}| = |\mathbf{b}_i|$ from Observation 4.7. Suppose for a contradiction that $\Gamma_i^{\min} \neq \mathbf{b}_i$ and consider $\sigma \in |\Gamma_i^{\min} - \mathbf{b}_i|$. The following chain

$$\Gamma \underset{\text{def.}}{=} \Gamma_i^{\min} - \frac{\Gamma_i^{\min}(\sigma)}{\Gamma_i^{\min}(\sigma) - \mathbf{b}_i(\sigma)} \left(\Gamma_i^{\min} - \mathbf{b}_i\right)$$

verifies $|\Gamma| \subset |\Gamma_i^{\min}| \setminus \{\sigma\}$ and thus $\Gamma \sqsubset_{lex} \Gamma_i^{\min}$. Also as $\mathbf{b}_i \in \Gamma_i + \text{span}(\Gamma_1, \ldots, \Gamma_{i-1})$, $\Gamma_i^{\min} - \mathbf{b}_i \in \text{span}(\Gamma_1, \ldots, \Gamma_{i-1})$ and $\Gamma \in \Gamma_i + \text{span}(\Gamma_1, \ldots, \Gamma_{i-1})$. We have a contradiction with $\Gamma_i^{\min}$ as minimum of the set $\Gamma_i + \text{span}(\Gamma_1, \ldots, \Gamma_{i-1})$. We conclude that $\Gamma_i^{\min} = \mathbf{b}_i$. □

The next two observations allow to compute the critical basis from matrix reduction algorithms previously described in Section 4.3 when $\mathbb{F} = \mathbb{Z}_2$.

Observation 4.53. *For $\mathbb{F} = \mathbb{Z}_2$, when considering a k-chain Γ as a vector of $\mathbb{Z}_2^m$ written in the basis of ordered k-simplices of dimension m, the* low *index corresponds to the index of the critical simplex:*

$$\text{crit}(\Gamma) = \sigma_{\text{low}(\Gamma)}$$

The third property of Equation (4.36) *can therefore be understood as verifying that the lows of the set $(\Gamma_i)_{i=1,\ldots,m}$, seen as vectors in $\mathbb{Z}_2^m$, are unique.*

Observation 4.54. *Consider a set $(\Gamma_i)_{i=1,\ldots,m}$ basis of $\mathbf{Z}_k^{\min}(K; \mathbb{Z}_2)$ and the matrix M whose columns are the corresponding vector representations in $\mathbb{Z}_2^m$ of $(\Gamma_i)_{i=1,\ldots,m}$ in the basis of ordered k-simplices of dimension m.*

- *The algorithm transforming this set $(\Gamma_i)_{i=1,\ldots,m}$ such that it verifies the properties of Equation* (4.36) *is the matrix reduction algorithm (Algorithm 2) of the matrix M, followed by a sort in increasing order along the lexicographic order on k-chains of the columns of M, resulting in a matrix R.*
- *The construction of each element $\mathbf{b}_i$ of the critical basis as described in Lemma 4.52 corresponds to the total reduction algorithm (Algorithm 3) on the column R_i with the submatrix of R formed by the first $(i-1)$ columns.*

4.8 Applications to urban reconstruction

We now explore three types of efficient reconstruction methods from the algorithms detailed in this chapter.

4.8.1 Closed surface reconstruction

The process of reconstruction for closed surfaces is illustrated in Figure 4.21. The Delaunay 3-complex is first computed from the input set of points $\mathbf{P}$, thanks to the CGAL library [JPT21]. When constructing its dual graph, we complete it into a topological 3-sphere by connecting, for any triangle on the convex hull of the Delaunay triangulation, its dual edge to an "infinite" dual vertex. Considering 2-cycles in this Delaunay 3-complex – which is an oriented 3-manifold – allows to use Algorithm 8, which was formulated as a lexicographic mincut problem between two dual vertices α_1, α_2, that we can respectively qualify as *interior* and *exterior* constraints. These vertices need to be provided as inputs to the reconstruction, although the exterior α_2 constraint can reasonably be set to the infinite dual vertex, representing the outside of the convex hull of $\mathbf{P}$. All triangles of the complex (i.e. dual edges) need to be sorted in decreasing order along the total order on 2-simplices defined in Lemma 4.14. Note that Algorithm 8 is purely combinatorial, the only geometrical information is given by this order on dual edges. This means the radii quantities appearing in the 2-simplex order can be computed in fixed precision and the sorting does not require exact predicates. The mincut formulation implies that the 2-chain resulting from Algorithm 8 is the boundary of a

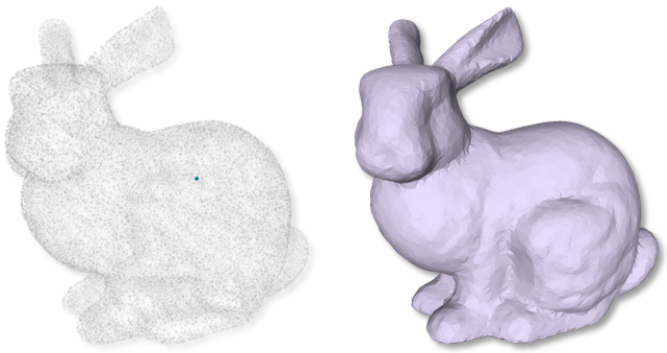


Figure 4.21: Illustration of the closed surface reconstruction on the Stanford Bunny [GT94]. Left: in addition to the set of points, an interior point (in blue) is given: the Delaunay cell in which it lies will be the α_1 interior constraint, the exterior constraint α_2 will be set to the "infinite" dual vertex. Right: the closed surface reconstruction formulated as lexicographic optimal chain.

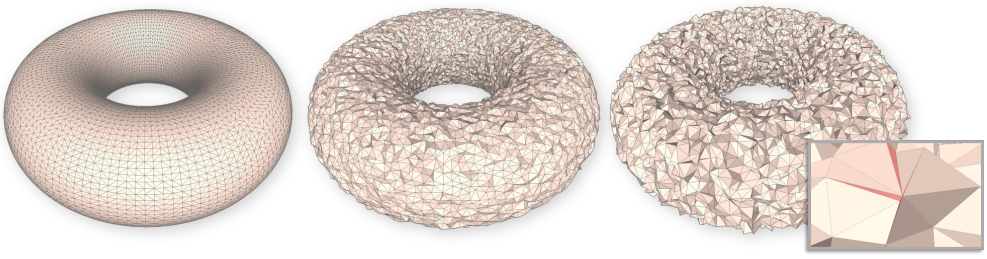


Figure 4.22: Reconstructions under different perturbations. Under no (left) or small (middle) perturbations, the reconstruction is a triangulation of the sampled manifold. A few non-manifold configurations can appear however under larger perturbations (right).

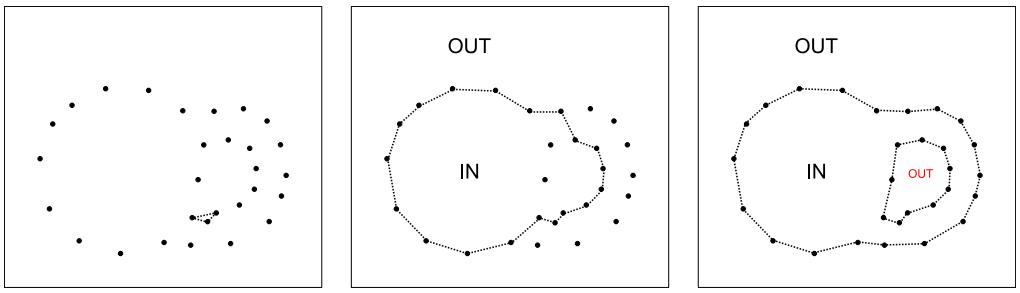


Figure 4.23: Illustration of how the topological information can effectively guide the shape reconstruction. The lexicographic order on 1-chains is induced by edge length comparison. Left: the global mincut is often not relevant when dealing with noisy or non-uniform sets of points. Middle and right: providing additional topological information can improve the result of the reconstruction.

union of tetrahedrons. It can be oriented and, although it might contain non-manifold configurations, these can be systematically dealt with.

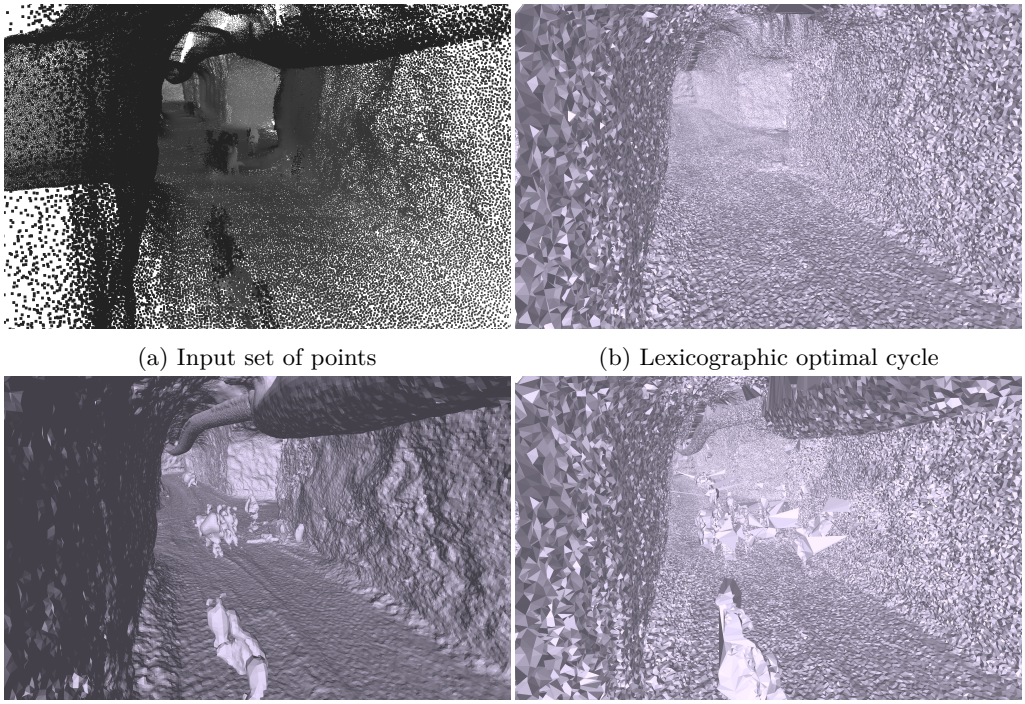
The total order on 2-simplices was justified in Section 4.2.3 by its connection to 2-dimensional Delaunay triangulations: this does not guarantee however that using the same order to reconstruct 2-surfaces in 3D will be relevant. However, triangles of 2D Delaunay triangulation are known to have some nice properties, such as maximizing the minimal angle or minimizing the largest circumcircle. When using the same total order to minimize 2-chains in dimension 3, we can hope some of these properties transfer to the reconstruction. In fact, it can been shown that for a Čech or Vietoris-Rips complex, under very strict conditions linking the point set sampling, the parameter of the complex and the reach of the underlying manifold of Euclidean space, the minimal lexicographic chain in the fundamental class of the complex using the described simplex order is a triangulation of the sampled manifold [CLV19a]. Experimental results (Figure 4.22) show that this property remains true relatively far from these theoretical conditions.

As urban scenes present a lot more open surfaces than closed surfaces, we illustrate the closed surface reconstruction on an interior structure (Figure 4.24). We observe that the formulation as a global optimization allows to filter out noise as well as small thin structures when compared to either a level set approximating method such as Poisson reconstruction [KH13] or the scale-space version of the advancing front interpolating reconstruction [DMSL11, CD04], both implemented in CGAL [ASG21, DC21, van21].

We investigate the effects of the input constraints α_1, α_2 on the reconstruction, as illustrated in Figure 4.23. Note first that a completely automatic method could be designed, which would compute the global lexicographic mincut problem: the lexicographic minimal chain separating the complex into two components. This would require only the point cloud as input and could work for clean and uniformly sampled points but would, however, not be robust in practical applications, as the global mincut is often irrelevant in these situations. On the other hand, additional constraints can better guide the reconstruction, especially when the input set of points is not sampled uniformly on the closed surface.

4.8.2 Open surface reconstruction

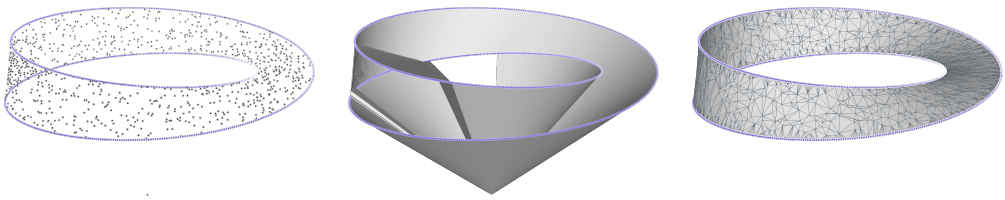
The context of urban reconstruction requires mostly to reconstruct portions of the Earth's surface and therefore open surfaces. Just as for closed surface reconstruction, we construct the 3-dimensional Delaunay triangulation of a set of points $\mathbf{P}$ and complete it into a topological 3-sphere. The process of open surface reconstruction is illustrated



(a) Input set of points (b) Lexicographic optimal cycle

(c) Poisson reconstruction (d) Advancing front reconstruction

Figure 4.24: Application of closed surface reconstruction as lexicographic optimal cycle. The formulation as an optimization problem helps avoid a lot of noise from the input point cloud.



(a) Inputs (b) Representative chain (c) Optimal chain

Figure 4.25: Illustration of the open surface reconstruction steps. From a set of points and a 1-cycle with coefficient in $\mathbb{Z}_2$ (a), a representative 2-chain bounding this 1-cycle is computed in the Delaunay 3-complex (b). Finally, the lexicographic optimal 2-chain homologous to the representative chain is computed (c). Note that an additional point far from the Mobius strip has been added to the set of points to better highlight the construction of the representative chain.

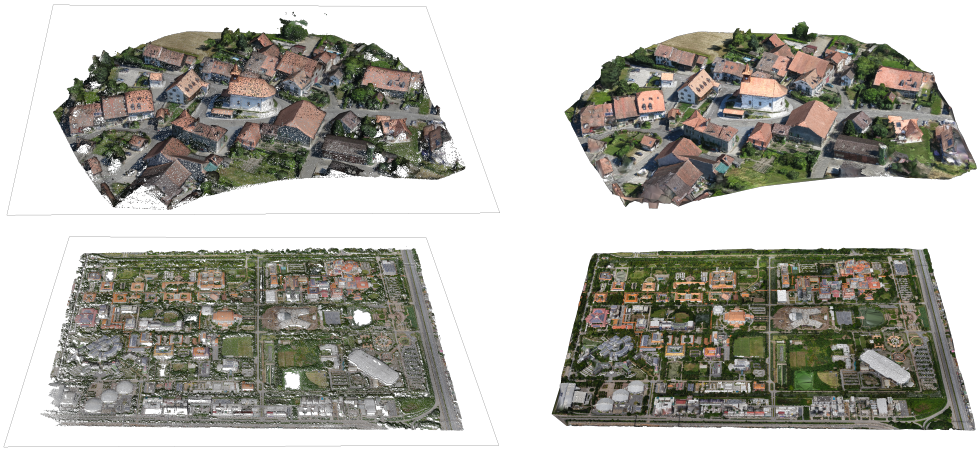


Figure 4.26: Example of open surface meshing on two models from the senseFly dataset [Sen11]. As the boundary is not part of the set of points, all triangles connecting any vertex of the boundary are removed in the resulting mesh.

in Figure 4.25. Assuming a 1-cycle embedded in the Delaunay triangulation is given, Algorithm 11 is used to compute a representative 2-chain having, as its boundary, the provided 1-cycle. Algorithm 10 then computes the optimal absolute chain homologous to the representative chain. As the 3-dimensional Delaunay complex has trivial 2-homology, this is equivalent to the optimal chain bounded by the provided 1-cycle. The resulting mesh is again the boundary of a union of tetrahedrons, whose arbitrary small offset is a pseudo-manifold with boundary. While closed surface reconstruction were formulated as the variant of a mincut problem and therefore the chain coefficient $\mathbb{Z}_2$ was used, different chain coefficients, namely $\mathbb{Z}_2$ and $\mathbb{Q}$, are interesting for open surface reconstruction. Indeed, Figure 4.25 shows that, when optimizing using $\mathbb{Z}_2$ coefficients, the resulting optimal chain in $\mathbb{Z}_2$ might not be orientable in $\mathbb{Q}$ and therefore the two minimization problems can give different results. In applications however, the choice of $\mathbb{Q}$ is more appealing to obtain an oriented result. We conjecture also, thanks to the unimodularity property of the boundary operator for oriented pseudomanifold [DHK11], that optimality problems with coefficients in $\mathbb{Z}$ can be relaxed to coefficient in $\mathbb{Q}$ and that both Algorithm 11 and Algorithm 10 can be extended to chains with coefficients in $\mathbb{Z}$.

Figure 4.26 illustrates the simplest use case of open surface reconstruction for terrains, where the boundary can be defined outside the set of points and the Delaunay triangulation is constructed on both the set of points and the points of this boundary. Because the points of the boundary are on the convex hull of the Delaunay triangulation, we are guaranteed that the edges of this boundary are in the Delaunay triangulation. We give in Table 4.2 a few performance information for each reconstruction case. The observed performance for the reconstruction correlates with the quasi-linear theoretical complexity. Like most reconstruction methods based on the interpolation of the input set of points, the reconstruction is far from being parsimonious, and the generated number of triangles can be a real limitation of the method for large sets of points.

When dealing with large urban scenes, point cloud data can be massive and meshing the whole dataset is prohibited by memory capacity. For instance, the 2015 Dublin LiDAR survey [LAA+17], shown in Figure 4.27 contains more than 1.4 billion points, organized into tiles. A tiling strategy for meshes is more difficult than for points, as it is expected that the mesh coincides correctly at the frontier of each tile. The ability to

Input		
Model name	Sullens	Thammasat
Number of points	1 056 038	5 225 819
Performance		
Delaunay triangulation	2 633 ms	14 416 ms
Representative chain (Algorithm 11)	27 ms	5 ms
Optimal chain (Algorithm 10)	1 548 ms	8 040 ms
Total elapsed time	4 208 ms	22 461 ms
Peak memory usage	1.4 GB	5.5 GB
Output		
Number of vertices	976 389	4 551 486
Number of triangles	1 953 211	9 108 074

Table 4.2: Performance on examples of the senseFly dataset [Sen11]

specify the boundary of our reconstruction method makes it a useful tool for this spatial decomposition. In Figure 4.27, the four exterior tiles are reconstructed independently – with the same technique as illustrated in Figure 4.26 – and the boundary of the resulting meshes can be combined to create a cycle which will define the boundary of the interior tile. The interior tile can then be reconstructed given this imposed boundary, the five meshes therefore coinciding at the frontier of each tiles. Note that the difficult question to automate this whole process is guaranteeing the presence of the provided cycle in the triangulation. This question is related to the challenging problem of conforming Delaunay tetrahedralization [CCY04, Rup95, Si15, Si08, MMG01, She98]. Here, to obtain the interior tile in Figure 4.26, some input points near the boundary have been removed to ensure the presence of the boundary in the Delaunay triangulation.

Finally, the previous open surface meshing applications were solved using the version of Algorithm 10 for absolute chains (by considering an empty subcomplex B). In Figure 4.28, we illustrate how the relative version of the algorithm is useful for remeshing purposes, for instance when a second partial acquisition needs to be merged with a previous larger mesh. With absolute chains, we can define a cycle on the mesh and find the lexicographic optimal chain bounding this cycle in the Delaunay triangulation of the new acquisition. However, defining this cycle depends strongly on the region impacted by the new acquisition and a badly localized cycle might generate defects along the junction between the old and new mesh. Instead, we can create a subcomplex B on the old mesh as an offset around this badly localized cycle and, after finding a representative 2-chain in the Delaunay triangulation K of the new acquisition, apply Algorithm 10 in the relative context to compute the lexicographic optimal relative chain homologous to the representative in the complex pair (K, B). This means that instead of bounding exactly the badly localized cycle, the resulting 2-relative chain can have its boundary anywhere inside B: roughly speaking, the resulting optimal 2-chain has the freedom of choosing where to place the junction between the old and new mesh. This junction should be cleaner than the one in the absolute context.

4.8.3 Boundary detection and critical basis

Finally, we described a process of reconstruction based on sharp feature detection and critical basis, illustrated in Figure 4.29. We first use the Voronoi-based curvature estimator called Voronoi Covariance Measure (VCM) [MOG11]. This method is able to estimate the curvature of a sampled surface from the covariance matrices of Voronoi cells. The idea of using the shape of Voronoi cells for the estimation of normal and

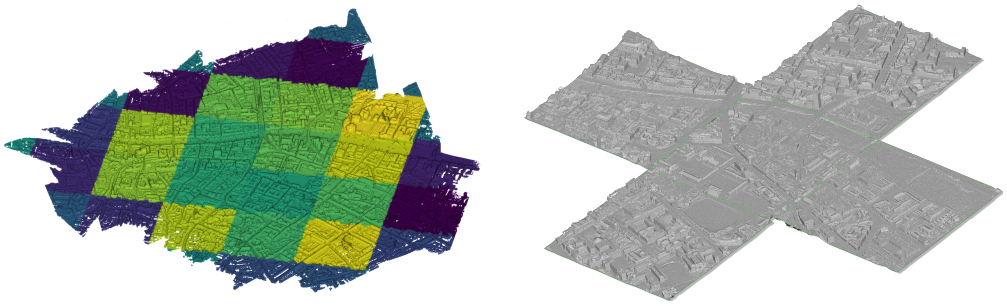
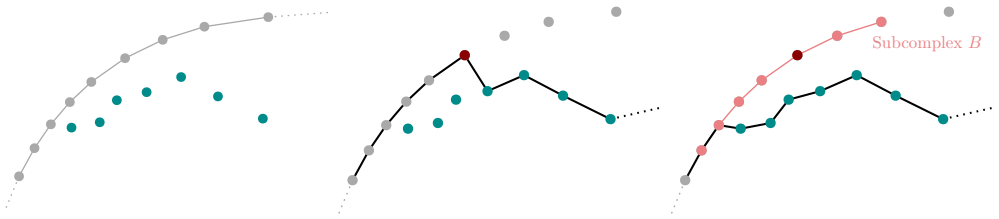
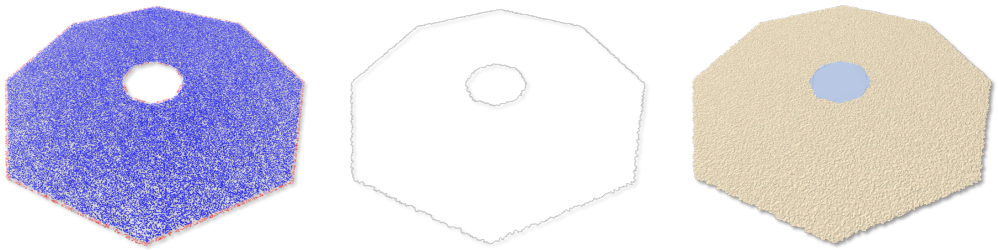


Figure 4.27: Example of reconstruction by spatial decomposition. Left: Visualization of the 2015 Dublin LiDAR survey [LAA+17], subsampled in order to visualize it in its entirety. Right: The center tile has been reconstructed under boundary imposed by its neighboring tiles. Each meshed tile is composed of about 9 million triangles.



(a) New Acquisition (b) Absolute context (c) Relative context

Figure 4.28: Illustration of the difference between absolute and relative optimal chains for remeshing purposes in 2 dimensions. (a) A new acquisition (points in cyan) needs to be reconstructed and merged with a previous reconstruction (shown in gray). The focus is put on one of the two junctions with the previous 1-chain. (b) A bad placement of the boundary (point in red) can lead to a partial and suboptimal reconstruction. (c) Using relative chains with a subcomplex B around the imposed boundary allows obtaining a smaller lexicographic optimal chain.



(a) VCM estimation (b) Optimal 1-cycles in B (c) Critical 2-basis of (K, B)

Figure 4.29: Illustration of a reconstruction process based on critical basis. An estimation of the boundary is computed on a set of points. By thresholding this estimator and constructing an subcomplex B offsetting a subset of points, a basis of $\mathbf{Z}_1^{\min}(B)$ is computed. By duality, this optimal basis is used to compute the critical basis of $\mathbf{Z}_2^{\min}(K, B)$.

curvature information is not new [ABK98, ACTD07, CCM10], however this method is resilient to non-local influences as well as noise by (i) computing the covariance matrix of a Voronoi cell $\mathcal{V}_P$ intersected by a small ball around the site P and (ii) considering a weighted average of neighboring covariance matrices, allowing for a better curvature estimation. The method can be efficiently computed for 3-dimensional point clouds. By thresholding this estimation, we can extract a subset $\mathcal{S}$ of vertices and construct a 2-subcomplex B of the Delaunay triangulation K, parameterized by a distance ϵ, as the union of all simplices up to dimension 2 that lie inside the ϵ-offset of $\mathcal{S}$.

By keeping the size of the subcomplex B, denoted by p, relatively small compared to the size n of the Delaunay 3-complex, we can use Algorithm 2 to compute the reduced boundary matrix of the complex B in $\mathcal{O}(p^3)$ time complexity. After extracting a basis of $\mathbf{Z}_1(B)$ by exploring the 1-skeleton of the subcomplex, we can use Algorithm 3 to compute the optimal homologous cycles for all cycles of B, and by filtering out all zero chains, we get a set of cycles $(\gamma_i)_{i=1,\dots,\beta} \in \mathbf{Z}_1(B)$ whose homology classes form a basis of $\mathcal{H}_1(B)$, with $\beta = \dim \mathcal{H}_1(B)$. Reduction algorithms are performed using the PHAT library [BKRW17].

As the Delaunay complex K triangulates the 3-sphere $\mathbb{S}^3$,

$$\mathcal{H}_1(K) = \mathcal{H}_1(\mathbb{S}^3) = \{0\}$$
$$\mathcal{H}_2(K) = \mathcal{H}_2(\mathbb{S}^3) = \{0\}$$

With $\rho^\star$ and $\iota^\star$ the homology morphisms induced respectively by the inclusion map $\rho : \mathbf{C}_2(K) \to \mathbf{C}_2(K, B)$ and the inclusion map $\iota : \mathbf{C}_1(B) \to \mathbf{C}_1(K)$, the following long exact sequence of relative homology

$$\{0\} = \mathcal{H}_2(K) \xrightarrow{\rho^\star} \mathcal{H}_2(K, B) \xrightarrow{\partial} \mathcal{H}_1(B) \xrightarrow{\iota^\star} \mathcal{H}_1(K) = \{0\}$$

implies that $\mathcal{H}_2(K, B) \xrightarrow{\partial} \mathcal{H}_1(B)$ is an isomorphism. By finding representative chains in $\mathbf{Z}_2(K, B)$ (using Algorithm 11) for each element of $(\gamma_i)_{i=1,\dots,\beta}$, we get a set of cycles whose homology classes form a basis of $\mathcal{H}_2(K, B)$.

Just as open surface reconstruction applications in Section 4.8.2, two optimizations of these representative chains are possible: either in the absolute complex K, meaning finding the lexicographic optimal chain having the same boundary as the representative, or in a simplicial pair (K, B), meaning finding the lexicographic optimal relative cycle homologous to the representative chain. When the simplicial complex B is of dimension 1, both approaches are of course equivalent. Again, the relative setting could improve the quality of the reconstruction when the detected cycles $(\gamma_i)_{i=1,\dots,\beta}$ of B are imprecise. However, one major inconvenient of the relative setting is that the reconstruction will be missing on the 2-skeleton of the subcomplex B. In the example shown in Figure 4.31, we will be considering the absolute setting.

Nonetheless, in both cases, the result of applying Algorithm 10 to each of these chains constructs a basis $(\Gamma_i)_{i=1,\dots,\beta}$ of $\mathbf{Z}_2^{\min}(K, B)$.

Finally, following Observation 4.54, the critical basis can be obtained from any basis of $\mathbf{Z}_2^{\min}(K, B)$ by computing the matrix reduction (Algorithm 2) of the matrix whose columns are vector representations in the basis of ordered k-simplices of $(\Gamma_i)_{i=1,\dots,\beta}$, followed by a sort in increasing order along the lexicographic order of these columns, and a final total reduction of each column. This construction has a $\mathcal{O}(\beta^2 n)$ time complexity. The whole process, excluding the complexity of computing the Delaunay triangulation and calling the feature estimator, has a $\mathcal{O}(p^3 + \beta n \log n + \beta^2 n)$ time complexity, where n is the size of the complex K, p the size of the subcomplex B, β the dimension of 1-homology of B (or equivalently the 2-homology of (K, B)).

A reconstruction result is presented in Figure 4.31 for mobile scanning input set of points, where occlusions make it a difficult reconstruction problem. However, from

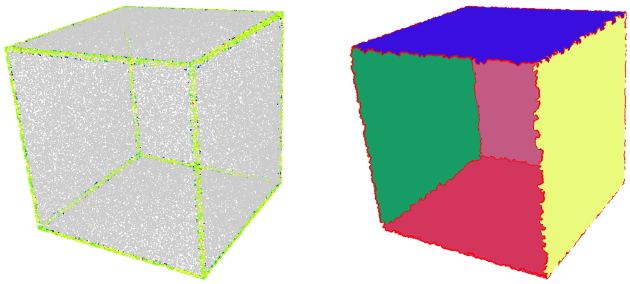
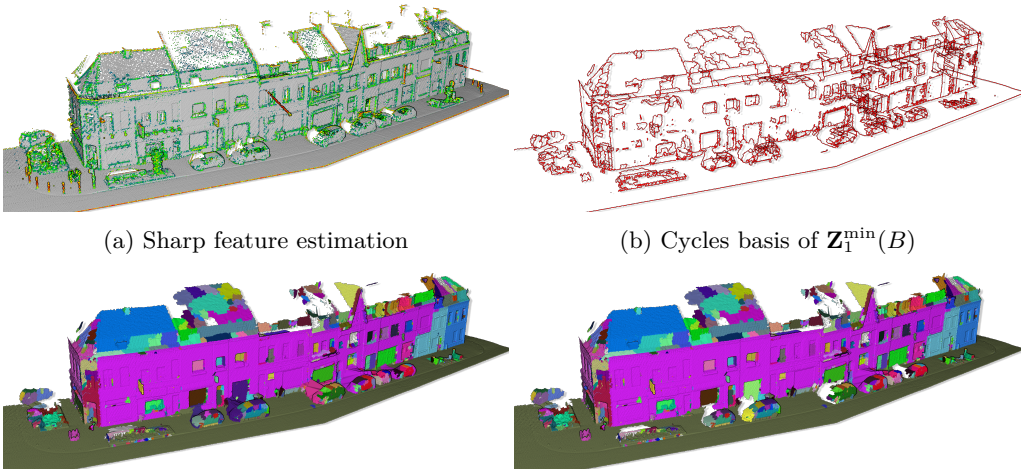


Figure 4.30: The reconstruction using critical basis of a sampled cube (i.e. closed surface) misses one face. Also, an imprecise localization of the sharp edges by the estimator leads to noisy lexicographic optimal 1-cycles.

the VCM estimation, a reasonable set of cycles can be extracted. The critical basis associated to these cycles creates a complete reconstruction of the scene as well as an interesting geometric decomposition: for instance, cars are represented in different elements of the critical basis than the ground, represented in a unique element. This decomposition also allows to either fill or keep the holes corresponding to occluded regions of the input set of points, by removing some of the largest elements of this critical basis.

This process knows a few limitations. In the presence of different densities in the input, using a single threshold for the curvature estimator is not appropriate. This can be seen on the roof part with low density in Figure 4.31, where the threshold leads to a fragmentation of the roof into multiple elements of the critical basis. Also, as shown in Figure 4.30, lexicographic optimal 1-chain are not always particularly well localized in the subcomplex B. Finally, as the homology classes of the critical basis form a basis of $\mathcal{H}_2(K, B)$, any reconstruction of a closed surface with this method will miss one part. We argue however that, in this situation, the first reconstruction method is far better suited to closed surfaces.



(a) Sharp feature estimation

(b) Cycles basis of $\mathbf{Z}_1^{\min}(B)$

(c) Critical basis of $\mathbf{Z}_2^{\min}(K, B)$

(d) Removing the first few chains

Figure 4.31: Example of reconstruction based on the construction of a critical basis. After choosing a threshold for the sharp feature estimation (a) and constructing a ϵ offset complex B ($\epsilon = 0.25$ meters), the basis of $\mathbf{Z}_1^{\min}(B)$ forms a "wireframe" representation of the scene (b). The critical basis correctly reconstructs most part of the scene (c). In this basis, the largest elements often correspond to the perceived holes of the original set of points (d). The input set of points was extracted from the Paris-Lille-3D dataset [RDG18].

Chapter 5

Conclusion and perspectives

Creating models for urban scenes is essential for tasks related to urban planning such as visualization, change tracking and simulation. The most adapted representation of 3-dimensional geometry fluctuates between either parsimonious representations, such as those defined by level of details in Geographic Information Systems, allowing less cluttered visualization, change tracking at the scale of entire cities and fast but coarse simulations, and dense representations, as triangular meshes reconstructed without priors, creating more realistic and precise visualization and allowing precise inspection and simulations at smaller scales. Advancements in acquisitions process, either from photography or laser scanning, allow to generate massive point clouds representing large urban scenes with a precision in the order of centimeters and reconstruct either geometry representations in a semi-automatic fashion.

Convinced that both representations have their own merits and flaws, two methods are presented in this work. The first reconstructs buildings using a parsimonious LOD2 representation with strong priors, in view of integration with GIS applications. The second reconstructs whole urban scenes as dense meshes without priors, focusing on efficiency and generality for large sets of points.

Parsimonious reconstructions The described pipeline for parsimonious LOD2 representations constructs intermediate 2-dimensional partitions. The following two observations have guided our contribution to this reconstruction pipeline:

— in order to correctly detect all relevant structures on noisy inputs, a large number of segments are used to create these 2-dimensional decompositions, often making them overly complex and irregular.

— the kinetic construction of these partitions creates many collinear segments.

The focus of our work is a simplification process of these 2D polygonal partitions which seeks a balance between the fidelity to input partition, the enforcement of canonical relationships between lines and a low complexity output. Because of the collinear segments of the initial partition, the simplification is formulated on the lines of the partition, written in homogeneous coordinates. We associate to each line a metric to measure the displacement from the original position. Applying a gradient descent algorithm to minimize our objective function requires a few precautions, namely a calculation of the gradient that takes into account the defined metrics and dealing with the degeneracy of these metrics. Our simplification alternates between continuous optimization and discrete merges of lines. When lifted in 3D, the simplified partitions allow for scalable generation of 3D models whose quality competes well with those of existing methods.

A major weakness of our simplification method is that the movement of the 2-dimensional partition might degrade the result of the lifted model. This is mainly due to the decorrelation between the formulated fidelity of the 2-dimensional partition and the input point clouds. An improvement of this simplification should consider this 3-dimensional information.

Future work could also explore a more iterative version of this pipeline. As an example, after having performed a simplification of a first partition, running the kinetic framework solely on the segments whose supporting line were not simplified might improve the quality of the partition, essentially filtering the unnecessary segments. Also, lifting the 2-dimensional partition gives a good information of fidelity to the input data, and regions where the fidelity is not sufficient could be located to either detect finer primitives or subdivide the partition locally.

Finally, investigating how our simplification process can be extended efficiently on 3D polyhedral arrangements could in turn lead to more scalable 3D piecewise-planar reconstruction methods.

Dense reconstructions We have proposed several dense reconstruction methods based on lexicographic optimal chains, for closed as well as open surfaces. The definition of lexicographic optimality is based on a total order on simplices which we justify by showing that Delaunay triangulations can be characterized by lexicographic optimal chains along this total order on simplices. We give polynomial-time algorithms allowing to compute lexicographic optimal homologous chains of any dimension of an arbitrary simplicial complex, whereas the more general problem of computing optimal homologous chains (OHCP) is NP-hard. Motivated by applications of reconstruction of 2-surfaces in 3D, we give more efficient algorithms when the simplicial complex is a pseudomanifold and the considered optimal chains are of codimension 1. This involves mainly leveraging the duality of the complex. In fact, the lexicographic optimal cycle problem is quite naturally restated as a lexicographic min-cut on the dual graph. In practice, this problem is used to reconstruct closed surfaces, where the method requires an input tetrahedron, the resulting chain being homologous to the boundary of this input in the simplicial complex where this tetrahedron has been removed. We also design an open surface reconstruction method, which requires to provide a 1-boundary. To derive an efficient algorithm in the dual graph of the complex, the method first requires to compute a "representative" chain, which can be any chain that is bounded by the provided boundary. A second algorithm computes the lexicographic optimal chain homologous to this representative chain. This requires to efficiently verify intersection products between 2-chains of the primal complex and 1-cycles of the dual graph. We introduce a modified version of the disjoint-sets data structure to do exactly that. In applications, when using the Delaunay triangulation as complex, finding the lexicographic optimal chain homologous to the representative chain is equivalent to finding the lexicographic optimal chain bounded by the provided boundary, thanks to the trivial homology of the Delaunay complex. We show this open surface reconstruction is particularly useful in the context of large urban scenes, as it allows to use a tiling meshing strategy for massive data points, where imposing the boundary guarantees correct stitching between meshes on adjacent tiles. Finally, we define a critical basis of the vector space of lexicographic optimal relative 2-chains in a simplicial pair (K, B), which is isomorphic to the homology groups $\mathcal{H}_2(K, B)$. One application for this critical basis constructs a subcomplex B of K from a sharp feature estimator and the critical basis of 2-cycles of this simplicial pair (K, B). We show how this reconstruction is useful for open surfaces where the boundary to reconstruct cannot easily be given manually.

The framework of lexicographic optimal chains offers many interesting continuation to this work. Although not exposed for sake of simplicity, most algorithms can be

explored in a persistent context: Given complexes $B \subset BB \subset K$, we believe algorithms can be derived to compute optimal homologous cycles of (K, B) persistent in (K, BB). The critical basis can also be defined for persistent homology. Whether these algorithms have interesting applications of reconstruction is still unknown at this point.

For closed surfaces, defining as input an interior region can be tedious. Small or under-sampled structures are also sometimes ignored by the method, and specifying additional interior constraints can improve the reconstruction. Because of its proximity with other Delaunay-based interpolatory approaches, finding automatically interior points is an interesting direction for this work, for instance with a classification of Voronoi poles [KSO04].

The second possible control over the method is by modifying the underlying complex used. A logical approach is to consider regular triangulations as we can derive a total order on these simplices and define a similar lexicographic order on chains for this complex. The choice of weights could be used either to guide the method toward a better reconstruction or to simplify the result, as the vertices of regular complexes can be a subset of the provided set of points, i.e. some input points might not appear as vertices of the complex. A few early experiments seem to show that this direction is promising, however the method choosing the weights automatically needs to be explored and justified. Approaches that have considered optimizing weighted triangulations [MMdD11, dGMMD14] could be relevant to this discussion.

Finally, the reconstruction process using critical basis is interesting, especially thanks to the control of which part of the basis we want to keep. We've mentioned limitations on the choice of a fixed threshold for the feature detector as well as the imprecise localization of produced 1-cycles. Further work could consider for instance adapting the threshold of the detector according to the local density of points, or improving the 1-cycles before computing its associated critical basis of 2-chains.

Bibliography

[ABK98] Nina Amenta, Marshall Bern, and Manolis Kamvysselis. A new Voronoi-based surface reconstruction algorithm. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques - SIGGRAPH '98*, pages 415–421, Not Known, 1998. ACM Press. doi:10.1145/280814.280947. (Cited on pages 11 and 95)

[ACDL00] N. Amenta, S. Choi, T. K. Dey, and N. Leekha. A simple algorithm for homeomorphic surface reconstruction. In *Proceedings of the Sixteenth Annual Symposium on Computational Geometry - SCG '00*, pages 213–222, Clear Water Bay, Kowloon, Hong Kong, 2000. ACM Press. doi:10.1145/336154.336207. (Cited on page 11)

[ACK01] Nina Amenta, Sunghee Choi, and Ravi Krishna Kolluri. The power crust. In *Proceedings of the Sixth ACM Symposium on Solid Modeling and Applications*, SMA '01, pages 249–266, New York, NY, USA, May 2001. Association for Computing Machinery. doi:10.1145/376957.376986. (Cited on page 12)

[ACTD07] P. Alliez, D. Cohen-Steiner, Y. Tong, and M. Desbrun. Voronoi-based variational reconstruction of unoriented point sets. In *Proceedings of the Fifth Eurographics Symposium on Geometry Processing*, SGP '07, pages 39–48, Goslar, DEU, July 2007. Eurographics Association. (Cited on page 95)

[ACYD05] Pierre Alliez, David Cohen-Steiner, Mariette Yvinec, and Mathieu Desbrun. Variational tetrahedral meshing. In *ACM SIGGRAPH 2005 Papers on - SIGGRAPH '05*, page 617, Los Angeles, California, 2005. ACM Press. doi:10.1145/1186822.1073238. (Cited on page 52)

[AS17] Radhakrishna Achanta and Sabine Susstrunk. Superpixels and Polygons Using Simple Non-iterative Clustering. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4895–4904, Honolulu, HI, July 2017. IEEE. doi:10.1109/CVPR.2017.520. (Cited on page 15)

[ASG21] Pierre Alliez, Laurent Saboret, and Gaël Guennebaud. Poisson surface reconstruction. In *CGAL User and Reference Manual*. CGAL Editorial Board, 5.2.2 edition, 2021. (Cited on page 90)

[ASS+09] Sameer Agarwal, Noah Snavely, Ian Simon, Steven M Seitz, and Richard Szeliski. Building Rome in a day. In *2009 IEEE 12th International Conference on Computer Vision*, pages 72–79, Kyoto, September 2009. IEEE. doi:10.1109/ICCV.2009.5459148. (Cited on page 7)

[Att98] D. Attali. R-regular shape reconstruction from unorganized points. *Computational Geometry*, 10(4):239–247, July 1998. doi:10.1016/S0925-7721(98)00013-3. (Cited on page 11)

[Att21] Dominique Attali. Reconstructing manifolds by weighted l1-norm minimization. https://dgdvc.sciencesconf.org/349134, 2021. (Cited on page 85)

[BB97] Fausto Bernardini and Chandrajit L Bajaj. Sampling and reconstructing manifolds using alpha-shapes. 1997. (Cited on page 11)

[BBPM08] Mathieu Brédif, Dider Boldo, Marc Pierrot-Deseilligny, and Henri Maître. 3D Building Model Fitting Using A New Kinetic Framework. *arXiv:0805.0648 [cs]*, May 2008. arXiv:0805.0648. (Cited on page 14)

[BC00] Jean-Daniel Boissonnat and Frédéric Cazals. Smooth surface reconstruction via natural neighbour interpolation of distance functions. In *Proceedings of the Sixteenth Annual Symposium on Computational Geometry - SCG '00*, pages 223–232, Clear Water Bay, Kowloon, Hong Kong, 2000. ACM Press. doi:10.1145/336154.336208. (Cited on page 12)

[BCC+12] Oleksiy Busaryev, Sergio Cabello, Chao Chen, Tamal K. Dey, and Yusu Wang. Annotating Simplices with a Homology Basis and Its Applications. In Fedor V. Fomin and Petteri Kaski, editors, *Algorithm Theory – SWAT 2012*, pages 189–200, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. (Cited on page 85)

[BCY18] Jean-Daniel Boissonnat, Frédéric Chazal, and Mariette Yvinec. *Geometric and Topological Inference*. Cambridge University Press, first edition, September 2018. doi:10.1017/9781108297806. (Cited on page 41)

[BdM14] Alexandre Boulch, Martin de La Gorce, and Renaud Marlet. Piecewise-Planar 3D Reconstruction with Edge and Corner Regularization. *Computer Graphics Forum*, 33(5):55–64, August 2014. doi:10.1111/cgf.12431. (Cited on page 15)

[Ber94] Marcel Berger. *Geometry*. Universitext. Springer-Verlag, Berlin ; New York, corr. 2nd print edition, 1994. (Cited on page 23)

[BF04] Jean-Daniel Boissonnat and Julia Flötotto. A coordinate system associated with points scattered on a surface. *Computer-Aided Design*, 36(2):161–174, February 2004. doi:10.1016/S0010-4485(03)00059-9. (Cited on page 17)

[BG14] Jean-Daniel Boissonnat and Arijit Ghosh. Manifold Reconstruction Using Tangential Delaunay Complexes. *Discrete & Computational Geometry*, 51(1):221–267, January 2014. doi:10.1007/s00454-013-9557-2. (Cited on page 17)

[Bie21] Bien'ici. Real estate service. https://www.bienici.com, 2021. (Cited on page 5)

[BKBH07] Matthew Bolitho, Michael Kazhdan, Randal Burns, and Hugues Hoppe. Multilevel streaming for out-of-core surface reconstruction. In *Proceedings of the Fifth Eurographics Symposium on Geometry Processing*, SGP '07, pages 69–78, Goslar, DEU, July 2007. Eurographics Association. (Cited on page 13)

[BKRW17] Ulrich Bauer, Michael Kerber, Jan Reininghaus, and Hubert Wagner. Phat – Persistent Homology Algorithms Toolbox. *Journal of Symbolic Computation*, 78:76–90, January 2017. doi:10.1016/j.jsc.2016.03.008. (Cited on page 95)

[BL18] Jean-Philippe Bauchet and Florent Lafarge. KIPPI: KInetic Polygonal Partitioning of Images. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3146–3154, Salt Lake City, UT, June 2018. IEEE. doi:10.1109/CVPR.2018.00332. (Cited on pages 15 and 19)

[BL19] J.-P. Bauchet and F. Lafarge. City reconstruction from airborne lidar: A computational geometry approach. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, IV-4/W8:19–26, September 2019. doi:10.5194/isprs-annals-IV-4-W8-19-2019. (Cited on page 15)

[BLS16] Filip Biljecki, Hugo Ledoux, and Jantien Stoter. An improved LOD specification for 3D building models. *Computers, Environment and Urban Systems*, 59:25–37, September 2016. doi:10.1016/j.compenvurbsys.2016.04.005. (Cited on page 4)

[BMN20] Glencora Borradaile, William Maxwell, and Amir Nayyeri. Minimum Bounded Chains and Minimum Homologous Chains in Embedded Simplicial Complexes. In Sergio Cabello and Danny Z. Chen, editors, *36th International Symposium on Computational Geometry (SoCG 2020)*, volume 164 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 21:1–21:15, Dagstuhl, Germany, 2020. Schloss Dagstuhl–Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.SoCG.2020.21. (Cited on page 85)

[BMR+99] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, and G. Taubin. The ball-pivoting algorithm for surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics*, 5(4):349–359, October 1999. doi:10.1109/2945.817351. (Cited on page 12)

[Boi84] Jean-Daniel Boissonnat. Geometric structures for three-dimensional shape representation. *ACM Transactions on Graphics*, 3(4):266–286, October 1984. doi:10.1145/357346.357349. (Cited on pages 12 and 17)

[Bou09] Christian Boucheny. Interactive scientific visualization of large datasets (towards a perception-based approach), 2009. (Cited on page 7)

[Bro79] Kevin Q. Brown. Voronoi diagrams from convex hulls. *Information Processing Letters*, 9(5):223–228, December 1979. doi:10.1016/0020-0190(79)90074-7. (Cited on page 46)

[BRVG15] Andras Bodis-Szomoru, Hayko Riemenschneider, and Luc Van Gool. Superpixel meshes for fast edge-preserving surface reconstruction. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2011–2020, Boston, MA, USA, June 2015. IEEE. doi:10.1109/CVPR.2015.7298812. (Cited on page 15)

[BSL+15] Filip Biljecki, Jantien Stoter, Hugo Ledoux, Sisi Zlatanova, and Arzu Çöltekin. Applications of 3D City Models: State of the Art Review. *ISPRS International Journal of Geo-Information*, 4(4):2842–2889, December 2015. doi:10.3390/ijgi4042842. (Cited on pages 4 and 5)

[BTS+14] Matthew Berger, Andrea Tagliasacchi, Lee M. Seversky, Pierre Alliez, Joshua A. Levine, Andrei Sharf, and Claudio T. Silva. State of the Art in Surface Reconstruction from Point Clouds. *Eurographics 2014 - State of the Art Reports*, page 25 pages, 2014. doi:10.2312/EGST.20141040. (Cited on page 11)

[BVZ01] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, Nov./2001. doi:10.1109/34.969114. (Cited on page 21)

[CA15] Daniela Cabiddu and Marco Attene. Large mesh simplification for distributed environments. *Computers & Graphics*, 51:81–89, October 2015. doi:10.1016/j.cag.2015.05.015. (Cited on page 13)

[CAD04] David Cohen-Steiner, Pierre Alliez, and Mathieu Desbrun. Variational shape approximation. *ACM Transactions on Graphics*, 23(3):905–914, August 2004. doi:10.1145/1015706.1015817. (Cited on pages 14, 33, 34, and 37)

[CBC+01] J. C. Carr, R. K. Beatson, J. B. Cherrie, T. J. Mitchell, W. R. Fright, B. C. McCallum, and T. R. Evans. Reconstruction and representation of 3D objects with radial basis functions. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '01, pages 67–76, New York, NY, USA, August 2001. Association for Computing Machinery. doi:10.1145/383259.383266. (Cited on page 13)

[CCM10] Frédéric Chazal, David Cohen-Steiner, and Quentin Mérigot. Boundary Measures for Geometric Inference. *Foundations of Computational Mathematics*, 10(2):221–240, April 2010. doi:10.1007/s10208-009-9056-2. (Cited on page 95)

[CCY04] David Cohen-Steiner, Éric Colin de Verdière, and Mariette Yvinec. Conforming Delaunay triangulations in 3D. *Computational Geometry*, 28(2-3):217–233, June 2004. doi:10.1016/j.comgeo.2004.03.001. (Cited on page 93)

[CD04] David Cohen-Steiner and Frank Da. A greedy Delaunay-based surface reconstruction algorithm. *The Visual Computer*, 20(1):4–16, April 2004. doi:10.1007/s00371-003-0217-z. (Cited on pages 12, 17, and 90)

[CDE+00] Siu-Wing Cheng, Tamal K. Dey, Herbert Edelsbrunner, Michael A. Facello, and Shang-Hua Teng. Silver exudation. *Journal of the ACM*, 47(5):883–904, September 2000. doi:10.1145/355483.355487. (Cited on page 17)

[CEN09] Erin W. Chambers, Jeff Erickson, and Amir Nayyeri. Minimum cuts and shortest homologous cycles. In *Proceedings of the 25th Annual Symposium on Computational Geometry - SCG '09*, page 377, Aarhus, Denmark, 2009. ACM Press. doi:10.1145/1542362.1542426. (Cited on pages 58 and 65)

[CF08] Chao Chen and Daniel Freedman. Quantifying Homology Classes. In Susanne Albers and Pascal Weil, editors, *25th International Symposium on Theoretical Aspects of Computer Science*, volume 1 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 169–180, Dagstuhl, Germany, 2008. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. doi:10.4230/LIPIcs.STACS.2008.1343. (Cited on page 85)

[CF10] Chao Chen and Daniel Freedman. Measuring and computing natural generators for homology groups. *Computational Geometry*, 43(2):169–181, February 2010. doi:10.1016/j.comgeo.2009.06.004. (Cited on page 85)

[CF11] Chao Chen and Daniel Freedman. Hardness Results for Homology Localization. *Discrete & Computational Geometry*, 45(3):425–448, April 2011. doi:10.1007/s00454-010-9322-8. (Cited on pages 58, 65, and 85)

[CG04] Frédéric Cazals and Joachim Giesen. Delaunay Triangulation Based Surface Reconstruction: Ideas and Algorithms. Report, INRIA, November 2004. doi:10/document. (Cited on page 11)

[Cha03] Raphaëlle Chaine. A geometric convection approach of 3-D reconstruction. In *Proceedings of the 2003 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*, SGP '03, pages 218–229, Goslar, DEU, June 2003. Eurographics Association. (Cited on page 12)

[Cit15] City of Helsinki. Solar Energy Potential. https://kartta.hel.fi/3d/solar/, 2015. (Cited on page 6)

[Cit19] City of Paris. Paris land use plan. https://www.paris.fr/pages/le-plan-local-d-urbanisme-plu-2329, 2019. (Cited on page 3)

[CL96] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '96, pages 303–312, New York, NY, USA, August 1996. Association for Computing Machinery. doi:10.1145/237170.237269. (Cited on page 13)

[CLP10] Anne-Laure Chauve, Patrick Labatut, and Jean-Philippe Pons. Robust piecewise-planar 3D reconstruction and completion from large-scale unstructured point data. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1261–1268, San Francisco, CA, USA, June 2010. IEEE. doi:10.1109/CVPR.2010.5539824. (Cited on pages 15, 33, 34, and 37)

[CLV19a] David Cohen-Steiner, André Lieutier, and Julien Vuillamy. Lexicographic optimal chains and manifold triangulations. Research Report, INRIA Sophia-Antipolis Méditerranée, December 2019. (Cited on pages 49, 53, 56, and 90)

[CLV19b] David Cohen-Steiner, André Lieutier, and Julien Vuillamy. Regular triangulations as lexicographic optimal chains. December 2019. (Cited on page 58)

[CV15] Erin Wolf Chambers and Mikael Vejdemo-Johansson. Computing Minimum Area Homologies: Computing Minimum Area Homologies. *Computer Graphics Forum*, 34(6):13–21, September 2015. doi:10.1111/cgf.12514. (Cited on page 85)

[CVKP18] J. Frederico Carvalho, Mikael Vejdemo-Johansson, Danica Kragic, and Florian T. Pokorny. An algorithm for calculating top-dimensional bounding chains. *PeerJ Computer Science*, 4:e153, May 2018. doi:10.7717/peerj-cs.153. (Cited on page 85)

[CX04] Long Chen and Jin-chao Xu. Optimal Delaunay Triangulations. *Journal of Computational Mathematics*, 22(2):299–308, 2004. (Cited on pages 52 and 53)

[DC21] Tran Kai Frank Da and David Cohen-Steiner. Advancing front surface reconstruction. In *CGAL User and Reference Manual*. CGAL Editorial Board, 5.2.2 edition, 2021. (Cited on page 90)

[dCAD11] Fernando de Goes, David Cohen-Steiner, Pierre Alliez, and Mathieu Desbrun. An Optimal Transport Approach to Robust Reconstruction and Simplification of 2D Shapes. *Computer Graphics Forum*, 30(5):1593–1602, August 2011. doi:10.1111/j.1467-8659.2011.02033.x. (Cited on page 14)

[DG03] Tamal K. Dey and Samrat Goswami. Tight cocone: A water-tight surface reconstructor. In *Proceedings of the Eighth ACM Symposium on Solid Modeling and Applications - SM '03*, page 127, Seattle, Washington, USA, 2003. ACM Press. doi:10.1145/781606.781627. (Cited on page 12)

[DG06] Tamal K. Dey and Samrat Goswami. Provable surface reconstruction from noisy samples. *Computational Geometry*, 35(1):124–141, August 2006. doi:10.1016/j.comgeo.2005.10.006. (Cited on page 12)

[dGMMD14] Fernando de Goes, Pooran Memari, Patrick Mullen, and Mathieu Desbrun. Weighted Triangulations for Geometry Processing. *ACM Transactions on Graphics*, 33(3):28:1–28:13, June 2014. doi:10.1145/2602143. (Cited on page 100)

[DHK11] Tamal K. Dey, Anil N. Hirani, and Bala Krishnamoorthy. Optimal Homologous Cycles, Total Unimodularity, and Linear Programming. *SIAM Journal on Computing*, 40(4):1026–1044, January 2011. doi:10.1137/100800245. (Cited on pages 58, 85, and 92)

[DHM20] Tamal K. Dey, Tao Hou, and Sayan Mandal. Computing minimal persistent cycles: Polynomial and hard cases. In *Proceedings of the Thirty-First Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '20, pages 2587–2606, USA, January 2020. Society for Industrial and Applied Mathematics. (Cited on pages 65 and 85)

[DL16] Liuyun Duan and Florent Lafarge. Towards Large-Scale City Reconstruction from Satellites. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, volume 9909, pages 89–104. Springer International Publishing, Cham, 2016. doi:10.1007/978-3-319-46454-1_6. (Cited on page 15)

[DLW18] Tamal K. Dey, Tianqi Li, and Yusu Wang. Efficient algorithms for computing a minimal homology basis. *arXiv:1801.06759 [cs, math]*, January 2018. arXiv:1801.06759. (Cited on page 85)

[DMSL11] Julie Digne, Jean-Michel Morel, Charyar-Mehdi Souzani, and Claire Lartigue. Scale Space Meshing of Raw Data Point Sets. *Computer Graphics Forum*, 30(6):1630–1642, September 2011. doi:10.1111/j.1467-8659.2011.01848.x. (Cited on page 90)

[DSW10] Tamal K. Dey, Jian Sun, and Yusu Wang. Approximating loops in a shortest homology basis from point data. In *Proceedings of the 2010 Annual*

Symposium on Computational Geometry - SoCG '10, page 166, Snowbird, Utah, USA, 2010. ACM Press. doi:10.1145/1810959.1810989. (Cited on page 85)

[Ede03] Herbert Edelsbrunner. Surface Reconstruction by Wrapping Finite Sets in Space. In Boris Aronov, Saugata Basu, János Pach, and Micha Sharir, editors, *Discrete and Computational Geometry*, volume 25, pages 379–404. Springer Berlin Heidelberg, Berlin, Heidelberg, 2003. doi:10.1007/978-3-642-55566-4_17. (Cited on page 12)

[EH10] Herbert Edelsbrunner and J. Harer. *Computational Topology: An Introduction*. American Mathematical Society, Providence, R.I, 2010. (Cited on pages 41, 58, 59, 60, and 75)

[EK72] Jack Edmonds and Richard M. Karp. Theoretical Improvements in Algorithmic Efficiency for Network Flow Problems. *Journal of the ACM*, 19(2):248–264, April 1972. doi:10.1145/321694.321699. (Cited on page 66)

[EM90] Herbert Edelsbrunner and Ernst Peter Mücke. Simulation of simplicity: A technique to cope with degenerate cases in geometric algorithms. *ACM Transactions on Graphics*, 9(1):66–104, January 1990. doi:10.1145/77635.77639. (Cited on page 79)

[Eur20] European Commission. BIM2TWIN: Optimal Construction Management & Production Control. https://cordis.europa.eu/project/id/958398, November 2020. (Cited on page 6)

[EV09] Sander Oude Elberink and George Vosselman. Building Reconstruction by Target Based Graph Matching on Incomplete Laser Data: Analysis and Limitations. *Sensors (Basel, Switzerland)*, 9(8):6101–6118, July 2009. doi:10.3390/s90806101. (Cited on page 16)

[EW05] Jeff Erickson and Kim Whittlesey. Greedy optimal homotopy and homology generators. In *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '05, pages 1038–1046, USA, January 2005. Society for Industrial and Applied Mathematics. (Cited on page 85)

[FH15] Yasutaka Furukawa and Carlos Hernández. Multi-View Stereo: A Tutorial. *Foundations and Trends® in Computer Graphics and Vision*, 9(1-2):1–148, 2015. doi:10.1561/0600000052. (Cited on page 7)

[FLBA20] Jean-Dominique Favreau, Florent Lafarge, Adrien Bousseau, and Alex Auvolat. Extracting Geometric Structures in Images with Delaunay Point Processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(4):837–850, April 2020. doi:10.1109/TPAMI.2018.2890586. (Cited on page 15)

[FS89] M. Fredman and M. Saks. The cell probe complexity of dynamic data structures. In *Proceedings of the Twenty-First Annual ACM Symposium on Theory of Computing - STOC '89*, pages 345–354, Seattle, Washington, United States, 1989. ACM Press. doi:10.1145/73007.73040. (Cited on page 68)

[FTK14] Chen Feng, Yuichi Taguchi, and Vineet R. Kamat. Fast plane extraction in organized point clouds using agglomerative hierarchical clustering. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6218–6225, Hong Kong, China, May 2014. IEEE. doi:10.1109/ICRA.2014.6907776. (Cited on page 19)

[FW16] Wolfgang Förstner and Bernhard P. Wrobel. *Photogrammetric Computer Vision: Statistics, Geometry, Orientation and Reconstruction.* Number 11 in Geometry and Computing. Springer International Publishing : Imprint: Springer, Cham, 1st ed. 2016 edition, 2016. doi:10.1007/978-3-319-11550-4. (Cited on page 25)

[GBK03] Stefan Gumhold, Pavel Borodin, and Reinhard Klein. Intersection free simplification. *International Journal of Shape Modeling*, 09(02):155–176, December 2003. doi:10.1142/S0218654303000097. (Cited on page 13)

[GCV+19] Kyle Genova, Forrester Cole, Daniel Vlasic, Aaron Sarna, William T. Freeman, and Thomas Funkhouser. Learning Shape Templates With Structured Implicit Functions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7154–7164, 2019. (Cited on page 13)

[GF64] Bernard A. Galler and Michael J. Fisher. An improved equivalence algorithm. *Communications of the ACM*, 7(5):301–303, May 1964. doi:10.1145/364099.364331. (Cited on page 68)

[GH97] Michael Garland and Paul S. Heckbert. Surface simplification using quadric error metrics. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques - SIGGRAPH '97*, pages 209–216, Not Known, 1997. ACM Press. doi:10.1145/258734.258849. (Cited on pages 13, 32, 33, 34, and 37)

[GJ+10] Gaël Guennebaud, Benoît Jacob, et al. Eigen v3, 2010. (Cited on page 33)

[GMH+20] Anahid Ghazanfarpour, Nicolas Mellado, Chems E. Himeur, Loïc Barthe, and Jean-Pierre Jessel. Proximity-aware multiple meshes decimation using quadric error metric. *Graphical Models*, 109:101062, May 2020. doi:10.1016/j.gmod.2020.101062. (Cited on page 13)

[Goo21] Google Maps. Online mapping service. https://www.google.com/maps, 2021. (Cited on page 5)

[GS97] T. Gevers and A.W.N. Smeulders. Combining region splitting and edge detection through guided Delaunay image subdivision. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1021–1026, San Juan, Puerto Rico, 1997. IEEE Comput. Soc. doi:10.1109/CVPR.1997.609455. (Cited on page 14)

[GT94] Marc Levoy Greg Turk. The Stanford 3D Scanning Repository. http://graphics.stanford.edu/data/3Dscanrep/, 1994. (Cited on page 89)

[Gui04] Leonidas J. Guibas. Kinetic data structures. In *Handbook of Data Structures and Applications*, 2004. (Cited on pages 15 and 19)

[HB13] Dirk Holz and Sven Behnke. Fast Range Image Segmentation and Smoothing Using Approximate Surface Reconstruction and Region Growing. In

Sukhan Lee, Hyungsuck Cho, Kwang-Joon Yoon, and Jangmyung Lee, editors, *Intelligent Autonomous Systems 12*, volume 194, pages 61–73. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013. doi:10.1007/978-3-642-33932-5_7. (Cited on page 19)

[HBS11] Hai Huang, Claus Brenner, and Monika Sester. 3D building roof reconstruction from point clouds via generative models. In *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems - GIS '11*, page 16, Chicago, Illinois, 2011. ACM Press. doi:10.1145/2093973.2093977. (Cited on page 15)

[HDD+92] Hugues Hoppe, Tony DeRose, Tom Duchamp, John McDonald, and Werner Stuetzle. Surface reconstruction from unorganized points. *ACM SIGGRAPH Computer Graphics*, 26(2):71–78, July 1992. doi:10.1145/142920.134011. (Cited on page 13)

[HGSP13] André Henn, Gerhard Gröger, Viktor Stroh, and Lutz Plümer. Model driven reconstruction of roofs from sparse LIDAR point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, 76:17–29, February 2013. doi:10.1016/j.isprsjprs.2012.11.004. (Cited on page 15)

[HK06] Alexander Hornung and Leif Kobbelt. Robust reconstruction of watertight 3D models from non-uniformly sampled point clouds without normal information. In *Proceedings of the Fourth Eurographics Symposium on Geometry Processing*, SGP '06, pages 41–50, Goslar, DEU, June 2006. Eurographics Association. (Cited on page 12)

[HK12] Martin Habbecke and Leif Kobbelt. Linear Analysis of Nonlinear Constraints for Interactive Geometric Modeling. *Computer Graphics Forum*, 31(2pt3):641–650, May 2012. doi:10.1111/j.1467-8659.2012.03043.x. (Cited on pages 15 and 16)

[HMFB18] Thomas Holzmann, Michael Maurer, Friedrich Fraundorfer, and Horst Bischof. Semantically Aware Urban 3D Reconstruction with Plane-Based Regularization. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision – ECCV 2018*, volume 11218, pages 487–503. Springer International Publishing, Cham, 2018. doi:10.1007/978-3-030-01264-9_29. (Cited on page 14)

[HTM17] Christian Häne, Shubham Tulsiani, and Jitendra Malik. Hierarchical Surface Prediction for 3D Object Reconstruction. *arXiv:1704.00710 [cs]*, November 2017. arXiv:1704.00710. (Cited on page 13)

[HY18] M A Hossain and J K W Yeoh. BIM for existing buildings: Potential opportunities and barriers. *IOP Conference Series: Materials Science and Engineering*, 371:012051, June 2018. doi:10.1088/1757-899x/371/1/012051. (Cited on page 4)

[JM00] Jianbo Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, Aug./2000. doi:10.1109/34.868688. (Cited on page 12)

[JPT21] Clément Jamin, Sylvain Pion, and Monique Teillaud. 3D Triangulations. In *CGAL User and Reference Manual*. CGAL Editorial Board, 5.2.2 edition, 2021. (Cited on page 88)

[Kad07] Martin Kada. Scale-Dependent Simplification of 3D Building Models Based on Cell Decomposition and Primitive Instancing. In Stephan Winter, Matt Duckham, Lars Kulik, and Ben Kuipers, editors, *Spatial Information Theory*, volume 4736, pages 222–237. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007. doi:10.1007/978-3-540-74788-8_14. (Cited on page 14)

[KBH06] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Proceedings of the Fourth Eurographics Symposium on Geometry Processing*, SGP '06, pages 61–70, Goslar, DEU, June 2006. Eurographics Association. (Cited on page 13)

[KH13] Michael Kazhdan and Hugues Hoppe. Screened poisson surface reconstruction. *ACM Transactions on Graphics*, 32(3):1–13, June 2013. doi:10.1145/2487228.2487237. (Cited on pages 13, 33, 37, and 90)

[KPZK17] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics*, 36(4):1–13, July 2017. doi:10.1145/3072959.3073599. (Cited on page 37)

[KSO04] Ravikrishna Kolluri, Jonathan Richard Shewchuk, and James F. O'Brien. Spectral surface reconstruction from noisy point clouds. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing - SGP '04*, page 11, Nice, France, 2004. ACM Press. doi:10.1145/1057432.1057434. (Cited on pages 12 and 100)

[KW18] Koulik Khamaru and Martin Wainwright. Convergence guarantees for a class of non-convex and non-smooth optimization problems. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2601–2610. PMLR, July 2018. (Cited on page 29)

[LAA+17] Debra F. Laefer, Saleh Abuwarda, Anh-Vu Vo, Linh Truong-Hong, and Hamid Gharibi. 2015 Aerial Laser and Photogrammetry Datasets for Dublin, Ireland's City Center, 2017. doi:10.17609/N8MQ0N. (Cited on pages 22, 92, and 94)

[LB07] Victor Lempitsky and Yuri Boykov. Global Optimization for Shape Fitting. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, Minneapolis, MN, USA, June 2007. IEEE. doi:10.1109/CVPR.2007.383293. (Cited on page 12)

[LC87] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3D surface construction algorithm. *ACM SIGGRAPH Computer Graphics*, 21(4):163–169, August 1987. doi:10.1145/37402.37422. (Cited on page 13)

[LCZ+19] Yi-Chun Lin, Yi-Ting Cheng, Tian Zhou, Radhika Ravi, Seyyed Meghdad Hasheminasab, John Evan Flatt, Cary Troy, and Ayman Habib. Evaluation of UAV LiDAR for Mapping Coastal Environments. *Remote Sensing*, 11(24):2893, January 2019. doi:10.3390/rs11242893. (Cited on page 7)

[Lee13] John M. Lee. *Introduction to Smooth Manifolds*. Number 218 in Graduate Texts in Mathematics. Springer, New York ; London, 2nd ed edition, 2013. (Cited on page 28)

[LGL+09] Nan Li, Pengdong Gao, Yongquan Lu, Chu Qiu, Jintao Wang, and Wenhua Yu. Parallel adaptive simplification of massive meshes. In *2009 11th IEEE International Conference on Computer-Aided Design and Computer Graphics*, pages 632–635, August 2009. doi:10.1109/CADCG.2009.5246823. (Cited on page 13)

[Lin00] Peter Lindstrom. Out-of-core simplification of large polygonal models. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques - SIGGRAPH '00*, pages 259–262, Not Known, 2000. ACM Press. doi:10.1145/344779.344912. (Cited on page 13)

[LKBV13] Florent Lafarge, Renaud Keriven, Mathieu Brédif, and Hoang-Hiep Vu. A hybrid multi-view stereo algorithm for modeling urban scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):5, January 2013. doi:10.1109/TPAMI.2012.84. (Cited on page 14)

[LLM20] Muxingzi Li, Florent Lafarge, and Renaud Marlet. Approximating shapes in images with low-complexity polygons. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8630–8638, Seattle, WA, USA, June 2020. IEEE. doi:10.1109/CVPR42600.2020.00866. (Cited on pages 15 and 16)

[LN21] Minglei Li and Liangliang Nan. Feature-preserving 3D mesh simplification for urban buildings. *ISPRS Journal of Photogrammetry and Remote Sensing*, 173:135–150, March 2021. doi:10.1016/j.isprsjprs.2021.01.006. (Cited on page 14)

[Low99] D.G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, pages 1150–1157 vol.2, Kerkyra, Greece, 1999. IEEE. doi:10.1109/ICCV.1999.790410. (Cited on page 7)

[LPK07] Patrick Labatut, Jean-Philippe Pons, and Renaud Keriven. Efficient Multi-View Reconstruction of Large-Scale Scenes using Interest Points, Delaunay Triangulation and Graph Cuts. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8, Rio de Janeiro, Brazil, 2007. IEEE. doi:10.1109/ICCV.2007.4408892. (Cited on pages 12 and 14)

[LPK09] P. Labatut, J.-P. Pons, and R. Keriven. Robust and Efficient Surface Reconstruction From Range Data. *Computer Graphics Forum*, 28(8):2275–2290, December 2009. doi:10.1111/j.1467-8659.2009.01530.x. (Cited on page 12)

[LTB19] Hélène Legrand, Jean-Marc Thiery, and Tamy Boubekeur. Filtered Quadrics for High-Speed Geometry Smoothing and Clustering. *Computer Graphics Forum*, 38(1):663–677, February 2019. doi:10.1111/cgf.13597. (Cited on page 13)

[MBH+95] M. V. Marathe, H. Breu, H. B. Hunt, S. S. Ravi, and D. J. Rosenkrantz. Simple heuristics for unit disk graphs. *Networks*, 25(2):59–68, March 1995. doi:10.1002/net.3230250205. (Cited on page 23)

[MDGD+10] Patrick Mullen, Fernando De Goes, Mathieu Desbrun, David Cohen-Steiner, and Pierre Alliez. Signing the Unsigned: Robust Surface Reconstruction from Raw Pointsets. *Computer Graphics Forum*, 29(5):1733–1741, September 2010. doi:10.1111/j.1467-8659.2010.01782.x. (Cited on page 13)

[MMBM15] Aron Monszpart, Nicolas Mellado, Gabriel J. Brostow, and Niloy J. Mitra. RAPter: Rebuilding man-made scenes with regular arrangements of planes. *ACM Transactions on Graphics*, 34(4):1–12, July 2015. doi:10.1145/2766995. (Cited on page 15)

[MMdD11] Patrick Mullen, Pooran Memari, Fernando de Goes, and Mathieu Desbrun. HOT: Hodge-optimized triangulations. *ACM Transactions on Graphics*, 30(4):1–12, July 2011. doi:10.1145/2010324.1964998. (Cited on page 100)

[MMG01] Michael Murphy, David M. Mount, and Carl W. Gable. A point-placement strategy for conforming Delaunay tetrahedralization. *International Journal of Computational Geometry & Applications*, 11(06):669–682, December 2001. doi:10.1142/S0218195901000699. (Cited on page 93)

[MMP16] C. Mura, O. Mattausch, and R. Pajarola. Piecewise-planar Reconstruction of Multi-room Interiors with Arbitrary Wall Arrangements. *Computer Graphics Forum*, 35(7):179–188, October 2016. (Cited on page 15)

[MOG11] Quentin Mérigot, M Ovsjanikov, and L J Guibas. Voronoi-Based Curvature and Feature Estimation from Point Clouds. *IEEE Transactions on Visualization and Computer Graphics*, 17(6):743–756, June 2011. doi:10.1109/TVCG.2010.261. (Cited on page 93)

[MON+19] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy Networks: Learning 3D Reconstruction in Function Space. *arXiv:1812.03828 [cs]*, April 2019. arXiv:1812.03828. (Cited on page 13)

[MPJ+19] Mateusz Michalkiewicz, Jhony K. Pontes, Dominic Jack, Mahsa Baktashmotlagh, and Anders Eriksson. Deep Level Sets: Implicit Surface Representations for 3D Shape Inference. *arXiv:1901.06802 [cs]*, January 2019. arXiv:1901.06802. (Cited on page 13)

[MPS08] J. Manson, G. Petrova, and S. Schaefer. Streaming surface reconstruction using wavelets. In *Proceedings of the Symposium on Geometry Processing*, SGP '08, pages 1411–1420, Goslar, DEU, July 2008. Eurographics Association. (Cited on page 13)

[Mun18] James R. Munkres. *Elements of Algebraic Topology*. CRC Press, first edition, March 2018. doi:10.1201/9780429493911. (Cited on page 41)

[Mus03] Oleg R. Musin. Construction of the Voronoi Diagram and Secondary Polytope. *arXiv:math/0312052*, December 2003. arXiv:math/0312052. (Cited on page 52)

[MV99] Hans-Gerd Maas and George Vosselman. Two algorithms for extracting building models from raw laser altimetry data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 54(2-3):153–163, July 1999. doi:10.1016/S0924-2716(99)00004-0. (Cited on page 15)

[MWA+13] P. Musialski, P. Wonka, D. G. Aliaga, M. Wimmer, L. van Gool, and W. Purgathofer. A Survey of Urban Reconstruction: A Survey of Urban Reconstruction. *Computer Graphics Forum*, 32(6):146–177, September 2013. doi:10.1111/cgf.12077. (Cited on page 14)

[Nes83] Y. Nesterov. A method for unconstrained convex minimization problem with the rate of convergence o(1/k$\hat{2}$). *undefined*, 1983. (Cited on page 29)

[NW99] Jorge Nocedal and Stephen J. Wright, editors. *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer-Verlag, New York, 1999. doi:10.1007/b98874. (Cited on page 29)

[Orl13] James B. Orlin. Max flows in O(nm) time, or better. In *Proceedings of the 45th Annual ACM Symposium on Symposium on Theory of Computing - STOC '13*, page 765, Palo Alto, California, USA, 2013. ACM Press. doi:10.1145/2488608.2488705. (Cited on pages 65 and 66)

[PFS+19] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation. *arXiv:1901.05103 [cs]*, January 2019. arXiv:1901.05103. (Cited on page 13)

[Qia99] Ning Qian. On the momentum term in gradient descent learning algorithms. *Neural Networks*, 12(1):145–151, January 1999. doi:10.1016/S0893-6080(98)00116-6. (Cited on page 29)

[RBDD18] Simon Rodriguez, Adrien Bousseau, Fredo Durand, and George Drettakis. Exploiting Repetitions for Image-Based Rendering of Facades. *Computer Graphics Forum*, 37(4):119–131, July 2018. doi:10.1111/cgf.13480. (Cited on page 15)

[RDG18] Xavier Roynard, Jean-Emmanuel Deschaud, and François Goulette. Paris-Lille-3D: A large and high-quality ground-truth urban point cloud dataset for automatic segmentation and classification. *The International Journal of Robotics Research*, 37(6):545–557, May 2018. doi:10.1177/0278364918767506. (Cited on page 97)

[RRKB11] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. ORB: An efficient alternative to SIFT or SURF. In *2011 International Conference on Computer Vision*, pages 2564–2571, Barcelona, Spain, November 2011. IEEE. doi:10.1109/ICCV.2011.6126544. (Cited on page 7)

[RS19] Rennes Métropole and Siradel. Ren'ondes. https://ondes.rennesmetropole.fr/, 2019. (Cited on page 6)

[Rup95] J. Ruppert. A Delaunay Refinement Algorithm for Quality 2-Dimensional Mesh Generation. *Journal of Algorithms*, 18(3):548–585, May 1995. doi:10.1006/jagm.1995.1021. (Cited on page 93)

[RvdHV06] T. Rabbani, F. A. van den Heuvel, and G. Vosselman. Segmentation of point clouds using smoothness constraints. In *ISPRS 2006 : Proceedings of the ISPRS Commission V Symposium Vol. 35, Part 6 : Image Engineering and Vision Metrology, Dresden, Germany 25-27 September 2006*, pages 248–253. International Society for Photogrammetry and Remote Sensing (ISPRS), 2006. (Cited on page 19)

[Sen11] Sensefly. Example dataset for mapping. https://www.sensefly.com/education/datasets/, 2011. (Cited on pages 92 and 93)

[SF16] Johannes L. Schonberger and Jan-Michael Frahm. Structure-from-Motion Revisited. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4104–4113, Las Vegas, NV, USA, June 2016. IEEE. doi:10.1109/CVPR.2016.445. (Cited on page 7)

[She98] Jonathan Richard Shewchuk. Tetrahedral mesh generation by Delaunay refinement. In *Proceedings of the Fourteenth Annual Symposium on Computational Geometry - SCG '98*, pages 86–95, Minneapolis, Minnesota, United States, 1998. ACM Press. doi:10.1145/276884.276894. (Cited on page 93)

[Si08] H. Si. Adaptive tetrahedral mesh generation by constrained Delaunay refinement. *International Journal for Numerical Methods in Engineering*, 75(7):856–880, August 2008. doi:10.1002/nme.2318. (Cited on page 93)

[Si15] Hang Si. TetGen, a Delaunay-Based Quality Tetrahedral Mesh Generator. *ACM Transactions on Mathematical Software*, 41(2):1–36, February 2015. doi:10.1145/2629697. (Cited on page 93)

[SJCSO01] Mark St. John, Michael B. Cowen, Harvey S. Smallman, and Heather M. Oonk. The Use of 2D and 3D Displays for Shape-Understanding versus Relative-Position Tasks. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 43(1):79–98, March 2001. doi:10.1518/001872001775992534. (Cited on page 5)

[SLA15] D. Salinas, F. Lafarge, and P. Alliez. Structure-Aware Mesh Decimation: Structure-Aware Mesh Decimation. *Computer Graphics Forum*, 34(6):211–227, September 2015. doi:10.1111/cgf.12531. (Cited on pages 14, 33, 34, and 37)

[SSZC10] Shy Shalom, Ariel Shamir, Hao Zhang, and Daniel Cohen-Or. Cone carving for surface reconstruction. *ACM Transactions on Graphics*, 29(6):1–10, December 2010. doi:10.1145/1882261.1866176. (Cited on page 13)

[Sta12] Standard OGC 12-019. OGC City Geography Markup Language (CityGML) Encoding Standard 2.0.0, 2012. (Cited on pages 3 and 4)

[Sul90] John Matthew Sullivan. *A Crystalline Approximation Theorem for Hypersurfaces*. PhD Thesis, Princeton Univ., October 1990. (Cited on page 65)

[SZBH15] Sajad Shiravi, Ming Zhong, Seyed Ahad Beykaei, and Faranak Hosseini. Estimation of Base-Year Floor Space Data: Comparison of Census-Based and Lidar–Geographic Information System Approaches. *Transportation Research Record: Journal of the Transportation Research Board*, 2494(1):21–28, January 2015. doi:10.3141/2494-03. (Cited on page 6)

[THA+10] Pingbo Tang, Daniel Huber, Burcu Akinci, Robert Lipman, and Alan Lytle. Automatic reconstruction of as-built building information models from laser-scanned point clouds: A review of related techniques. *Automation in Construction*, 19(7):829–843, November 2010. doi:10.1016/j.autcon.2010.06.007. (Cited on page 4)

[TMHF00] Bill Triggs, Philip F. McLauchlan, Richard I. Hartley, and Andrew W. Fitzgibbon. Bundle Adjustment — A Modern Synthesis. In Gerhard Goos, Juris Hartmanis, Jan van Leeuwen, Bill Triggs, Andrew Zisserman, and Richard Szeliski, editors, *Vision Algorithms: Theory and Practice*, volume

1883, pages 298–372. Springer Berlin Heidelberg, Berlin, Heidelberg, 2000. doi:10.1007/3-540-44480-7_21. (Cited on page 7)

[Tv84] Robert E. Tarjan and Jan van Leeuwen. Worst-case Analysis of Set Union Algorithms. *Journal of the ACM*, 31(2):245–281, March 1984. doi:10.1145/62.2160. (Cited on pages 68 and 70)

[ULW19] Kaleem Ullah, Irene Lill, and Emlyn Witt. An Overview of BIM Adoption in the Construction Industry: Benefits and Barriers. In Irene Lill and Emlyn Witt, editors, *Emerald Reach Proceedings Series*, pages 297–303. Emerald Publishing Limited, May 2019. doi:10.1108/S2516-285320190000002052. (Cited on page 4)

[van21] Thijs van Lankveld. Scale-space surface reconstruction. In *CGAL User and Reference Manual*. CGAL Editorial Board, 5.2.2 edition, 2021. (Cited on page 90)

[VKH06] V. Verma, R. Kumar, and S. Hsu. 3D Building Detection and Modeling from Aerial LIDAR Data. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2 (CVPR'06)*, volume 2, pages 2213–2220, New York, NY, USA, 2006. IEEE. doi:10.1109/CVPR.2006.12. (Cited on page 15)

[VL12] Yannick Verdié and Florent Lafarge. Efficient Monte Carlo Sampler for Detecting Parametric Objects in Large Scenes. In David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell, Moni Naor, Oscar Nierstrasz, C. Pandu Rangan, Bernhard Steffen, Madhu Sudan, Demetri Terzopoulos, Doug Tygar, Moshe Y. Vardi, Gerhard Weikum, Andrew Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi Sato, and Cordelia Schmid, editors, *Computer Vision – ECCV 2012*, volume 7574, pages 539–552. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. doi:10.1007/978-3-642-33712-3_39. (Cited on page 15)

[VLA15] Yannick Verdie, Florent Lafarge, and Pierre Alliez. LOD Generation for Urban Scenes. *ACM Transactions on Graphics*, 34(3):1–14, May 2015. doi:10.1145/2732527. (Cited on page 14)

[WCW+17] Pengxiang Wu, Chao Chen, Yusu Wang, Shaoting Zhang, Changhe Yuan, Zhen Qian, Dimitris Metaxas, and Leon Axel. Optimal Topological Cycles and Their Application in Cardiac Trabeculae Restoration. In Marc Niethammer, Martin Styner, Stephen Aylward, Hongtu Zhu, Ipek Oguz, Pew-Thian Yap, and Dinggang Shen, editors, *Information Processing in Medical Imaging*, pages 80–92, Cham, 2017. Springer International Publishing. (Cited on page 85)

[XAAH13] Xuehan Xiong, Antonio Adan, Burcu Akinci, and Daniel Huber. Automatic creation of semantically rich 3D building models from laser scanner data. *Automation in Construction*, 31:325–337, May 2013. doi:10.1016/j.autcon.2012.10.006. (Cited on page 4)

[XOEV14] B. Xiong, S. Oude Elberink, and G. Vosselman. A graph edit dictionary for correcting errors in roof topology graphs reconstructed from point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, 93:227–242, July 2014. doi:10.1016/j.isprsjprs.2014.01.007. (Cited on page 16)

[ZBKB08] Lukas Zebedin, Joachim Bauer, Konrad Karner, and Horst Bischof. Fusion of Feature- and Area-Based Information for Urban Buildings Modeling from Aerial Imagery. In *Computer Vision – ECCV 2008*, pages 873–886. Springer, Berlin, Heidelberg, October 2008. doi:10.1007/978-3-540-88693-8_64. (Cited on pages 15 and 16)

[ZDJF14] Enliang Zheng, Enrique Dunn, Vladimir Jojic, and Jan-Michael Frahm. PatchMatch Based Joint View Selection and Depthmap Estimation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1510–1517, Columbus, OH, USA, June 2014. IEEE. doi:10.1109/CVPR.2014.196. (Cited on page 7)

[ZN08] Qian-Yi Zhou and Ulrich Neumann. Fast and extensible building modeling from airborne LiDAR data. In *Proceedings of the 16th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS '08, pages 1–8, New York, NY, USA, November 2008. Association for Computing Machinery. doi:10.1145/1463434.1463444. (Cited on pages 15 and 16)

[ZN10] Qian-Yi Zhou and Ulrich Neumann. 2.5D Dual Contouring: A Robust Approach to Creating Building Models from Aerial LiDAR Point Clouds. In *Computer Vision – ECCV 2010*, pages 115–128. Springer, Berlin, Heidelberg, September 2010. doi:10.1007/978-3-642-15558-1_9. (Cited on pages 15 and 16)

[ZWF18] Huayi Zeng, Jiaye Wu, and Yasutaka Furukawa. Neural Procedural Reconstruction for Residential Buildings. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 737–753, 2018. (Cited on page 15)

Appendices

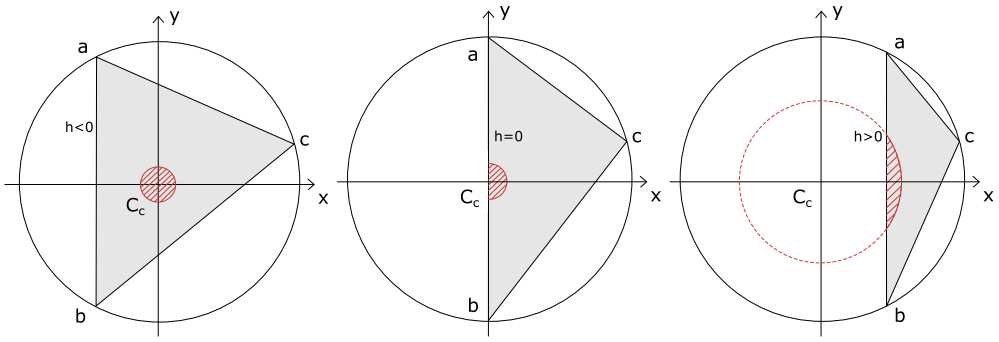


Figure 1: Illustration of the three nature of triangles

A.1 Asymptotical behaviours for 2-simplices (Proof of Lemma 4.13)

We recall the definition of the weight associated to any triangle $\sigma = \{P_0, P_1, P_2\}$ given by Equation (4.8):

$$w_p(\sigma)^p = \int_{|\sigma|} f_\sigma(x)^p dx$$

where, for a point $x = \sum_i \lambda_i P_i$ of $|\sigma|$ written in barycentric coordinates $(\lambda_i)_{i=0,1,2}$ (i.e. verifying $\sum_i \lambda_i = 1$ and $\forall 0 \leq i \leq 2,\ 0 \leq \lambda_i \leq 1$,),

$$f_\sigma(x) = \sum_{i=0}^{2} \lambda_i \left\|P_i\right\|^2 - \left\|x\right\|^2$$

Applying a uniform scale λ to a triangle σ multiplies by λ^2 both its area and the function f_σ. Therefore, the quantity $w_p(\sigma)^p$ is multiplied by λ^{2+2p}. Therefore, without loss of generality, we consider now and for the rest of the proof a triangle abc with ab as its longest edge and verifying:

$$\mathrm{R_B}(abc) = 1$$

For any triangle σ similar to abc, the final asymptotical behaviour of the triangle σ will then be multiplied by $\mathrm{R_B}(\sigma)^{2+2p}$ compared to behaviour associated to abc.

Observe the function f_σ is invariant by translation and rotation around the origin. We introduce the 2-dimensional Cartesian coordinate system of the supporting plane Π_{abc} of abc, centered at the circumcenter $C_C(abc)$ of abc and verifying:

$$\begin{cases} x_a = x_b \\ x_c > x_a \end{cases}$$

We call h the x-coordinate of a and b. This coordinate system is illustrated in Figure 1 according to the nature of the triangle abc. It is particularly helpful in the study of $w_p(abc)^p$ as the function f_{abc} in this referential verifying:

$$w_p(\sigma)^p = \int_{(x,y)\in|abc|} f_{abc}(x,y)^p dxdy$$

is given by the following equation:

$$f_{abc}(x,y) = \mathrm{R_C}(abc)^2 - x^2 - y^2$$

.A.1. Asymptotical behaviours for 2-simplices (Proof of Lemma 4.13)

The proof will require to show that an asymptotical behaviour of $w_p(abc)^p$ can be found in a small region of $|abc|$ close to the circumcenter: this is the goal of the following definition and lemma.

Definition .1 (Upper set measure D_ϕ)**.** *Consider a compact set $\mathcal{D} \subset \mathbb{R}^d$. For a continuous function $\phi : \mathcal{D} \to [0,1]$ we denote by D_ϕ the* upper set measure *of ϕ, the map $D_\phi : [0,1] \to \mathbb{R}^+$ defined as:*

$$D_\phi(t) \underset{def.}{=} \mu_L\left(\{u \in \mathcal{D}, \phi(u) \geq 1 - t\}\right)$$

where μ_L denotes the Lebesgues measure.

From this definition, the upper set measure $t \mapsto D_\phi(t)$ is non decreasing. Recall that, by Rademacher Theorem, Lipschitz function are differentiable almost everywhere.

Lemma .2. *Consider a compact set $\mathcal{D} \subset \mathbb{R}^d$ and a continuous function $\phi : \mathcal{D} \to [0,1]$. Assumes the upper set measure D_ϕ of ϕ is Lipschitz and denote by D'_ϕ its derivative defined almost everywhere. Then one has:*

$$\int_{\mathcal{D}} \phi(u)^p = \int_0^1 D'_\phi(t)(1-t)^p dt$$

Moreover, if there is $\beta > 0$ such that D'_ϕ is defined and continuous on $(0,\beta)$ with:

$$\forall t \in (0,\beta), D'_\phi(t) > 0$$

then, for any given $0 < \alpha \leq 1$:

$$\int_{\mathcal{D}} \phi(u)^p \underset{p\to\infty}{\sim} \int_0^\alpha D'_\phi(t)(1-t)^p dt$$

Proof. On the interval $[0,1]$, the map $s \mapsto s^p$ is p-Lipschitz (its derivative is upper bounded by p), therefore:

$$\forall a, b \in [0,1], |a^p - b^p| \leq p|b-a|$$

In particular, for a positive integer k:

$$\forall a \in [0,1], \left| a^p - \left(\frac{1}{k}\lfloor ka \rfloor\right)^p \right| \leq \frac{p}{k}$$

For a compact set $\mathcal{D} \subset \mathbb{R}^d$ and a continuous function $\phi : \mathcal{D} \to [0,1]$ one has therefore:

$$\left| \int_{\mathcal{D}} \phi(u)^p - \int_{\mathcal{D}} \left(\frac{1}{k}\lfloor k\phi(u) \rfloor\right)^p \right| \leq \frac{p}{k}\mu_L(\mathcal{D})$$

Since:

$$\int_{\mathcal{D}} \left(\frac{1}{k}\lfloor k\phi(u) \rfloor\right)^p = \sum_{i=0}^{k-1} \left(D_\phi\left(\frac{i+1}{k}\right) - D_\phi\left(\frac{i}{k}\right)\right)\left(1 - \frac{i}{k}\right)^p$$

Using the Lebesgue integrability of ϕ and the Riemann integrability of $t \mapsto p(1 -$

$t)^{p-1}D_\phi(t)$, we get:

$$\begin{aligned}
\int_{\mathcal{D}} \phi(u)^p &= \lim_{k\to\infty} \sum_{i=0}^{k-1} \left(D_\phi\left(\frac{i+1}{k}\right) - D_\phi\left(\frac{i}{k}\right)\right)\left(1-\frac{i}{k}\right)^p \\
&= \lim_{k\to\infty} \sum_{i=1}^{k-1} \left(\left(1-\frac{i-1}{k}\right)^p - \left(1-\frac{i}{k}\right)^p \right) D_\phi\left(\frac{i}{k}\right) \\
&= \int_0^1 -D_\phi(t)p(1-t)^{p-1}dt \\
&= \left[-D_\phi(t)(1-t)^p\right]_0^1 + \int_0^1 D'_\phi(t)(1-t)^p dt \\
&= \int_0^1 D'_\phi(t)(1-t)^p dt \qquad (1)
\end{aligned}$$

For the second part of the lemma, one has:

$$\frac{\int_{\mathcal{D}} \phi(u)^p}{\int_0^\alpha D'_\phi(t)(1-t)^p dt} = \frac{\int_0^1 D'_\phi(t)(1-t)^p dt}{\int_0^\alpha D'_\phi(t)(1-t)^p dt} = 1 + \frac{\int_\alpha^1 D'_\phi(t)(1-t)^p dt}{\int_0^\alpha D'_\phi(t)(1-t)^p dt}$$

Since D_ϕ is increasing and K-Lipschitz for some constant K, one has $D'_\phi(t) \in [0, K]$ whenever it is defined which gives:

$$\int_\alpha^1 D'_\phi(t)(1-t)^p dt \leq K(1-\alpha)^p$$

and:

$$\int_0^\alpha D'_\phi(t)(1-t)^p dt \geq \int_0^{\alpha/2} D'_\phi(t)(1-t)^p dt \geq A(1-\alpha/2)^p$$

with:

$$A = \int_0^{\alpha/2} D'_\phi(t)dt$$

Since there is $\beta > 0$ such that D'_ϕ is defined and continuous on $(0,\beta)$ with $\forall t \in (0,\beta), D'_\phi(t) > 0$ we have $A > 0$ and:

$$\left| \frac{\int_{\mathcal{D}} \phi(u)^p}{\int_0^\alpha D'_\phi(t)(1-t)^p dt} - 1 \right| \leq \frac{K(1-\alpha)^p}{A(1-\alpha/2)^p} = \frac{K}{A}\left(\frac{1-\alpha}{1-\alpha/2}\right)^p$$

□

When looking at the upper measure of f_{abc}, $D_{f_{abc}}$ is Lipschitz with constant π since for $0 \leq t < t+u \leq 1$:

$$0 \leq D_{f_{abc}}(t+u) - D_{f_{abc}}(t) \leq \mu_L\left((x,y) \in \mathbb{R}^2, h^2 + t \leq x^2 + y^2 \leq h^2 + t + u\}\right) \leq \pi u$$

We now study each case illustrated in Figure 1 separately, depending on the nature of the triangle *abc*.

abc is strictly acute We have that $\mathrm{R_C}(abc) = \mathrm{R_B}(abc) = 1$ and $h < 0$. Also, $(0,0)$ is in the interior of *abc* and therefore, there is $\alpha > 0$ small enough such that:

$$\forall t \in [0,\alpha],\ f_{abc}^{-1}([1-t,1]) = \{(x,y) \in \mathbb{R}^2, x^2 + y^2 \leq t\}$$

which gives for the upper set measure $D_{f_{abc}}$:

$$\forall t \in [0, \alpha], D_{f_{abc}}(t) = \pi t$$

and then, if $D'_{f_{abc}}$ denotes the derivative of $D_{f_{abc}}$:

$$\forall t \in [0, \alpha], D'_{f_{abc}}(t) = \pi$$

By applying Lemma .2, one gets the following equivalence:

$$\int_{|abc|} f_{abc}(u)^p \underset{p\to\infty}{\sim} \int_0^\alpha \pi(1-t)^p dt = \frac{1-(1-\alpha)^p}{p+1}$$

Finally,

$$w_p(abc)^p \underset{p\to\infty}{\sim} \frac{\pi}{p}$$

abc is right Very similarly than the strictly acute case, one now has $\mathrm{R_C}(abc) = \mathrm{R_B}(abc) = 1$ and $h = 0$. There is $\alpha > 0$ small enough such that:

$$\forall t \in [0, \alpha], f_{abc}^{-1}([1-t, 1]) = \{(x, y) \in \mathbb{R}^2, x^2 + y^2 \leq t \text{ and } x \geq 0\}$$

which gives for the upper set measure $D_{f_{abc}}$:

$$\forall t \in [0, \alpha], D_{f_{abc}}(t) = \frac{\pi}{2} t$$

and therefore:

$$\forall t \in [0, \alpha], D'_{f_{abc}}(t) = \frac{\pi}{2}$$

Applying again Lemma .2 we get

$$w_p(abc)^p \underset{p\to\infty}{\sim} \frac{\pi}{2p}$$

abc is strictly obtuse One now has $h > 0$, $\mathrm{R_C}(abc) > 1$ and $h^2 = \mathrm{R_C}(abc)^2 - 1$. There is $\alpha > 0$ small enough such that :

$$\forall t \in [0, \alpha], f_{abc}^{-1}([1-t, 1]) = \{(x, y), x^2 + y^2 \leq h^2 + t \text{ and } x \geq h\}$$

which gives:

$$\forall t \in [0, \alpha], D_{f_{abc}}(t) = (h^2 + t)\left(\frac{\pi}{2} - \arcsin\sqrt{\frac{h^2}{h^2+t}}\right) - \sqrt{h^2 t}$$

and then:

$$\forall t \in [0, \alpha], D'_{f_{abc}}(t) = \arccos\sqrt{\frac{h^2}{h^2+t}} \tag{2}$$

This gives from Lemma .2:

$$w_p(abc)^p \underset{p\to\infty}{\sim} \int_0^\alpha (1-t)^p \arccos\sqrt{\frac{h^2}{h^2+t}} \tag{3}$$

Using a change of variable $u = \frac{\sqrt{t}}{h}$, $t = h^2u^2$, $dt = 2h^2u\,du$ and $\arccos\sqrt{\frac{h^2}{h^2+t}} = \arctan(u)$ we get:

$$\int_0^\alpha (1-t)^p \arccos\sqrt{\frac{h^2}{h^2+t}} dt = 2h^2 \int_0^{\sqrt{\alpha}/h} (1-h^2u^2)^p u \arctan(u)\, du \tag{4}$$

.A.1. Asymptotical behaviours for 2-simplices (Proof of Lemma 4.13)

Since $\lim_{u\to 0} \frac{\arctan(u)}{u} = 1$ and, for positive u, $\arctan(u) < u$, one has:

$$\forall \epsilon > 0,\ \exists \alpha' > 0,\ u \in (0, \alpha') \Rightarrow (1-\epsilon)u \leq \arctan(u) \leq u$$

So that, for a given h and arbitrary small $\epsilon > 0$, there is $\alpha > 0$ such that:

$$\begin{aligned}(1-\epsilon)\int_0^{\sqrt{\alpha}/h} u^2(1-h^2u^2)^p\,du &\leq \int_0^{\sqrt{\alpha}/h} (1-h^2u^2)^p u\arctan(u)\,du \\ &\leq \int_0^{\sqrt{\alpha}/h} u^2(1-h^2u^2)^p\,du \end{aligned} \tag{5}$$

Notice that:

$$u^2(1-h^2u^2)^p = u^2\exp\left(p\log(1-h^2u^2)\right)$$

and, using the first order Taylor expansion of $t \to \log(1-t)$ and since the second derivative of log is negative at 1, for $\alpha > 0$ small enough, there is a constant $C > 0$ such that, when $u \in (0, \sqrt{\alpha}/h)$, one has:

$$-pC(h^2u^2)^2 - ph^2u^2 \leq p\log(1-h^2u^2) \leq -ph^2u^2$$

we get, then, when $u \in (0, \sqrt{\alpha}/h)$:

$$\exp\left(-pCh^4u^4\right)\exp\left(-ph^2u^2\right) \leq (1-h^2u^2)^p \leq \exp\left(-ph^2u^2\right) \tag{6}$$

The second inequality in Equation (6) gives, for $\alpha > 0$ small enough:

$$\int_0^{\sqrt{\alpha}/h} u^2(1-h^2u^2)^p\,du \leq \int_0^{\sqrt{\alpha}/h} u^2\exp\left(-ph^2u^2\right)\,du \tag{7}$$

One can check, by derivation of the right hand term, that, for $\beta > 0$:

$$\int_0^{\beta} u^2\exp\left(-ph^2u^2\right)\,du = \frac{\sqrt{\pi}\,\mathrm{erf}(h\sqrt{p}\beta) - 2h\sqrt{p}\beta\exp\left(-ph^2\beta^2\right)}{4h^3p^{\frac{3}{2}}} \tag{8}$$

where the so called "error function" erf is the normalized primitive of the Gaussian:

$$\mathrm{erf}(z) \underset{\text{def.}}{=} \frac{2}{\sqrt{\pi}}\int_0^z \exp(-t^2)dt$$

and the normalization gives that $\lim_{z\to\infty}\mathrm{erf}(z) = 1$. Observe that Equation (8) gives, for any p and $\beta > 0$:

$$\int_0^{\beta} u^2\exp\left(-ph^2u^2\right)\,du \leq \frac{\sqrt{\pi}}{4h^3p^{\frac{3}{2}}}$$

This, with Equation (7), gives that, for fixed α small enough:

$$\lim_{p\to\infty} \frac{\int_0^{\sqrt{\alpha}/h} u^2(1-h^2u^2)^p\,du}{\frac{\sqrt{\pi}}{4h^3p^{\frac{3}{2}}}} \leq 1 \tag{9}$$

We consider now the lower bound given by the first inequality in Equation (6). First observes that, for p large enough, one has $p^{-1/3} < \sqrt{\alpha}/h$, and since the integrand is non negative, one has:

$$\lim_{p\to\infty} \frac{\int_0^{\sqrt{\alpha}/h} u^2\exp\left(-pCh^4u^4\right)\exp\left(-ph^2u^2\right)\,du}{\int_0^{p^{-1/3}} u^2\exp\left(-pCh^4u^4\right)\exp\left(-ph^2u^2\right)\,du} \geq 1$$

Since $u \mapsto \exp\left(pCh^4u^4\right)$ is increasing for $u > 0$, this gives:

$$\lim_{p\to\infty} \frac{\int_0^{\sqrt{\alpha}/h} u^2 \exp\left(-pCh^4u^4\right) \exp\left(-ph^2u^2\right) du}{\exp\left(-pCh^4(p^{-1/3})^4\right) \int_0^{p^{-1/3}} u^2 \exp\left(-ph^2u^2\right) du} \geq 1$$

Since $\lim_{p\to\infty} \exp\left(pCh^4(p^{-1/3})^4\right) = \lim_{p\to\infty} \exp\left(p^{-1/3}Ch^4\right) = 1$, we get:

$$\lim_{p\to\infty} \frac{\int_0^{\sqrt{\alpha}/h} u^2 \exp\left(-pCh^4u^4\right) \exp\left(-ph^2u^2\right) du}{\int_0^{p^{-1/3}} u^2 \exp\left(-ph^2u^2\right) du} \geq 1 \tag{10}$$

Taking $\beta = p^{-1/3}$ in Equation (8) we get:

$$\lim_{p\to\infty} \frac{\int_0^{p^{-1/3}} u^2 \exp\left(-ph^2u^2\right) du}{\frac{\sqrt{\pi}}{4h^3p^{\frac{3}{2}}}} = 1$$

This, with Equation (10) and the first inequality in Equation (6), we get:

$$\lim_{p\to\infty} \frac{\int_0^{\sqrt{\alpha}/h} u^2(1-h^2u^2)^p \, du}{\frac{\sqrt{\pi}}{4h^3p^{\frac{3}{2}}}} \geq 1$$

This with Equation (9) gives, for any $\alpha > 0$ small enough

$$\lim_{p\to\infty} \frac{\int_0^{\sqrt{\alpha}/h} u^2(1-h^2u^2)^p \, du}{\frac{\sqrt{\pi}}{4h^3p^{\frac{3}{2}}}} = 1 \tag{11}$$

Since Equation (5) holds for any $\epsilon > 0$, and since Equation (11) holds for arbitrary small α, we get:

$$\lim_{p\to\infty} \frac{\int_0^{\sqrt{\alpha}/h} (1-h^2u^2)^p u \arctan(u) \, du}{\frac{\sqrt{\pi}}{4h^3p^{\frac{3}{2}}}} = 1$$

and, with Equation (4) and Equation (3), we finally get:

$$w_p(\sigma)^p \underset{p\to\infty}{\sim} \sqrt{\frac{\pi}{4h^2}\frac{1}{p^3}}$$