Structured Learning of Geospatial Data

Loic Landrieu

HABILITATION À DIRIGER DES RECHERCHES DE L'UNIVERSITÉ GUSTAVE EIFFEL

Spécialité: Sciences et Technologies de l'Information Géographique

Loïc LANDRIEU

LIGM, Ecole des Ponts, CNRS, Univ. Gustave Eiffel
LASTIG, IGN/ENSG

STRUCTURED LEARNING OF GEOSPATIAL DATA

Defended on May 9 2023 in front of a jury composed of:

Reviewers :	
Pierre ALLIEZ	Inria Sophia-Antipolis (TITANE)
Nicolas COURTY	Univ. Bretagne Sud (IRISA/Obelix)
Maks OVSJANIKOV	Ecole Polytechnique (GeoVIC/LIX)
Examinators :	
Camille COUPRIE	Meta (FAIR)
Christian HEIPKE	Univ. of Hannover (IPG)
Konrad SCHINDLER	ETH Zürich (IGP)
Martin BURGER	Univ. of Erlangen (Dep. of Mathematics)
Carola-Bibiane SCHÖNLIEB	Univ. of Cambridge (DAMTP, CIA)

Acknowledgments

I can't thank my students enough for their hard work, enthusiasm, and creativity. It has been a privilege to guide such motivated future researchers through their first steps in academia.

I am thankful to the LASTIG lab, IGN, and the ISPRS for making me feel welcome in their community. In particular, I appreciated the trust and continuous support of Clément Mallet, Mathieu Brédif, Bruno Vallet, and Nicolas Paparoditis.

Lastly, I want to thank Alix for her world-class editing skills and for making me take a much-needed break once in a while.

Preamble

I started my academic career with a Ph.D. in so-called "pure applied maths" [Loi16], working on functional optimization[RL15, LO16] and probabilistic graphical models [LO14]. While I immensely enjoyed the powerful tools I got to study and later develop, I also regretted not tackling concrete issues. I decided that after my Ph.D., I would start with real problems and let their characteristics and constraints guide my research. Circumstances and opportunities led me to study geospatial machine learning at IGN, and I could not have asked for a more exciting, impactful, and fertile playing field.

The message of this manuscript is simple: geospatial data has distinctive characteristics leading to unique machine-learning challenges and impactful applications. I draw two conclusions from this observation: (i) the analysis of geospatial data should rely on bespoke methods to leverage its unique structure; (ii) geospatial tasks are great candidates for evaluating and motivating new machine learning approaches. This habilitation showcases various methods involving abstract mathematical reasoning, sensor-specific considerations, and everything in between. The common motto is always to leverage the specificities of the considered problem into the architecture of the algorithm for added precision, speed, and parsimony.

In the title of this manuscript, I use the phrase "structured learning", a reference to "structured optimization". This sub-field of functional optimization consists in designing efficient algorithms that exploit the structure of the functional to minimize. Although most of the work here is more applied than pure mathematical optimization, I have applied this principle throughout all my research.

The term "geospatial data" refers to acquisitions with a large-scale spatial structure, without necessarily referencing their absolute position on Earth. This encompasses remote sensing data, such as satellite or aerial acquisitions, but also 3D scans or images taken from large scenes.

Abstract

Abstract

This manuscript presents an overview of my work in the field of geospatial machine learning, a rapidly growing interdisciplinary field that poses many methodological challenges and has a wide range of impactful applications. Throughout my research, I have focused on developing bespoke approaches that leverage the unique properties of geospatial data to create more efficient, precise, and parsimonious models.

This manuscript is divided into four main chapters, each covering a different property of geospatial data structures that can be leveraged algorithmically. The first chapter presents a versatile mathematical framework formalizing the concept of spatial regularity with graphs. We propose an efficient algorithm that tackles a broad family of spatial problems and provides novel convergence guarantees and significant speed-ups compared to generic approaches.

The second chapter introduces a deep learning method that extends the idea of exploiting graph regularity to the case of massive 3D point clouds. We simplify the task of large-scale semantic segmentation by formulating it as as a small graph labelling problem. Our compact models reach high precision at a fraction of the computational cost of other approaches.

In the third chapter, we present a collection of methods designed to take advantage of the data structure inherited from 3D sensors. By considering the sensors' structure, we develop powerful networks with state-of-the-art accuracy, latency, and robustness for various applications and data types.

The last chapter dives into the real-life challenge of automated satellite time series analysis for crop mapping. Recognizing the difference between such data and standard formats used in computer vision, we propose novel and streamlined architectures that achieve unprecedented precision while remaining efficient and economical in memory and preprocessing. We also introduce the task of panoptic segmentation for satellite time series and an efficient architecture to solve this problem at scale.

In summary, this manuscript argues that geospatial problems represent a challenging and impactful venue for evaluating the newest machine learning and vision methods and a fertile source of inspiration for designing novel approaches.

An extended abstract in French is available at the end of the manuscript.

Contents

CHAPTER 1

Introduction

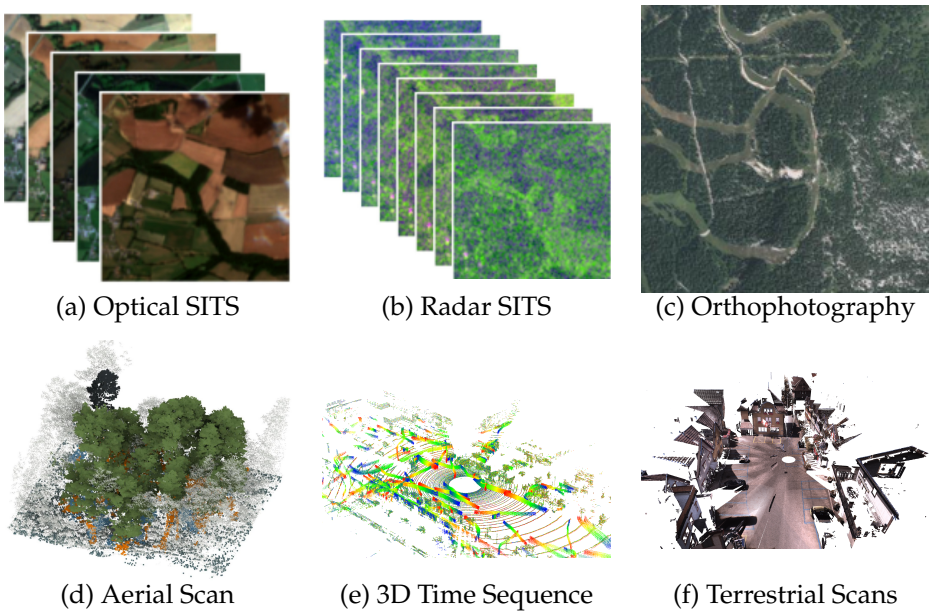
Geospatial data analysis has obvious assets as a scientific field: virtually unlimited data, abundant annotations, challenging structure, unparalleled scale, and exciting applications. This combination makes it a promising and exciting field for evaluating and designing novel machine learning algorithms. This chapter details the specificities of geospatial data and presents some of its most motivating applications. We then argue for the benefits of using dedicated architectures, and present an overview of how the works in this manuscript leverage the unique structures of geospatial data.

1 The Unique Structure of Geospatial Data

Geospatial data refers to data augmented with their location on Earth. By extension, we use this term to designate data with a large-scale spatial extent without necessarily referring to an absolute positioning system. Such data are typically collected with various remote sensing sensors and exhibit particular properties not commonly encountered in natural images and videos. In particular, the diversity of complex sensors requires special considerations. Geospatial problems are typically large-scale, with entangled spatial, temporal, and spectral dimensions. Lastly, the absolute temporal and spatial frame of reference is a crucial characteristic that can be exploited for added precision and speed.

Multi-Sensor. As for photography, geospatial data inherits the structure of the sensors used for acquisition. While intrinsic and extrinsic parameters are often sufficient to characterize the acquisition of a natural image, remote sensing sensors require specific considerations. Geospatial data can be collected by various sensors: active or passive, mobile or static, terrestrial, aerial or spatial, within or beyond the visible light spectrum. In this work, we focus mainly on two types of data:

- *LiDAR Scans.* This active sensor uses lasers whose return times can be used to infer the location of reflective surfaces. LiDARs can be mounted on a fixed frame [HSL+17, ASZ+16], a mobile platform [GLSU13, LAL22], manned or unmanned aircrafts [KLMC22a, VAG20, IGN], and even satellite [DBG+20]. The high precision and



(a) Optical SITS (b) Radar SITS (c) Orthophotography

(d) Aerial Scan (e) 3D Time Sequence (f) Terrestrial Scans

Figure 1.1: Example of geospatial data types considered in this manuscript.

low latency of LiDARs have contributed to their wide adoption for industrial applications. Spinning LiDAR, in particular, have a complex acquisition geometry in which the sensor spins while being mounted on a mobile platform in a dynamic environment.

- *Satellite Image Time Series (SITS).* Satellites offer a privileged perspective on our planet in terms of extent and viewpoints. Some satellites provide optical information across a broad spectrum of frequencies thanks to sophisticated sensors [DDBC+12], while others use radar waves to gather rich information about objects reflecting specific bandwidths [PRG+16]. With carefully planned orbits, the satellites fly over certain areas at a periodical rate of a few days or less, producing temporal sequences of images.

Large-Scale. Earth Observation (EO) is one of the primary sources of geospatial data. Global coverage is standard for satellite [DDBC+12, PRG+16, DBG+20], and country-scale acquisition campaigns are frequent for aerial imagery [IGN]. Geospatial data analysis tasks often require processing data at a near-global scale, *e.g.* biomass estimation for carbon capture assessment [LKA+22] or agricultural monitoring for worldwide food security [MRB+19, PI18].

Geospatial processes display influences at different spatial and temporal scales simultaneously: the crop cultivated in a field is influenced by the parcel's slopes, but also its bioclimatic regions; the fields' appearance dur-

ing an acquisition will be influenced by the position of the sun and the atmospheric conditions [RMM20], but also the cumulative weather patterns of the last few months [NPA22] and local agricultural practices. Some processes require considering large areas simultaneously: some urban land use classes can only be decided based on their position within complex urban configurations. The extent of EO is typically arbitrary and only depends on the memory and computation available. Consequently, modelling large-scale geospatial processes necessitates efficient and parsimonious methods. In contrast, computer vision considers images with a fixed and predictable extent and prioritizes precision over speed and scalability.

Geospatial processes also exhibit multiple and intricate data shifts. We observe a combination of *prior-shift* (more vines in Italy than Norway), *concept-shift* (roofs look different in the South or North of France), and *covariate-shift* (meteorological conditions vary year to year) [KL18].

Multidimensional. The influence of the spatial dimensions on geospatial processes is evident; "*everything is related to everything else. But near things are more related than distant things*" is often referred to as the first law of geography [Tob70]. However, the temporal and spectral dimensions also play a crucial role in remote sensing data analysis.

The multi-temporality of geospatial data can be a consequence of the orbit of satellites or the very nature of the sensor, such as spinning LiDARs. For many applications, the temporal dimension of the acquisitions is crucial for differentiating different classes, such as tree species or crop types [GLGC19], which may appear similar at specific dates and only differs in their evolution. For other applications, time is a structuring constraint, *e.g.* real-time 3D perception for autonomous driving [LAL22].

Satellite images often have a greater spectral extent than natural images. This goes from a fourth *near-infrared* channel or the 13 bands of Sentinel-2, to hyperspectral imagery, which uses hundreds [MPH+22]. This information proves decisive when analyzing materials with specific spectral responses, such as vegetation [CRC05, TDH+13], minerals [CS95], water resources [GCB07], and anthropogenic structures [WHL08].

Geo-referenced. A key difference between natural images and geospatial data is the absolute spatio-temporal coordinates of the latter. One of video analysis' most challenging aspects is tracking the objects of interest across frames. The positions of static geospatial objects such as roads or agricultural parcels are given in an absolute referential (typically WGS84 [SM98], or Lambert93 [Tho52]), greatly simplifying the co-registration between acquisitions taken at different dates or with different sensors.

The spatio-temporal coordinates are also structuring information. Knowing that a video frame is the 13-th of the sequence does not give much

information about its content nor impacts its interpretation. In contrast, the time of year when a satellite image is taken drastically influences its analysis, and the exact time of day also impacts the radiometry through the sun's position. Weather patterns also significantly affect geospatial processes such as plant growth [NPA22], and dictate cloud cover, which completely blocks optical observations. Likewise, the exact coordinate of a pixel in an image is usually not informative. In contrast, the absolute position of a geospatial acquisition may strongly influence its interpretation due to spatial domain shift.

Richly Annotated. Language models have access to a near-infinite corpus by training on the entire Internet with efficient self-supervised tasks [RWC+19, GBB+20]. Many aligned text-image pairs [RKH+21] are also available online as captioned images. However, direct annotations of text and images often require costly and time-consuming human intervention.

Geospatial data benefit from extensively available annotations. Indeed, Geographical Information Systems (GIS) [Goo10, Cha08] are widely used by public and private actors alike for numerous tasks such as urban and terrain modelling [For08], driving guidance systems [LWZ20], or econometrics [BJL+02]. Thanks to increasingly open public data policies [geo], crowd-sourced initiatives [Map], and academic efforts [GL21b, LAL22, GLSU13, SBK21], the amount of open-access annotated geospatial data has starkly increased. For example, countries such as France and Denmark release yearly the cultivated crops in each agricultural parcel of the country [RPG], amounting to millions of labels every year. Many governments publish publicly the cadastre (extent of properties) of their entire territory [BDT]. **The wide availability of annotated geospatial data results in a uniquely favourable setting for developing and evaluating machine learning algorithms.**

Despite a surface-level resemblance, the structure of geospatial data differs vastly from natural images and videos. In particular, the data is gathered with a variety of active or passive sensors with unique properties. The data also displays a complex multidimensional structure with entangled spatial, temporal, and spectral dimensions. The scale of data is also more extensive than typically encountered in computer vision, requiring scalable algorithms for training and inference. However, remote sensing data benefits from absolute spatial and temporal referencing, making tracking more accessible and conditioning information extraction. Contrary to many machine learning applications, remote sensing data can be paired with a large amount of open-access annotations at a low cost.

2 Impactful Applications of Geospatial Analysis

After years of unbridled enthusiasm from the industry as well as the general public, some voices have called the true potential of computer vision for real-life applications to be over-hyped [Man18], unsustainable [TGLM21], and even unethical [VN20]. Such declarations can appear inflammatory and provocative. Notwithstanding, we argue that automated geospatial analysis assuredly leads to impactful and beneficial applications. Below is a non-exhaustive list of applications that motivated this manuscript's work.

Crop Mapping. The ability to classify and predict the yield of crops at a large scale is a crucial part of the growing concern for global food security [MRB+19, PI18]. Through its Common Agricultural Policy, Europe distributes over 57B euros each year in subsidies [CAP] (25B dollars in the US). The fair allocation of this assistance implies at least partial automation, as France alone counts close to 10 million individual parcels [RPG]. Moreover, crop type classification can also ensure that the best agricultural practices are employed to preserve soil [AAM+11, Bul92] and to maximize yield without relying on pesticides [KKN+15, SL08].

Forest Inventory. Automated vegetation analysis from aerial observation is an essential step for many forestry applications [BLBK17], such as forest management [JSM01], biomass estimation [FSM+16, Lu06], or forest fire modelling and prevention [MR11, Mac96, SOC01]. In turn, this allows for further ecological studies of the forest biome, such as estimating habitat suitability [HDW+06, MVG+09] or deriving biodiversity indicators [BD05, Nag01, IK98].

LiDAR-equipped satellite [DBG+20] offers the perspective of global monitoring of forest resources [LKA+22]. However, aerial or space-borne remote sensing of natural forests is often limited to canopy observation and misses its multilayer structure of [KLMC22a, FMJ+15]. Furthermore, annotations often require costly in-situ observations [KLMC22b].

Land Use/Cover Mapping. Thanks to aerial/spatial points of view, remote sensing allows states to oversee their territory's land usage extensively. This has proven especially important to monitor the increase in impervious surface [AJG96, SBT+05] and its harmful environmental effects. Automated and large-scale land-use classification is also helpful in monitoring urban sprawl [HL03, VOOR19] and is a step towards sustainable urban development [TL16].

Another use of automated remote sensing analysis is the possibility of emergency mapping [BGT15] to better facilitate disaster response and

damage assessment, for floods [ASM+22], earthquakes [DG12, DBC+11], tsunamis [KMMB20], or forest fires [CC89], and even conflicts [LS19].

Digital Twins. Extensive acquisition capabilities combined with efficient processing of 3D data allow for the perspective of digitizing entire cities [Bat18, KNL+20] as well as extensive industrial facilities [NSGW21, KFC+19]. Both prospects represent significant environmental and economic stakes. For example, digital cities allow to better design and monitor urban ecological corridors [PZL17]. Modelling of industrial infrastructures can increase productivity [VBR+17, MLL+19] or facilitate their decommissioning [PTB16].

Autonomous Driving. Autonomous driving is predicted to become a norm-shattering reality in the near future. Nevertheless, there are still several significant hurdles to overcome for large-scale deployment, not only regarding social, ethical, and engineering issues but also in terms of machine learning, computer vision, and remote sensing research. As the debate rages on over whether or not cameras are the only necessary sensor for fully autonomous vehicles, several major players have opted for a highly diversified array of sensors on their prototypes: thermal cameras, radar, LiDAR in various configurations, ultrasounds sensors, audio sensors, and so on [Zoo, GKM+20, Val, Lyf]. This diversity and complementary of sensors, combined with the entangled spatial and temporal dimensions of the acquisitions, make some aspects of this problem a flagship remote sensing / geospatial application.

Cultural Heritage / Archaeology. Remote sensing technologies have been used successfully for monitoring heritage sites and discovering hidden archaeological sites for several decades [LWG+19]. Aerial LiDAR has allowed archaeologists to find remnants of forgotten civilizations hidden under heavy forest cover in the Amazon [PBI+22] and the jungles of Angkor [EFP+13]. LiDAR scanning has also been used to save endangered artifacts [Ico, LL19] or help their protection [PBM+18], and more generally, backing up our shared cultural and ecological heritage [FLE+22].

3 Exploiting the Structure of Geospatial Data

Due to their unique properties, geospatial problems present unique challenges, making them a compelling opportunity for evaluating and designing new approaches in machine learning and computer vision.

3.1 Geospatial Analysis as an Evaluation

Machine learning methods are typically evaluated on problems taken from vision, language, or medicine. Geospatial tasks constitute an excellent standard for measuring models' performance, scalability, and expressiveness. We provide here some pointers on how to increase the appeal of geospatial problems as a standard evaluation task for learning algorithms.

Accessible Datasets: Accessibility and ease of use are central to the success of image datasets: images are easy to open and manipulate with most modern programming frameworks. This is far from the case for geospatial data, which tends to use dedicated formats which are typically harder to manipulate, *e.g.* .tiff for georeferenced images or .las for 3D scans. Using generic formats such as .npy or .h5 format can significantly lower the barrier of entry for researchers interested in applying their method to geospatial data. Datasets should also, of course, be freely distributed and accessible. Platforms like Zenodo offer free hosting with unlimited downloads for academic datasets.

Cleaned and Focused Problems: Large-scale spatial data exhibits a variety of complex domain shifts. For example, the composition of acquisitions at different dates, times, or days, translates into radiometric discrepancies. Rare classes can only appear in select regions, complicating the creation of meaningful train/validation/test sets. The class nomenclatures useful for practitioners can often be problematic for learning because of their ambiguity, excessively rare classes, or significant inter-class similarities.

When constructing datasets to assess the performance of learning algorithms, it is essential to curate the data to insulate the precise task evaluated. Indeed, spatial domain shifts or rare classes may bias the metrics in favour of specific architectures for reasons unrelated to the evaluated tasks. Adaptive sampling or reduced nomenclature can help mitigate these issues. Keeping adequate metadata can allow specific evaluation settings focused on the messiness and complexity of real-world data.

Geospatial benchmarks need to consider the spatial auto-correlation of spatial processes in their train/test split. There should be a clear demarcation and buffer between these parts of the dataset, as failing to do so may overtly favour large models overfitting the training set. However, unless the goal is to evaluate low-shot learning or domain adaptation, the data distribution of the training set should be relatively similar to the test sets. It is a tricky balance to find, and these questions should be attentively considered when designing geospatial benchmarks.

Reproducible Code: The geospatial community should systematize the release of open-access code for every publication. This will encourage authors to compare their work with recent baselines and allow for more thorough studies of the impact of a paper's contributions. The (almost) systematic release of research code played a significant part in computer vision's rapid evolution and success. In contrast, many geospatial articles still present methods evaluated on closed datasets without meaningful comparison to the state-of-the-art, open-access implementation or enough detail to re-implement them. Regardless of the brilliance of the ideas they explore, such papers have nearly no scientific value.

3.2 The Need for Dedicated Architectures

Remote sensing as a field shares the same goal as computer vision: analyze an object or environment from information gathered by a sensor situated at a distance. As detailed in Section 1, geospatial data and remote sensing differ from natural images in several key ways. However, a substantial part of geospatial analysis research directly applies computer vision methods to remote sensing data methods with little adaptation, limiting their efficiency [RCVS+19, TRW+21]. Developing algorithms and architectures that exploit the uniqueness of geospatial data leads to several benefits, which we list below.

Adaptive. Geospatial processes are typically large-scale and require networks with large receptive fields to capture. Likewise, as remote sensing sensors are typically further away than cameras, they have lower resolution and density, necessitating adapted processing. Computer vision focuses on analyzing objects or indoor scenes from dense acquisitions, which does not benefit from modelling long-range interactions to the same extent. For example, the nadir or near-nadir point of view of aerial scans differs from acquisitions at ground level. This results in distinct sampling density and a unique perspective on the world. Terrestrial LiDAR in urban areas captures building facades with centimetric precision, whose geometry is well characterized by grid-based approaches [CGS19] or dense convolutions. On the contrary, aerial scans have a density of only a handful (10-20) echos per square meter and are mostly limited to roofs, resulting in significant gaps between related entities. This motivates the use of networks that are robust to stark changes in density and able to connect distant entities.

Efficient. Computer vision favours large models with a high-level understanding of natural images. Remote sensing tasks are typically more grounded, less abstract, and require smaller models. Furthermore, the size of remote sensing acquisition favours algorithms that can scale to large in-

puts. Using gigantic pretrained computer vision models such as Foundation Models [BHA+21] may give satisfying results out of the box, but incurs superfluous computation and have unnecessarily high hardware requirements. Streamlining architectures to the bare essential results in faster and more data-efficient approaches, at virtually no cost in precision.

Multimodal. While multimodality is explored in computer vision, it is an essential aspect of geospatial data frequently captured by many different sensors with complementary characteristics. Furthermore, georeferencing leads to a natural alignment across modalities, which is often difficult or impossible with natural images. While this alignment facilitates the fusion of various sources of information, this remains a nontrivial task due to their differing nature and resolution.

Attractive. The use of unaltered vision models for geospatial tasks may wrongly lead students and researchers outside of the community to believe that geospatial analysis is purely an applied field without methodological challenges. On the contrary, the successful use of novel and dedicated architectures for spatial problems promotes the field as a vector of innovation. This is a key endeavour to present geospatial tasks as a challenging and impactful source of evaluation and inspiration for novel methods.

4 Organization of the Manuscript

We present approaches to leveraging geospatial data's structure into more efficient, parsimonious, and precise algorithms. The works are organized into chapters covering different characteristics of geospatial data that can be exploited.

Chapter 2: Exploiting Graph Regularity. We first present an abstract formalization of spatial regularity into a generic graph framework. We introduce the cut pursuit algorithm to minimize a large family of functionals commonly encountered in geospatial learning and other applications. We present a unified analysis for several versions of the method, including strong and unique theoretical guarantees and speed improvements of several orders of magnitude. This chapter contains this manuscript's most theoretical work, which will serve as the basis or inspiration for much of the work presented thereafter.

Chapter 3: Exploiting Spatial Regularity of 3D Data. This chapter presents a deep learning implementation of the ideas of the last chapter

for the particular case of large-scale 3D data analysis. By computing adaptive partitions of point clouds or 3D space, we transform complex problems of semantic segmentation or surface reconstruction into small-scale graph analysis problems. This approach allows us to reach state-of-the-art results with compact and efficient methods.

Chapter 4: Exploiting Sensor Structure of 3D Data. This chapter describes three distinct approaches that exploit the specific structure of 3D acquisition sensors. Our methods apply to LiDAR time sequences, joint image and point cloud scans, and large point clouds in the wild. Our approaches result in significant improvements in terms of precision, latency, and model size for semantic segmentation and surface reconstruction tasks.

Chapter 5: Exploiting the Structure of Satellite Time Series. This final chapter presents our work on satellite image time series (SITS) for automated crop mapping. By identifying the profound differences between SITS and videos, we design new dedicated approaches to exploit their specificities, which results in significant gains in precision and efficiency. We also present the first state-of-the-art for SITS panoptic segmentation and a unique dedicated dataset.

CHAPTER 2

Exploiting Graph Regularity

Many geospatial analysis problems can be formulated as the minimization of a functional defined on a graph and whose solutions are spatially regular. We first formalize this property as a form of graph-structured sparsity, then introduce the cut pursuit algorithm, which exploits this property for computational efficiency. Our approach can be adapted to functionals with various degrees of smoothness and continuity, covering a wide array of spatially structured problems such as inverse brain imaging or large-scale surface reconstruction. When applied to problems whose solution is spatially regular, the cut pursuit algorithm provides a considerable acceleration compared to other widely used optimization algorithms. In some settings, we also prove unique convergence guarantees that do not require smoothness or convexity of the minimized functional.
This chapter is organized around the following publications:

† **[LO17]** Loic Landrieu, Guillaume Obozinski, "Cut pursuit: Fast Algorithms to Learn Piecewise Constant Functions on General Weighted Graphs", *SIAM Journal of Imaging Science*, 2017
[RL18] Raguet Hugo, Loic Landrieu, "Cut-Pursuit Algorithm for Regularizing Nonsmooth Functionals with Graph Total Variation", *ICML*, 2018
[RL19] Raguet Hugo, Loic Landrieu, "Parallel Cut Pursuit For Minimization of the Graph Total Variation", *ICML Workshop on Graph Reasonning*, 2019
[GLCV19] Stéphane Guinard, Loic Landrieu, Laurent Caraffa, Bruno Vallet, "Piecewise-planar Approximation Of Large 3D Data As Graph-Structured Optimization", *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, 2019

† Work presented in the PhD thesis of the defendant [Loi16].

1 The Cut Pursuit Algorithm

Graphs are commonly used to describe the structure of spatial data, such as point clouds or vector maps, by encoding the adjacency between individual *elements*. As we will exemplify in our numerical experiments, many real-life geospatial problems can be expressed as the minimization of a well-chosen functional whose variables are associated with the vertices of such a graph. We can express the spatial regularity of the minimizers as a form of graph-structured sparsity. We first propose formalizing these concepts mathematically and then introducing an efficient optimization algorithm capable of leveraging this graph sparsity.

We define an unoriented graph $G = (V, E, w)$ with edge weights $w \in \mathbf{R}_+^E$. We consider a functional $F : \Omega^V \mapsto \mathbf{R}$ whose input variable x can be decomposed with respect to the vertices of G: each x_v belongs to the set Ω. We seek $x^\star$ a minimizer of F: [1]

$$x^\star \underset{x \in \Omega^V}{\text{ minimizes }} F(x) \,. \tag{2.1}$$

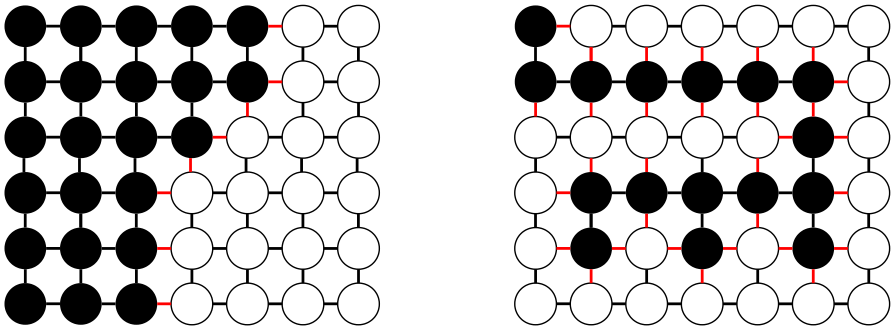
We are interested in functionals F whose minimizers $x^\star$ are constant with respect to a partition of V into a small number of components *w.r.t* G. This property corresponds to a regularity prior on spatial processes and can be exploited algorithmically for faster solving.

1.1 Graph-Structured Sparsity

We define two notions of regularity, which are related but not equivalent.

Graph-Coarseness. Mumford and Shah describe an image as *simple* if it is piecewise smooth, *i.e.* can be decomposed into a small number of regions with short contours and smooth variations [MS89]. This chapter focuses on a stricter form of simplicity: the signal is constant within each region. We can translate this concept for a signal x defined on the vertices of a graph G: x is simple if it is constant within the regions of a partition $\mathcal{V}$ of V with much fewer connected components than vertices, *i.e.* $|\mathcal{V}| \ll |V|$. This property is referred to as *graph-coarseness* and is a widely used prior for natural images [ROF92] and medical imagery [EZE07] images. It is also commonly encountered when the vertices of V correspond to elements positioned in space, such as pixels or 3D points. Since the concept of connectedness in a graph is inherently discontinuous, directly encouraging graph-coarseness makes the optimization problem significantly more complicated.

[1] Depending on its properties, *minimizing* F can mean finding a global or local optimum, a critical point, or simply trying to achieve low values of F.



(a) Gradient has 89% zeros. (b) Gradient has 58% zeros.

Figure 2.1: **Graph-Sparsity and Graph-Coarseness.** Example of two vertex-valued signals which are both graph-coarse: they only have two constant connected components characterized by white or black vertices. We represent the non-zeros values of the graph-gradient as red edges. The signal in (a) is also graph-sparse, while the signal in (b) is not.

Graph-Sparsity. We qualify x as *graph-sparse* if most vertices linked by an edge share the same value: $|\{(u,v) \in E \mid x_u \neq x_v\}| \ll |E|$. When Ω is equipped with a vector structure, this property translates into the sparsity of the graph-gradient, *i.e.* $\{x_u - x_v \mid (u,v) \in E\}$ is mostly zero. This corresponds to a form of structured sparsity as described by Bach *et. al* [BJMO12], and hints at the possibility of more parsimonious and efficient solving methods.

Encouraging Coarseness. As illustrated in Figure 2.1, graph-sparsity implies graph-coarseness. Indeed, if G itself is connected, the number of constant connected components of x is bounded by the number of non-zero values of its graph-gradient plus one. However, the converse is false: a graph-coarse signal is not necessarily graph-sparse.

Motivated by this observation, a straightforward way to impose a coarseness prior to a graph signal x is to use a sparsity-inducing regularizer on the graph gradient. This observation leads to the following family of functionals, first characterized by Geman and Reynolds [GR92]:

$$F(x) := f(x) + \sum_{(u,v)\in E} w_{u,v} h(x_u, x_v)\,, \tag{2.2}$$

with $f : \Omega^V \mapsto \mathbf{R}$ and $h : \Omega \mapsto \mathbf{R}$ a function that reaches its minimum only when $x_u = x_v$. More precisely, we are interested in functions h that encourage strict equality between x_u and x_v. [2] Consequently, we can expect

[2] When Ω is continuous and equipped with a vector structure, we typically have $h(x_u, x_v) = \rho(x_u - x_v)$ with ρ a sparsity-inducing function, *e.g.* a non-smooth norm [BJMO11a].

a minimizer $x^\star$ of such functional F to be graph-sparse and hence graph-coarse. In particular, with $h(x_u, x_v) = \|x_u - x_v\|$ we obtain the well-studied regularity-inducing graph total variation (TV).

The functional optimization literature explores many approaches to exploit the sparsity of the solution of some large-scale problems with fast and parsimonious algorithms [BJMO11b]. Analogously, we propose to exploit the graph-coarseness of functional minimizers of the form Equation 2.2 with a working set strategy. [3]

1.2 Principle of Cut Pursuit

The cut pursuit algorithm maintains a current partition $\mathcal{V}$ of V and alternates between two steps: the *reduction step* and the *refinement step*, see Figure 2.2 The reduction step finds minimizers of F that are also piecewise constant *w.r.t* $\mathcal{V}$. The refinement step splits $\mathcal{V}$ into a finer partition. Both steps can be performed efficiently by exploiting the structure of their corresponding optimization problems.

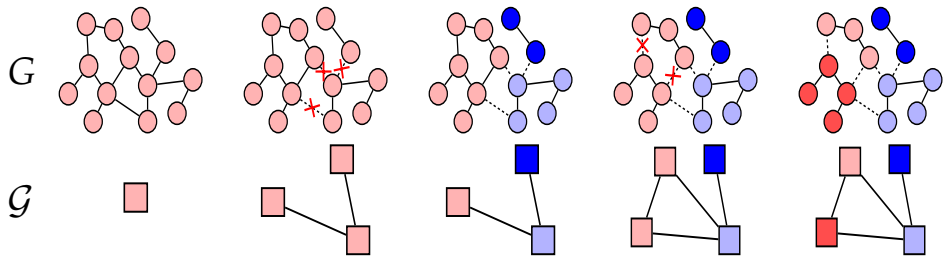


Figure 2.2: **Steps of the Cut Pursuit Algorithm.** At first, the vertices of the graph G are combined into a single component, defining a single-vertex reduced graph $\mathcal{G}$, and the resulting reduced problem is solved (a). The partition is then iteratively refined (b,d) and the corresponding reduced functionals minimized (c,e). The reduction steps are performed *w.r.t* the smaller reduced graph $\mathcal{G}$, while the refinement step involves the full graph G. The vertex colours represent their values, from blue to red.

Problem Reduction. From a partition $\mathcal{V}$, we can define a *reduced graph* $\mathcal{G} = (\mathcal{V}, \mathcal{E}, W)$ whose vertices are the components of $\mathcal{V}$ itself, the edges

[3] By analogy with the literature on sparse optimization, we distinguish between *working set* and *active set* algorithms. The former, also called *column generation algorithms*, maintains a set of constraints that can be active or not, while the latter considers the exact set of active constraints.

denote the adjacent components, and W their weights:

$$\mathcal{E} := \{(U, U') \in \mathcal{V}^2 \mid (U \times U') \cap E \neq \varnothing\} \tag{2.3}$$

$$W_{U,U'} := \sum_{(u,v)\in U\times U'} w_{u,v} \,. \tag{2.4}$$

We can use this reduced graph to advantageously express the target functional F for iterates x that are piecewise constant *w.r.t* $\mathcal{V}$. We first notice that such x can be written as $\mathbb{1}(r, \mathcal{V})$ with $r \in \Omega^{\mathcal{V}}$ such that:

$$[\mathbb{1}(r, \mathcal{V})]_v = \begin{cases} r_U & \text{if } v \in U \\ 0 & \text{otherwise.} \end{cases} \tag{2.5}$$

In other words, a signal that is piecewise constant *w.r.t* $\mathcal{V}$ is completely characterized by only $|\mathcal{V}|$ values. In this case, we call r the *reduced variable*, indicating the value of x for each component of $\mathcal{V}$.

Without loss of generality, we can assume that $h(x, x) = 0$ in Eq. (2.2). In this case, $h(x_u, x_v)$ is zero for all edges (u, v) linking vertices in the same constant component. Consequently, the regularization cancels out inside constant components. We introduce the *reduced functional* $F^{(\mathcal{V})} : \Omega^{\mathcal{V}} \mapsto \mathbf{R}$, defined as the restriction of F to variables that are piecewise constant *w.r.t* $\mathcal{V}$, and which can be written as such:

$$F^{(\mathcal{V})}(r) := F(\mathbb{1}(r, \mathcal{V})) = f(\mathbb{1}(r, \mathcal{V})) + \sum_{(U,U')\in\mathcal{E}} W_{U,U'} h(r_U, r_{U'}) \,. \tag{2.6}$$

We remark that $F^{(\mathcal{V})}$ has a very similar structure to the target functional F and inherits its regularity. Furthermore, $F^{(\mathcal{V})}$ is defined *w.r.t* the reduced graph $\mathcal{G}$, which is much smaller than G as long as $\mathcal{V}$ is coarse. Under this hypothesis, minimizing $F^{(\mathcal{V})}$ is significantly easier than minimizing F. If we had access to a constant partition of the sought solution $x^\star$, solving Eq. (2.6) would also solve the full problem defined in Eq. (2.1), but at a drastically reduced cost. The next step consists in approximating this partition.

Refining $\mathcal{V}$. While the reduction step allows us to solve a simplified version of the problem, the resulting minimizer is only as good as the current partition $\mathcal{V}$. The goal of the refinement step is to add new degrees of freedom to Eq. (2.6) so that its minimization yields a lower value for F. In practice, we seek a *refining partition* $\mathcal{V}'$ that can be combined with $\mathcal{V}$ to obtain a new finer partition through cross-partitioning:

$$\text{refine}(\mathcal{V}, \mathcal{V}') := \{U \cap U' \mid U \in \mathcal{V}, U' \in \mathcal{V}'\} \,. \tag{2.7}$$

In cases where F is not convex, it can be useful to define a more complex refining operator $\text{refine}(\mathcal{V}, \mathcal{V}')$ to mitigate the influence of suboptimal early decisions [SIBD11], see Section 3.3.

We consider a class of graph-functionals whose minimizers are piecewise constant with respect to a coarse partition and hence only defined by a few values. The cut pursuit algorithm exploits this property by iteratively refining a graph partition and solving a constrained version of the objective functionals. Both of these steps have specific properties allowing for their efficient computation.

Algorithmic Structure. Due to the structure of F defined in Eq. (2.2), there exist minimizers $x^\star$ of F that are piecewise constant with respect to a partition with few components. Hence, we iteratively refine a partition $\mathcal{V}$ with a top-down approach until we find a partition that lets us compute a satisfactory minimizer of F. The different steps of the cut pursuit algorithm are represented in Figure 2.2, and can be summarized as follows:

- **Init.** Set $\mathcal{V}$ as the connected components of V, *i.e.* the coarsest possible partition of V.
- **Reduction.** Minimize the reduced functional $F^{(\mathcal{V})}$ with respect to the current partition $\mathcal{V}$:

$$r^{(\mathcal{V})} \leftarrow \underset{r \in \Omega^{\mathcal{V}}}{\text{minimizes}}\, F^{(\mathcal{V})}(r) \,. \tag{2.8}$$

  The complexity of this step depends on the characteristics of F and the coarseness of $\mathcal{V}$.
- **Refinement.** We now split the current partition $\mathcal{V}$ into finer components by computing a *refining partition* $\mathcal{V}'$:

$$\mathcal{V} \leftarrow \text{refine}(\mathcal{V}, \mathcal{V}') \,. \tag{2.9}$$

  The rationale for finding $\mathcal{V}'$ depends on F and Ω. In favourable cases, this amounts to finding a minimum cut in a well-chosen flow graph.
- **Stopping Criterion.** If the chosen stopping criterion is not reached, we return to the refinement step.

In the following sections, we explain how this approach can be implemented for different choices of f, h, and Ω.

2 Cut Pursuit for Graph-Total Variation

With $h(x, y) = \|x - y\|$, finding the stationary points of F as defined in Eq. (2.2) amounts to regularize f with the graph total variation. This setting is well-studied and understood, and we are able to prove the convergence of our proposed scheme. We first provide a quick survey of the functional optimization literature on total variation. Then, we explain how to implement the cut pursuit algorithm for $\Omega = \mathbf{R}$ when f is differentiable, and

to what extent these hypotheses can be relaxed. We then present heuristic approaches to deal with data from multivariate vertices, *e.g.* $\Omega = \mathbf{R}^n$, and to parallelize the graph cut-based refinement stage of cut pursuit. Finally, we provide several numerical experiments showcasing the considerable acceleration provided by the cut pursuit algorithm for problems with coarse solutions.

2.1 Background on Total Variation

Rudin, Osher and Fatemi [ROF92] first introduced the total variation regularizer as a tractable surrogate of the Mumford-Shah functional [MS89]. It has been at the center of sustained interest in the image processing community as a simple and versatile prior on natural and medical images [EZE07]. The total variation is also the subject of numerous works in the functional optimization community. Large-scale problems regularized by the total variation are typically solved by first-order proximal splitting algorithms [CP08, RFP13, CGN$^+$13] with or without pre-conditioning [PC11, RL15, MYWC18]. While these iterative methods enjoy convergence guarantees in the convex case, they require many iterations over thousands or even millions of variables. Luckily, most approaches can be parallelized in a straightforward fashion, which can mitigate this drawback [BS14, KBJ$^+$15]. Furthermore, a large array of functions can be regularized with such schemes [LP15].

An alternative approach exploits a deep connection with graph cuts: the total variation is the Lovász extension of the submodular cut function in a well-chosen flow graph [Bac11]. This link has led to several fast algorithms, among which the one of Chambole & Darbon [CD09] who reformulate finding the proximal of the graph total variation as a parametric max-flow problem. Such methods are typically very fast thanks to efficient graph cut solvers [BK04]. However, these approaches can only be applied to regularize functions f with restrictive properties (convex, separable, smooth) and cannot be easily parallelized.

The cut pursuit algorithm for graph-total variation regularizing combines all the advantages of the previously mentioned methods:

(i) the speed of graph cut-based approaches,
(ii) the parsimony of sparsity-aware optimizers,
(iii) the ease of parallelization and
(iv) versatility of iterative proximal-based algorithms.

2.2 Regularizing Smooth Functionals

We first describe the cut pursuit algorithm in its most straightforward setting: $\Omega = \mathbf{R}$ and the function $f : \Omega^V \mapsto \mathbf{R}$ is differentiable everywhere, but not necessarily convex.

Reduction Step. This step consists in retrieving a critical point $r^{(\mathcal{V})}$ of the reduced functional $F^{(\mathcal{V})}$ defined *w.r.t* a current partition $\mathcal{V}$, see Eq. (2.6). The difficulty in finding $r^{(\mathcal{V})}$ depends on the structure of f. If f corresponds to an inverse problem, *e.g.* $f(x) = \|Hx - y\|^2$ with H a linear operator, then the minimization of $F^{(\mathcal{V})}$ has the exact same form as F only with $|\mathcal{V}|$ vertices instead of $|V|$. For other forms of f involving, for example, nonlinear operators across multiple coordinates of x, the reduced functional may not necessarily be easier to minimize than F. In this case, one should not use the cut pursuit algorithm.

The reduced functional can often be efficiently minimized with a first-order proximal-based algorithm, such as Generalized Forward Backward Splitting [RFP13]. Given the small number of variables, we can run many iterations and provide high-precision critical points. However, the reduced functional is typically poorly conditioned[4] since the size of the components of $\mathcal{V}$ and the length of their interface can vary significantly. We can address this difficulty with a preconditioning strategy such as the one proposed by Raguet and Landrieu [RL15].

Refinement Step. The goal of the refinement step is to split the components of the current partition $\mathcal{V}$ such that the following reduction step can decrease the objective functional F as much as possible. Intuitively, this can be achieved by refining the components of $\mathcal{V}$ to unlock new steep descent directions. The directional derivative of F at the current iterate x in direction $d \in \Omega^V$ can be written as follows:

$$F'(x,d) = \sum_{v\in V} \delta_v(x) d_v + \sum_{(u,v)\in E_=^{(x)}} w_{u,v} \|d_u - d_v\| \ \text{, with} \tag{2.10}$$

$$\delta_v(x) := \nabla_v f(x) + \sum_{v\in V} \sum_{u\in\mathcal{N}(v)} w_e \operatorname{sign}(x_v - x_u) \,, \tag{2.11}$$

where $\mathcal{N}(v) = \{u \in V \mid (u,v) \text{ or } (v,u) \in E\}$, $\operatorname{sign}(t) = -1, 0, \text{or } 1$ depending on $t < 0, t = 0$, or $t > 0$, and $E_=^{(x)} = \{(u,v) \in E \mid x_u = x_v\}$. We aim to find a steep descent direction d, *i.e.* leading to a small directional derivative $F'(x,d)$. The first term of Eq. (2.10) indicates the sensitivity of each vertex v to an increase of their iterate x_v. This value depends on the gradient of

[4] Some directions are much more *sensitive* than others, which complicates finding a fitting gradient step size.

(a) Observed signal.

(b) First binary partition.

(c) First reduced variable.

(d) Second binary partition.

(e) Second reduced variable.

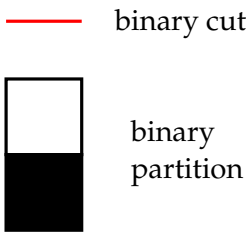
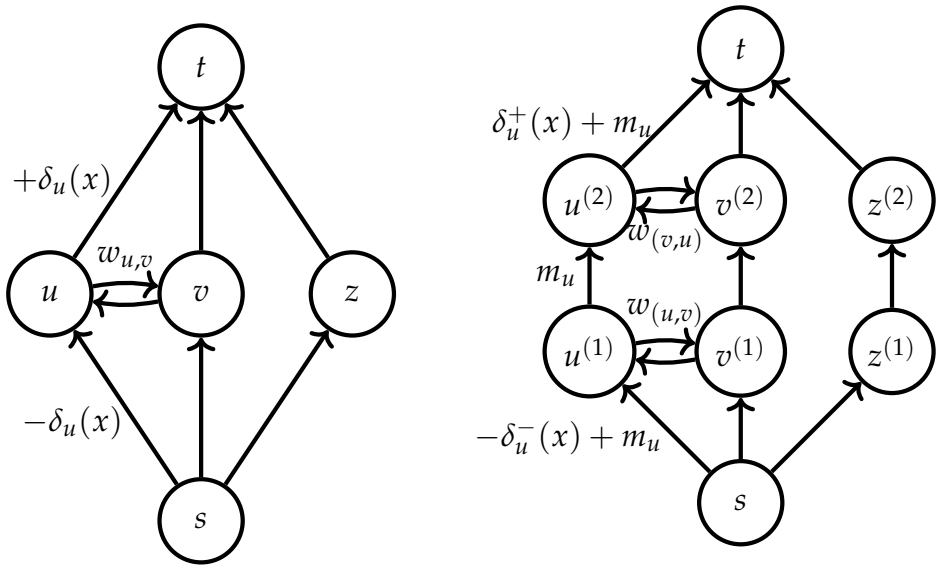


Figure 2.3: **Illustration on an Image.** We construct a graph from the pixels' adjacency, and use the pixels' intensity to assign a value y_v to each vertex v. We visualize the first two steps of the cut pursuit algorithm for a simple TV denoising problem: $\Omega = \mathbf{R}$, $f(x, y) = \|x - y\|^2$ and $h(x_u, x_v) = \|x_u - x_v\|$. We represent the graph cuts from the refinement stage as black (source-size) and white (sink-size) pixels separated by a red line representing the resulting minimal cut.

f and the values of adjacent vertices, see Eq. (2.11). The second term of Eq. (2.10) favours descent directions with identical amplitude for adjacent vertices with equal iterates. In other words, we must make a compromise between the individual tendency of each vertex to increase or decrease and avoid the creation of new discontinuities between adjacent vertices.

In general, finding the steepest direction at x is a hard problem, potentially as hard as minimizing F itself. However, we need not necessarily find the steepest direction but add relevant degrees of freedom to $F^{(\mathcal{V})}$. To make the problem more tractable, we restrict the space of potential directions to $D = \{-1, 1\}$: increase or decrease. Now, we can write the search for the steepest binary descent direction $d^{(x)}$ as a combinatorial optimization problem:

$$d^{(x)} \in \operatorname*{argmin}_{d \in D^V} F'(x, d) \,. \tag{2.12}$$



(a) Binary direction flow graph.

(b) Ternary direction flow graph.

Figure 2.4: **Flow Graphs.** Finding a minimal cut in the flow graphs above provides the steepest binary/ternary direction, and thus a refining partition. In this illustration, $x_u = x_v \neq x_z$. The *vertical* edges' capacity of the flow graphs are derived from the δ_v in the smooth setting, and from δ_v^+, δ_v^- as well $m_u := \max(0, \delta_v^-(x), -\delta_v^+(x))$ in the nonsmooth setting. The *horizontal* edges are defined by the edge weights of G.

Fortunately, this problem has a favourable structure and can be efficiently solved by finding a minimum cut in a well-chosen *flow graph*, see Figure 2.4a. Such a cut can be retrieved in quasi-linear time with efficient solvers such as the one of Boykov and Kolmogorov [BK04]. The refining partition $\mathcal{V}'$ is then defined as the maximal constant connected components of the steepest binary descent direction $d^{(x)}$.

Convergence Guarantees. Despite restricting the search directions to $\{-1, 1\}^V$, the cut pursuit algorithm is guaranteed to find a stationary point of F in a finite number of steps. If x is not a stationary point of F, then $d^{(x)}$ is a *strict* descent direction. In other words, the following reduction step will decrease F. On the contrary, if d_x does not refine $\mathcal{V}$, then x is provably a stationary point of F. In practice, since the partition induced by $x^\star$ is coarse and the size of the current partition increases exponentially with each refinement step, we only need a few iterations to reach convergence at machine precision.

In [LO17], Guillaume Obozinski and I provided a first convergence proof by identifying the flows in the flow graph with the sub-gradient of

F. The idea of this proof is that if no non-trivial cut [5] of the flow graph can be found, no descent direction of f at x is steep enough to justify creating a new discontinuity—more formally: $-\nabla f(x) \in \partial TV(x)$. We also provided additional motivation for our method by viewing the total variation as an atomic gauge [CRPW12] and the steepest binary direction as the Frank-Wolf direction of the regularization problem [Jag13]. In [RL18], Hugo Raguet and I developed a more elementary proof based on a systematic analysis of the directional derivative of F. The gist of the proof is that for a wide variety of functionals, if a strict descent direction exists, there also exists another strict descent direction whose vector representation is only composed of 1s and -1s. A major benefit of this last proof is that it does not assume the convexity of F.

The cut pursuit algorithm can efficiently recover critical points of a non-convex functional regularized by the total variation with no more than a few graph cuts. Furthermore, the graph gradient of our solutions is *exactly* sparse since all considered iterates are piecewise constant. This is in contrast to iterative algorithms, which are only sparse after thresholding procedures which often yield artifacts. Finally, our top-down strategy favours retrieving partitions with few components, offering a solution to the undesirable staircasing effects often encountered using iterative methods.

The cut pursuit algorithm provably minimizes F in just a few graph cuts. It provides convergence guarantees to machine precision, and its solutions are coarse without post-processing.

2.3 Extension to Non-Smooth Functionals

Proximal-based methods can be used to regularize a large array of functionals with the graph total variation [RFP13, LP15], including nonsmooth ones. In [RL18], we show that when $\Omega = \mathbf{R}$, the cut pursuit algorithm can be extended for f consisting of a smooth part s and a non-differentiable, vertex-separable part. In other words, f can be written as follows:

$$f(x) = s(x) + \sum_{v \in V} g_v(x_v) \,, \tag{2.13}$$

with $s : \Omega^V \mapsto \mathbf{R}$ differentiable everywhere and $g_v : \Omega \mapsto]-\infty, \infty]$ for all $v \in V$. We only assume that all g_v are *directionally differentiable* in the sense that $\lim_{t \downarrow 0} \frac{1}{t}(g_v(x + td) - g_v(x))$ exists in $]-\infty, \infty]$ for all x such that $g_v(x) < \infty$ and all $d \in \Omega^V$. Note that this definition is different from regular differentiability. In fact, any convex function is directionally differentiable regardless of smoothness [HUL04, part D], while the converse does not

[5] A cut is trivial if all vertices are on the same side: source or sink.

hold. In practice, we choose g_v as set indicators of convex subsets of Ω, or $g_v = \|\cdot\|$ to obtain the LASSO regularizer [EHJT04].

Reduction Step. The reduction step is unchanged from the differentiable case as long as a sufficiently versatile solver is used to find a stationary point of the reduced function $F^{(\mathcal{V})}$, such as the Generalized Forward Backward [RFP13].

Refinement Step. The rationale of this step is also unchanged: we add degrees of freedom to the current partition by looking for a steep descent direction. However, since the functions g_v are not differentiable, we must adapt Eq. (2.10). In particular, the directional derivative of F at $x \in \Omega^V$ in direction $d \in \Omega^V$ now writes:

$$F'(x,d) = \sum_{\substack{v\in V \\ d_v>0}} \delta_v^+(x) d_v + \sum_{\substack{v\in V \\ d_v<0}} \delta_v^-(x) d_v + \sum_{(u,v)\in E_=^{(x)}} w_{u,v}\|d_u - d_v\| \,, \quad (2.14)$$

with

$$\delta_v^+(x) := \nabla_v f(x) + g_v'(x_v, +1) + \sum_{v\in V}\sum_{u\in\mathcal{N}(v)} w_e \operatorname{sign}(x_v - x_u) \quad (2.15)$$

$$\delta_v^-(x) := \nabla_v f(x) - g_v'(x_v, -1) + \sum_{v\in V}\sum_{u\in\mathcal{N}(v)} w_e \operatorname{sign}(x_v - x_u) \,. \quad (2.16)$$

The main difference between Eq. (2.14) and Eq. (2.10) is that non-smooth functions g_v may have different directional derivatives in the increasing $(+1)$ or decreasing directions (-1). Consequently, it is now possible for a given vertex to not favour either the increasing or decreasing directions. This leads us to add the null value to the discrete set of possible directions D when simplifying the search for a steep descent direction Eq. (2.12): $D = \{-1, 0, 1\}$. Solving this steepest ternary descent direction problem corresponds to finding a minimum cut in a suitable flow graph, represented in Figure 2.4b.

Convergence Guarantees. The non-smooth setting benefits from the same guarantees as the smooth setting: if x is not a stationary point, then a strict ternary descent direction exists, *i.e.* $d^{(x)} \in D^V$ such that $F'(x, d^{(x)}) < 0$. Consequently, if solving Eq. (2.12) returns a trivial cut, then x is a stationary point. In practice, we only need a few iterations (less than 10) to achieve machine precision for most standard total variation parameterizations.

2.4 Extension to Multidimensional Values

We can extend the cut pursuit algorithm to the case where each vertex is associated with a multidimensional variable. Although our convergence

guarantees do not hold, this approach efficiently leads to satisfactory approximate solutions.

Choosing a Direction Set. In the previous setting, restricting the set of candidate directions D to increasing $(+1)$, decreasing (-1), or null (0) is enough to obtain our optimality certificate. This allows us to efficiently find the steepest binary/ternary descent directions with a single graph cut. In the setting where Ω is multidimensional, *i.e.* when the signal x has several values for each vertex, such restricted directions may not be sufficient. To preserve optimality guarantees, D would have to be large or even infinite, preventing us from efficiently finding a steep descent direction. However, we can choose D heuristically to achieve empirical efficiency without a theoretical convergence guarantee.

A key point to note is that the set of potential directions D can be chosen for each vertex independently and change according to the current iterate. If each vertex v has its own set of candidate directions D_v, then the total set of directions in Eq. (2.12) is the Cartesian product of all directions $\bigtimes_{v\in V} D_v$.

Examples. In the case of $\Omega = \mathbf{R}^n$, we can define a heuristic set of directions for each component U of $\mathcal{V}$ by computing e_U the first eigenvector [Pea01] of the centered values of the iterate for the component: $\{x_v\}_{v\in U}$. We can then define $D_v = \{-e_U, +e_U\}$ for all $v \in U$, or add even more directions corresponding to other eigenvalues. Each component can adapt its descent directions according to its content, resulting in better splits and fewer computations than generic candidate directions.

We also propose a meaningful heuristic direction set when $\Omega = \Delta_K$, the K-dimensional simplex. For each vertex $v \in V$, we denote by $k_v := \operatorname{argmax}_{k\in[1,K]} x_{v,k}$ the corner of the simplex which is closest to x_v, *i.e.* the label for which x_v associates the highest predicted probability. We then define $D_v = \{\mathbb{1}_k - \mathbb{1}_{k_v} \mid k \in [1, K]\}$, with $[\mathbb{1}_k]_l = 1$ if $k = l$ and 0 otherwise. In other words, each vertex can decide to transfer probability from a label to its most likely label or remain unchanged. Our split can factor in various configurations by adapting the descent direction at the vertex level.

2.5 Parallel Refinement

In practice, the refinement step is often the computational bottleneck of the cut-pursuit algorithm. The reduction step only involves $|\mathcal{V}|$ variables and relies on easily parallelizable solvers. In contrast, finding a steep descent direction implies finding minimal cuts in a flow graph with at least $V + 2$ vertices. Although the Ford-Fulkerson graph cut solver can be surprisingly fast [FF56], it is not easily parallelizable.

Parallelizing Graph Cuts. All binary terms in Eq. (2.10) and (2.14) correspond to edges *within* the constant components of the current partition $\mathcal{V}$. There are no binary terms connecting vertices in different components. Consequently, $F'(x, d)$ is separable along the components of $\mathcal{V}$. For the binary case, this means that we can rewrite $F'(x, d)$ as follows:

$$F'(x,d) = \sum_{U \in \mathcal{V}} F'_U(x_U, d_U)\,, \quad \text{with for all } U \in \mathcal{V}, \tag{2.17}$$

$$F'_U(x_U, d_U) = \sum_{v \in U} \delta_v(x) d_v + \sum_{(u,v) \in E^{(x)}_{=} \cap U^2} w_{u,v} \|d_u - d_v\|\,, \tag{2.18}$$

In terms of graph cuts, no flow can be exchanged between the components of $\mathcal{V}$, allowing us to compute the cuts in each component *independently*. We implemented a *multi-threaded* version of the already efficient implementation of the Ford-Fulkerson algorithm [FF56] by Boykov and Kolmogorov [BK04].

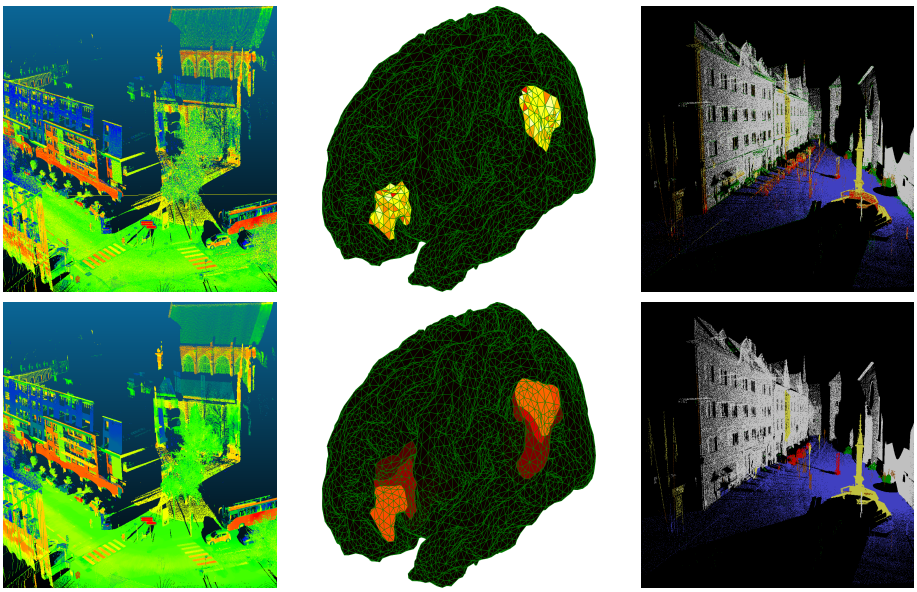
[Taking advantage of the particular structure of the refinement problem, we propose the first multi-threaded application of the Ford-Fulkerson algorithm.]

Balancing the Parallel Workload Distribution. While our parallelizing strategy already considerably accelerates the cut pursuit algorithm, we can improve thread utilization even further. Indeed, when performing the refinement step in parallel, each component is assigned to a single thread. Consequently, the computation takes at least as long as the time required to refine the most challenging component, typically the largest one. If the partition is unbalanced, this can lead to poor threads usage. Consider the first iteration, where the partition only has one component, preventing parallelization.

For this reason, we advocate decomposing the largest components into smaller *balancing* components with a maximum size defined by the problem size and the number of available threads. This can be done efficiently with a breadth-first search from random starting vertices. In the refinement step, we set the border edge capacities between balancing components to 0 in Eq. (2.17). In the reduction step, we can ignore the balancing components. In theory, this partitioning strategy may lead to worse local minima or reduce the accuracy of the solution. In practice, the acceleration provided might offset the disadvantage of slightly less optimal solutions depending on the application.

Let us finally note that there is still room for improvement. For instance, we could replace the greedy breadth-first construction of the balancing components with better domain-specific heuristics. This is especially true for spatially embedded graphs as many fast and data-adaptive

partition algorithms exist for 3D point clouds [MD03, Mea82, YL22]. Furthermore, the compelling time and space complexity of the Ford-Fulkerson algorithm does not simply depend on the size of the flow graphs but also on the ratio between *vertical* and *horizontal* capacities: large horizontal capacities tend to incur longer computation time. This observation may lead to better parallel scheduling. However, an important limitation of this line of improvement is that the balancing step must be efficient or risk negating any computational benefit otherwise provided.



(a) Reflectance Regularization

(b) Brain Source Identification

(c) Segmentation Regularization

Figure 2.5: **Qualitative Results.** Illustration of the solutions given by the cut-pursuit algorithms. (a) noisy and regularized point cloud reflectance (detail of a much larger scene); (c) synthetic and recovered ground truth brain activity; (b) noisy prediction and regularized point cloud classification.

2.6 Numerical Experiments

We showcase the performance of the cut pursuit algorithm across three different experiments. See Figure 2.5 for a visual representation of these tasks.

Point Cloud Reflectance Regularization. We spatially regularize the reflectance of 3D points acquired with a mobile LiDAR scanning vehicle as described by [PPC$^+$12]. This corresponds to a denoising problem $f : x \mapsto$

$\|x - y\|^2$ with y the observed noisy reflectance. G is the 5-nearest neighbours graph of the point cloud ($|V| = 109$m and $|E| = 317$m). We expect the solution to be simple as anthropic scenes typically contain parts with uniform materials and reflectance.

Brain Source Identification in Electroencephalography. We consider the inverse problem of brain source identification from electroencephalography. The brain of a patient is mapped to a triangular mesh whose adjacency structure is described by a graph $G = (V, E)$ with $|V| = 20$k and $|E| = 29$k. A set of $N = 91$ electrodes records the patient's brain activity $y \in \mathbb{R}^N$, and the goal is to retrieve the neuron activation on the detailed mesh. The relationship between the electrodes' output and the brain activity is given by a *lead-field* linear operator $\phi : \mathbb{R}^V \mapsto \mathbb{R}^N$, which is derived from physical and physiological considerations by experts of the domain [BAC+14]. To model the regularity, sparsity, and positivity of brain signals, we set

$$f : x \mapsto \frac{1}{2} \|y - \phi x\|^2 + \sum_{v \in V} \left(\lambda_v |x_v| + \iota_{\mathbb{R}_+}(x_v)\right) , \tag{2.19}$$

with λ_v the parameters of the weighted LASSO regularization and $\iota_{\mathbb{R}_+}$ the set indicator function of $\mathbb{R}_+$.

Point Cloud Classification Regularization. We consider the problem of spatially regularizing noisy semantic labellings of point clouds among a class set K [LRV+17]. G is the 10-nearest-neighbours graph of the point cloud, ($|V| = 3\,000\,111$ and $|E| = 17\,206\,938$). A noisy classification $y \in \mathbf{R}^{V \times K}$ is given by a random forest classifier operating on handcrafted vertex geometric features as described in [GL17]. Noting ι_{Δ_K} the convex indicator of the $|K|$-dimensional simplex, and $\mathrm{KL}(r, s) = \sum_{k \in K} r_k \log(r_k / s_k)$ the *Kullback–Leibler divergence*, we choose

$$f : x \mapsto \mathrm{KL}(y_v, x_v) + \sum_{v \in V} \iota_{\Delta_K}(x_v). \tag{2.20}$$

In this multidimensional setting, we use the heuristic direction set described in Section 2.4 for simplices.

Competing Methods. We compare the performance of the cut pursuit algorithm against highly specialized graph cut-based approaches and flexible proximal algorithms. The competing methods—all implemented in C++—are as follows:

- **PMF:** the parametric max flow-based algorithm of [CD09] for the proximity operator of the graph total variation;

- **PFDR:** the Forward-Douglas–Rachford splitting algorithm of [Rag18] preconditioned according to [RL15], with all proximal steps parallelized;
- **CP:** cut pursuit algorithm with PFDR to solve the reduced problem;
- **PCP:** CP with parallel refinement, without balancing the parallel workload distribution;
- **PCP-balanced:** PCP with component balancing.

Analysis. We report the performance across methods and tasks in Figure 2.6. We first compute a high-precision estimate of the optimal value for these convex problems by running PFDR for $1e5$ iterations, which leads to an iterate evolution under $10^{-}8$ as measured by the largest coordinate change. This allows us to approximate and plot the distance towards optimality.

When the sought solution is coarse, the cut pursuit algorithm can accelerate by several orders of magnitude compared to state-of-the-art solvers. In simple settings such as TV-denoising, our approach outperforms the highly specialized PMF algorithm [CD09] by a significant margin. We can also employ cut pursuit to more complex settings, such as the regularization of nonconvex and nondifferentiable functions, and observe comparable improvements.

In all experiments, the cut pursuit algorithms converge in less than 10 iterations, translating to faster recovery than iterative methods as detailed in Table 2.1. In the point cloud smoothing experiment, the cut pursuit algorithm performs as well as the highly specialized Parametric Max-Flow formulation. Parallelizing the refinement steps brings a significant speed-up, and is further improved when using the component balancing strategy.

In the brain source identification, the final partition contains only 20 constant component. This setting advantages cut pursuit, which can solve this complex, poorly conditioned, and medium-sized problem several orders of magnitude faster than iterative methods. However, parallelization does not bring significant improvements in this setting, and the balancing strategy leads to a suboptimal solution.

Cut pursuit also brings significant acceleration to the semantic segmentation regularization problem. Despite the non-applicability of our convergence guarantees, our method can retrieve high-quality solutions. The balancing strategy further accelerates the recovery.

Warm Restart Strategy. In contrast to methods using dual [CD09] or auxiliary variables [Rag18, RFP13], the cut pursuit algorithm can leverage a

Table 2.1: **Recovery Speed.** Time for the method to reach a solution within 10^{-2} of the approximate optimal. PMF cannot be used to minimize the functionals in the second and third experiments. The cut pursuit algorithm offers acceleration of several orders of magnitude.

	Reflectance Regularization	Brain Source Identification	Segmentation Regularization
PFDR	1123	1.40	838
PMF	732	-	-
CP	511	**0.04**	271
PCP	255	**0.04**	145
PCP-Bal.	**150**	0.11	**39**

candidate iterate believed to be close to the solution with a simple warm restart strategy. Indeed, instead of starting the current partition $\mathcal{V}$ with $\{V\}$, we can use the maximal constant connected components of the candidate iterate. When computing a sequence of problems with decreasing regularization strength, we can use the previous partition as initialization, see [LO17][2.8]. This strategy allows us to approximate the entire regularization path of TV for a cost comparable to solving the lowest regularization setting.

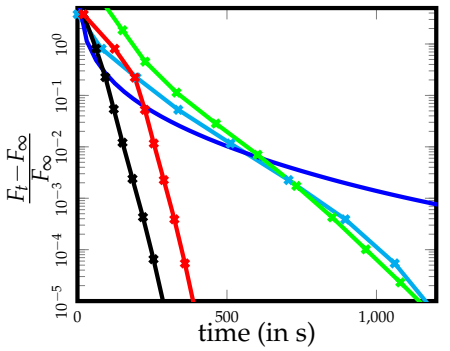
Conclusion. We have provided a theoretical and practical framework for harnessing the speed of efficient graph-cut algorithms for a large class of graph-structured problems involving nondifferentiable terms alongside the total variation. Cut pursuit is not meant to be an all-purpose algorithm. It should only be used if the solution is expected to be piecewise constant *w.r.t* a coarse partition, and this is not generally the case with natural images. However, we believe that cut pursuit addresses some of the limitations of common solvers for such problems:

(i) Cut pursuit is both fast and versatile as does not require convexity or differentiability. In contrast, competing methods are either fast but very specialized (PMF) [CD09], or generic but slow [Rag18].

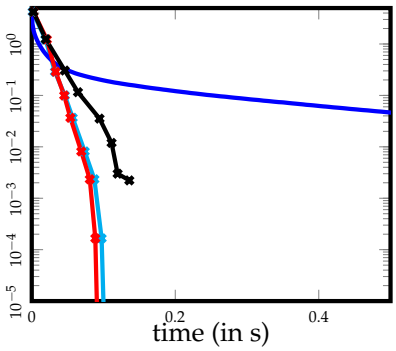
(ii) Cut pursuit retrieves a solution that is *exactly* piecewise constant. In contrast, the iterative method requires a thresholding step which can engender artifacts.

(iii) By construction, the cut pursuit algorithm retrieves a coarse solution. In contrast, iterative methods generally find solutions with many small constant components due to the *staircasing effect* of total variation [Jal16].

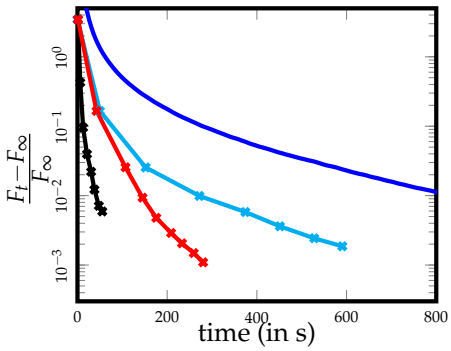
(iv) Cut pursuit allows us to run the fast Ford-Fulkerson graph cut algorithm in parallel to solve continuous optimization problems.



(a) Reflectance Regularization.
109 M vertices, 317 M edges.



(b) Brain Source Identification.
19 k vertices, 29 k edges.



(c) Segmentation Regularization.
3 M vertices, 17 M edges.

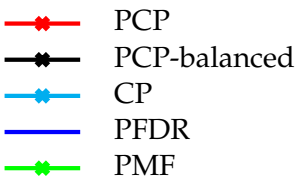


Figure 2.6: Relative distance to the approximate optimal against the running time of the algorithms; optimal values are estimated by longer, high-precision runs.

3 ℓ_0-Cut Pursuit for Contour Regularizing

We now consider the case where $h(x_u, x_v) = [x_u - x_v]$ with $[\,\cdot\,] : \Omega \mapsto \mathbf{R}$ equals to 0 at 0_Ω and 1 elsewhere. In other words, the graph-structured regularization in Eq. (2.2) becomes the cut between constant components, *i.e.* the total contour length of the induced partition. This regularization can be linked to several other commonly used regularizers and problems. We can see this penalizer as a continuous-space version of the Potts penalty [Pot52], a generalized version of the minimal partition problem in which the number of components is not predefined [Lec89], a discrete (graph) version of *Caccioppoli partitions* [TC96], or an ℓ_0 version of the total variation

following the terminology of Candes *et. al* for the LASSO penalty [CWB08].

In this setting, the functional F is neither continuous nor convex. Consequently, finding a global optimal is generally not an option regardless of the convexity of f. However, we show that our approach can be adapted into a greedy formulation, dubbed ℓ_0-cut pursuit, which can find quickly and reliably good local minima of F. We restrict ourselves to the setting in which f is separable into continuous functions: $f(x) = \sum_{v \in V} f_v(x_v)$ with $f_v : \Omega \mapsto \mathbf{R}$. This assumption is not necessary but greatly simplifies the formulation of the following sub-problems.

3.1 Reduction.

For a given partition $\mathcal{V}$ with reduced graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$, we define the reduced functional $F^{(\mathcal{V})}$ for the reduced variable $r \in \Omega^{\mathcal{V}}$ as follows:

$$F^{(\mathcal{V})}(r) = \sum_{U \in \mathcal{V}} \sum_{v \in U} f_v(r_U) \,. \tag{2.21}$$

Note that we drop the edge-wise regularization term altogether. The rationale is that during the reduction step, we only change the values of components and do not add nor remove existing *borders*. Since the ℓ_0 norm penalizes all non-zeros values identically, we can completely ignore the penalty term in the reduction phase. We can now solve the reduced problem for each component independently and in parallel. A caveat of this simplification is that adjacent components with very close values may benefit from adjusting their iterates to remove their mutual borders. This case is handled by a backward step defined below.

3.2 Refinement.

In the original cut pursuit formulation, we perform the refinement step by minimizing an energy determined by the directional derivative of F. Since the functional F is non-continuous in the ℓ_0 setting, we do not have access to such values. Instead, we define a greedy counterpart of the steepest binary partition that we dub the *optimal binary partition*, which we use to split the current partition $\mathcal{V}$. The optimal binary partition $B^\star$ is a solution to the following bilevel optimization problem:

$$\begin{aligned} B^\star \in \operatorname*{argmin}_{B \subset V} \min_{h,h' \in \Omega^{\mathcal{V}}} \sum_{U \in \mathcal{V}} \left[\sum_{v \in U \cap B} f_v(h_U) + \sum_{v \in U \cap B^c} f_v(h'_U) \right] \\ + \sum_{U \in \mathcal{V}} w\left(U \cap B, U \cap B^c\right) \,, \end{aligned} \tag{2.22}$$

with B^c the complementary of B *w.r.t* V. The rationale of this step is to split each component U of the current partition $\mathcal{V}$ along a binary partition $(B \cap$

$U/B^c \cap U)$, associated with the constant values $h_U, h'_U \in \Omega^2$ respectively. The second term quantifies the additional boundary incurred by the new partition.

The optimization problem defined in Eq. (2.22) is difficult as the second term is still nonconvex and noncontinuous. We can, however, find an approximate solution by alternating between solving for (h, h') and B. The partition retrieved is provably a local minimum of the nonconvex optimization problem defined in Eq. (2.22). In practice, only a few (two or three) iterations are necessary.

3.3 Backward-Step.

It is common for greedy working set methods to implement a backward step in which previous decisions can be reversed, see the Single Best Replacement method of Soussen *et. al* [SIBD11]. In our setting, this amounts to removing existing borders by merging adjacent components. While we explored different strategies, the simplest one is, in most cases, sufficient.

We perform this step after the refinement step. We start by computing for each adjacent component $(U, U') \in \mathcal{E}$ in the reduced graph the change in the objective functional F incurred by their merging. If this value is negative, the reduction in contours length outweighs the removal of one degree of liberty incurred by forcing the vertices in $U \cup U'$ to share the same value. Such cases typically happen for borders added at early iterations that are no longer relevant after further refinement. We then greedily select the most *profitable* move, *i.e.* which most decreases the objective functional and update the potential gains of all adjacent components. This step is repeated until there are no more profitable fusions. With f vertex-separable, this step is performed by considering the reduced graph only, and is hence very fast.

While the contour penalty defines a difficult nonconvex, noncontinuous problem, the ℓ_0-cut pursuit algorithm can efficiently find approximate solutions, outperforming celebrated approaches such as α-expansion [BVZ01] with observed accelerations up to 100-fold.

3.4 Numerical Experiments.

We evaluate the performance of the ℓ_0-cut pursuit algorithm for a 3D point cloud segmentation. This large-scale optimization problem consists of grouping points into clusters with homogeneous properties and simple contours, called *superpoints*. We consider 271 point clouds from the S3DIS dataset [ASZ+16], averaging 116 000 points each after subsampling. In the next chapter, we give more details about this experiment and its motivation.

For a 3D point cloud, we define a graph $G = (V, E, w)$ such that V is the set of 3D points, E encodes their adjacency, and w their proximity. We associate with each vertex v a value $y_v \in \mathbf{R}^7$, encoding its radiometry and local geometry. We aim to compute a piecewise constant approximation of this signal into few components. This amounts to minimizing the following functional for $x \in \Omega^V$:

$$F(x) := \sum_{v \in V} \|x_v - y_v\|^2 + \sum_{(u,v) \in E} w_{u,v}[x_u - x_v] \,. \tag{2.23}$$

We can approximately minimize this functional with the ℓ_0-cut pursuit algorithm. As a baseline, we define the following approach:

- We set the number of classes K.
- We run the K-means algorithm on the observations $y \in \mathbf{R}^7 \times V$ to obtain class values $k \in \mathbf{R}^{7 \times K}$. The resulting centroids define a discrete set of values that each vertex can take.
- We compute the combination of classes minimizing F. This corresponds to a classical energy minimization problem which can be solved approximately with α-expansion.
- We update the class values by computing their feature average. This step is homologous to the debiasing post-processing often applied to the solution of regularized problems [DPS15].

Note that this baseline is only partially comparable to cut pursuit: the former operates on a discrete set of values while the latter works in a continuous space. Since the problem itself is defined *w.r.t* a continuous domain, we can expect the cut pursuit approach to reach lower energies. However, we can compare the execution time of the two approaches. Indeed, both heavily rely on graph cuts to minimize the energy, albeit in a different manner: α-expansion solves a graph cut problem for each class at each iteration, while the cut pursuit algorithm uses the cuts to split existing components. Comparing the execution speed between these methods may inform us which graph cut formulation is more efficient for piecewise constant approximation.

In Figure 2.7, we represent the evolution of the performance of ℓ_0-cut pursuit, with or without parallelization, as well as two versions of our baseline with $K = 10$ and 20. We observe that ℓ_0-cut pursuit reaches lower energies, which we expected. More interestingly, our approach provides an acceleration of over 100 fold compared to α-expansion. This indicates that the component-wise splitting of ℓ_0-cut pursuit might be a much better strategy than the class-wise splitting of α-expansion when the sought signal is coarse. Furthermore, we can bring further speed-ups with parallelization.

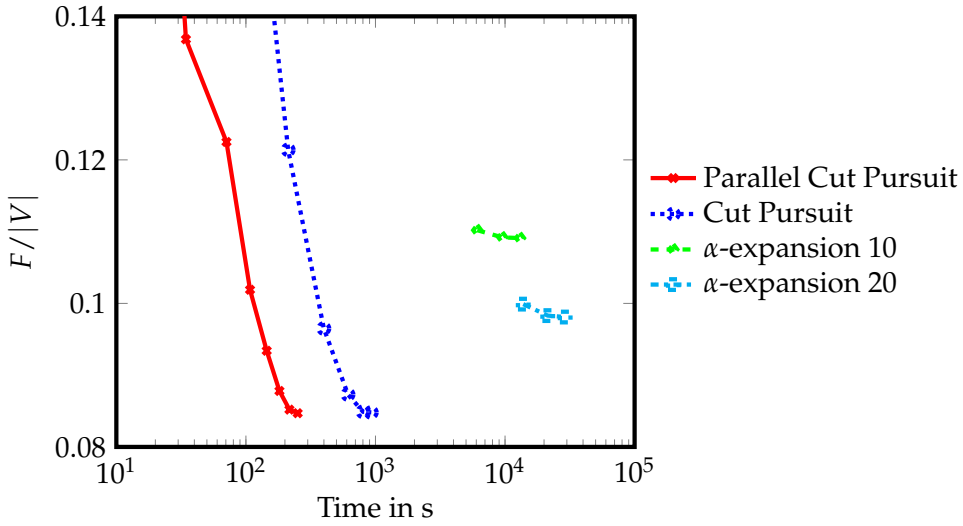


Figure 2.7: **Numerical Experiment for ℓ_0-cut pursuit.** We represent the evolution of $F/|V|$ with respect to computation time in seconds. The cut pursuit algorithm produces its approximation in little more than 2 minutes with parallelization, and under 7 minutes without. In contrast, α-expansion takes up between 2 and 8 hours depending on the chosen number of labels and desired precision.

4 The Plane-Pursuit Algorithm

Our approach is particularly well-suited for data exhibiting spatial regularity and well-represented with graphs, such as 3D point clouds. This setting has attracted most of the interest in cut pursuit in terms of applications: [TGB19, GL17, LRV+17, LS18, MSF20]. In the next chapter, we will detail an approach to scale the analysis of deep networks to point clouds containing millions of points with cut pursuit. This section presents a specialized adaptation of cut pursuit for computing piecewise-planar approximations of large 3D point clouds.

We derive the graph $G = (V, E, w)$ from an unstructured 3D point cloud: V is the set of 3D points, E captures local neighbourhoods, and $w \in \mathbf{R}_+^E$ the proximity of points. We aim to approximate a point cloud with a small number of 2D planes embedded in $\mathbf{R}^3$. We propose to formulate this task as the recovery of a graph-coarse signal on graph G.

Let $\mathcal{P}$ be the set of all planes of $\mathbb{R}^3$. We denote by $d(v, \pi)$ the distance between a point v and a plane $\pi \in \mathcal{P}$. We define $\Pi : V \mapsto \mathcal{P}$ as the function that associates each vertex of V with a plane Π_v in $\mathcal{P}$. We want Π to define as few planes as possible while remaining a good approximation of the point cloud. This task amounts to recovering a signal Π on the set of vertices V for which (i) the points are well approximated by their pro-

jection in their respective plane: $d(v, \Pi_v)$ is small and (ii) the signal Π is graph-coarse. We can translate these objectives into a unique functional F to minimize:

$$F(\Pi) := \sum_{v \in V} d(v, \Pi_v)^2 + \mu \sum_{(u,v) \in E} w_{u,v} [\Pi_u \neq \Pi_v] \,, \tag{2.24}$$

where $[\pi \neq \pi']$ is the function of $\mathcal{P}^2 \to \{0, 1\}$ equal to 0 when π and π' are identical planes and 1 otherwise. The parameter $\mu \in \mathbb{R}_+$ is the regularization strength. The first term of Eq. (2.24) measures the fidelity of the planar reconstruction defined by Π. The second part of Eq. (2.24) encourages planes associated with adjacent vertices to be identical.

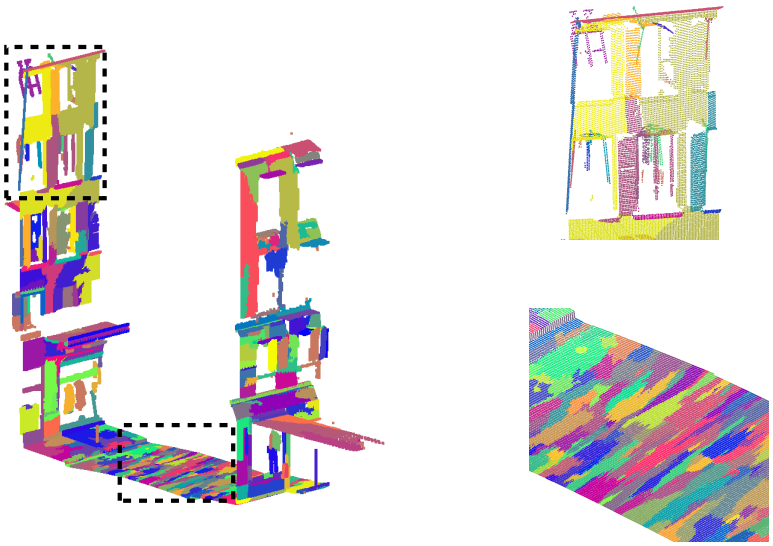
The functional defined in Eq. (2.24) is similar in structure to Eq. (2.2), and we can apply the same algorithmic scheme as in the previous sections with minor adjustments: the reduction step now perform plane-fitting for each component, and the refinement step use RANSAC [FB81] to initialize the splitting procedure described in Section 3.2.

Numerical Experiments. We compare our results with a traditional planar segmentation approach based on a region-growing clustering by Cohen *et. al* [CSAD04]. For a given number of components/regions, plane pursuit improves the geometric error by an entire order of magnitude and speed by more than 50-fold. In Figure 2.8, we represent qualitative results illustrating the reason behind this performance: plane pursuit fits large planes where the geometry is simple, adapting the size of planes to the local geometry's complexity.

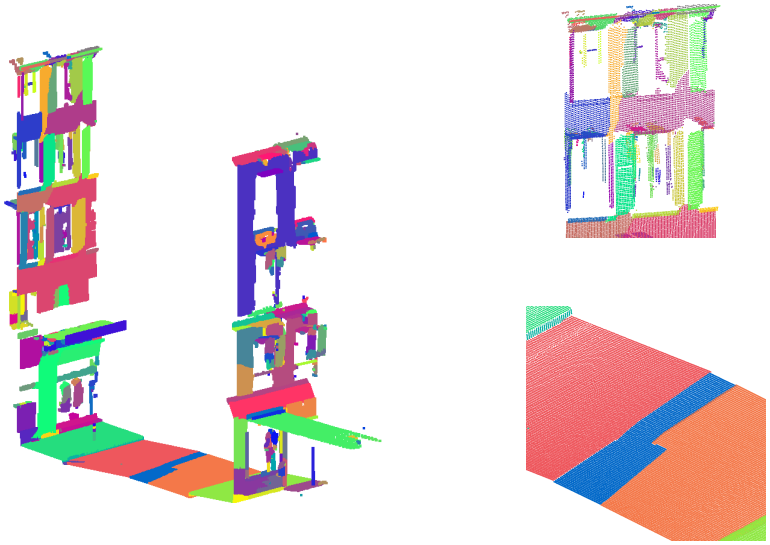
The cut pursuit algorithm can be adapted for the efficient planar reconstruction of unstructured 3D point clouds. The resulting plane-pursuit algorithm takes advantage of the regularity of the solution to provide both a significant acceleration and a better approximation.

Conclusion

Cut pursuit is a meta-algorithm, which can take different forms depending on the property of the functional to minimize. We provide convergence proofs for a wide range of data functionals when using the total variation as graph regularizer. The cut pursuit algorithm can be seen as an efficient heuristic strategy in more complex settings. In all cases, we provide considerable acceleration when the sought signal exhibits graph-coarseness. This results from the algorithmic design that exploits such structured sparsity but also from an efficient parallel implementation.



(a) Region-growing baseline with 492 regions.



(b) Plane-Pursuit with 492 components.

Figure 2.8: **Qualitative Results for Plane-Pursuit.** We represent the projection of each point on their supporting plane, and each colour represents a different component/region. Our method yields planes of adaptive size: large in geometrically simple regions such as roads, and smaller for complex areas such as facades and window panes.

CHAPTER 3

Exploiting the Spatial Regularity of 3D Data

This chapter presents different methods for exploiting the spatial regularity of large 3D point clouds to accelerate their automated analysis. We first propose a versatile family of spatial regularizers for semantic labelling. We then introduce superpoint graph, a compact yet rich representation of large point clouds. This structure can be computed efficiently and leads to significant improvement in terms of precision and scalability. We also present a hybrid approach combining deep learning and combinatorial optimization to mesh large point clouds.
This chapter is organized around the following publications:

[LS18]: Loic Landrieu, Martin Simonovsky, "Large-scale Point Cloud Semantic Segmentation with Superpoint Graphs", *CVPR*, 2018
[LB19]: Loic Landrieu, Mohamed Boussaha, "Point Cloud Oversegmentation with Graph-Structured Deep Metric Learning", *CVPR*, 2019
[CCHL20]: Chaton, Thomas, Nicolas Chaulet, Sofiane Horache, and Loic Landrieu, "Torch-Points3D: A modular multi-task framework for reproducible deep learning on 3D point clouds", *3DV*, 2020
[SLMV21]: Raphael Sulzer, Loic Landrieu, Renaud Marlet, Bruno Vallet, "Scalable Surface Reconstruction with Delaunay-Graph Neural Networks", *Symposium on Geometry Processing (SGP)*, 2021
[LRV+17]: Loic Landrieu, Hugo Raguet, Bruno Vallet, Clément Mallet, Martin Weinmann, "A structured regularization framework for spatially smoothing semantic labelings of 3D point clouds", *ISPRS Journal of Photogrammetry and Remote Sensing*, 2017
[GL17]: Stéphane Guinard, Loic Landrieu, "Weakly supervised segmentation-aided classification of urban scenes from 3D LiDAR point clouds", *ISPRS - International Archives of the Photogrammetry and Remote Sensing and Spatial Information Sciences*, 2017

1 Regularizing 3D Point Clouds Segmentation

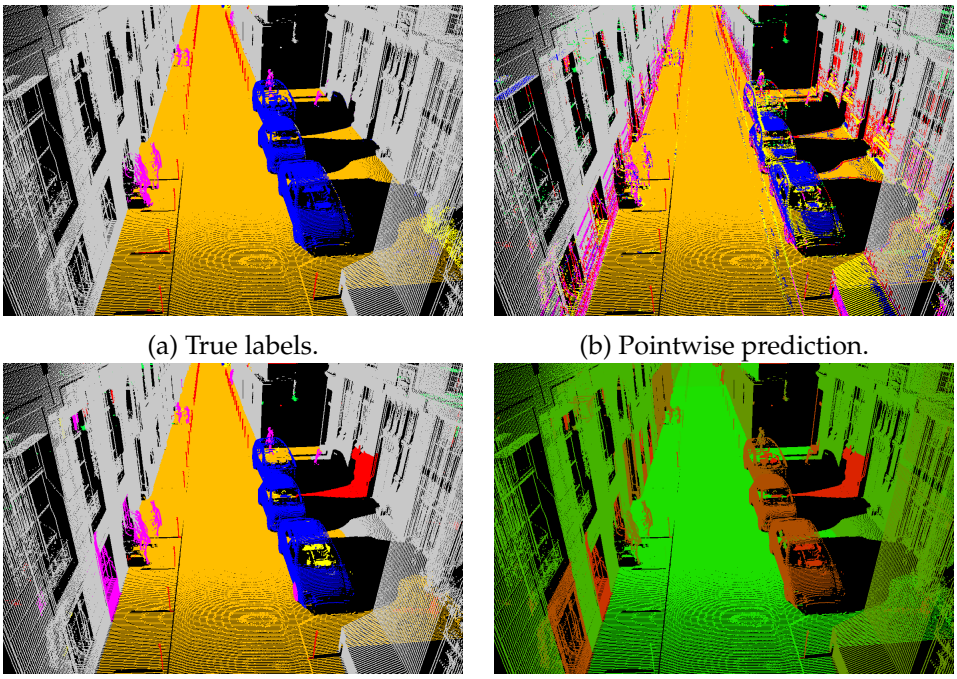
We present a structured-optimization approach to impose a spatial regularity prior to probabilistic 3D point classifications. Our approach increases performance while retaining the probabilistic nature of the prediction.

Classical Approaches to 3D Semantic Segmentation. Steady improvements in sensor technology allow the acquisition of 3D point clouds with high resolution and precision. A straightforward consequence of the point density is that the resulting point clouds exhibit a strong spatial regularity. This property manifests itself in the semantic labels associated with each 3D point: a point *surrounded* by points belonging to the class "car" most likely has the "car" label as well, as seen in Figure 3.1a.

Until the recent rise of 3D deep learning [QSMG17a, GWH+20], most 3D point cloud semantic segmentation pipelines classified individual points based on hand-crafted features [WJHM15]. These features are derived from spatial neighbourhoods of each point, which can be fixed [DMDV11], adaptive [WSM+15], or multiscale [TGD+18]. However, the label associated with each point is ultimately decided independently for each point, leading to the low spatial regularity of the labels as observed in Figure 3.1b.

Many works use graphical models such as Markov Random Fields (MRF) [MBVH09, SVB10, LR12, NNSP14] or their discriminative counterpart, Conditional Random Fields (CRF) [SNRS14] to increase the spatial regularity of the prediction. Computing marginal inference on a large scale is costly and can lead to poor results. Maximum a Posteriori (MAP) inference can be efficiently and reliably approximated with graph cut-based algorithms [BVZ01]. However, this type of inference produces a single label for each point and discards any confidence information from the prediction. Calibrated predictions, as seen in Figure 3.1d, are crucial in 3D analysis, particularly for applications such as autonomous driving [HWB+13] or surface reconstruction [CBV17, CKR04].

Spatial Smoothing as Structured Optimization. Let $G = (V, E, w)$ be an edge-weighted graph characterizing the adjacency structure of a 3D point cloud indexed by V. We consider a probabilistic labelling $\delta \in \Delta^V$ of the points of V, with Δ_K the K-simplex when K is the number of classes. We denote $\delta_{v,k}$ the probabilistic prediction associated with the vertex v and class k. δ is typically obtained by classifying each point individually as proposed in [WJHM15]. We propose to advantageously formulate the problem of increasing the spatial regularity of δ as a structured optimization problem in



(a) True labels. (b) Pointwise prediction.

(c) Regularized prediction. (d) Confidence.

Figure 3.1: **Smoothing a Pointwise Prediction.**:A 3D point cloud taken from the Paris-rue-Cassette Dataset [VBS+15] and with true annotation (a). A pointwise classifier first gives a noisy semantic segmentation (b) which we smooth to improve the spatial regularity (c). Our approach allows to retain the probabilistic nature of the segmentation (d) : confident uncertain. Note that misclassifications in (c) correspond to the least confident area in (d).

the domain Ω^V:

$$q^\star \in \operatorname*{argmin}_{q \in \Omega^V} \sum_{v \in V} f(\delta_v, q_v) + \sum_{(u,v) \in E} w_{u,v} h(q_u - q_v) \,. \tag{3.1}$$

For different choices for Ω, f, and h, this versatile formulation allows us to define both classic and new smoothing problems:

- **MAP Inference in a CRF. (MAP)** We set Ω to $\mathbf{\Delta}_K$ the corner of the K-simplex, $f(\delta_v, q_v) = \sum_{k=1}^K q_{v,k} \log(\delta_{v,k})$, and $h = [\cdot = 0_\Omega]$ equal to 1 at 0_Ω and 0 otherwise. This formulation corresponds to the MAP problem in a CRF defined by G. This combinatorial problem can be efficiently approximated with α-expansion, but the solution is a simple labeling without confidence values.
- **TV-regularized KL-Divergence. (KL-TV)** We set $\Omega = \Delta_K$, $f(\delta_v, q_v) = \sum_{k=1}^K \delta_{v,k} \log(q_{v,k})$, and $h(x) = \|x\|_{1,1} = \sum_{k=1}^K |x_k|$. The choice of f corresponds to the (variable part of the) Kullback-Leiber divergence, a similarity measure between distributions based on information theory [KL51]. The choice of h leads to the graph-total variation presented in Section 2.1. The resulting convex problem is typically solved using first-order proximal methods such as the Generalized Forward-Backward Splitting algorithm [RFP13].
- **Continuous Potts with KL-Divergence. (KL-Potts)** In this formulation, we change h to $[\cdot = 0_\Omega]$. This results in a continuous-space, nonconvex, and noncontinuous optimization problem, which we have extensively described in Chapter 2. This functional can be efficiently minimized with the ℓ_0-cut pursuit algorithm.

As a baseline, we also consider the marginal inference in the CRF defined by G (Marginal), solved with Loopy-Belief Propagation [Pea82].

Numerical Experiments. We evaluate these four approaches on three open-access medium-scale 3D scenes containing between 1.3 and 12 million points and between 3 and 7 distinct semantic classes. We first perform adaptive-scale geometric feature extraction [WSM$^+$15] and train a random forest classifier on a small subset of 1000 points to obtain noisy pointwise predictions. We observe in Figure 3.2 that the performance of marginal inference provides significantly lower results than the other methods in terms of the F1-score. MAP inference leads to results of comparable quality to KL-TV and KL-Potts, but loses the probabilistic nature of the prediction. In contrast, we can associate confidence with each prediction of KL-TV and KL-Potts by computing the entropy of the class prediction. As seen in Figure 3.2 that almost all errors occur for the 30% least confident points, illustrating the interest of this approach when precision is critical.

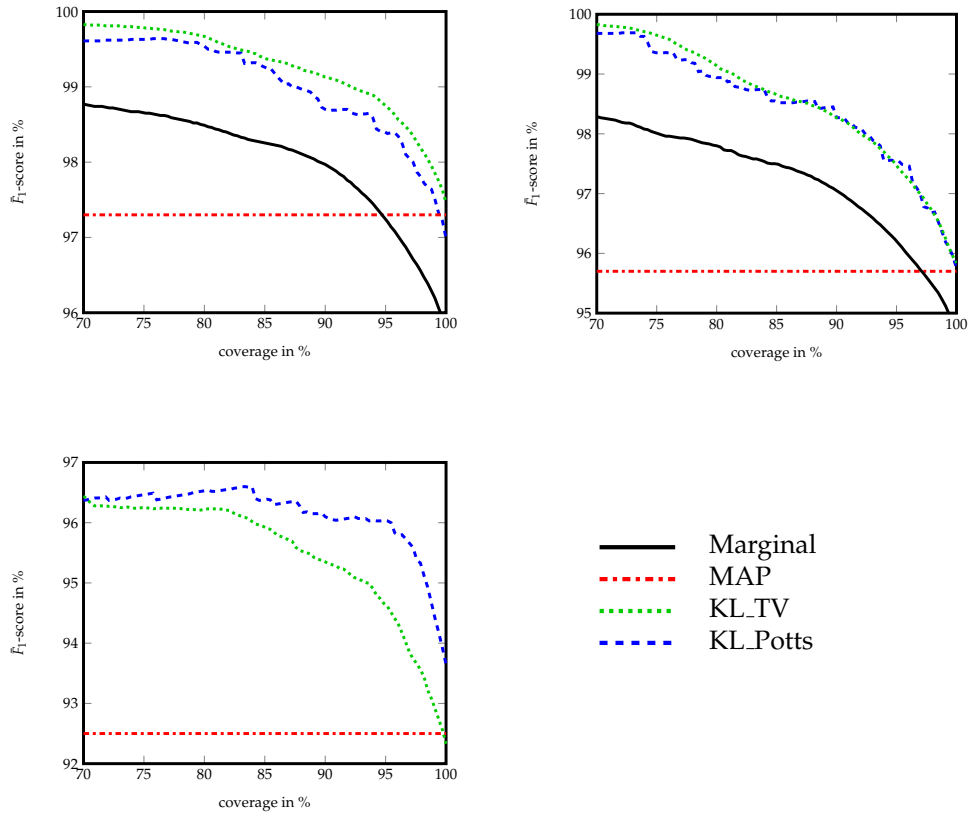


Figure 3.2: **Quantitative Results for Label Smoothing.** Accuracy/coverage plot: we sort the points by decreasing confidence for the Oakland-3C dataset (top left), the Oakland-5C dataset (top right) and the Paris-rue-Cassette database (bottom left). The precision reaches nearly 100% for probabilistic methods at coverage 70%, *i.e.* once removing the 30% least confident points

We propose a versatile optimization-based framework for spatially regularizing pointwise prediction on 3D point clouds. In particular, our approach increases the precision of the prediction while retaining the probabilistic nature of the pointwise prediction, which is critical for many applications.

2 The Superpoint Approach

Following the steps of image analysis, 3D point cloud analysis has now fully adopted the deep learning paradigm [GWH+20]. However, neural networks are limited in the size of the inputs they can handle simultane-

ously. LiDAR scans often reach several million points and require sliding window strategies [QYSG17, CGS19, TQD+19, CCHL20]. Although this is sufficient for problems such as indoor segmentation, the loss of global structure can be a limiting factor for large-scale geospatial data.

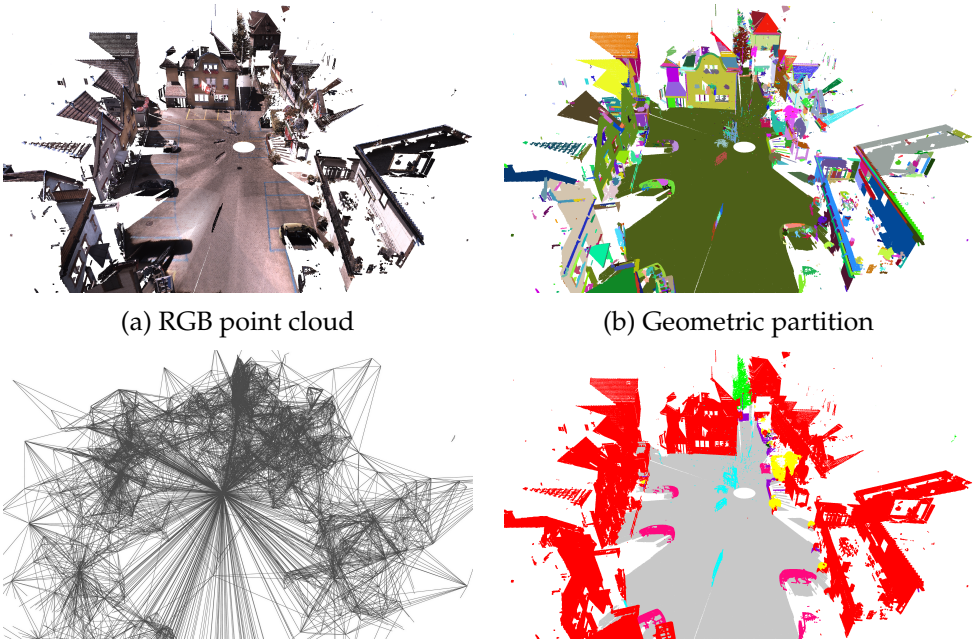
The semantic labels of 3D point clouds are generally regular, as explored in the previous section. Furthermore, the transitions between objects typically occur in regions with geometric or radiometric discontinuities. In other words, the interface between objects of different natures is often characterized by a sharp shift in shape and/or colour—or a learned combination of such features. Consequently, by grouping adjacent points with similar local geometry and radiometry, the resulting partition should also be semantically homogeneous: each component contains mainly the same semantic or instance labels. Instead of using this prior *ex post* to improve a noisy prediction, we propose to exploit this property algorithmically to improve processing efficiency. The resulting deep learning approach we propose can handle millions of points simultaneously.

Principle of SuperPoint Graph. We represent large 3D point clouds as collections of interconnected shapes called superpoints, in the spirit of superpixel methods for images [ASS+12]. As illustrated in Figure 3.3, we represent this structure with an attributed directed graph called the SuperPoint Graph (SPG). The vertices are the shapes, and the edges represent their adjacency relationship in 3D space. We also equip each edge with a descriptor of the nature of the adjacency between superpoints (*e.g.* max and average distance, interface size, size ratio).

Instead of considering individual points or voxels, we directly classify superpoints. Since the number of superpoints is typically several orders of magnitude smaller than the number of points, this allows us to consider large point clouds and model long-range interactions. The proposed SPG representation divides the semantic segmentation problem into three distinct problems of different complexity:

1 **Semantically homogeneous partition:** We first partition the point cloud into geometrically simple superpoints. We can easily derive the SPG from this partition.
2 **Superpoint embedding:** Since superpoints are geometrically simple, we can embed them with a lightweight point set network.
3 **Contextual segmentation:** The SPG typically contains a few hundred vertices and a few thousand edges at most, even for complex scenes. This small scale allows us to employ powerful graph convolution networks to leverage the context of superpoints.

Note that only the partitioning step considers the entire point cloud; all other steps operate directly on the superpoint graph. An essential hypothesis is that geometrically simple superpoints are semantically homogeneous,



(a) RGB point cloud (b) Geometric partition

(c) Superpoint graph (d) Semantic segmentation

Figure 3.3: **Superpoint Graph.** An input point cloud (a) is divided into geometrically simple shapes, called superpoints (b). We construct a graph by linking nearby superpoints with superedges (c). We then use graph convolutions to classify the superpoints according to their shape and context.

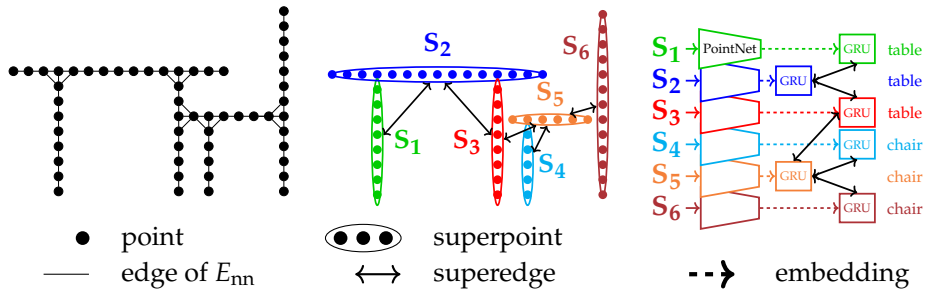
i.e. contain mostly points with the same label. In Section 2.1, we first assume that geometric and radiometric homogeneity implies semantic purity. We then present in Section 2.2 a method to directly learn to partition the point cloud into semantically homogeneous regions.

> By grouping points into geometrically and semantically homogeneous regions, we cast the semantic segmentation of a large 3D point cloud as a vertex classification problem on a small graph.

2.1 Geometrically Homogeneous Point Cloud Partition

This section details the partition of an input point cloud V into geometrically and radiometrically simple and contiguous regions. Our objective is not to retrieve individual instances, such as cars or chairs, but to break down objects into geometrically simple parts, as seen in Figure 3.3.

Geometrically Homogeneous Partition. We associate with each point a set of geometric features based on hand-made dimensionality features



(a) Input point cloud. (b) Superpoint graph (c) Network architecture

Figure 3.4: **Pipeline.** We illustrate our approach on a toy scene with a table and a chair. We partition the point cloud (a), which allows us to build the superpoint graph (b). Each superpoint is embedded with a PointNet network. The embeddings are then refined with message passing along superedges to produce the final labelling (c).

[DMDV11, GL17] and, if available, radiometric information (*e.g.* colour, intensity). We denote these features by $f \in \mathbf{R}^{d_v \times V}$. We also compute $G_{\text{nn}} = (V, E_{\text{nn}}, w)$ the k-nearest neighbours *adjacency graph* of the point cloud in 3D space (this is *not* the SPG). Each edge (u, v) in E_{nn} is attributed with a non-negative weight $w_{(u,v)} \in \mathbb{R}^+$ encoding the proximity of the points.

Partitioning amounts to computing a piecewise constant approximation of f with respect to the graph G_{nn}. We define the superpoints as the constant connected components of the solution of the following optimization problem:

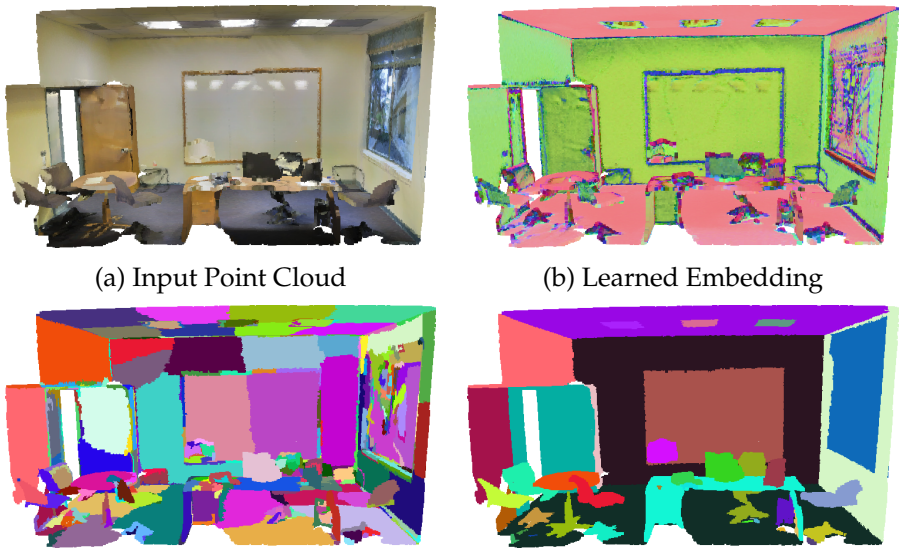
$$\underset{g \in \mathbb{R}^{d_v \times V}}{\text{argmin}} \sum_{v \in V} \|g_v - f_v\|^2 + \lambda \sum_{(u,v) \in E_{\text{nn}}} w_{u,v}[x_u - x_v] \,, \tag{3.2}$$

with $[x_u - x_v] = 0$ if $x_u = x_v$ and 1 else, and λ the regularization strength that determines the coarseness of the resulting partition. By construction, the recovered superpoints $\mathcal{S} = \{S_1, \cdots, S_k\}$ are geometrically and radiometrically homogeneous regions of the point cloud. The defined functional is neither convex nor continuous, making its exact minimization unrealistic for large point clouds. However, the ℓ_0-cut pursuit algorithm introduced in Chapter 2 can quickly find an approximate solution with a few graph cuts.

Computing SuperEdges. The SPG is an oriented attributed graph $\mathcal{G} = (\mathcal{S}, \mathcal{E}, F)$ whose vertices are the superpoints $\mathcal{S}$ and whose edges $\mathcal{E}$ (referred to as *superedges*) represent the adjacency between superpoints. We link superpoints whose points are connected in the Delaunay tetrahedralization [JT92, WH94] of the point cloud. Using this adjacency definition instead of

nearest neighbours allows us to model long-range relationships not typically captured by nearest neighbour graphs. We attribute each superedge with features that describe the adjacency relationships between the connected superpoints, such as the offset vector between their centroids and comparisons between the superpoint size and shapes. Note that the asymmetry in the superedge features makes the SPG a directed graph.

2.2 Learning to Partition 3D Point Clouds



(a) Input Point Cloud (b) Learned Embedding

(c) Oversegmentation (d) True Objects

Figure 3.5: Learning to Partition. We process a point cloud (a) with our point embedding network (b). We compute a piecewise constant partition of these embeddings (c), which closely follows the ground truth objects (d).

Not all interfaces between objects are characterized by a change in geometry or colour. For example, whiteboards and walls are flat, white, and vertical. The transition between these objects is subtle yet immediately recognizable to the human eye. Additionally, certain objects present heterogeneous radiometry and geometry: trees are composed of trunks (flat, brown, and vertical) and foliage (scattered, green, and volumic). In this section, we propose learning point descriptors with high contrast along object transitions; see Figure 3.5.

Graph-Structured Contrastive Loss. We denote by $\sigma : V \mapsto \mathbf{S}_m^V$ a neural network which extracts local geometric and radiometric features of 3D points, with $\mathbf{S}$ the m-dimensional hypersphere. This model is typically

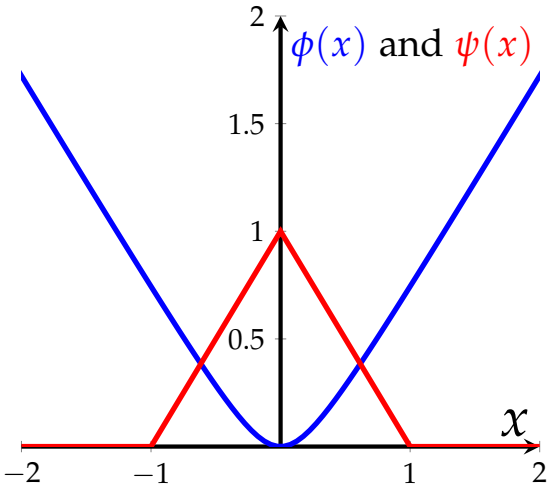


Figure 3.6: **Contrastive Functionals.** The homogeneity-promoting functional ϕ (in blue) and the contrastive functional ψ (in red) used in the graph-structured contrastive loss.

straightforward and operates with a limited receptive field (*e.g.* 50 nearest neighbours of each point). Our goal is for the vertex-valued function $f = \sigma(V)$ to be homogeneous within objects and to present high contrast at their interface. To this end, we define the set of *intra-edges* E_{intra} as the edges of E_{nn} connecting points within the same object. Equivalently, E_{inter} is the set of *inter-edges* and represents the transition between distinct objects.

We propose to learn f by minimizing the following graph-structured contrastive loss:

$$\ell(f)|E| := \sum_{(u,v)\in E_{\text{intra}}} \phi\left(f_u - f_v\right) + \sum_{(u,v)\in E_{\text{inter}}} \mu_{u,v}\psi\left(f_u - f_v\right) , \tag{3.3}$$

with ϕ (resp. ψ) a function minimal (resp. maximal) in 0, and $\mu_{u,v} \in \mathbb{R}^{E_{\text{inter}}}$ a weight on inter-edges. A point embedding function that minimizes this loss should be uniform within objects and have sharp contrast at their interface. Consequently, the components of the piecewise constant approximation of Eq. (3.2) should follow the objects' borders.

We chose ϕ, the function that promotes intra-object homogeneity as $\phi(x) := \delta(\sqrt{\|x\|^2/\delta^2 + 1} - 1)$ with $\delta = 0.3$. The first term of ℓ becomes the pseudo-Huber graph-total variation on the E_{intra} edges [H$^+$73, CBFAB97], and promotes the homogeneity of embeddings within the same object. With $\psi(x) = \max(1 - \|x\|, 0)$, the second part of ℓ penalizes vertices connected across objects' borders and that share similar embeddings. A point embedding constant within objects and with *enough* contrast between adjacent objects will have a loss of 0, which implies perfect segmentation with a well-chosen parameterization of Eq. (3.2). The four-colour theorem

[Gon08] tells us that such embedding can be achieved for hyperspheres with a dimension of 3 or more. However, because σ operates on point features, borders that do not present sufficiently distinct geometric or radiometric signatures will remain undetected.

Cross-Partition Weighting. Equation Eq. (3.3) views the problem of learning to partition as an edge classification problem. However, a single missed edge can erroneously fuse two large superpoints that cover different objects and drastically impact the quality of the partition. We propose to translate the influence of each edge on the resulting partition into a set of inter-edge weights $\mu_{u,v}$, see Figure 3.7.

We compute the partition $\mathcal{S}$ obtained by the piecewise constant approximation of the point features f as defined by Eq. (3.2). We define the cross-partition $\mathcal{V}_{\text{cross}}$ between $\mathcal{S}$ and the true partition $\mathcal{O}$ of V into objects: $\mathcal{V}_{\text{cross}} := \{O \cap S \mid O \in \mathcal{O}, S \in \mathcal{S}\}$. For all inter-edge $(u, v) \in E_{\text{inter}}$, we denote by U (resp. V) the element of $\mathcal{V}_{\text{cross}}$ containing u (resp. v) and associate the following edge weight:

$$\mu_{u,v} := \mu_0 \frac{\min\left(|U|, |V|\right)}{|\ U \times V \cap E_{\text{inter}}\ |}\,, \tag{3.4}$$

with μ_0 a normalizing parameter. The rationale of this formula is the following: since (u, v) is an inter-edge, u and v belong to different objects. If (u, v) is not detected as a transition, then U and V will be erroneously merged Since U and V cover different objects, this would incur at least $\min\left(|\ U\ |, |\ V\ |\right)$ *trespassing* vertices, *i.e.* vertices that do not belong to the same object as the majority of the vertices of their superpoint. The weights are also divided by the size of the interface between U and V to evenly distribute the penalty over the number of edges that make up an interface. This prevents long borders from being overrepresented in the loss and allows the network to handle *choke points*.

We partition large point clouds into semantically homogeneous regions by computing a piecewise constant approximation of local point descriptors on an adjacency graph. These descriptors can be either hand-crafted or learned to present high contrast at the interface between objects.

2.3 SuperPoint Classification

After constructing the superpoint graph, our goal is to assign a semantic label to each superpoint based on its shape and the global context of the scene, see Figure 3.4.

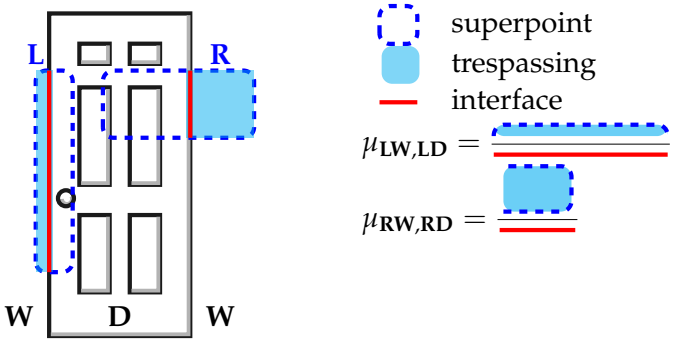


Figure 3.7: **Cross-Partition Weighting.** Scene comprised of a door (**D**) and a wall (**W**). Two superpoints **L** (left) and **R** (right) overlap the door. The edges of $(\mathbf{LW}, \mathbf{LD})$ (resp. $(\mathbf{RW}, \mathbf{RD})$) connect the vertices of the left (resp. right) superpoint that belong to the wall and the door. With fewer trespassing vertices and a longer interface than $(\mathbf{RW}, \mathbf{RD})$, the weights of $(\mathbf{LW}, \mathbf{LD})$ are smaller.

Superpoint Embedding. Since superpoints are geometrically simple by construction, we sub-sample them on-the-fly to $n_p = 128$ points and normalize them to the unit cube. This adaptive sampling increases the computational and memory efficiency, and doubles as a powerful data augmentation. We then embed each superpoint $U \in \mathcal{S}$ independently and in parallel into a vector z_U of size $n_z = 32$ with a simple shape-embedding network $\pi : \mathbf{R}^{3 \times n_p} \mapsto \mathbf{R}^{n_z}$ such as PointNet [QSMG17b]. To take the superpoint size into account, we concatenate its diameter to the maxpooled shape vector.

Contextual Segmentation. In order for each superpoint to refine its embedding based on its local context, we employ a message-passing scheme inspired by Edge-Conditioned Convolutions (ECC) [SK17]:

(i) We initialize the state of each superpoint U with the shape embedding $z_U^{(0)} = z_U$.

(ii) Each superedge (U, V) is assigned a filter $\Theta_{U,V} \in \mathbf{R}^{n_z}$ mapped from its features by an MLP.

(iii) Each superpoint U sends a message $\Theta_{U,V} \odot z_V^{(t)}$ to each of its neighbours $V \in N_U$ in the SPG, consisting of its current state modulated by the edge filter.

(iv) Once all messages are sent, each superpoint U combines the incoming messages and updates its current state accordingly.

(v) After a set number of $(iii) \mapsto (iv)$ steps, the last current state is mapped to a vector of class scores by another MLP.

The following equation summarizes the main mechanism behind this scheme:

$$z_U^{(t+1)} = \text{update}\left(z_U^{(t)}, \text{pool}\left(\left\{\Theta_{U,V} \odot z_V^{(t)}\right\}_{V \in N_U}\right)\right), \tag{3.5}$$

with pool the averaging operator, $\odot$ the Hadamard product, and update a modified Gated Recurrent Unit (GRU) [CvMG+14]. We added a gating mechanism allowing the recurrent network to block parts of the input vector depending on its current state. The GRU can ignore specific channels in straightforward cases, freeing their usage for ambiguous situations. Note that the superpoint and superedge embedding networks, as well as the GRU, are all trained end-to-end simultaneously.

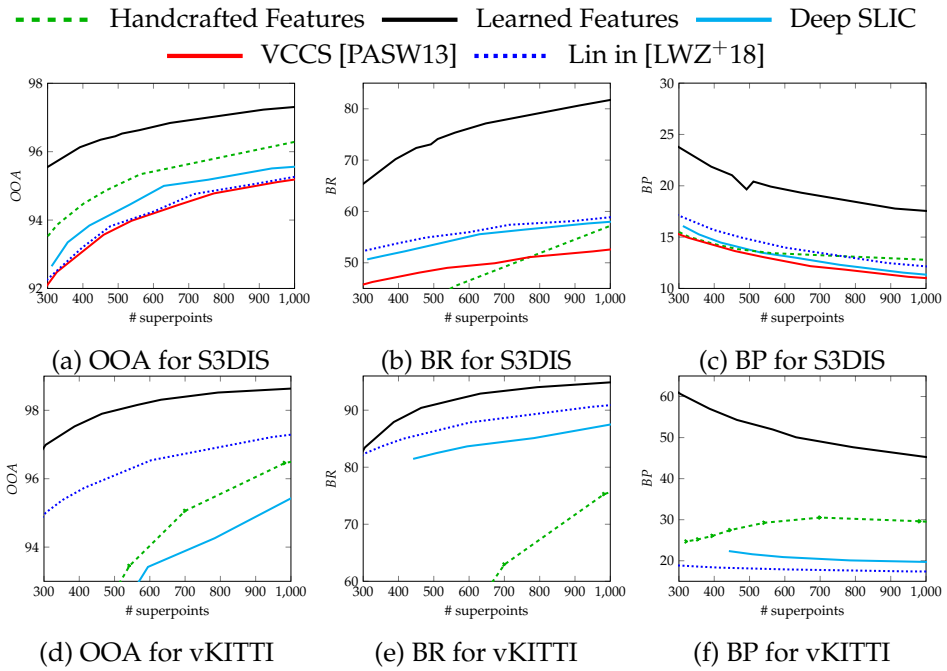
This message passing scheme is similar to some inference methods used in CRFs, such as Loopy Belief Propagation [MWJ13], or its numerous deep learning-based reformulations [ZJR+15, SU15, LSvdHR16, CK16, LKZ+17]. The main difference is that the messages operate directly on learned representations instead of relating to class compatibility. This allows the network to leverage weak unaries or to discover latent sub-classes corresponding to object parts, *e.g.* the legs of tables and chairs.

2.4 Numerical Experiments

Datasets. We evaluate the performance of our 3D point cloud analysis algorithm on three datasets of different natures:

- **S3DIS [ASZ+16]**. This indoor dataset of office buildings contains more than 278 million semantically annotated 3D coloured points in 6 building areas and annotated with 13 classes and individual instances.
- **Semantic3D [HSL+17].** This large outdoor dense RGB LiDAR dataset consists of over 3 billion points from various urban scenes. The dataset consists of 15 training scans and 15 test scans with withheld labels and a reduced test set of 4 subsampled scans.
- **vKITTI3D [GWCV16, EKHL17].** This autonomous driving dataset is made up of virtual LiDAR scans [GWCV16] with point annotations [EKHL17].

Point Cloud Oversegmentation. We first evaluate the partition quality obtained from hand-crafted and learned features. We compare our results with two state-of-the-art superpoint methods: VCCS [PASW13] and the method of Lin *et. al* [LWZ+18]. We also implemented a 3D version of the deep superpixel algorithms proposed by [JSL+18] and inspired by SLIC superpixels [ASS+12]. We propose three metrics to evaluate the quality of a partition. Oracle Overall Accuracy (OOA), which is the accuracy obtained when associating each superpoint to its majority label; Border Recall and



(a) OOA for S3DIS (b) BR for S3DIS (c) BP for S3DIS

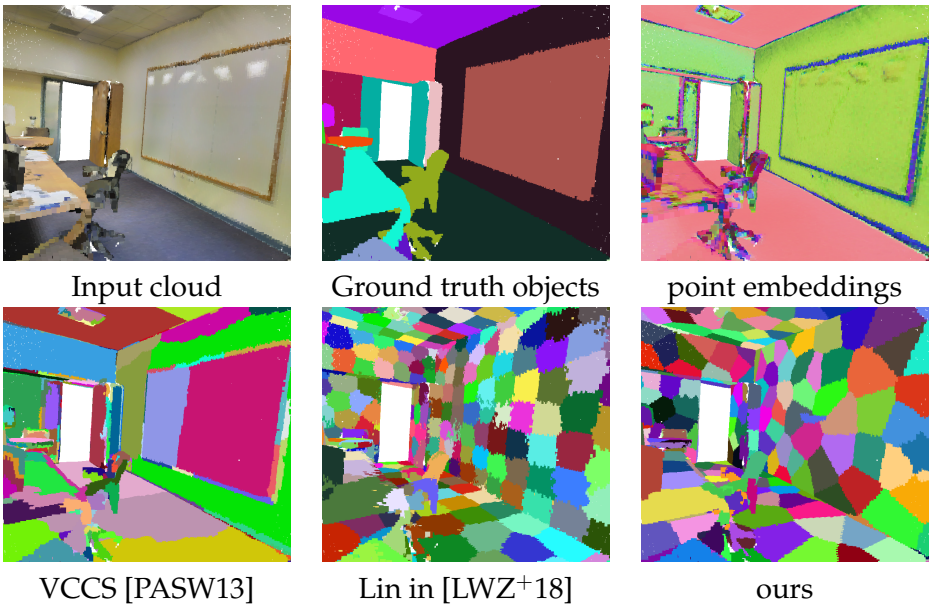
(d) OOA for vKITTI (e) BR for vKITTI (f) BP for vKITTI

Figure 3.8: Quantitative Oversegmentation Results. Performance of the different algorithms on the 6-fold S3DIS dataset (a, b, c), and the 6-fold vKITTI dataset (d, e, f). SSP-Cluster and VCCS are not represented for vKITTI for the sake of legibility as their performance is too low.

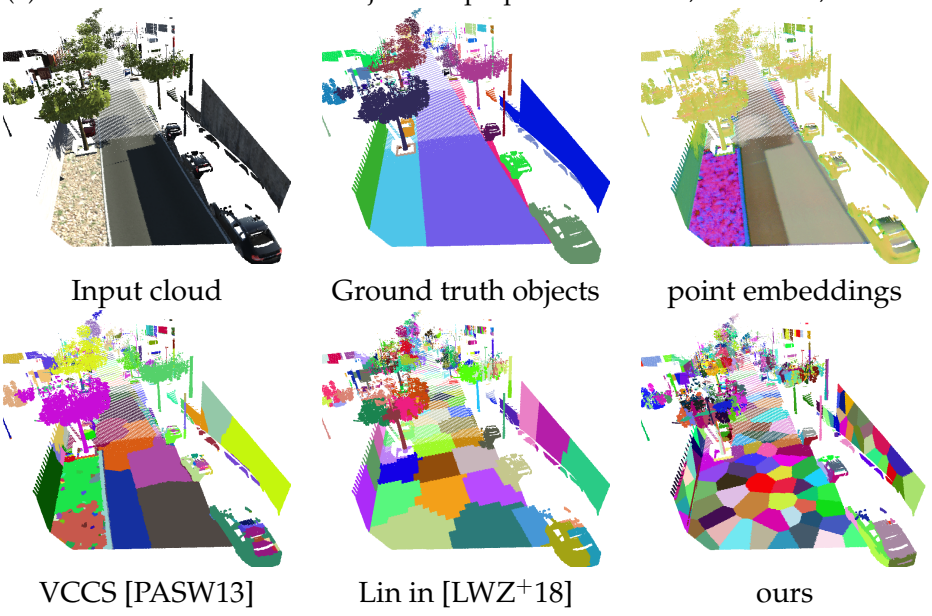
Precision (BR and BP), assessing how well the contour of objects is correctly recovered (with a set tolerance).

In Figure 3.8, we present the results of the different oversegmentation approaches. Using learned features leads to partitions requiring five times fewer superpoints than competing methods to reach the same purity and border quality. We can observe in the qualitative results displayed in Figure 3.9 that networks learn features with high contrast around objects, even ones with subtle interfaces such as whiteboards and white walls. Furthermore, the features are homogeneous within objects even when the radiometry and geometry change due to light reflections or composite objects such as trees.

Point Cloud Semantic Segmentation. We report in Table 3.1 the performance of SPG, with and without learned partition and for different datasets. Since our approach classifies superpoints and not points, its precision is bound by the quality of the partition. However, with only a fraction of the parameters of large and complex neural networks such as KPConv [TQD+19], or MinkowskiNet [CGS19], our method achieves comparable performance. Learning the partition provides a consistent improve-



(a) S3DIS scene with 58 true objects. Superpoints : SSP 442, VCCS 436, Lin 423.



(b) vKITTI scene with true 233 objects. Superpoints: SSP 420, VCCS 422, Lin 425.

Figure 3.9: **Qualitative Oversegmentation Results.** Our learned point features are homogeneous within objects and present high contrast at their border, allowing for their precise segmentation.

ment, with few additional parameters (under 15k). SPG training epochs are at least an order of magnitude faster than its competitors, and the entire model can be trained in under 2h on a single commercial GPU, compared

to days on GPU clusters for the most demanding networks.

Table 3.1: **Quantitative Semantic Segmentation Results.** Results given in % of mIoU over several datasets.

Model	**Size** $\times 10^6$	**S3DIS** 6Fold	**Semantic3D** Red/Full	**vKITTI3D**
PointNet [QSMG17a]	3.5	47.6	-	34.4
PointNet++ [QYSG17]	12.4	54.5	-	-
SPG [LS18]	0.25	62.1	73.2/76.2	55.4
MinkoNet [CGS19]	31	65.9	-	-
ConvPoint [Bou19]	2.8	68.2	-/76.5	-
SSP + SPG [LB19]	0.25	68.4	-	57.0
SPNet [HYC+21]	-	68.7	-	57.0
RandLA-Net [HYX+20a]	1.2	70.0	77.4/-	-
KPConv [TQD+19]	14.9	70.6	-	-

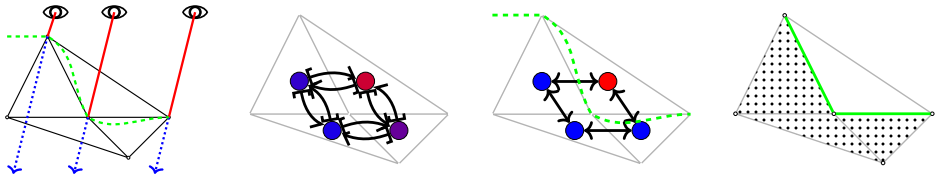
3 Large-Scale Surface Reconstruction

Most surface reconstruction methods rely either on visibility-based energy formulations [BRV16, CBV17, JP11, JP14, VLPK12, ZSH19, LPK09] or shape-based deep learning approaches [GFK+18, YFST18, SO20, LZS20, DGF+19]. Energy-based methods are typically scalable and robust, and provide useful topological guarantees. Deep learning approaches can learn surface priors directly from training data leading to high-performance in-distribution. This section presents a hybrid method in which a graph neural network predicts the parameter of a graph-cut problem, combining the advantages of both approaches, see Figure 3.10.

3.1 Tetrahedralization-based Surface Reconstruction

The point density of a 3D point cloud is highest near the surface of the scanned objects. A standard approach to exploit this regularity is to discretize the 3D space with a Delaunay tetrahedralization [CG+04] and directly classify each cell as inside or outside. Indeed, the size of the resulting cells is inversely proportional to the density of points, simplifying the decision for large empty regions of space. The interface between the inside and outside cells determines the predicted surface.

We denote by $\mathcal{T}$ the set of tetrahedron partitioning the 3D space, and $\mathcal{E} \subset \mathcal{T} \times \mathcal{T}$ the edges connecting cells with a shared facet. We aim to find a labelling l_t in $\{0, 1\}$ for each tetrahedron t, with 0 meaning outside and 1 inside. Each cell is attributed a potential U_t quantifying its likelihood of



(a) Delaunay triangulation and ray casting.

(b) Local and contextual learning with gnn.

(c) Global optimization with graph cuts.

(d) Surface reconstruction.

Figure 3.10: **Pipeline.** We discretize the input point cloud (a) and compute tetrahedron visibility information from lines of sight —⋯→ and from camera positions 👁. We use a graph neural network (b) to predict an occupancy score for each tetrahedron, which in turn define an energy (c) leading to a minimal cut - - - in an adapted flow graph. The reconstructed surface (d) — is the interface between cells with different labels.

being inside based on visibility considerations, *i.e.* cameras are outside, and the immediate continuation of a line-of-sight after impact is inside. We also define *binary potential* $B_{s,t}$, whose value indicates the propensity of a facet to be part of the sought surface. The optimal label assignment minimizes the following energy:

$$E(l) = \sum_{t \in \mathcal{T}} U_t(l_t) + \lambda \sum_{(s,t) \in \mathcal{E}} B_{s,t}(l_s, l_t) \,, \tag{3.6}$$

where $\lambda \geq 0$ is the regularization strength. We can find a global minimizer of the energy E by computing a minimum cut in an appropriate flow graph [BVZ01] or using a linear programming approach [BdLGM14]. Of course, the quality of the resulting surface highly depends on the relevance of the unary and binary terms. Handcrafted potentials are usually quick to compute but may be insufficient in complex cases.

We can exploit the irregular density of 3D scans with a partition of space that is adaptive to the point density. We then use a hybrid approach to classify each cell as inside or outside. Our method benefits from the scalability and robustness of energy-based models and the adaptability of deep learning.

3.2 Visibility-Based Occupancy Prediction

We propose to use a graph neural network to predict unary potentials instead of using handcrafted values. We first associate each cell with a set of visibility features based on ray traversal from the sensor positions. We then use the scalable Graph-SAGE [HYL17] graph-convolution scheme to

Table 3.2: **Object Reconstruction.** Evaluated on shapes from Berger *et. al* [BLN+13] with different scanning settings, our approach outperforms both traditional and deep approaches despite only using a fraction of the training set of ConvONet [PNM+20].

Method	**Chamfer [↓] distance** (point ave. %)	**Volumetric [↑] IoU (%)**	**Number [↓] of components**
ConvONet [PNM+20]	2.53	64.1	7.6
IGR [GYH+20]	5.13	62.6	38.3
Poisson [KH13]	0.74	86.1	7.8
Labatut et al. [LPK09]	0.72	86.4	2.0
Ours	**0.65**	**88.5**	**1.1**

leverage information from adjacent cells and predict for each t a pair of class scores i_t, o_t. These scores can be combined to form a soft occupancy prediction with a softmax function. This network is supervized with the Kullback-Leibler divergence between the predicted and true occupancy ratio of each cell. Finally, we define the unary terms according to the occupancy scores:

$$U(l_t) = i_t \left[l_t = 0\right] + o_t \left[l_t = 1\right], \tag{3.7}$$

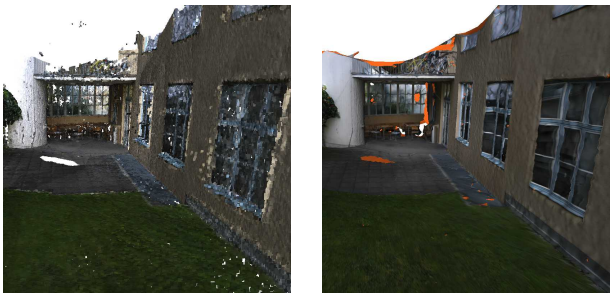
with $[x = y]$ the Iverson bracket, equal to 1 if $x = y$ and 0 otherwise.

3.3 Numerical Experiments

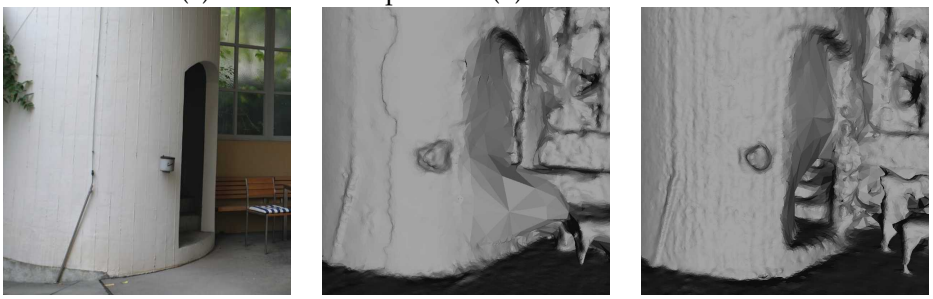
We present numerical experiments to show the performance of our reconstruction method for both objects and large-scale scenes. In both settings, our method is only trained on a small synthetic dataset (130 shapes from ShapeNet [CFG+15], artificially scanned) and yet outperforms state-of-the-art learning and non-learning-based methods, highlighting its capacity for generalization.

Object Reconstruction. We scan virtually 5 shapes from Berger *et. al* [BLN+13], each with 5 noise, resolution, and outlier ratio settings. As seen in Table 3.2, our approach outperforms both traditional and deep learning approaches trained on the entirety of ShapeNet (50k objects).

Scene Reconstruction. Even though we only train our model on a small set of synthetic shapes, it can generalize to large real-life scenes, see Figure 3.11. Our method consistently outperforms traditional methods in terms of precision. It is also faster and more memory efficient than deep learning methods such as ConvONet.



(a) Dense MVS input. (b) Our textured mesh.



(c) Details Image. (d) [JP14]. (e) Ours.

Figure 3.11: **Qualitative Results on ETH3D.** Our mesh reconstruction method takes as input a dense MVS point cloud (a) and produces a mesh (b), simultaneously preserving fine details and completing missing parts (here textured with [WMG14]). We represent: in (c), a cropped image of a detail from the *terrace* scene of the ETH3D benchmark [SSG+17b]; in (d), the reconstruction by Jancosek and *et. al* [JP14]; and in (e), our reconstruction. Notice the missing staircase and spurious vertical pattern on the concrete wall in (d). In contrast, our method (e) reconstructs part of the staircase as well as the fine-grained wall textures.

Conclusion

We presented three methods to leverage the underlying structure of 3D point clouds for three tasks: (i) improving pointwise predictions, (ii) analyzing large point clouds, (iii) large-scale surface reconstruction. By exploiting spatial regularity, our approaches resulted in higher precision, reduction in model size, faster training, and higher generalization.

CHAPTER 4

Exploiting the Structure of 3D Sensors

This chapter presents four methods that exploit the acquisition structure of 3D data for added speed and precision. We study the structure of LiDAR time sequences and show how a sensor-aware method can reach real-time speed without sacrificing accuracy. We then propose an end-to-end approach for merging images and raw 3D point clouds which defines a new state-of-the-art with minimalistic pre-processing. We introduce a simple way to consider the acquisition geometry of 3D scans to infuse visibility information into any deep surface reconstruction methods. Finally, we present a deep learning approach for automated forest inventory that use the ability of LiDAR to penetrate the tree canopy to predict a multi-layer vegetation structure.

This chapter is based on the following publications:

[LAL22]: Romain Loiseau, Mathieu Aubry, Loic Landrieu, "Online Segmentation of LiDAR Sequences: Dataset and Algorithm", *ECCV*, 2022

[RVL22]: Damien Robert, Bruno Vallet, Loic Landrieu, "Learning Multi-View Aggregation In the Wild for Large-Scale 3D Semantic Segmentation", *CVPR*, 2022

[SLB+22]: Raphael Sulzer, Loic Landrieu, Alexandre Boulch, Renaud Marlet, Bruno Vallet, "Deep Surface Reconstruction from Point Clouds with Visibility Information", *ICPR*, 2022

[KLMC22a]: Ekaterina Kalinicheva, Loic Landrieu, Clément Mallet, Nesrine Chehata, "Multi-Layer Modeling of Dense Vegetation from Aerial LiDAR Scans", *CVPR Workshops*, 2022

[KLMC22b]: Ekaterina Kalinicheva, Loic Landrieu, Clément Mallet, Nesrine Chehata, "Predicting Vegetation Stratum Occupancy from Airborne LiDAR Data with Deep Learning", *Journal of Applied Earth Observation and Geoinformation*, 2022

1 Online LiDAR Segmentation

Roof-mounted spinning LiDAR sensors are widely used by autonomous vehicles [RBG19]. Most semantic datasets [BGM+19, JOWS21, LXG21] and algorithms used for LiDAR sequence segmentation operate on frames corresponding to a 360° degree arc around the sensor. This incurs an acquisition latency of over 100ms, which is incompatible with real-time applications. To remedy this limitation, we introduce HelixNet, the largest dataset of LiDAR time sequences with fine-grained point information, allowing for precise latency estimation. We also introduce Helix4D, a spatio-temporal transformer architecture operating on *slices* of acquisition corresponding to a fraction of a complete sensor rotation, significantly reducing the latency without hampering precision, see Figure 4.1.

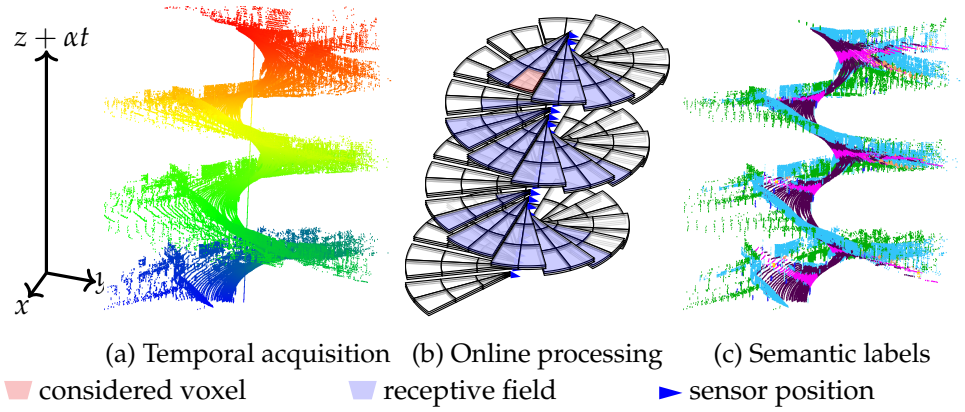


Figure 4.1: **Online LiDAR Segmentation**. The 3D point sequences of rotating LiDAR follow a complex helix-like structure in space and time, represented in (a). We propose an efficient spatio-temporal transformer operating on angular slices centred on the sensor's position (b). The slices are partitioned into voxels, each gathering information with voxels from past slices to build a large spatio-temporal receptive field. Our proposed model can segment the LiDAR point stream (c) with state-of-the-art accuracy and in real-time.

1.1 HelixNet: A Dataset for Online LiDAR Segmentation

We introduce HelixNet, a large-scale, open-access LiDAR benchmark for real-time semantic segmentation algorithms. HelixNet contains 20 3D sequences from the STEREOPOLIS II dataset [PPC+12], corresponding to 10 billion 3D points across 78 800 frames and 8.85 billion individual labels. As shown in Table 4.1, HelixNet is the largest densely annotated open-access rotating LiDAR dataset by a significant margin.

Table 4.1: **Embarked LiDAR Datasets with Semantic Point Annotations.** With over 8.8B annotated 3D points, HelixNet is 70% larger than SemanticKITTI, and includes diverse scenes spanning 6 different French cities. HelixNet arranges points with respect to the sensor rotation and contains fine-grained information about their release time.

Dataset	labels	frames	classes	span	format
HelixNet (Ours)	8.85B	78k	9	6 cities	sensor rotation
SemanticKITTI [BGM+19]	5.2B	43k	19	1 city	frame
Rellis3D [JOWS21]	1.5B	13k	16	1 city	frame
KITTI-360 [LXG21]	1.0B	81k	37	1 city	frame
A2D2 [GKM+20]	387M	41k	38	3 cities	frames
Paris-Lille-3D [RDG18]	143M	N/A	50	2 cities	multi-frame
Toronto3D [TQM+20]	78M	N/A	8	1 city	multi-frame

We format the sequences to closely follow the LiDAR data stream released by the sensor. More specifically, points are grouped by packets sharing the same *release time*, *i.e.* the moment they are made available to the segmentation algorithm. We also associate each point with the position and angle of the sensor, rather than only once per frame for SemanticKITTI [BGM+19]. This fine-grained information is critical to measuring the acquisition latency (acquisition to release) and inference latency (release to classification). The fine-grained information about the sensor position allows us to group the points into meaningful acquisition slices, a critical step for the approach detailed in the next section.

1.2 Helix4D: Fast LiDAR Segmentation with Transformers

We consider a sequence of 3D points acquired by a rotating LiDAR on a mobile platform, which we split into chronologically ordered slices of acquisition. As represented in Figure 4.3, we process each slice with a U-Net architecture [RFB15] with cylindrical convolutions [ZZW+21]. At the lowest resolution, a spatio-temporal transformer network connects neighbouring voxels in space and time, resulting in a large receptive field.

Temporal Slicing Instead of processing the data frame-by-frame, we propose to split the sequence into slices covering a fixed portion $\Delta\theta \in]0, 2\pi]$ of the sensor rotation, resulting in a shorter acquisition time and lower latency. Choosing $\Delta\theta = 2\pi$ corresponds to the classic frame-by-frame setting and implies an acquisition latency of 104ms in HelixNet or SemanticKITTI [BGM+19]. A slice size of $\Delta\theta = 2\pi/5$ leads to an acquisition latency of

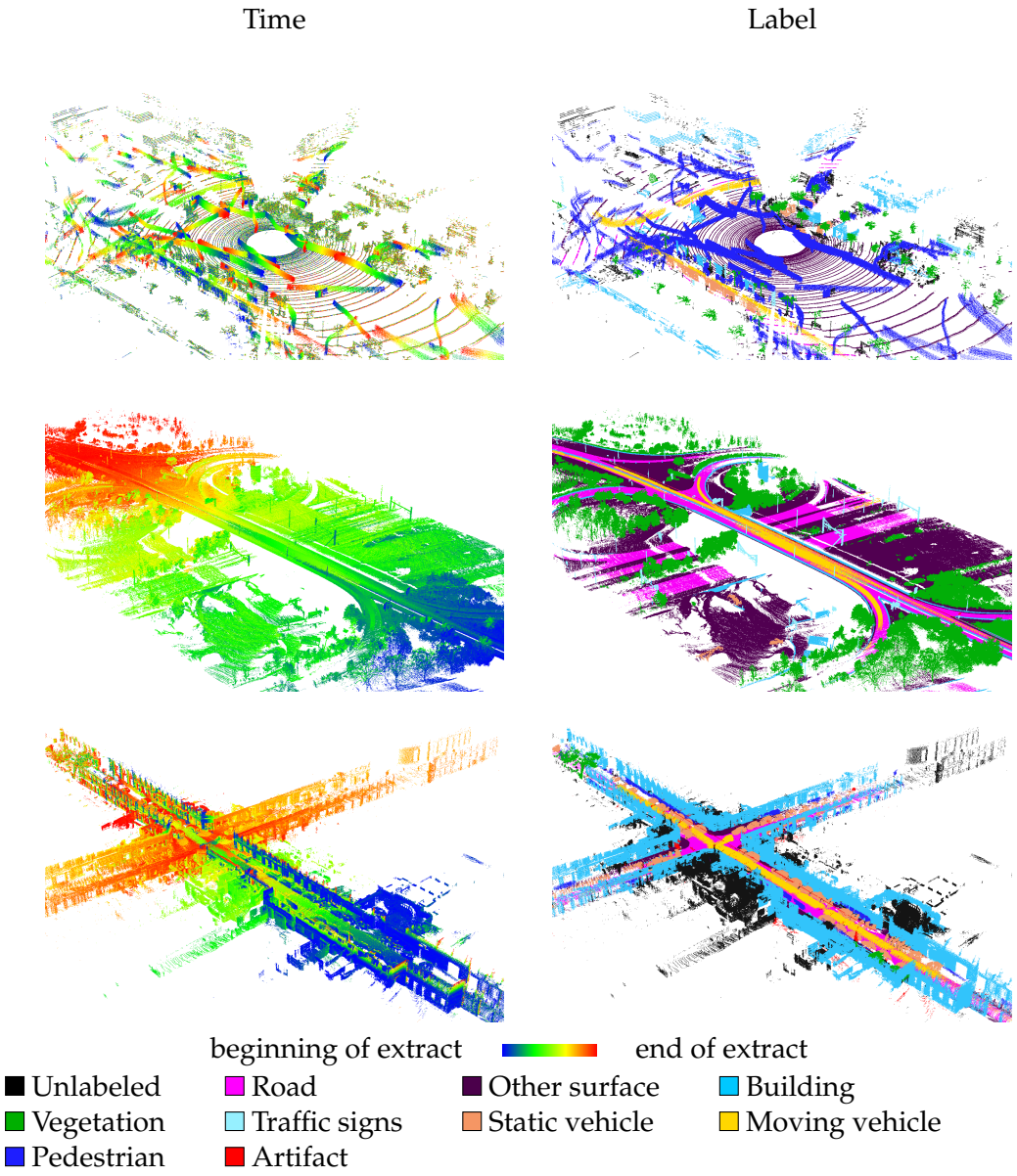


Figure 4.2: **Extracts from HelixNet.** Our proposed dataset contains various urban scenes from motorways to pedestrian plazas and historical centers.

21ms, which is more conducive to the real-time processing of driving data.

Spatio-Temporal Transformer Inspired by the Cylinder3D model [ZZW+21], we discretize into grids of increasing coarseness. We use a cylindrical convolutional encoder to produce feature maps at the lowest resolution. These features are processed by a spatio-temporal

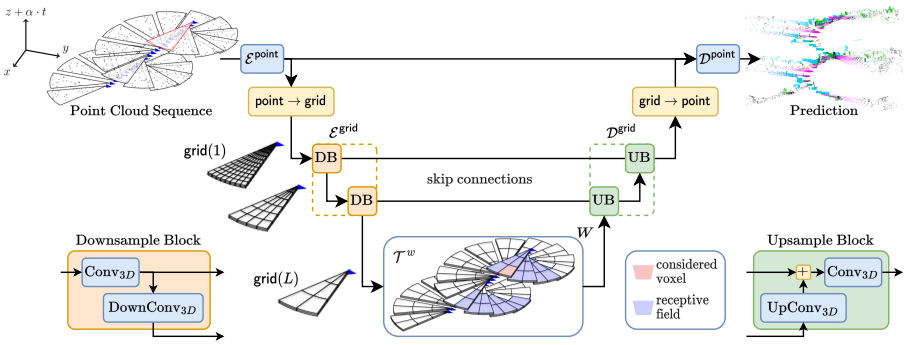


Figure 4.3: **Helix4D Architecture.** A point sequence is split into angular slices and encoded with a cylindrical convolutional encoder $\mathcal{E}^{\text{grid}}$. At the lowest resolution, we apply W consecutive spatio-temporal transformer blocks $\mathcal{T}^w$ with attention spanning current and past slices. The resulting features are up-sampled to full resolution with a convolutional decoder $\mathcal{D}^{\text{grid}}$ and used to classify the individual points.

transformer and, in turn, upsampled to full resolution with a convolutional decoder. The points are then individually classified to produce a semantic segmentation of the sequence. Our simplified architecture results in a lighter computational and memory load but can still learn rich spatio-temporal features thanks to the addition of the transformer module described below.

We consider all non-empty voxels at the lowest resolution, forming a non-strictly ordered time sequence. We apply to each voxel W independent transformer blocks $\mathcal{T}^1, \cdots, \mathcal{T}^W$ successively. For added efficiency, the transformers use a sparse attention scheme by only comparing each voxel v with other voxels within a mask determined by a spatial radius R and a set of rotation offsets $P \subset \mathbb{N}$. In the context of autonomous driving, we choose $R = 6\text{m}$ and $P = \{0, 5, 10\}$, corresponding to slices 0.5 and 1 seconds in the past along with the current one. We encode the relative position between voxels based on their spatio-temporal offset in the manner of Wu *et. al* [WPC+21].

We design a simplified transformer architecture [VSP+17] with only two learnable modules: a linear block generating the keys and values and the relative positional encoding. We save further computation at inference time by storing in memory the keys, values, and absolute positions of the voxels in past slices with a fixed buffer of $\max(P)$ rotations. This allows us to allocate a large spatio-temporal receptive field to each voxel without supplementary computations.

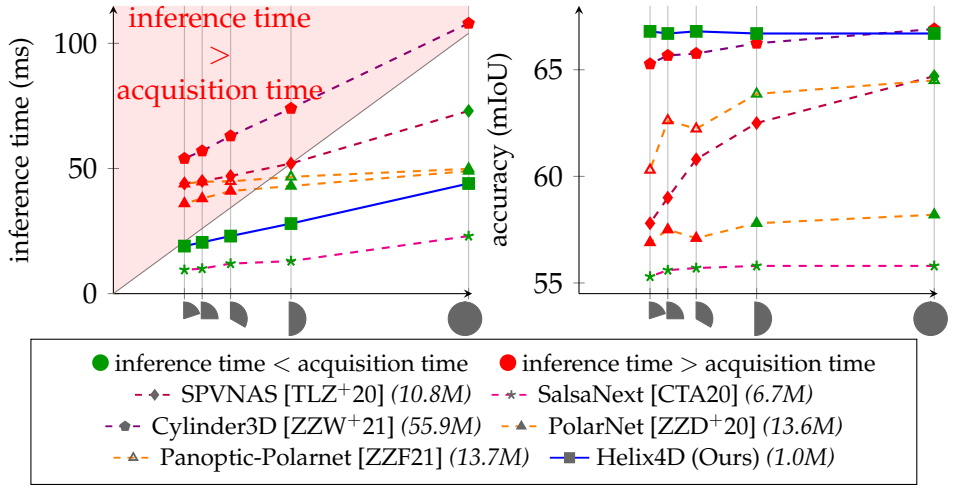


Figure 4.4: **Influence of Slice Size.** We plot the processing time (left, in ms) and precision (right, in mIoU) of different methods for the considered slices sizes, estimated on the validation set of SemanticKITTI [BGM+19]. Methods whose inference time is slower than the acquisition time of the slice (red shaded area) do not meet the real-time requirement.

1.3 Numerical Experiments

Evaluating Online Semantic Segmentation We evaluate the performance and inference time for our approach and other state-of-the-art methods on our proposed dataset HelixNet and the standard SemanticKITTI dataset. We consider different slice sizes, from $\Delta\theta = 2\pi/5$ to $\Delta\theta = 2\pi$, *i.e.* frame-by-frame. We measure the inference latency of a segmentation method as the average time between the release of the last point of a slice and its segmentation. To meet the real-time requirement, the classification of a slice must be faster than its acquisition, as slower processing would cause the prediction module to continuously fall behind. Although thinner slices directly reduce acquisition latency, they also make the real-time requirement more strict: a complete turn must be processed in less than 104ms, and a fifth turn must be in at most 21ms.

Because SemanticKITTI [BGM+19, GLSU13] lacks the pointwise information we provide in HelixNet, we cannot directly run semantic segmentation algorithms in the online setting. This forces us to make several approximations concerning the laser alignments, the movement of the vehicle, the rotation speed of the sensor, and the release time of the points. We must also adapt existing frame-by-frame methods to process slice and not full frames. We propose to process the point clouds corresponding to each slice independently and sequentially. This approach restricts the receptive field to the extent of the slices. However, since the sensor moves, the relative

positions given in past slices whose may no longer be valid. By explicitly modelling the spatio-temporal offset between voxels, Helix4D does not suffer from this limitation.

Analysis. In Figure 4.4, we report the inference time and mIoU for different slice sizes. Due to its design, the performance of Helix4D is not affected by the slice size. In contrast, competing methods perform worse with smaller slices. Helix4D yields state-of-the-art accuracy in the frame-by-frame setting, with mIoU scores only matched by Cylinder3D [ZZW+21]. However, Cylinder3D has 50 times more parameters and is twice slower, failing to meet the real-time requirement.

Only two approaches can perform real-time classification of slices with $\Delta\theta \leq 2\pi/3$: SalsaNeXt [CTA20] and Helix4D. Our approach outperforms SalsaNeXt by over 10 mIoU points for both frame and slices. Our model's total latency (acquisition plus inference time) in the online setting is 40ms ($21 + 19$ms). However, it reaches the same performance as Cylinder3D evaluated on full frame with a latency of 212ms ($104 + 108$ms), an acceleration of more than 5 folds. In short, Helix4D is as accurate as the largest and slowest models, with an inference speed comparable to that of the fastest and least accurate models.

We exploit the helix-like structure of point cloud sequences acquired with a rotating LiDAR mounted on a mobile platform. Instead of waiting for a complete sensor rotation to start inference, we process slices of rotation, drastically decreasing the acquisition and processing latency. Evaluated on our proposed dataset and a classic benchmark, our approach reaches the performance of state-of-the-art methods with a latency reduced by $5\times$ and a model size reduced by $50\times$.

2 Image and LiDAR Fusion

LiDAR scans capture geometric information with high precision; images provide rich textural and contextual cues. We can exploit this complementarity by processing each modality with a dedicated network and projecting onto 3D points the 2D features learned from real [DN18, HZJ+21, JGS19] or virtual images [KYF+20, CLLH19]. However, merging large-scale point clouds and images raises challenges such as mapping pixels with points and aggregating features between multiple views. Current methods rely on a costly mesh reconstruction or specialized sensors to recover occlusions, and use heuristics to merge available image information. In contrast, we propose an end-to-end trainable multiview aggregation model to combine features from images taken at arbitrary positions.

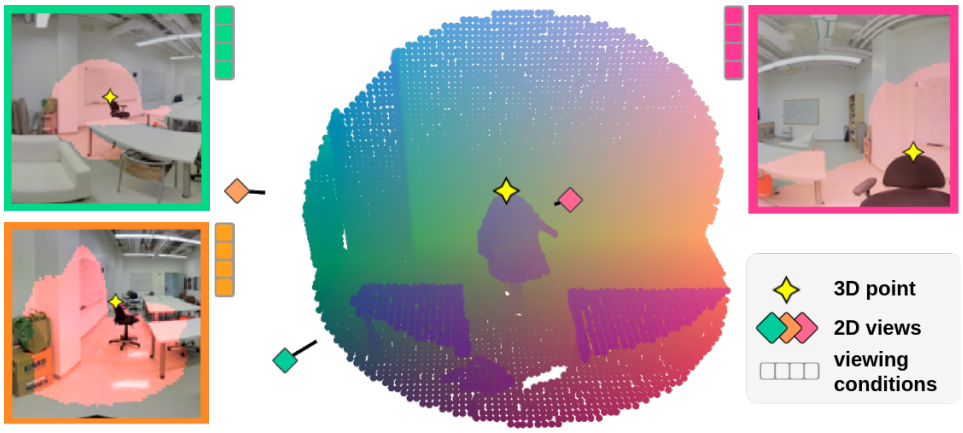


Figure 4.5: **Multiview Information.** A 3D point ✦ is seen in several images with different insights. Here, the **green** image contains contextual information, while the **pink** image captures the local texture. The **orange** image sees the point at a slanted angle and may not contain additional relevant information.

2.1 Deep Multi-View Aggregation

We consider P a set of 3D points and I a collection of co-registered images from the same scene. We seek to exploit the correspondence between 3D points and 2D pixels to perform 3D point cloud semantic segmentation with features learned from both modalities. As illustrated in Figure 4.5, the conditions under which a pixel sees a 3D point heavily influence the quality of the image information. Intuitively, closer images contain textural information, while far-away images are more informative about the context. On the other hand, a slanted viewing angle may result in unreliable or irrelevant 2D features. We propose a method to learn to aggregate relevant image features onto 3D point clouds based on such viewing conditions.

Point-Pixel Mapping. Combining point clouds and images involves computing point-pixel mappings that takes occlusions into account. This generally necessitates accurate depth maps from specialized sensors [VM98, CHLS17] or a costly meshing step [BGLSA18]. We propose a GPU implementation of the straightforward Z-buffering method [Str82], which efficiently computes the sought mapping directly from point clouds, images, and poses. This algorithm defines for each point p the set $v(p) \subset I$ of images in which it is visible. If $i \in v(p)$, we say that image i and point p are *compatible*, and denote by $\text{pix}(p, i)$ the index of the pixel of i which sees p.

We associate with each compatible point-pixel pair (p, i) a vector $o(p, i)$ describing the conditions under which the point p is seen in i. We use a

set of 8 handcrafted features describing the point-pixel distance, local geometry, viewing angle, distortion, local density, and occlusion rate, *i.e.* the ratio of neighbouring points also seen in the image. While we could learn to describe the viewing conditions based on learned 2D and 3D features in an end-to-end fashion, this incurs a significant increase in memory load and did not improve performance in our experiments. Our proposed features contain enough information to learn point-image compatibility and are easy to compute.

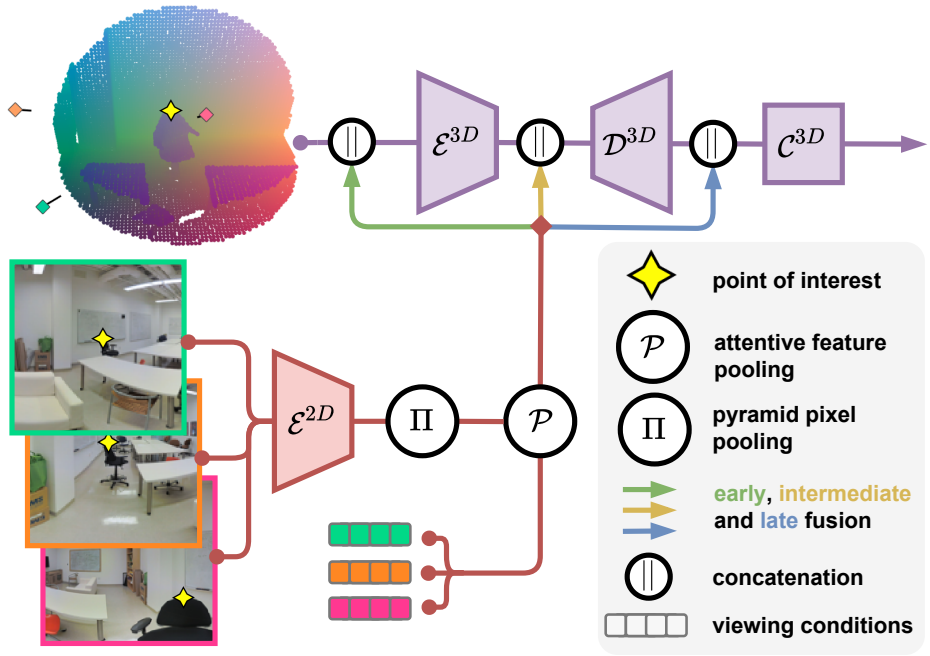


Figure 4.6: **Bimodal 2D/3D Architecture.** We combine a 2D convolutional encoder $\mathcal{E}^{2D}$ and a 3D network composed of an encoder $\mathcal{E}^{3D}$, a decoder $\mathcal{D}^{3D}$, and a classifier $\mathcal{C}^{3D}$ with three different 2D/3D fusion strategies: early (our choice in the experiments), intermediate, and late fusion.

Learning Multi-View Aggregation. We denote by $\{f_i^{2D}\}_{i \in I}$ a set of 2D feature maps of width C associated with the images I, typically obtained with a convolutional neural network. We aim to transfer these features to the 3D points by exploiting the correspondence between points and images. However, not all viewing images contain equally relevant information for a given 3D point. For example, an image viewing a point from a distance may give important contextual cues, while an image taken close and at a straight angle may give detailed textural information. In contrast, an image in which a point is seen from a slanted angle or under high distortion may not contain relevant information and may need to be discarded. Note that

the relevance of an image depends on the other views: while an image may see a point with a less-than-ideal viewing angle, it may be the only one with contextual cues and therefore should be kept.

To model these complex dependencies, we predict for each compatible point-image pair (p, i) a *quality score* $x(p, i) \in \mathbf{R}$ describing the relevance of image i for point p. The scores $x(p, i)$ are defined as a deep set function [ZKR+17] of the viewing conditions $o(p, i)$ for all images i seeing p:

$$\{x(p,i)\}_{i \in v(p)} = \text{DeepSet}(\{o(p,i)\}_{i \in v(p)}) \, . \tag{4.1}$$

We use a softmax function to compute attention scores $a_{(p,i)}$ in $[0, 1]$ representing the relative relevance of image i for point p among the image set $v(p)$. We associate to the point p the 2D features $\mathcal{P}(f^{2D}, p)$ defined as the sum the 2D features for all pixels seeing p weighted by their respective attention scores $a(p, i)$:

$$\mathcal{P}(f^{2D}, p) = \sum_{i \in v(p)} a(p,i) \phi \left(f_i^{2D}[\text{pix}(p,i)] \right) \, , \tag{4.2}$$

with ϕ a learned linear function. The vector $\mathcal{P}(f^{2D}, p)$ corresponds to the image information gathered from all views of p according to their relevance. In Figure 4.7, we observe the influence of several viewing conditions on the attention scores.

We propose to exploit the synergy between 3D point clouds and images by learning a multiview aggregation scheme based on the viewing conditions of points in images: distance, viewing angle, etc. Combined with standard 2D and 3D networks, our methods define a new state-of-the-art while only requiring raw point clouds, images, and poses. In contrast, all other methods operate on colourized point clouds, and 2D/3D fusion approaches require either a meshing step or specialized depth sensors.

Extensions. We propose two extensions of this approach: view gating and feature grouping. View gating is a mechanism that allows us to completely block the image features for a point if no viewing conditions are satisfactory. The gating parameter $g(p) \in [0, 1]$ for a point p is a non-decreasing function of the maximum class score $x_{(p,i)}$ for all viewing images $i \in v(p)$. We then multiply $\mathcal{P}(f^{2D}, p)$ by this parameter, allowing the module to block possibly corrupted information from poor scanning conditions.

The motivation for feature grouping is that there may be several manners for an image to be relevant or not for a point p, such as contextual cues, textural information, and colour. We group the channels of

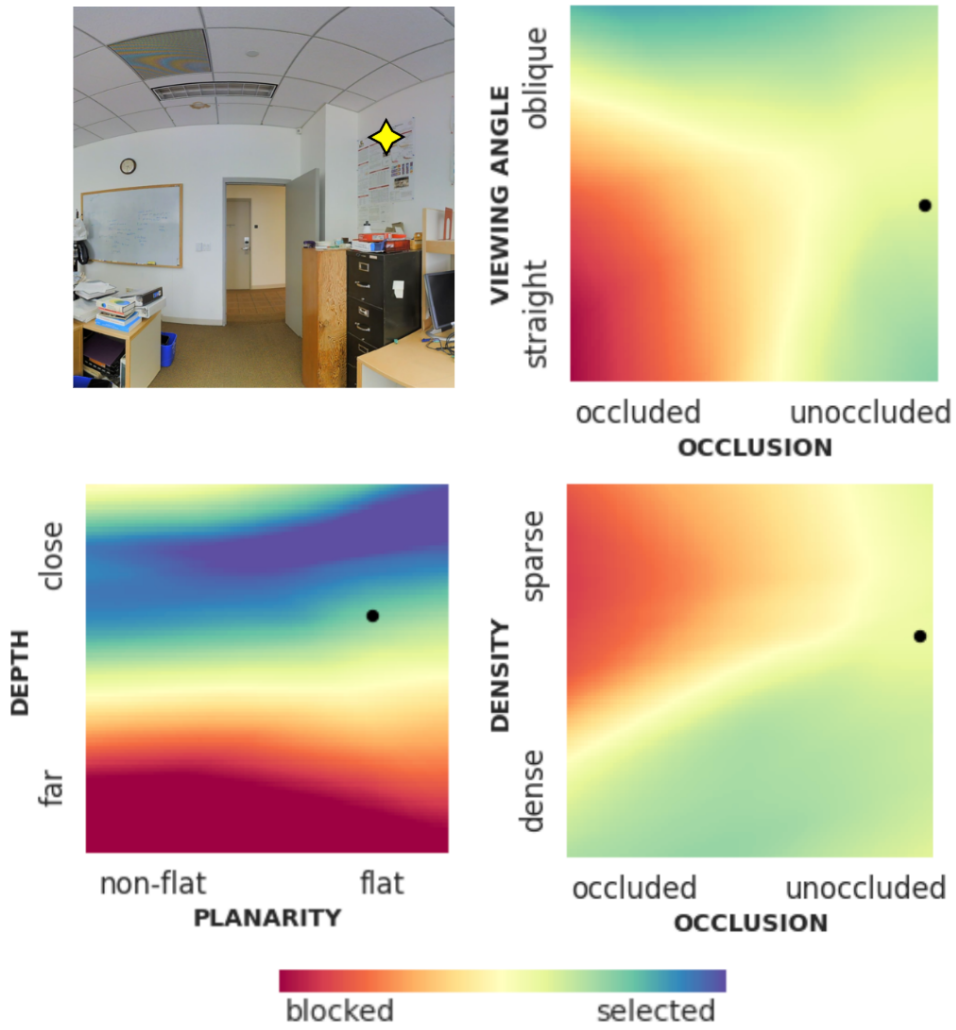


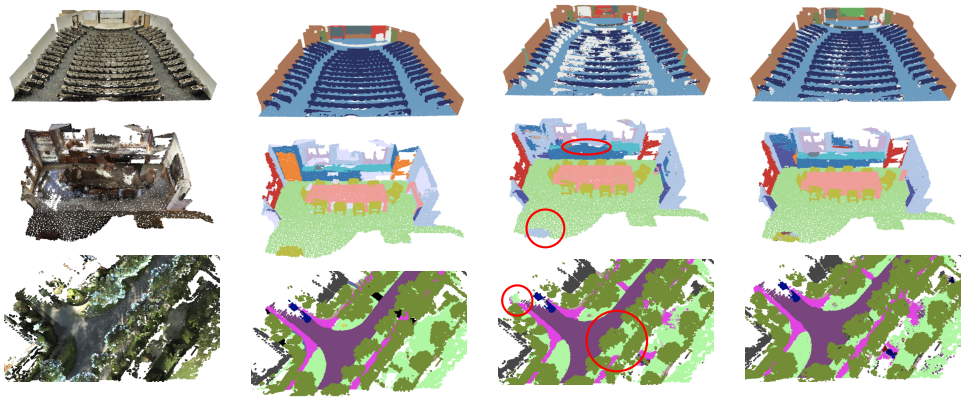
Figure 4.7: **Influence of Viewing Descriptors.** Given a point ✧ seen in an image (top left), we compute the quality scores when varying the viewing conditions from their initial values •. We observe some interesting behaviours for the aggregation module which can combine multiple viewing conditions in a nontrivial way. Top right, images with a high occlusion rate despite a straight viewing angle are blocked from transferring information. Bottom left, the module selects close images, but not too close—except if the surface is planar. Bottom right, images with a high occlusion rate despite a low point density are discarded.

$\phi\left(f_i^{\text{2D}}[\text{pix}(p,i)]\right)$ into K contiguous groups. We now associate for each point-image pair (p,i) a set of K quality values, one for each feature group. We then define K attention scores, each measuring the relative importance of images for its feature group. After their weighted summation, we con-

catenate the features channelwise.

Modality Fusion Network. We represent in Figure 4.6 the different fusion strategy that we have explored. In practice, we observed that replacing the RGB values of point clouds in colourized point clouds with the pooled image features leads to the best performance. We use the implementation of the Minkowski Engine [CGS19] from torch-Points3D [CCHL20] as 3D backbone, and a pretrained ResNet18 network for embedding images.

2.2 Numerical Experiments



(a) Colorized Point Cloud (b) Ground Truth (c) 3D Backbone Predictions (d) Our Model Predictions.

Figure 4.8: **Qualitative illustration.** Scenes from our considered datasets (top: S3DIS, middle: ScanNet, bottom: KITTI-360) with (a) colorized point clouds, (b) ground truth point annotations, (c) prediction of the backbone network operating on the colorized point cloud, and (d) our method operating on raw uncolored point clouds and images. Our approach can use images to resolve cases in which the geometry is ambiguous or unusual, such as a large amphitheatre with tiered rows of seats (top row).

We present in Table 4.2 the performance of our approach compared to state-of-the-art semantic segmentation methods. Using our image fusion module consistently improves the performance of our 3D backbone by a large margin of 3 to 5 points. Remarkably, we outperform the recent Point-Transformer network even though its performance is much higher than our backbone. We also perform better than all 2D/3D fusion methods. Note that all evaluated methods use colourized point clouds except ours, which only uses raw point clouds and images. See Figure 4.8 for qualitative illustrations. The gating mechanism accounts for 0.4 to 3 mIoU points depending on the considered dataset, and channel grouping for 0.4 to 4.8 points.

Table 4.2: **Quantitative Evaluation.** Mean Intersection-over-Union of different state-of-the-art methods on S3DIS's Fold 5 and 6-fold, and KITTI-360 Test. All methods except the last line are trained on colourized point clouds. **State-of-the-art**, second highest.

Model	S3DIS		KITTI	
	Fold 5	6-Fold	Val	360 Test
Methods operating on colorized point clouds				
PointNet++ [QYSG17]	-	56.7	67.6	35.7
SPG+SSP [LS18, LB19]	61.7	68.4	-	-
MinkowskiNet [CGS19]	65.4	65.9	72.4	-
KPConv [TQD+19]	67.1	70.6	69.3	-
RandLANet [HYX+20b]	-	70.0	-	-
PointTrans.[EBD21]	**70.4**	73.5	-	-
Our 3D Backbone	64.7	69.5	69.0	53.9
Methods operating on point clouds and images				
MVPNet [JGS19]	62.4	-	68.3	-
VMVF [KYF+20]	65.4	-	**76.4**	-
BPNet [HZJ+21]	-	-	69.7[1]	-
3D Backbone+ DeepViewAgg (ours)	67.2	**74.7**	71.0	**58.3**

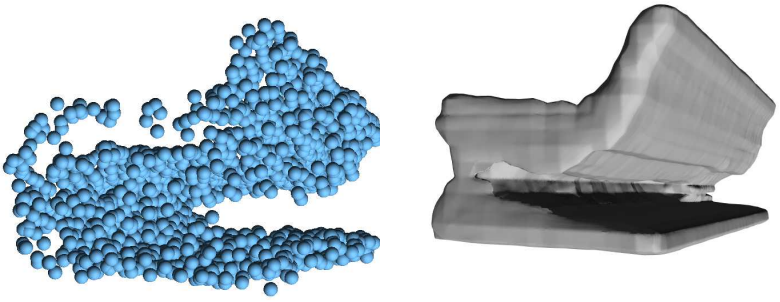
3 Surface Reconstruction with Visibility Information

Most deep surface reconstruction methods from point clouds ignore pose information and only operate on point location. However, sensor visibility holds valuable information regarding space occupancy and surface orientation. This section presents two simple ways to augment point clouds with visibility information that surface reconstruction networks can leverage with minimal adaptation. Our proposed modifications consistently improve the accuracy of generated surfaces as well as the generalization capability of the networks to unseen domains.

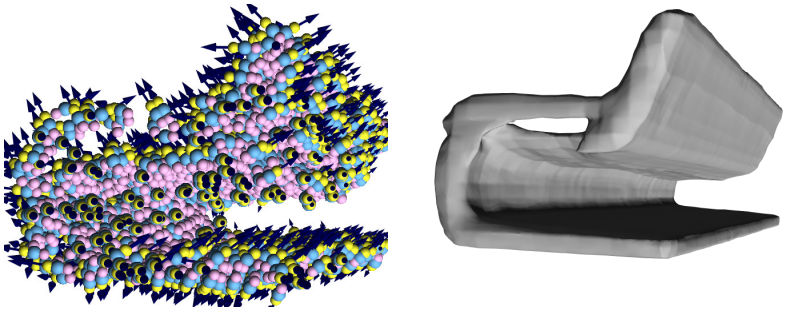
3.1 Visibility-Augmented Point Clouds

By incorporating visibility information into common Deep Surface Reconstruction (DSR) architectures, we aim to improve the occupancy [LPK09, VLPK12, JP11] and orientation [SSG17a] of the predicted surface, see Figure 4.9. While DSR methods have achieved impressive results due to their ability to learn shape priors [PFS+19, MON+19] and shape similarities [HMGCO20, DDN20], they tend to generalize poorly to unseen categories or settings. We aim to increase their generalization capacities by learning a more grounded and *physical* visibility model.

Real-world point cloud acquisition techniques, such as LiDAR scanning or multi-view stereo (MVS), naturally provide the sensor position. We can



(a) Reconstruction using only the point positions.



(b) Reconstruction with visibility augmented point cloud.

Figure 4.9: **Reconstruction with Visibility Information.** We augment each 3D point ● with a sightline vector → and two auxiliary points are placed before ● and after ● the observed point. We can then easily adapt surface reconstruction algorithm to reconstruct significantly more accurate surfaces.

then form lines of sight between the sensor and the acquired points. For each point, we associate (i) a unit *line-of-sight vector* pointing in the direction of the sensor; (ii) two auxiliary points situated before and after the point along the line of sight, see Figure 4.10 for an illustration. By construction, the *before* is likely outside the scanned object, and the *after point* is likely inside. We position these auxiliary points at a distance d on each side of the real point, where d is a typical level-of-detail value, *e.g.* the scanning resolution at a set distance.

We can adapt most DSR networks to handle visibility-augmented point clouds with only two simple modifications:

- We concatenate the 3 coordinates of the line-of-sight vectors to the point location and features.
- We add auxiliary points to the point cloud, thus tripling the num-

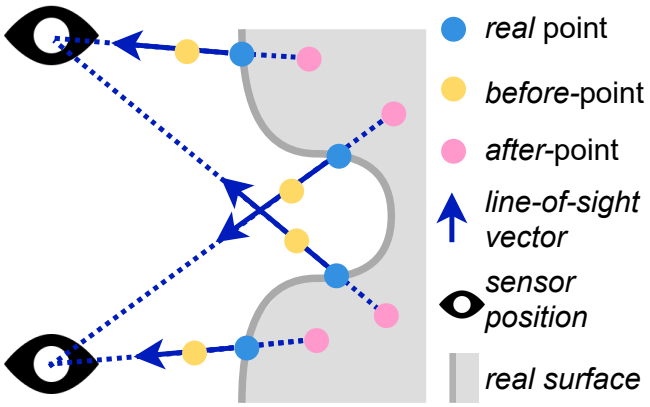


Figure 4.10: **Visibility-Augmented Point Cloud.** Each observed point is associated with a sightline unit vector pointing toward its sensor. New points are added before and after each point to help disambiguate occupancy.

ber of input points. We can perform a volumetric sampling for each category of points (before, after, and real points) for efficiency.

Since we only add point and input features, most existing architectures can be used *as is*. As most of the memory and computation is taken up by the computation of wide feature maps for subsampled clouds in the inner layers, our modification typically does not result in significant overheads.

3.2 Numerical Experiments

We report the results of experiments on object- and scene-level point clouds by evaluating 5 of the top-performing DSR approaches with and without our proposed improvements. Because sensor position is typically not given in current datasets and benchmarks, we perform a synthetic scan of all 3991 shapes from ModelNet10 to form a synthetic training set.

Object-Level Reconstruction We report in Table 4.3 the performance on ModelNet10 for various models, with and without sightline vectors or auxiliary points. We observe a general performance improvement across all methods, as illustrated in Figure 4.11. In practice, adding line-of-sight vectors and auxiliary points is comparable to directly adding the *true oriented normals* from the sought surface in terms of performance. However, these normals are not accessible without ground truth surface, while sensor positions are freely given.

The effect on the run time of adding sightline vectors and auxiliary points depends on the method: under 2% increase for POCO and Con-

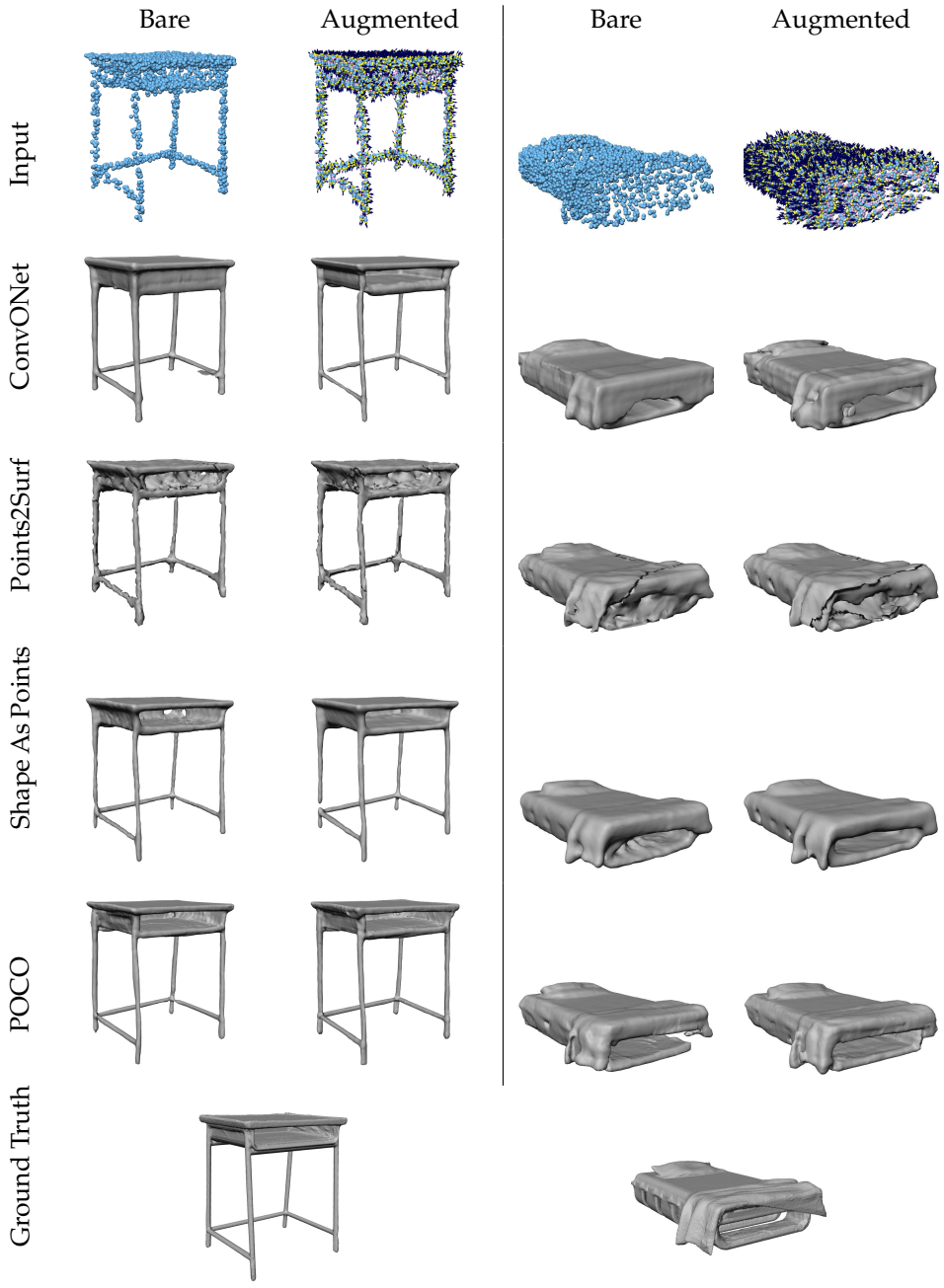


Figure 4.11: **Object-Level Reconstruction.** Shapes from the ModelNet10 test set reconstructed using four different DSR methods operating on point clouds with and without our proposed visibility-augmentation.

vONet, 25% for Shape As Points, and 125% for Points2Surf whose bottleneck is the nearest neighbour computation between points.

Table 4.3: **Object-Level Reconstruction.** We report the volumetric IoU (IoU), average Chamfer distance (CD) and normal consistency (NC) for DSR methods trained and tested on ModelNet10, with and without line-of-sight vectors (SV) or auxiliary points (AP). † Trained on ShapeNet.

Model	SV	AP	IoU ↑	CD ↓	NC ↑
ConvONet-2D [PNM+20]			0.853	0.618	0.934
ConvONet-2D [PNM+20]	✓		0.871	0.557	0.936
ConvONet-2D [PNM+20]	✓	✓	**0.889**	**0.508**	**0.944**
ConvONet-3D [PNM+20]			0.885	0.493	0.949
ConvONet-3D [PNM+20]	✓		0.911	0.424	0.956
ConvONet-3D [PNM+20]	✓	✓	**0.923**	**0.393**	**0.959**
Points2Surf [EOMW20]			0.842	0.590	0.890
Points2Surf [EOMW20]	✓		**0.859**	**0.544**	0.896
Points2Surf [EOMW20]	✓	✓	0.856	0.548	**0.897**
Shape As Points [PJL+21]			0.903	0.438	0.948
Shape As Points [PJL+21]	✓		0.907	0.430	0.950
Shape As Points [PJL+21]	✓	✓	**0.914**	**0.410**	**0.954**
POCO [BM22]			0.907	0.422	0.945
POCO [BM22]	✓		0.915	0.408	**0.950**
POCO [BM22]	✓	✓	**0.917**	**0.406**	**0.950**
† LIG [JSM+20]			–	0.974	0.849
† LIG [JSM+20]	✓		–	**0.880**	**0.882**
DGNN [SLMV21]	✓		0.866	0.543	0.884

Out-of-Domain Reconstruction We evaluate the impact of adding visibility information to existing networks in terms of generalization capacity. We trained POCO on ModelNet and tested it on large-scale scenes from ScanNet (real) and SceneNet (synthetic). In both cases, the visibility-augmented models produce smoother reconstructions with better completeness, see Figure 4.12. Quantitatively, the surface IoU increased by almost 3%.

We also evaluate out-of-category reconstructions by testing models trained on ModelNet on shapes from ShapeNet [CFG+15]. We observe a significant performance increase for all evaluated methods, up to +44% surface IoU points. This illustrates the benefit of learning a visibility model instead of shapes. Lastly, we evaluate models trained on synthetics shapes from ModelNet10 on real-world scans acquired with LiDAR or MVS. In Figure 4.13, we show that all considered networks reconstruct more accurate surfaces using visibility information.

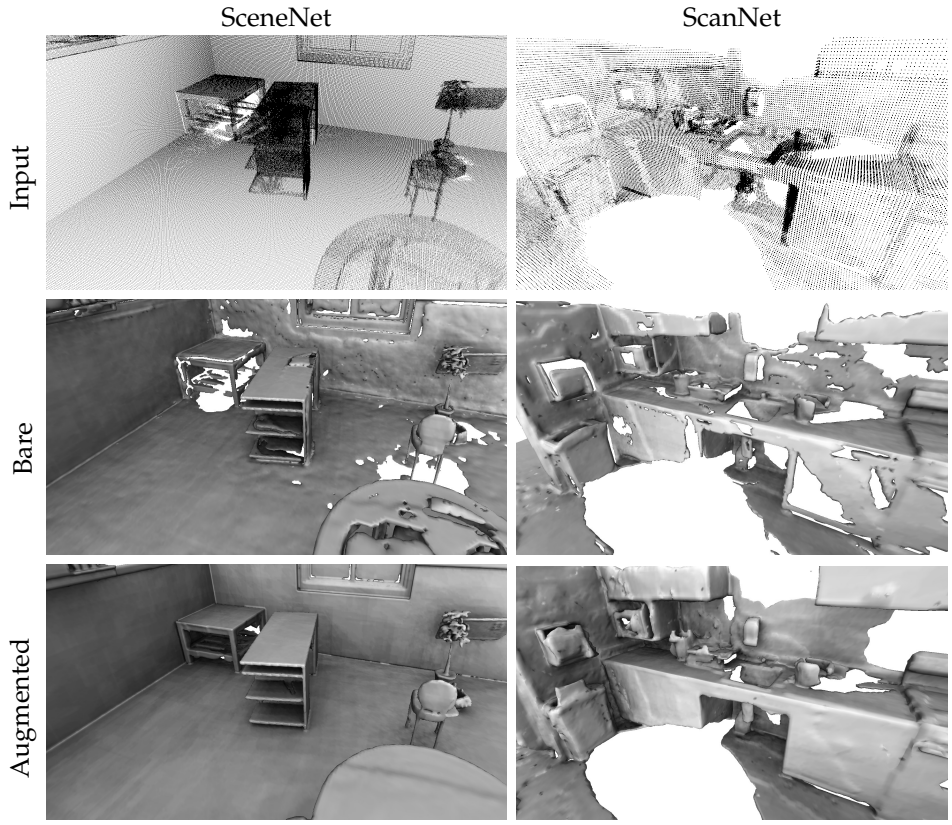


Figure 4.12: **Out-of-Domain Scene-Level Reconstruction.** POCO trained on ModelNet10, with and without visibility information, is run on scenes from SceneNet (synthetic RGB-D scan) and ScanNet (real RGB-D scan).

We propose to incorporate visibility information into deep surface reconstruction methods by adding for each point a unit vector pointing towards the sensor and a pair of auxiliary points along this vector. This only requires minimal adaptation of the considered architecture and results in consistently improved predicted surfaces. More strikingly, the generalization capacity of the networks is significantly increased, and models trained on small synthetic objects can be applied to unseen classes, entire scenes, and even real scans.

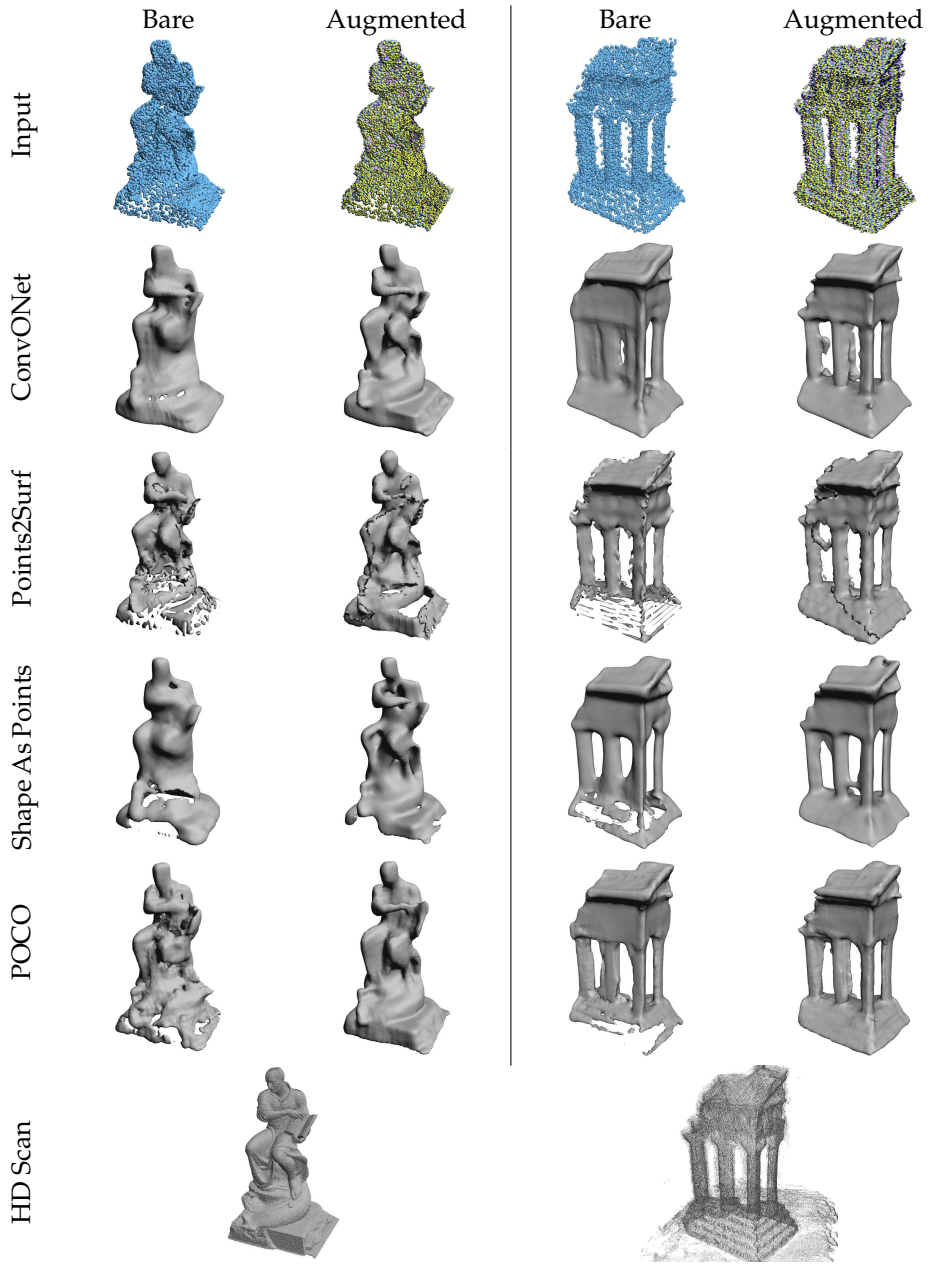


Figure 4.13: **Out-of-Domain Reconstruction.** Reconstructed shapes from a LiDAR point cloud (left, *Ignatius* from Tanks And Temples) and a MVS point cloud (right, *TempleRing* from Middlebury) using four different DSR methods trained on ModelNet10, with and without visibility information.

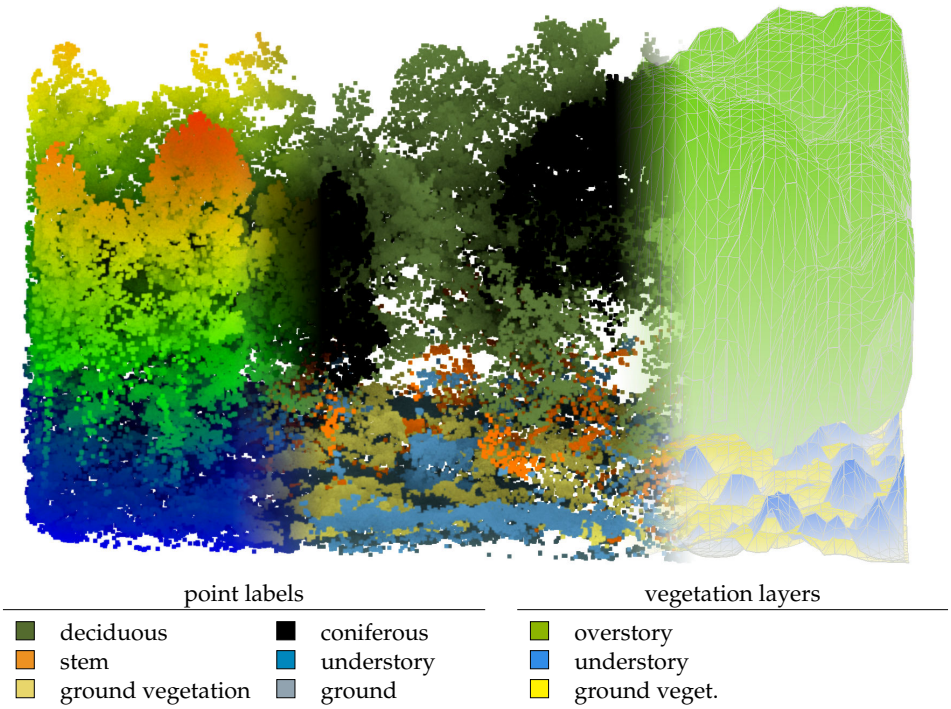


Figure 4.14: **Multi-Layer Forest Analysis.** We introduce WildForest3D, a novel dataset of annotated UAV-LS point clouds of dense forest (left), and a new model for the multi-layer analysis of vegetation. Our network performs 3D semantic segmentation (middle), and produces height maps and watertight meshes for three vegetation layers (right).

4 Forestry Analysis from Aerial LiDAR

The automated analysis of vegetation is one of the key challenges of environment monitoring, allowing us to measure carbon stock and biodiversity [LKA+22, FSM+16]. Since cameras and low-resolution LiDAR cannot capture meaningful information underneath the tree tops [TVB+10], most methods focus on analyzing the canopy [HCZ17]. However, such analysis is limited: (i) tree height makes a poor substitute for biomass [Lu06], (ii) canopy-only analysis ignores the critical biodiversity and natural habitats from the understory [GAMP12], and (iii) ground vegetation is pivotal for forest fire modelling [MR11]. On the other hand, aerial LiDAR scanners with sufficient resolutions can perfectly capture precise geometric information below tree tops [FMJ+15]. We propose to directly model the multi-layer structure of natural forests, whose observation is unlocked by LIDAR.

4.1 Multi-Layer Vegetation Modeling

We release the first dataset with pointwise information for natural forests and introduce a network to explicitly model the multi-layer structure of natural forests.

WildForest3D. We introduce a first-of-its-kind dataset of 29 scans of dense natural forest, which contains 7 million 3D points and 2.1 million individual labels. This corresponds to over 2000 tree instances which were individually located and classified by *in situ* observations from forestry experts. The labels indicate the nature of the vegetation: deciduous canopy, coniferous canopy, understory, stems, and ground. We also produce high-resolution maps of the occupancy and thickness of three vegetation layers: ground vegetation, understory, and overstory.

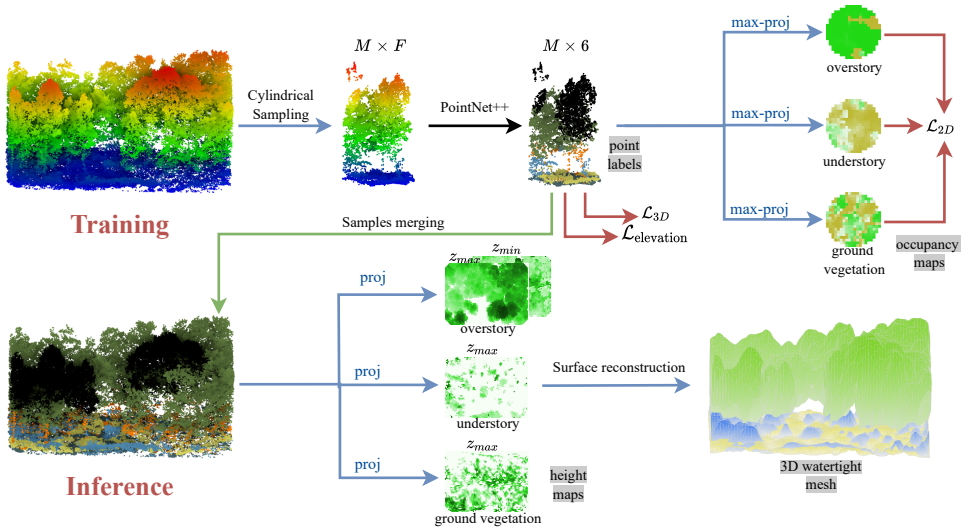
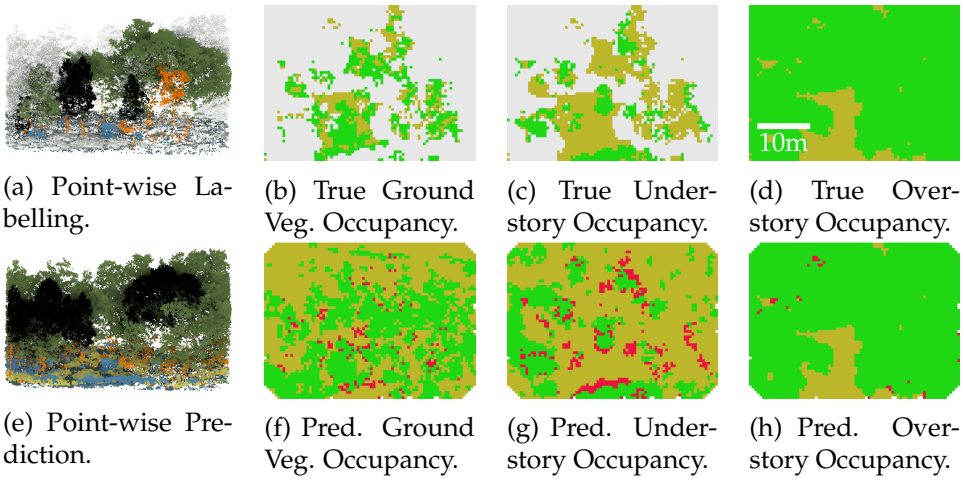


Figure 4.15: **Multi-Layer Modeling.** Our network yields label predictions for each 3D point, which are projected onto rasters to obtain soft occupancy maps for 3 different vegetation layers. The network is supervised using 2D and 3D annotations. During inference, the predictions are used to derive the minimum and maximum elevation of each layer, which we convert in a watertight 3D mesh.

Layer-Wise Modeling. We design a model for the automated analysis of multi-layer vegetation from aerial laser scans. Our network operates directly on the 3D points to perform semantic segmentation of the point clouds and generate layer occupancy rasters, see Figure 4.15. Both 2D and 3D tasks are supervised end-to-end and simultaneously. Once trained, our



(a) Point-wise Labelling. (b) True Ground Veg. Occupancy. (c) True Understory Occupancy. (d) True Overstory Occupancy.

(e) Point-wise Prediction. (f) Pred. Ground Veg. Occupancy. (g) Pred. Understory Occupancy. (h) Pred. Overstory Occupancy.

Figure 4.16: **Qualitative Results.** In the top row, we show the ground truth for the point labels and layer occupancy maps, with un-annotated 3D points in grey. In the bottom row, we report our predictions with errors in red.

model produces height maps for all layers, which we transform into watertight meshes. This surface-based representation is helpful for downstream applications such as biomass, carbon stock, and fire fuel estimation [FBJ+12, GCVY21], soil illumination [HvOW03], or vegetation parameter extraction for the forest inventory [AHK+21], see Figure 4.14.

Trained on our WildForest3D, our model can accurately predict the height and thickness of all vegetation layers at a resolution of 0.25m^2. The estimation error is divided tenfold compared to classical approaches such as regression trees and linear regressions operating on handcrafted descriptors, see Figure 4.16 for a qualitative illustration.

4.2 Weakly-Supervised Learning for Forestry Analysis

Pointwise annotation is exceptionally costly and tedious for forestry data. Contrary to most 3D modalities whose scans can be annotated by specialized companies, forest data require experts to take physical measurements and determine tree species *in-situ*. This task is particularly complex for natural forests, which may be poorly accessible, and whose trees are often interlocking. Experts produce tree-wise annotations that need to be converted to a useful format for training machine learning algorithms, such as point labels, which is a laborious and error-prone process. We propose a weakly-supervised scheme from coarse annotations to remediate this critical limitation. The *in-situ* forestry experts are asked to estimate the coverage ratio of three vegetation layers (overstory, understood, ground vegeta-

tion) in a 10m radius around a geo-localized position, see Figure 4.17. This task is much simpler than detailed tree inventory and can be performed in minutes for each plot of over 300m^2. We release a novel dataset of 200 cylindrical plots with dense 3D scans and cover ratios to evaluate if such annotations are sufficient for training deep models.

(a) LiDAR scan. (b) Aerial image. (c) *In situ* view.

Figure 4.17: **Plot-Based Annotations.** We represent in (a) the 3D point clouds corresponding to two distinct plots. In (b), we show an aerial view of the plots, and in (c) the perspective of the *in situ* annotator. The top plots has a layer coverage of 0%, 10%, 50% for overstory, understory, ground vegetation, and the bottom plot 40%, 60%, 50%.

We consider the problem of strata occupancy regression presented in the last section. We train a network that predicts fine-grained occupancy maps, only supervised with plot-aggregated coverage ratio. We combine this approach with two auxiliary losses: (i) pixel-wise negative cross-entropy to discourage low confidence occupancy (*i.e.* 50%) and produce *crisper* maps; (ii) we model the distribution of point elevation as a mixture of two Gamma distributions, which can be fitted with the expectation–conditional–maximization (ECM) algorithm [YCHNP19]. Models trained on purely plot-aggregated data can predict occupancy maps at 0.25m^2 with high precision.

We exploit the unique perspective offered by aerial LiDAR on the multi-layer structure of natural forests. We propose a model that performs jointly 3D semantic segmentation and layer occupancy regression. Our model yields a precise estimation of the density and height of different vegetation strata, which is helpful for a variety of forestry applications. We also propose a weakly-supervised training scheme that greatly simplifies the annotation process while yielding precise occupancy maps.

Conclusion

We have shown through four different 3D models and applications the advantage of considering the specificities of the sensor when designing network architectures. Considering the viewing conditions of images in point clouds allows us to overtake state-of-the-art 2D/3D hybrid segmentation models while reducing costly preprocessing. Exploiting the rotating LiDAR structure motivates a novel architecture with high precision and significantly decreased latency and model size. We propose a simple alteration to recent deep surface reconstruction methods to incorporate visibility information. Our approach leads to across-the-board improvements and a significant increase in generalization capacity. Finally, we exploit the ability of LiDAR to penetrate tree canopy to learn the multi-layer structure of natural forests.

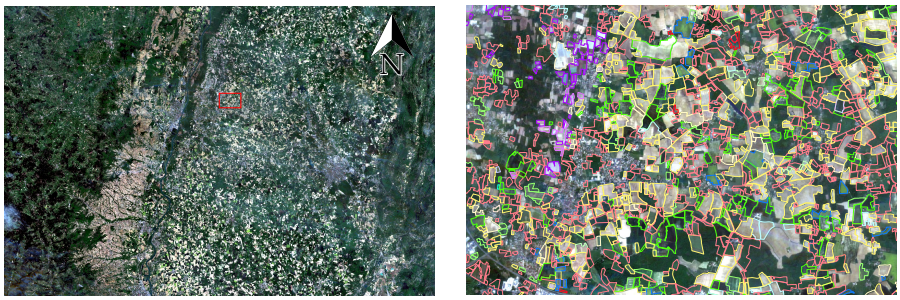
CHAPTER 5

Exploiting the Structure of Satellite Time Series

This chapter presents our work on the automated analysis of Satellite Image Time Series, or SITS. Due to accessible data and plentiful annotations, SITS analysis is a prime example of a remote sensing task which benefits from the deep learning approach. However, SITS follow a complex multi-modal, spatial, temporal, and spectral structure. We propose a family of attention-based algorithms exploiting these properties to increase the speed and precision of crop mapping.
This chapter is organized around the following publications:

[GLGC20]: Vivien Sainte Fare Garnot, Loic Landrieu, Sebastien Giordano, Nesrine Chehata, "Satellite image time series classification with pixel-set encoders and temporal self-attention", *CVPR*, 2020
[GL21b]: Vivien Sainte Fare Garnot, Loic Landrieu, "Panoptic Segmentation of Satellite Image Time Series with Convolutional Temporal Attention Networks", *ICCV*, 2021
[GL20]: Vivien Sainte Fare Garnot, Loic Landrieu, "Lightweight Temporal Self-Attention for Classifying Satellite Image Time Series", *ECML Workshop on Advanced Analysis and Learning on Temporal Data*, 2020
[GL21a]: Vivien Sainte Fare Garnot, Loic Landrieu, "Leveraging Class Hierarchies with Metric-Guided Prototype Learning", *BMVC*, 2021
[GL22]: Vivien Sainte Fare Garnot, Loic Landrieu, "Multi-Modal Temporal Attention Models for Crop Mapping from Satellite Time Series", *ISPRS Journal*, 2021
[QL21]: Félix Quinton, Loic Landrieu, "Crop Rotation Modeling for Deep Learning-Based Parcel Classification from Satellite Time Series", *Remote Sensing*, 2021
[GBLC20]: Sébastien Giordano, Simon Bailly, Loic Landrieu, Nesrine Chehata, "Improved crop classification with rotation knowledge using Sentinel-1 and -2 time series", *Photogrammetric Engineering & Remote Sensing*, 2021
[GLGC19]: Vivien Sainte Fare Garnot, Loic Landrieu, Sebastien Giordano, Nesrine Chehata, "Time-space trade-off in deep learning models for crop classification on satellite multi-spectral image time series", *IGARSS*, 2020

1 Stakes and Challenges of SITS Analysis



(a) RGB image (b) Agricultural Parcels

Figure 5.1: **Sentinel-2 Image.** RGB values for the 31TFM tile, covering 110×110 km and containing over 200 000 agricultural parcels.

Advances in space-based remote sensing, such as the launch of the Planet [BMKS14] and the open-access Sentinel constellations [DDBC+12], have allowed for sustained improvements in the precision and availability of Earth observation. In particular, satellites with high revisit frequency are ideal for analyzing phenomena with complex temporal dynamics [GLGC19] such as crop mapping—the driving application of this chapter, see Figure 5.1.

The automated analysis of Earth observation creates impactful financial and environmental opportunities for private and public actors. Crop monitoring is necessary for the fair allocation of farming subsidies (57 and 22 billion euros per year in Europe and the US, respectively [CAP]). It can help enforce crop rotation best practices for maximizing yields [KKN+15, SL08] and soil protection [AAM+11, Bul92]. Automated SITS analysis is also used for other applications such as surveying urban development [TL16] and deforestation [RHV+18].

Despite the inherent difficulty of differentiating between the complex growth patterns of cultivated plants, virtually limitless access to data and annotations have encouraged the development of algorithmic solutions for automated crop monitoring from satellites [KDB+19]. Indeed, Sentinel-2 averages one multi-spectral observation every five days, which is beneficial for characterizing crop phenology [SBAK20, VMD+18], see Figure 5.2. Moreover, farmers declare the kind of crop cultivated in their parcels yearly. This represents over 10 million yearly annotations for France alone [RPG], all openly accessible in the Land-Parcel Identification System.

SITS present a set of unique challenges and structures:

- **Large Scale.** The amount of data acquired by Sentinel-2 is considerable: more than 25 To / year in Europe for the optical modality alone. Training and applying deep models require computation- and

memory-efficient architectures.

- **Absolute Spatio-Temporal Frame of Reference.** The position of a pixel in a picture or the frame number in a video is typically considered arbitrary and uninformative information. In contrast, absolute spatial and temporal coordinates are crucial information in SITS as they significantly influence observations: wheat crops have different spectral signatures in Denmark or Italy, and in October or March.
- **Cloud Occlusion.** Agricultural parcels generally do not occlude each other from the near-nadir perspective of a satellite. [1] However, cloud cover impacts acquisitions in several ways, from total occlusion to direct and indirect (shadow) radiometric corruption [LSL+17, GYL+20]. [2]
- **Low Spatio-Temporal Resolution.** The spatial resolution and revisit time of the Sentinel-2 satellites are nothing short of a technological marvel and revolutionized Earth observation. However, agricultural parcels are structured by elements such as hedges and furrows, which are typically smaller than the size of Sentinel-2 pixel (10m). Likewise, the evolution of agricultural land is organized around technical acts such as harvest or mowing, which are typically shorter than the satellite's revisit time of five days. This differs from the analysis of natural images or videos, in which the target objects or events are typically covered by multiple pixels and frames.
- **Multi-Modality.** Earth's surface is constantly monitored from space by an array of public, private, and military sensors that differ in nature and resolution. The Sentinel constellation itself comprises synthetic-aperture radar (S1), optical, microwave, and infrared radiometers (S2,S3), spectrometers (S3,S4,S5), and altimeters (S3,S6). In particular, C-band radar and visible/near-infrared radiometers collect complementary information [VTGGP18], which is useful for crop and land mapping. Furthermore, while cloud cover is highly disruptive for optical imagery, radar waves are essentially unaffected [SWNW18].

These characteristics make the analysis of SITS for crop monitoring an exciting and challenging machine learning task, deserving of uniquely tailored solutions. Throughout this chapter, we define SITS as 4-dimensional tensors of size $T \times C \times H \times W$, with T the length of the sequence, C the number of channels per pixel, and H, W the spatial extent of the images. Note that in the general case, the acquisition dates depend on the region

[1] Note that images with oblique viewing conditions also exists, causing self-occlusions in mountainous areas.

[2] Clouds are surprisingly pervasive: up to 65% of the Earth's surface is occluded from space at any given time, and around 35% for land masses [JR08]. This makes crop monitoring [ESR+16] and emergency mapping [RRW19] particularly difficult in humid tropical regions.

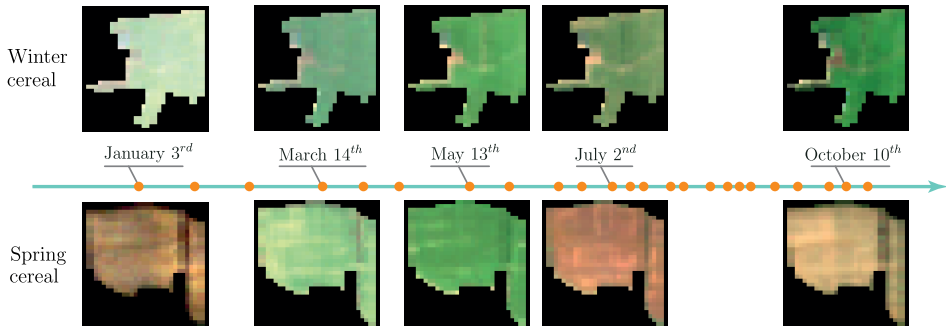


Figure 5.2: **Sentinel-2 Time Series.** Images (RGB bands, 10m per pixel) for parcels of *Winter cereal* and *Spring cereal*. The dots on the horizontal axis represent the unevenly distributed acquisition dates throughout the period of interest. Note the importance of the temporal evolution of the parcels to discriminate between classes.

(tile) considered, a complication that our proposed solutions handle well.

2 Parcel Classification

In this setting, we know the exact extent of each parcel and try to predict the nature of the crop from a time sequence of observations. This is the operational setting in countries with an open-access Land Parcel Identification System (LPIS), such as France or Denmark.

2.1 Spatial Encoding

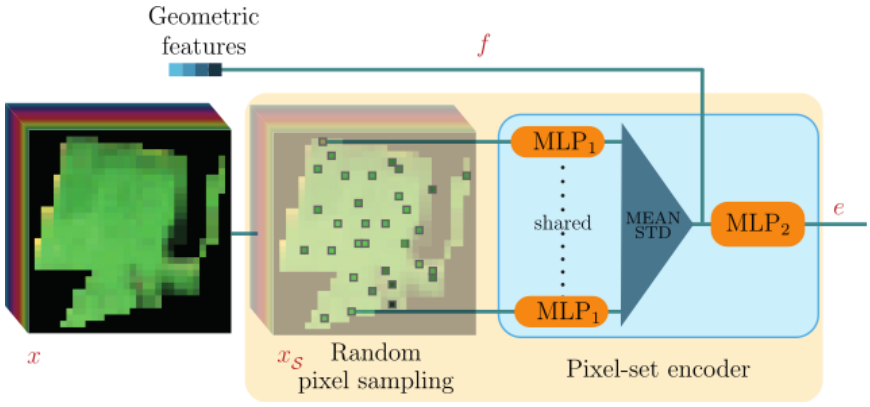


Figure 5.3: **Architecture of the PSE.** Each image of the considered sequence is processed as an unordered set of pixels.

The spatial resolution of 10m per pixel of the Sentinel-2 optical sensor limits the nature of the spatial information that can be captured. In particular, textural information, such as furrows and hedges, is typically lost. This coarseness questions the widespread use of CNNs [KLSS17, RK18, GLGC19] as spatial encoders for SITS. In fact, CNNs are known to rely mainly on texture to extract spatial features [GRM+19].

Instead, we propose to consider each image as an unordered set of pixel, *i.e.* a point cloud in spectral space. Our proposed *Pixel-Set Encoder* (PSE) network randomly samples pixels for each date, and uses a modified set-based architecture [QSMG17a, ZKR+17] to learn statistical descriptors of the parcels' spectral distribution. While this approach destroys spatial information, the randomness of sampling acts as a powerful augmentation strategy to fight overfitting. We also incorporate a set of handcrafted spatial features (*e.g.* perimeter, surface), which are easy to compute and allow the network to retain high-level information about the parcel shape. Another advantage of this approach compared to image-based approaches is that parcels can be processed in parallel, regardless of their size. This allows us to forego the memory-intensive and information-altering resizing and padding steps required to embed images of various shapes with CNNs.

Recognizing that the spatial resolution of Sentinel-2 images may be too low to represent texture, we view parcels as unordered sets of pixels and learn statistical descriptors of the parcels' spectra with a simple set-based network.

2.2 Attention-Based Temporal Encoding

The temporal dimension of remote sensing time series can be handled in a number of ways, such as temporal concatenation [KLG+16], temporal statistics [PVI+16], histograms [BMT+15], time kernels [TMC+17], shapelets [YK09], or probabilistic graphical models [GBLC20]. However, attention-based networks [VSP+17] have initiated a new era for sequential information analysis. Initially designed for Natural-Language Processing (NP), Transformers have proved more expressive and faster to train than RNNs for various tasks. We propose to adapt this concept to satellite image time series. Given the difference between the analysis of NLP and SITS, this requires several adjustments.

Transformer Network. In the original formulation, a *query-key-value* triplet $q^{(t)}, k^{(t)}, v^{(t)}$ is computed simultaneously for each element $x^{(t)}$ of the input sequence. The key $k^{(t)}$ conveys information about the nature of the content, the value $v^{(t)}$ encodes the content itself, and the query $q^{(t)}$ the type

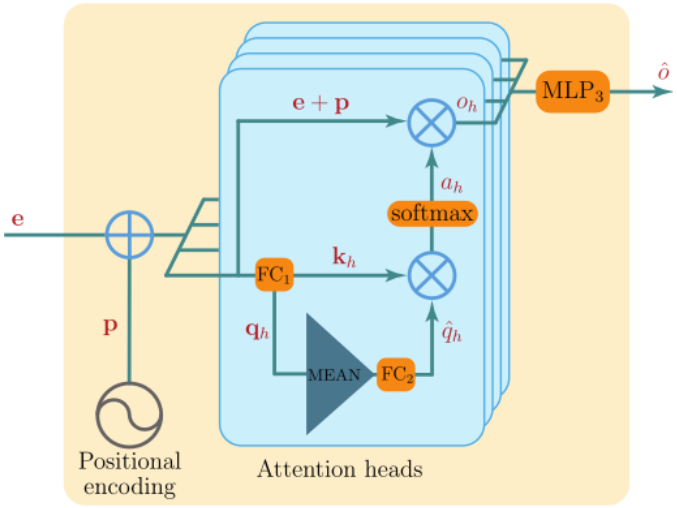


Figure 5.4: **Architecture of the Temporal Attention Encoder.** Our simplified transformer architecture can efficiently embed sequences of image embeddings. Variables in bold are tensors concatenated along the temporal dimension, *e.g.* $\mathbf{e} = [e^{(0)}, \cdots, e^{(T)}]$.

of contextual information needed to embed $x^{(t)}$. Each element of the sequence associates an attention score to the preceding elements by computing the dot product compatibility between $k^{(t<T)}$ and $q^{(t)}$, rescaled with a modified softmax layer.

This procedure can be computed several times in parallel with different sets of independent parameters, or *heads*. This approach, called *multi-head attention*, allows for the specialization of different query-key compatibility. Since all computations are parallel, the Transformer takes full advantage of modern GPU architecture and yields a significant speed increase compared to recurrent architectures.

Positional Encoding. In their paper on text translation, Vaswani *et. al* encode the position of elements in the sequence with discrete Fourier coefficients, as defined in Equation 5.1. Instead of the position in the sequence, we use the number of days since the first observation to help to account for inconsistent temporal sampling (see Figure 5.2).

$$[p^{(t)}]_{i=1}^{E} = \sin\left(\text{day}(t)/\tau^{\frac{2i}{E}} + \frac{\pi}{2}\text{mod}(i,2)\right) , \tag{5.1}$$

with E the size of the spatial embedding.

End-to-End Encoding. Since we train our spatial and temporal encoders simultaneously, we can directly use spatial features as values. This removes

needless computation and avoids a potential information bottleneck. Using a single linear layer $\text{FC}_1^{(h)}$, we generate the keys $k_h^{(t)}$ and queries $k_h^{(t)}$ for each element t of the sequence and each head h in parallel :

$$k_h^{(t)}, q_h^{(t)} = \text{FC}_1^{(h)} \left(e^{(t)} + p^{(t)} \right) \tag{5.2}$$

Master Query Generation. Our goal is to encode the sequence corresponding to one year of observation for each parcel. In NLP, this corresponds to the sentence classification setting, *e.g.* for sentiment analysis, and is typically done using a special "beginning of sentence" token [DCLT18]. Here, we generate a single *master-query* $\hat{q}_h$ per sequence and head h as the temporal average of all queries of the sequence, processed by a single fully-connected layer $\text{FC}_2^{(h)}$:

$$\hat{q}_h = \text{FC}_2^{(h)} \left(\text{mean} \left(\{q_h^{(t)}\}_{t=1}^T \right) \right) \tag{5.3}$$

Two benefits of this approach are that the query can adapt to the content of the sequence and is linear in time instead of quadratic for token embedding tasks.

Multi-Head Temporal Attention. We multiply the master query with the keys to produce attention weights $a^{(h)} \in [0,1]^T$ for each element of the sequence, determining which dates contain the most helpful information. We use the attention weight to obtain the output o_h as the temporal average of the input embeddings. Finally, we concatenate the output of all heads and process the results with MLP_3:

$$a_h = \text{softmax} \left(\frac{1}{\sqrt{d_k}} \left[\hat{q}_h \cdot k_h^{(t)} \right]_{t=1}^T \right) \tag{5.4}$$

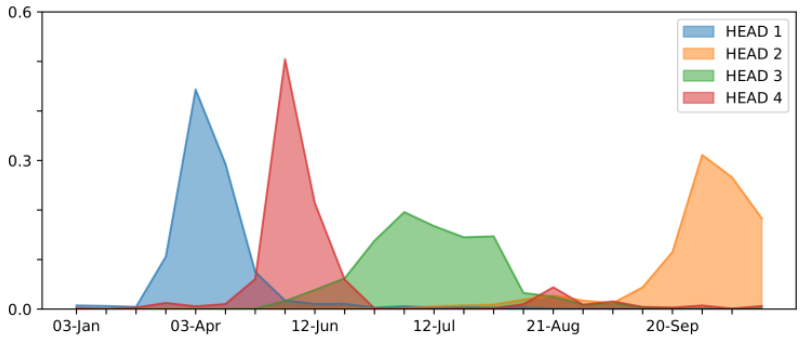
$$o_h = \sum_{t=1}^{T} a_h[t] \left(e^{(t)} + p^{(t)} \right) \tag{5.5}$$

$$\hat{o} = \text{MLP}_3 \left([o_1, \cdots, o_H] \right) . \tag{5.6}$$

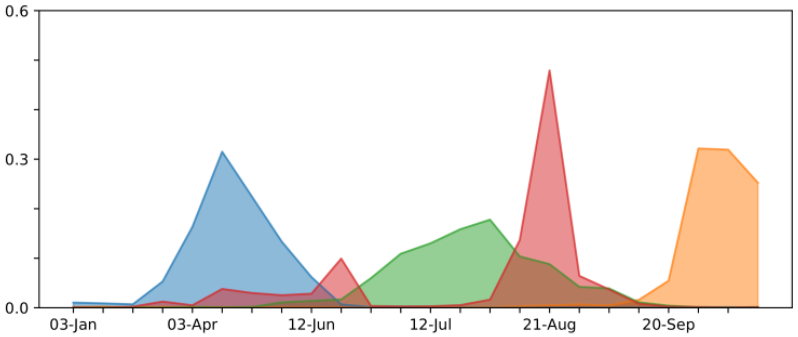
In Figure 5.5, we illustrate head specialization by plotting the average attention masks for two types of cereals. We can see that each of the four heads specializes in a specific portion of the time series. Additionally, the dependency of the attention mask to the input is apparent for head 4, which focuses on late spring for *Spring Cereal* samples and late summer for *Summer Cereal* samples.

2.3 Efficient Temporal Attention

An important observation from the previously presented TAE is that increasing the number of heads is beneficial for precision, but incurs a high



(a) Spring Cereal



(b) Summer Cereal

Figure 5.5: Average attention masks of the TAE heads, obtained from 128 samples of spring (a), and summer (b) cereal parcels. Head 4 adapts to the different classes to detect specific events.

computational and memory cost. We thus propose the Lightweight Temporal Attention Encoder (L-TAE), a stripped-down version of the TAE allowing for the efficient use of many attention heads. We present the main proposed changes below and in Figure 5.6.

Channel Grouping: we split the E channels of the sequence of spatial embeddings into H groups of size $E' = E/H$ with H the number of heads. We denote by $e_h^{(t)}$ the groups of input channels for the h-th head of the t-th element of the input sequence.

Query-as-Parameter: We save computations by directly defining the master queries as trainable parameters of the network. While such queries are not adaptive to the input, we can use many heads to maximize the expressivity of the learned features. Only the keys are obtained with a learned linear layer.

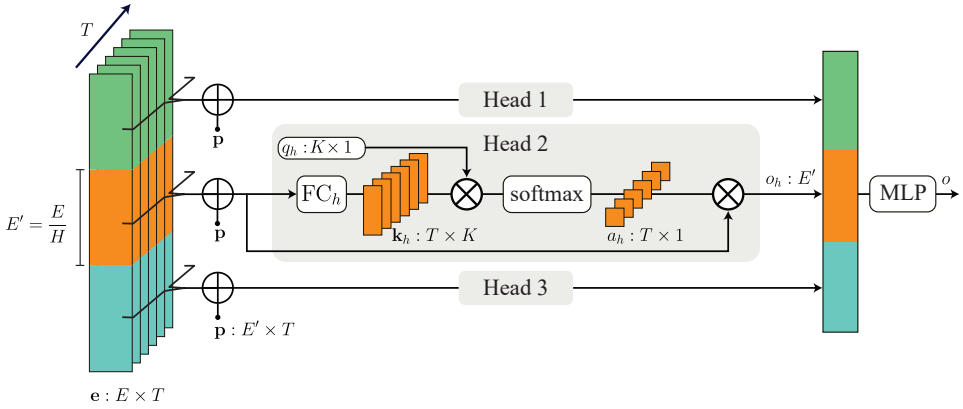


Figure 5.6: **L-TAE Architecture.** An input sequence **e** of T vectors of size E, with $H = 3$ heads and keys of size K. The channels of the input embeddings are distributed among heads. Each head uses a learned query $\hat{q}_h$, while a linear layer FC$_h$ maps inputs to keys. We concatenate the outputs of all heads into a vector the same size as the input embeddings.

Table 5.1: **Asymptotic Complexity of Sequence Embedding Networks.** The GRU's memory update complexity is given in the Keys and Mask columns. X is the size of the output vector. M is the size of the hidden state of the GRU.

Method	Keys	Mask	Output
L-TAE	$O(TEK)$	$O(HTK)$	$O(EX)$
TAE	$O(HTEK)$	$O(HTK)$	$O(HEX)$
Transf.	$O(HTEK)$	$O(HT^2K)$	$O(HEX)$
GRU	$O\left(MT(E+M)\right)$		$O(MX)$

Multi-Head Attention: The rest of the network operates similarly to the TAE: outputs o_h of each head are defined as the temporal average of inputs weighted by attention scores derived from key-query compatibility. Finally, the outputs of all heads of size E/H are concatenated into a vector of size E and processed by a multi-layer perceptron MLP to the desired width.

Computational Complexity In Table 5.1, we report the asymptotic complexity of different sequence embedding algorithms. The L-TAE channel grouping strategy removes the influence of H in the computation of keys and outputs compared to a TAE or a Transformer. The complexity of the L-TAE is also lower than the GRU's as M, the size of the hidden state, is typically larger than K (130 vs 8 in the experiments presented in Table 5.2).

2.4 Quantitative Results

Table 5.2: **Quantitative Evaluation.** We report the overall Accuracy (OA) and mIoU Classification metrics for different architectures, as well as training time (one epoch) and inference time for the entire dataset, and the size of the dataset on the disk . [1] disk space required for training and pure inference, [2] time for the entire training step, [3] preprocessing and inference time, [4] dataset before and after preprocessing.

	OA	mIoU	Training (s/epoch)	Inference (s/dataset)	Disk Size Gb
PSE+TAE (ours)	**94.2** ±0.1	**50.9** ±0.8	158	**149**	**28.6 / 12.3**[1]
CNN+GRU [GLGC19]	93.8 ±0.3	48.1 ±0.6	656	633	98.1
CNN+TCNN [PWP19]	93.3 ±0.2	47.5 ±1.0	635	608	98.1
Transformer [RK19]	92.2 ±0.3	42.8 ±1.1	13	420 + 4[3]	**28.6 / 0.22**[4]
ConvLSTM [RK18]	92.5 ±0.5	42.1 ±1.2	1 283	666	98.1
Random Forest [GBLC20]	91.6 ±1.7	32.5 ±1.4	**293**[2]	420 + 4[3]	**28.6 / 0.44** [4]

We report in Table 5.2 the performance of our approach and different competing networks on a proposed open-access dataset.

Sentinel2-Agri. We propose a dataset comprising 191 703 image time sequences corresponding to agricultural parcels of the T31TFM Tile in southern France. All sequences are composed of 24 observations spanning from January to October 2017, with a spatial resolution of 10m and 10 spectral bands (we discard B01, B09, and B10). We associate each sequence with the label corresponding to the majority culture retrieved from the French Land Parcel Identification System records[3] with a 20 class nomenclature designed by the subsidy allocation authority of France. The dataset is highly imbalanced: four classes cover 90% of the samples.

We propose two versions of the dataset, depending if images are stored as patches or sets. In the *patch format*, we resize each parcel into a tensor of size $T \times C \times 32 \times 32$ with nearest neighbour spatial interpolation and zero-padding. In the *set format*, the pixels of each parcel are stored in arbitrary order into a tensor of size $T \times C \times N$, with N the total number of pixels in a given parcel. Note that this format neither loses nor creates information, regardless of parcel size. Hence, this setup saves up to 70% disk space compared to the patch format (28.6Gb *vs.* 98.1Gb). This dataset constitutes the first large-scale dataset for object-based agricultural parcel classification and is freely accessible at github.com/VSainteuf/pytorch-psetae.

[3] http://professionnels.ign.fr/rpg

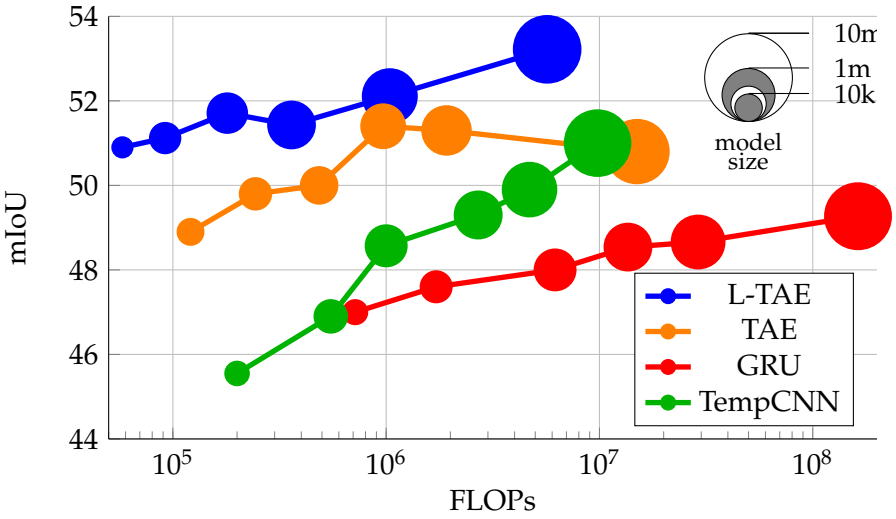


Figure 5.7: **Precision, Speed, and Model Size.** We report the mIoU averaged over 5 runs for different temporal encoders plotted with respect to the number of FLOPs necessary to process one sequence. The marker size represents the number of trainable parameters in the model. The L-TAE outperforms other models across all model sizes and processing requirements.

Competing Methods. We compare our approach to recent algorithms operating on similar datasets. These methods use CNNs or spatial pooling for spatial encoding and recurrent neural networks, temporal convolutions, or transformers for temporal encoding. We parametrize all models to have roughly 150k parameters. We report in Table 5.2 the performance of all methods in terms of precision, speed, and memory usage.

Efficient Temporal Attention. We focus on the advantages of the L-TAE by using the same PSE network for spatial encoding and varying the temporal encoders. We report the performance in Figure 5.7.

We propose to adapt the successful attention model from NLP to SITS. Not only does our approach define a new state-of-the-art in terms of precision, but it is four times faster and uses a parsimonious data format only requiring a quarter of disk size. A variation of our temporal model reaches even higher performance with smaller models: a model with only 9k parameters overtakes a GRU with over 3m parameters. Several other studies confirmed the superiority of our models for SITS [SK20, KTR+21].

3 Panoptic Segmentation of SITS

We now study crop mapping when both the content and contour of parcels must be retrieved, which is the most common operational setting. This problem can be framed as panoptic segmentation of an image sequence [KGHD19, MV21]. However, since agricultural parcels are geo-referenced and static, we only need to consider the geo-referenced pixels and not all images in the sequences. This removes the complex task of spatially tracking objects [TSA19]. While some approaches propose to perform instance segmentation [Rie17], delineation (border detection) [GPGMLS17, MPT20, WD20], or oversegmentation [PKPG12], there exists no dedicated approach for detecting individual objects consistently across an entire satellite image sequence.

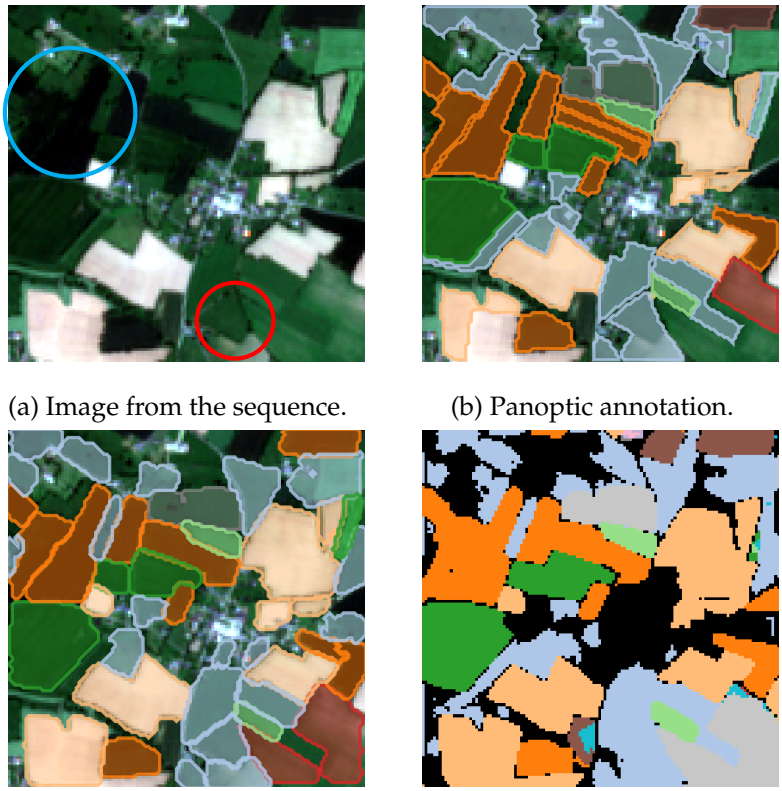
3.1 Spatio-Temporal Pixel Encoding

The first step of our pipeline is to learn pixel-wise features encoding the spatio-temporal dynamics of SITS containing multiple parcels. The key insight of our proposed architecture is to use temporal attention to compute meaningful spatio-temporal maps at several resolution levels *simultaneously*. Temporal U-Net networks [SPI$^+$19, RCW$^+$19, PVK21] only process the temporal dimension of the lowest resolution and collapse the skip connections with spatial averaging. This prevents the extraction of spatially adaptive and parcel-specific temporal patterns at higher resolutions. Conversely, convolution-recurrent encoders [RK18] process the temporal dimension at the highest resolution, which results in an increased memory requirement and ignores the low spatial regularity of the data. Instead, we propose to efficiently compute temporal attention masks at all resolutions using an upsampling strategy on the attention masks themselves, see Figure 5.9.

Our model, dubbed U-TAE (U-Net with Temporal Attention Encoder), encodes a sequence x of dimension $T \times C \times H \times W$ in three steps:

- **(a)** Each image in the sequence is embedded simultaneously and independently by a shared multi-level spatial convolutional encoder.
- **(b)** We apply an L-TAE to all the pixels of the lowest resolution feature map independently to obtain pixelwise temporal attention masks. We then use bilinear interpolation to obtain masks at all spatial resolutions and collapse the spatial dimension of the maps at all levels.
- **(c)** The resulting spatial feature maps are processed by a convolutional decoder to produce rich feature maps at all levels.

A subtlety of our spatial encoder is that the image batches mix acquisitions taken at different dates. Consequently, the samples are not identically distributed, which prevents the use of BatchNorms [IS15]. To address this issue, we use Group Normalization [WH18] with 4 groups instead in the



(a) Image from the sequence. (b) Panoptic annotation.

(c) Panoptic segmentation. (d) Semantic segmentation.

Figure 5.8: **Problem Statement.** We consider an image sequence (a) with panoptic annotations (b). Our objective is to learn spatio-temporal pixel features, which can be used to perform panoptic (c) and semantic segmentation (d). We observe some difficult configurations: boundary ambiguities that can not be resolved from a single image (cyan circle ○), and conversely, visually fragmented areas annotated as a single instance (red circle ○).

encoder. As shown later, the impact of this simple change on the performance is drastic for all *temporal U-Net* and not only ours.

PASTIS Dataset. We introduce the PASTIS (Panoptic Agricultural Satellite TIme Series) dataset, the first large-scale, publicly available SITS dataset with semantic and panoptic annotations. PASTIS is comprised of 2 433 sequences of multi-spectral images of size $10 \times 128 \times 128$. Each sequence contains between 38 and 61 observations taken between September 2018 and November 2019, for over 2 billion pixels spanning 4000 km^2. We estimate that close to 28% of images have at least partial cloud cover.

Each pixel of PASTIS is associated with a semantic label taken from a nomenclature of 18 crop types plus a background class. Each non-

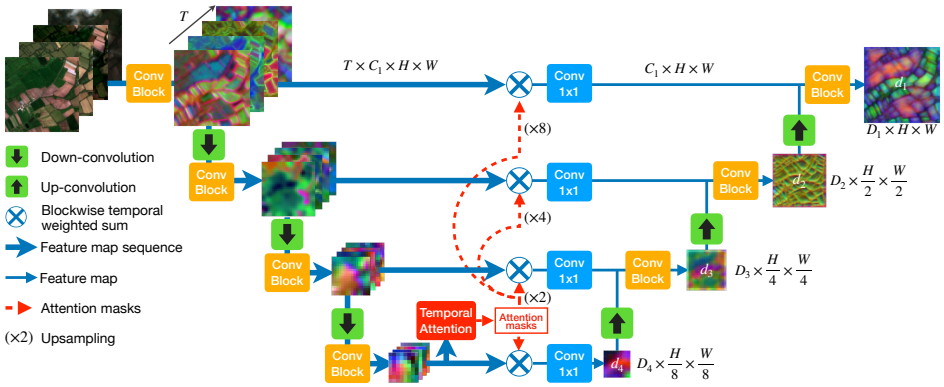


Figure 5.9: **Spatio-temporal Encoding.** A sequence of images is processed in parallel by a shared convolutional encoder. At the lowest resolution, an attention-based temporal encoder produces a set of temporal attention masks for each pixel, which are then spatially interpolated at all resolutions. These masks are then used to collapse the temporal dimension of the feature map sequences into a single map per resolution. Finally, a convolutional decoder computes features at all resolution levels.

background pixel has a unique instance label corresponding to its parcel index. In total, 124 422 parcels are available, each with bounding box, pixel-precise mask, and crop type. The French Payment Agency estimates the accuracy of the crop annotations via in situ control to be over 98% and the relative error in surface to be under 0.3%. To allow for fair cross-validation, we split PASTIS into 5 folds with a 1km buffer between images.

Semantic Segmentation. By setting the width of the highest resolution layer to the number of classes, the U-TAE can perform semantic segmentation. As seen in Table 5.3 the U-TAE significantly outperforms the other methods from the state-of-the-art. Interestingly, most of the advantage of our approach is attributable to the use of attention maps at all resolution levels. Collapsing the skip connections with simple temporal average results in a performance comparable to the other evaluated methods. The main advantage of temporal attention is that it can be easily used at different resolutions, in contrast to recurrent network-based approaches.

3.2 Panoptic Segmentation

We now use the multi-scale feature maps learned with the U-TAE to perform panoptic segmentation. Since the U-TAE operates on many images for each sequence (often over 50), we favour an efficient design. Furthermore, given the relative simplicity of parcel borders, we can avoid com-

Table 5.3: **Semantic Segmentation.** We report for our method and six competing methods the model size in trainable parameters, Overall Accuracy (OA), mean Intersection over Union (mIoU), and Inference Time for one fold of $\sim$ 490 sequences (IT).

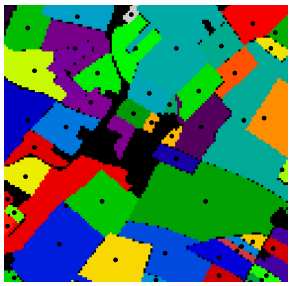
Model	# param $\times 1000$	OA	mIoU	IT (s)
U-TAE (ours)	1 087	**83.2**	**63.1**	**25.7**
3D-Unet [RCW$^+$19]	1 554	81.3	58.4	29.5
U-ConvLSTM [RCW$^+$19]	1 508	82.1	57.8	28.3
FPN-ConvLSTM [MLRF$^+$21]	1 261	81.6	57.1	103.6
U-BiConvLSTM [MLRF$^+$21]	1 434	81.8	55.9	32.7
ConvGRU [BYPC16]	1 040	79.8	54.2	49.0
ConvLSTM [RK18]	1 010	77.9	49.1	49.1
U-TAE w. Skip Mean	1 074	82.0	58.3	24.5

plex region proposal networks such as Mask-RCNN [HGDG17]. Instead, we adapt the single-stage Object-as-Point instance segmentation network [ZWK19, WXS$^+$20] and name our network *Parcels-as-Points* (PaPs) to highlight this inspiration.

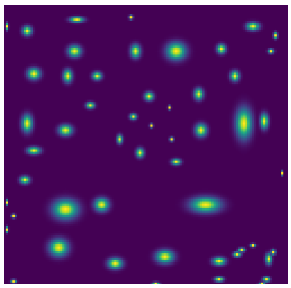
Parcel Detection. We define a *centerness heatmap* from the ground truth parcel bounding boxes by associating each centroid with a heteroscedastic Gaussian kernel depending on their spatial extent. This allows us to frame the detection problem as the regression of this heatmap, supervised using a modified logistic loss. We predict parcel centerpoints at all spatial local maxima of the predicted centerness heatmap, which can can be efficiently retrieved with a single max-pooling operation, see Figure 5.10.

Size, Class, and Shape Prediction. We attribute to each detected centerpoint a multi-scale descriptor by concatenating the feature maps at all resolutions in the channel dimension. From this descriptor, we predict a class, a size, and a small shape patch of fixed size S, typically 16 pixels. This patch is reshaped using the predicted size to obtain a rough shape prediction. To obtain a pixel-precise instance prediction, we also predict a saliency map at full resolution, shared for all detected centerpoints of the sequence. This saliency map is cropped along the predicted parcel bounding box, and combined with the rough shape prediction in a residual fashion to obtain a soft occupancy mask for each predicted centerpoint, see Figure 5.11

We supervise the class predictions with the cross-entropy and the predicted sizes with a rescaled L1 loss. The shape patches and saliency map are supervised with the binary cross entropy between the predicted and

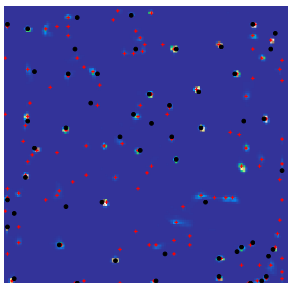


(a) Instance masks



(b) Target heatmap

(c) Observation from sequence.



(d) Predicted centerpoints

Figure 5.10: **Centerpoint Detection.** The ground truth instance masks (a) is used to construct a target heatmap (b). Our parcel detection module maps the observation (c) to a predicted heatmap (d). The predicted centerpoints (red crosses) are the local maxima of the predicted heatmap (d). The black dots are the true parcel centers.

Table 5.4: **Panoptic Segmentation Experiment.** We report class-averaged panoptic metrics: SQ, RQ, PQ. The U-TAE appears essential to the quality of the prediction.

	SQ	RQ	PQ
U-TAE + PaPs	81.3	49.2	40.4
U-ConvLSTM + Paps	80.9	40.8	33.4

true occupancy mask of the parcels. We combine all losses without scaling.

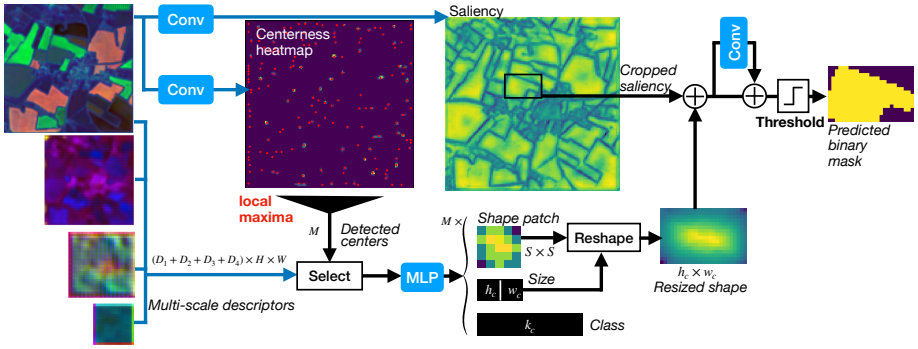
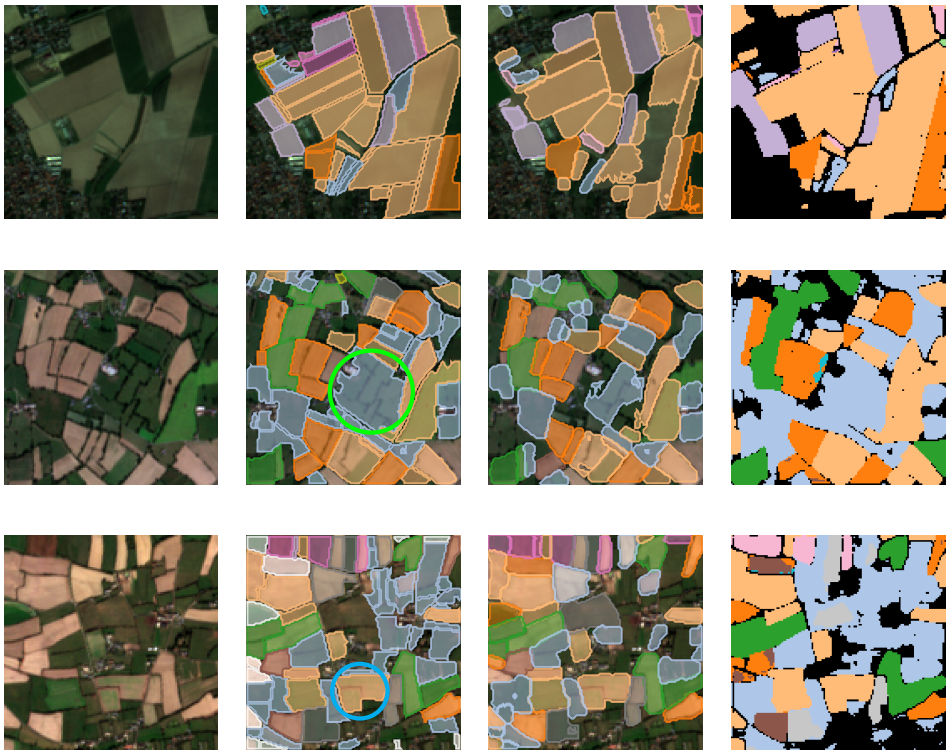


Figure 5.11: **Panoptic Segmentation.** The local maxima of the predicted centerness heatmap define M tentative parcels. We then predict a bounding box size, a semantic class, and an $S \times S$ shape patch for each detected centerpoint. Combined with a global saliency map, the shape patch allows us to predict pixel-precise masks.

Converting to Panoptic Segmentation. Panoptic segmentation consists in associating to each pixel a semantic label and, for non-background pixels (our only *stuff* class), an instance label [KGHD19]. We resolve the overlaps of the previously described instance proposals with Non-Maximum Suppression. We report in Table 5.4 the performance of our approach, constituting the first state-of-the-art of SITS panoptic segmentation. See Figure 5.12 for qualitative illustrations of the advantage of our approach.

We adapt our temporal attention encoder to analyze remote sensing image time series. We propose a hybrid convolutional transformer architecture which defines a new state-of-the-art for semantic segmentation of agricultural parcels. We also propose the first dataset for evaluating the task of agricultural parcel panoptic segmentation, for which our approach sets up the first state-of-the-art.



(a) Single image (b) Panoptic GT (c) Panoptic seg (d) Semantic seg

Figure 5.12: **Qualitative Panoptic Segmentation Results.** We present a single image from the sequence using the RGB channels (a), and whose ground truth parcel boudaries and types are known (b). We then present our predicted panoptic segmentation (c), and semantic segmentation (d). Large, fragmented parcel are sometime declared as one single field ○, making their recovery difficult. Conversely, fragmented parcels ○ are correctly predicted as a single instance, suggesting that our network is able to use the temporal dynamics to recover ambiguous borders.

4 Multi-Modal TAEs

The various sensors of Earth Observation satellites capture information of different natures and distinct spatial and temporal resolutions, with varying resilience to atmospheric conditions. In particular, C-band radar and optical images possess well-known synergies for automated crop mapping [VTGGP18, SWNW18, CTGHM+19]. Optical time series contain highly relevant information for monitoring the evolution of plant phenology [VMD+18, SBAK20], but are highly susceptible to cloud cover and atmospheric distortion [STAL20]. Conversely, radar acquisitions are impervious to cloud cover, which makes them uniquely well-suited for monitoring rapidly changing biological processes [MKL+14]. However, radar is influenced by extrinsic factors such as humidity and terrain, making it harder to extract robust features.

The fusion of optical and radar time series has been extensively explored with traditional machine learning methods [VTGGP18, SWNW18, GBLC20], and more recently, recurrent neural networks [IIGM19]. As illustrated in Figure 5.13, we leverage this multimodality with temporal attention networks. We extend the work of [OAPL21] by assessing the benefit of combining optical for improving the precision and robustness to cloud cover of several crop mapping tasks. We also evaluate the effect of two simple training enhancements.

Fusion Schemes. As represented in Figure 5.14, we list four different schemes for merging image time series of different modalities:

- Early Fusion. We choose a pivot modality (*e.g.* optical) and temporally interpolate all other modalities to the acquisition dates of this pivot modality. We stack the resulting images in the channel dimension and process the sequence with a single spatio-temporal encoder.
- Mid Fusion. Each modality is processed by a dedicated spatial encoder. We merge the resulting embeddings into a single time series processed by a single temporal encoder.
- Late Fusion. Each modality is processed by a dedicated spatio-temporal encode, and the resulting feature maps are concatenated channel-wise before classification.
- Decision Fusion. Each modality is classified independently, and we average their class scores.

Training Enhancements. Optical images contain richer information for crop mapping than radar acquisitions. This imbalance can cause the decision to overly rely on the optical modality, resulting in a weak supervisory signal for the other modality. We present two simple training enhancements to mitigate this issue:

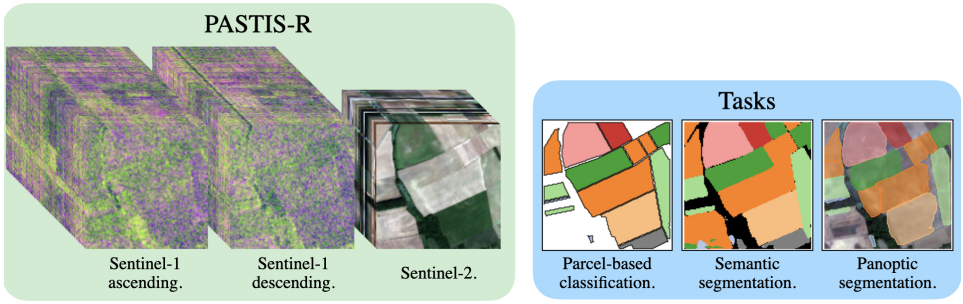
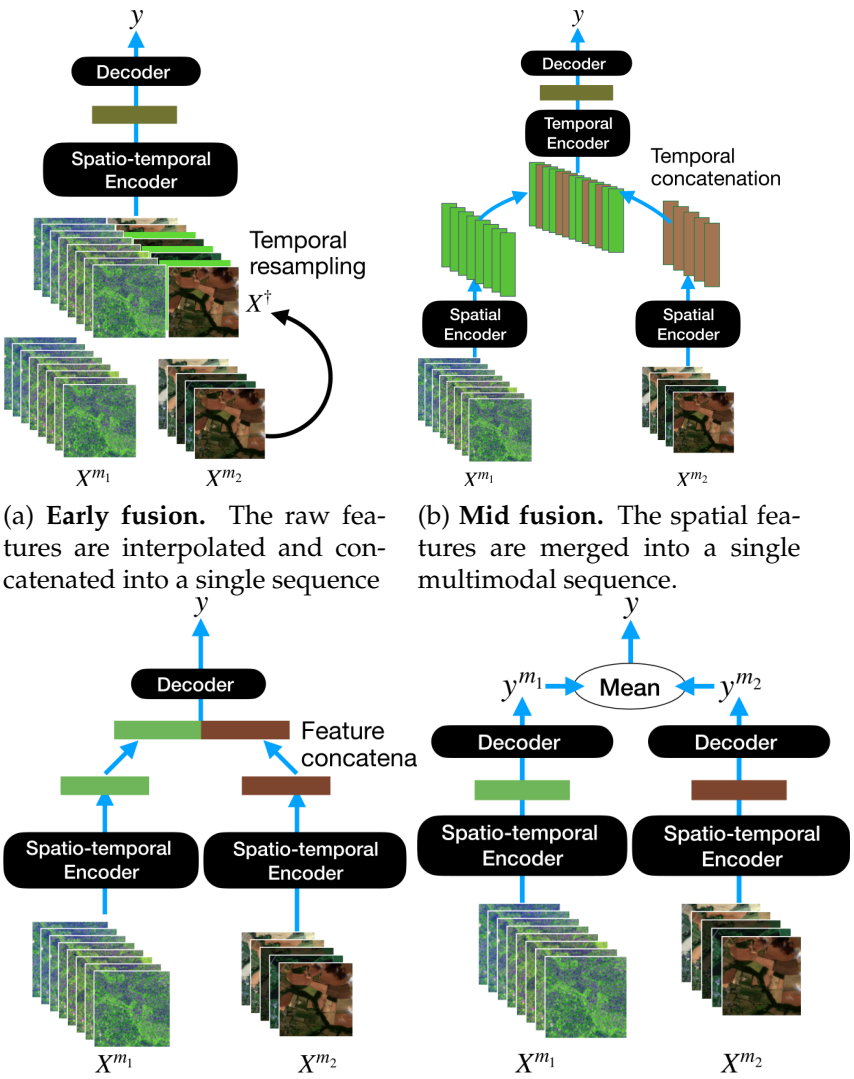


Figure 5.13: **MultiModal Dataset.** We introduce the PASTIS-R dataset containing 2433 multimodal image time series of Sentinel-2 optical and Sentinel-1 radar data. We evaluate different fusion strategies and enhancements on parcel-based classification, semantic segmentation, and panoptic segmentation.

- Auxiliary Supervision. All schemes except early fusion can benefit from the addition of auxiliary loss supervising each modality independently. This ensures that all modalities contain enough information to perform meaningful classification on their own.
- Temporal Dropout. During training, we randomly drop some acquisitions from the time series, with an increased probability for the optical modality. This prevents the network from overly relying on a single modality. Note that this enhancement also reduces the memory required for training, which can prove necessary for large models or image sequences.

PASTIS-Radar. To evaluate the benefit of multimodality, we extend the PASTIS dataset with corresponding Sentinel-1 observations. We separate the observations made in ascending and descending orbits into two different time series of around 70 3-channel images: vertical polarization (VV), horizontal polarisation (VH), and the ratio of vertical over horizontal polarization (VV/VH). We use the Ground Range Detected format processed into backscattering coefficient in decibels, orthorectified at a 10m spatial resolution. We do not apply spatial or temporal speckle filtering or radiometric terrain correction. The resulting data consists of 339k radar images and is available at github.com/VSainteuf/pastis-benchmark.

Multi-Task Evaluation. We report in Table 5.5 the improvements brought by the proposed fusion strategies and enhancements. We observe a consistent increase in precision compared to the individual modalities across all schemes, with late fusion being particularly well suited for parcel classification and semantic segmentation, while early fusion gives the best panoptic results. See Figure 5.16a for qualitative illustrations of our results.



(a) **Early fusion.** The raw features are interpolated and concatenated into a single sequence

(b) **Mid fusion.** The spatial features are merged into a single multimodal sequence.

(c) **Late fusion.** The learned spatio-temporal features of each modality are concatenated prior to classification

(d) **Decision fusion.** each modality is processed independently, and the resulting decision averaged

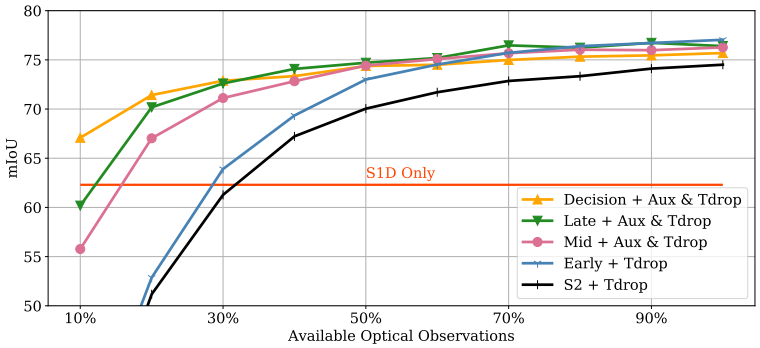
Figure 5.14: **Evaluated Fusion Schemes.** We implemented four ways to embed time series of different modalities.

Table 5.5: **Quantitative Evaluation**. We report the 5-fold cross-validated performance of the individual modalities and fusion schemes across three tasks. When it is possible to isolate their effect, we report the improvement of our proposed enhancements in parenthesis. ‡ indicates that the enhancements are detrimental, and ⋆ that the enhancements are necessary for fitting the model into memory. A dash − means that this model was not evaluated for memory issues or because its design is incompatible with the task.

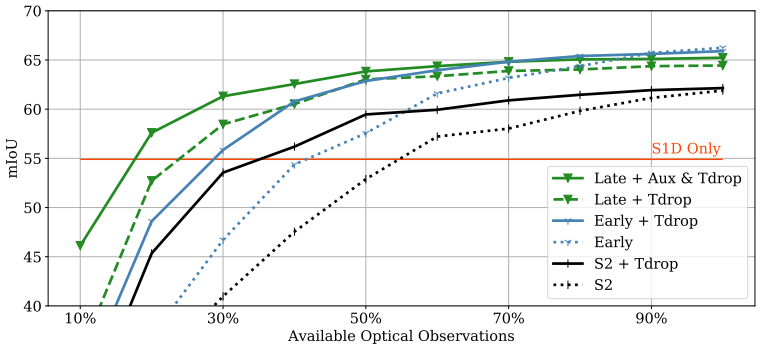
Modality	Parcel Classification	Semantic Segmentation	Panoptic Segmentation		
	mIoU	mIoU	SQ	RQ	PQ
S1D	64.7 (+0.2)	54.9 ‡	77.0	39.3	30.9
S1A	63.3 ‡	53.8 ‡	77.4	38.8	30.6
S2	74.5 (+0.6)	63.6 (+0.5)	81.3	49.2	40.4
Early Fusion	76.5 (+1.6)	65.8 (+0.9)	**82.2**	**50.6**	**42.0**
Mid Fusion	76.5 (+1.4)	-	-	-	-
Late Fusion	**77.2 (+4.2)**	**66.3** ⋆	81.6	50.5	41.6
Decision Fusion	75.8 (+3.3)	64.3 ⋆	-	-	-

Robustness to Clouds. In Figure 5.15b, we report the robustness to cloud cover brought by the fusion schemes and enhancements by artificially masking optical acquisitions during inference for parcel classification and semantic segmentation. Even with 70% to 90% of optical acquisitions missing, the fusion schemes outperform the model operating on radar time series.

Optical and radar acquisitions have known synergies for crop mapping. We design several multimodal schemes and related enhancements, resulting in higher precision and robustness to varying cloud cover across different crop mapping tasks.

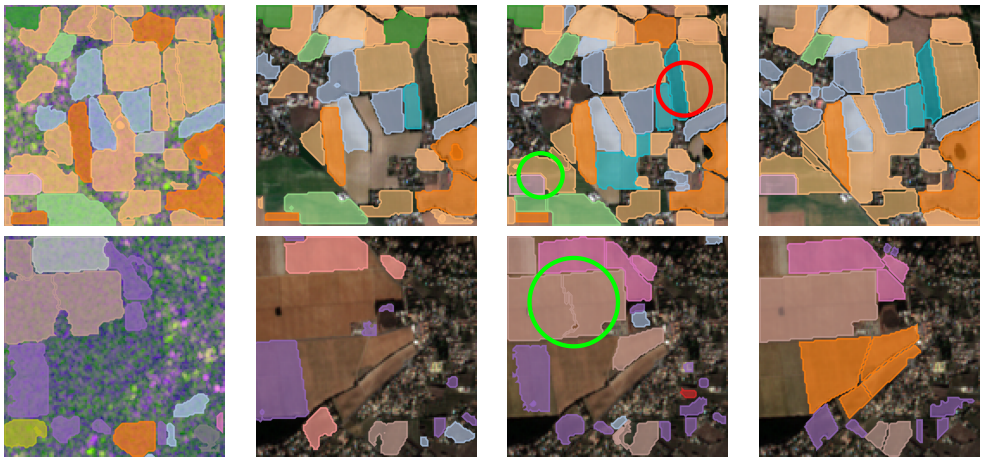


(a) Parcel-based classification



(b) Semantic segmentation

Figure 5.15: **Varying Cloud Cover Experiment.** We evaluate different models with varying ratios of available optical observations remaining. In both parcel-based classification (a) and semantic segmentation (b), the fusion models prove robust to a reduced number of optical observations.



(a) S1D (b) S2 (c) Fusion (d) GT

Figure 5.16: **Qualitative Results for Panoptic Segmentation.** We compare the predictions made by unimodal models operating on S1D (a), S2 (b), the late fusion model (c), and the ground truth annotations (d). Some parcels are misclassified when only using the optical modality but are successfully recovered by the radar and fusion models (green circle ○). Some parcels are only detected using both modalities (red circle ○).

5 Modelling Crop Rotations

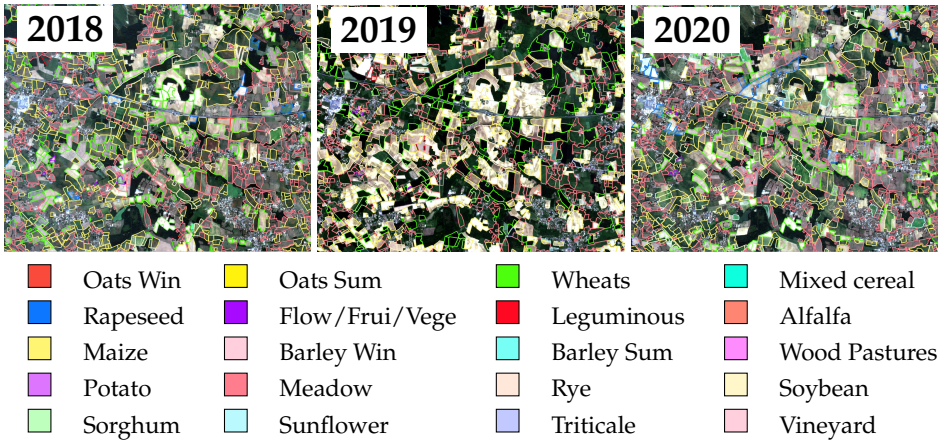


Figure 5.17: **Multiyear Sentinel-2 Data.** Crop type for a part of our area of interest across three years.

The impact of crop rotations is extensively studied in the agricultural optimization literature [DSG+12, DRVI03, BF15]. Given their measurable impact on crop yields [KKN+15], it is natural to assume that crop rotations significantly influence the choice of cultivated species. Consequently, taking past cultures into account when classifying parcels should improve our precision.

However, most crop mapping methods operate on a single year's worth of temporal acquisitions and ignore the inter-annual crop dynamics represented in Figure 5.17. Previous works propose to use probabilistic Markov models [OID15] or Conditional Random Field (CRF) [GBLC20] to learn transition statistics explicitly. Yaramasu *et. al* [YBP20] analyze multi-year data with a deep convolutional-recurrent model, but only select one image per year, thus ignoring intra-year dynamics. In contrast, we propose a model that operates at the intra-year scale through yearly sequences of observations and at the inter-annual scale by considering past cultures.

Multi-Year Dataset. In order to evaluate the impact of crop rotation, we present Sentinel2Agri-Multi, a version of the Sentinel2Agri dataset spanning 2018, 2019, and 2020. We select 103 602 *stable* parcels only, *i.e.* whose contours only undergo minor changes across the three studied years. Our dataset associates each parcel with three image time series and three crop annotations corresponding to the years 2018, 2019, and 2020.

Inter-Year Training. Given several years of annotated acquisitions, we observe that training a single model on a single multi-year dataset is ben-

eficial compared to training different models for each year independently. Indeed, the increased volume of data and varying temporal domains result in a network with better generalization, and mitigate the rare class issue. Implemented with a PSE+L-TAE baseline, the mixed-year approach outperforms specialized networks trained on a single year worth of data *even for their year of specialization* by up to 5 mIoU points for parcel classification. These results should encourage practitioners to combine several years of annotations when constituting SITS datasets.

Modeling Rotations. We consider the problem of classifying a parcel given SITS corresponding to several years, along with ground truth labels for the *past years*. We propose three strategies to exploit this data:

- M_{obs}: we process all yearly SITS independently and concatenate their spatio-temporal embeddings before classification.
- M_{lab}: we only process the SITS of the target year and concatenate to the resulting spatio-temporal feature the sum of the one-hot-encoded past labels.
- M_{CRF}: we only process the SITS of the target year and use a second-order CRF to model crop rotations from past labels.

These models can learn to resolve ambiguous observations by considering the history of a parcel. By processing past knowledge of labels and current observations end-to-end, the model M_{lab} performs best. As shown in Table 5.6, this results in an appreciable improvement of over 6.3 points compared to only considering the last SITS. Further analysis shows that the most significant increase (16.9%) is for permanent crops (*e.g.* vineyards, meadow), followed by crops with statistically significant rotation patterns (*e.g.* rapeseed, soybeans) with 7.6%. Even for crops with no apparent rotation rules, this model provides an increase of 2.3% on average.

> We model multi-year patterns of agricultural cultivation by considering the parcel crop type history. Combined with our temporal attention model, this leads to a significant improvement across all cultures.

Table 5.6: **Performance by model.** Performances of the models M_{obs} and M_{lab} for the year 2020. We compare with the model M_{single} trained on only the target year, and a CRF baseline M_{CRF}.

Model	Description	OA	mIoU
M_{single}	last SITS only with L-TAE	96.8	68.7
M_{obs}	SITS for all 3 years with L-TAE	96.8	69.3
M_{CRF}	last SITS with L-TAE + past labels with CRF	96.8	74.4
M_{lab}	last SITS + past labels with L-TAE	97.5	75.0

6 Leveraging Class Hierarchies

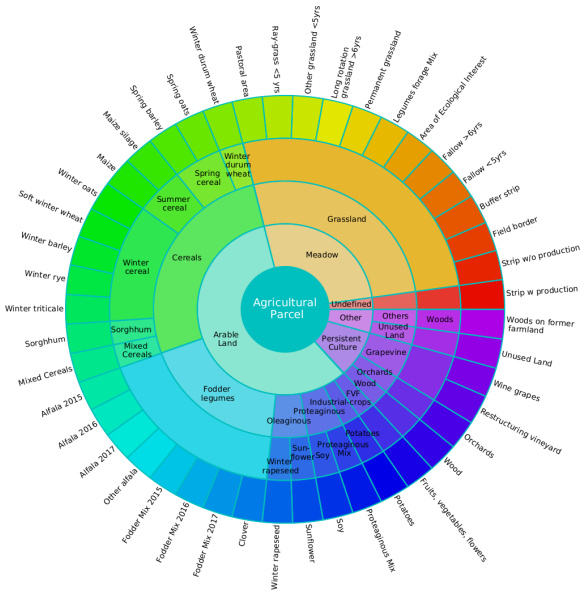
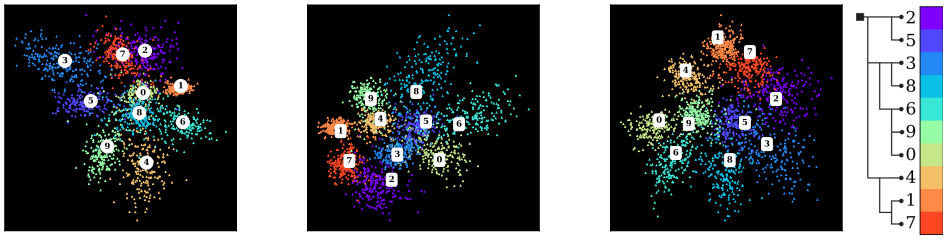


Figure 5.18: **Class hierarchy for S2-Agri.**

Classification models typically focus on maximizing prediction accuracy, regardless of the semantic nature of errors. A step towards more reliable and interpretable algorithms would be to explicitly model and measure this gravity of errors [BMT+20]. In many classification problems, we can organize the class set according to a tree whose structure encapsulates the semantic similarity and discrepancy between classes in a hierarchical fashion. For a classification task over a set $\mathcal{K}$ of K classes, we represent the class tree by a finite metric $D \in \mathbf{R}_+^{K\times K}$ such that $D[k,l]$ is the length of the shortest path between class k and l in the nomenclature tree. For a dataset indexed by $\mathcal{N}$, the *Average Hierarchical Cost* (AHC) between class predictions $y \in \mathcal{K}^{\mathcal{N}}$ and the true labels $z \in \mathcal{K}^{\mathcal{N}}$ is defined as [RDS+15, DBLFF10]:

$$\mathrm{AHC}(y,z) = \frac{1}{|\mathcal{N}|}\sum_{n\in\mathcal{N}} D[y_n, z_n]\ . \tag{5.7}$$

Crop mapping especially benefits from predictions with a low hierarchical cost. Indeed, payment agencies monitor the allocation of agricultural subsidies and whether crop rotations follow best practice recommendations [Gra97]. The monetary and environmental impact of misclassifications are typically reflected in the class hierarchy designed by domain experts [BF15, Bul92], see Figure 5.18. By achieving a low AHC, we ensure that these downstream tasks can be meaningfully realized.



(a) Linear Classifier, = 0.42, ER= 15.2%, AHC = 0.81

(b) Prototypical Classifier , distortion= 0.47, ER= 14.2%, AHC = 0.75

(c) Guided Prototypes, distortion= 0.22, ER= 11.9%, AHC = 0.52

Figure 5.19: **Metric Guided Prototype.** We propose a hierarchical tree representing the visual similarity between digits. The baselines (a) linear classifier and (b) prototypical classifier ignore this prior. The proposed regularization term used in (c) leads to prototypes whose arrangement is consistent with the class hierarchy. This results in a decrease of *Average Hierarchical Cost* (AHC), as well as *Error Rate* (ER), indicating that the taxonomy may contain useful information for learning better visual features.

Beyond reducing the AHC, the class hierarchy may also contain information about the data structure. Although it is not always the case, cohyponyms (*i.e.* siblings) in a class tree tend to share some structural properties. Encouraging such classes to have similar representations could lead to more efficient learning, *e.g.* by leveraging shared feature detectors, as exemplified in Figure 5.19.

Metric-Guided Regularization. We propose to integrate a pre-defined class hierarchy into a prototype-based network [YZYL18, CLT+19]. We introduce a regularization term encouraging the pairwise distance between prototypes to reflect the error cost defined by the class tree. We define a function $f : \mathcal{X} \mapsto \Omega$ mapping inputs in $\mathcal{X}$ to the representation space Ω. Each class k is associated with a prototypes $\pi_k \in \Omega$. A sample x_n is classified with the label k if $f(x_n)$ is closer to π_k than all other prototypes. Both f and π can be learned jointly with log-likelihood maximization [SSZ17].

We equip the embedding space Ω with a distance function $d : \Omega \times \Omega \mapsto \mathbb{R}_+$, such that (Ω, d) forms a continuous metric space. We say that the prototypes π are consistent with the finite metric D if their pairwise distance in (Ω, d) reflects their distance in D: the mapping $k \mapsto \pi_k$ has low distortion, as defined by De Sa *et. al* [DSGRS18]. We argue that this property should lead to a lower hierarchical error: *near* misclassifications are more likely to result in a *related* class. However, the scale induced by the tree may be in conflict with the optimal arrangement of prototypes to minimize the classification loss. We propose a new regularizer which is a

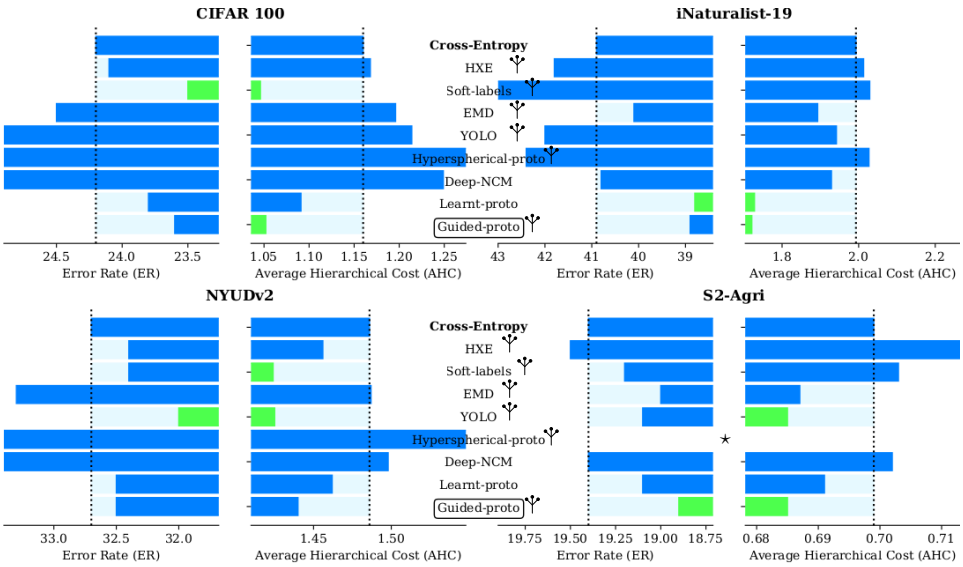


Figure 5.20: **Quantitative Results.** Error Rate (ER) in % and Average Hierarchical Cost (AHC) on four datasets for `Guided-proto`, the Linear Classifier baseline (`Cross-Entropy`), and competing approaches. Methods that use the hierarchical knowledge are indicated with the symbol $\curlyvee$. We plot the best performances on each dataset in green. Our guided prototype approach improves the ER and AHC across the four datasets compared to the baseline. $\star$: would not meaningfully converge.

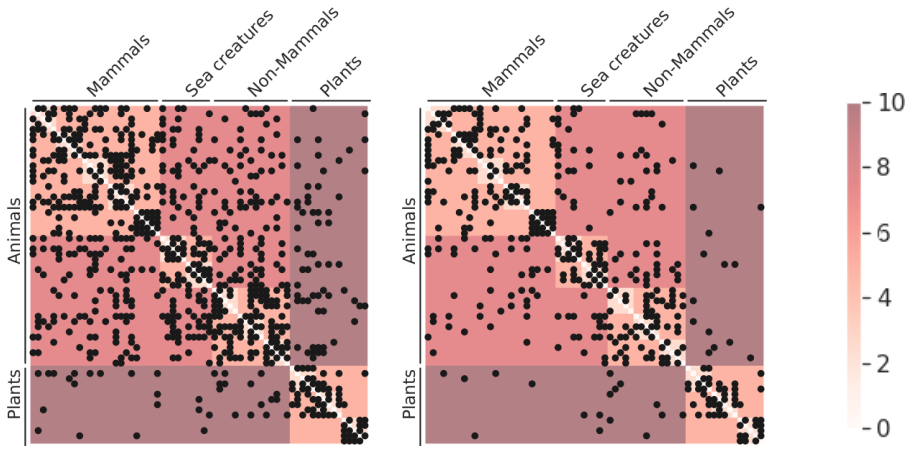
scale-free and differentiable surrogate of distortion:

$$\mathcal{L}_{\text{disto}}(\pi) = \frac{1}{K(K-1)} \min_{s \in \mathbf{R}_+} \sum_{k,l \in \mathcal{K}^2, k \neq l} \left(\frac{s d(\pi_k, \pi_l) - D[k,l]}{D[k,l]} \right)^2 . \tag{5.8}$$

Minimizing this regularizer encourages the pairwise distances between prototypes $d(\pi_k, \pi_l)$ and their classes $D[k,l]$ to be close. Note that the inner minimization problem can be efficiently solved in closed form.

6.1 Numerical Experiments.

We evaluate our approach across different tasks and datasets with fine-grained class hierarchies: image classification on CIFAR100 [KH+09] and the 1010-class iNaturalist-19 [VHMAS+18], RGB-D image segmentation on NYUDv2 [NSF12], and image sequence classification on S2-Agri. We define class hierarchies for each dataset, such as the one represented in Figure 5.18. We only use classical backbone networks: ResNet18 [HZRS16] for CIFAR100 and iNaturalist-19, FuseNet [HMDC16] for NYUDv2, and PSe+LTAE for S2-Agri.



(a) Linear Classifier (b) Ours (c) Cost

Figure 5.21: **Effect of Regularization.** Partial confusion matrices for the "living organism" class subset of CIFAR100 for the linear classifier baseline (a) and our approach (b). For readability, we only display (in black) entries of the matrices with at least one confusion. We also represent the cost of confusing different classes in shades of reds (c). We note that our approach yields fewer confusions between pairs of classes with high costs, such as plants and animals.

We compare our approach with a linear classifier baseline and several reimplemented methods designed to leverage extrinsic class hierarchies. As seen in Figure 5.20, the benefits provided by our approach appear on all datasets. Compared to the linear classifier baseline, our model improves the AHC by 3% on NYUDv2 and S2-Agri, and up to 9% and 14% for CIFAR100 and iNat-19, respectively. While some methods perform on par or better than ours for some datasets, only our metric-guided prototypes consistently reduce the hierarchical cost across all tasks and datasets. We also observe a relative decrease of the error rate by 3 to 4% across all datasets compared to the linear classifier baseline. This indicates that cost matrices derived from the class hierarchies can help neural networks learn more discriminative representations.

In Figure 5.21, we represent the partial confusion matrices of the cross entropy baseline and our approach on CIFAR100, illustrating that guiding prototypes with a finite metric lead to fewer "high-cost" errors.

We model the agreement between a prediction and a tree-shaped class hierarchy as the distortion between two metric spaces. Combined with a versatile prototype-based approach, our model signifi-

cantly reduces the weighted and unweighted error rates across various datasets and tasks.

Conclusion

SITS analysis is a high-impact and complex machine learning task with unique challenges. We propose a series of methods leveraging the particular structure of SITS for crop-type mapping applications. Our models lead to higher precision, faster inference, and lower memory requirement for parcel classification problems. We also propose the first method for panoptic segmentation on SITS, and new ways to exploit the synergy between remote sensing modalities and the semantic structure of complex class sets. We release 4 unique and novel datasets aiming to popularize SITS analysis as a machine learning task and to encourage more reproducible science.

CHAPTER 6

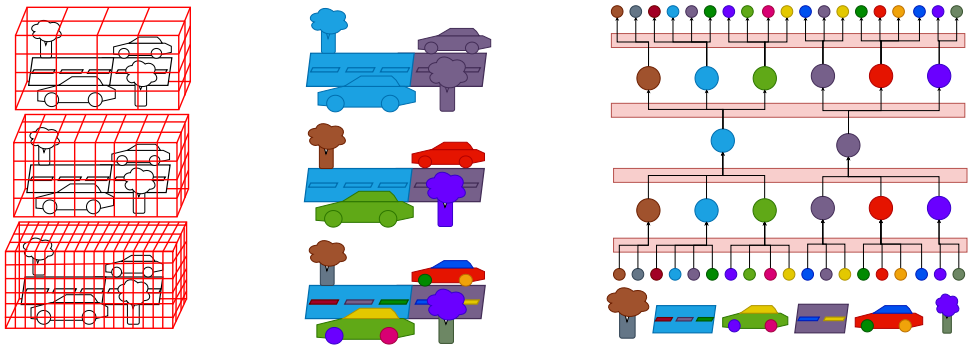
Perspectives

I identify two main directions for future work on geospatial machine learning: efficient learning with hierarchical partitions and reciprocal learning with many modalities.

1 Efficient Learning with Hierarchical Partitions

The cost of training state-of-the-art neural networks has reached unprecedented levels, effectively pricing out most public actors and practitioners [BHA+21]. Indeed, recent deep learning advances favour generality and scalability over efficiency [JGB+21], raising questions regarding the sustainability of the field. Exploiting the spatial regularity of geospatial processes could be the key to more sober and efficient machine learning. When analyzing a scene, we do not give the same attention to each square centimetre of every surface. Instead, we instinctively group our environment into homogeneous regions of various sizes and analyze their relationships at different scales: *this car is composed of four tires, a hood, and a roof; is on a road and under a tree; is in the middle of a commercial district.* However, traditional deep learning methods process the input's atomic elements (pixels, voxels, 3D points) *uniformly* despite the highly redundant information they carry, which leads to wasteful computations. Furthermore, this restricts the quantity of data considered simultaneously and prevents the modelling of long-range interactions. We could achieve more efficient learning by mimicking this two-step process: first compute a hierarchical partition of the input into homogeneous regions dubbed super-elements (*e.g.* , super-pixel, super-points), then perform a multi-scale analysis of the super-elements. This would allow efficient and parsimonious high-level and large-scale reasoning without considering millions of individual atoms.

This approach goes exactly against the trend of large, generic, and powerful models [BHA+21] which aim to be free from any sort of *inductive bias* or human expertise [JGB+21]. The main idea is to leverage the spatial regularity of many spatial processes [Tob70]. This property translates into the spatial regularity of many latent variables of interest such as semantic or instance labels, or surface parameterization. Consequently, the solution to many spatial analysis problems is constant with respect to a partition of the input with much fewer components than the number of atomic elements. By computing such partition at multiple nested scale, we can define



(a) Grid Partition. (b) Hierarchical Partition. (c) Network Architecture.

Figure 6.1: **Architecture from Structure.** Instead of using cookie-cutter grid partitions (a), we build a hierarchical partition of a scene (b). We can then define a network whose architecture (c) is directly derived from the partition itself instead of from uniformly spaced convolutions. This allows the analysis of the input data to adapt to its local complexity. The red shaded rectangles indicate a step in which super-elements of the same level exchange information, *e.g.* with a transformer architecture.

a hierarchical partition encompassing both small details and large structures. Instead of using this prior to improve atomic predictions *ex-post* [KK11, LRV+17, MKH19], we could define a network whose entire structure adapts to the data, while reducing the complexity of learning problems by several orders of magnitude. This would significantly accelerate inference, training, and lead to more compact models with lighter hardware requirements. Furthermore, multi-level hierarchical partitions allow us to model long-range interactions, which would be particularly beneficial for geospatial data.

2 Cross-Modal Reciprocal Learning

The supervised learning paradigm requires costly annotations, which can be particularly expensive in some geospatial applications. For example, forestry labels must be collected *in situ* in poorly accessible areas [KLMC22a, KLMC22b], making large-scale annotation prohibitively costly. Instead, we could train models by exploiting the diversity of remote sensing sensors, their global scope, and systematic georeferencing.

Text-Image contrastive pretraining [CHL05] has shown impressive results in computer vision [OLV18], leading to influential foundation models [RKH+21, YCC+21]. Contrastive learning has been recently explored for EO by exploiting the spatial alignment between time series [AUM+21] and for cross-modal localization [TLB+22]. We could generalize this ap-

proach to the multi-modal setting with georeferenced observations. In other words, we want to align spatially the features extracted from acquisitions with different modalities: the descriptor of an area should be consistent regardless of the scanning sensor and different from other areas. By forcing spatial alignment across sensors capturing different information, the features must describe the only shared latent variable: the actual content of the considered area.

Generalizing contrastive learning to the multi-domain setting raises several theoretical and technical challenges. First, the classic two-modalities formulation leads to an exponential complexity *w.r.t* the number of sensors, quickly becoming impractical. Furthermore, current work produces a single descriptor per image, whereas we will consider pixel descriptors, worsening the combinatorics of the problem. Second, cross-modal learning implies the simultaneous training of several large networks and the manipulation of costly multi-modal batches. This raises technical issues such as inefficiency in memory use and prolonged training times. Lastly, if we only optimize the encoders associated with different sensors to produce spatially aligned features, sensor-specific information may be ignored. This would result in weaker individual representations discarding the unique strength of each sensor at the benefit of the *lowest common denominator*. However, such training would have a considerable impact as we could train single and multi-modal models using nothing but geolocalized observations. The resulting feature extractors could be fine-tuned on downstream tasks with much fewer annotations than if trained from scratch.

3 Other Works

I have also had the pleasure of collaborating with Mathieu Aubry (ENPC) on unsupervised object discovery in object-level [LMAL21] and large-scale [LVAL23] 3D datasets. Our approach involves learning representations in input space, rather than in an abstract feature space, allowing the models to provide interpretable and editable visualizations of their reasoning. This property is beneficial for application to industrial and public-policy settings. We also propose an application to sound discovery [LBT+22], showcasing the versatility of our proposed approach.

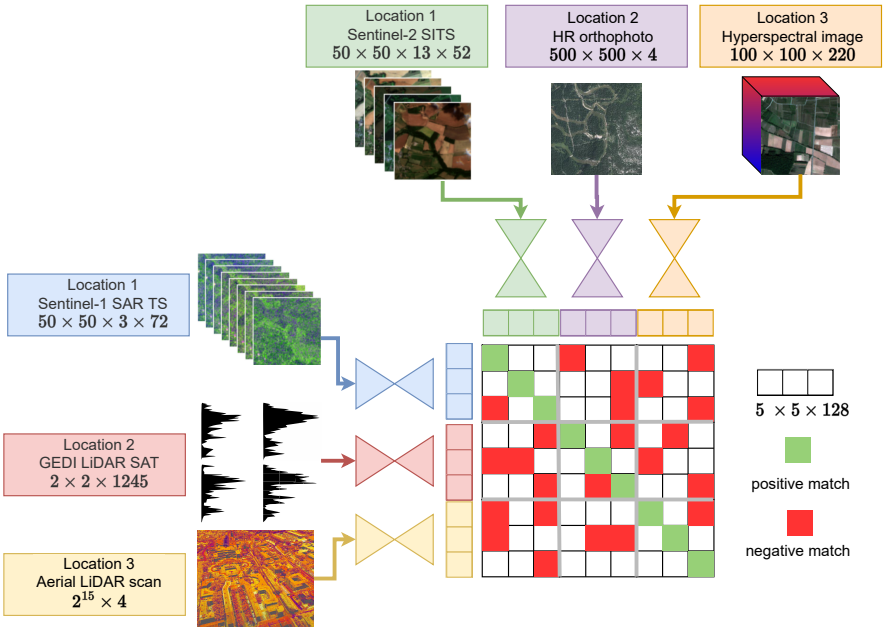


Figure 6.2: **Cross-Modal Contrastive Learning.** We consider a set of encoders for different inputs: image time series, LiDAR scans, hyperspectral images, etc. Each multi-modal patch is embedded by a dedicated encoder into a small 5×5 feature map of width 128, whose cells are trained to be co-linear with the corresponding cell in the same modality and contrasted with others.

CHAPTER 7

Curriculum Vitae

Short CV

Summary I am a machine learning/computer vision researcher with a strong interest in geospatial applications. My multi-disciplinary research aims to exploit the specific structure of complex tasks to develop more efficient solutions. I strive to create links between machine learning and geospatial communities by organizing workshops and conferences and co-chairing working groups.

Positions and Education

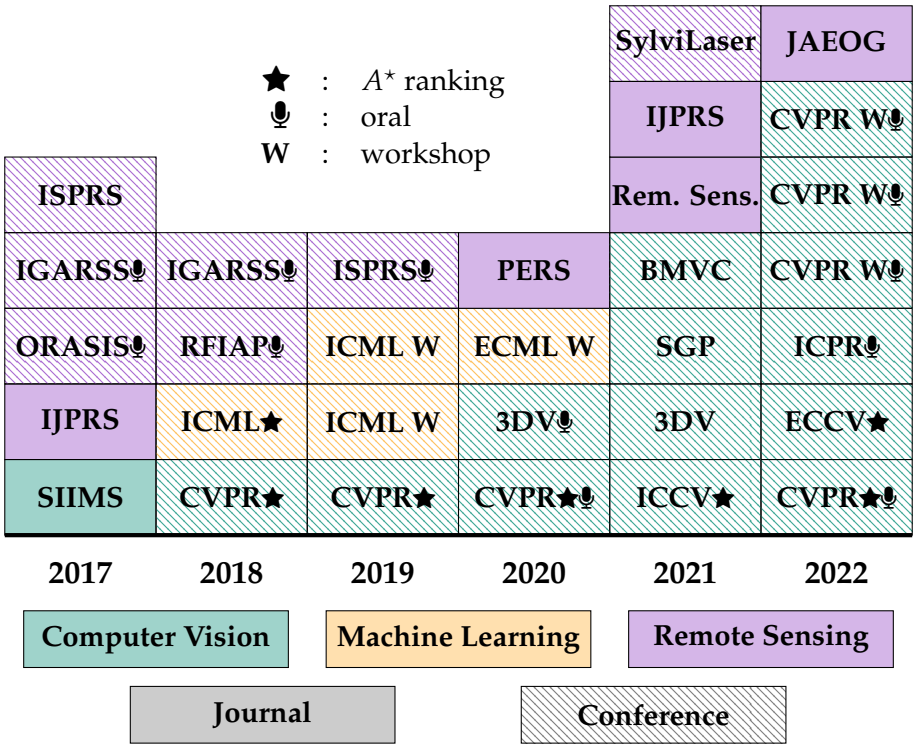
Since 2012, I have the status of civil servant of the French Ministry of Ecology (IPEF).

- **2023-, IMAGINE, LIGM, ENPC** : *Researcher*,
- **2015–2023, IGN/ENSG, LASTIG, UGE** : *Researcher*,
- **Sep 2012 - Sep 2016, ENPC ParisTech - INRIA, PhD**
  Learning structured models on weighted graphs, with applications to spatial data analysis. **Advisors:** Francis Bach and Guillaume Obozinski.
- **2011 - 2012, ENS Cachan, MSc.** Master MVA, machine learning.
- **2011 - 2012, ENPC ParisTech, MSc.** Master IMI, computer science.
- **2007 - 2011, Ecole Polytechnique, MSc.** Algorithmics.

Research

Software Development I am committed to reproducible research: every published article comes with its open-source implementation. I authored or participated as an advisor to over 30 repositories, including the following highlights:

- [GitHub]loicland/superpoint-graph 642★ 207
- [GitHub]loicland/cut-pursuit 64★ 18
- [GitHub]torch-points3d/torch-points3d 2001 ★ 338
- [GitHub]VSainteuf/pytorch-psetae 128 ★ 29
- [GitHub]drprojects/DeepViewAgg 148 ★ 15



Breakdown of my post-PhD Publications across different fields.

Datasets My students and I released 7 open-access datasets, totalling 2500+ downloads or over 300To :

- **HelixNet:** largest 3D datasets (10B points) for autonomous driving, contains unique sensor information for precise latency measurements.
- **PASTIS:** first dataset of satellite image time series with parcel borders and content (4000km^2, 2B pixels).
- **PASTIS-R & Multi-Year Sentinel:** companion datasets focusing on multimodality and multi-year acquisition, respectively.
- **StrataNet & WildForest3D:** first densely and weakly annotated aerial LiDAR datasets for forest inventory (over 2000 individual trees).

Invited Talks I have given over 20 invited talks, both for **academic institutions** (ETH Zurich, Ecole Polytechnique, Univ. Bologna, Politecnico di Milano, Univ. Erlangen, Tech. Univ. Danemark, Norwegian Inst. for Bioeconomics Research, Univ. Aix-Marseille, Univ. Montpelier, Univ. Paris-Est, SIAM symposium) and **leading industry research centers** (Valeo AI, Facebook AI Research, Sony CSL, ENGIE CRIGEN, Quantcube Tech.).

I also gave **keynote addresses** to the Urban3D ICCV Workshop (2021), the French Robotic Research Symposium (2019), the French Workshop on Mobile and Airborne LiDAR (2022) , the 2nd International Workshops on

Point Cloud Processing (2019,2023), and the French Workshop on Graph Signals Processing (2019).

Supervision and Community

Graduated PhD students (with percentage of supervision):

Raphael Sulzer	2018-2022	40%	now at INRIA Titane, FRA
Vivien S.F Garnot	2018-2021	90%	now at Univ. of Zurich, CHE
Stéphane Guinard	2017-2020	50%	now at Univ. of Laval, CAN
Mohamed Boussaha	2017-2020	50%	now at Gambi-M, FRA

Ongoing students (with percentage of supervision):

Damien Robert	2020-2023	100%	with ENGIE & B. Vallet
Romain Loiseau	2020-2023	80%	with M. Aubry (ENPC)

Ongoing post-Doctoral Fellow: Ekaterina Kalinincheva Jan 2021-Jan 2023
Internships: 14 master & industry interns

Organization & Community

- **Program chair** of the XXIV ISPRS Congress (2022), **main event of the Remote Sensing community** (every 4 years), 743 papers published and 1500 participants from 59 countries, coordinating 200+ area chairs [LRE+22].
- **Organizing committee** of Earth Vision (2021,2022,2023), a **leading CVPR worskhop** at the intersection between remote sensing and computer vision.
- **Program chair** of the Conference on IGN Research (2019, 2020), (800+ participants) and LASTIG Research Seminar (250+ participants).
- **Editorial.**
  - Editorial advisory board of IJPRS (Elsevier, IF:11.8)
  - Reviewing Committee of Remote Sensing (MDPI, IF:5.3)
  - Guest editor for IJPRS (Elsevier, IF:11.8).
- **Scientific Society:**
  - Co-chair of the ISPRS WG *Temporal Geospatial Data Understanding*.
  - Co-lead of the IEEE GRSS WG *Image and Signal Processing*.
- **Reviewing:** ICML , CVPR , ISPRS Congress, NeurIPS, ICCV, ECCV, ICLR, BMVC, IJCV, PAMI, IJDSA, IJPRS.
- **Awards:**
  - Outstanding reviewer: ICML2021 (top 10%), CVPR2021 (top 10%), ECCV2022 (top 5%), ISPRS Congress 2022 (top 5%).
  - CVPR 2022 Best paper finalist (top 33 / 2065).
- **Expertise:** ANR Grants, Dutch Research Council (NWO), Canadian Centres of Excellence (Mitacs).
- **Jury Member:** 3 PhD jury, 3 PhD committees, 1 assistant professor Jury.
- **Scientific Advisory Board** of the AI-based startup SAMP (samp.ai).

Teaching

- ENSG, *Course Instructor and Creator*, Deep Learning for Remote Sensing and Introduction to Machine Learning (24 hours yearly).
- EduSERV for EUROSDR, *MOOC instructor and course creator*, Deep learning for remote sensing (2 weeks).
- Master IMI (ENPC), *Teaching assistant*, Machine learning.
- Master MVA (ENS), *Teaching assistant*: Probabilistic graphical models, *Invited Speaker*: 3D Deep Learning
- Master AMMI (Rwanda), *Teaching assistant*, Probabilistic Graphical Models (2 weeks intensive training).

Projects and Grants

- **Principal investigator** of the ANR JCJC **READY3D**: REal-Time Analysis of DYnamic LiDAR 3D Point Clouds (total cost: 476k€, 194k€ subsidy).
- Participant of the **BIOM** ANR Project (total cost: 1 776k€, 723k€ subsidy).
- DGA PhD Grant (100k€ subsidy).
- ASP Agence de Services et de Paiement - 300k€ research Grant.
- Joint PhD between ENGIE, IGN and Univ. Paris Est (250k€).

Synthèse en Français

De par leur richesse, échelle et complexité, les données géospatiales sont un excellent terrain d'inv-estigation pour l'apprentissage automatique. Plutôt que d'utiliser des outils existant issus de la vision par ordinateur ou de l'apprentissage machine, notre motivation est d'exploiter la structure particulière des données géospatiales pour proposer des algorithmes adaptés et innovants. En accordant nos méthodes aux tâches considérées, nous atteignons d'important gains de vitesses, de performance, et une réduction drastique de la taille des modèles considérés.

1 Contributions Scientifiques

Je présente dans cette section un rapide aperçu de mes contributions scientifiques et de mes charges d'encadrements et organisationnelles à l'IGN au cours de ces 6 dernières années.

J'ai ensuite rejoint l'IGN (LASTIG, ENSG, UGE), où je me suis intéressé à l'apprentissage automatique pour les données géospatiales.

Publications. Je situe mes recherches à l'interface entre la vision par ordinateur et l'apprentissage automatique d'une part, et la télédétection et photogrammétrie d'autre part. Je publie avec mes étudiants dans des journaux spécialisés et thématiques de la communauté de télédétection et de photogrammétrie (IJPRS, JAEOG, PERS, Remote Sensing) mais aussi dans les conférences méthodologiques très sélectives de vision et d'apprentissage. En particulier, j'ai depuis ma thèse publié à CVPR (4 papiers dont 2 oraux et un *best paper finalist*, 3 workshops), ICML (1 papier et 2 workshops), et 2 papier à I/E.CCV. J'ai aussi publié dans des conférences plus spécialisées comme 3DV (2 papiers dont 1 oral), BMVC, SGP Eurographics et ICPR.

Logiciel et Donnée Libre. Mes étudiants et moi avons publié plus de 22 dépôts de code libres liés à nos projets, accumulant plus 3000 étoiles sur GitHub et 500 branches. Nous avons aussi constitué 7 jeux de données libres aux propriétés uniques, totalisant plus de 2000 téléchargements.

Organisation. J'ai été *Program Chair* du congrès 2022 de l'International Society of Photogrammetry and Remote Sensing (ISPRS), le plus grand évènement de la communauté, qui a lieu une fois tous les 4 ans et a rassemblé 1500 chercheurs de 68 pays différents. Je participe aussi à l'organisation du Workshop CVPR EarthVision (2021 et 2022), qui a pour but de rassembler les chercheurs en vision intéressés par les

problématiques d'observation de la Terre. J'ai aussi organisé les journées de la recherche de l'IGN sur le thème de l'IA pour la géomatique (800 participants, virtuels) et plusieurs séminaires inter-disciplinaires sur le sujet rassemblant jusqu'à 250 participants (virtuellement).

Responsabilité Éditoriale. Je suis au comité de rédaction du journal de l'ISPRS (IF: 9.0) et de Remote Sensing (IF: 5.3). Je fait partie des comités de lecture de nombreuses conférences et journaux de vision par ordinateurs (CVPR, ECCV, ICCV, ACCV, BMVC, PAMI, IJCV) et d'apprentissage (ICLR, NeurIPS, ICML). J'ai été distingué plusieurs fois comme *outstanding reviewer*: CVPR21, ICML21, ISPRS22. Depuis 2022, je suis également *Working Group Officer* pour le pôle Understanding Temporal Data de l'ISPRS, et *Co-lead* du *Working Group* IEEE GRSS sur Image and Signal Processing.

Encadrements. J'ai co-encadré 3 doctorants qui ont défendus avec succès leur thèse, et je co-encadre actuellement 3 autres étudiants et une post-doctorante. J'ai encadré 14 stages de master ou industriels.

Financements. Je suis porteur principal de l'ANR JCJC ReADy3D sur la perception 3D temps réel pour la conduite autonome (476k€, 194k€de subvention). Je participe aussi à ANR (BIOM, 1 776k€), et ai contribué à l'obtention de financements auprès d'acteurs privés (ENGIE: 250k €) et publics (ASP: 300k €, DGA : 100k €).

2 Spécificités des Données Géospatiales

L'analyse des données géospatiales a plusieurs atouts qui en font un excellent terrain d'application pour l'apprentissage automatique. En particulier, ces données sont typiquement acquises à grande échelle par une variété de capteurs, et possèdent une structure complexe qui mêle les dimensions spatiale, temporelle, et spectrale. Il existe de nombreuses sources de données géospatiales et d'annotations en libre accès. Leur analyse automatique à grande échelle mène de nombreuses applications à fort impact, autant économique qu'environnemental et social.

Malgré une ressemblance superficielle, les données géospatiales présentent des différences importantes avec les images ou vidéos *naturelles* typiquement traitées en vision par ordinateur. La structure unique des données géospatiales nécessite donc de développer des méthodes adaptées. Le principe guidant l'ensemble de mes travaux est d'exploiter cette structure pour mettre au point des algorithmes et architectures adaptés afin d'améliorer la vitesse et précision de leur analyse.

Dans ce manuscrit, nous présentons certains des travaux que j'ai conduit avec mes étudiants depuis l'obtention de mon doctorat en septembre

2016. En particulier, nous développons la manière dont nous avons exploité les structures suivantes:

- *Structure de régularité sur Graphe:* nous proposons une approche mathématique pour résoudre certains problèmes d'optimisation en tirant parti d'une propriété de régularité habituellement rencontrée lors de l'analyse de données géospatiales.
- *Structure de Régularité des Données 3D:* ces méthodes exploitent le même principe que le chapitre précédent mais dans le cadre de l'apprentissage profond pour l'analyse automatique de grands nuages de points.
- *Structure des Capteurs de Données 3D:* nous proposons plusieurs approches exploitant la structure particulière de certains capteurs 3D pour améliorer la rapidité et la précision de leur analyse.
- *Structure des Séquences Temporelles Satellite:* nous présentons une série d'algorithmes pour l'analyse automatique de séries temporelles d'images satellites, améliorant significativement l'état de l'art de ce domaine.

3 Structure de Régularité sur Graphe

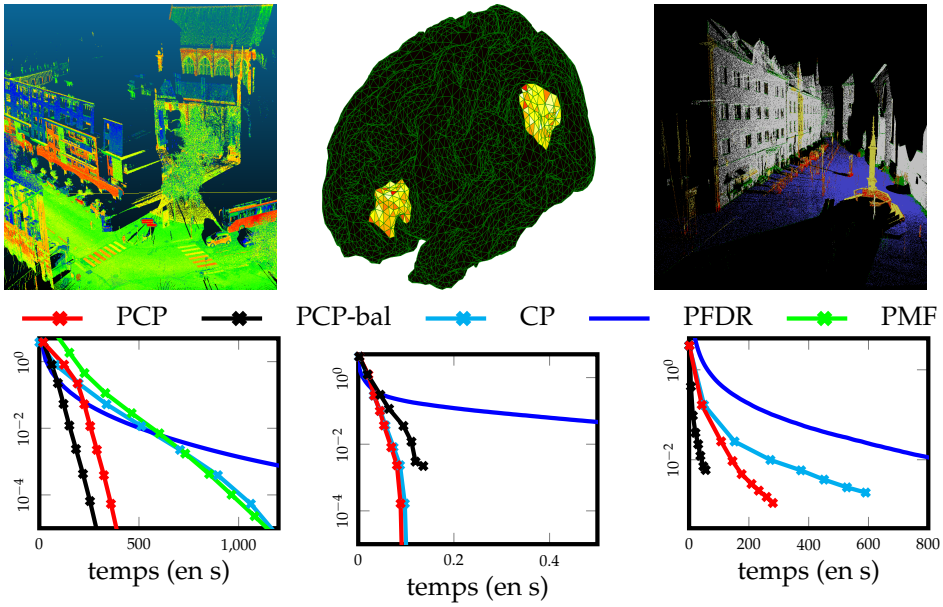
Cette première section présente nos travaux les plus théoriques, où nous montrons comment une propriété abstraite de régularité définie par rapport à un graphe général peut être exploitée pour considérablement accélérer la résolution d'une large classe de problèmes d'optimisation. Dans certaines circonstances, notre approche apporte également des garanties théoriques inédites et nécessitant peu d'hypothèses sur les fonctions considérées. Nos travaux s'appliquent à de nombreux problèmes géo-spatiaux, pour lesquels nous parvenons à réduire les temps de calculs de plusieurs ordres de grandeur, *c.f.* Figure 7.1. Cette section reprend les publications suivantes:

[LO17] Loic Landrieu, Guillaume Obozinski, "Cut Pursuit: Fast Algorithms to Learn Piecewise Constant Functions on General Weighted Graphs", *SIAM Journal of Imaging Science*, 2017

[RL18] Raguet Hugo, Loic Landrieu, "Cut-Pursuit Algorithm for Regularizing Nonsmooth Functionals with Graph Total Variation", *ICML*, 2018

[RL19] Raguet Hugo, Loic Landrieu, "Parallel Cut Pursuit For Minimization of the Graph Total Variation", *ICML Workshop on Graph Reasonning*, 2019

[GLCV19] Stéphane Guinard, Loic Landrieu, Laurent Caraffa, Bruno Vallet, "Piecewise-Planar Approximation of Large 3D Data ss Graph-Structured Optimization", *ISPRS Annals*, 2019



(a) Regularisation LiDAR. 109m vertices, 317m edges.

(b) Inversion d'ECG. 19k sommets, 29k arêtes.

(c) Régularisation sémantique. 3m sommets, 17m arêtes.

Figure 7.1: **Cut Pursuit.** Évolution de la distance à la solution pour différentes versions de Cut Pursuit (CP, PCP, PCP-bal) et autres méthodes (PFDR [RL15], PMF [CD09]) et différentes tâches. L'accélération peut atteindre plusieurs ordres de grandeurs.

Contexte. Nous considérons le problème d'optimisation consistant à minimiser une fonction dont les variables $x \in \Omega^V$ sont définies selon les sommets d'un graphe $G = (V, E, w)$ avec des arêtes aux poids $w \in \mathbf{R}_+^E$. En particulier, nous nous intéressons aux fonctions F qui peuvent se décomposer de la manière suivante :

$$F(x) := f(x) + \sum_{(u,v)\in E} w_{u,v} h(x_u, x_v) \,, \tag{7.1}$$

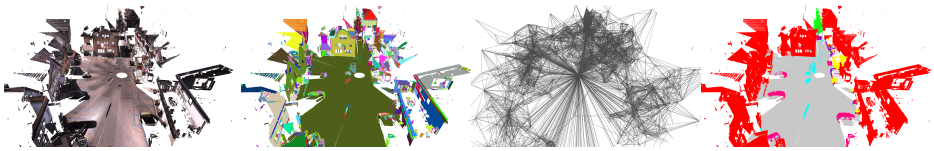
avec $f : \Omega^V \mapsto \mathbf{R}$ et $h : \Omega \mapsto \mathbf{R}$ une fonction qui atteint son minimum uniquement quand $x_u = x_v$. Ces fonctions apparaissent naturellement lors de l'analyse de problèmes ayant une composante spatiale, comme les données géospatiales ou l'imagerie médicale. Cette formulation couvre une large famille de problèmes d'optimisation fonctionnelle, telle que la régularisation par variation totale sur un graphe général ($h = \|\cdot\|$) et la régularisation par longueur des contours ($h = [\cdot \neq 0]$).

Cut Pursuit. Les points critiques $x^\star$ de F exhibent typiquement une forme de régularité par rapport au graphe G: les sommets connectés ont *souvent* des valeurs identiques. Par exemple, le label sémantique associé à un point 3D est partagé par la majorité de ses voisins. En conséquence, les solution $x^\star$ sont constantes par rapport à une partition $\mathcal{V}$ de V avec un faible nombre de composantes. En imposant que les variables x soient constantes par rapport à $\mathcal{V}$, la minimisation de F devient plus simple. En effet, ce problème *réduit* n'a que $|\mathcal{V}|$ variables au de $|V|$, tout en gardant une structure similaire.

Cette propriété peut être exploitée algorithmiquement pour accélérer la minimisation de F. En effet, chaque solution est entièrement définie par une partition et la valeur associée à chacune de ses composantes. L'algorithme cut pursuit alterne entre la minimisation du problème réduit contraint par la partition courante $\mathcal{V}$ et le raffinement de $\mathcal{V}$ en composantes plus petites. La première étape peut être effectuée efficacement de par sa faible dimension, et la seconde peut se formuler sous la forme d'un problème de coupe minimale dans un graphe de flot bien choisi.

Performance. L'approche proposée est avantageuse quand il existe une solution $x^\star$ constante par rapport à une partition $\mathcal{V}$ telle que $|\mathcal{V}| \ll |V|$. Dans ce cas, l'algorithme cut pursuit apporte une accélération pouvant atteindre 1000x, même pour des problèmes inverses complexes. De plus, notre approche est entièrement parallèle grâce a la première implémentation *multi-thread* de l'algorithme de Ford-Fulkerson [FF56].

Garanties. L'algorithme cut pursuit s'applique à une grande variété de problèmes. Pour le cas de la variation totale, notre approche permet



(a) Nuage de points RGB (b) Partition géométrique (c) Superpoint graph (d) Segmentation Semantique

Figure 7.2: **Superpoint Graph.** Un nuage de points (a) est partitionné en formes géométriquement simples (b). Un graphe reliant les superpoints adjacents est construits (c), permettant leur classification (d).

non seulement une résolution efficace, mais apporte un certificat de criticalité à précision machine et une garantie de convergence en un nombre fini d'itérations. Contrairement à la plupart des méthodes existantes, nos preuves ne nécessitent ni la convexité ni la differentiablité de f.

Versatilité. Notre approche peut aussi se décliner pour la régularisation par longueur des contours des zones constantes, qui n'est ni continue, ni différentiable, ni convexe. Néanmoins, notre approche reste plus rapide que le populaire algorithme α-expansion quand la solution recherchée est simple. Nous proposons aussi une variation de notre approche pour l'approximation de grands nuages de points 3D en un faible nombre de plans. Ici encore, quand la solution est bien simple (*i.e.* peu de plans), notre méthode apporte une accélération de plus d'un ordre de grandeur comparé à l'état de l'art.

4 Structure de Régularité des Données 3D

Dans cette section, nous proposons des approches pour la segmentation sémantique et la reconstruction de surface pour de très grands nuages de points. Ces méthodes sont basées sur les mêmes idées que la section précédente, mais adaptées au cadre de l'apprentissage profond. Cette section est principalement basée sur les articles suivants:

[LS18]: Loic Landrieu, Martin Simonovsky, "Large-scale Point Cloud Semantic Segmentation with Superpoint Graphs", *CVPR*, 2018

[LB19]: Loic Landrieu, Mohamed Boussaha, "Point Cloud Oversegmentation with Graph-Structured Deep Metric Learning", *CVPR*, 2019

[SLMV21]: Raphael Sulzer, Loic Landrieu, Renaud Marlet, Bruno Vallet, "Scalable Surface Reconstruction with Delaunay-Graph Neural Networks", *Symposium on Geometry Processing (SGP)*, 2021

SuperPoint Graph. Nous introduisons l'approche Superpoint Graph pour la segmentation sémantique de nuages de points 3D à large échelle. Cet algorithme consiste à calculer une partition du nuage de points en formes simples, appelées *superpoints*, et la création d'une structure de graphe liant les superpoints voisins. Au lieu de classifier chaque point indépendamment, nous classifions les superpoints eux-même, réduisant ainsi la complexité du problème de plusieurs ordres de grandeur. En d'autre terme, nous transformons le problème de segmentation sémantique d'un grand nuage de points en la classification des sommets d'un graphe avec peu de sommets. Cela nous permet d'utiliser des puissants outils de convolution sur graphe pour analyser le contexte de grandes scènes, *c.f.* Figure 7.2. Notre approche permet de classifier avec d'excellents résultats des nuages contenant des dizaines de millions de points avec des réseaux de neurones efficaces en temps d'inférences et en nombre de paramètres.

Segmentation Supervisée. La limitation de l'approche précédente est que toute erreur dans la partition entraîne des mauvaises classifications. Nous proposons donc d'entraîner un réseau de neurones à segmenter un grand nuage de points en superpoints sémantiquement purs. Nous mettons au point une nouvelle fonction de perte contrastive permettant d'apprendre une représentation de points qui soit homogène au sein des objets et présente de forts contrastes à leur interface. La partition en découlant aboutit à une amélioration importante de la qualité de la segmentation sémantique.

Reconstruction Large Échelle. Nous proposons une approche pour la reconstruction grande de surface à grande échelle basée sur une discrétisation de l'espace 3D par une tetraèdrisation de Delaunay. Nous adaptons les classiques modèles d'énergie sous-modulaire, qui peuvent être efficacement résolus par minimisation de flots, mais dont les potentiels sont appris par un réseau de neurones sur graphe. Notre méthode combine donc la richesse des méthodes d'apprentissage et la rapidité et les garanties des méthodes énergétiques. Entraînée sur un faible nombre de scans virtuelles, notre approche généralise à de nouveaux domaines et à de très grandes scènes avec une qualité supérieure aux modèle d'apprentissage et énergétiques.

5 Structure des Capteurs de Données 3D

Dans cette section, nous présentons nos travaux ayant pour but d'exploiter la géométrie particulière des acquisitions de télédétection 3D pour améliorer la précision et la vitesse d'algorithmes de segmentation

(a) liDAR Rotatif (b) Image/LiDAR (c) Visibilité

Figure 7.3: **Structure des Capteurs.**
Nous exploitons la structure d'acquisition des LiDARs rotatifs, la complémentarité images/ LiDAR, et les informations de visibilités qui peuvent être déduites de la pose des capteurs.

sémantique et de reconstruction de surface. Ce chapitre est basé sur les publications suivantes:

[LAL22]: Romain Loiseau, Mathieu Aubry, Loic Landrieu, "Online Segmentation of LiDAR Sequences: Dataset and Algorithm", *ECCV*, 2022
[RVL22]: Damien Robert, Bruno Vallet, Loic Landrieu, "Learning Multi-View Aggregation In the Wild for Large-Scale 3D Semantic Segmentation", *CVPR*, 2022
[SLB+22]: Raphael Sulzer, Loic Landrieu, Alexandre Boulch, Renaud Marlet, Bruno Vallet, "Deep Surface Reconstruction from Point Clouds with Visibility Information", *ICPR*, 2022

LiDAR Rotatif Mobile. Les LiDARs rotatifs sont très utilisés dans le cadre de la conduite autonome, motivant ainsi le besoin d'outils d'analyse temps réel de séquences temporelles de points 3D. La structure de ces séquences est complexe: le capteur tourne rapidement sur lui-même, la plate-forme d'acquisition est en mouvement, et les environnements urbains sont typiquement dynamiques; voir Figure 7.3a pour une illustration.

La plupart des jeux de données et algorithmes proposés pour l'analyse de telles séquences opèrent sur un découpage en trames couvrant 3600°, ce qui conduit à une latence d'acquisition incompatible avec les applications temps réels. Nous répondons à ce problème avec deux contributions principales. Tout d'abord, nous présentons HelixNet, un dataset de 10 milliards de points 3D annotés individuellement avec labels sémantiques et les informations de capteur nécessaires à la mesure précise de la capacité

temps réel des algorithmes de segmentation. Nous proposons également Helix4D, un réseau spatio-temporel compact et efficace spécialement conçu pour l'analyse de séquences de points 3D issues d'un LiDAR rotatif. Helix4D opère sur des tranches d'acquisition couvrant une fraction de tour du capteur sur lui-même, réduisant significativement la latence totale.

Nous évaluons la performance et latence de plusieurs algorithmes de l'état de l'art sur HelixNet et SemanticKITTI. Helix4D atteint une précision comparable aux algorithmes de segmentations les plus récents avec une réduction de plus de $5\times$ en terme de latence, et $50\times$ du nombre de paramètres.

Complémentarité Image/LiDAR. Les images et les nuages de points contiennent des informations différentes et complémentaires: les images capturent la texture et le contexte des objets, alors que les nuages de points reflètent leur géométrie avec précision, *c.f.* Figure 7.3b. Les approches hybrides 2D/3D proposent de combiner ces informations pour améliorer la précision de l'analyse de grandes scènes.

Combiner images et nuages de points issus de grandes scènes présente de nombreuses difficultés, telles que la mise en relation des points et des pixels et l'agrégation des descripteurs issus de différentes images. Les méthodes actuelles nécessitent souvent la construction d'un maillage ou l'usage de capteurs spécialisés pour trouver les occlusions, et utilisent des heuristiques pour combiner les images.

Nous proposons une méthode d'agrégation multi-vues exploitant les conditions d'observations (angle de vue, distance, occlusion, etc.) des points 3D dans les images. Notre approche permet de combiner des réseaux 2D et 3D standards et d'obtenir des résultats supérieurs aux réseaux classiques ou hybrides sans nécessiter de colorisation, de maillage, ni de capteur de profondeur. Notre approche opère sur les nuages de points bruts, et un nombre arbitraire d'images avec pose. Nous définissons un nouvel état de l'art pour un jeu de donnée de segmentation de scènes d'intérieur et un autre jeu de données centré sur la conduite autonome.

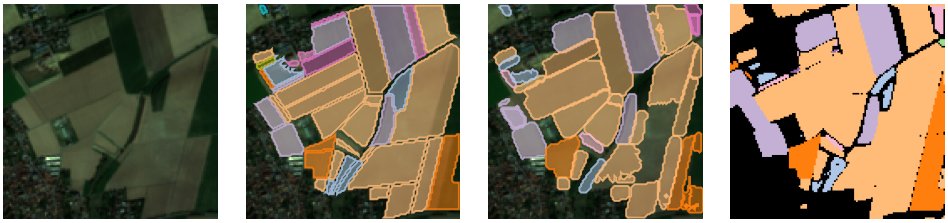
Reconstruction avec Information de Visibilité. Les méthodes actuelles de reconstruction de surface à partir de nuages de points 3D ignorent les informations de pose et n'utilisent que la position des points. Cependant, la pose des capteurs permet de déduire des informations de visibilité (ligne de vue) qui sont précieuses pour la reconstruction et l'orientation des surfaces prédites. Nous proposons deux manières simples d'enrichir chaque point 3D avec ses informations de visibilité: (i) ajout d'un vecteur pointant vers le capteur, (ii) ajouts de points virtuels le long de la ligne de vue, *c.f.* Figure 7.3c. Ces deux modifications peuvent s'intégrer facilement à de nombreuses approches existantes avec modifications minimales.

Notre méthode améliore la précision des surfaces générées mais aussi les capacités de généralisation des réseaux à des nouveaux domaines: nouvelles classes, scènes entières, nouveaux capteurs.

6 Structure des Séquences Temporelles Satellite

Cette section présente nos travaux sur l'analyse automatique de séries temporelles d'images satellites (SITS) comme représenta à la Figure 7.4. De par l'accessibilité de grandes quantités de données annotées, l'analyse automatique de SITS est un exemple de premier plan de l'intérêt des approches d'apprentissage pour l'analyse des données géospatiales. Cependant, les SITS suivent une structure particulière nécessitant des approches adaptées: multimodalité, faible résolution spatiale, spatio-spectro-temporalité, et grande échelle. Nous proposons une série d'algorithmes exploitant cette structure pour améliorer la vitesse et précision de l'analyse des SITS. Cette section est basée sur les articles suivants:
[GLGC20]: Vivien Sainte Fare Garnot, Loic Landrieu, Sebastien Giordano, Nesrine Chehata, "Satellite image time series classification with pixel-set encoders and temporal self-attention", *CVPR*, 2020
[GL21b]: Vivien Sainte Fare Garnot, Loic Landrieu, "Panoptic Segmentation of Satellite Image Time Series with Convolutional Temporal Attention Networks", *ICCV*, 2021
[GL20]: Vivien Sainte Fare Garnot, Loic Landrieu, "Lightweight Temporal Self-Attention for Classifying Satellite Image Time Series", *ECML Workshop on Advanced Analysis and Learning on Temporal Data*, 2020
[GL21a]: Vivien Sainte Fare Garnot, Loic Landrieu, "Leveraging Class Hierarchies with Metric-Guided Prototype Learning", *BMVC*, 2021
[GL22]: Vivien Sainte Fare Garnot, Loic Landrieu, "Multi-Modal Temporal Attention Models for Crop Mapping from Satellite Time Series", *ISPRS Journal*, 2021
[QL21]: Félix Quinton, Loic Landrieu, "Crop Rotation Modeling for Deep Learning-Based Parcel Classification from Satellite Time Series", *Remote Sensing*, 2021
[GBLC20]: Giordano, Sébastien and Bailly, Simon and Landrieu, Loic and Chehata, Nesrine, "Improved crop classification with rotation knowledge using Sentinel-1 and -2 time series", *Photogrammetric Engineering & Remote Sensing*, 2021
[GLGC19]: Vivien Sainte Fare Garnot, Loic Landrieu, Sebastien Giordano, Nesrine Chehata, "Time-space trade-off in deep learning models for crop classification on satellite multi-spectral image time series", *IGARSS*, 2020

Classification de Parcelles. De par leur grande accessibilité, les séries temporelles d'images satellites sont au centre d'un effort d'automatisation



(a) Single image (b) Panoptic GT (c) Panoptic seg (d) Semantic seg

Figure 7.4: **Segmentation Panoptique de SITS.** Notre objectif est de prédire automatiquement les cultures et contours de parcelles agricoles à partir de séries temporelles d'images satellite.

des données d'observation de la terre. En particulier, le suivi des cultures agricoles est un sujet d'importance écologique, économique, et politique majeure. En effet, plus de 57 milliards de subvention par an de subvention sont distribuées chaque année en Europe aux agriculteurs, et la rotation des cultures a un important impact environnemental mais aussi le rendement des cultures.

Les approches les plus couramment utilisées pour cette tâche mêlent des réseaux convolutionnels pour l'aspect spatial et des réseaux récurrents pour la dimension temporelle. Observant que la résolution des images satellites à faible temps de revisite ne permet pas de capturer la texture des cultures, nous proposons de considérer chaque parcelle comme un ensemble non ordonné d'observations radiométriques multi-spectrales. Cette approche permet d'apprendre efficacement des descripteurs de la réponse spectrale des parcelles sans avoir à les redimensionner, ce qui économise à la fois du temps de calcul et de la mémoire.

Nous adaptons également les méthodes d'auto-attention si efficace en traitement de la langue au contexte des séquences d'images satellite. Nous proposons une série de modifications visant à améliorer la performance et de la vitesse de ces méthodes pour notre cas d'usage, aboutissant à une architecture légère et performante qui est maintenant largement utilisée dans le domaine de la classification de séries temporelles. Pour l'évaluation, nous mettons à disposition un dataset de parcelles annotées et leurs séries temporelles correspondantes.

Segmentation Panoptique de Parcelles. Nous proposons une extension de nos travaux précédents pour prédire non seulement les cultures mais aussi la forme des parcelles agricoles. Ce problème peut se formuler comme la segmentation panoptique de séries temporelles d'images: chaque pixel est associé à un label sémantique, et les pixels de cultures sont associés à l'index de leur parcelle.

Ce problème a été étudié pour les images satellite, mais la dimension

temporelle reste ignorée alors qu'elle est critique pour la classification des espèces végétales. Nous présentons la première approche d'apprentissage pour la segmentation panoptique de série temporelle d'images satellites. Un premier module combine des convolutions spatiales et l'approches d'attention temporelle décrite ci-dessus pour extraire de riches descripteurs spatio-temporels multi-échelles. Un second module permet de prédire efficacement et précisément le contour des parcelles agricoles.

Nous introduisons aussi PASTIS, le premier jeu de données libre de séries temporelles d'images satellites avec annotation panoptique, et couvrant plus de 4000 mk^2. Évalué sur cet dataset, nous démontrons la supériorité de notre encodeur spatio-temporel pour la segmentation sémantique, et proposons le premier état de l'art de la segmentation panoptique de séries d'images satellite.

Extension Multi-Années. Les rotations de cultures jouent un rôle important sur l'impact environnemental et le rendement des cultures. Nous proposons un modèle capable d'apprendre les rotations de cultures pour améliorer leur classification. Nous introduisons également le premier jeu de données multi-années de séries temporelles d'images satellite. Notre approche améliore les performances de classification comparés aux modèle mono-années, mais aussi par rapport aux approches existantes pour prendre en compte les rotations.

Multimodalité Optique-Radar. Les capteurs optiques passifs superspectraux et le radar à synthèse d'ouverture permettent d'acquérir depuis l'espace des informations complémentaires sur les cultures. Nous proposons une nouvelle approche pour combiner ces informations pour l'analyse de SITS. Notre méthode repose sur les modules d'attention temporelle décrits ci-dessus, et peut combiner différentes modalités sans nécessiter d'alignement temporel, d'interpolation, ou de pré-traitement.

Nous proposons également une extension du jeu de données PASTIS avec les séries radar associées à chaque série optique. Nous évaluons nos approches et ses variations sur différentes tâches, allant de la classification de parcelles à la segmentation panoptique. Nous définissons un nouvel état de l'art pour chacune des tâches considérées grâce à l'apport de la multimodalité.

Classification Hiérarchique. Les espèces agricoles peuvent être organisées selon une hiérarchie de classes, à l'instar de nombreuses autres nomenclatures. Cette structure induit une distance sémantique entre classes, et définit ainsi une métrique discrète.

Nous proposons une régularisation permettant d'apprendre conjointement des descripteurs et des prototypes de classes qui suivent un arrange-

ment en accord avec la distance sémantique entre leur classes. Formellement, nous minimisons la distorsion enter la métrique discrète entre classes et la métrique continues entre prototypes. Nous évaluons notre approche sur différentes tâches, de la classification de séries temporelles à la segmentation sémantique d'images de profondeur, et observons une baisse systématique du taux d'erreur pondéré par leur coût. De façon plus surprenante, dans certains cas le taux d'erreur non pondéré diminue également, démontrant que l'injection de connaissance sur la hiérarchie des classes permet aussi d'améliorer la qualité des descripteurs appris.

Bibliography

Defendant's References

[CCHL20] Thomas Chaton, Nicolas Chaulet, Sofiane Horache, and **Loic Landrieu**. Torch-Points3D: A modular multi-task framework for reproducible deep learning on 3D point clouds. *3DV*, 2020.

[GBLC20] Sébastien Giordano, Simon Bailly, **Loic Landrieu**, and Nesrine Chehata. Improved crop classification with rotation knowledge using sentinel-1 and-2 time series. *Photogrammetric Engineering & Remote Sensing*, 2020.

[GL17] Stéphane Guinard and **Loic Landrieu**. Weakly supervised segmentation-aided classification of urban scenes from 3D LiDAR point clouds. *ISPRS Workshops 2017*, 2017.

[GL20] Vivien Sainte Fare Garnot and **Loic Landrieu**. Lightweight temporal self-attention for classifying satellite images time series. *ECML Workshop on Advanced and Learning on Temporal Data*, 2020.

[GL21a] Vivien Sainte Fare Garnot and **Loic Landrieu**. Leveraging class hierarchies with metric-guided prototype learning. *BMVC*, 2021.

[GL21b] Vivien Sainte Fare Garnot and **Loic Landrieu**. Panoptic segmentation of satellite image time series with convolutional temporal attention networks satellite image time series with convolutional temporal attention networks. *ICCV*, 2021.

[GL22] Vivien Sainte Fare Garnot and **Loic Landrieu**. Multi-modal temporal at-tention models for crop mapping from satellite time series. *ISPRS Journal of Photogrammetry and Remote Sensing*, 2022.

[GLCV19] Stéphane Guinard, **Loic Landrieu**, Laurent Caraffa, and Bruno Vallet. Piecewise-planar approximation of large 3D data as graph-structured optimization. *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, 2019.

[GLGC19] Vivien Sainte Fare Garnot, **Loic Landrieu**, Sebastien Giordano, and Nesrine Chehata. Time-space trade-off in deep learning models for crop classification on satellite multi-spectral image time series. *IGARSS*, 2019.

[GLGC20] Vivien Sainte Fare Garnot, **Loic Landrieu**, Sebastien Giordano, and Nesrine Chehata. Satellite image time series classification with pixel-set encoders and temporal self-attention. *CVPR*, 2020.

[KLMC22a] Ekaterina Kalinicheva, **Loic Landrieu**, Clément Mallet, and Nesrine Chehata. Multi-layer modeling of dense vegetation from aerial LiDAR scans. *CVPR Workshops*, 2022.

[KLMC22b] Ekaterina Kalinicheva, **Loic Landrieu**, Clément Mallet, and Nesrine Chehata. Predicting vegetation stratum occupancy from airborne LiDAR data with deep learning. *Journal of Applied Earth Observation and Geoinformation*, 2022.

[LAL22] Romain Loiseau, Mathieu Aubry, and **Loic Landrieu**. Online segmentation of LiDAR sequences: Dataset and algorithm. *ECCV*, 2022.

[LB19] **Loic Landrieu** and Mohamed Boussaha. Point cloud oversegmentation with graph-structured deep metric learning. *CVPR*, 2019.

[LBT+22] Romain Loiseau, Baptiste Bouvier, Yann Teytaut, Elliot Vincent, Mathieu Aubry, and **Loic Landrieu**. A model you can hear: Audio identification with playable prototypes. *arXiv preprint arXiv:2208.03311*, 2022.

[LMAL21] Romain Loiseau, Tom Monnier, Mathieu Aubry, and **Loic Landrieu**. Representing shape collections with alignment-aware linear models. In *3DV*, 2021.

[LO14] Loic Landrieu and Guillaume Obozinski. Continuously indexed potts models on unoriented graphs. *UAI*, 2014.

[LO16] **Loic Landrieu** and Guillaume Obozinski. Cut pursuit: Fast algorithms to learn piecewise constant functions. *AISTATS*, 2016.

[LO17] **Loic Landrieu** and Guillaume Obozinski. Cut pursuit: Fast algorithms to learn piecewise constant functions on general weighted graphs. *SIAM Journal on Imaging Sciences*, 2017.

[Loi16] Landrieu Loic. *Learning structured models on weighted graphs, with applications to spatial data analysis*. PhD thesis, Paris Sciences et Lettres (ComUE), 2016.

[LRE+22] **Loic Landrieu** , Ewelina Rupnik, S Oude Elberink, Clément Mallet, and Nicolas Paparoditis. Preface: the 2022 edition of the xxivth isprs congress. In *XXIVth ISPRS Congress*, 2022.

[LRV+17] **Loic Landrieu** , Hugo Raguet, Bruno Vallet, Clément Mallet, and Martin Weinmann. A structured regularization framework for spatially smoothing semantic labelings of 3D point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, 2017.

[LS18] **Loic Landrieu** and Martin Simonovsky. Large-scale point cloud semantic segmentation with superpoint graphs. *CVPR*, 2018.

[LVAL23] Romain Loiseau, Elliot Vincent, Mathieu Aubry, and **Loic Landrieu**. Learnable earth parser: Discovering 3d prototypes in aerial scans. *arXiv preprint arXiv:2304.09704*, 2023.

[QL21] Félix Quinton and **Loic Landrieu**. Crop rotation modeling for deep learning-based parcel classification from satellite time series. *Remote Sensing*, 2021.

[RL15] Hugo Raguet and **Loic Landrieu**. Preconditioning of a generalized forward-backward splitting and application to optimization on graphs. *SIAM Journal on Imaging Sciences*, 2015.

[RL18] Hugo Raguet and **Loic Landrieu**. Cut-pursuit algorithm for regularizing nonsmooth functionals with graph total variation. *ICML*, 2018.

[RL19] Hugo Raguet and **Loic Landrieu**. Parallel cut pursuit for minimization of the graph total variation. *CVPR Workshop on Graph Reasoning*, 2019.

[RVL22] Damien Robert, Bruno Vallet, and **Loic Landrieu**. Learning multi-view aggregation in the wild for large-scale 3D semantic segmentation. *CVPR*, 2022.

[SLB+22] Raphael Sulzer, **Loic Landrieu**, Alexandre Boulch, Renaud Marlet, and Bruno Vallet. Deep surface reconstruction from point clouds with visibility information. *ICPR*, 2022.

[SLMV21] Raphael Sulzer, **Loic Landrieu**, Renaud Marlet, and Bruno Vallet. Scalable surface reconstruction with delaunay-graph neural networks. *Symposium on Geometry Processing*, 2021.

Other References

[AAM+11] I Aziz, M Ashraf, T Mahmood, KR Islam, et al. Crop rotation impact on soil quality. *Pakistan Journal of Botany*, 2011.

[AHK+21] Elias Ayrey, Daniel J. Hayes, John B. Kilbride, Shawn Fraver, John A. Kershaw, Bruce D. Cook, and Aaron R. Weiskittel. Synthesizing disparate LiDAR and satellite datasets through deep learning to generate wall-to-wall regional inventories for the complex, mixed-species forests of the Eastern United States. *Remote Sensing*, 2021.

[AJG96] Chester L Arnold Jr and C James Gibbons. Impervious surface coverage: the emergence of a key environmental indicator. *Journal of the American planning Association*, 1996.

[ASM+22] Sheikh Kamran Abid, Noralfishah Sulaiman, Nur Putri Najwa Mahmud, Umber Nazir, and Nur Azhani Adnan. A review on the application of remote sensing and geographic information system in flood crisis management. *Conference on Broad Exposure to Science and Technology*, 2022.

[ASS+12] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. SLIC superpixels compared to state-of-the-art superpixel methods. *Transactions on Pattern Analysis and Machine Intelligence*, 2012.

[ASZ+16] Iro Armeni, Ozan Sener, Amir R Zamir, Helen Jiang, Ioannis Brilakis, Martin Fischer, and Silvio Savarese. 3D semantic parsing of large-scale indoor spaces. *CVPR*, 2016.

[AUM+21] Kumar Ayush, Burak Uzkent, Chenlin Meng, Kumar Tanmay, Marshall Burke, David Lobell, and Stefano Ermon. Geography-aware self-supervised learning. *ICCV*, 2021.

[Bac11] Francis Bach. Shaping level sets with submodular functions. *NeurIPS*, 2011.

[BAC+14] Hanna Becker, Laurent Albera, Pierre Comon, Rémi Gribonval, and Isabelle Merlet. Fast, variation-based methods for

the analysis of extended brain sources. *European Signal Processing Conference*, 2014.

[Bat18] Michael Batty. Digital twins. *Environment and Planning B: Urban Analytics and City Science*, 2018.

[BD05] DS Boyd and FM Danson. Satellite remote sensing of forest resources: three decades of research development. *Progress in Physical Geography*, 2005.

[BdLGM14] Alexandre Boulch, Martin de La Gorce, and Renaud Marlet. Piecewise-planar 3D reconstruction with edge and corner regularization. *Computer Graphic Forum*, 2014.

[BF15] Gerhard Brankatschk and Matthias Finkbeiner. Modeling crop rotation in agricultural lcas—challenges and potential solutions. *Agricultural Systems*, 2015.

[BGLSA18] Alexandre Boulch, Joris Guerry, Bertrand Le Saux, and Nicolas Audebert. Snapnet: 3D point cloud semantic labeling with 2D deep segmentation networks. *Computers & Graphics*, 2018.

[BGM+19] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall. SemanticKITTI: A dataset for semantic scene understanding of LiDAR sequences. *ICCV*, 2019.

[BGT15] Piero Boccardo and Fabio Giulio Tonolo. Remote sensing role in emergency mapping for disaster response. *Engineering Geology for Society and Territory-Volume 5*, 2015.

[BHA+21] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

[BJL+02] Ian J Bateman, Andrew P Jones, Andrew A Lovett, IR Lake, and BH Day. Applying geographical information systems (GIS) to environmental and resource economics. *Environmental and Resource Economics*, 2002.

[BJMO11a] Francis Bach, Rodolphe Jenatton, Julien Mairal, and Guillaume Obozinski. Convex optimization with sparsity-inducing norms. *Optimization for Machine Learning*, 2011.

[BJMO11b] Francis Bach, Rodolphe Jenatton, Julien Mairal, and Guillaume Obozinski. Optimization with sparsity-inducing penalties. *Foundations and Trends in Machine Learning*, 2011.

[BJMO12] Francis Bach, Rodolphe Jenatton, Julien Mairal, and Guillaume Obozinski. Structured sparsity through convex optimization. *Statistical Science*, 2012.

[BK04] Yuri Boykov and Vladimir Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *Transactions on Pattern Analysis and Machine Intelligence*, 2004.

[BLBK17] Benjamin Brede, Alvaro Lau, Harm M. Bartholomeus, and Lammert Kooistra. Comparing RIEGL RiCOPTER UAV LiDAR derived canopy height and DBH with terrestrial LiDAR. *Sensors*, 2017.

[BLN+13] Matthew Berger, Joshua A. Levine, Luis Gustavo Nonato, Gabriel Taubin, and Claudio T. Silva. A benchmark for surface reconstruction. *Transaction on Graphics*, 2013.

[BM22] Alexandre Boulch and Renaud Marlet. POCO: Point convolution for surface reconstruction. *CVPR*, 2022.

[BMKS14] Christopher Boshuizen, James Mason, Pete Klupar, and Shannon Spanhake. Results from the planet labs flock constellation. *AIAA/USU Conference on Small Satellites*, 2014.

[BMT+15] Adeline Bailly, Simon Malinowski, Romain Tavenard, Laetitia Chapel, and Thomas Guyet. Dense bag-of-temporal-sift-words for time series classification. *ECML Workshop on Advanced Analytics and Learning on Temporal Data*, 2015.

[BMT+20] Luca Bertinetto, Romain Mueller, Konstantinos Tertikas, Sina Samangooei, and Nicholas A Lord. Making better mistakes: Leveraging class hierarchies with deep networks. *CVPR*, 2020.

[Bou19] Alexandre Boulch. Generalizing discrete convolutions for unstructured point clouds. *3DOR Eurographics*, 2019.

[BRV16] A. Bódis-Szomorú, H. Riemenschneider, and L. Van Gool. Efficient volumetric fusion of airborne and street-side data for urban reconstruction. *ICPR*, 2016.

[BS14] Alvaro Barbero and Suvrit Sra. Modular proximal optimization for multidimensional total-variation regularization. *The Journal of Machine Learning Research*, 2014.

[Bul92] Donald G Bullock. Crop rotation. *Critical Reviews in Plant Sciences*, 1992.

[BVZ01] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *Transactions on Pattern Analysis and Machine Intelligence*, 2001.

[BYPC16] Nicolas Ballas, Li Yao, Chris Pal, and Aaron Courville. Delving deeper into convolutional networks for learning video representations. *ICLR*, 2016.

[CBFAB97] Pierre Charbonnier, Laure Blanc-Féraud, Gilles Aubert, and Michel Barlaud. Deterministic edge-preserving regularization in computed imaging. *Transactions on Image Processing*, 1997.

[CBV17] Laurent Caraffa, Mathieu Brédif, and Bruno Vallet. 3D watertight mesh generation with uncertainties from ubiquitous data. *ACCV*, 2017.

[CC89] Emilio Chuvieco and Russell G Congalton. Application of remote sensing and geographic information systems to forest fire hazard mapping. *Remote Sensing of Environment*, 1989.

[CD09] Antonin Chambolle and Jérôme Darbon. On total variation minimization and surface evolution using parametric maximum flows. *International Journal of Computer Vision*, 2009.

[CFG+15] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: an information-rich 3D model repository. Technical report, Stanford University, Princeton University, Toyota Technological Institute at Chicago, 2015.

[CG+04] Frédéric Cazals, Joachim Giesen, et al. *Delaunay triangulation based surface reconstruction: a short survey*. PhD thesis, INRIA, 2004.

[CGN+13] Camille Couprie, Leo Grady, Laurent Najman, Jean-Christophe Pesquet, and Hugues Talbot. Dual constrained TV-based regularization on graphs. *SIAM Journal on Imaging Sciences*, 2013.

[CGS19] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4D spatio-temporal convnets: Minkowski convolutional neural networks. *CVPR*, 2019.

[Cha08] Kang-Tsung Chang. *Introduction to geographic information systems*. McGraw-Hill Boston, 2008.

[CHL05] Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. *CVPR*, 2005.

[CHLS17] Ziyun Cai, Jungong Han, Li Liu, and Ling Shao. RGB-D datasets using Microsoft Kinect or similar sensors: a survey. *Multimedia Tools and Applications*, 2017.

[CK16] Siddhartha Chandra and Iasonas Kokkinos. Fast, exact and multi-scale inference for semantic image segmentation with deep gaussian crfs. *ECCV*, 2016.

[CKR04] Simon Clode, Peter J Kootsookos, and Franz Rottensteiner. The automatic extraction of roads from LiDAR data. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2004.

[CLLH19] Hung-Yueh Chiang, Yen-Liang Lin, Yueh-Cheng Liu, and Winston H Hsu. A unified point-based framework for 3D segmentation. *3DV*, 2019.

[CLT+19] Chaofan Chen, Oscar Li, Daniel Tao, Alina Barnett, Cynthia Rudin, and Jonathan K Su. This looks like that: Deep learning for interpretable image recognition. *NeurIPS*, 2019.

[CP08] Patrick L Combettes and Jean-Christophe Pesquet. A proximal decomposition method for solving convex variational inverse problems. *Inverse Problems*, 2008.

[CRC05] Matthew L Clark, Dar A Roberts, and David B Clark. Hyperspectral discrimination of tropical rain forest tree species at leaf to crown scales. *Remote Sensing of Environment*, 2005.

[CRPW12] Venkat Chandrasekaran, Benjamin Recht, Pablo A Parrilo, and Alan S Willsky. The convex geometry of linear inverse problems. *Foundations of Computational Mathematics*, 2012.

[CS95] Roger N Clark and Gregg A Swayze. Mapping minerals, amorphous materials, environmental materials, vegetation, water, ice and snow, and other materials: the USGS Tricorder algorithm. *JPL, Summaries of the Fifth Annual JPL Airborne Earth Science Workshop*, 1995.

[CSAD04] David Cohen-Steiner, Pierre Alliez, and Mathieu Desbrun. Variational shape approximation. *Transactions on Graphics*, 2004.

[CTA20] Tiago Cortinhal, George Tzelepis, and Eren Erdal Aksoy. SalsaNext: Fast, uncertainty-aware semantic segmentation of LiDAR point clouds for autonomous driving. *arXiv:2003.03653*, 2020.

[CTGHM+19] Manuel Campos-Taberner, Francisco Javier García-Haro, Beatriz Martínez, Sergio Sánchez-Ruíz, and María Amparo Gilabert. A Copernicus Sentinel-1 and Sentinel-2 classification framework for the 2020+ European common agricultural policy: A case study in València (Spain). *Agronomy*, 2019.

[CvMG+14] Kyunghyun Cho, Bart van Merriënboer, Çağlar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. *EMNLP*, 2014.

[CWB08] Emmanuel J Candes, Michael B Wakin, and Stephen P Boyd. Enhancing sparsity by reweighted ℓ_1 minimization. *Journal of Fourier Analysis and Applications*, 2008.

[DBC+11] Fabio Dell'Acqua, Christian Bignami, Marco Chini, Gianni Lisini, Diego Aldo Polli, and Salvatore Stramondo. Earthquake damages rapid mapping by satellite remote sensing data: L'aquila april 6th, 2009 event. *Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2011.

[DBG+20] Ralph Dubayah, James Bryan Blair, Scott Goetz, Lola Fatoyinbo, Matthew Hansen, Sean Healey, Michelle Hofton, George Hurtt, James Kellner, Scott Luthcke, et al. The global ecosystem dynamics investigation: High-resolution laser ranging of the earth's forests and topography. *Science of Remote Sensing*, 2020.

[DBLFF10] Jia Deng, Alexander C Berg, Kai Li, and Li Fei-Fei. What does classifying more than 10,000 image categories tell us? *ECCV*, 2010.

[DCLT18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. *NAACL*, 2018.

[DDBC+12] Matthias Drusch, Umberto Del Bello, Sébastien Carlier, Olivier Colin, Veronica Fernandez, Ferran Gascon, Bianca

Hoersch, Claudia Isola, Paolo Laberinti, Philippe Martimort, et al. Sentinel-2: ESA's optical high-resolution mission for GMES operational services. *Remote Sensing of Environment*, 2012.

[DDN20] Angela Dai, Christian Diller, and Matthias Nießner. SG-NN: Sparse generative neural networks for self-supervised scene completion of RGB-D scans. *CVPR*, 2020.

[DG12] Fabio Dell'Acqua and Paolo Gamba. Remote sensing and earthquake damage assessment: Experiences, limits, and perspectives. *Proceedings of the IEEE*, 2012.

[DGF+19] Theo Deprelle, Thibault Groueix, Matthew Fisher, Vladimir Kim, Bryan Russell, and Mathieu Aubry. Learning elementary structures for 3D shape generation and matching. *NeurIPS*, 2019.

[DMDV11] Jérôme Demantké, Clément Mallet, Nicolas David, and Bruno Vallet. Dimensionality based scale selection in 3D LiDAR point clouds. *Laserscanning*, 2011.

[DN18] Angela Dai and Matthias Nießner. 3DMV: Joint 3D-multi-view prediction for 3D semantic scene segmentation. *ECCV*, 2018.

[DPS15] Charles-Alban Deledalle, Nicolas Papadakis, and Joseph Salmon. On debiasing restoration algorithms: Applications to total-variation and nonlocal-means. *International Conference on Scale Space and Variational Methods in Computer Vision*, 2015.

[DRVI03] S Dogliotti, WAH Rossing, and MK Van Ittersum. Rotat, a tool for systematically generating crop rotations. *European Journal of Agronomy*, 2003.

[DSG+12] Jérôme Dury, Noémie Schaller, Frédérick Garcia, Arnaud Reynaud, and Jacques Eric Bergez. Models to support cropping plan and crop rotation decisions. a review. *Agronomy for Sustainable Development*, 2012.

[DSGRS18] Christopher De Sa, Albert Gu, Christopher Ré, and Frederic Sala. Representation tradeoffs for hyperbolic embeddings. *Proceedings of Machine Learning Research*, 2018.

[EBD21] Nico Engel, Vasileios Belagiannis, and Klaus Dietmayer. Point transformer. *ICCV*, 2021.

[EFP+13] Damian H Evans, Roland J Fletcher, Christophe Pottier, Jean-Baptiste Chevance, Dominique Soutif, Boun Suy Tan, Sokrithy Im, Darith Ea, Tina Tin, Samnang Kim, et al. Uncovering archaeological landscapes at Angkor using LiDAR. *Proceedings of the National Academy of Sciences*, 2013.

[EHJT04] Bradley Efron, Trevor Hastie, Iain Johnstone, and Robert Tibshirani. Least angle regression. *The Annals of Statistics*, 2004.

[EKHL17] Francis Engelmann, Theodora Kontogianni, Alexander Hermans, and Bastian Leibe. Exploring spatial context for 3D semantic segmentation of point clouds. *ICCV Workshops*, 2017.

[EOMW20] Philipp Erler, Stefan Ohrhallinger, Niloy Mitra, and Michael Wimmer. Points2Surf: Learning implicit surfaces from point clouds. *ECCV*, 2020.

[ESR+16] Isaque Daniel Rocha Eberhardt, Bruno Schultz, Rodrigo Rizzi, Ieda Del'Arco Sanches, Antonio Roberto Formaggio, Clement Atzberger, Marcio Pupin Mello, Markus Immitzer, Kleber Trabaquini, William Foschiera, et al. Cloud cover assessment for operational crop monitoring systems in tropical areas. *Remote Sensing*, 2016.

[EZE07] Noha Youssry El-Zehiry and Adel Elmaghraby. Brain MRI tissue classification using graph cut optimization of the Mumford-Shah functional. *Proceedings of the International Vision Conference of New Zealand*, 2007.

[FB81] Martin A Fischler and Robert C Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 1981.

[FBJ+12] António Ferraz, Frédéric Bretar, Stéphane Jacquemoud, Gil Gonçalves, Luisa Pereira, Margarida Tomé, and Paula Soares. 3D mapping of a multi-layered Mediterranean forest using ALS data. *Remote Sensing of Environment*, 2012.

[FF56] Lester Randolph Ford and Delbert R Fulkerson. Maximal flow through a network. *Canadian journal of Mathematics*, 1956.

[FLE+22] Christopher Fisher, Stephen Leisz, Damian Evans, Diana H Wall, Kathleen Galvin, Melinda Laituri, Geoffrey Henebry,

James Zeidler, Juan Carlos Fernandez-Diaz, Shrideep Pallickara, et al. Creating an earth archive. *Proceedings of the National Academy of Sciences*, 2022.

[FMJ+15] António Ferraz, Clément Mallet, Stéphane Jacquemoud, Gil Gonçalves, Margarida Tomé, Paula Soares, Luísa Pereira, and Frédéric Bretar. Canopy density model: A new als-derived product to generate multilayer crown cover maps. *Transactions on Geoscience and Remote Sensing*, 2015.

[For08] Eric Kwabena Forkuo. *Digital terrain modeling in a GIS environment*. Citeseer, 2008.

[FSM+16] António Ferraz, Sassan Saatchi, Clément Mallet, Stéphane Jacquemoud, Gil Gonçalves, Carlos Alberto Silva, Paula Soares, Margarida Tomé, and Luisa Pereira. Airborne LiDAR estimation of aboveground forest biomass in the absence of field inventory. *Remote Sensing*, 2016.

[GAMP12] Etienne Gaujour, Bernard Amiaud, Catherine Mignolet, and Sylvain Plantureux. Factors and processes affecting plant biodiversity in permanent grasslands. a review. *Agronomy for Sustainable Development*, 2012.

[GBB+20] Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.

[GCB07] Megandhren Govender, Kershani Chetty, and Hartley Bulcock. A review of hyperspectral remote sensing and its application in vegetation and water resource studies. *Water SA*, 2007.

[GCVY21] Matthew G. Gale, Geoffrey J. Cary, Albert I.J.M. Van Dijk, and Marta Yebra. Forest fire fuel through the lens of remote sensing: Review of approaches, challenges and future directions in the remote sensing of biotic determinants of fire behaviour. *Remote Sensing of Environment*, 2021.

[GFK+18] Thibault Groueix, Matthew Fisher, Vladimir G. Kim, Bryan C. Russell, and Mathieu Aubry. AtlasNet: A papier-mâché approach to learning 3D surface generation. *CVPR*, 2018.

[GKM+20] Jakob Geyer, Yohannes Kassahun, Mentar Mahmudi, Xavier Ricou, Rupesh Durgesh, Andrew S Chung, Lorenz Hauswald, Viet Hoang Pham, Maximilian Mühlegg, Sebastian Dorn, et al. A2D2: Audi autonomous driving dataset. *arXiv preprint arXiv:2004.06320*, 2020.

[GLSU13] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The KITTI dataset. *The International Journal of Robotics Research*, 2013.

[Gon08] Georges Gonthier. Formal proof–the four-color theorem. *Notices of the AMS*, 2008.

[Goo10] Michael Goodchild. Twenty years of progress: GIScience in 2010. *Journal of Spatial Information Science*, 2010.

[GPGMLS17] Angel Garcia-Pedrero, Consuelo Gonzalo-Martin, and M Lillo-Saavedra. A machine learning approach for agricultural parcel delineation through agglomerative segmentation. *International Journal of Remote Sensing*, 2017.

[GR92] Donald Geman and George Reynolds. Constrained restoration and the recovery of discontinuities. *Transactions on Pattern Analysis and Machine Intelligence*, 1992.

[Gra97] Wyn Grant. *The common agricultural policy*. Macmillan International Higher Education, 1997.

[GRM+19] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. *ICLR*, 2019.

[GWCV16] Adrien Gaidon, Qiao Wang, Yohann Cabon, and Eleonora Vig. Virtual worlds as proxy for multi-object tracking analysis. *CVPR*, 2016.

[GWH+20] Yulan Guo, Hanyun Wang, Qingyong Hu, Hao Liu, Li Liu, and Mohammed Bennamoun. Deep learning for 3D point clouds: A survey. *Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[GYH+20] Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. *Machine Learning and Systems*, 2020.

[GYL+20] Jianhao Gao, Qiangqiang Yuan, Jie Li, Hai Zhang, and Xin Su. Cloud removal with fusion of high resolution optical and sar images using generative adversarial networks. *Remote Sensing*, 2020.

[H+73] Peter J Huber et al. Robust regression: Asymptotics, conjectures and monte carlo. *The Annals of Statistics*, 1973.

[HCZ17] Hamid Hamraz, Marco Contreras, and Jun Zhang. Forest understory trees can be segmented accurately within sufficiently dense airborne laser scanning point clouds. *Scientific Reports*, 2017.

[HDW+06] Peter Hyde, Ralph Dubayah, Wayne Walker, J Bryan Blair, Michelle Hofton, and Carolyn Hunsaker. Mapping forest structure for wildlife habitat analysis using multi-sensor (LiDAR, SAR/InSAR, ETM+, Quickbird) synergy. *Remote Sensing of Environment*, 2006.

[HGDG17] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. *ICCV*, 2017.

[HL03] John E Hasse and Richard G Lathrop. Land resource impact indicators of urban sprawl. *Applied Geography*, 2003.

[HMDC16] Caner Hazirbas, Lingni Ma, Csaba Domokos, and Daniel Cremers. FuseNet: Incorporating depth into semantic segmentation via fusion-based CNN architecture. *ACCV*, 2016.

[HMGCO20] Rana Hanocka, Gal Metzer, Raja Giryes, and Daniel Cohen-Or. Point2Mesh: A self-prior for deformable meshes. *Transaction on Graphics*, 2020.

[HSL+17] Timo Hackel, Nikolay Savinov, Lubor Ladicky, Jan D Wegner, Konrad Schindler, and Marc Pollefeys. Semantic3D.net: A new large-scale point cloud classification benchmark. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2017.

[HUL04] Jean-Baptiste Hiriart-Urruty and Claude Lemaréchal. *Fundamentals of convex analysis*. Springer Science & Business Media, 2004.

[HvOW03] Werner Härdtle, Goddert von Oheimb, and Christina Westphal. The effects of light and soil conditions on the species richness of the ground vegetation of deciduous forests in northern germany (schleswig-holstein). *Forest Ecology and Management*, 2003.

[HWB+13] Armin Hornung, Kai M Wurm, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard. OctoMap: An efficient probabilistic 3D mapping framework based on octrees. *Autonomous robots*, 2013.

[HYC+21] Le Hui, Jia Yuan, Mingmei Cheng, Jin Xie, Xiaoya Zhang, and Jian Yang. Superpoint network for point cloud oversegmentation. *ICCV*, 2021.

[HYL17] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *NeurIPS*, 2017.

[HYX+20a] Qingyong Hu, Bo Yang, Linhai Xie, Stefano Rosa, Yulan Guo, Zhihua Wang, Niki Trigoni, and Andrew Markham. Randla-net: Efficient semantic segmentation of large-scale point clouds. *CVPR*, 2020.

[HYX+20b] Qingyong Hu, Bo Yang, Linhai Xie, Stefano Rosa, Yulan Guo, Zhihua Wang, Niki Trigoni, and Andrew Markham. RandLA-Net: efficient semantic segmentation of large-scale point clouds. *CVPR*, 2020.

[HZJ+21] Wenbo Hu, Hengshuang Zhao, Li Jiang, Jiaya Jia, and Tien-Tsin Wong. Bidirectional projection network for cross dimension scene understanding. *CVPR*, 2021.

[HZRS16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CVPR*, 2016.

[IIGM19] Dino Ienco, Roberto Interdonato, Raffaele Gaetano, and Dinh Ho Tong Minh. Combining Sentinel-1 and Sentinel-2 satellite image time series for land cover mapping via a multi-source deep learning architecture. *ISPRS Journal of Photogrammetry and Remote Sensing*, 2019.

[IK98] John L Innes and Barbara Koch. Forest biodiversity and its assessment by remote sensing. *Global Ecology & Biogeography Letters*, 1998.

[IS15] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *ICML*, 2015.

[Jag13] Martin Jaggi. Revisiting frank-wolfe: Projection-free sparse convex optimization. *International Conference on Machine Learning*, 2013.

[Jal16] Khalid Jalalzai. Some remarks on the staircasing phenomenon in total variation-based image denoising. *Journal of Mathematical Imaging and Vision*, 2016.

[JGB+21] Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. Perceiver: general perception with iterative attention. *ICLR*, 2021.

[JGS19] Maximilian Jaritz, Jiayuan Gu, and Hao Su. Multi-view pointNet for 3D scene understanding. *CVPR Workshops*, 2019.

[JOWS21] Peng Jiang, Philip Osteen, Maggie Wigness, and Srikanth Saripalli. Rellis-3D dataset: data, benchmarks and analysis. *ICRA*, 2021.

[JP11] Michal Jancosek and Tomas Pajdla. Multi-view reconstruction preserving weakly-supported surfaces. *CVPR*, 2011.

[JP14] Michal Jancosek and Tomas Pajdla. Exploiting visibility information in surface reconstruction to preserve weakly supported surfaces. *International Scholarly Research Notices*, 2014.

[JR08] Junchang Ju and David P Roy. The availability of cloud-free landsat ETM+ data over the conterminous United States and globally. *Remote Sensing of Environment*, 2008.

[JSL+18] Varun Jampani, Deqing Sun, Ming-Yu Liu, Ming-Hsuan Yang, and Jan Kautz. Superpixel sampling networks. *ECCV*, 2018.

[JSM01] Juha Jantunen, Kimmo Saarinen, and Olli Marttila. Effects of forest management on field layer vegetation: a comparison between finnish and russian karelia. *Scandinavian Journal of Forest Research*, 2001.

[JSM+20] Chiyu Max Jiang, Avneesh Sud, Ameesh Makadia, Jingwei Huang, Matthias Nießner, and Thomas Funkhouser. Local implicit grid representations for 3D scenes. *CVPR*, 2020.

[JT92] Jerzy W Jaromczyk and Godfried T Toussaint. Relative neighborhood graphs and their relatives. *Proceedings of the IEEE*, 1992.

[KBJ+15] KS Kumar, Alvaro Barbero, Stefanie Jegelka, Suvrit Sra, and Francis Bach. Convex optimization for parallel energy minimization. *arXiv preprint arXiv:1503.01563*, 2015.

[KDB+19] B Koetz, P Defourny, S Bontemps, K Bajec, C Cara, L de Vendictis, L Kucera, P Malcorps, G Milcinski, L Nicola, et al. Sen4cap sentinels for cap monitoring approach. *JRC IACS Workshop*, 2019.

[KFC+19] Nadezhda Kurganova, Michael Filin, Dmitry Cherniaev, Artem Shaklein, and Dmitry Namiot. Digital twins introduction as one of the major directions of industrial digitalization. *International Journal of Open Information Technologies*, 2019.

[KGHD19] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. *CVPR*, 2019.

[KH+09] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

[KH13] Michael Kazhdan and Hugues Hoppe. Screened Poisson surface reconstruction. *Transaction on Graphics*, 2013.

[KK11] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected CRFs with Gaussian edge potentials. *NeurIPS*, 2011.

[KKN+15] Chris Kollas, Kurt Christian Kersebaum, Claas Nendel, Kiril Manevski, Christoph Müller, Taru Palosuo, Cecilia M Armas-Herrera, Nicolas Beaudoin, Marco Bindi, Monia Charfeddine, et al. Crop rotation modelling—a European model intercomparison. *European Journal of Agronomy*, 2015.

[KL51] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 1951.

[KL18] Wouter M Kouw and Marco Loog. An introduction to domain adaptation and transfer learning. *arXiv preprint arXiv:1812.11806*, 2018.

[KLG+16] Nataliia Kussul, Guido Lemoine, Francisco Javier Gallego, Sergii V Skakun, Mykola Lavreniuk, and Andrii Yu Shelestov. Parcel-based crop classification in ukraine using Landsat-8 data and Sentinel-1A data. *Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2016.

[KLSS17] Nataliia Kussul, Mykola Lavreniuk, Sergii Skakun, and Andrii Shelestov. Deep learning classification of land cover and crop types using remote sensing data. *Geoscience and Remote Sensing Letters*, 2017.

[KMMB20] Shunichi Koshimura, Luis Moya, Erick Mas, and Yanbing Bai. Tsunami damage detection with remote sensing: a review. *Geosciences*, 2020.

[KNL+20] Bernd Ketzler, Vasilis Naserentin, Fabio Latino, Christopher Zangelidis, Liane Thuvander, and Anders Logg. Digital twins for cities: A state of the art review. *Built Environment*, 2020.

[KTR+21] Lukas Kondmann, Aysim Toker, Marc Rußwurm, Andrés Camero, Devis Peressuti, Grega Milcinski, Pierre-Philippe Mathieu, Nicolas Longépé, Timothy Davis, Giovanni Marchisio, et al. DENETHOR: The DynamicEarthNET dataset for Harmonized, inter-operable, analysis-Ready, daily crop monitoring from space. *NeurIPS Datasets and Benchmarks Track*, 2021.

[KYF+20] Abhijit Kundu, Xiaoqi Yin, Alireza Fathi, David Ross, Brian Brewington, Thomas Funkhouser, and Caroline Pantofaru. Virtual multi-view fusion for 3D semantic segmentation. *ECCV*, 2020.

[Lec89] Yvan G Leclerc. Constructing simple stable descriptions for image partitioning. *International Journal of Computer Vision*, 1989.

[LKA+22] Nico Lang, Nikolai Kalischek, John Armston, Konrad Schindler, Ralph Dubayah, and Jan Dirk Wegner. Global canopy height regression and uncertainty estimation from GEDI LIDAR waveforms with deep ensembles. *Remote Sensing of Environment*, 2022.

[LKZ+17] Måns Larsson, Fredrik Kahl, Shuai Zheng, Anurag Arnab, Philip H. S. Torr, and Richard I. Hartley. Learning arbitrary potentials in CRFs with gradient descent. *CoRR*, abs/1701.06805, 2017.

[LL19] gregory Lepère and Mathias Lemmens. Laser scanning of damaged historical icons. *GIM International*, 2019.

[LP15] Dirk A Lorenz and Thomas Pock. An inertial forward-backward algorithm for monotone inclusions. *Journal of Mathematical Imaging and Vision*, 2015.

[LPK09] P. Labatut, J. P. Pons, and R. Keriven. Robust and efficient surface reconstruction from range data. *Computer Graphics Forum*, 2009.

[LR12] Yan Lu and Christopher Rasmussen. Simplified Markov random fields for efficient semantic labeling of 3D point clouds. *International Conference on Intelligent Robots and Systems*, 2012.

[LS19] A Lubin and A Saleem. Remote sensing-based mapping of the destruction to aleppo during the syrian civil war between 2011 and 2017. *Applied Geography*, 2019.

[LSL+17] Zhiwei Li, Huanfeng Shen, Huifang Li, Guisong Xia, Paolo Gamba, and Liangpei Zhang. Multi-feature combined cloud and cloud shadow detection in GaoFen- wide field of view imagery. *Remote Sensing of Environment*, 2017.

[LSvdHR16] Guosheng Lin, Chunhua Shen, Anton van den Hengel, and Ian D. Reid. Efficient piecewise training of deep structured models for semantic segmentation. *CVPR*, 2016.

[Lu06] Dengsheng Lu. The potential and challenge of remote sensing-based biomass estimation. *International Journal of Remote Sensing*, 2006.

[LWG+19] Lei Luo, Xinyuan Wang, Huadong Guo, Rosa Lasaponara, Xin Zong, Nicola Masini, Guizhou Wang, Pilong Shi, Houcine Khatteli, Fulong Chen, et al. Airborne and spaceborne remote sensing for archaeological and cultural heritage applications: A review of the century (1907–2017). *Remote Sensing of Environment*, 2019.

[LWZ+18] Yangbin Lin, Cheng Wang, Dawei Zhai, Wei Li, and Jonathan Li. Toward better boundary preserved supervoxel segmentation for 3D point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, 2018.

[LWZ20] Rong Liu, Jinling Wang, and Bingqi Zhang. High definition map for automated driving: Overview and analysis. *The Journal of Navigation*, 2020.

[LXG21] Yiyi Liao, Jun Xie, and Andreas Geiger. KITTI-360: A novel dataset and benchmarks for urban scene understanding in 2D and 3D. *Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[LZS20] Minghua Liu, Xiaoshuai Zhang, and Hao Su. Meshing point clouds with predicted intrinsic-extrinsic ratio guidance. *ECCV*, 2020.

[Mac96] David A MacLean. Forest management strategies to reduce spruce budworm damage in the fundy model forest. *The Forestry Chronicle*, 1996.

[Man18] Cedric Manlhiot. Machine learning for predictive analytics in medicine: real opportunity or overblown hype? *European Heart Journal-Cardiovascular Imaging*, 2018.

[MBVH09] Daniel Munoz, J Andrew Bagnell, Nicolas Vandapel, and Martial Hebert. Contextual classification with functional max-margin Markov networks. *CVPR*, 2009.

[MD03] Carsten Moenning and Neil A Dodgson. Fast marching farthest point sampling. *Eurographics*, 2003.

[Mea82] Donald Meagher. Geometric modeling using octree encoding. *Computer Graphics and Image Processing*, 1982.

[MKH19] Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. When does label smoothing help? *NeurIPS*, 2019.

[MKL+14] Heather McNairn, Angela Kross, David Lapen, R Caves, and Jiali Shang. Early season monitoring of corn and soybeans with terrasar-x and radarsat-2. *International Journal of Applied Earth Observation and Geoinformation*, 2014.

[MLL+19] Qingfei Min, Yangguang Lu, Zhiyong Liu, Chao Su, and Bo Wang. Machine learning based digital twin framework for production optimization in petrochemical industry. *International Journal of Information Management*, 2019.

[MLRF+21] Jorge Andres Chamorro Martinez, Laura Elena Cué La Rosa, Raul Queiroz Feitosa, Ieda Del'Arco Sanches, and Patrick Nigri Happ. Fully convolutional recurrent networks for multidate crop recognition from multitemporal image sequences. *ISPRS Journal of Photogrammetry and Remote Sensing*, 2021.

[MON+19] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: learning 3D reconstruction in function space. *CVPR*, 2019.

[MPH+22] Matti Mõttus, Phu Pham, Eelis Halme, Matthieu Molinier, Hai Cu, and Jorma Laaksonen. TAIGA: a novel dataset for multitask learning of continuous and categorical forest variables from hyperspectral imagery. *Transactions on Geoscience and Remote Sensing*, 2022.

[MPT20] Khairiya Mudrik Masoud, Claudio Persello, and Valentyn A Tolpekin. Delineation of agricultural field boundaries from Sentinel-2 images using a novel super-resolution contour detector based on fully convolutional networks. *Remote sensing*, 2020.

[MR11] Donald McKenzie and Crystal Raymond. Modeling understory vegetation and its response to fire. *Models for Planning Wildlife Conservation in Large Landscapes*, 2011.

[MRB+19] Cheikh Mbow, Cynthia Rosenzweig, Luis G Barioni, Tim G Benton, Mario Herrero, Murukesan Krishnapillai, Emma Liwenga, Prajal Pradhan, M-G Rivera-Ferre, T Sapkota, et al. *Food security*. Intergovernmental Panel on Climate Change, 2019.

[MS89] David Bryant Mumford and Jayant Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on Pure and Applied Mathematics*, 1989.

[MSF20] Saba Mehmood, Muhammad Shahzad, and Muhammad Moazam Fraz. Dcarn: Deep context aware recurrent neural network for semantic segmentation of large scale unstructured 3D point cloud. *Neural Processing Letters*, 2020.

[MV21] Rohit Mohan and Abhinav Valada. Efficientps: Efficient panoptic segmentation. *International Journal of Computer Vision*, 2021.

[MVG+09] Sebastian Martinuzzi, Lee Vierling, William Gould, Michael Falkowski, Jeffrey Evans, Andrew Hudak, and Kerri Vierling. Mapping snags and understory shrubs for a LiDAR-based assessment of wildlife habitat suitability. *Remote Sensing of Environment*, 2009.

[MWJ13] Kevin Murphy, Yair Weiss, and Michael I Jordan. Loopy belief propagation for approximate inference: An empirical study. *UAI*, 2013.

[MYWC18] Thomas Möllenhoff, Zhenzhang Ye, Tao Wu, and Daniel Cremers. Combinatorial preconditioners for proximal algorithms on graphs. *AISTATS*, 2018.

[Nag01] Harini Nagendra. Using remote sensing to assess biodiversity. *International Journal of Remote Sensing*, 2001.

[NNSP14] Mohammad Najafi, Sarah Taghavi Namin, Mathieu Salzmann, and Lars Petersson. Non-associative higher-order Markov networks for point cloud classification. *ECCV*, 2014.

[NPA22] Joachim Nyborg, Charlotte Pelletier, and Ira Assent. Generalized classification of satellite image time series with thermal positional encoding. *CVPR Workshops*, 2022.

[NSF12] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from RGBD images. *ECCV*, 2012.

[NSGW21] Steven A Niederer, Michael S Sacks, Mark Girolami, and Karen Willcox. Scaling digital twins from the artisanal to the industrial. *Nature Computational Science*, 2021.

[OAPL21] Stella Ofori-Ampofo, Charlotte Pelletier, and Stefan Lang. Crop type mapping from optical and radar time series using attention-based deep learning. *Remote Sensing*, 2021.

[OID15] Julien Osman, Jordi Inglada, and Jean-François Dejoux. Assessment of a markov logic model of crop rotations for early crop mapping. *Computers and Electronics in Agriculture*, 2015.

[OLV18] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[PASW13] Jeremie Papon, Alexey Abramov, Markus Schoeler, and Florentin Wörgötter. Voxel cloud connectivity segmentation - supervoxels for point clouds. *CVPR*, 2013.

[PBI+22] Heiko Prümers, Carla Jaimes Betancourt, José Iriarte, Mark Robinson, and Martin Schaich. LiDAR reveals pre-Hispanic low-density urbanism in the Bolivian Amazon. *Nature*, 2022.

[PBM+18] Laura Pastonchi, Anna Barra, Oriol Monserrat, Guido Luzi, Lorenzo Solari, and Veronica Tofani. Satellite data to improve the knowledge of geohazards in world heritage sites. *Remote Sensing*, 2018.

[PC11] Thomas Pock and Antonin Chambolle. Diagonal preconditioning for first order primal-dual algorithms in convex optimization. *ICCV*, 2011.

[Pea01] Karl Pearson. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 1901.

[Pea82] Judea Pearl. *Reverend Bayes on inference engines: A distributed hierarchical approach*. Cognitive Systems Laboratory, School of Engineering and Applied Science, 1982.

[PFS+19] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning continuous signed distance functions for shape representation. *CVPR*, 2019.

[PI18] Alexander Y Prosekov and Svetlana A Ivanova. Food security: The challenge of the present. *Geoforum*, 2018.

[PJL+21] Songyou Peng, Chiyu "Max" Jiang, Yiyi Liao, Michael Niemeyer, Marc Pollefeys, and Andreas Geiger. Shape as points: A differentiable poisson solver. *NeurIPS*, 2021.

[PKPG12] François Petitjean, Camille Kurtz, Nicolas Passat, and Pierre Gançarski. Spatio-temporal reasoning for the classification of satellite image time series. *Pattern Recognition Letters*, 2012.

[PNM+20] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. *ECCV*, 2020.

[Pot52] Renfrey Burnard Potts. Some generalized order-disorder transformations. *Mathematical Proceedings of the Cambridge Philosophical Society*, 1952.

[PPC+12] Nicolas Paparoditis, Jean-Pierre Papelard, Bertrand Cannelle, Alexandre Devaux, Bahman Soheilian, Nicolas David, and Erwann Houzay. Stereopolis II: A multi-purpose and multi-sensor 3D mobile mapping system for street visualisation and 3D metrology. *Revue Française de Photogrammétrie et de Télédétection*, 2012.

[PRG+16] Pierre Potin, Betlem Rosich, Patrick Grimont, Nuno Miranda, Ian Shurmer, Alistair O'Connell, Ramon Torres, and Mike Krassenburg. Sentinel-1 mission status. *European Conference on Synthetic Aperture Radar*, 2016.

[PTB16] Eann A Patterson, Richard J Taylor, and Mark Bankhead. A framework for an integrated nuclear digital environment. *Progress in Nuclear Energy*, 2016.

[PVI+16] Charlotte Pelletier, Silvia Valero, Jordi Inglada, Nicolas Champion, and Gérard Dedieu. Assessing the robustness of random forests to map land cover with high resolution satellite image time series over large areas. *Remote Sensing of Environment*, 2016.

[PVK21] Maria Papadomanolaki, Maria Vakalopoulou, and Konstantinos Karantzalos. A deep multi-task learning framework coupling semantic segmentation and fully convolutional LSTM networks for urban change detection. *Transactions on Geoscience and Remote Sensing*, 2021.

[PWP19] Charlotte Pelletier, Geoffrey I Webb, and François Petitjean. Temporal convolutional neural network for the classification of satellite image time series. *Remote Sensing*, 2019.

[PZL17] Jian Peng, Huijuan Zhao, and Yanxu Liu. Urban ecological corridors construction: A review. *Acta Ecologica Sinica*, 2017.

[QSMG17a] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3D classification and segmentation. *CVPR*, 2017.

[QSMG17b] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. *CVPR*, 2017.

[QYSG17] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *NeurIPS*, 2017.

[Rag18] Hugo Raguet. A note on the forward-douglas–rachford splitting for monotone inclusion and convex optimization. *Optimization Letters*, 2018.

[RBG19] Santiago Royo and Maria Ballesta-Garcia. An overview of LiDAR imaging systems for autonomous vehicles. *Applied Sciences*, 2019.

[RCVS+19] Markus Reichstein, Gustau Camps-Valls, Bjorn Stevens, Martin Jung, Joachim Denzler, Nuno Carvalhais, et al. Deep learning and process understanding for data-driven earth system science. *Nature*, 2019.

[RCW+19] Rose Rustowicz, Robin Cheong, Lijing Wang, Stefano Ermon, Marshall Burke, and David Lobell. Semantic segmentation of crop type in africa: A novel dataset and analysis of deep learning methods. *CVPR Workshops*, 2019.

[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*, 2018.

[RDS+15] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 2015.

[RFB15] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *MICCAI*, 2015.

[RFP13] Hugo Raguet, Jalal Fadili, and Gabriel Peyré. A generalized forward-backward splitting. *SIAM Journal on Imaging Sciences*, 2013.

[RHV+18] Johannes Reiche, Eliakim Hamunyela, Jan Verbesselt, Dirk Hoekman, and Martin Herold. Improving near-real time deforestation monitoring in tropical dry forests by combining dense sentinel-1 time series with landsat and alos-2 palsar-2. *Remote Sensing of Environment*, 2018.

[RK18] Marc Rußwurm and Marco Körner. Convolutional LSTMs for cloud-robust segmentation of remote sensing imagery. *NeurIPS Workshops*, 2018.

[RK19] Marc Rußwurm and Marco Körner. Self-attention for raw optical satellite time series classification. *ISPRS Journal of Photogrammetry and Remote Sensing*, 2019.

[RKH+21] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. *ICML*, 2021.

[RMM20] Luka Rumora, Mario Miler, and Damir Medak. Impact of various atmospheric corrections on sentinel-2 land cover classification accuracy using machine learning classifiers. *ISPRS International Journal of Geo-Information*, 2020.

[ROF92] Leonid I Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 1992.

[RRW19] Tom R Robinson, Nick Rosser, and Richard J Walters. The spatial and temporal influence of cloud cover on satellite-based emergency mapping of earthquake disasters. *Scientific Reports*, 2019.

[SBAK20] Joel Segarra, Maria Luisa Buchaillot, Jose Luis Araus, and Shawn C Kefauver. Remote sensing for precision agriculture: Sentinel-2 improved features and applications. *Agronomy*, 2020.

[SBK21] Maja Schneider, Amelie Broszeit, and Marco Körner. Eurocrops: A pan-European dataset for time series crop type classification. *Proceedings of the 2021 Conference on Big Data from Space*, 2021.

[SBT+05] William D Shuster, James Bonta, Hale Thurston, Elizabeth Warnemuende, and DR Smith. Impacts of impervious surface on watershed hydrology: A review. *Urban Water Journal*, 2005.

[SIBD11] Charles Soussen, Jérôme Idier, David Brie, and Junbo Duan. From bernoulli–gaussian deconvolution to sparse signal restoration. *Transactions on Signal Processing*, 2011.

[SK17] Martin Simonovsky and Nikos Komodakis. Dynamic edge-conditioned filters in convolutional neural networks on graphs. *CVPR*, 2017.

[SK20] Maja Schneider and Marco Körner. [re] satellite image time series classification with pixel-set encoders and temporal self-attention. *ML Reproducibility Challenge*, 2020.

[SL08] Trenton F Stanger and Joseph G Lauer. Corn grain yield response to crop rotation and nitrogen over 35 years. *Agronomy Journal*, 2008.

[SM98] James A Slater and Stephen Malys. WGS 84—past, present and future. *Advances in Positioning and Reference Frames*, 1998.

[SNRS14] Alena Schmidt, Joachim Niemeyer, Franz Rottensteiner, and Uwe Soergel. Contextual classification of full waveform LiDAR data in the Wadden Sea. *Geoscience and Remote Sensing Letters*, 2014.

[SO20] Nicholas Sharp and Maks Ovsjanikov. PointTriNet: Learned triangulation of 3D point sets. *ECCV*, 2020.

[SOC01] David V. Sandberg, Roger D. Ottmar, and Geoffrey H. Cushon. Characterizing fuels in the 21st century. *International Journal of Wildland Fire*, 2001.

[SPI+19] Andrei Stoian, Vincent Poulain, Jordi Inglada, Victor Poughon, and Dawa Derksen. Land cover maps production with high resolution satellite image time series and convolutional neural networks: Adaptations and limits for operational systems. *Remote Sensing*, 2019.

[SSG17a] Nico Schertler, Bogdan Savchynskyy, and Stefan Gumhold. Towards globally optimal normal orientations for large point clouds. *Computer Graphics Forum*, 2017.

[SSG+17b] Thomas Schöps, Johannes L. Schönberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. *CVPR*, 2017.

[SSZ17] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *NeurIPS*, 2017.

[STAL20] Martin Sudmanns, Dirk Tiede, Hannah Augustin, and Stefan Lang. Assessing global sentinel-2 coverage dynamics and data availability for operational earth observation (eo) applications using the eo-compass. *International Journal of Digital Earth*, 2020.

[Str82] Wolfgang Strasser. Fast curve and surface generation for interactive shape design. *Computers in Industry*, 1982.

[SU15] Alexander G. Schwing and Raquel Urtasun. Fully connected deep structured networks. *CoRR*, 2015.

[SVB10] Roman Shapovalov, Er Velizhev, and Olga Barinova. Nonassociative Markov networks for 3D point cloud classification. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2010.

[SWNW18] Max J Steinhausen, Paul D Wagner, Balaji Narasimhan, and Björn Waske. Combining sentinel-1 and sentinel-2 data for improved land use and land cover mapping of monsoon regions. *International Journal of Applied Earth Observation and Geoinformation*, 2018.

[TC96] Italo Tamanini and Giuseppe Congedo. Optimal segmentation of unbounded functions. *Rendiconti del Seminario Matematico della Universita di Padova*, 1996.

[TDH+13] Mustafa Teke, Hüsne Seda Deveci, Onur Haliloğlu, Sevgi Zübeyde Gürbüz, and Ufuk Sakarya. A short survey of hyperspectral remote sensing applications in agriculture. *International Conference on Recent Advances in Space Technologies*, 2013.

[TGB19] Daniel Tenbrinck, Fjedor Gaede, and Martin Burger. Variational graph methods for efficient point cloud sparsification. *arXiv preprint arXiv:1903.02858*, 2019.

[TGD+18] Hugues Thomas, François Goulette, Jean-Emmanuel Deschaud, Beatriz Marcotegui, and Yann LeGall. Semantic classification of 3D point clouds with multiscale spherical neighborhoods. *3DV*, 2018.

[TGLM21] Neil C Thompson, Kristjan Greenewald, Keeheon Lee, and Gabriel F Manso. Deep learning's diminishing returns: the cost of improvement is becoming unsustainable. *Spectrum*, 2021.

[Tho52] Paul D Thomas. *Conformal projections in geodesy and cartography*. US Government Printing Office, 1952.

[TL16] Hui-Ting Tang and Yuh-Ming Lee. The making of sustainable urban development: a synthesis framework. *Sustainability*, 2016.

[TLB+22] Wei-Hsin Tseng, Hoàng-Ân Lê, Alexandre Boulch, Sébastien Lefèvre, and Dirk Tiede. Croco: Cross-modal contrastive learning for localization of earth observation data. *International Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2022.

[TLZ+20] Haotian* Tang, Zhijian* Liu, Shengyu Zhao, Yujun Lin, Ji Lin, Hanrui Wang, and Song Han. Searching efficient 3D architectures with sparse point-voxel convolution. *ECCV*, 2020.

[TMC+17] Romain Tavenard, Simon Malinowski, Laetitia Chapel, Adeline Bailly, Heider Sanchez, and Benjamin Bustos. Efficient temporal kernels between feature sets for time series classification. *ECML Workshop on Advanced Analytics and Learning on Temporal Data*, 2017.

[Tob70] Waldo R Tobler. A computer movie simulating urban growth in the detroit region. *Economic Geography*, 1970.

[TQD+19] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. KPConv: Flexible and deformable convolution for point clouds. *CVPR*, 2019.

[TQM+20] Weikai Tan, Nannan Qin, Lingfei Ma, Ying Li, Jing Du, Guorong Cai, Ke Yang, and Jonathan Li. Toronto-3D: A large-scale mobile LiDAR dataset for semantic segmentation of urban roadways. *CVPR Workshops*, 2020.

[TRW+21] Devis Tuia, Ribana Roscher, Jan Dirk Wegner, Nathan Jacobs, Xiaoxiang Zhu, and Gustau Camps-Valls. Toward a collective agenda on AI for earth science data analysis. *Geoscience and Remote Sensing Magazine*, 2021.

[TSA19] Pavel Tokmakov, Cordelia Schmid, and Karteek Alahari. Learning to segment moving objects. *International Journal of Computer Vision*, 2019.

[TVB+10] Mao-Ning Tuanmu, Andrés Viña, Scott Bearer, Weihua Xu, Zhiyun Ouyang, Hemin Zhang, and Jianguo Liu. Mapping understory vegetation using phenological characteristics derived from remotely sensed data. *Remote Sensing of Environment*, 2010.

[VAG20] Nina Varney, Vijayan K Asari, and Quinn Graehling. DALES: A large-scale aerial LiDAR data set for semantic segmentation. *CVPR Workshops*, 2020.

[VBR+17] Ján Vachálek, Lukás Bartalskỳ, Oliver Rovnỳ, Dana Šišmišová, Martin Morháč, and Milan Lokšík. The digital twin of an industrial production line within the industry 4.0 concept. *International Conference on Process Control*, 2017.

[VBS+15] Bruno Vallet, Mathieu Brédif, Andrés Serna, Beatriz Marcotegui, and Nicolas Paparoditis. TerraMobilita/iQmulus urban point cloud analysis benchmark. *Computers & Graphics*, 2015.

[VHMAS+18] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The iNaturalist species classification and detection dataset. *CVPR*, 2018.

[VLPK12] Hoang Hiep Vu, Patrick Labatut, Jean Philippe Pons, and Renaud Keriven. High accuracy and visibility-consistent dense multiview stereo. *Transactions on Pattern Analysis and Machine Intelligence*, 2012.

[VM98] Robert J Valkenburg and Alan M McIvor. Accurate 3D measurement using a structured light system. *Image and Vision Computing*, 1998.

[VMD+18] Anton Vrieling, Michele Meroni, Roshanak Darvishzadeh, Andrew K Skidmore, Tiejun Wang, Raul Zurita-Milla, Kees Oosterbeek, Brian O'Connor, and Marc Paganini. Vegetation phenology from Sentinel-2 and field cameras for a Dutch barrier island. *Remote Sensing of Environment*, 2018.

[VN20] Richard Van Noorden. The ethical questions that haunt facial-recognition research. *Nature*, 2020.

[VOOR19] Cláudia M Viana, Sandra Oliveira, Sérgio C Oliveira, and Jorge Rocha. Land use/land cover change detection and urban sprawl analysis. *Spatial modeling in GIS and R for Earth and environmental sciences*, 2019.

[VSP+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 2017.

[VTGGP18] Kristof Van Tricht, Anne Gobin, Sven Gilliams, and Isabelle Piccard. Synergistic use of radar sentinel-1 and optical sentinel-2 imagery for crop mapping: A case study for belgium. *Remote Sensing*, 2018.

[WD20] François Waldner and Foivos I Diakogiannis. Deep learning on edge: Extracting field boundaries from satellite images with a convolutional neural network. *Remote Sensing of Environment*, 2020.

[WH94] Nigel P Weatherill and Oubay Hassan. Efficient three-dimensional delaunay triangulation with automatic point creation and imposed boundary constraints. *International Journal for Numerical Methods in Engineering*, 1994.

[WH18] Yuxin Wu and Kaiming He. Group normalization. *ECCV*, 2018.

[WHL08] Qihao Weng, Xuefei Hu, and Dengsheng Lu. Extracting impervious surfaces from medium spatial resolution multispectral and hyperspectral imagery: a comparison. *International Journal of Remote Sensing*, 2008.

[WJHM15] Martin Weinmann, Boris Jutzi, Stefan Hinz, and Clément Mallet. Semantic point cloud interpretation based on op-

timal neighborhoods, relevant features and efficient classifiers. *ISPRS Journal of Photogrammetry and Remote Sensing*, 2015.

[WMG14] Michael Waechter, Nils Moehrle, and Michael Goesele. Let there be color! Large-scale texturing of 3D reconstructions. *ECCV*, 2014.

[WPC+21] Kan Wu, Houwen Peng, Minghao Chen, Jianlong Fu, and Hongyang Chao. Rethinking and improving relative position encoding for vision transformer. *ICCV*, 2021.

[WSM+15] Martin Weinmann, Alena Schmidt, Clément Mallet, Stefan Hinz, Franz Rottensteiner, and Boris Jutzi. Contextual classification of point cloud data by exploiting individual 3D neigbourhoods. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2015.

[WXS+20] Yuqing Wang, Zhaoliang Xu, Hao Shen, Baoshan Cheng, and Lirong Yang. Centermask: Single shot instance segmentation with point representation. *CVPR*, 2020.

[YBP20] Raghu Yaramasu, Varaprasad Bandaru, and Koutilya Pnvr. Pre-season crop type mapping using deep neural networks. *Computers and Electronics in Agriculture*, 2020.

[YCC+21] Lu Yuan, Dongdong Chen, Yi-Ling Chen, Noel Codella, Xiyang Dai, Jianfeng Gao, Houdong Hu, Xuedong Huang, Boxin Li, Chunyuan Li, et al. Florence: A new foundation model for computer vision. *arXiv preprint arXiv:2111.11432*, 2021.

[YCHNP19] Derek S Young, Xi Chen, Dilrukshi C Hewage, and Ricardo Nilo-Poyanco. Finite mixture-of-gamma distributions: estimation, inference, and model-based clustering. *Advances in Data Analysis and Classification*, 2019.

[YFST18] Yaoqing Yang, Chen Feng, Yiru Shen, and Dong Tian. FoldingNet: Point cloud auto-encoder via deep grid deformation. *CVPR*, 2018.

[YK09] Lexiang Ye and Eamonn Keogh. Time series shapelets: A new primitive for data mining. *ACM SIGKDD*, 2009.

[YL22] Mulin Yu and Florent Lafarge. Finding Good Configurations of Planar Primitives in Unorganized Point Clouds. *CVPR*, 2022.

[YZYL18] Hong-Ming Yang, Xu-Yao Zhang, Fei Yin, and Cheng-Lin Liu. Robust classification with convolutional prototype learning. *CVPR*, 2018.

[ZJR+15] Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip H. S. Torr. Conditional random fields as recurrent neural networks. *ICCV*, 2015.

[ZKR+17] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Ruslan R Salakhutdinov, and Alexander J Smola. Deep sets. *NeurIPS*, 2017.

[ZSH19] Yang Zhou, Shuhan Shen, and Zhanyi Hu. Detail preserved surface reconstruction from point cloud. *Sensors*, 2019.

[ZWK19] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019.

[ZZD+20] Yang Zhang, Zixiang Zhou, Philip David, Xiangyu Yue, Zerong Xi, Boqing Gong, and Hassan Foroosh. Polarnet: An improved grid representation for online LiDAR point clouds semantic segmentation. *CVPR*, 2020.

[ZZF21] Zixiang Zhou, Yang Zhang, and Hassan Foroosh. Panoptic-polarnet: Proposal-free LiDAR point cloud panoptic segmentation. *CVPR*, 2021.

[ZZW+21] Xinge Zhu, Hui Zhou, Tai Wang, Fangzhou Hong, Yuexin Ma, Wei Li, Hongsheng Li, and Dahua Lin. Cylindrical and asymmetrical 3D convolution networks for LiDAR segmentation. *CVPR*, 2021.

Miscellaneous

[BDT] BD TOPO. https://geoservices.ign.fr/documentation/donnees/vecteur/bdtopo. Accessed: 2022-06-24.

[CAP] The common agricultural policy at a glance. https://ec.europa.eu/info/food-farming-fisheries/key-policies/common-agricultural-policy/cap-glance_en. Accessed: 2021-09-24.

[geo] La démarche géocommuns. https://www.ign.fr/la-demarche-geocommuns. Accessed: 2022-07-01.

[Ico] Iconem. Iconem. https://iconem.com/en/. Accessed: 2022-06-24.

[IGN] IGN. LIDAR HD: une cartographie 3D du sol et du sursol de la france. https://geoservices.ign.fr/lidarhd. Accessed: 2022-07-24.

[Lyf] Lyft. Lyft self-driving safety report. https://autonomous.lyft.com/wp-content/uploads/2020/06/Safety_Report_2020.pdf. Accessed: 2022-07-24.

[Map] Open Street Map. Open street map. https://www.openstreetmap.org/. Accessed: 2022-07-01.

[Rie17] Christoph Rieke. Deep learning for instance segmentation of agricultural fields. https://github.com/chrieke/InstanceSegmentation_Sentinel2, 2017. Accessed: 2022-09-27.

[RPG] Registre parcellaire graphique (RPG) : Contours des parcelles et îlots culturaux et leur groupe de cultures majoritaire. https://www.data.gouv.fr/en/datasets/registre-parcellaire-graphique-rpg-contours-des-parcelles-et-ilots-culturaux-et-leur-groupe-de-cultures-majoritaire/. Accessed: 2021-09-24.

[RWC+19] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI Blog*, 2019. Accessed: 2021-10-21.

[Val] Valeo.ai. Valeo drive4u, the first autonomous car to be demonstrated on the streets of paris. https://www.valeo.com/en/valeo-drive4u-the-first-autonomous-car-to-be-demonstrated-on-the-streets-of-paris/. Accessed: 2022-07-24.

[Zoo] Zoox. Zoox: Built for riders, not drivers. https://zoox.com/vehicle/. Accessed: 2022-07-24.