ISPRS Journal of Photogrammetry and Remote Sensing 132 (2017) 102–118

Contents lists available at ScienceDirect

ISPRS Journal of Photogrammetry and Remote Sensing

journal homepage: www.elsevier.com/locate/isprsjprs

A structured regularization framework for spatially smoothing semantic labelings of 3D point clouds

Loïc Landrieu [a,*], Hugo Raguet [b], Bruno Vallet [a], Clément Mallet [a], Martin Weinmann [a,c,*]

[a] Univ. Paris-Est, LASTIG MATIS, IGN, ENSG, F-94160 Saint-Mande, France
[b] Institut de Mathématiques de Marseille, Aix-Marseille Université, 39 rue Frédéric Joliot-Curie, F-13453 Marseille, France
[c] Institute of Photogrammetry and Remote Sensing, Karlsruhe Institute of Technology (KIT), Englerstraße 7, D-76131 Karlsruhe, Germany

ARTICLE INFO

Article history:
Received 7 April 2017
Received in revised form 25 August 2017
Accepted 26 August 2017
Available online 11 September 2017

Keywords:
Semantic labeling
Point cloud
Classification
Regularization
Scene interpretation
Structured optimization
Graphical models

ABSTRACT

In this paper, we introduce a mathematical framework for obtaining spatially smooth semantic labelings of 3D point clouds from a pointwise classification. We argue that structured regularization offers a more versatile alternative to the standard graphical model approach. Indeed, our framework allows us to choose between a wide range of fidelity functions and regularizers, influencing the properties of the solution. In particular, we investigate the conditions under which the smoothed labeling remains probabilistic in nature, allowing us to measure the uncertainty associated with each label. Finally, we present efficient algorithms to solve the corresponding optimization problems.

To demonstrate the performance of our approach, we present classification results derived for standard benchmark datasets. We demonstrate that the structured regularization framework offers higher accuracy at a lighter computational cost in comparison to the classic graphical model approach.

1. Introduction

Due to the steadily increasing availability of geospatial information, their automated analysis has become a topic of major interest in photogrammetry, remote sensing, robotics, and computer vision. In particular, the representation of a scene in the form of a 3D point cloud and a subsequent semantic interpretation of this point cloud serve as the basis for many applications, such as scene modeling, autonomous navigation, or object detection. For instance, the analysis of 3D point cloud data acquired within urban environments benefits from a semantic labeling since the latter can be exploited for the creation of large-scale city models (Lafarge and Mallet, 2012) or urban accessibility diagnosis (Serna and Marcotegui, 2013).

The semantic interpretation typically consists in assigning a semantic label (e.g. *building*, *ground* or *vegetation*) to each point of the considered 3D point cloud, as shown in Fig. 1a–c. This assignment can be accompanied by an estimation of the confidence of the labeling of each point in the form of a probability distribution over the labels, as illustrated in Fig. 1d. Such a certainty assessment can prove useful when either the precision or the recall of the classification is more crucial for a given application. In the case of autonomous navigation for example, merely the possibility of an obstacle can be enough to alter course, and a probabilistic occupancy map is preferred to a binary one (Moravec and Elfes, 1985; Hornung et al., 2013). In the case of reconstruction tasks which necessitate the removal of a specific semantic class beforehand (Clode et al., 2004), precision is the focus in order to not accidentally remove relevant information. In a context of active learning, an assessment of the labeling certainty can guide an operator to the areas of the point cloud in which the classification is least certain, as they are more prone to be labeled incorrectly and might require manual re-labeling (Jing et al., 2004). The nature of the assignment, either a probability or a label, depends on the choice of the method used for inference.

The semantic labeling of 3D point clouds has been addressed by numerous investigations (Munoz et al., 2009; Shapovalov et al., 2010; Mallet et al., 2011; Niemeyer et al., 2014; Xu et al., 2014; Guo et al., 2015; Weinmann et al., 2015a; Weinmann, 2016; Hackel et al., 2016b). However, this problem remains challenging due to the irregular point sampling, occlusions, and the complexity of the considered scenes, which induce a loose yet meaningful structure to the data. Furthermore, the consideration of larger scenes typically results in a huge amount of data and efficient

* Corresponding authors at: Univ. Paris-Est, LASTIG MATIS, IGN, ENSG, F-94160 Saint-Mande, France (L. Landrieu and M. Weinmann).
E-mail addresses: loic.landrieu@ign.fr (L. Landrieu), hugo.raguet@gmail.com (H. Raguet), bruno.vallet@ign.fr (B. Vallet), clement.mallet@ign.fr (C. Mallet), martin.weinmann@kit.edu (M. Weinmann).

http://dx.doi.org/10.1016/j.isprsjprs.2017.08.010
0924-2716/

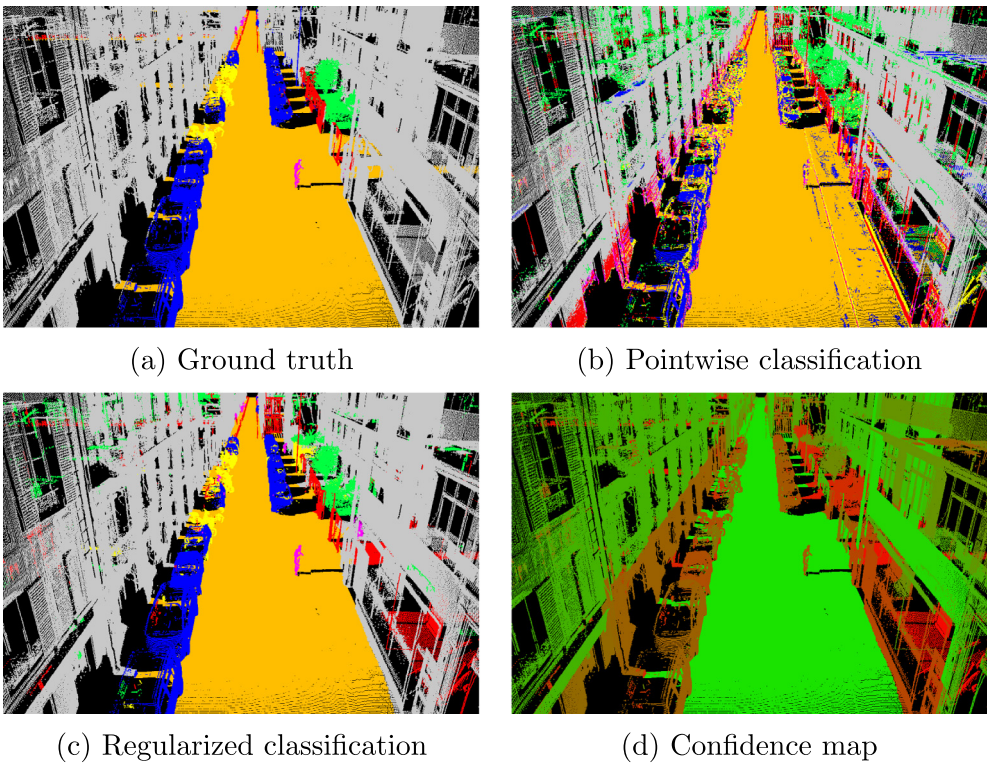


Fig. 1. Visualization of a 3D point cloud labeling for a part of the Paris-rue-Cassette Database (Vallet et al., 2014). In (a), (b), and (c), the color encoding addresses the classes *Façade* (gray), *Ground* (orange), *Cars* (blue), *2-Wheelers* (yellow), *Road Inventory* (red), *Pedestrians* (magenta) and *Vegetation* (green). In (d), the confidence is represented from green to red: confident uncertain. Remark that misclassifications in (c) correspond to the least confident areas in (d).

techniques for semantic labeling are therefore desirable. To foster research regarding semantic labeling of 3D point cloud data, a variety of benchmark datasets acquired within urban environments have been released (Munoz et al., 2009; Serna et al., 2014; Vallet et al., 2015; Hackel et al., 2016b).

The straightforward approach for semantically labeling a considered 3D point cloud consists in extracting a variety of features for all points, concatenating these features to a feature vector, which is then classified with a classifier trained on representative training examples. This strategy has for instance been followed in the framework introduced by Weinmann et al. (2015a), in which a diversity of distinctive geometric low-level features serve as input for a standard supervised classification scheme. While this rather simple approach already yields good classification rates due to the use of distinctive features, the visualization of the classified 3D point cloud reveals a noisy behavior as each point is treated individually by only considering the respective feature vector for classification. To illustrate this effect, we provide a ground truth labeling for a considered 3D point cloud in Fig. 1a and the result of a pointwise classification based on distinctive geometric low-level features in Fig. 1b. Considering the ground truth labeling, one can observe a high spatial regularity of the labeling. Indeed, as the number of 3D points far exceeds the number of objects in the scene, it is reasonable to assume that most 3D points are surrounded by points of the same label.

To impose spatial smoothness on this classification result, contextual information among neighboring 3D points is typically taken into account. For this purpose, the spatial structure of a 3D point cloud can be captured by a graph encoding the adjacency relationship between 3D points. Thereby, the adjacency relationship can be derived from the local neighborhood of each 3D point (Weinmann et al., 2015b), pre-segmentations (Niemeyer et al., 2016), or super-voxel structures (Lim and Suter, 2009). Based on the defined adjacency relationship, a context model is typically derived in the form of a graphical model, e.g. a Markov random field (MRF) (Munoz et al., 2009; Shapovalov et al., 2010; Lu and Rasmussen, 2012; Najafi et al., 2014) or its discriminative counterpart, the conditional random field (CRF) (Niemeyer et al., 2011; Niemeyer et al., 2014; Schmidt et al., 2014; Weinmann et al., 2015b). As a consequence of imposing spatial smoothness on the derived labeling, the corresponding classification results are often significantly improved as can be observed in Fig. 1c. However, the choice of the inference strategy (marginal, maximum-a-posteriori) will have a profound impact on the precision and nature of the solution (probabilistic or labeling), as well as the computation times.

In this paper, we propose to consider the problem of spatially smoothing semantic labelings of 3D point clouds from a structured regularization perspective. While using such a model results in a loss of interpretability compared to a probabilistic approach, it offers several advantages. In particular, the structured regularization approach allows:

- the choice from a wide range of fidelity functions and regularizers,[1]
- the choice to retain or not the probabilistic aspect of the input labeling, and
- the use of fast solving algorithms, compared to slow and memory-intensive message-passing algorithms.

After briefly introducing the used notation and the formal description of the considered problem, we summarize related work

[1] Notably, this framework allows us to express the graphical model approach as a special instance.

in Section 2. Subsequently, in Section 3, we outline the main idea of the fundamental framework for pointwise point cloud classification, which follows the work presented in (Weinmann et al., 2015a; Weinmann, 2016) but additionally allows the choice between *hard labelings*, in which each 3D point is assigned one unique class label, and *soft labelings* in the form of class probabilities.

As the labelings obtained via pointwise classification are not generally spatially smooth, we present a general regularization framework that takes into account the fact that class labels of neighboring 3D points tend to be correlated. In Section 4, we present the label-smoothing problem as a regularization problem structured by an adjacency graph, and we present four fidelity functions as well as two graph-structured regularizers and two possible search spaces. The choice of a fitting minimizing algorithm hinges on the respective properties of the fidelity and regularizing functions. In Section 5, we present efficient, state-of-the-art algorithms to solve the different cases encountered. Furthermore, we present a novel extension of the ℓ_0-cut pursuit algorithm presented by Landrieu and Obozinski (2016a), allowing the input to take the form of multi-dimensional probabilities instead of one-dimensional values only. To demonstrate the performance of our methodology, we present in Section 6 experimental results for different configurations of our framework as well as for state-of-the-art approaches. We focus on the quality of the resulting classifications, and to stress the advantages of probabilistic labelings, we present the partial coverage classification as well. Finally, in Section 7, we provide concluding remarks and suggestions for future work.

1.1. Notation

We denote V the finite set of 3D points to label, and $\mathcal{K}$ the finite set of potential semantic labels for each 3D point. Throughout this paper, we make an important distinction between hard and soft labelings. In that regard, we denote by $\mathcal{S}$ the simplex:

$$\mathcal{S} = \{p \in [0,1]^{\mathcal{K}} | \sum_{k \in \mathcal{K}} p_k = 1\}. \tag{1}$$

Elements of $\mathcal{S}$ are called *soft labelings*. Similarly, we denote $\mathbf{S}$ the corners of the simplex:

$$\mathbf{S} = \{p \in \{0,1\}^{\mathcal{K}} | \sum_{k \in \mathcal{K}} p_k = 1\}. \tag{2}$$

Elements of $\mathbf{S}$ are called *hard labelings*. For a labeling $p \in \mathcal{S}$ or $\mathbf{S}$ and a semantic class $k \in \mathcal{K}$, we denote the probability associated with class k by p_k, and consider the probability p as a vector of size $|\mathcal{K}|$.

Throughout this paper, we denote labelings of a single 3D point in lowercase, and global labelings, relative to the entire point cloud V, in uppercase. For such a global labeling $P \in \mathcal{S}^V$ or $\mathbf{S}^V$, we denote the labeling of a point $i \in V$ by P_i and its probability for class k by $P_{i,k}$.

1.2. Problem statement

We consider a set of 3D points V for which we have a soft labeling $P \in \mathcal{S}^V$ obtained via a classification algorithm which does not directly account for spatial smoothness. The goal is to find P^*, an improved labeling with increased spatial smoothness while remaining as close as possible to the input labeling P. Our proposed approach is to define P^* as the solution of a well-chosen optimization problem, whose *objective functional* is structured by an adjacency graph capturing the spatial relationship between the 3D points. P^* can be either a soft labeling or a hard labeling depending on the parameterization of the regularization problem. This process can be broken down into three parts:

- **Computing the initial labeling:** The proposed regularization framework is not affected by the choice of the method used to obtain the initial classification. However, our approach is more suited when the initial labeling is probabilistic. In this paper, we use a classification framework which is described in Section 3 and relies on the use of a diversity of low-level geometric 3D and 2D features as input for a standard random forest classifier (Weinmann et al., 2015a).
- **Parameterizing the regularization problem:** We define P^* as the result of an optimization problem with the following structure:

$$P^* \in \arg\min_{Q \in \Omega^V} \{\Phi(P,Q) + \lambda \Psi(Q)\}, \tag{3}$$

  where Φ is the *fidelity term*, Ψ the *regularizer*, $\lambda > 0$ the *regularization strength*, and Ω the search space. The fidelity term $\Phi(P,Q)$ enforces the influence of the initial labeling P, in the sense that it decreases as Q is closer to P. The regularizer Ψ favors solutions that are spatially smooth, in the sense that most adjacent nodes share the same label. The regularization strength λ is a user-defined parameter which dictates the influence of the regularization with respect to the fidelity term. In Section 4, we present the respective advantages of four fidelity terms, two regularizers, and two different search spaces.
- **Solving the optimization problem:** The choice of the minimizing algorithm to solve the regularization problem (3) hinges on the respective properties of the fidelity term and regularizer functions. In particular, we distinguish three settings: combinatorial, convex continuous and non-convex continuous problems. In Section 5, we present efficient, state-of-the-art algorithms for each case.

2. Related work

In recent years, a lot of attention has been paid to the semantic classification of 3D point clouds. Many investigations focus on a pointwise classification (Section 2.1) which serves as initial labeling for our framework. Spatially smooth labelings are subsequently obtained by computing an adjacency structure (Section 2.2) which allows for contextual classification (Section 2.3).

2.1. Semantic classification of 3D point clouds

The classic approach for point cloud classification is to treat each point individually by extracting a set of handcrafted features describing that point and using the respective feature vector as input for a standard supervised classification algorithm. Consequently, much effort has been spent on feature extraction and the classification procedure itself. In the following paragraphs, we summarize the main ideas behind both aspects.

A variety of handcrafted 3D shape features derived from the 3D structure tensor have been presented in different investigations (West et al., 2004; Pauly et al., 2003). Those features are advantageously completed by further characterization of the local 3D structure, e.g. in terms of angular statistics (Munoz et al., 2009), height and plane characteristics (Mallet et al., 2011; Guo et al., 2015), low-level 3D and 2D features (Weinmann et al., 2015a), or moments and height features (Hackel et al., 2016b).

The definition of an appropriate local neighborhood that comprises the local 3D structure is a crucial issue as it serves as the basis for feature extraction. Such neighborhoods are parameterized with a single parameter, commonly referred to as the *scale* and typically represented by the radius (Lee and Schenk, 2002; Filin and

Pfeifer, 2005) or the number of nearest neighbors considered (Linsen and Prautzsch, 2001). To avoid invoking prior knowledge about the scene, a data-driven solution for selecting the optimal neighborhood size of each point is desirable. Respective approaches are for instance based on the local surface variation (Pauly et al., 2003; Belton and Lichti, 2006) and the combined consideration of curvature, point density and noise of normal estimation (Mitra and Nguyen, 2003; Lalonde et al., 2005). Further approaches have been presented with dimensionality-based scale selection (Demantké et al., 2011), and eigenentropy-based scale selection (Weinmann et al., 2015a; Weinmann, 2016).

Other approaches focus on the computation of local 3D features at different scales. In this regard, it has been proposed to consider a collection of spherical neighborhoods (Brodu and Lague, 2012), a collection of cylindrical neighborhoods (Niemeyer et al., 2014), a combination of cylindrical and spherical neighborhoods (Blomley et al., 2016), a combination of neighborhoods in the form of voxels, blocks and pillars (Hu et al., 2013), or a combination of neighborhoods in the form of spatial bins, planar segments and local neighborhoods (Gevaert et al., 2016). A different strategy has been followed with the generation of a scale pyramid by repeated downsampling via a voxel-grid filter (Hackel et al., 2016b), which allows for calculating features based on a fixed, small number of nearest neighbors for each of these scales.

Depending on the system used for data acquisition, further types of data can be recorded in addition to spatial coordinates. Accordingly, complementary types of features can be derived based on the additional data, e.g. echo-based features (Chehata et al., 2009; Mallet et al., 2011; Waldhauser et al., 2014), full-waveform features (Chehata et al., 2009; Mallet et al., 2011) or radiometric features (Niemeyer et al., 2014; Schmidt et al., 2014).

The extracted features are concatenated to feature vectors that are provided as input to a classifier. In most cases, the focus is put on supervised classification. Accordingly, representative training data is required to train the involved classifier so that it is afterwards able to generalize to unseen data and thus able to assign a (semantic) class label to each point of the point cloud.

A variety of techniques may be applied for supervised classification, such as random forest classifiers (Chehata et al., 2009), support vector machine classifiers (Mallet et al., 2011), or Bayesian discriminant analysis classifiers (Khoshelham and Oude Elberink, 2012). Those classifiers are rather easy to use and meanwhile available in numerous software tools. A variety of such standard classifiers has recently been involved in a comprehensive study focusing on the classification of mobile laser scanning data (Weinmann et al., 2015a; Weinmann, 2016), where the derived results reveal that a random forest classifier provides a good trade-off between classification accuracy and computational efficiency. However, due to the pointwise consideration relying only on a feature vector per point, the labeling derived with such standard classifiers typically lacks spatial regularity, i.e. the classified point cloud typically reveals a "noisy" behavior, although it should be taken into account that class labels of neighboring 3D points tend to be correlated.

Instead of this classic approach, recent advances in deep learning have shown promising results for the semantic classification of point clouds with learned features. The straightforward approach in this regard consists in the direct adaptation of Convolutional Neural Networks (CNNs) to 3D data. Accordingly, the considered point cloud is typically transformed to a regular 3D voxel grid to allow for efficient convolutions. While different approaches for point cloud classification have been presented, the most promising ones focus on classifying each 3D point of a point cloud based on a transformation of all points within its local neighborhood to a voxel-occupancy grid serving as input for a 3D-CNN (Savinov, 2017; Hackel et al., 2016a; Huang and You, 2016). Instead of relying on a voxelization, the most recent progress focuses on the use of 2D image projections as input for a 2D-CNN designed for semantic segmentation and a subsequent backprojection of predicted labels to 3D space (Boulch et al., 2017; Lawin et al., 2017), on directly taking the considered point cloud as input for a neural network (Qi et al., 2016) or on generalizing the convolution operator from regular grids to arbitrary graphs (Simonovsky and Komodakis, 2017). It is interesting to note that the classification results of deep learning approaches tend to be smoother than those of pointwise classification approaches. However, the regularity is a by-product of the meta-parameterization of the network, such as the size of the convolution masks. In our approach, the regularity is adaptive in the strength of semantic consensus of neighboring points as well as the local connectivity, and can be controlled with a single scalar parameter.

Since the regularization framework proposed in this paper is independent of the choice of the method used for the initial labeling, we focus on the use of standard techniques for the latter. More specifically, we use an existing classification framework[2] presented by Weinmann et al. (2015a), which is based on the use of a variety of low-level geometric 3D and 2D features as input for a standard random forest classifier. The choice of this classification framework is motivated by the fact that it (1) focuses on a data-driven neighborhood recovery and is thus applicable for different point clouds without involving prior knowledge about the scene and/or the data, (2) exploits a set of informative features that are still interpretable, (3) already provides a reasonable initial labeling, and (4) can easily be adapted to produce a soft labeling rather than a hard one.

2.2. Graph structure of point clouds

Statistical context models are commonly used for modeling the relationship between neighboring points, and, consequently, imposing spatial regularity on the semantic labelings of 3D points clouds. In general, context models are based on the construction of an adjacency graph defining the extent to which interactions in a local neighborhood are considered, i.e. it is possible to take into account short-, mid- and long-range dependencies. It is important to note that this adjacency graph is in general not the same as the neighborhood graph used to compute local geometric features.

There are numerous approaches to obtain such adjacency graphs. The most common is to derive the graph from a neighborhood relationship, from nearest neighbors graph (Shapovalov et al., 2010) to cylindrical (Filin and Pfeifer, 2005; Niemeyer et al., 2014) or adaptive neighborhoods (Demantké et al., 2011; Weinmann et al., 2015b). Thereby, a simplifying assumption is typically made by only considering short-range dependencies. This is motivated by the observation that the quality of derived classification results reveals a saturation effect when increasing the scale parameter of the local neighborhood used for defining the adjacency graph. In this regard, the average number of involved neighbors was 7 in an investigation focusing on the classification of airborne laser scanning data (Niemeyer et al., 2011, 2014). Similar observations have been made in an investigation focusing on the classification of mobile laser scanning data (Weinmann et al., 2015b), where the consideration of short-range dependencies already delivers classification results of high quality. However, the latter investigation also indicates that adapting the size of the local neighborhood used for defining the adjacency graph with respect to the locally-adaptive neighborhood used for feature extraction is favorable in comparison to fixed neighborhoods. In these experiments, the average number of involved neighbors was between 15 and 21,

[2] Respective implementations can be found at http://www.ipf.kit.edu/code.php.

depending on the approach used for deriving locally-adaptive neighborhoods.

Instead of defining context models on the basis of neighboring points, different entities may be used as well. In this regard, several investigations advocate a super-voxel-based approach to represent the higher order structure (Lim and Suter, 2009; Niemeyer et al., 2016; Guignard and Landrieu, 2017).

2.3. Spatially smooth labeling

To derive a labeling with a higher spatial regularity, smooth labeling techniques (Schindler, 2012) or approaches for contextual classification can be used. The latter consider an initial labeling derived with a standard classifier and use a statistical context model to increase spatial regularity. Thereby, the classification of a given point does not only take into account the feature vector corresponding to this considered point, but also the labels (and possibly the feature vectors) corresponding to neighboring points.

Respective approaches have for instance been used in the form of associative Markov networks (Munoz et al., 2009), non-associative Markov networks (Shapovalov et al., 2010), conditional random fields (Niemeyer et al., 2014; Schmidt et al., 2014; Weinmann et al., 2015b; Landrieu et al., 2017), simplified Markov random fields (Lu and Rasmussen, 2012), multi-stage inference procedures focusing on point cloud statistics and relational information over different scales (Xiong et al., 2011), spatial inference machines modeling mid- and long-range dependencies inherent in the data (Shapovalov et al., 2013) and 3D entangled forests (Wolf et al., 2016).

Some of the presented approaches rely on the consideration of point cloud segments, e.g. (Shapovalov et al., 2010; Xiong et al., 2011), whereas others directly classify points, e.g. (Niemeyer et al., 2014). In this paper, we take into account that segment-based methods heavily depend on the quality of the results of the segmentation algorithm, and we therefore focus on the regularization of point clouds without the use of pre-segmentations or super-voxels.

In pairwise context models, such as CRFs or MRFs, retrieving the most likely spatially smooth labeling is referred to as maximum-a-posteriori inference. In practice, this labeling can only be approximated using efficient combinatorial optimization techniques, notably based on graph cuts. This form of inference produces a hard labeling, and hence loses the probabilistic nature of the initial labeling.

Alternatively, the label distribution can be computed for each node with marginal inference, which allows us to keep the probabilistic nature of the classification. However, marginal inference is typically approximated with message-passing algorithms such as loopy belief propagation (Niemeyer et al., 2011, 2014; Weinmann et al., 2015b), which are significantly slower and lead to classifications of lower likelihood and less accurate classification results, as detailed in (Landrieu et al., 2017).

Lellmann et al. (2013) propose a variational approach, in which the discrete label set is relaxed into a continuous one. This approach permits the use of convex optimization algorithms, and yields results with arguably less artifacts than the solutions of combinatorial optimization techniques. The article shows that, with reasonable assumptions, the solution of the optimization can easily be discretized into a spatially smooth labeling.

3. Probabilistic classification of point clouds

The complete methodology that we use for point cloud classification consists of two major steps summarized in Fig. 2. The first step focuses on computing a pointwise probabilistic classification $P \in \mathcal{S}^V$ that serves as input for our regularization framework. While the exact method used to obtain said classification does not impact its smoothing, provided it is reasonably good, we present the fundamentals of pointwise semantic labeling for the sake of completeness. We follow a slightly modified version of the framework presented by Weinmann et al. (2015a), in which the output is a soft labeling rather than a hard one. In the following, we briefly address the recovery of a local neighborhood for each 3D point (Section 3.1) which allows to compute low-level geometric 2D and 3D features

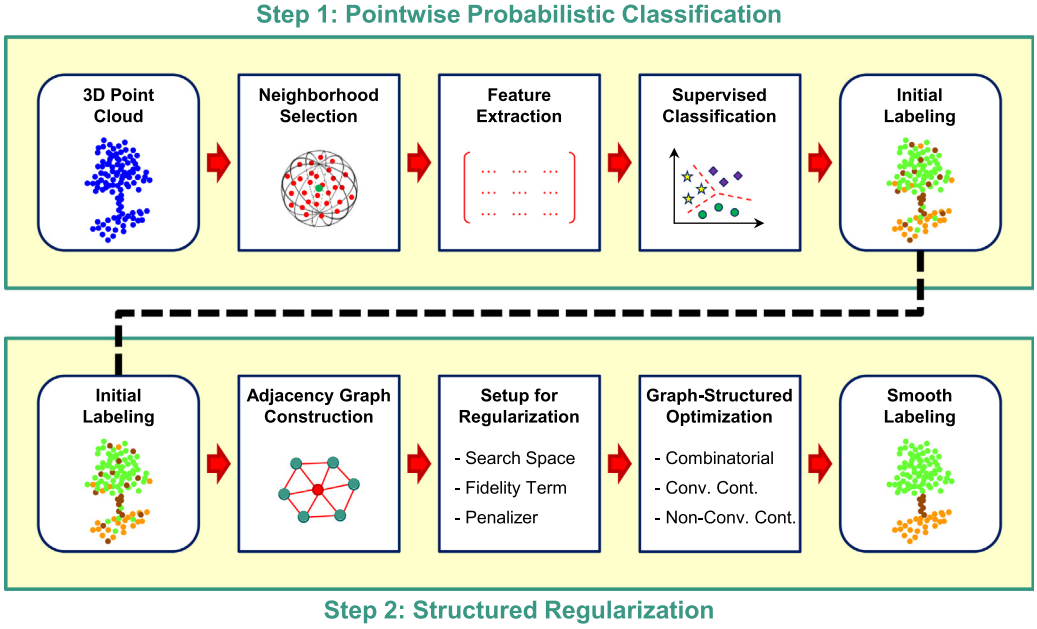


Fig. 2. Overview on the proposed methodology for the semantic labeling of 3D point clouds: the first step delivers a standard pointwise probabilistic classification, whereas the second step smoothes the initial labeling by structured regularization. For the latter, we involve two different search spaces ($\Omega = \mathbf{S}$ and $\Omega = \mathcal{S}$, i.e. hard and soft labeling), four different fidelity terms (linear, linear-logarithmic, quadratic and Kullback-Leibler fidelity), two penalizers (Potts penalty and total variation) and three settings for graph-structured optimization (combinatorial, convex continuous and non-convex continuous functional).

based on the point statistics within (Section 3.2), and finally the supervised classification based on the extracted features (Section 3.3).

3.1. Recovery of local neighborhoods

To appropriately describe the local 3D structure at a considered point of the point cloud, we take into account that we are dealing with mobile laser scanning data acquired in urban environments. In such scenarios, the point density varies strongly according to the distance of the target such that it is advisable to use a spherical neighborhood definition relying on a scale parameter in the form of either a radius or the number of nearest neighbors that are considered. To allow for flexibility with respect to the given data, we focus on a data-driven approach to determine the neighborhood size by selecting the number of nearest neighbors in the local 3D neighborhood of each individual point with *eigenentropy-based scale selection* (Weinmann et al., 2015a). This approach has proven to compare favorably to a variety of other approaches, and neither involves parameter tuning nor prior knowledge about the scene.

More specifically, for varying values of the scale parameter $s \in \mathbb{N}$, we use the 3D coordinates of each point and its s nearest neighbors to derive the respective 3D structure tensor $\mathbf{T} \in \mathbb{R}^{3\times3}$ as a function of the scale parameter s. The 3D structure tensor $\mathbf{T}$ is a symmetric positive semi-definite matrix, i.e. its three eigenvalues exist, are non-negative and correspond to an orthogonal system of eigenvectors. Once normalized by their sum, the eigenvalues $\mu_1(s) \geqslant \mu_2(s) \geqslant \mu_3(s) \geqslant 0$ can be considered as "quasi-probabilities" allowing us to define an energy function for optimal neighborhood size selection on the basis of the Shannon entropy:

$$E_\mu(s) = -\sum_{j=1}^{3} \mu_j(s)\, \ln(\mu_j(s)). \tag{4}$$

This energy function of the scale parameter s is known as the *eigenentropy* and describes the order/disorder of 3D points within the local 3D neighborhood. We select the parameter s_{opt} by minimizing the eigenentropy $E_\mu(s)$ across varying values of the scale parameter s:

$$s_{\text{opt}} = \arg\min_{s\in\mathcal{K}} E_\mu(s). \tag{5}$$

In the scope of our work, we test different values of s within a predefined set, with a lower boundary of $s_{\min} = 10$ neighbors to remain statistically meaningful (Demantké et al., 2011; Weinmann et al., 2015a; Weinmann, 2016) and an upper boundary of $s_{\max} = 100$ to limit the computational effort.

3.2. Extraction of low-level geometric 3D and 2D features

In the scope of this work, we describe each 3D point by considering all points within its local neighborhood of optimal size and calculating the respective values for a set of handcrafted geometric features proposed in (Weinmann et al., 2015a; Weinmann, 2016). These features are rather intuitive, and each feature is only represented by a single value.

The considered feature set comprises 14 low-level geometric 3D features. Eight of them are derived from the normalized eigenvalues μ_j and represented by linearity, planarity, sphericity, omnivariance, anisotropy, eigenentropy, sum of eigenvalues and change of curvature (West et al., 2004; Pauly et al., 2003). The other features are derived from the optimal neighborhood itself and given by the height of the considered point, the radius of the local neighborhood, the local point density, the verticality, and the maximum difference as well as the standard deviation of the height values corresponding to those points within the local neighborhood.

To take into account the particular role played by the vertical dimension in urban environments, we also consider 2D features defined in analogy to the 3D case. More specifically, we use the normalized eigenvalues of the 2D structure tensor derived from the 2D projections of a considered point and its s_{opt} nearest neighbors onto a horizontally oriented plane. We consider the 2D features determined by the sum and ratio of these normalized eigenvalues, as well as the radius of the local neighborhood and the local point density in the 2D projection.

The derived values for all features extracted for a point are finally concatenated to a feature vector. Taking into account that the defined features correspond to different quantities with different units, we introduce a normalization across all feature vectors which maps each dimension onto the interval $[0, 1]$. Thereby, the mapping function is defined based on the training data.

3.3. Supervised classification

To obtain an initial labeling based on the derived feature vectors, we focus on *ensemble learning* which relies on the idea of strategically generating a set of weak learners and combining them in order to create a single strong learner. A rather intuitive combination of weak learners is realized via bagging (Breiman, 1996), where bootstrapped replica of the training data (i.e. randomly drawn subsets) are used to train a set of weak learners of the same type. As each of the weak learners is trained on an independent subset, the weak learners are all randomly different from one another. This, in turn, results in a de-correlation between individual hypotheses and thus an improved generalization and robustness may be expected when taking the respective majority vote over all hypotheses (Criminisi and Shotton, 2013).

The most popular example for bagging is represented by a random forest classifier (Breiman, 2001) which relies on a set of N_T decision trees as weak learners and typically yields a good trade-off between accuracy and computational effort (Weinmann et al., 2015a; Weinmann, 2016). To obtain an initial global soft labeling, we use a random forest classifier whose parameters are cross-validated, and define the probability that a point i belongs to the class k as:

$$P_{i,k} = \frac{N_k}{N_T}, \tag{6}$$

where N_k is the number of decision trees having voted for class k. This soft labeling in the form of classwise probabilities may not be spatially regular, and hence may be used as the basis for a subsequent regularization as we explain in the following section.

4. Regularizing soft labelings on a weighted graph

In the second step of our methodology (cf. Fig. 2), we consider $P \in \mathcal{S}^n$ a global soft labeling of a 3D point cloud as input and seek an alternative labeling P^* with increased spatial regularity. In that regard, we define P^* as the solution of the structured optimization problem (3) with well-chosen search space, fidelity terms, and regularizing functions. This problem is said to be *structured*, as both fidelity terms and regularizers have a specific form derived from an adjacency graph $G = (V, E, w)$, defined in Section 4.1. In the following, we present and discuss the respective properties of different options for the search space (Section 4.2), the fidelity term (Section 4.3), and the regularizer (Section 4.4).

4.1. Adjacency graph of point clouds

In this paper, we focus on the regularization of semantic labelings, while investigating the respective benefits of different adjacency graphs is beyond the scope of our work. We chose a symmetrized 10-neighborhood adjacency graph with constant edge weight for its simplicity of implementation (Indyk and Motwani, 1998). However, our framework can naturally handle graphs with weighted edges, and we directly incorporate edge weights into our objective function.

4.2. Search space for probabilistic labelings

In this paper, we restrict the choice of Ω to hard or soft labelings: $\Omega = \mathbf{S}$ or $\Omega = \mathcal{S}$. While hard labelings assign a unique class, soft labelings assign a probability for each class, and consequently contain more information. However, producing a hard labeling for each point remains the main objective of semantic classification.

The most straightforward way to produce a hard labeling from a soft labeling is to assign the label which has the highest probability for each point independently, assuming it is unique. If it is not, the label can be chosen arbitrarily from the classes of highest probability, with the lowest index for example. In other words, for a soft labeling $P \in \mathcal{S}^n$, we define the associated hard labeling $\widehat{P} \in \mathbf{S}^n$ such that for all nodes $i \in V$:

$$\widehat{P}_{i,k} = \begin{cases} 1 & \text{if } k = \min \arg\max_{l \in \mathcal{K}} P_{i,l} \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

The main advantage of choosing $\Omega = \mathcal{S}$ over $\mathbf{S}$ is that a soft labeling allows the confidence assessment of the associated hard labeling through the computation of its entropy. This can be useful when the focus of the classification is precision rather than the full coverage of the point cloud. In such circumstances, the global labeling can be sorted by increasing entropy, ensuring that the first points have higher confidence.

4.3. Fidelity terms for regularizing distributions

The fidelity term $\Phi(P, \cdot) : \mathcal{S}^n \mapsto \mathbb{R}$ of the optimization problem defined in (3) enforces the influence of the soft labeling P in $\mathcal{S}^n$. In the scope of this paper, we focus on fidelity terms that are separable with respect to V:

$$\Phi(P, Q) = \sum_{i \in V} \phi(P_i, Q_i),$$

where $\phi(p, \cdot) : \mathbf{S} \mapsto \mathbb{R}$ is a smooth and convex function called the *fidelity function*. Such a function $\phi(p, q)$ must be minimal for $q = p$, and increases as q differs from p. It is important to note that the fidelity function must be defined on the convex domain $\mathcal{S}$, but can be restricted to $\Omega = \mathbf{S}$ since $\mathbf{S} \subset \mathcal{S}$.

In this section, we present four different choices for ϕ. This list is not comprehensive, and could be extended with the Riemannian distance (Aström et al., 2016), or non-differentiable norms such as the L_1-norm or its variant presented by Huber (1964).

4.3.1. Linear fidelity

The *linear fidelity* is traditionally used as a convex relaxation of unary potentials for labeling problems. We define it here as the opposite of the scalar product with the observed probability p:

$$\phi_{\text{linear}}(p, q) \doteq -\langle p, q \rangle = -\sum_{k \in \mathcal{K}} p_k q_k. \tag{8}$$

In accordance with general results of linear programming, the linear fidelity function encourages solutions that lie in the corner of the feasibility set, i.e. $\mathbf{S}$, as illustrated in Fig. 3a. Although the choice of the regularizer can alter this behavior, this fidelity function should be used when a hard labeling is preferred.

The main advantage of this fidelity function is its simplicity: it is a simple scalar product and its gradient is constant. However, linearly factoring the observed probability might be too simplistic, in particular when it comes to low observed probability. For example, the penalty for choosing two labels with probability 0.5 is the same than choosing one label with probability 0.

4.3.2. Linear-logarithmic fidelity

The *linear-logarithmic fidelity* is defined as the opposite of the scalar product with the logarithm of the observed probability p:

$$\phi_{\log}(p, q) \doteq -\langle q, \log(\hat{p}) \rangle = -\sum_{k \in \mathcal{K}} q_k \log(\hat{p}_k), \tag{9}$$

where log denotes the entrywise logarithm and $\hat{p}$ is a version of the observed probability which is smoothed to prevent numerical issues: $\hat{p}_k = \frac{\alpha}{K} + \alpha p_k$ with $\alpha \in [0, 1[$. Due to its linearity, this function tends to induce a hard labeling as well, as illustrated in Fig. 3a. We remark that the fidelity term of the log-likelihood in graphical models such as MRFs or CRFs corresponds to this fidelity function. However, the probabilistic modeling setting is restricted to hard labelings for q while we extend it to the simplex $\mathcal{S}$.

This fidelity, while still simple to compute, necessitates the tuning of a supplementary tuning parameter. In our experiments, the influence of α was fairly minimal and we chose $\alpha = 0.05$ across all experiments. The main advantage of this function is that it heavily penalizes choosing labels with low probability.

4.3.3. Quadratic fidelity

The *quadratic fidelity* corresponds to the sum of squared differences (SSD) between distributions:

$$\phi_{\text{quadratic}}(p, q) \doteq \|p - q\|^2 = \sum_{k \in \mathcal{K}} (p_k - q_k)^2. \tag{10}$$

Unlike the two linear functions presented above, this fidelity function does not favor hard labelings, and it hence retains the probabilistic nature of P. The penalty is proportional to the Euclidean distance on the simplex between the observed and the assigned probability, as represented in Fig. 3b.

4.3.4. Kullback-Leibler fidelity

The *Kullback-Leibler (KL) fidelity* relies on the Kullback-Leibler divergence $\text{KL}(p, q)$ which has been introduced by Kullback and Leibler (1951) as a measure of similarity between two distributions p and q:

$$\text{KL}(p, q) \doteq \sum_{k \in \mathcal{K}} p_k \log\left(\frac{p_k}{q_k}\right) = -\sum_{k \in \mathcal{K}} p_k \log(q_k) + \text{function of } p. \tag{11}$$

Since we are only interested in optimizing the fidelity function with respect to q, we can discard the constant part of the previous equation. As with the linear-logarithmic fidelity, we smooth both p and q using a convex combination with the uniform distribution parameterized by $\alpha \in [0, 1[$:

$$\phi_{\text{KL}}(p, q) \doteq -\sum_{k \in \mathcal{K}} \hat{p}_k \log(\hat{q}_k), \tag{12}$$

with $\hat{p}_k = \frac{\alpha}{K} + (1 - \alpha) p_k$ and likewise $\hat{q}_k = \frac{\alpha}{K} + (1 - \alpha) q_k$.

This fidelity term is better suited for comparing distributions than the quadratic fidelity. Indeed, it penalizes more heavily the disparity between p and q if p is a confident labeling, meaning that one label dominates the other. On the other hand, if p is such that all classes have similar probabilities, then differences between q and p should be less penalized. This property, illustrated in Fig. 3c, reflects that the observed probability should be most

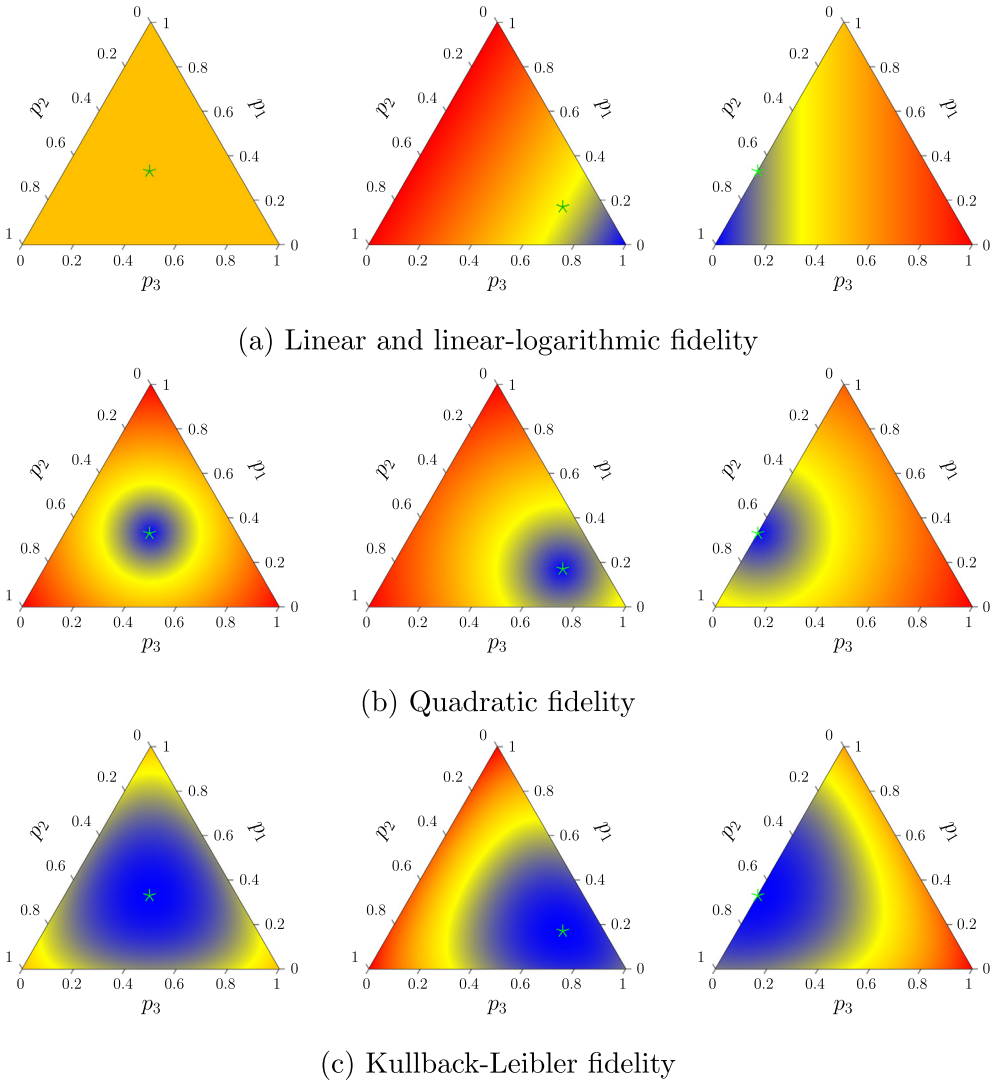


Fig. 3. Surface plot of the fidelity functions over the simplex for $|\mathcal{K}| = 3$. The observed distribution p is represented by a green star ★, while the value of the fidelity function of the corresponding point of the simplex is represented with the following normalized colormap: low high. We remark that the quadratic divergence illustrated in (b) only takes the radial distance into account. On the other hand, the linear fidelities illustrated in (a) are minimal at the simplex corner closer to the observed distribution, or constant if the distribution is uniform, as seen on the top left figure. Finally, the confidence of the observed distribution is taken into account when estimating the Kullback-Leibler fidelity illustrated in (c).

influential when confident, while regularity should be the deciding factor for ambiguous labelings.

4.4. Penalizers inducing spatial regularity

The regularizer $\Psi(\cdot) : \mathcal{S}^n \mapsto \mathbb{R}$ favors solutions of (3) which are spatially smooth, in the sense that most adjacent nodes in the graph G share the same label. In this section, we present two popular spatial regularity-inducing penalizers and their respective properties. For this purpose, we consider the graph $G = (V, E, w)$ defined in Section 4.1. As with the fidelity terms, all penalizers are defined on the convex domain.

We define a global labeling Q as spatially smooth if the number of non-zero values in $\{x_i - x_j | (i,j) \in E\}$ is small compared to the number of edges. Indeed, for such a labeling, most nodes are surrounded by neighbors of the same label. Such a labeling is constant with respect to a partition of G which is *coarse*, i.e. with a number of constant connected components that is small with respect to the number of nodes.

We restrict ourselves to regularizers that factorize over the edges of G, i.e. that can be written under the following form:

$$\Psi(Q) \doteq \sum_{(i,j)\in E} w_{ij} \psi(Q_i - Q_j), \tag{13}$$

with $\psi : \mathbb{R}^{\mathcal{K}} \mapsto \mathbb{R}$ a functional minimal in 0, encouraging spatially smooth solutions. Regularizers of this form were first introduced by Geman and Reynolds (1992), and include many of the most commonly used spatial regularity-inducing penalties.

4.4.1. Potts penalty

Pairwise graphical models such as MRFs and CRFs encode the influence of the context with an interaction potential between adjacent nodes whose value is zero when the labels are identical and strictly non-negative when they are different (Potts, 1952). This translates into choosing $\Omega = \mathbf{S}$ and a functional ψ_{Potts} equal to 0 in 0 and 1 everywhere else:

$$\psi_{\text{Potts}}(q) = \begin{cases} 0 & \text{if } q = 0 \\ 1 & \text{else.} \end{cases} \tag{14}$$

This functional can naturally be extended to the case where $\Omega = \mathcal{S}$. In this case, the *Potts penalty* corresponds to the total weighted cut between constant components of G, and is referred to as the total boundary size.

4.4.2. Total variation

The *graph total variation* (TV) can be seen as a convex relaxation of the above Potts penalty, which, to a certain extent, can also enforce piecewise-constant solutions (Rudin et al., 1992). Its definition depends on the context but usually consists in setting ψ as a norm over $\mathbb{R}^{\mathcal{K}}$.

Over vector spaces, the use of an Euclidean norm is often considered, enjoying theoretical isotropy. Here, however, ψ is applied to the difference between two discrete distributions, for which the notion of isotropy is not relevant. Observe that the use of an Euclidean norm would enforce equality of neighboring distributions only as a whole, that is to say in the solution set either the two distributions are exactly the same, or they differ over each label for which the observation P differs. As a consequence, if two neighboring labelings in P disagree over the probability of one given label, and thus their equality is not favored in the solution set, then equality of the discrete probabilities for the other labels would not be favored either.

Although we do not investigate its practical advantage further, we prefer using an ℓ_1-norm, which is the separable sum of the absolute values of the finite differences, thus favoring the equality of all neighboring discrete probabilities more independently:

$$\psi_{\mathrm{TV}}(q) = \sum_{k \in \mathcal{K}} |q_k|. \tag{15}$$

5. Graph-structured optimization

The choice of a fitting algorithm to minimize objective functionals of the form (3) hinges on the respective properties of the fidelity and regularizing functions as well as the search space. We distinguish three settings, as they necessitate vastly different approaches to be solved:

- combinatorial;
- continuous space and non-convex functional;
- continuous space and convex functional.

5.1. Combinatorial

If the search space $\Omega = \mathbf{S}$ is discrete, the problem is said to be *combinatorial*. The sheer number of combinations and the lack of continuity prevent the retrieval of a global minimizer in general. When considering only two labels however, the objective functional can be solved with graph cuts algorithms due to its submodularity (Boykov et al., 2001).

If the number of labels exceeds two, the functional is no longer submodular and can only be approximately minimized. The α-expansion algorithm introduced by Boykov and Kolmogorov (2004) allows us to approximately solve such problems through a sequence of binary labeling problems, which can in turn be solved efficiently with graph cuts. This algorithm is widely used because of its performance, its theoretical guarantees and the availability of its implementation.

5.2. Continuous space and non-convex functional

We consider the case when $\Omega = \mathcal{S}$ is continuous, but the regularizer is non-convex, typically the Potts penalty extended to $\mathcal{S}$. In this setting, no guarantee on the global optimality can be established, however numerous approximated algorithms exist. A first approach proposed by Ishikawa (2003) is to discretize the search space and to treat the problem as a combinatorial one. A more recent approach proposed by Landrieu and Obozinski (2016b) allows us to keep the continuity of the search space and provides better results with fewer cuts. As this algorithm has only been presented for one-dimensional values, we focus on a natural extension to multi-dimensional, simplex-constrained values in the following.

This greedy algorithm, dubbed ℓ_0-*cut pursuit*, exploits the fact that spatially smooth labelings can be broken down into a small number of constant connected components to accelerate the resolution of the corresponding optimization problem. The ℓ_0-cut pursuit algorithm maintains a current partition of the graph in which the nodes of each component share the same value. This partition is initialized such that all the nodes are in the same component, and is then refined by computing binary partitions, called *optimal binary cuts*, and enumerating their connected components. A backward-step is then performed to check if merging existing adjacent components can decrease the objective function.

However, in (Landrieu and Obozinski, 2016b), ℓ_0-*cut pursuit* is only defined in the one-dimensional setting in which only one scalar value is associated with each node. In this paper, we extend this algorithm to a multi-dimensional setting in which we associate a multi-dimensional, simplex-constrained value to each node. This extension is made easy by the separability hypothesis of the fidelity term, which ensures that, given a partition of V, the associated optimal distribution can be computed independently for each component by minimizing the sum of the associated fidelity terms.

Furthermore, the four fidelity functions listed in Section 4.3 are such that the constant distribution q_A minimizing the sum of fidelity terms for a subset of nodes $A \subset V$ is also simplex-bound, and easy to compute. Indeed, for the linear and linear-logarithmic fidelity, q_A is the hard labeling corresponding to the class maximizing the sum of the distributions associated to the nodes of A. For the quadratic and Kullback-Leibler fidelity, q_A is the average of the distributions of the nodes of A.

Computing such piecewise-constant labelings is the critical step in each of the three main steps of ℓ_0-cut pursuit, namely the computation of the optimal binary cuts and associated optimal distribution, and the backward step. Consequently the extension of ℓ_0-cut pursuit to multi-dimensional, simplex-bound data can be implemented easily and remains very efficient.[3]

5.3. Continuous space and convex functional

In this last setting, on top of the convexity of the search space $\Omega = \mathcal{S}$, we consider a functional ψ (and hence Ψ), which is also convex. In order to favor the sparsity of $\{x_i - x_j | (i,j) \in E\}$, and hence a small number of constant connected components in the set of minimizers, the graph total variation is however non-differentiable.

Given the level of uncertainty over the data and over the parameters of our regularization framework, high precision is not required for the minimization of the objective functional. We thus resort to first-order *proximal splitting algorithms*, well-adapted to such large-scale situations where the functional is a sum of simple terms. This approach has been considered for instance by Lellmann et al. (2009), who use a *Douglas–Rachford splitting algorithm* to solve a specific instance of (3). Since this publication, more powerful splitting schemes have been developed. Although primal-dual schemes are popular thanks to their generality (Chambolle and Pock, 2011), the *preconditioned generalized forward-backward splitting algorithm* (Raguet and Landrieu, 2015) is more suited to our graph-structured problem, while taking full advantage of the

[3] A C++ implementation can be downloaded at www.loiclandrieu.com/.

smoothness of the data-fidelity term. We refer to the latter and references therein for more details.[4]

6. Experimental results

In this section, we first present the involved benchmark datasets (Section 6.1), the considered evaluation metrics (Section 6.2) and the competing methods (Section 6.3). Subsequently, we present and discuss the derived results (Sections 6.4 and 6.5). The experimental framework will be available at www.loiclandrieu.com/.

6.1. Datasets

Since our main goal consists in testing the applicability of the involved methods and the reproducibility of derived results, we want to facilitate an objective comparison to other methodologies. Hence, we test our framework on three publicly available and labeled 3D point cloud datasets which are described in the following subsections.

6.1.1. Oakland-5C Dataset and Oakland-3C Dataset

The *Oakland 3D Point Cloud Dataset*[5] (Munoz et al., 2009) is a labeled benchmark dataset which has often been used to evaluate approaches focusing on a semantic labeling of 3D point clouds. This dataset has been acquired in the vicinity of the CMU campus in Oakland, USA, with a moving platform equipped with a side-looking Sick laser scanner used in push-broom mode (Munoz et al., 2008). During data acquisition, the speed of the platform reached up to 20 km/h, and the acquired 3D point clouds reveal a point density with significant variation. A separation of the dataset into training data (about 37k labeled 3D points) and test data (about 1.32M labeled 3D points) is already provided.

The *Oakland-5C Dataset* refers to the provided reference labeling with respect to five semantic classes. These classes are defined as *Wire*, *Pole/Trunk*, *Façade*, *Ground* and *Vegetation*.

The *Oakland-3C Dataset* refers to the provided reference labeling with respect to three structural classes. These classes are defined as *Linear Structures*, *Planar Structures* and *Volumetric Structures*.

For both the Oakland-5C Dataset and the Oakland-3C Dataset, the number of training examples per class is very unbalanced which can have a detrimental effect on the training process (Chen et al., 2004; Criminisi and Shotton, 2013). To avoid such effects, we introduce a class re-balancing which relies on randomly selecting $1,000$ labeled 3D points per class as new training set and discarding all other points (Weinmann et al., 2015a).

6.1.2. Paris-rue-Cassette Database

To include larger MLS datasets in our experiments, we also make use of the *Paris-rue-Cassette Database*[6] (Vallet et al., 2014), a point cloud dataset which has been acquired in January 2013 with the mobile laser scanning system called *Stereopolis II* (Paparoditis et al., 2012). This system involves two plane sweep lidars of type *Riegl LMS-Q120i* and a 3D lidar of type *Velodyne HDL-64E* to capture the local 3D geometry of the scene. The Riegl devices are placed on each side of the vehicle and serve for observing the building façades with a centimeter accuracy, whereas the Velodyne device mainly serves for observing the bottom part in between. In total, the dataset contains 12M points corresponding to a street section with a length of approximately 200 m as well as a reference labeling which includes both pointwise labels and segmented objects. The annotation has been carried out by recovering a regular 2D topology for the point cloud stream during data acquisition and an offline human interaction via a graph editing tool based on standard 2D image segmentation techniques (Brédif et al., 2014).

In our experiments, we consider the seven dominant classes defined as *Façade*, *Ground*, *Cars*, *2-Wheelers*, *Road Inventory*, *Pedestrians* and *Vegetation*. All 3D points belonging to the other classes are removed as these classes are not considered as representative (Weinmann et al., 2015c). To separate the dataset into training data and test data, we randomly select $1,000$ labeled 3D points per class as training set and all remaining labeled 3D points as test set.

6.2. Evaluation metrics

To evaluate the performance of the benchmarked approaches, we compare the derived labeling to the reference labeling on a per-point basis. For this purpose, we consider both classwise and global evaluation metrics. The classwise evaluation metrics are represented by recall (R), precision (P) and F_1-score. Whereas recall represents a measure of completeness or quantity, precision represents a measure of exactness or quality. The F_1-score is a compound metric combining precision and recall with equal weights. The global evaluation metrics are represented by overall accuracy (OA) and the unweighted average of the F_1-score over all classes ($\overline{F}_1$). In this regard, it should be taken into account that a consideration of the overall accuracy might not be sufficient if the number of examples per class is very inhomogeneous for the test data. The indicator $\overline{F}_1$ allows judging about the quality of classification results based on classwise evaluation metrics.

As stated in Section 4.2, the advantage of probabilistic labelings is that their certainty can be estimated. To each point-level assignment, we associate a certainty measure by computing its entropy H. A low entropy designates a high confidence assignment (for example $H([1\ 0\ 0]) = 0$), while a high entropy denotes an ambiguous assignment (for example $H([\frac{1}{3}\ \frac{1}{3}\ \frac{1}{3}]) = \log(3)$). We define the partial assignment at coverage $f\%$ as the fraction of an assignment P when only considering the $f\%$ lowest entropy pointwise assignments, i.e. only the most confident points. We can evaluate the $\overline{F}_1$-score of such a partial assignment by comparing it against the corresponding partial ground truth.

To demonstrate the benefit of this confidence assignment, we provide the *accuracy/coverage plots* of the best performing methods in Fig. 4. Those plots are obtained by sorting the points by increasing entropy, and computing the accuracy of the partial assignment for coverage going from 70% to full coverage.

6.3. Competing methods

In this subsection, we briefly summarize the benchmarked algorithms (i.e. the considered configurations of our framework) and some state-of-the-art methods that are involved for comparison. In Section 4, we listed two search spaces ($\Omega = \mathcal{S}$ or $\mathbf{S}$), four fidelity functions (*linear*, *linear-logarithmic*, *quadratic* and *Kullback-Leibler*), as well as the two regularizers (*Potts penalty*, *total variation*). Of the 16 possible combinations, only 8 are unique and relevant. Indeed, when considering fidelity functions that induce a hard assignment (namely *linear* and *linear-logarithmic*) with the Potts penalty, the choice of Ω is irrelevant as all solutions belong to $\mathbf{S}$. Consequently, ℓ_0-cut pursuit and α-expansion will minimize the same functional. As both approaches approximate the global solution, one could expect different results. However, in all our numerical experiments, the difference in the final value of the functional was small enough that its corresponding assignment was almost identical.

[4] Efficient implementations in C++, interfaced with MEX for MATLAB/GNU Octave users, can be found at https://www.ceremade.dauphine.fr/raguet/pgfb/.

[5] The Oakland 3D Point Cloud Dataset is publicly available at http://www.cs.cmu.edu/vmr/datasets/oakland_3d/cvpr09/doc/ (last access: 17 November 2016).

[6] The Paris-rue-Cassette Database is publicly available at http://data.ign.fr/benchmarks/UrbanAnalysis/ (last access: 17 November 2016).

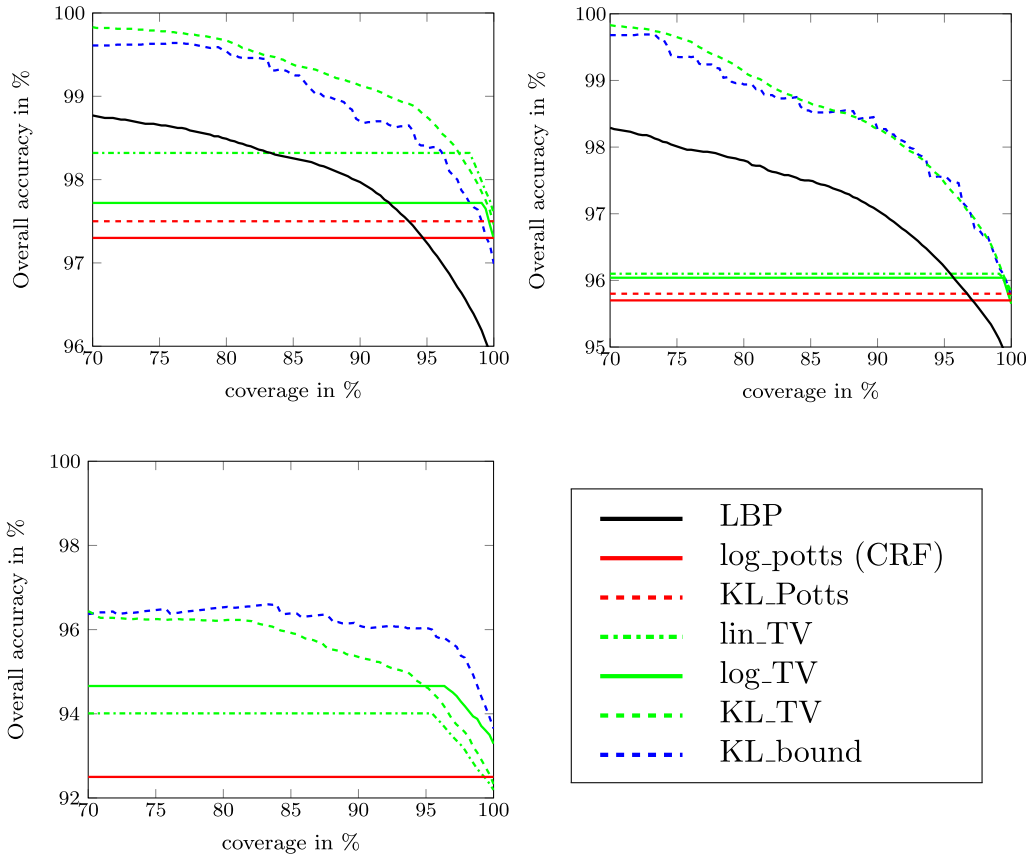


Fig. 4. Accuracy/coverage plot for the three datasets and the top performing methods from 70% to 100% coverage for the Oakland-3C Dataset (top left), the Oakland-5C Dataset (top right) and the Paris-rue-Cassette Database (bottom left).

Some simple calculus shows that, for the Potts penalty, the values of the linear, quadratic and Kullback-Leibler fidelities at the corners of the simplex amount to the same penalty with a different regularization strength. Similarly, we do not consider the total variation regularizer with a discrete search space, as it is redundant with the Potts penalty. We list the 8 combinations of regularizers and fidelity functions that correspond to unique algorithms in Table 1.

To compare with the approach advocated by Niemeyer et al. (2014) and based on a graphical model in the form of a CRF, we compute the solutions provided by loopy belief propagation, both for the marginal inference (LBP) and MAP-inference (LBP_MAP). The MAP-inference can also be advantageously computed with α-expansion, as mentioned by Landrieu et al. (2017), and corresponds to the log_potts shorthand in Table 1. We also considered alternative message-passing algorithms presented in literature on graphical models. In particular, the inference scheme based on the mean-field approximation (Xing et al., 2003) yields very similar results to LBP, and is not represented here for the sake of clarity. The tree-reweighted variant of the belief propagation algorithm (Wainwright et al., 2003) was not able to produce results in a reasonable time, as the spanning tree search turned out to be very costly. The implementations of the inference algorithms were obtained at (Schmidt, 2012).

The regularizing approach proposed by Lellmann et al. (2009) for image labeling corresponds to lin_TV. Since our minimization problem is strongly convex, the choice of the solving algorithm does not influence the solution. Consequently, we use the preconditioned generalized forward-backward splitting algorithm (PGFB) algorithm which is much faster.

6.4. Experimental results

In Table 2, we provide the results of the full classifications for the methods invoked in Section 6.3, and the accuracy/coverage plot is represented in Fig. 4. The classwise results are displayed in Tables 3–5. The computation time, referenced in Table 6, is mainly dependent on the chosen regularizer as it dictates the used algorithm.

To obtain a visual impression about the quality of the derived classification results, a visualization of the classified point clouds is provided for the Oakland-3C Dataset in Fig. 5, for the Oakland-5C Dataset in Fig. 6 and for the Paris-rue-Cassette Database in Fig. 1. All these figures contain an illustration of the ground truth labeling, the initial labeling derived via pointwise classification, the labeling derived via structured regularization relying on the KL_bound method and the confidence of the derived labeling.

6.5. Discussion

From Table 2, we can observe that regularization does indeed improve both the accuracy and the $\overline{F}_1$-score of the pointwise classification. We observe that message-passing algorithms such as LBP and LBP_MAP underperform when compared to other approaches, while their computation time is among the highest. This is in accordance with the observations made by Landrieu et al. (2017).

In Tables 3–5, we observe that although our framework barely improves the F_1-score of easy-to-classify classes such as *Ground* or *Façade*, our methods display significant improvement over hard

Table 1
List of the benchmarked algorithms with their characteristics.

Method	Fidelity	Regularizer	Domain	Minimizing algorithm
lin_potts	Linear	Potts penalty	$\mathbf{S}$	α-expansion
log_potts	Linear-logarithmic	Potts penalty	$\mathbf{S}$	α-expansion
lin_TV	Linear	Total variation	$\mathcal{S}$	PGFB
log_TV	Linear-logarithmic	Total variation	$\mathcal{S}$	PGFB
l22_TV	Quadratic	Total variation	$\mathcal{S}$	PGFB
KL_TV	Kullback-Leibler	Total variation	$\mathcal{S}$	PGFB
l22_bound	Quadratic	Potts penalty	$\mathcal{S}$	ℓ_0-cut pursuit
KL_bound	Kullback-Leibler	Potts penalty	$\mathcal{S}$	ℓ_0-cut pursuit

Table 2
Classification results (in %) derived with the considered methods for the three datasets. OA is the overall accuracy and $\overline{F}_1$ is the unweighted average of the F_1-scores over all classes. Bold values indicate the highest rates achieved for the respective dataset.

Method	Oakland-3C Dataset		Oakland-5C Dataset		Paris-rue-Cassette Database	
	OA	$\overline{F}_1$	OA	$\overline{F}_1$	OA	$\overline{F}_1$
Pointwise	93.8	71.4	92.3	63.5	81.0	41.0
LBP	95.8	75.6	94.7	70.9	83.2	44.3
LBP_MAP	95.5	74.9	94.5	69.8	82.6	43.6
log_potts	97.3	78.2	95.7	74.4	92.5	65.4
lin_potts	**97.5**	**78.9**	95.8	75.1	92.3	61.3
l22_bound	97.2	78.3	95.7	73.6	93.6	**65.6**
KL_bound	97.0	78.3	95.7	74.7	**93.7**	64.4
lin_TV	**97.5**	78.7	95.7	74.9	92.4	61.2
log_TV	97.3	78.2	95.7	74.4	93.1	60.9
l22_TV	**97.5**	78.7	95.7	74.8	92.4	61.2
KL_TV	**97.5**	78.7	**95.9**	**75.3**	91.4	61.2

Table 3
Classification results (in %) for the Oakland-3C Dataset. We present the overall accuracy (OA), the unweighted average of the F_1-score over all classes ($\overline{F}_1$), and the classwise F_1-scores. Bold values indicate the highest rates achieved for this dataset.

Method	OA	$\overline{F}_1$	F_1(*Linear Structures*)	F_1(*Planar Structures*)	F_1(*Volumetric Structures*)
Pointwise	93.8	71.4	28.7	97.1	88.5
LBP	95.8	75.6	36.2	98.1	92.6
LBP_MAP	95.5	74.9	34.8	98.0	91.9
log_potts	97.3	78.2	40.1	98.8	95.6
lin_potts	**97.5**	**78.9**	**41.6**	98.9	96.0
l22_bound	97.2	78.3	40.5	98.8	95.5
KL_bound	97.0	78.3	41.1	98.7	95.0
lin_TV	**97.5**	78.7	41.0	98.9	96.0
log_TV	97.3	78.2	40.0	98.9	95.7
l22_TV	**97.5**	78.7	41.0	98.9	96.1
KL_TV	**97.5**	78.7	40.9	**99.0**	**96.2**

Table 4
Classification results (in %) for the Oakland-5C Dataset. We present the overall accuracy (OA), the unweighted average of the F_1-score over all classes ($\overline{F}_1$), and the classwise F_1-scores. Bold values indicate the highest rates achieved for this dataset.

Method	OA	$\overline{F}_1$	F_1(*Wire*)	F_1(*Pole/Trunk*)	F_1(*Façade*)	F_1(*Ground*)	F_1(*Vegetation*)
Pointwise	92.3	63.5	18.2	38.0	76.3	97.6	87.2
LBP	94.7	70.9	22.2	59.8	81.2	98.0	93.4
LBP_MAP	94.5	69.8	21.2	55.6	81.0	98.0	93.0
lin_potts	95.8	75.1	28.2	**70.0**	83.3	**98.2**	95.8
log_potts	95.7	74.4	29.1	65.7	**83.4**	**98.2**	95.7
l22_bound	95.7	73.6	26.5	64.7	83.1	98.1	95.9
KL_bound	95.7	74.7	**29.4**	67.3	82.9	**98.2**	95.7
lin_TV	95.7	74.9	27.5	69.3	**83.4**	98.1	95.9
log_TV	95.7	74.4	28.4	66.5	83.1	98.1	95.7
l22_TV	95.7	74.8	27.8	68.9	83.2	98.1	95.8
KL_TV	**95.9**	**75.3**	29.1	69.6	83.2	98.1	**96.3**

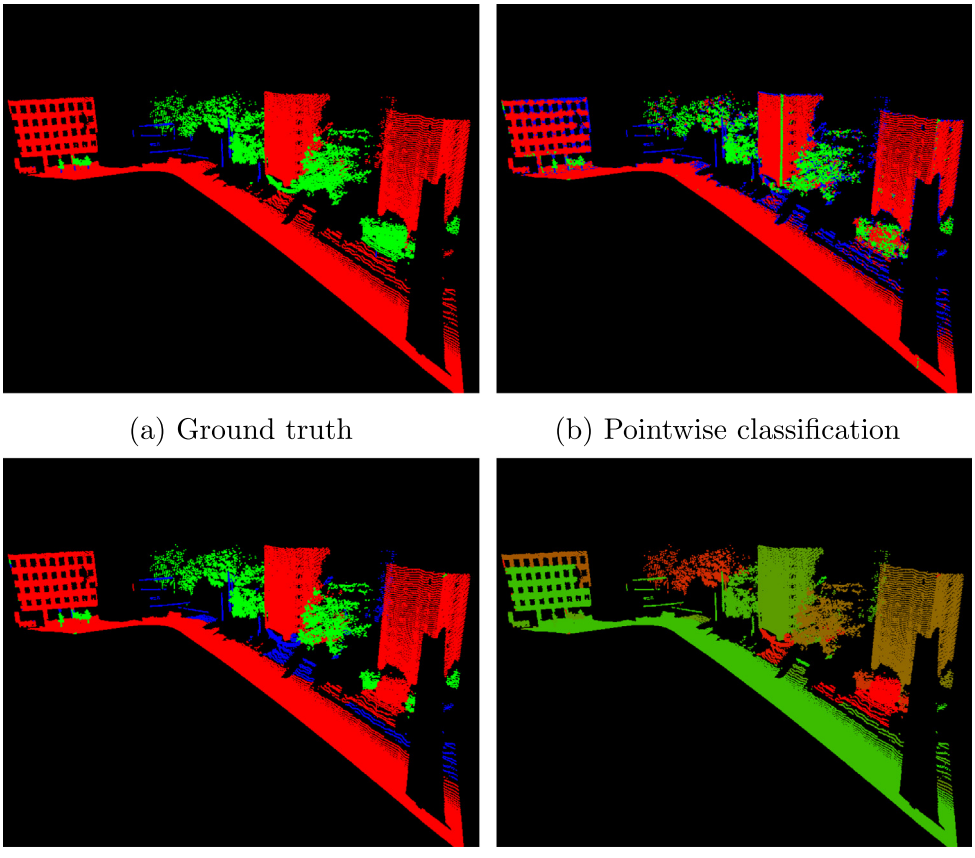
Table 5
Classification results (in %) for the Paris-rue-Cassette Database with 7 classes represented by *Façade* (F), *Ground* (G), *Cars* (C), *2-Wheelers* (2 W), *Road Inventory* (RI), *Pedestrians* (P) and *Vegetation* (V). We present the overall accuracy (OA), the unweighted average of the F_1-score over all classes ($\overline{F}_1$), and the classwise F_1-scores. Bold values indicate the highest rates achieved for this dataset.

Method	OA	$\overline{F}_1$	F_1(F)	F_1(G)	F_1(C)	F_1(2 W)	F_1(RI)	F_1(P)	F_1(V)
Pointwise	81.0	41.0	85.4	96.8	45.4	10.6	10.7	5.0	33.3
LBP	83.2	44.3	87.0	97.6	55.5	14.3	12.8	6.4	36.4
LBP_MAP	82.6	43.6	86.6	97.4	54.3	13.9	12.7	5.6	35.0
log_potts	92.5	65.4	94.7	95.1	82.2	48.7	16.5	**65.9**	**54.9**
lin_potts	92.3	61.3	94.5	95.8	78.9	48.1	17.7	43.7	50.6
l22_bound	93.6	**65.6**	**95.7**	97.4	**83.5**	**66.7**	19.5	46.6	50.0
KL_bound	**93.7**	64.4	**95.7**	98.1	82.4	46.9	**32.7**	44.1	51.2
lin_TV	92.4	61.2	94.4	97.4	81.8	42.4	28.7	35.7	48.0
log_TV	93.1	60.9	95.0	**98.2**	82.9	41.2	29.8	29.2	50.1
l22_TV	92.4	61.2	94.4	96.5	80.5	44.4	23.1	38.9	50.5
KL_TV	91.4	61.2	93.7	93.9	76.4	48.3	18.6	43.8	53.6

Table 6
Required time in seconds for solving the optimization problem depending on the dataset and the chosen algorithm on an i7-4790 CPU 3.60 GHz with 8 GB of RAM. Bold values indicate the most efficient solution for the respective dataset.

Method	Oakland-3C Dataset (1.3M points)	Oakland-5C Dataset (1.3M points)	Paris-rue-Cassette Database (12M points)
LBP	35	45	720
LBP_MAP	31	54	800
α-expansion	**15**	**24**	**400**
ℓ_0-cut pursuit	21	27	600
PGFB	43	72	2100

classes, such as *Wire*, *Pole/Trunk*, *Cars*, *2-Wheelers* and *Pedestrians*. Indeed, the F_1-score of *Wire* (from 18.2% to 29.4%) and *Pole/Trunk* (from 38.0% to 70.0%) is almost doubled for the Oakland-5C Dataset for the best performing methods. More impressively, for the Paris-rue-Cassette Database, our methods are able to retrieve decent classifications for classes that were mostly mislabeled by the pointwise labeling. For example, *2-Wheelers* were initially classified with a 10.6% F_1-score, but the best regularized labeling boasts a classification score of 66.7%. Likewise, regularization was able to improve the classification of the class *Pedestrians* from 5.0% to 65.9%, and the classification of the class *Cars* from 45.4% to 83.5%.



(a) Ground truth (b) Pointwise classification

(c) Regularized classification (d) Confidence map

Fig. 5. Visualization of a 3D point cloud labeling for a part of the Oakland-3C Dataset. In (a), (b), and (c), the color encoding addresses the classes *Linear Structures* (blue), *Planar Structures* (red) and *Volumetric Structures* (green). In (d), the confidence is represented from green to red: confident uncertain. Remark that misclassifications in (c) correspond to the least confident areas in (d).

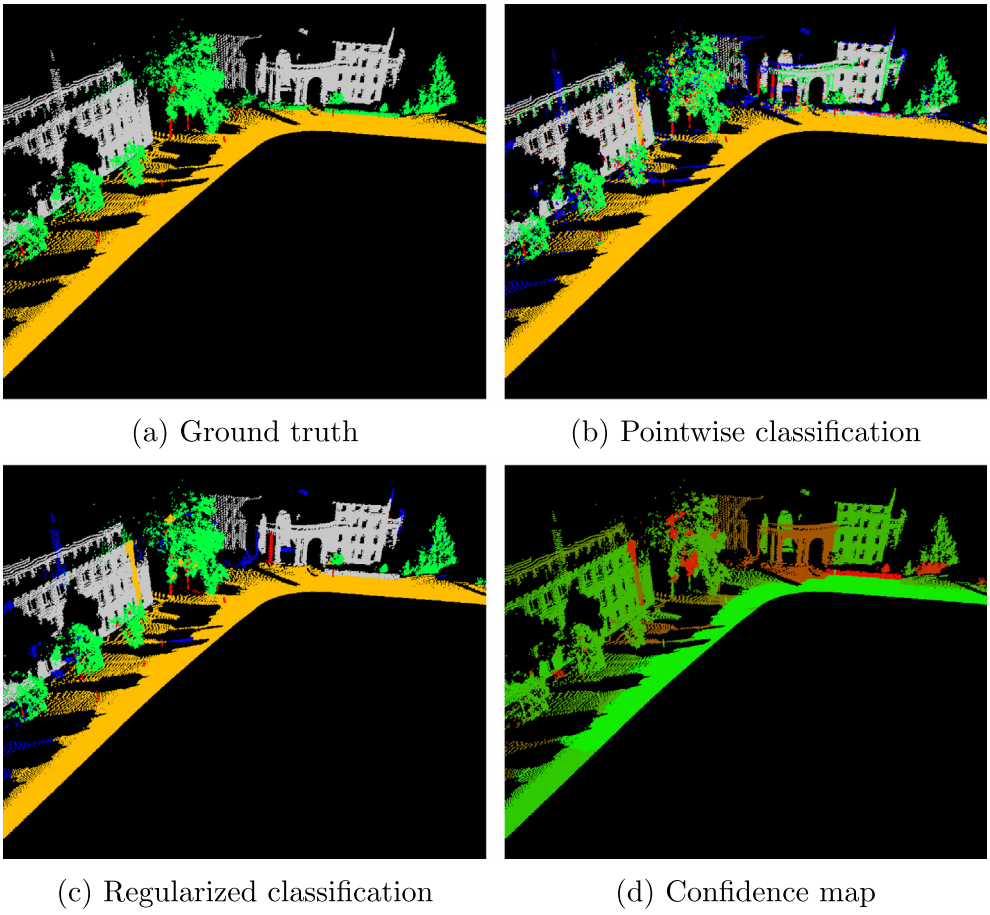


Fig. 6. Visualization of a 3D point cloud labeling for a part of the Oakland-5C Dataset. In (a), (b), and (c), the color encoding addresses the classes *Wire* (blue), *Pole*/*Trunk* (red), *Façade* (gray), *Ground* (orange) and *Vegetation* (green). In (d), the confidence is represented from green to red: confident uncertain. Remark that misclassifications in (c) correspond to the least confident areas in (d).

We explain this large improvement over hard classes with the fact that regularization removes isolated misclassified points scattered over the point cloud. Those classes are also the least represented, their recall is particularly sensitive to such misclassifications. The regularization increases the precision as well by enforcing homogeneity of a tightly connected set of points, which often belong to the same class. Remark that this will only improve the classification if the initial labeling is mostly right to begin with.

Among the methods implemented in our framework, the difference of performance is rather small, with a difference of less than 1% in accuracy. The benefits in choosing a given configuration lies elsewhere, namely in its computational load and the nature of the obtained smoothed assignment.

In addition, we remark that some classes are much harder to correctly classify than others because of systematic issues. For instance, for the Oakland-3C and Oakland-5C Datasets, the class *Wire* has a significantly lower classification score compared to other classes. This can be explained by the nature of the lidar data acquisition. Indeed, linear structures have a lower chance of triggering echoes, resulting in a sparser acquisition. Furthermore, since the angle between the scanlines is constant for a TLS/MLS system, the actual distance between the scanlines increases with the distance. As a consequence, the acquisition of rather distant façades tends to look like a collection of parallel linear structures. This problem is inherent to the acquisition and the pointwise classification methods, yet it is beyond the scope of this paper. However, we can notice that the regularization improves the quality of the prediction for such classes significantly (over 50% increase with respect to the F_1-score referring to the initial labeling).

A closer look at the visualizations of derived results reveals that the initial labeling derived via pointwise classification (cf. Figs. 1b, 5b, 6b) contains many misclassified points in those regions of the point cloud, which are later characterized by a rather low confidence when using the KL_bound method (cf. Figs. 1d, 5d, 6d), while the labeling itself is spatially regular and more appropriate in almost all cases (cf. Figs. 1c, 5c, 6c). In particular, the spatially regular labeling remains uncertain in regions of the point cloud, where labels of several classes occur rather often after the initial labeling and with no clearly dominant label. For this reason, the confidence is for instance lower for one of the building façades of the Oakland-3C Dataset (cf. upper left part of Fig. 5d), lower for one of the building façades of the Oakland-5C Dataset (cf. upper part of Fig. 6d) and for most of the cars and pedestrians of the Paris-rue-Cassette Database (cf. Fig. 1d). As the degree to which isolated misclassified points occur varies significantly for different parts of the point cloud, we can for instance observe that the assigned confidence is different for the trees of the Oakland-3C Dataset (cf. Fig. 5d).

6.5.1. Choosing the fidelity

When combined with the same penalizer, the influence of the fidelity function seems limited in terms of accuracy. However, this choice influences qualitative properties of the solution.

For example, when combined with either the TV or boundary penalty, the linear and linear-logarithmic fidelity yield hard smoothed assignments, while the quadratic and KL fidelity yield probabilistic assignments. Both the linear-logarithmic and KL fidelity involve entrywise logarithms of probabilities, which can induce numerical issues, and consequently require a supplemen-

tary smoothing parameter. However, a smoothing parameter of $\alpha = 0.05$ seems to yield good results in general, and does not require extensive cross-validation. Finally, both quadratic and linear fidelities have fewer parameters and easier computation since both their gradient and proximal operator are very straightforward to obtain.

The linear-logarithmic fidelity takes into account the observed probability in a non-linear way, penalizing assignments with low probability much more than it favors assignments with high probability. This penalty should hence be preferred when considering hard assignments. Conversely, the KL fidelity takes the observed probability into account linearly, while penalizing strong confidence outputs. This penalty should hence be preferred when a probabilistic output is expected.

The choice of the fidelity should in general be cross-validated as a meta-parameter. However, depending on the nature of the expected output, this choice can be restricted.

- **Hard assignment expected:** When only the 100% coverage is relevant for the application, both linear and linear-logarithmic fidelity can be employed.
- **Soft assignment expected:** The quadratic and KL fidelity provide a probabilistic output when combined with the TV or boundary penalty.

6.5.2. Choosing the penalty

The influence of the penalty is more drastic, as it impacts both the nature of the output and the computational efficiency as well. The Potts penalty combined with the α-expansion algorithm is the fastest of all approaches. In terms of accuracy and $\overline{F}_1$-score, it offers excellent performances as well, making it a solid choice when only a hard assignment is expected. The boundary penalty, when solved with the ℓ_0-cut pursuit algorithm, is slightly slower to compute, for comparable performance in terms of classification. However, when combined with the quadratic or KL fidelity, it provides a probabilistic classification as output which allows us to evaluate the confidence of each assignment. Finally, the TV penalty reaches excellent performance, with a probabilistic output as well.

However, solving the convex problem takes more time than the other approaches. We observe that, when combined with linear or linear-logarithmic fidelity, most of the assignments of the regularized solution lie within a corner of the simplex, in accordance with the principles of linear programming. In this case, the probabilistic nature of the solution cannot be exploited beyond removing a small proportion (about 2–4%) of uncertain points. Furthermore, the accuracy of assignments obtained with the TV regularizer seems to evolve smoothly and monotonically with respect to the coverage. On the other hand, the boundary size seems to induce an accuracy-coverage plot which is more subject to sharp breaks and irregularities. This can be explained by the non-convexity of the associated objective function, and the greedy nature of its solving algorithm.

As for the choice of the fidelity function, this choice can be cross-validated as a meta-parameter, as the effect of regularization can vary depending on the dataset and the quality of the initial labeling. However, a general guideline is presented in the following:

- **Hard assignment expected:** When only the 100% coverage is relevant for the application, the Potts penalty should it be used, with α-expansion as a solver, for its speed and the quality of its output.
- **Soft assignment expected:**
  - **Speed is the priority:** In this case, the boundary size penalty combined with the ℓ_0-cut pursuit algorithm is more advantageous.
  - **Quality is the priority:** The total variation penalty offers excellent precision, both for partial and complete coverage, at the price of a longer computation time.

6.5.3. Extension

It is important to note that the cut pursuit algorithm cannot handle different values for the transition between classes. If such a transition matrix can be either inferred or cross-validated, then the other penalty shall always be favored.

The only drawback of the TV penalty in our application is its speed. However, this issue could be addressed by adapting the cut pursuit algorithm to multi-dimensional simplex-constrained values. In this case, the TV penalty would combine the benefits of both the Potts penalty and the boundary penalty, while retaining the robustness associated with its convex nature.

We did not benchmark the effect of choosing a different adjacency tree structure, nor the different weighting schemes that can be applied. When using varying edge weights with the TV penalty, preconditioning strategies such as the one used by PGFB are an absolute must to avoid drastic convergence speed increase.

6.5.4. Comparison to other approaches

In comparison to other approaches for semantically labeling 3D point clouds, our methodology has several strengths. Due to the straightforward extension of the works presented by Weinmann et al. (2015a) in terms of additionally considering structured regularization, the achieved classification results reveal an improved spatial regularity, while the quantitative evaluation reveals an improvement of several % in OA and in the classwise evaluation metrics. Furthermore, we compare the results achieved with our methodology to the ones obtained by a recent approach in which the considered point cloud is projected to an elevation image and, subsequently, image processing techniques relying on mathematical morphology are applied in combination with machine learning techniques (Serna and Marcotegui, 2014). While the qualitative evaluation of that approach reveals a rather high quality of the classification results derived for the Paris-rue-Cassette Database (Vallet et al., 2015), a tendency to over-detection with unnatural segment boundaries becomes visible. In contrast, our methodology produces smoother object boundaries, as the regularization function penalizes the length of the interfaces between objects with different labels. A closer look reveals that the derived boundaries are better which is due to the fact that we use spatial regularizing penalties, and due to the fact that we regularize a semantic assignment derived via point-based classification, which means that the boundaries between classes tend to be the boundaries between objects. Currently, the only approach delivering classification results of higher quality (with $OA = 95.4\%$ and $\overline{F}_1 = 70.2\%$) for the Paris-rue-Cassette Database is the approach presented by Hackel et al. (2016b) that can be seen as fast multi-scale extension of the approach presented by Weinmann et al. (2015a). Due to the consideration of multi-scale neighborhoods, contextual information across different scales is considered in the respectively extracted features, resulting in improved performance. Yet, we point out that this is a limitation of the method used to obtain the initial labeling, while the proposed regularization framework only focuses on obtaining an improved labeling with increased spatial smoothness. Therefore, the regularization framework could for instance also be applied to further improve the results presented by Hackel et al. (2016b).

When directly comparing the results achieved for the Oakland-5C Dataset with the ones reported for other contextual classification approaches, one can observe that our results are slightly better than standard CRF-based classification results (Weinmann et al., 2015b), where an overall accuracy between 94.2% and 95.5% was

achieved. The classification results derived by Munoz et al. (2009) with associative Markov networks for very similar training and test datasets are slightly better in terms of overall accuracy for which values between 95.7% and 97.2% were reported for the three presented approaches. However, our methods better address hard-to-retrieve classes such as *Wire* and *Pole*/*Trunk*.

With a proper parameterization, our method even allows to retain the probabilistic nature of the original classification. To the best of our knowledge, competing methods only provide overall accuracy for 100% coverage, which prevents us from comparing the accuracy-coverage behavior to the one of other algorithms. Our code is available at www.loiclandrieu.com/code, allowing future works to easily use our framework as a post-processing unit and evaluate potential improvements.

7. Conclusions

In this paper, we presented a regularization framework based on structured optimization to smooth semantic labelings on 3D point clouds. We demonstrated that this approach is superior to the classically used belief propagation algorithm. Furthermore, we presented a family of regularizers and fidelity functions which allows to retain the probabilistic nature of the labeling after smoothing, allowing us to estimate its confidence at each point. We also presented an efficient algorithm to solve the subsequent optimization problem, and extended the existing ℓ_0-cut pursuit algorithm to our multi-dimensional, simplex-constrained setting.

Besides different extensions of our regularization framework, we also intend to investigate the potential of Convolutional Neural Networks (CNNs) adapted to 3D data. Among different strategies, particularly the one involving a 3D-CNN to classify each 3D point of a point cloud by considering a voxel-occupancy grid corresponding to the respective local neighborhood seems to be promising (Savinov, 2017; Hackel et al., 2016a; Huang and You, 2016). However, 3D-CNNs typically require a large amount of training data, and the network architecture as well as its internal settings are often heuristically defined by the user. The framework presented in the scope of this paper provides a competitive baseline for such approaches, as it represents an important alternative which is given by a theoretically well-founded structured regularization delivering classification results of high quality at a lighter computational cost and also for scenarios, where only smaller amounts of training data are available. Furthermore, although labelings obtained with CNNs tend to be spatially smoother than pointwise labelings, the degree of spatial regularity depends on the width of the convolutional filters, and hence is not easily tunable. Because our framework is agnostic with respect to how the initial labeling is obtained, it could be used to precisely set the level of smoothness in post-processing, at a light computational cost.

References

Aström, F., Petra, S., Schmitzer, B., Schnörr, C., 2016. Image labeling by assignment. arXiv:1603.05285, pp. 1–9.

Belton, D., Lichti, D.D., 2006. Classification and segmentation of terrestrial laser scanner point clouds using local variance information. Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. XXXVI-5, 44–49.

Blomley, R., Jutzi, B., Weinmann, M., 2016. Classification of airborne laser scanning data using geometric multi-scale features and different neighbourhood types. ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci. III-3, 169–176.

Boulch, A., Le Saux, B., Audebert, N., 2017. Unstructured point cloud semantic labeling using deep segmentation networks. In: Proceedings of the Eurographics Workshop on 3D Object Retrieval, The Eurographics Association, Lyon, France, 23–34 April, pp. 17–24.

Boykov, Y., Kolmogorov, V., 2004. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. IEEE Trans. Pattern Anal. Mach. Intell. 26 (9), 1124–1137.

Boykov, Y., Veksler, O., Zabih, R., 2001. Fast approximate energy minimization via graph cuts. IEEE Trans. Pattern Anal. Mach. Intell. 23 (11), 1222–1239.

Brédif, M., Vallet, B., Serna, A., Marcotegui, B., Paparoditis, N., 2014. TerraMobilita/IQmulus urban point cloud classification benchmark. In: Proceedings of the IQmulus Workshop on Processing Large Geospatial Data, Cardiff, UK, 8 July, pp. 1–6.

Breiman, L., 1996. Bagging predictors. Mach. Learn. 24 (2), 123–140.

Breiman, L., 2001. Random forests. Mach. Learn. 45 (1), 5–32.

Brodu, N., Lague, D., 2012. 3d terrestrial lidar data classification of complex natural scenes using a multi-scale dimensionality criterion: applications in geomorphology. ISPRS J. Photogramm. Remote Sens. 68, 121–134.

Chambolle, A., Pock, T., 2011. A first-order primal-dual algorithm for convex problems with applications to imaging. J. Math. Imaging Vis. 40 (1), 120–145.

Chehata, N., Guo, L., Mallet, C., 2009. Airborne lidar feature selection for urban classification using random forests. Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. XXXVIII-3/W8, 207–212.

Chen, C., Liaw, A., Breiman, L., 2004. Using random forest to learn imbalanced data. Technical report, University of California, Berkeley, CA, USA.

Clode, S., Kootsookos, P.J., Rottensteiner, F., 2004. The automatic extraction of roads from lidar data. Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. XXXV-B3, 231–236.

Criminisi, A., Shotton, J., 2013. Decision forests for computer vision and medical image analysis. In: Advances in Computer Vision and Pattern Recognition. Springer, London, UK.

Demantké, J., Mallet, C., David, N., Vallet, B., 2011. Dimensionality based scale selection in 3d lidar point clouds. Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. XXXVIII-5/W12, 97–102.

Filin, S., Pfeifer, N., 2005. Neighborhood systems for airborne laser data. Photogramm. Eng. Remote Sens. 71 (6), 743–755.

Geman, D., Reynolds, G., 1992. Constrained restoration and the recovery of discontinuities. IEEE Trans. Pattern Anal. Mach. Intell. 14 (3), 367–383.

Gevaert, C.M., Persello, C., Vosselman, G., 2016. Optimizing multiple kernel learning for the classification of UAV data. Remote Sens. 8 (12), 1–22.

Guignard, S., Landrieu, L., 2017. Weakly supervised segmentation-aided classification of urban scenes from 3d lidar point clouds. Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. XLII-1/W1, 151–157.

Guo, B., Huang, X., Zhang, F., Sohn, G., 2015. Classification of airborne laser scanning data using JointBoost. ISPRS J. Photogramm. Remote Sens. 100, 71–83.

Hackel, T., Savinov, N., Ladicky, L., Wegner, J.D., Schindler, K., Pollefeys, M., 2016a. Large-scale point cloud classification benchmark. <http://www.semantic3d.net>.

Hackel, T., Wegner, J.D., Schindler, K., 2016b. Fast semantic segmentation of 3d point clouds with strongly varying density. ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci. III-3, 177–184.

Hornung, A., Wurm, K.M., Bennewitz, M., Stachniss, C., Burgard, W., 2013. Octomap: an efficient probabilistic 3d mapping framework based on octrees. Autonom. Robots 34 (3), 189–206.

Hu, H., Munoz, D., Bagnell, J.A., Hebert, M., 2013. Efficient 3-d scene analysis from streaming data. In: Proceedings of the IEEE International Conference on Robotics and Automation, IEEE, Karlsruhe, Germany, 6–10 May, pp. 2297–2304.

Huang, J., You, S., 2016. Point cloud labeling using 3d convolutional neural network. In: Proceedings of the International Conference on Pattern Recognition, IEEE, Cancun, Mexico, 4–8 December, pp. 2670–2675.

Huber, P.J., 1964. Robust estimation of a location parameter. Ann. Math. Stat. 35 (1), 73–101.

Indyk, P., Motwani, R., 1998. Approximate nearest neighbors: towards removing the curse of dimensionality. In: Proceedings of the 30th Annual ACM Symposium on Theory of Computing. ACM, Dallas, TX, USA. 24–26 May, pp. 604–613.

Ishikawa, H., 2003. Exact optimization for Markov random fields with convex priors. IEEE Trans. Pattern Anal. Mach. Intell. 25 (10), 1333–1336.

Jing, F., Li, M., Zhang, H.J., Zhang, B., 2004. Entropy-based active learning with support vector machines for content-based image retrieval. In: Proceedings of the IEEE International Conference on Multimedia and Expo, IEEE, Taipei, Taiwan, 27–30 June, pp. 85–88.

Khoshelham, K., Oude Elberink, S.J., 2012. Role of dimensionality reduction in segment-based classification of damaged building roofs in airborne laser scanning data. In: Proceedings of the International Conference on Geographic Object-Based Image Analysis, Rio de Janeiro, Brazil, 7–9 May, pp. 372–377.

Kullback, S., Leibler, R.A., 1951. On information and sufficiency. Ann. Math. Stat. 22 (1), 79–86.

Lafarge, F., Mallet, C., 2012. Creating large-scale city models from 3d-point clouds: a robust approach with hybrid representation. Int. J. Comput. Vision 99 (1), 69–85.

Lalonde, J.F., Unnikrishnan, R., Vandapel, N., Hebert, M., 2005. Scale selection for classification of point-sampled 3d surfaces. In: Proceedings of the International Conference on 3-D Digital Imaging and Modeling, IEEE, Ottawa, Canada, 13–16 June, pp. 285–292.

Landrieu, L., Mallet, C., Weinmann, M., 2017. Comparison of belief propagation and graph-cut approaches for contextual classification of 3d lidar point cloud data. In: Proceedings of the IEEE International Geoscience and Remote Sensing Symposium, IEEE, Fort Worth, TX, USA, 23–28 July, pp. 1–4.

Landrieu, L., Obozinski, G., 2016a. Cut pursuit: fast algorithms to learn piecewise constant functions. In: Proceedings of the International Conference on Artificial Intelligence and Statistics, Microtome Publishing, Cadiz, Spain, 9–11 May, pp. 97–104.

Landrieu, L., Obozinski, G., 2016b. Cut pursuit: fast algorithms to learn piecewise constant functions on general weighted graphs. HAL hal-01306779, pp. 1–35.

Lawin, F.J., Danelljan, M., Tosteberg, P., Bhat, G., Khan, F.S., Felsberg, M., 2017. Deep projective 3d semantic segmentation. arXiv:1705.03428, pp. 1–12.

Lee, I., Schenk, T., 2002. Perceptual organization of 3d surface points. Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. XXXIV-3A, 193–198.
Lellmann, J., Kappes, J., Yuan, J., Becker, F., Schnörr, C., 2009. Convex multi-class image labeling by simplex-constrained total variation. In: Proceedings of the Second International Conference on Scale Space and Variational Methods in Computer Vision, Springer, Voss, Norway, 1–5 June, pp. 150–162.
Lellmann, J., Lellmann, B., Widmann, F., Schnörr, C., 2013. Discrete and continuous models for partitioning problems. Int. J. Comput. Vis. 104 (3), 241–269.
Lim, E.H., Suter, D., 2009. 3d terrestrial lidar classifications with super-voxels and multi-scale conditional random fields. Comput.-Aid. Des. 41 (10), 701–710.
Linsen, L., Prautzsch, H., 2001. Local versus global triangulations. In: Proceedings of Eurographics, Manchester, UK, 5–7 September, pp. 257–263.
Lu, Y., Rasmussen, C., 2012. Simplified Markov random fields for efficient semantic labeling of 3d point clouds. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, Vilamoura, Portugal, 7–12 October, pp. 2690–2697.
Mallet, C., Bretar, F., Roux, M., Soergel, U., Heipke, C., 2011. Relevance assessment of full-waveform lidar data for urban area classification. ISPRS J. Photogramm. Remote Sens. 66 (6), S71–S84.
Mitra, N.J., Nguyen, A., 2003. Estimating surface normals in noisy point cloud data. In: Proceedings of the Annual Symposium on Computational Geometry, ACM, San Diego, CA, USA, 8–10 June, pp. 322–328.
Moravec, H., Elfes, A., 1985. High resolution maps from wide angle sonar. In: Proceedings of the IEEE International Conference on Robotics and Automation, IEEE, St. Louis, MO, USA, 25–28 March, pp. 116–121.
Munoz, D., Bagnell, J.A., Vandapel, N., Hebert, M., 2009. Contextual classification with functional max-margin Markov networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, Miami, FL, USA, 20–25 June, pp. 975–982.
Munoz, D., Vandapel, N., Hebert, M., 2008. Directional associative Markov network for 3-d point cloud classification. In: Proceedings of the International Symposium on 3D Data Processing, Visualization and Transmission, Georgia Institute of Technology, Atlanta, GA, USA, 18–20 June, pp. 63–70.
Najafi, M., Taghavi Namin, S., Salzmann, M., Petersson, L., 2014. Non-associative higher-order Markov networks for point cloud classification. In: Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September, pp. 500–515.
Niemeyer, J., Rottensteiner, F., Soergel, U., 2014. Contextual classification of lidar data and building object detection in urban areas. ISPRS J. Photogramm. Remote Sens. 87, 152–165.
Niemeyer, J., Rottensteiner, F., Soergel, U., Heipke, C., 2016. Hierarchical higher order CRF for the classification of airborne lidar point clouds in urban areas. Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. XLI-B3, 655–662.
Niemeyer, J., Wegner, J.D., Mallet, C., Rottensteiner, F., Soergel, U., 2011. Conditional random fields for urban scene classification with full waveform lidar data. In: Proceedings of the ISPRS Conference on Photogrammetric Image Analysis, Springer, Munich, Germany, 5–7 October, pp. 233–244.
Paparoditis, N., Papelard, J.P., Cannelle, B., Devaux, A., Soheilian, B., David, N., Houzay, E., 2012. 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 200 (1), 69–79.
Pauly, M., Keiser, R., Gross, M., 2003. Multi-scale feature extraction on point-sampled surfaces. Comput. Graph. Forum 22 (3), 281–289.
Potts, R.B., 1952. Some generalized order-disorder transformations. Math. Proc. Cambridge Philos. Soc. 48 (1), 106–109.
Qi, C.R., Su, H., Mo, K., Guibas, L.J., 2016. PointNet: deep learning on point sets for 3d classification and segmentation. arXiv:1612.00593, pp. 1–19.
Raguet, H., Landrieu, L., 2015. Preconditioning of a generalized forward-backward splitting and application to optimization on graphs. SIAM J. Imaging Sci. 8 (4), 2706–2739.
Rudin, L.I., Osher, S., Fatemi, E., 1992. Nonlinear total variation based noise removal algorithms. Physica D: Nonlinear Phenom. 60 (1–4), 259–268.
Savinov, N., 2017. Point cloud semantic segmentation via deep 3D convolutional neural network. <https://github.com/nsavinov/semantic3dnet>.
Schindler, K., 2012. An overview and comparison of smooth labeling methods for land-cover classification. IEEE Trans. Geosci. Remote Sens. 50 (11), 4534–4545.
Schmidt, A., Niemeyer, J., Rottensteiner, F., Soergel, U., 2014. Contextual classification of full waveform lidar data in the Wadden Sea. IEEE Geosci. Remote Sens. Lett. 11 (9), 1614–1618.
Schmidt, M., 2012. UGM: Matlab code for undirected graphical models. <http://www.cs.ubc.ca/schmidtm/Software/UGM.html> (last access: 17 November 2016).
Serna, A., Marcotegui, B., 2013. Urban accessibility diagnosis from mobile laser scanning data. ISPRS J. Photogramm. Remote Sens. 84, 23–32.
Serna, A., Marcotegui, B., 2014. Detection, segmentation and classification of 3d urban objects using mathematical morphology and supervised learning. ISPRS J. Photogramm. Remote Sens. 93, 243–255.
Serna, A., Marcotegui, B., Goulette, F., Deschaud, J.E., 2014. Paris-rue-Madame database: a 3d mobile laser scanner dataset for benchmarking urban detection, segmentation and classification methods. In: Proceedings of the International Conference on Pattern Recognition Applications and Methods, ACM, Angers, France, 6–8 March, pp. 819–824.
Shapovalov, R., Velizhev, A., Barinova, O., 2010. Non-associative Markov networks for 3d point cloud classification. Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. XXXVIII-3A, 103–108.
Shapovalov, R., Vetrov, D., Kohli, P., 2013. Spatial inference machines. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, Portland, OR, USA, 23–28 June, pp. 2985–2992.
Simonovsky, M., Komodakis, N., 2017. Dynamic edge-conditioned filters in convolutional neural networks on graphs. arXiv:1704.02901, pp. 1–12.
Vallet, B., Brédif, M., Serna, A., Marcotegui, B., Paparoditis, N., 2014. IQmulus & TerraMobilita Contest – Analysis of mobile laser scans (MLS) in dense urban environments. MATIS Laboratory, French National Mapping Agency (IGN) and Center for Mathematical Morphology (CMM), MINES ParisTech. <http://data.ign.fr/benchmarks/UrbanAnalysis/> (last access: 17 November 2016).
Vallet, B., Brédif, M., Serna, A., Marcotegui, B., Paparoditis, N., 2015. TerraMobilita/iQmulus urban point cloud analysis benchmark. Comput. Graph. 49, 126–133.
Wainwright, M.J., Jaakkola, T.S., Willsky, A.S., 2003. Tree-reweighted belief propagation algorithms and approximate ML estimation by pseudo-moment matching. In: Proceedings of the International Workshop on Artificial Intelligence and Statistics, Society for Artificial Intelligence and Statistics, Key West, FL, USA, 3–6 January, pp. 1–8.
Waldhauser, C., Hochreiter, R., Otepka, J., Pfeifer, N., Ghuffar, S., Korzeniowska, K., Wagner, G., 2014. Automated classification of airborne laser scanning point clouds. In: Koziel, S., Leifsson, L., Yang, X.-S. (Eds.), Solving Computationally Expensive Engineering Problems: Methods and Applications. Springer, New York, USA, pp. 269–292.
Weinmann, M., 2016. Reconstruction and Analysis of 3D Scenes – From Irregularly Distributed 3D Points to Object Classes. Springer, Cham, Switzerland.
Weinmann, M., Jutzi, B., Hinz, S., Mallet, C., 2015a. Semantic point cloud interpretation based on optimal neighborhoods, relevant features and efficient classifiers. ISPRS J. Photogramm. Remote Sens. 105, 286–304.
Weinmann, M., Schmidt, A., Mallet, C., Hinz, S., Rottensteiner, F., Jutzi, B., 2015b. Contextual classification of point cloud data by exploiting individual 3d neighborhoods. ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci. II-3/W4, 271–278.
Weinmann, M., Urban, S., Hinz, S., Jutzi, B., Mallet, C., 2015c. Distinctive 2d and 3d features for automated large-scale scene analysis in urban areas. Comput. Graph. 49, 47–57.
West, K.F., Webb, B.N., Lersch, J.R., Pothier, S., Triscari, J.M., Iverson, A.E., 2004. Context-driven automated target detection in 3-d data. In: Proceedings of SPIE 5426, Automatic Target Recognition XIV, SPIE, Orlando, FL, USA, 12 April, pp. 133–143.
Wolf, D., Prankl, J., Vincze, M., 2016. Enhancing semantic segmentation for robotics: the power of 3-d entangled forests. IEEE Robot. Autom. Lett. 1 (1), 49–56.
Xing, E.P., Jordan, M.I., Russell, S., 2003. A generalized mean field algorithm for variational inference in exponential families. In: Proceedings of the 19th Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann, Acapulco, Mexico, 7–10 August, pp. 583–591.
Xiong, X., Munoz, D., Bagnell, J.A., Hebert, M., 2011. 3-d scene analysis via sequenced predictions over points and regions. In: Proceedings of the IEEE International Conference on Robotics and Automation, IEEE, Shanghai, China, 9–13 May, pp. 2609–2616.
Xu, S., Vosselman, G., Oude Elberink, S., 2014. Multiple-entity based classification of airborne laser scanning data in urban areas. ISPRS J. Photogramm. Remote Sens. 88, 1–15.