Detection, segmentation and classification of 3D urban objects using mathematical morphology and supervised learning

Andrés Serna, Beatriz Marcotegui

Detection, segmentation and classification of 3D urban objects using mathematical morphology and supervised learning

Andrés Serna, Beatriz Marcotegui

MINES ParisTech, CMM–Centre de Morphologie Mathématique,
35 rue St Honoré 77305-Fontainebleau-CEDEX, France
{*andres.serna_morales, beatriz.marcotegui*} *@mines-paristech.fr*

Abstract

In this paper, we propose an automatic and robust approach to detect, segment and classify urban objects from 3D point clouds. Processing is carried out using elevation images, called also digital elevation models, and the final result is presented reprojecting the image onto the 3D point cloud. First, the ground is segmented and objects are detected as discontinuities on the ground. Then, connected objects are segmented using a watershed constrained by the significant maxima. Finally, objects are classified in several categories using a support vector machine (SVM) approach with geometrical and contextual features.

Our methodology is qualitatively and quantitatively evaluated on three databases: one from Ohio (USA) and two from Paris (France). In the former, our method retrieves 98% of the objects in the detection step, 78% of them are correctly segmented and 82% of the well–segmented objects are correctly classified. In the latter, our method leads to an improvement of about 15% on the classification step with respect to previous works. Additionally, our approach is robust to noise since small and isolated structures are eliminated by morphological filtering. Quantitative results prove that our method not only provides a good performance but is also faster than other works reported in the literature on the same databases under similar conditions.

Keywords: 3D urban analysis, laser scanning, detection, segmentation, classification, mathematical morphology, support vector machine (SVM)

1. Introduction

Thanks to new 3D data availability, an increasing number of geographic applications such as Google Earth, Geoportail, iTowns and Elyx-3D is flourishing nowadays. Most of them have been recently enhanced with pedestrian navigation options and realistic 3D models. In general, 3D city models are useful for many applications: urban planning, emergency response simulation, cultural heritage documentation, virtual tourism, itinerary planning, accessibility analysis for different types of mobility, among others. Some of these applications not only require to look realistic but have also to be faithful to reality. Thus, semantic analysis from real data (images and 3D point clouds) are required in order to give faithfulness to 3D city models. These analyses are usually carried out by manual assisted approaches, leading to time consuming procedures, unsuitable for large scale applications. In that sense, automatic methods for urban semantic analysis are required.

Our work is part of TerraMobilita project[1]—"3D mapping of roads and urban public space, accessibility and soft–mobility". The project is built around two main topics: i) to develop new methods and tools to create and update 3D urban maps using laser scanning and digital imagery; ii) to develop innovative applications for soft–mobility itinerary planning. The focus of this work is automatic detection, segmentation and classification of urban objects from laser scanning data. Our method is based on elevation images, mathematical morphology and supervised learning. It is validated on three databases in order to get comparative results with the state of the art: two Mobile Laser Scanning (MLS) datasets from Paris (France) and an Aerial/Terrestrial Laser Scanning (ALS/TLS) dataset from Ohio (USA).

This paper is organized as follows. Section 2 reviews related work in the state of the art. Section 3 describes our method to process 3D point clouds using elevation images. Section 4 presents experiments and comparative results with the state of the art. Finally, Section 5 concludes the work.

2. Related work

Even though 3D acquisition systems have a high maturity level, 3D automatic analysis of urban areas is still an active research area. In the last years, several automatic solutions have been developed with different aims.

[1] http://cmm.ensmp.fr/TerraMobilita/

Table 1 summarizes representative papers related to our work. The detection-segmentation method, the classification strategy, the data structure and the accuracy reported on each paper is summed up in the table. Performance ranges from 58% to 95% but results are not comparable because they use different databases and different object classes, have different aims, use different data structures and process data in different ways. This table only offers an idea on each method performance. As a general observation, several authors use elevation images, clustering methods for detection-segmentation, and supervised techniques for classification. Further details are given below.

Table 1: Comparison of the state of the art (P: Precision, R: Recall). Colors indicate similar methods used by different authors.

Authors	Detection & Segmentation	Classification	Number of classes	Accuracy
Mallet et al. (2008)	Full–waveform analysis, Mathematical morphology	SVM	3 (buildings, ground, vegetation)	P=95.0%
Golovinskiy et al. (2009)	*Elevation images*, Graphs, contextual analysis	Hierarchical clustering, SVM	16 (cars, pole-like objects, trash cans, parking meters, ...)	P=58%, R=65%
Hernández and Marcotegui (2009b)	*Elevation images*, Mathematical morphology	SVM, Linear Discriminant Analysis	4 (cars, lampposts, pedestrians, others)	P=86.21%
Munoz et al. (2009)	Contextual analysis, clustering	High-order Markov models	5 (vegetation, wires, poles/trunks, load bearing, facades)	P=87.1%
Owechko et al. (2010)	3D strip by strip processing	Decision trees	17 (Buildings, ground, cars, bollards, lampposts, trees,...)	P=70.0%
Zhu et al. (2010)	*Elevation images*, Graph–cuts	SVM, Decision trees	7 (buildings, bushes, cars, trees, pedestrians, bicycles, others)	P=89.6%
Demantke et al. (2010)	3D adaptive neighborhood, Principal Component Analysis	Decision trees, dimensionality features	4 (lines, planes, volumes, noise)	P=69.3%
Douillard et al. (2011)	Voxelisation, Hierarchical clustering	Decision trees, RANSAC, clustering	16 (ground and several urban objects)	P=89.0%
Rutzinger et al. (2011)	3D Hough transform, region growing	Shape models, 3D alpha shapes	2 (trees, non-tree)	P=93%, R=86%
Pu et al. (2011)	geometrical and topological analysis	Decision trees	3 (poles, trees, others)	P=73.5%
Velizhev et al. (2012)	RANSAC, hierarchical clustering, spin images	Implicit shape models	2 (cars, light poles)	P=69%, R=80%

Several methods project 3D information onto a 2D grid in order to reduce the problem complexity and to speed up the computational processing. As each pixel of the projected grid contains elevation information, it is called elevation image or digital elevation model. This kind of 2.5D images has a long tradition in the scientific community (Hoover et al., 1996) and it is of great interest nowadays due to technological developments in remote sensing equipments such as Riegl, Velodyne and Kinect sensors. Gorte (2007) presents a method to segment planes on TLS data using range images. The

3D point cloud is projected from the sensor point of view. As a result, a 'panoramic' range image is obtained and plane estimations are done for each pixel on the image. Then, a region growing approach is performed in order to segment pixels belonging to the same plane. In a similar way, Zhu et al. (2010) project MLS data to a 'panoramic' range image in which rows represent the acquisition time of each laser scan–line, columns represent the sequential order of measurement and pixel values code the distance from the sensor to the point. They propose a segmentation-classification pipeline using graphs, SVM and decision trees. Hernández and Marcotegui (2009b) propose a method projecting MLS data to elevation images, i.e. a nadir view of the scene. Ground and objects are segmented using morphological transformations and objects are classified in four categories (cars, lampposts, pedestrians, and others) using SVM.

Since processing based on elevation images is both precise and fast, real-time applications such as guiding autonomous vehicles have been addressed. Kammel et al. (2008) and Ferguson et al. (2008) have developed autonomous vehicles, for the DARPA Challenge 2007, able to drive through urban scenarios. They use off-line processed aerial images and 2D maps in order to determine road structure. Then, on-board laser scanners are used to build elevation images in order to detect static and mobile obstacles. Munoz et al. (2009), extending the work by Anguelov et al. (2005), propose High Order Markov Random Fields for on–board contextual classification. In general, approaches for autonomous vehicles do not require high (centimetre) accuracy but high speed in order to detect and predict obstacles in real time. More accurate but slower methods process the 3D point cloud directly. These approaches are suitable for applications with high accuracy requirements but no strict time constraints. One of the major problems is the 3D neighborhood definition, which is not as trivial as it is in the 2D case using elevation images. Demantke et al. (2010) propose a method to adapt 3D neighborhood radius based on local features. Radius selection is carried out optimizing local entropy. Then, dimensionality features are calculated on spherical neighborhoods in order to characterize lines (1D), planes (2D) and volumes (3D). Douillard et al. (2011) present a set of 3D segmentation methods based on voxelisation and meshing. Their algorithms are evaluated on manually labeled datasets and the best performance is achieved using clustering approaches.

Several general segmentation and classification frameworks can be also found in the literature. Golovinskiy et al. (2009) develop a set of algorithms

to detect, segment, characterize and classify urban objects. Their method is evaluated on an ALS/TLS database from Ohio (USA). Their pipeline is as follows: i) ground segmentation using graph cuts, ii) object detection and segmentation using hierarchical clustering, iii) object characterization using geometrical and contextual descriptors, and iv) object classification using SVM. Recently, Velizhev et al. (2012) have improved this workflow including spin images and implicit shape models. The major problems of these approaches are noise, sparse sampling and proximity between objects. Moreover, some prior knowledge about the object scale is required to set up thresholds. Schnabel et al. (2008) present a semantic system for 3D shape detection. Their algorithm consists in two main steps: i) a topology graph is built with primitive shapes extracted from the data; ii) a search is carried out in order to detect characteristic subgraphs of semantic entities. The main drawback is the graph complexity when dealing with non-trivial objects. Pu et al. (2011) propose a framework for segmenting and classifying urban objects from MLS data. This work starts with a rough classification into three large categories: ground, on-ground objects and off-ground objects. Then, based on geometrical attributes and topological relations, more detailed classes such as traffic signs, trees, building walls and barriers are recognized. Owechko et al. (2010) describe a similar pipeline: first, a spatial cueing is applied in order to identify potential objects; then, statistical classifiers based on decision trees are trained with geometrical and contextual features. Using these methods, occlusions and point density distribution are critical. Additionally, there is barely any problem recognizing large flat features such as ground, barriers and walls. However, there are some problems classifying pole–like objects such as trees, bollards and lampposts.

In order to solve these problems, several specific approaches have been proposed. For instance, Mallet et al. (2011) investigate the potential of full-waveform LiDAR data for urban areas classification. In that work, waveform features are used as input for a SVM classifier. Their results show that echo amplitude and radiometric features are suitable to classify buildings, ground and vegetation. Rutzinger et al. (2011) describe an automated workflow to segment and to model trees from MLS data. First, the input point cloud is segmented into planar regions using the 3D Hough Transform and surface growing algorithms. Then, the remaining small segments are merged applying a connectivity analysis. Next, non-tree objects are removed from the analysis using statistical measures. Finally, trees are thinned using 3D alpha shapes (Edelsbrunner and Mücke, 1994) and realistic 3D models are

generated. Zhou and Vosselman (2012) segment and model curbstones from ALS/MLS data. Their process is performed directly on the 3D point cloud, on a strip by strip basis, so intrinsic information between the neighboring strips is missing. Recently, Serna and Marcotegui (2013b) solved this problem by processing all strips at the same time using elevation images.

In the present work, we aim at developing a method to detect, segment and classify urban objects, suitable for large scale applications. We adopt a method based on elevation images because of their demonstrated efficiency in terms of result quality and computational time. Our method is fully-automatic using few a priori information, it is based on robust morphological operators and supervised classification. It can manage partial occlusions, it is robust to noise, and re-segmentation process is carried out in order to separate connected objects. Simple geometrical and contextual features lead to better results than other works reported in the literature. Additionally, computation is faster than other works because image elevation reduces the amount of data to be processed.

This work provides an incremental contribution over Hernández and Marcotegui (2009b) work. The main contributions of this paper are the improvements in the detection and classification steps: i) an improved object detection is provided dealing with objects located at the border of the scene and also thin vertical objects, such as poles (Section 3.3); ii) classification is carried out in an effective way using simple geometrical and contextual features and a hierarchical classification is proposed (Section 3.5); iii) finally, this work presents quantitative results on Paris and Ohio databases, leading to comparisons with the state of the art (Section 4).

3. Proposed methodology

Our general workflow is shown in Figure 1. First, the 3D point cloud is projected to elevation images. At that point, a digital terrain model (DTM) is automatically created and object hypotheses are generated as discontinuities on the ground. Facades are automatically segmented as the highest vertical structures in the elevation image. Then, small and isolated regions are eliminated and connected objects are segmented. As a result of the segmentation process, a label image is created containing a unique identifier for each segmented object. Next, several geometrical and contextual features are computed for each object and classification is carried out. As a result of the classification process, a class image is created containing a category

for each segmented object. Having labels and classes in two different images is useful in the case of connected objects belonging to the same class, e.g. alignments of parked cars. Finally, the label and class images are reprojected to the 3D point cloud in order to get the final result. This reprojection step transforms the 2D resulting images into a 3D point cloud. For this purpose, all 3D points projected on a given pixel take the label and the class from that pixel. This step is required only if the result have to be displayed in 3D. Detailed descriptions are presented in following subsections.

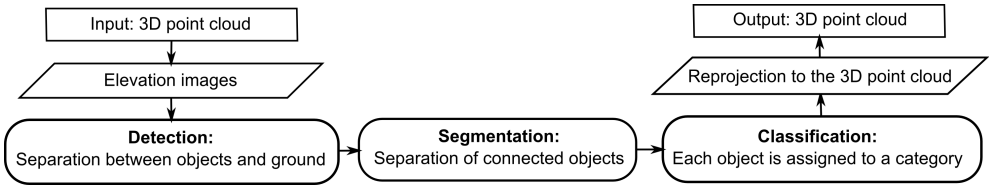


Figure 1: Detection, segmentation and classification of urban objects from 3D point clouds.

3.1. Elevation images

Elevation images are 2.5D structures that contain altitude information at each pixel. 3D point clouds are projected to elevation images because they are convenient structures to visualize and to process data. One can utilize all the large collection of existing image processing tools, in particular mathematical morphology (Matheron, 1975; Serra, 1988; Soille, 2003). Additionally, images can be processed quickly, implicitly define neighborhood relationships and require less memory than 3D data.

Elevation images are generated by an orthographic projection of the 3D point cloud using a virtual camera. This projection is a transformation from $\mathbf{R}^3 \rightarrow \mathbf{N}^2$. The virtual camera is located on the horizontal plane with normal vector $\overrightarrow{n} = (0, 0, 1)$ and crossing the lowest point in the point cloud $(0, 0, z_{min})$. Thus, each pixel on the elevation image contains the elevation of the grid cell above z_{min}. The only free parameter of this projection is the spatial pixel size (pw), which has to be carefully chosen. On one hand, if pw is too large, too many points would be projected on the same pixel losing fine details. On the other hand, too small pw implies connectivity problems and large image sizes, which would no longer justify the use of elevation images instead of 3D point clouds. To avoid connectivity problems and loss of infor-

mation, *pw* is chosen according to the point cloud resolution, as explained in Section 4.

In general, several points are projected on the same pixel. Thus, four images are defined: i) *maximal elevation image*, or simply elevation image, stores the maximal elevation among all projected points on the same pixel; ii) *minimal elevation image*, which stores the minimal elevation among all projected points on the same pixel; iii) *height difference image*, which contains the difference between maximal and minimal elevation images; and, iv) *accumulation image*, which stores the number of points projected on each pixel. In general, processing steps are performed on the elevation image. The other images are used to support some decisions during the analysis or to compute object features.

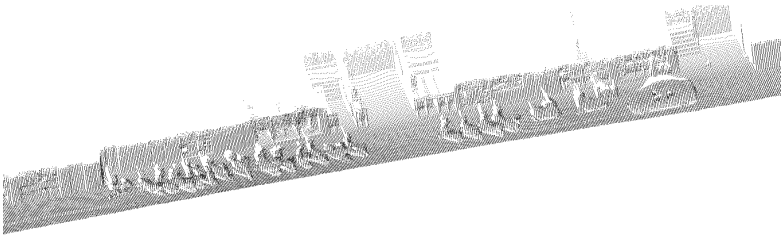
After image creation, a morphological interpolation is performed in order to fill holes caused by occlusions and missing scan lines. An interpolation technique based on the morphological *fill holes* operation (*Fill(f)*) is preferred since this transformation does not create new regional maxima in the image. In the most simple sense, a hole is a dark region (i.e. surrounded by brighter pixels), not connected to the image border. This interpolation strategy has been proposed by Hernández and Marcotegui (2009a) and a detailed explanation can be found in (Serna and Marcotegui, 2013b).

When detection, segmentation and classification have been carried out, images are reprojected to the 3D point cloud. Figure 2 describes the 3D point cloud processing using elevation images. A detailed explanation is presented in the following subsections.

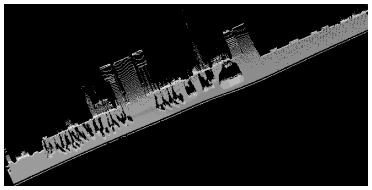
3.2. Ground segmentation

Ground segmentation is a critical step since urban objects are assumed to be located on it. When objects are filtered from the ground mask, the DTM can be generated. With the aim of segmenting ground, we use the approach proposed by Hernández and Marcotegui (2009a). It is based on the λ–flat zones labeling algorithm, firstly introduced in image processing by Nagao et al. (1979), defined by Meyer (1998) as:

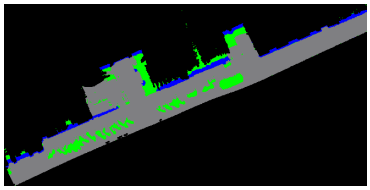
Definition 1. *Let f be a digital gray–scale image $f : D \rightarrow V$, with $D \subset Z^2$ the image domain and $V = [0, ..., R]$ the set of gray levels. Two neighboring pixels p, q belong to the same λ–flat zone of f, if their difference $|f_p - f_q|$ is smaller than or equal to a given λ value. For all $x \in D$, let $A_x(\lambda)$ be the*



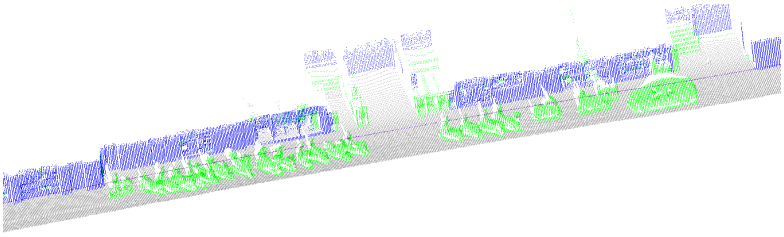
(a) Input point cloud



(b) Elevation image



(c) Segmented image



(d) Reprojection to the 3D point cloud

Figure 2: 3D point cloud processing using elevation images. Segmentation results: ground (gray), facade (blue), objects (green).

λ–flat zone of image f containing pixel x.

$$A_x(\lambda) = \{x\} \cup \{q | \exists \wp = (p_1 = x, ..., p_n = q) \textit{ such that } |f_{p_j} - f_{p_{j+1}}| \leq \lambda\} \quad (1)$$

The λ-flat zone labeling leads to a segmentation of the image, "that is, a partition into disjoint connected subsets (called segments) such that there exists a logical predicate returning true on each segment but false on any union of adjacent segments" (Horowitz and Pavlidis, 1974). With this definition, we want to obtain the ground mask $g_m(f) = \max \arg\{|(A_x(\lambda)|\}$ as the largest λ–flat zone in the elevation image. We set $\lambda = 20$ cm because it is usually high enough to merge road and sidewalk without merging other objects, even if there is no ramp access for the sidewalk.

3.3. Object detection

Our object detection method is based on mathematical morphology, inspired by Hernández and Marcotegui (2009a). They propose to detect urban objects using the top-hat by filling holes (THFH) followed by an area opening. In the first step, THFH is an effective and parameterless way to extract objects that appear as bumps on the elevation image. However, it fails extracting objects touching the image border because they are not considered as bumps. In the second step, an area opening $\gamma_{A_{min}}$ (Vincent, 1992) is performed in order to filter out small and noisy structures. Area opening is a morphological filter that removes objects with an area smaller than a given threshold A_{min}. This procedure is effective to get rid of noisy and isolated regions. However, it also removes thin objects such as bollards. In general, pole-like objects have a small area when they are seen from a nadir view, so they are suppressed by this filter. In this section, we propose an object detection framework that solves these two problems.

In order to solve the drawbacks of THFH step, a twofold strategy is proposed. A structure is considered to be object candidate if at least one of the two following conditions are fulfilled: i) it has not been reached by the λ–flat zones algorithm, i.e. it does not belong to the ground mask; ii) it appears as a bump on the elevation image. Therefore, the first set of object candidates is the ground residue, which is computed by the arithmetic difference between the elevation image and the ground mask ($f - g_m(f)$). The second set of object candidates is extracted using the THFH(f), as proposed originally by Hernández and Marcotegui (2009a). Then, the union of these sets is performed in order to get all object candidates. In order to solve the $\gamma_{A_{min}}$ drawbacks, the accumulation image is used. In general, vertical structures have high accumulation values. Thus, pole-like objects can be easily reinserted since their accumulation is higher than the accumulation for noisy objects.

Let us explain our detection method with an example. Figure 3 illustrates a typical acquisition profile. The urban profile contains the following urban objects enumerated from 1 to 7: 1) car, 2) pedestrian, 3) noisy structure, 4) dog, 5) pedestrian, 6) house facade, and 7) chimney. Note that this is only an illustrative example on a 1D profile. The processing is performed on the entire 2.5D elevation image.

The first step consists in interpolating occluded zones using a fill holes transformation, as explained in Subsection 3.1. Figure 3(a) presents the interpolated profile f. Using this transformation, each hole is filled with the

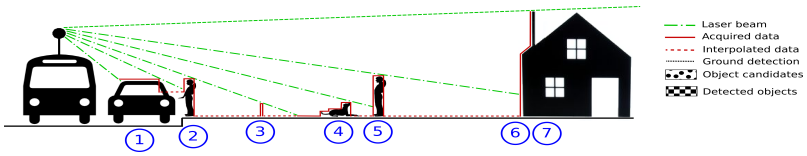
minimal value surrounding the hole. For example, consider the hole in the left part, between objects 2 and 4. This hole is filled at the ground level because in 2.5D it is connected to ground pixels. Additionally, consider the hole in the right part, between objects 5 and 6. This hole is also filled at the ground level even if the ground is not the minimal value surrounding that hole in this 1D profile. We assume that this hole can be filled at that level because the ground is not occluded by the pedestrian (object 5) in the previous or following profiles.

Figure 3(b) presents the first set of object candidates obtained as the ground residue. Note that almost all objects are retrieved. However, the dog in the middle of the sidewalk (object 4) is not detected because it is too low, thus it has been reached by the λ–flat zones propagation.

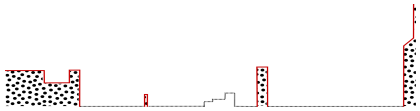
In order to obtain the second set of object candidates, the profile is inverted and holes are filled using the morphological fill holes transformation, as shown in Figure 3(c). Then, the transformation THFH(f)=Fill($\hat{f}$) $- \hat{f}$ consists in subtracting the inverted image $\hat{f}$ from the inverted filled image Fill($\hat{f}$), as shown in Figure 3(d). Note that this transformation correctly detects the dog in the middle of the sidewalk (object 4). However, the car in the left part (object 1) and the house in the right part (objects 6 and 7) are not retrieved because they are touching the border and then they do not become holes in the inverted profile. Figure 3(e) presents the complete set of object candidates, computed as the supremum between the two aforementioned sets of candidates $(f - g_m(f)) \vee$ THFH(f).

Figure 3(f) illustrates the effect of $\gamma_{A_{min}}$ in order to eliminate small and noisy structures. Note that the noisy structure in the middle of the sidewalk (object 3) has been correctly eliminated. However, the chimney (object 7) has also been suppressed. Finally, Figure 3(g) shows the result of the detection process, where the chimney has been reinserted because it has an important accumulation value.

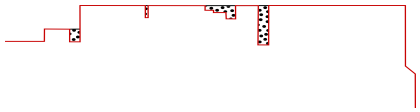
Figure 4 illustrates the detection process on real data. Note that all objects are detected by our method. For a better understanding, facades are marked in a different color. In our experiments, facades are automatically segmented as the highest vertical structures in the elevation image using a controlled reconstruction from markers, as explained in our previous work (Serna and Marcotegui, 2013a). Facade segmentation is out of scope of this work, but if one is interested in detecting facades independently, several other works are available in the literature (Boulaassal et al., 2007; Hammoudi, 2011; Rutzinger et al., 2011; Poreba and Goulette, 2012).



(a) Acquisition scheme and interpolated profile f.

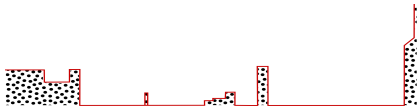


(b) In black: ground mask $g_m(f)$. In red: first set of object candidates $f - g_m(f)$



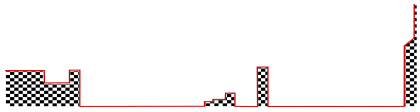
(c) Inverted profile $\hat{f}$ and fill holes Fill($\hat{f}$)

(d) Second set of object candidate: THFH(f)=Fill($\hat{f}$) $- \hat{f}$



(e) Complete set of object candidates: $(f - g_m(f)) \vee$ THFH(f)

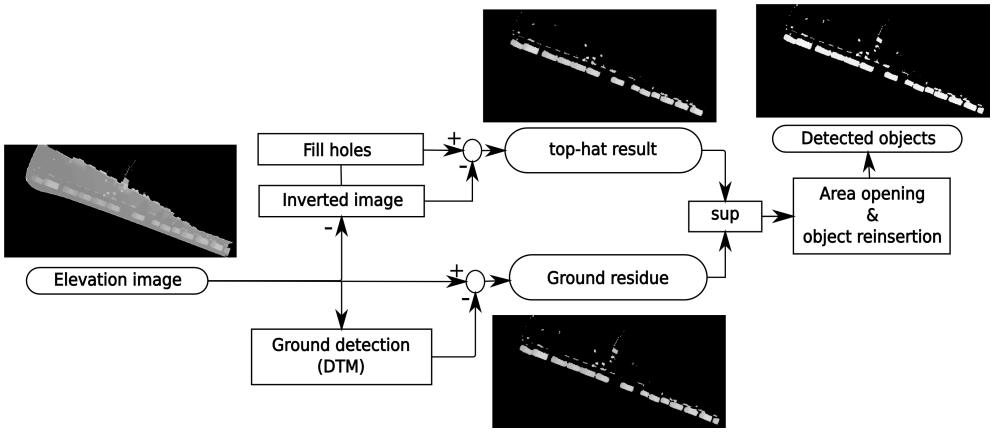
(f) Area opening: $\gamma_{A_{min}}((f - g_m(f)) \vee THFH(f))$



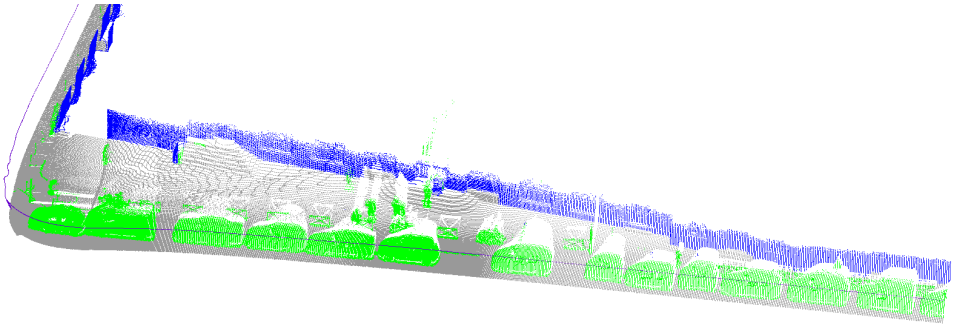
(g) Objects and DTM. Objects with significant accumulation are reinserted.

Figure 3: Detection method on a 1D profile.

Figure 5(a) illustrates the pole-like object reinsertion. Note that several pole–like objects are removed by an area opening filter with A_{min}=0.1 m^2. In Figure 5(b), objects with an accumulation higher than 10 are reinserted



(a) Detection scheme using elevation images



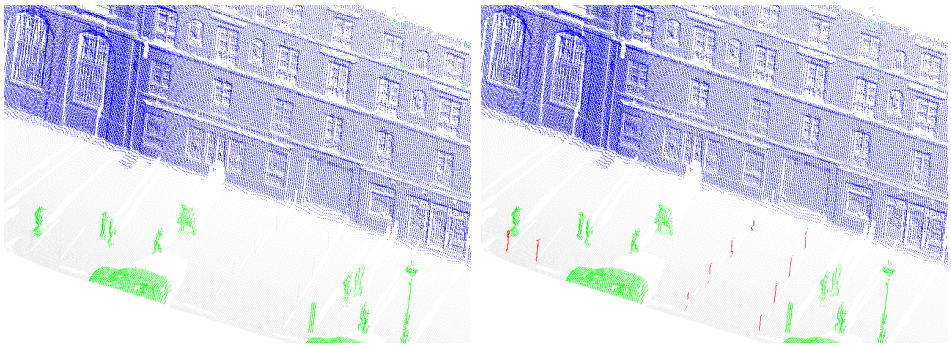
(b) Reprojection to the 3D point cloud: ground (gray), objects (green), facade (blue) and acquisition trajectory (violet).

Figure 4: Object detection using the top–hat by filling holes and the ground residue.

(in red). Note that a tilted bollard (black) is not recovered because it has not enough accumulation. A lower threshold can be used in order to retrieve this tilted bollard but at the risk of preserving other noisy structures.

3.4. Object Segmentation

Using our detection approach, it is possible to get several objects, close to each other, merged into a single connected component (CC). For example, in the left part of Figure 3, a pedestrian (object 2) and a car (object 1) are detected in the same CC. Another example is shown in Figure 6(a), where several cars are merged into a single CC. In order to solve this problem, we



(a) Detected objects after area opening

(b) Reinserted poles (red) and a tilted bollard not retrieved (black)

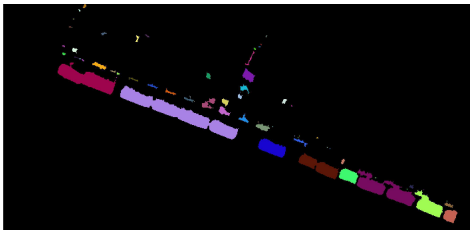
Figure 5: Pole reinsertion using accumulation. In the real scene there are 10 bollards. However, one of them is not reinserted because it is tilted, thus it has not enough accumulated points on the same pixel.

apply the solution proposed by Hernández and Marcotegui (2009b): "the number of connected objects in the same CC is equal to the number of significant maxima on it". With the aim of preserving only the most significant maxima, i.e. to get rid of maxima due to texture and noise on the upper part of the objects, a morphological h-Maxima filter is used (Soille, 2003). The h-Maxima filter eliminates maxima with a low local contrast whose relative height is less than or equal to a given threshold h. Using filtered maxima as markers, a constrained watershed on the elevation image is applied in order to segment connected objects. Figure 6 illustrates the performance of this re–segmentation.

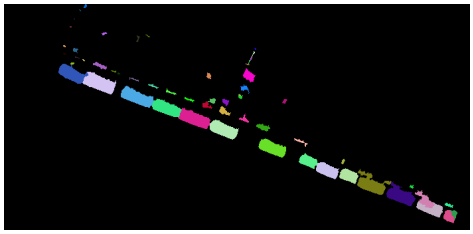
The main disadvantage appears when segmenting objects such as bikes, fences, trees or lampposts with several arms. They could be over–segmented because they have more than one significant maximum in the elevation image.

3.5. Object classification

Several classification methods have already been applied to 3D data in urban areas. In general, supervised classifiers are preferred since they offer a higher performance. In addition to the feature vector, a set of labels associated to each training sample is required. This set is called the training dataset, which is used to estimate the parameters of the classifier. An important underlying assumption is that the whole dataset has similar feature



(a) Elevation image: connected objects



(b) Elevation image: segmented objects

(c) 3D point cloud: segmented objects

Figure 6: Object segmentation using a constraint watershed from object maxima. Each color represents a different object.

distribution with respect to the training dataset. This means that test and training datasets must have similar features in order to achieve a good performance. To prevent over–fitting, bootstrapping or cross-validation techniques can be used.

In our work, SVM is chosen because it has remarkable abilities to deal with both high–dimensional data and limited training sets, it is easy to implement, a simple set of features is used as input and a good performance is reported in the literature for similar applications (Mallet et al., 2008; Hernández and Marcotegui, 2009b; Alexander et al., 2010; Mountrakis et al., 2011). Other methods, such as random forests and high order Markov models could also be suitable and they are known for providing similar performance (Anguelov et al., 2005; Mallet et al., 2008; Munoz et al., 2009).

In order to build the feature vector, three set of features are used:

- **Geometrical features:** object area and perimeter; bounding box area; maximum, mean, standard deviation and mode (the most frequent value) of the object height; object volume, computed as the integral of the elevation image over each object.

- **Contextual features:** Neighboring objects N_{neigh}, defined as the number of regions touching the object, using 8-connectivity on the elevation image. This feature is very discriminative in the case of group of trees and cars parked next to each other; confidence index $C_{ind} = n_{real}/(n_{real} + n_{interp})$, where n_{real} and n_{interp} are the number of non-empty object pixels before and after elevation image interpolation, respectively. In general, occluded and far objects have a low confidence index.

- **Color features:** Average red, green and blue over the object. These features are used if available.

The reliability of these features depends on the acquisition system. Accurate and calibrated sensors contribute to compute accurate features and to get a good classification performance. Note that geometrical features can be adapted to any XYZ point cloud, taking into account the acquisition system resolution. In our experiments, geometrical features are computed in the international unit system (SI units).

3.5.1. Hierarchical classification

With the aim of reducing confusion between classes with similar features and few examples in the database, we propose a hierarchical classification approach. The idea of hierarchical classification comes directly from the study of biological perceptual systems (Hubel and Wiesel, 1962; Poggio and Shelton, 1999), and it has been also applied in the remote sensing community (Avci and Akyurek, 2004; Pu et al., 2011).

First, data are separated into two parts: training and test sets. The definition of the hierarchical steps is entirely carried out on the training dataset.

Our hierarchical classification begins using general classes, then it continues in a top-down approach until obtaining more detailed classes.

This approach can be implemented as follows: i) an analysis is carried out on the training dataset applying a global classification taking all available classes into account; ii) training errors are computed using a k-fold cross-validation approach. In k-fold cross-validation, we first divide the training set into k subsets of equal size. In our experiments, we have used k=10. Sequentially one subset is tested using the classifier trained on the remaining k-1 subsets. Thus, each instance of the whole training set is predicted once. iii) classical Precision $P(train)$, Recall $R(train)$ and

$f_{mean}(train) = (2 \times P(train) \times R(train))/(P(train) + R(train))$ statistics are computed in order to evaluate our training results. Classes with high confusion rates ($f_{mean}(train)$ lower than 80%) are identified. In general, these classes correspond to heterogeneous objects with few examples. These classes are gathered in more general new classes; iv) using the whole training dataset, two kind of classifiers are trained: the first one is a classifier trained with the well-distinguished original classes and the new general ones; the second one is a more specific classifier used for each new general class aiming at obtaining more detailed classes; v) the process can be iterated. In our experiments, only two levels of hierarchy have been used.

Then, the resulting classifier is used to predict the test dataset. Precision $P(test)$, Recall $R(test)$ and $f_{mean}(test)$ results reported in Section 4 have been computed on the test dataset and reflect the performances of our system on real operation conditions.

4. Results

Our methodology is evaluated on three databases: rue Soufflot (Paris), Ohio (USA) and rues Vaugirard-Madame (Paris). As a general remark, our experiments demonstrate that almost all objects are retrieved by our detection approach. Then, segmentation is useful to separate connected objects such as pedestrians and cars. However, bikes and trees can be over–segmented. Finally, classification is carried out in an effective way using simple geometrical and contextual features. In our experiments, spatial pixel size is set to 0.04 m^2 (pw=20 cm width) and 0.01 m^2 (pw=10 cm width) for Ohio and Paris datasets, respectively.

It is noteworthy that our algorithms were initially developed to process Paris databases in the framework of TerraMobilita project. One of the main advantages of our method is that it can be easily generalized to other datasets without any major modification. This is underlined by the good results obtained on the Ohio database. Detailed results are presented below.

4.1. Rue Soufflot, Paris

For this experiment, we use a manual labeled dataset from rue Soufflot, a street approximatively 500 m long in the 5^{th} Parisian district. Acquisition was done by the Stereopolis MLS system from the French National Mapping Agency (IGN) (Paparoditis et al., 2012). A typical scene is shown in Figure 7. It contains pedestrians, cars, lampposts, motorcycles, among others. This

database was firstly used by Hernández and Marcotegui (2009b) to classify objects in four categories: cars, lampposts, pedestrians and others. However, his original annotation is no longer available. For the sake of comparison, we have manually annotated the database again and managed to reproduce results consistent with those reported by the author (shown in brackets in Table 2).

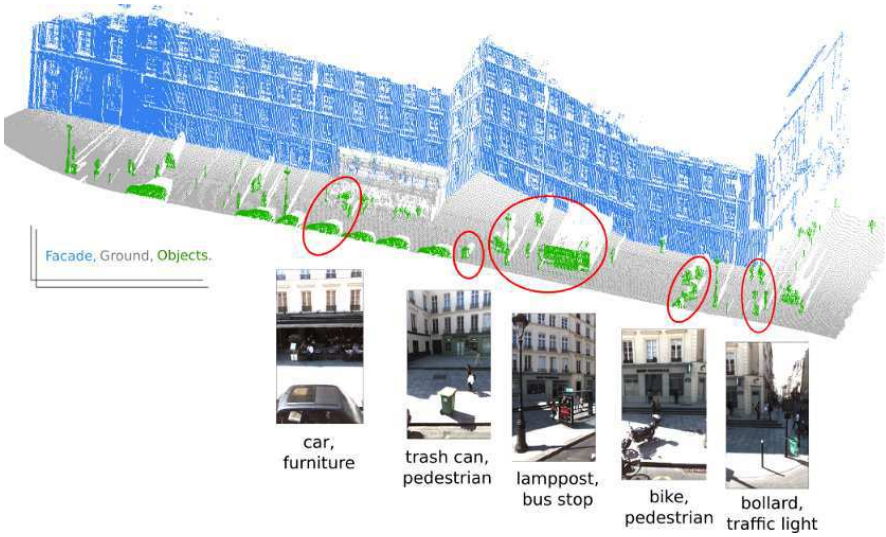
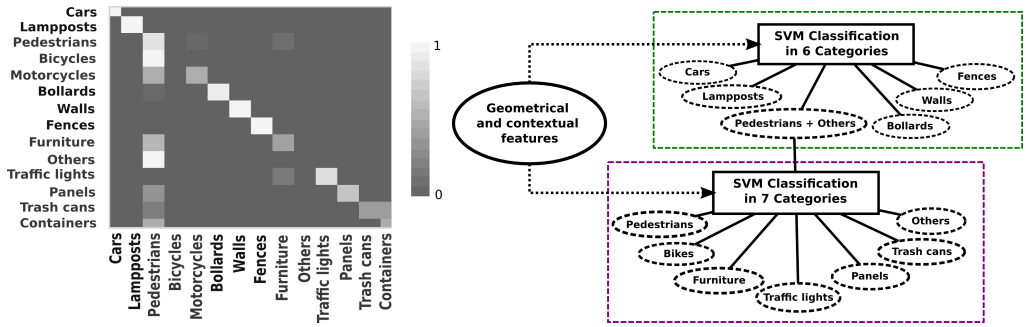


Figure 7: Example of urban objects manually annotated in the rue Soufflot dataset (Paris). Acquired by IGN–Stereopolis system.

First, data are separated into two parts, training and test sets. This separation has been randomly done keeping 50% of the objects of each class in the training set and the rest in the test set.

Color is not available in this database, thus only geometrical and contextual features have been used. In a first attempt, a single SVM classifier has been trained for all available categories. Training errors have been computed using 10–fold cross validation and high confusion rates were found between heterogeneous classes and classes with few examples, as shown in Figure 8(a). To solve these problems, the hierarchical classification proposed in Section 3.5 is applied, as shown in Figure 8(b). The first SVM classifies well–discriminated objects ($f_{mean}(train)$ greater than 80%), while the second one is exclusively dedicated to classes with higher confusion rates ($f_{mean}(train)$ lower than 80%). Table 2 presents our classification results on

the test set.



(a) Confusion matrix using all available categories on the training set.

(b) Hierarchical classification

Figure 8: Hierarchical SVM classification on rue Soufflot dataset.

Our main contribution in the classification step is the use of contextual features and hierarchical SVM. With respect to Hernández and Marcotegui (2009b) work, classification results have been improved. On the one hand, cars and lamppost classification have the same maximal accuracy (100%) while the performance on the pedestrian class has been improved by about 15%. On the other hand, we use all available categories preserving the performance on cars and lampposts categories. The main problems appear with classes *furniture* and *others* because they are very heterogeneous. The same problem appears for *traffic lights* and *trash cans* classes because there are not enough samples in the database (4 and 5 samples, respectively).

4.2. Ohio database

The Ohio database has also been used by Golovinskiy et al. (2009) and Velizhev et al. (2012) in order to evaluate their detection, segmentation and classification methods. This dataset is a combination of ALS and TLS data scanned in Ottawa city (Ohio, USA). It contains 26 tiles, 100 $\times$ 100 meters (approximatively $4{\times}10^6$ points) each, as shown in Figure 9. A typical scene contains trees, cars, lampposts, among others. The ground-truth (GT) consists in a labeled point marking the center of each object and its class.

Since our method is sequential, i.e. the input of each processing step is the output of the previous one, its evaluation is carried out in the same way. First, the detection process is applied to the entire database; second,

Table 2: Classification results on rue Soufflot test set. In brackets results from Hernández and Marcotegui (2009b).

Class	samples	Precision (%)	Recall (%)	f_{mean} (%)
Cars	27	100 (100)	100 (100)	100 (100)
Lampposts	12	100 (100)	100 (100)	100 (100)
Bollards	39	89	100	94
Walls	12	100	100	100
Fences	5	100	100	100
Pedestrians	101	86 (70)	84 (71)	85 (71)
Bikes	14	100	54	70
Furniture	30	67	67	67
Others	23	50	100	66.6
Traffic lights	4	0	0	0
Panels	7	100	100	100
Trash cans	5	0	0	0

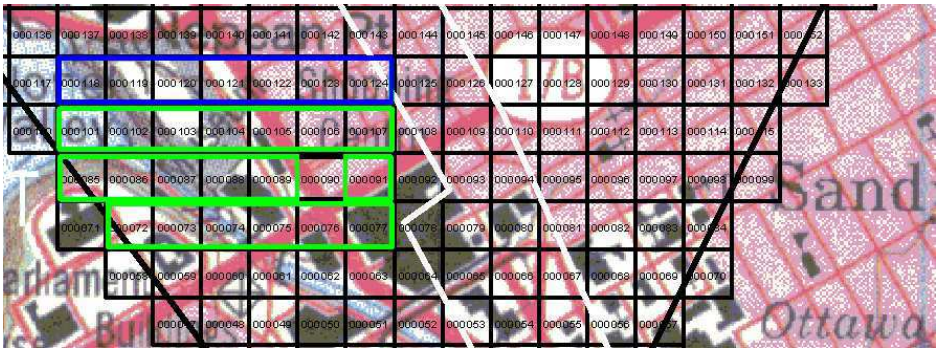


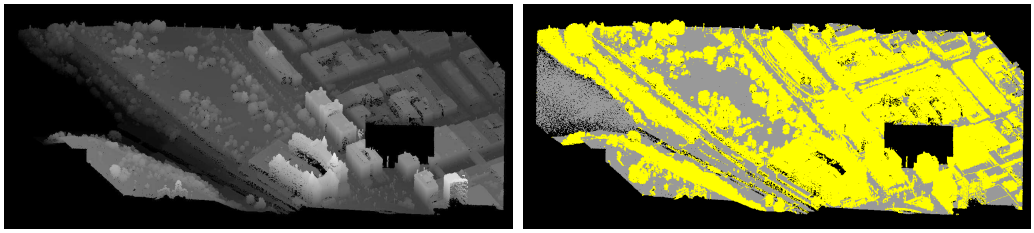
Figure 9: Ottawa city, Ohio (USA). The database contains 26 annotated tiles 100×100 meters each. Blue: train set; green: test set; black: non-annotated data.

detected objects are used as input for the segmentation step; and third, correctly segmented objects are separated in two subsets (train and test) in order to perform the classification. Let us to explain each processing step and its evaluation.

4.2.1. Evaluation: Detection

In order to evaluate our detection approach, an object is considered to be correctly detected if its GT center is included in the object hypotheses mask (Subsection 3.3), i.e. it has not been suppressed by any noise filter and it has not been wrongly merged with the ground. Note that an object

hypothesis may contain several connected objects or only a part of an object. In the detection step, we are only interested in keeping all possible objects. This is important because non detected objects cannot be recovered in the subsequent steps. Table 3 presents the percentage of retrieved objects in this database. Our detection method retrieves 98% of the objects, which outperforms other methods reported in the literature (92% by Golovinskiy et al. (2009) and 96% by Velizhev et al. (2012)). The number of false alarms cannot be estimated because many objects located on building roofs and in the forest are detected by our method (since they are real objects) , but they have not been annotated in the database. Figure 10 shows the detection results on the 3D point cloud.



(a) Z-coordinate (b) Object detection and DTM generation

Figure 10: The Ohio database: object detection (yellow) and DTM generation (gray).

4.2.2. Evaluation: Segmentation

In order to evaluate our segmentation approach, an object is considered to be correctly segmented if it is isolated as a single object, i.e. connected objects are correctly separated (there is no under-segmentation) and each individual object is inside one and only one connected component (there is no over-segmentation). However, an estimation of under-segmentation and over-segmentation errors cannot be done on the Ohio database because it only contains a GT point for each object. In that sense, an object is considered to be correctly segmented if it is marked with one and only one GT point.

As shown in Table 3, our method segments correctly 76% of the detected objects. Objects such as cars, lampposts, parking meters and signs are correctly segmented (Recall greater than 80%). The main problem comes from under-segmentation of connected objects such as light poles, posts and trees. Since this kind of clusters has only one maximum on the elevation image (the

highest object), they are not correctly segmented by our method. Note that trees recall is 90%, which means a good segmentation. However, trees represent approximatively 34% of the objects in the database, which implies that under-segmented trees affect seriously the recall of other classes, in particular for classes with few objects.

Table 3: Detection and segmentation results on the Ohio dataset.

			Detection		Segmentation	
Class	Name	GT	Detected	Recall	Segmented	Recall
1	Ad cylinder	6	6	100 %	5	83 %
2	Bush	29	28	97 %	23	82 %
3	Car	240	237	99 %	195	82 %
4	Dumpster	1	1	100 %	1	100 %
5	Fire hydrant	19	16	84 %	13	81 %
6	Flagpole	2	2	100 %	2	100 %
7	Lamppost	146	143	98 %	117	82 %
8	Light pole	62	60	97 %	46	77 %
9	Mailing box	4	4	100 %	1	25 %
10	Newspaper box	42	35	83 %	5	14 %
11	Parking meter	10	10	100 %	10	100 %
12	Post	377	376	100 %	208	55 %
13	Recycle bin	6	6	100 %	3	50 %
14	Sign	96	92	96 %	79	86 %
15	Telephone booth	4	4	100 %	2	50 %
16	Traffic ctrl. box	8	5	63 %	2	40 %
17	Traffic light	42	42	100 %	34	81 %
18	Trash can	19	19	100 %	8	42 %
19	Tree	552	543	98 %	490	90 %
20	Box transformer	2	2	100 %	0	0 %
Total		**1667**	**1631**	**98 %**	**1244**	**76 %**

4.2.3. Evaluation: Classification

For the classification experiments, segmented objects in the north quarter of the city (7 tiles, 458 objects) are used for training and the rest (19 tiles, 677 objects) for testing. Training and testing tiles are the same as in (Golovinskiy et al., 2009), for comparison purposes. The number of objects per class on both training and test sets are detailed in Table 5.

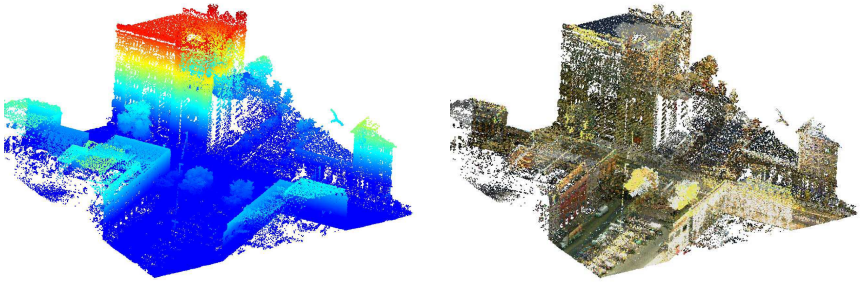
Geometrical, contextual and color features (Subsection 3.5) are combined in this experiment in order to define the best classification features. Classi-

fication performance obtained using different combinations of them is given in Table 4. The best overall accuracy (82%), defined as the ratio between the number of correctly classified objects and the total number of objects, is obtained combining geometrical and contextual features. Detailed results are presented in Table 5.

It is noteworthy that including color information degrades the classification accuracy. The reason is that in this database, color information is the result of overlapping several aerial and terrestrial scans. During acquisitions, color sensors were not calibrated, thus their superposition is not perceptually coherent, as shown in Figure 11.

Table 4: Classification accuracy using different features combination.

Features	Overall accuracy
Geometrical	75%
Geometrical + C_{ind}	77%
Geometrical + C_{ind} + N_{neigh}	**82%**
Geometrical + C_{ind} + N_{neigh} + Color	72%



(a) Tile #72: Height information (b) Tile #72: Color information

Figure 11: The Ohio database

Table 5 shows detailed classification results. Precision, Recall and f_{mean} for each class are presented. In this experiment, classes with less than 5 objects, either in the training set or in the testing set, are not considered in the classification process. Therefore, only 6 categories have been used. It is noteworthy that cars, trees and posts are correctly classified. However, lampposts, lights and signs have lower accuracies.

Table 5: Classification results on the Ohio dataset. Classes with less than 5 objects, either in the training set or in the testing set, are not considered. Pred: predicted, TP: true positives, P: Precision, R: Recall.

		Correctly segmented		Classification				
Class	Name	Train	Test	Pred	TP	P	R	f_{mean}
1	Ad cylinder	2	3					
2	bush	1	22					
3	car	108	87	85	75	88%	86%	87%
4	dumpster	1	0					
5	Fire hydrant	3	10					
6	flagpole	1	1					
7	Lamppost	33	84	78	51	65%	61%	63%
8	Light pole	14	32	22	16	73%	50%	59%
9	Mailing box	0	1					
10	Newspaper box	0	5					
11	Parking meter	10	0					
12	post	132	76	85	66	78%	87%	82%
13	Recycle bin	1	2					
14	sign	34	45	44	33	75%	73%	74%
15	Telephone booth	1	1					
16	Traffic ctrl. box	1	1					
17	Traffic light	4	30					
18	Trash can	0	8					
19	tree	137	353	363	317	87%	90%	89%
20	Box transformer	0	0					
Total (used classes)		**458**	**677**	**677**	**558**	**82%**	**82%**	**82%**
Total (all objects)		**483**	**761**					

For a better understanding, Table 6 shows the confusion matrix. Note that cars are correctly classified while lampposts, lights, posts, signs and trees are mixed up, which is comprehensible because they are pole-like objects.

In an attempt to solve these confusion problems, the hierarchical classification approach (proposed in Section 3.5) has been studied. Lampposts, lights, posts and signs have been put together in a new class, while cars and trees are preserved in their original classes. A first classifier is applied to separate correctly discriminated objects, and a second one is exclusively dedicated to classes with higher confusion rates. After our experiments, we have noted that this approach does not provide any global improvement in this database since f_{mean} increases by 16% for lampposts and lights, but it

decreases by 15% for posts and signs. The conclusion here is that a hierarchical approach is not enough to solve confusion problems since objects are too similar. A possible solution is the use of other features which allow a clearer separation between classes.

Table 6: Confusion matrix for classification in 6 classes on the Ohio database.

GT\Predict.	Cars	Lampposts	Light	Post	Sign	Tree	**Total**
Car	**75**	0	0	0	1	11	87
Lamppost	1	**51**	1	11	1	19	84
Light	0	6	**16**	0	0	10	32
Post	0	3	1	**66**	2	4	76
Sign	0	3	0	7	**33**	2	45
Tree	9	15	4	1	7	**317**	353
Total	85	78	22	85	44	363	

Table 7 presents results gathering lampposts, lights, posts, and signs in a more general category called pole-like objects. With 3 classes, the overall accuracy rises up to 88%.

Table 7: Confusion matrix gathering lampposts, lights, posts, and signs in the same category. Results on the Ohio dataset.

GT\Predict.	Cars	Pole-like	Trees	Total	Precision	Recall	f_{mean}
Car	**75**	1	11	87	88%	86%	87%
Pole-like	1	**201**	35	237	88%	85%	86%
Trees	9	27	**317**	353	87%	90%	89%
Total	85	229	363				

4.2.4. Comparison with the state of the art

The Ohio database has been chosen because it contains many different objects, it is large enough to exemplify a large-scale application, and comparison with the state of the art is possible since it has been used in other works (Golovinskiy et al., 2009; Velizhev et al., 2012).

We present our results on 26 tiles. However, in the original publication by Golovinskiy et al. (2009) (the website containing the dataset is not longer available), they report 27 tiles. Therefore, the number of objects is not the

same due to this missing tile. Additionally, some important differences have been noticed with respect to the aforementioned authors: on the one hand, with respect to Velizhev et al. (2012), they have only used 2 classes (cars and light poles), thus only a partial comparison can be done; on the other hand, with respect to Golovinskiy et al. (2009), the main difference comes from the fact that they do not consider trees nor bushes in their analysis.

Table 8 presents a quantitative comparison with the state of the art. Taking into account only 6 categories, the ones used during classification, our detection method (accuracy equal to 99%) performs better than the other two reported in the literature; our classification accuracy is equal to 82%, whereas Golovinskiy et al. (2009) correctly classify 65% of the objects considered by their method; with respect to the segmentation method, results from Velizhev are not available and our accuracy (78%) is 8% lower than that reported by Golovinskiy et al. (2009). On the one hand, our major under-segmentation problem is due to clusters formed by trees and pole-like objects, where the highest object is the only significant maximum. On the other hand, our major over-segmentation problem is when segmenting objects with several regional maxima such as trees. To summarize, our sequential method correctly detects, segments and classifies 99%×78%×82% = 64% of the annotated objects.

Table 8: Summarized comparison with other methods reported in the literature. The percent values indicate the accuracy in each stage of the workflow.

	Golovinskiy et al. (2009)	Velizhev et al. (2012)	Our method (2013)
Detection	92%	96%	99%
Segmentation	86%	N/A	78%
Classification	65%	67%	82%
Overall accuracy			**64%**
Computational time	7.3 min/tile (3 GHz PC)	5 $\sim$ 10 min/tile (4×2.4 GHz PC)	1 min/tile (4×2.4 GHz PC)

With respect to computational time (last row in Table 8), our method is up to 10 times faster than the other two works. In spite of hardware differences, these three works use general-purpose machines and they are not specially optimized nor parallelized. The aim of this comparison is to give

an idea to the reader about the computational time and the potential to large-scale or other time-constrained applications. One of the reasons of our faster processing is due to the use of elevation images and image processing algorithms since their computational cost is lower than that on the 3D case.

Note that the typical speed of a MLS system is 30 km/h, which corresponds approximatively to a covered area of 10,000 m^2/minute on a 20 m wide street without considering stops nor traffic lights. In this database, our processing speed is 10,000 m^2/minute. This is a very fast off-line processing since acquisition and processing times are equal.

4.3. Paris database: Rues Vaugirard-Madame

Dealing with cars has a particular interest in the framework of the TerraMobilita project since one of the applications consists in computing automatic parking statistics. In order to evaluate the potential of an automatic method, several 3D point clouds of the same street in Paris (Rues Vaugirard-Madame, approximatively a 500 m long section) have been acquired at different hours. Acquisition was done by the Stereopolis MLS system from the French National Mapping Agency (IGN) (Paparoditis et al., 2012). Then, we apply our automatic methodology in order to detect, segment and classify cars. For the classification step, urban objects were manually labeled. We use 2307 objects (129 *cars* and 2178 *others*) as training set, and 970 objects (53 *cars* and 917 *others*) as testing set. Note that a hierarchical classification is not applied since we are only interested in cars.

Color information is not available. Therefore, only geometrical and contextual features have been used. Table 9 presents our classification results using a binary SVM. The performance of our method is proved since 99.7% of the objects are correctly classified. Note that 5.4% of the cars have not been properly identified due to occlusion and over–segmentation problems.

Table 9: Results: car classification

Class	Precision	Recall	f_{mean}
Cars	100.0%	94.6%	97.2%
Others	99.7%	100.0%	99.9%

In order to demonstrate that our method can be easily generalized, we have used a classifier trained on the Ohio dataset in order to classify rue

Vaugirard-Madame cars. A f_{mean} equal to 90.0% has been obtained. This result is slightly lower than that reported in Table 9 (97.2%). However, the great advantage is that a new annotation may not be required when working with a new database.

At this point, our system is able to correctly extract cars and present some additional information such as the geographic position, geometric features and GPS time at the acquisition moment. However, a comparison between cars parked in the same place at different moments is required to compute parking duration statistics. In order to avoid confusions between those cars, geometrical and color features should be used. Additionally, relative sensor precision between different acquisitions becomes a critical issue. In efficiency terms, an automatic method seems to be suitable for this problem since the acquisition vehicle can go up to 20 times faster than a person. Additionally, the automatic processing takes only a few minutes and it is comparable to the acquisition time.

5. Conclusions

We propose an automatic and robust approach to detect, segment and classify urban objects from 3D point clouds. Processing is carried out using elevation images and the final result is presented reprojecting the image onto the 3D point cloud.

First, the ground is segmented using a lambda-flat zones propagation. Next, objects are detected using a two-fold strategy considering both structures connected to the boundary of the scene as well as ground discontinuities. Then, a filtering step is performed in order to reduce noise but preserving thin vertical structures. Subsequent, connected objects are segmented assuming that the number of significant maxima is equal to the number of connected objects. Finally, objects are classified in several categories using a SVM approach with geometrical and contextual features. Our geometrical features have can be adapted to any XYZ point cloud. Thus, the classification can be easily generalized, i.e. training on a database and testing on another one, as shown in rues Vaugirard-Madame dataset. This is a significant advantage because the model learned for a database can be applied to another one, even acquired by a different acquisition system, without the tedious manual annotation.

Our methodology is qualitatively and quantitatively evaluated on MLS and ALS/TLS databases from Paris (France) and Ohio (USA). Our results

on the Ohio dataset show that our method retrieves 99% of the objects in the detection step, 78% of connected objects are correctly segmented, and 82% of correctly segmented ones are correctly classified using geometrical and contextual features. On Paris dataset, our proposed hierarchical classification leads to an improvement of about 15% on the pedestrian class with respect to previous works while preserving a good performance in other classes. Moreover, new classes (not considered in previous works) have been taken into account.

Our method is robust to noise since small and isolated structures are eliminated using morphological filters. Additionally, it is fast because we project 3D points onto an elevation image and we process them as a complete set using digital image processing techniques.

Even if our method presents good results, it is noteworthy that several improvements should be done before developing a mature application. Our main problem, common to all methods in the literature, is due to large occluded regions. Several scans of the same zone could reduce this problem. Some under–segmentation and over–segmentation problems have been also pointed out. A possible solution can include shape/texture analysis to help deciding whether or not an object should be re-segmented.

Up to now, we have only used the spatial information available in the point cloud. However, additional features such as laser intensity and texture could improve our performance. Additionally, in the future we are planning to use Velodyne[2] data in order to distinguish static from mobile obstacles and to reduce occlusion problems.

Acknowledgements

The work reported in this paper has been performed as part of Cap Digital Business Cluster TerraMobilita project.
We want to thank Alexander Velizhev for providing us with his results on the Ohio database and the dataset itself.

Alexander, C., Tansey, K., Kaduk, J., Holland, D., Tate, N. J., 2010. Backscatter coefficient as an attribute for the classification of full-waveform

[2]Velodyne LiDAR: several lasers are mounted on upper and lower blocks of 32/64 lasers each and the entire unit spins. Taken from `http://velodynelidar.com/` [Last accessed: September 16, 2013]

airborne laser scanning data in urban areas. ISPRS Journal of Photogrammetry and Remote Sensing 65 (5), 423–432.

Anguelov, D., Taskarf, B., Chatalbashev, V., Koller, D., Gupta, D., Heitz, G., Ng, A., 2005. Discriminative learning of Markov random fields for segmentation of 3D scan data. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR. Vol. 2. pp. 169–176.

Avci, M., Akyurek, Z., 2004. A Hierarchical Classificaton of Landsat Tm Imagery for Landcover Mapping. In: The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. Vol. XXXV-B4. pp. 511–516.

Boulaassal, H., Grussenmeyer, P., Tarsha-kurdi, F., 2007. Automatic segmentation of building facades using terrestrial laser data. In: The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. Vol. XXXVI-3/W52. pp. 65–70.

Demantke, J., Mallet, C., David, N., Vallet, B., 2010. Dimensionality based scale selection in 3D LiDAR point clouds. In: The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. Vol. XXXVIII-5/W12. pp. 97–102.

Douillard, B., Underwood, J., Kuntz, N., Vlaskine, V., Quadros, A., Morton, P., Frenkel, A., 2011. On the segmentation of 3D LIDAR point clouds. In: IEEE International Conference on Robotics and Automation, ICRA'11. pp. 2798–2805.

Edelsbrunner, H., Mücke, E. P., 1994. Three-dimensional alpha shapes. ACM Transactions on Graphics 13, 43–72.

Ferguson, D., Darms, M., Urmson, C., Kolski, S., 2008. Detection, prediction, and avoidance of dynamic obstacles in urban environments. In: IEEE Intelligent Vehicles Symposium. pp. 1149–1154.

Golovinskiy, A., Kim, V. G., Funkhouser, T., 2009. Shape-based recognition of 3D point clouds in urban environments. In: 12th IEEE International Conference on Computer Vision. pp. 2154–2161.

Gorte, B., 2007. Planar feature extraction in terrestrial laser scans using gradient based range image segmentation. In: The International Archives

of the Photogrammetry, Remote Sensing and Spatial Information Sciences. Vol. XXXVI-3/W52. pp. 173–177.

Hammoudi, K., 2011. Contributions to the 3D city modeling. Ph.D. thesis, Université Paris-Est.

Hernández, J., Marcotegui, B., 2009a. Filtering of artifacts and pavement segmentation from mobile LiDAR data. In: The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. Vol. XXXVIII-3/W8. pp. 329–333.

Hernández, J., Marcotegui, B., 2009b. Point cloud segmentation towards urban ground modeling. In: The 5th GRSS/ISPRS Joint Urban Remote Sensing Event (URBAN2009). Shangai, China, pp. 1–5.

Hoover, A., Jean-baptiste, G., Jiang, X., Flynn, P. J., Bunke, H., Goldgof, D. B., Bowyer, K., Eggert, D. W., Fitzgibbon, A., Fisher, R. B., 1996. An explerimental comparison of range image segmentation algorithm. IEEE Transactions on Pattern Analysis and Machine Intelligence 18 (7), 673–689.

Horowitz, S. L., Pavlidis, T., 1974. Picture segmentation by a directed split-and-merge procedure. In: Proceedings of the 2nd International Joint Conference on Pattern Recognition. pp. 424–433.

Hubel, D. H., Wiesel, T. N., 1962. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. The Journal of Physiology 160 (1), 106–154.

Kammel, S., Ziegler, J., Pitzer, B., Werling, M., Gindele, T., Jagzent, D., Schröder, J., Thuy, M., Goebl, M., Hundelshausen, F. v., Pink, O., Frese, C., Stiller, C., 2008. Team AnnieWAY's autonomous system for the 2007 DARPA Urban Challenge. Journal of Field Robot. 25 (9), 615–639.

Mallet, C., Bretar, F., Roux, M., Soergel, U., Heipke, C., 2011. Relevance assessment of full-waveform LiDAR data for urban area classification. ISPRS Journal of Photogrammetry and Remote Sensing 66 (6), 71–84.

Mallet, C., Bretar, F., Soergel, U., 2008. Analysis of Full-Waveform LiDAR data for classification of urban areas. Photogrammetrie - Fernerkundung - Geoinformation (PFG) 5, 337–349.

Matheron, G., 1975. Random Sets and Integral Geometry. John Wiley & Sons, New York.

Meyer, F., 1998. From connected operators to levelings. In: Mathematical Morphology and its Applications to Image and Signal Processing. Vol. 12 of Computational Imaging and Vision. Kluwer Academic Publishers, pp. 191–198.

Mountrakis, G., Im, J., Ogole, C., 2011. Support vector machines in remote sensing: A review. ISPRS Journal of Photogrammetry and Remote Sensing 66 (3), 247–259.

Munoz, D., Vandapel, N. D., Hebert, M., 2009. Onboard contextual classification of 3-D point clouds with learned high-order Markov Random Fields. In: IEEE International Conference on Robotics and Automation, ICRA '09. pp. 2009–2016.

Nagao, M., Matsuyama, T., Ikeda, Y., 1979. Region extraction and shape analysis in aerial photographs. Computer Graphics and Image Processing 10 (3), 195–223.

Owechko, Y., Medasani, S., Korah, T., 2010. Automatic recognition of diverse 3-D objects and analysis of large urban scenes using ground and aerial LiDAR sensors. In: Conference on Lasers and Electro-Optics (CLEO) and Quantum Electronics and Laser Science Conference (QELS). pp. 16–21.

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.

Poggio, T., Shelton, C. R., 1999. Machine learning, machine vision, and the brain. AI Magazine 20 (3), 37–56.

Poreba, M., Goulette, F., 2012. RANSAC algorithm and elements of graph theory for automatic plane detection in 3D point clouds. Archives of Photogrammetry, Cartography and Remote Sensing 24, 301–310.

Pu, S., Rutzinger, M., Vosselman, G., Elberink, S. O., 2011. Recognizing basic structures from mobile laser scanning data for road inventory studies. ISPRS Journal of Photogrammetry and Remote Sensing 66 (6), 28–39.

Rutzinger, M., Pratihast, A. K., Oude Elberink, S. J., Vosselman, G., 2011. Tree modelling from mobile laser scanning data-sets. The Photogrammetric Record 26 (135), 361–372.

Schnabel, R., Wessel, R., Wahl, R., Klein, R., 2008. Shape recognition in 3D point clouds. In: Skala, V. (Ed.), The 16th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision. Union Agency-Science Press, pp. 1–8.

Serna, A., Marcotegui, B., 2013a. Attribute controlled reconstruction and adaptive mathematical morphology. In: 11th International Symposium on Mathematical Morphology. Uppsala, Sweden, pp. 205–216.

Serna, A., Marcotegui, B., 2013b. Urban accessibility diagnosis from mobile laser scanning data. ISPRS Journal of Photogrammetry and Remote Sensing 84, 23–32.

Serra, J., 1988. Image Analysis and Mathematical Morphology. Vol. 2. Academic Press, London.

Soille, P., 2003. Morphological Image Analysis: Principles and Applications. Springer-Verlag, Secaucus, NJ, USA.

Velizhev, A., Shapovalov, R., Schindler, K., 2012. Implicit shape model for object detection in 3D point clouds. In: The ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences. Vol. I-3. pp. 179–184.

Vincent, L., 1992. Morphological area openings and closings for grey-scale images. In: Proceedings of the Workshop: shape in picture. Springer, Driebergen, The Netherlands, pp. 197–208.

Zhou, L., Vosselman, G., 2012. Mapping curbstones in airborne and mobile laser scanning data. International Journal of Applied Earth Observation and Geoinformation 18, 293–304.

Zhu, X., Zhao, H., Liu, Y., Zhao, Y., Zha, H., 2010. Segmentation and classification of range image from an intelligent vehicle in urban environment. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS2010). pp. 1457–1462.