Expert Systems With Applications 58 (2016) 130–142

Contents lists available at ScienceDirect

Expert Systems With Applications

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

Building detection from orthophotos using a machine learning approach: An empirical study on image segmentation and descriptors

Fadi Dornaika[a,b,*], Abdelmalik Moujahid[a], Youssef El Merabet[c], Yassine Ruichek[d]

[a] University of the Basque Country UPV/EHU, Manuel Lardizabal 1, San Sebastian 20018, Spain
[b] IKERBASQUE, Basque Foundation for Science, Maria Diaz de Haro, 3, Bilbao 48013, Spain
[c] Faculté des Sciences, Université Ibn Tofail, Campus Universitaire, BP 133, Kénitra 14000, Morocco
[d] IRTES-SET, University of Technology of Belfort-Montbeliard, 90010 Belfort, France

ARTICLE INFO

Article history:
Received 22 May 2015
Revised 10 March 2016
Accepted 11 March 2016
Available online 31 March 2016

Keywords:
Automatic building detection and delineation
Orthophotos
Image segmentation
Image descriptors
Supervised learning
Classifier

ABSTRACT

Building detection from aerial images has many applications in fields like urban planning, real-estate management, and disaster relief. In the last two decades, a large variety of methods on automatic building detection have been proposed in the remote sensing literature. Many of these approaches make use of local features to classify each pixel or segment to an object label, therefore involving an extra step to fuse pixelwise decisions. This paper presents a generic framework that exploits recent advances in image segmentation and region descriptors extraction for the automatic and accurate detection of buildings on aerial orthophotos. The proposed solution is supervised in the sense that appearances of buildings are learnt from examples. For the first time in the context of building detection, we use the matrix covariance descriptor, which proves to be very informative and compact. Moreover, we introduce a principled evaluation that allows selecting the best pair segmentation algorithm-region descriptor for the task of building detection. Finally, we provide a performance evaluation at pixel level using different classifiers. This evaluation is conducted over 200 buildings using different segmentation algorithms and descriptors. The performance analysis quantifies the quality of both the image segmentation and the descriptor used. The proposed approach presents several advantages in terms of scalability, suitability and simplicity with respect to the existing methods. Furthermore, the proposed scheme (detection chain and evaluation) can be deployed for detecting multiple object categories that are present in images and can be used by intelligent systems requiring scene perception and parsing such as intelligent unmanned aerial vehicle navigation and automatic 3D city modeling.

1. Introduction

1.1. Motivation

Nowadays, automatic object recognition is a topic of growing interest for the machine vision community. In particular, the automatic building detection from monocular satellite and aerial images has been an important tool for many applications such as creation and update of maps and the Geographical Information Systems database, land use analysis, change detection and urban monitoring applications (Quang, Thuy, Sang, & Binh, 2015; Sirmacek & Unsalan, 2009; Sun et al., 2016; Unsalan & Boyer, 2005). Due to the rapidly growing urbanization, detecting buildings from images is a hot topic and an active field of research. Recently, vision and photogrammetry tools have been increasingly used in the processing of Geographical Information Systems, cultural heritage modeling, risk management, and monitoring of urban regions. More specifically, extracting objects such as roads and buildings has gain significant attention over the last decade. Aerial data are very useful for the coverage of large areas such as cities and several aerial-based approaches have been proposed for the extraction of buildings.

More precisely, the data employed as input to these approaches are either optical aerial images and derived Digital Surface Models (e.g., Tournaire, Brédif, Boldo, & Durupt, 2010) or aerial LiDAR 3D point clouds (e.g., Wang, Lodha, & Helmbold, 2006). It is well-known that segmenting buildings in aerial images is a challenging task. This problem is generally considered when we talk about high-level image processing in order to produce numerical or symbolic information. In this context, many techniques have

* Corresponding author at: University of the Basque Country UPV/EHU, Manuel Lardizabal 1, 20018 San Sebastian, Spain, Tel.: +34943018034.
E-mail addresses: fadi.dornaika@ehu.es, fdornaika@gmail.com (F. Dornaika), jibmomoa@gmail.com (A. Moujahid), elmerabet113@gmail.com (Y. El Merabet), yassine.ruichek@utbm.fr (Y. Ruichek).

http://dx.doi.org/10.1016/j.eswa.2016.03.024
0957-4174/

been proposed in the literature. Among the techniques most frequently used, one can cite semi-automatic methods that need user interaction in order to extract desired targets or objects of interest from images. Generally, this category of methods has been introduced to overcome the problems associated with the full automatic segmentation which is usually not perfect. It consists in dividing an image into two classes: "object" and "background". The interactivity consists in imposing some constraints to the segmentation that stipulate that some pixels (seeds) should belong to the object and some pixels should belong to the background. Rother, Kolmogorov, and Blake (2004) presented an iterative algorithm called GrabCut by simplifying user interaction. Their method combines image segmentation using graph cut and Gaussian mixture models (with the Orchard–Bouman clustering algorithm) of foreground and background structures in color space. A useful segmentation platform has recently been introduced by McGuinness and O'Connor (2010). The authors compared segmentation methods such as seeded region growing (SRG) (Adams & Bischof, 1994), Iterative Graph Cuts (Boykov & Jolly, 2001), and simple interactive object extraction (SIOX) (Friedland, Jantz, & Rojas, 2005).

1.2. Contribution

From the point of view of machine learning paradigms, it is desirable to keep the user interaction at the training phase only and to fully automate the detection and recognition at the test phase. In this paper, we propose an image-based approach for object detection and classification namely, detecting roof building in orthophotos. We use image segmentation algorithms to get an over-segmented orthophoto (e.g., Arbelaez, Maire, Fowlkes, & Malik, 2011; Nock & Nielsen, 2004). The obtained regions are then described by holistic and hybrid descriptors for detection of roof building in orthophotos. First, an over-segmentation is applied on the orthophoto. This over-segmentation is applied on both the training and test images. Second, holistic descriptors including color and texture are fused in order to get the feature descriptor of a given region. Third, the segmented regions in a test image are then classified using machine learning tools. We investigate the good combination (segmentation, descriptors) that can lead to optimal detection results via a case study over a set of aerial images. The main contributions of the paper are as follows. Firstly, we apply the matrix covariance descriptor to the building detection problem. To the best of our knowledge, this recent descriptor was not used in the context of building detection. This descriptor has proved to be very informative and compact. Secondly, we introduce a principled evaluation that studies the performances of the two main modules used in the detection chain, namely the image over-segmentation algorithm and the descriptor extractor. This study can provide and select the best pair segmentation algorithm-region descriptor in the context of building detection. Thirdly, we provide a performance study on classifiers whose role is to decide if any arbitrary region is a building or not. We provide evaluation performances over 200 buildings using different segmentation algorithms and descriptors.

While the application of the covariance descriptor to the building detection problem can be considered as one novel aspect of this current work, we point out that the objective of our work is not to propose a novel processing algorithm. Rather we are interested in studying the performance of a machine learning approach and its processing pipeline that combines several modules: image segmentation, image descriptor extraction, and classification. Thus, the work studies the influence of different modules on the final performance of building detection in orthophotos. Based on this study, we can identify the configurations that should be adopted for the task at hand.

The rest of the paper is organized as follows. Section 2 presents some related, state-of-the-art work. Section 3.1 describes the proposed machine learning approach as well as its main differences with existing work. It also presents the studied image descriptors together with their implementation details. Section 4 presents the performance evaluation through the use of image segmentation and descriptors classification. It presents an extended performance study of several combinations of pairs segmentation algorithm-image descriptor as well as of several classifiers. Finally, Section 5 provides some discussions and Section 6 concludes the paper.

2. Related work

This section is split in two main subsections. The first subsection describes briefly four image segmentation algorithms. The second subsection provides with an overview of the state-of-the art in building detection.

2.1. General purpose image segmentation

Before we proceed to the analysis of the effect of different segmentation methods on building roofs detection, we briefly review four popular segmentation techniques of the literature: Roerdink and Meijster (2001), Statistical Region Merging (SRM) (Nock & Nielsen, 2004), mean shift-based segmentation (MS) (Comaniciu & Meer, 2002), and Superpixels (Ren & Malik, 2003). These segmentation methods are well known and often used for building segmentation purposes. Most of these methods have several control parameters. Some parameters specify the image size or the output format. Other parameters are essential for the segmentation process.

Watershed algorithm: The Watershed algorithm is widely encountered and various definitions can be found in the literature (Roerdink & Meijster, 2001), (Vincent & Soille, 1991). In order to obtain the segmented image, the watershed technique uses a gradient image as input image, calculated using Di-zenzo's operators (Di-Zenzo, 1986) applied on the initial image. This segmentation technique has two main advantages: (1) it produces regions that are closed and connected; (2) the boundaries coincide with the most significant edges in the image.

Statistical Region Merging (SRM): Introduced by Nielsen and Nock, SRM is a fast and robust image segmentation technique (Nock & Nielsen, 2004). The algorithm considers each pixel as a region and it merges connected regions when their intensities are sufficiently similar according to a statistical test. This algorithm presents the advantage of not requiring any quantization or color space transformations. The number of regions is controlled by only a simple parameter Q, which represents the statistical complexity of the image and controls the level of segmentation.

Mean shift algorithm (MS): Based on gradient estimation, MS is an iterative statistical procedure to mode detection and clustering (Comaniciu & Meer, 2002). The mean shift approach is composed of two main steps: (1) filtering of the initial image and (2) clustering of the filtered image. The size and number of the obtained regions are controlled by two bandwidth parameters: the spatial bandwidth h_s related to the two spatial features and the range bandwidth h_r related to the color coordinate part of the feature vector. It is noted that the contours and the small regions are preserved after filtering.

Superpixels algorithm: Introduced by Ren and Malik (2003), Superpixels method is based on the graph cut algorithm that operates on graphs whose nodes are pixel values and whose edges represent affinities between pixel pairs. The advantage of the superpixels concerns their autonomous adaptation to the image structure where the produced regions, called "superpixels", are

small, local, compact and quasi-uniform. A recent state of the art of superpixel methods could be found in Achanta et al. (2012).

2.2. Building detection

Of the numerous man-made objects to be detected in aerial images, buildings can be the most relevant due to their distribution and complexity. Many approaches were proposed for building detection using aerial imagery. Most of these approaches adopt the hypothesis/verification paradigm. Thus, they first attempt to generate a hypothesis of building by clustering low-level image features such as edges and lines junctions, then try to check the generated hypothesis using either heuristics that depend on a geometric-model, or a statistical model such as a Markov Random Field (MRF). This technique requires the low-level image features to be computed explicitly, which may add some noises and inaccurate measurements in the image (Kumar & Hebert, 2003). This technique also requires to produce the hypothesis from a large number of image features extracted from images. The hypothesis/verification paradigm is very often based on some assumptions about buildings. For instance, the assumption of a rectangular footprint shape or flat rectilinear roofs is commonly used. This assumption can reduce the computational complexity, i.e., the number of building hypotheses. However, even with such assumptions the analysis of monocular images remains a challenging task since it generally leads to ambiguous solutions.

In Shorter and Kasparis (2009), the authors proposed an unsupervised method able to detect vegetation, building and non-building objects in aerial images. The assumption is that the building structures should have convex rooftop sections. In Sirmacek and Unsalan (2008), the authors exploit color invariant features in order to detect buildings in color aerial images. In this work, red roofs and shadows are extracted. Then the search for buildings is guided and improved by exploiting the detected shadow segments. In Ok, Senaras, and Yuksel (2013), the authors address the automated building detection using very high resolution optical satellite images. First, they search for the shadow regions to focus on building regions. To this end, the authors propose a new fuzzy landscape generation scheme to model the directional spatial relationship between buildings and their shadows. After all landscapes are known, a selection process is invoked to eliminate the landscapes due to non-building objects. The final building segments are estimated by the GrabCut partitioning method.

In Ngo, Collet, and Mazet (2015), an efficient approach is proposed for automatic rectangular building detection from monocular aerial images. Image is first decomposed into small homogeneous regions using superpixel segmentation of a masked image. Regions are then grouped into clusters by a region-level MRF segmentation method. Regions bordering shadows in the opposite direction of the illumination angle are flagged as building segments. The proposed approach cannot detect building regions whose shadow is not visible or is missing.

In Liasis and Stavrou (2016), the authors detect buildings in satellite images. The images are processed by applying a clustering technique using color features to eliminate vegetation areas and shadows that may adversely affect the performance of the algorithm. Subsequently, the Hue Saturation Value (HSV) representation of the image is used and a new active contour model was developed and applied for building extraction.

Two recent works started to exploit deep learning paradigms (Wu, Xu, Zhao, Li, & Xiang, 2015). Some attempted to apply Convolutional Neural Networks (CNN) to aerial images in order to either retrieve features or classes. In Saito, Yamashita, and Aoki (2016), the authors use CNNs that produce local label maps from rectangular aerial image patches. In this work, three categories (buildings, roads, and others) were considered. In Vakalopoulou, Karantzalos, Komodakis, and Paragios (2016), the authors propose an automated building detection framework from very high resolution remote sensing data based on deep CNNs. The core of their method is based on a supervised classification procedure employing a very large training dataset. An MRF model is then responsible for obtaining the optimal labels regarding the detection of scene buildings.

3. Proposed machine learning approach

3.1. Overview of the proposed framework and main differences with related work

The general flowchart of the proposed building-detection method is illustrated in Fig. 1. It should be noticed that the training set is formed by a set of labeled regions together with their image descriptor. As can be seen, many existing works are based on local features to classify each pixel to an object label, so that these approaches need class-conditional distribution of pixel values. For instance the work in Vakalopoulou et al. (2016) attempts to derive pixel based descriptors using CNNs, then a binary classifier is used at pixel level with an extra regularization step in order to get the real detected object of interest. This scheme needs to specify the ideal local image support as well as the parameters of the final regularization. In Saito et al. (2016), again the choice of the size of both the rectangular input patch and of the output label map seem to be ad hoc. Our proposed framework considers semantic patches that are provided by well studied image segmentation algorithms. Each local patch is represented by a rich descriptor that globally describes the entire region, and used to infer its category via a supervised scheme. Our proposed method also differs from existing works by the fact that it does not impose any constraint on building shapes whereas many works assume a quadrilateral shape for buildings.

3.2. Studied image descriptors

Image textures and color can characterize intensity variations of object surfaces. Texture representation and analysis are a main focus of machine vision. Texture classification subject to changes and perturbations in image acquisition process, such as illumination, noise, or scale, is very challenging, which often leads to a large intra-class variability. In this section, we present the image descriptors used in our work as well as some implementation details.

3.2.1. Color

Color was among the main descriptors that are used in order to characterize image regions. Indeed, existing methods have exploited color invariant descriptors in order to detect objects in aerial images (e.g., Sirmacek & Unsalan, 2008). Color invariant descriptors are object properties that are not affected by external conditions (Gevers & Smeulder, 2000). For image regions, color histograms can be considered as a simple and fast descriptor. These histograms computed in any color space quantifies color distribution in a given region and hence can be used as a discriminant signature. Fig. 3 illustrates the color descriptor associated with two different segmented regions, each belonging to a different class.

In our work, we use color histograms in RGB space. In order to computer color histograms, we uniformly quantize each color channel into 16 bins and then the color histogram of each region is computed in the feature space of $16 \times 16 \times 16 = 4096$ bins. Obviously, quantization reduces the information regarding the content of regions and it is used as trade-off to reduce processing time.

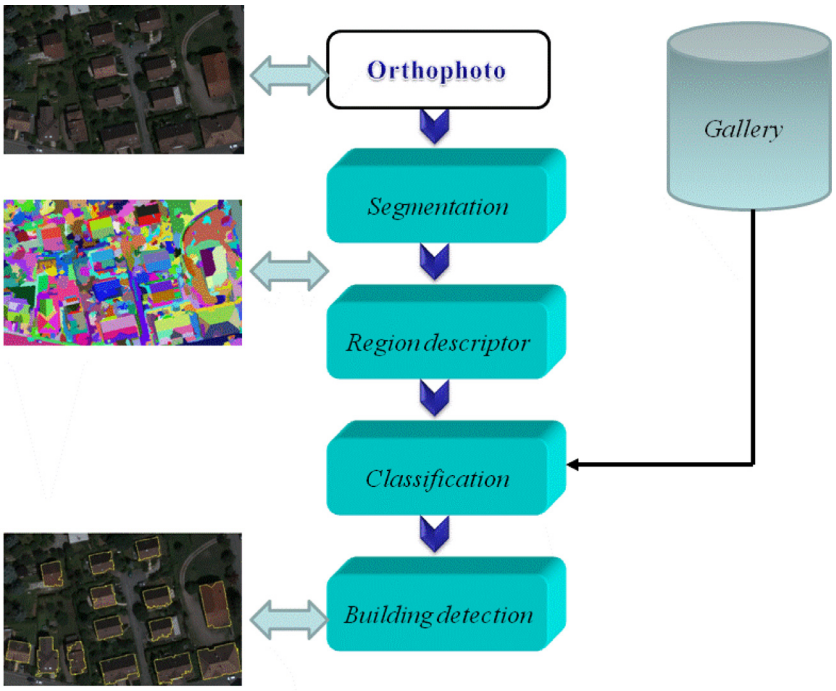


Fig. 1. General flowchart of the proposed machine learning building-detection method. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

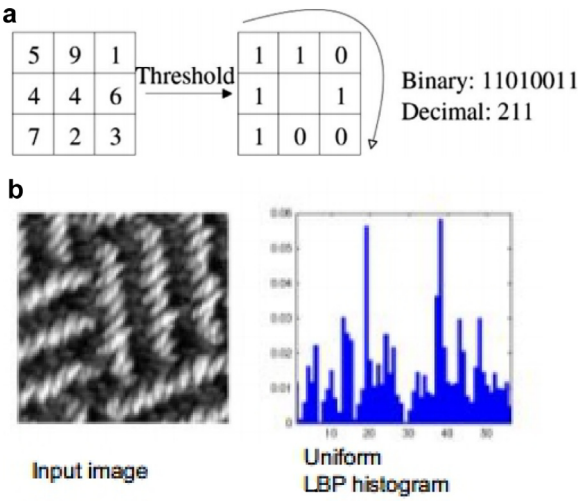


Fig. 2. (a) Example of basic LBP operator. (b) Example of LBP descriptor.

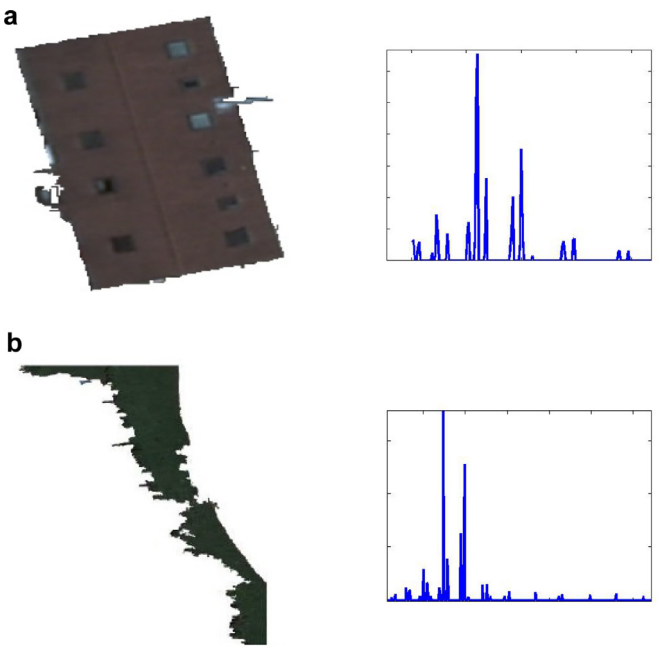


Fig. 3. (a) A segmented roof region and its color histogram. (b) A segmented background region and its color histogram. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

3.2.2. Local Binary Patterns

Local Binary Patterns are among the recent texture descriptors. The original LBP operator replace the value of the pixels of an image with decimal numbers, which are called LBPs or LBP codes that encode the local structure around each pixel (Ahonen, Hadid, & Pietikäinen, 2006; Ojala, Pietikäinen, & Maenpaa, 2002; Takala, Ahonen, & Pietikäinen, 2005). It proceeds thus, as illustrated in Fig. 2: each central pixel is compared with its eight neighbors; the neighbors having smaller value than that of the central pixel will have the bit 0, and the other neighbors having value equal to or greater than that of the central pixel will have the bit 1. For each given central pixel, one can generate a binary number that is obtained by concatenating all these binary bits in a clockwise manner, which starts from the one of its top-left neighbor. The resulting decimal value of the generated binary number replaces the central pixel value. The histogram of LBP labels (the frequency of occurrence of each code) calculated over a region or an image can be used as a texture descriptor of that image.

The size of the histogram is 2^P since the operator $LBP(P, r)$ is able to generate 2^P different binary codes, formed by the P neighboring pixels. Recently, several LBP variants have been developed in order to improve the texture description

(Bereta, Karczmarek, Pedrycz, & Reformat, 2013; Huang, Shan, Ardabilian, & Wang, 2011; Wolf, Hassner, & Taigman, 2008).

For describing a segmented region with LBP descriptors, in our work, we use eight neighboring points ($P = 8$) with three radii ($r = 1,\ r = 2,\ r = 3$) each with three modes (uniform, rotation invariant, uniform and rotation invariant). Thus, there are nine LBP descriptors. The final descriptor is given by the concatenation of all. It is worth noting that despite the use of nine LBP descriptors the final one is described by 3 × (59 + 36 +10) = 315 variables only.

3.2.3. Covariance descriptors

Tuzel et al. introduced the covariance descriptor (Tuzel & Meer, 2006). This descriptor represents an image or an image region using a sample covariance matrix. Let J denote a $M \times N$ intensity or color image, and V be the $M \times N \times d$ dimensional feature image extracted from J. Thus, V can be understood as a set of d 2D arrays (channels) where every array can correspond to a given image feature such as horizontal coordinate, vertical coordinate, color, image derivatives, and filter responses, etc. This 3D array can be written as $V(x; y) = \phi(J; x; y)$ where ϕ is a function that extracts image features. For a given image region $\mathcal{R} \in J$ containing n pixels, let $\{\mathrm{v}_i\}_{i=1\ldots n}$ denote the d-dimensional feature vectors obtained by ϕ within $\mathcal{R}$. According to Tuzel and Meer (2006), the region $\mathcal{R}$ can be described by a $d \times d$ covariance matrix:

$$\mathbf{S}_{\mathcal{R}} = \frac{1}{n-1}\sum_{i=1}^{n}(\mathrm{v}_i - \mathrm{m})(\mathrm{v}_i - \mathrm{m})^T$$

where m is the mean vector of $\{\mathrm{v}_i\}_{i=1\ldots n}$.

Since covariance matrices do not live in the Euclidean space, the difference between two matrices would not quantify the similarity or dissimilarity between the corresponding regions.

Under the Log-Euclidean Riemannian metric, it is possible to measure the distance between covariance matrices.

Given two covariance matrices S_1 and S_2, their distance is given by,

$$d(\mathrm{S}_1, \mathrm{S}_2) = ||\log(\mathrm{S}_1) - \log(\mathrm{S}_2)||_{\ell_2}$$

where $||.||_{\ell_2}$ is the ℓ_2 vector norm and log(S) is the matrix logarithm of the square matrix S.

Thus, every image region, $\mathcal{R}$, can be characterized by $\log(\mathrm{S}_{\mathcal{R}})$. Since this is a symmetric matrix, then the feature vector can be described by a $d \times (d+1)/2$ where d is the number of channels.

In our work, the image covariance descriptor is computed as follows. We consider 23 channels $(x, y, R, G, B, H, S, V, I_x, I_y, I_{xx}, I_{yy}, I_{xy}, I_{xy} LBP_{u2}(r = 1), LBP_{ri}(r = 1), LBP_{riu2}(r = 1), LBP_{u2}(r = 2), LBP_{ri}(r = 2), LBP_{riu2}(r = 2), LBP_{u2}(r = 3), LBP_{ri}(r = 3), LBP_{riu2}(r = 3))$. x denotes the channel that contains the horizontal coordinate of pixels, y denotes the channel that contains the vertical coordinate of pixels, R, G, B denote the three color components, H, S, V, the color channels in HSV space, I_x, I_y, I_{xx}, I_{yy}, I_{xy}, I_{yx} denote the first order and second order image partial derivatives, and $LBP_{mode}(R)$ denotes the LBP image obtained for a given $mode$ and a given radius R. In our work, we use nine LBP images associated with three different modes $mode \in \{uniform, rotation\ invariant, uniform\&rotation\ invariant\}$ and three different radii $R \in \{1, 2, 3\}$. For all LBP images, the number of neighboring points is fixed to 8. Since the number of channels used is 23, it follows that the descriptor of each region is described by 276 features. We stress the fact that even if the covariance matrix descriptor used color channels and LBP images, its descriptor is still different from that of color histograms and LBP histograms.

3.2.4. Hybrid descriptors

Hybrid descriptors can be obtained by concatenating the feature vectors provided by different descriptors. While this can enrich the discrimination capacity of the resulting descriptor, it has the disadvantage that the dimensionality of the resulting feature vector can be very high.

3.3. Training and testing

As in any training process, we need a set of segmented regions with known labels. In other words, each segmented region that will be used as a training example should have been identified as background or building. In our case, this process is semi-automatic. In order to get a training set which contains regions belonging to the two classes (background and building) with ground-truth labels, we proceed as follows. The buildings footprints are first manually delineated in each training orthophoto. Each such ground-truth map is then overlapped with the corresponding automatically over-segmented orthophoto.

The label of any segmented region can be inferred by using the size of the intersection with the ground-truth building region/pixels. Any segmented region whose overlap with a building footprint exceeds 90% of its size will be labeled as building. Any segmented region whose overlap with the building footprint is below 3% of its size will have the non-building label. The segmented regions that do not meet any of the two conditions are discarded and will not be included in the set of training samples. This selection scheme makes sure that the used descriptors are associated with their own classes. The reason behind using these thresholds is the fact that an automatically segmented region may be shared by a building region and a background region.

Fig. 4 illustrates the semi-automatic training process. (a) Depicts a part of an original orthophoto. (b) Illustrates the output of the SRM image segmentation algorithm. (c) Illustrates the ground-truth building footprints obtained manually. (d) Depicts the regions labeled as building and those labeled as background. In total, there are 19 regions that are labeled as positive examples (building samples). In brief, the training process consists of (i) obtaining a set of training descriptors together with their labels, and (ii) learning a classifier that can separate between building regions and non-building ones.

At testing phase, the same processing pipeline is adopted. Firstly, The orthophoto will be automatically segmented using the same segmentation algorithm used in training. Secondly, the descriptor associated with each segmented region in the test orthophoto will be computed. Thirdly, a classifier will decide the class of every segmented region, in the test orthophoto, using the set of trained descriptors. Based on this decision the region pixels will be labeled as building and non-building. The building pixels will be grouped into connected components that will represent the detected buildings in the test orthophoto.

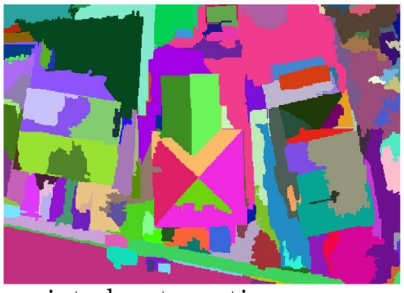
4. Performance evaluation

In this section, we present three groups of experiments. The first group studies the segmentation algorithms ability to provide non-mixed segmented regions. The second group evaluates the pair segmentation algorithm-descriptor. The third group of experiments studies the performance of different classifiers.

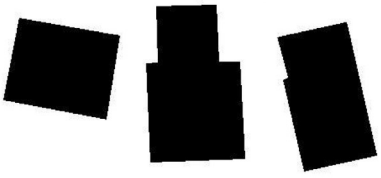
4.1. Dataset

The dataset used in this research to evaluate the accuracy of the proposed framework corresponds to 12 large orthophotos depicting several zones in the region of Belfort city situated on the

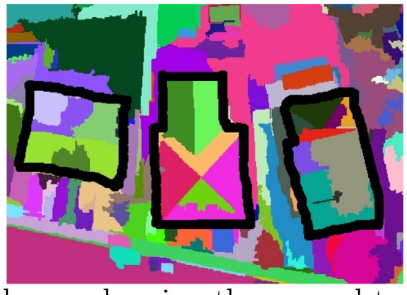
(a) Original orthophoto portion.



(b) Associated automatic oversegmentation.



(c) Associated ground-truth building delineation (manual delineation).



(d) Labeled regions by overlapping the ground-truth footprints with the segmented regions.

Fig. 4. Semi-automatic generation of training examples.

north-eastern of France. The spatial resolution of these orthophotos, provided by Communauté de l'Agglomération Belfortaine (CAB 2008), is 16 cm/pixel. These orthophotos contain about 200 buildings. In these orthophotos, the building roofs have different colors and textures. Furthermore, the background contains highly varying appearances corresponding to vegetation, cars, roads, and other objects.

4.2. Evaluation metrics

In order to get a quantitative evaluation, we use the ground-truth building maps. The manually delineated buildings (ground-truth buildings) were used as a reference building set to evaluate the whole automated building-extraction accuracy. The automatically detected buildings and the ground-truth buildings are compared pixel-by-pixel. All pixels in the test orthophoto are grouped into four categories.

1. True positive (TP). The automated and manual techniques classify the given pixel as belonging to the buildings.
2. True negative (TN). The automated and manual techniques classify the given pixel as belonging to the background.
3. False positive (FP). The automated technique misclassifies the given pixel as belonging to a building.
4. False negative (FN). The automated technique incorrectly classifies the given pixel as belonging to the background.

From these measures it is straightforward to compute the following scores associated with the building regions in the test image: recall, precision, F1 measure, accuracy, and Matthews correlation coefficient (MCC). The MCC returns a score between −1 and

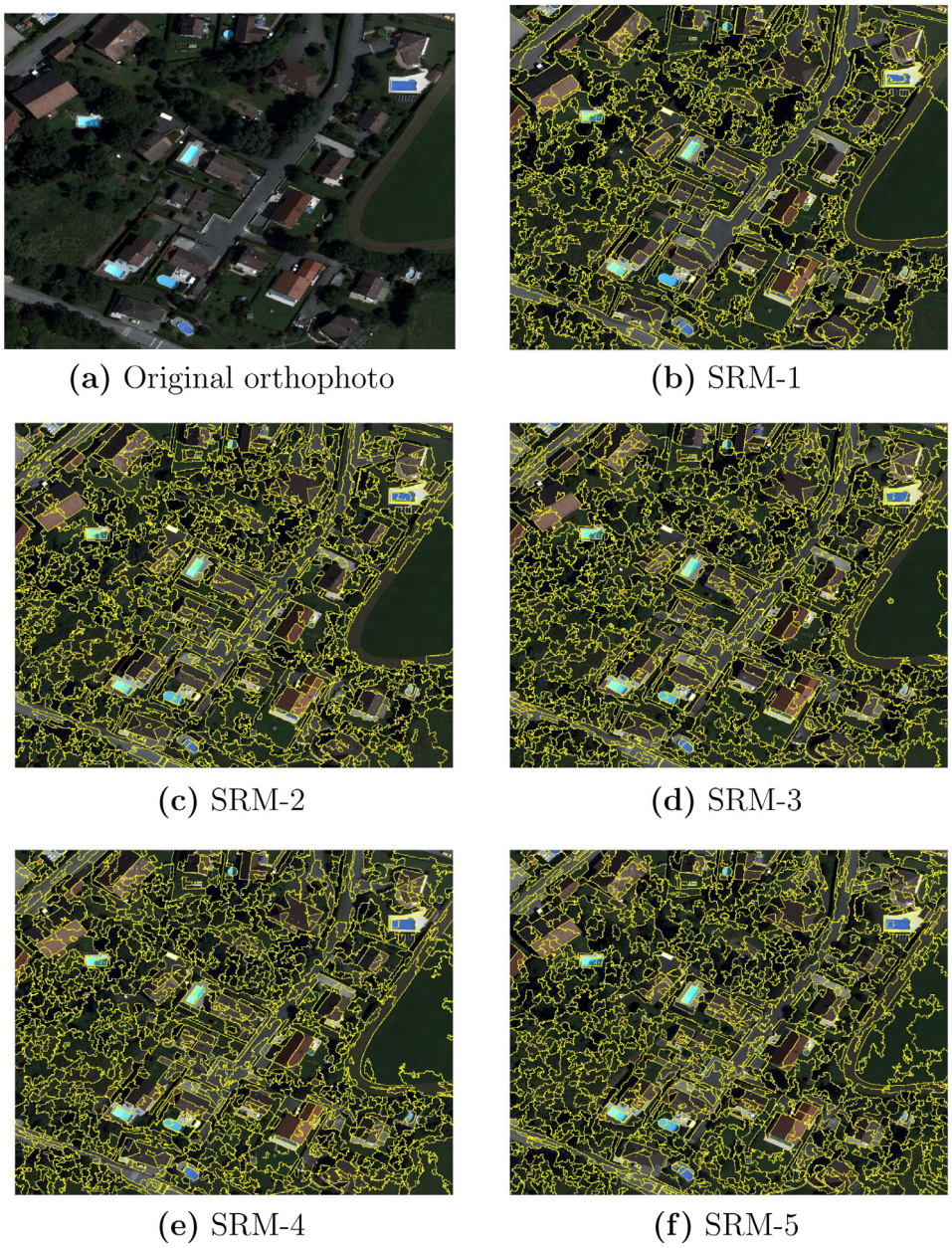


Fig. 5. SRM segmentation algorithm with different parameters.

+1. A value of +1 means a perfect prediction, 0 no better than random prediction and −1 means total disagreement between prediction and observation. The MCC is considered as being one of the best scores that can represent the confusion matrix of true and false positives and negatives by a single number. It should be noted that the recall and precision scores are also called "Completeness" and "Correctness", respectively.

The three groups of experiments adopt the following evaluation protocol. The whole set of orthophotos is divided evenly in two subsets: training subset and test subset. The segmented regions in the training subset are used to learn image descriptors and a given classifier. The segmented regions in the test orthophotos are used to evaluate the automatic detection using the recall, precision, F1 measure, accuracy, and MCC. This process is repeated 20 times and the statistical scores are averaged over these 20 splits.

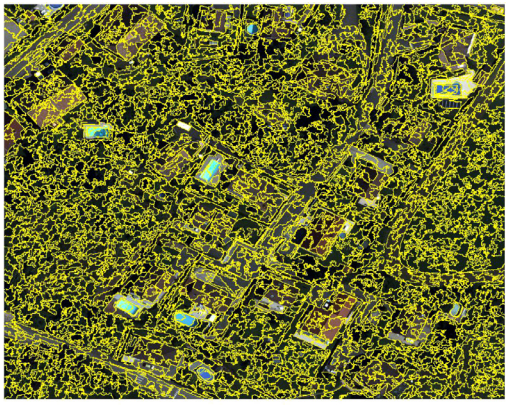
4.3. Influence of mixed regions

As segmentation methods, we consider (1) the SRM algorithm adopting five levels of segmentation, (2) the mean shift algorithm adopting two levels of segmentation, and (3) turbopixel (Levinshtein et al., 2009) adopting one level of segmentation. Table 1 summarizes the acronyms of these algorithms as well as their parameters. Figs. 5 and 6 illustrate the segmentation of an orthophoto obtained by the eight segmentation algorithms and levels. As can be seen, the quality and the number of regions depend on the segmentation technique and its parameters.

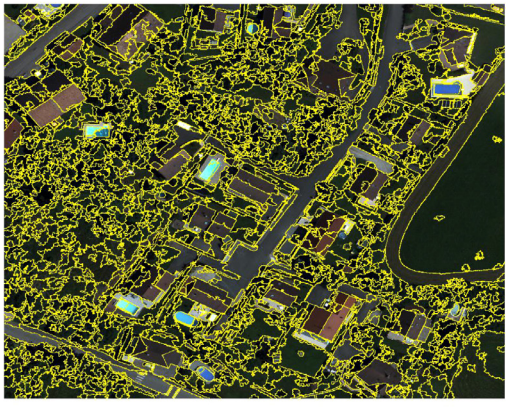
Before presenting the performance evaluation of the proposed framework, we study the segmentation algorithms' ability to provide segmented regions that are either purely building regions or purely background regions. Indeed, such behavior is very desired since mixed segmented regions (i.e., a region that contains pixels

Table 1
Segmentation algorithms: acronyms and parameters.

SRM-1	Statistical Region Merging	$Q = 2000$
SRM-2	Statistical Region Merging	$Q = 10{,}000$
SRM-3	Statistical Region Merging	$Q = 15{,}000$
SRM-4	Statistical Region Merging	$Q = 20{,}000$
SRM-5	Statistical Region Merging	$Q = 30{,}000$
MS-1	Mean shift	$h_s = 9$, $h_r = 1$
MS-2	Mean shift	$h_s = 9$, $h_r = 3$
Turbo	Turbopixel	$N = 7000$



(a) Mean Shift-1



(b) Mean Shift-2

(c) Turbopixel

Fig. 6. Mean shift and turbopixel segmentation results.

belonging to both classes) lead to pixel misclassification. This is due to the fact that the classifier will affect the same label to the whole pixels belonging to any segmented region. Thus, mixed regions will introduce some errors independently of the descriptor and of the classifier being used. Therefore the interest to quantify the capacity of each segmentation algorithm to provide non-mixed regions.

At first glance, it would be tempting to increase the number of segmented regions (i.e., using highly over-segmented images) in order to reduce the misclassification resulting from mixed regions. However, in the sequel, we will show that the use of highly over-segmented orthophotos is not a good idea. In order to investigate this, we have conducted the following experiment. Given a set of orthophotos together with their segmentation and the ground-truth building footprints, we are able to quantify the pixel misclassification without using image descriptors and classification. To this end, we proceed as follows.

First, all segmented regions are grouped into two categories: (i) pure regions, and (ii) mixed regions. Second, for every mixed region we compute the percentage of the intersection size to the region size, where the intersection is the region pixels that belong to a real building. By adopting a realistic assumption, we are able to quantify the misclassified pixels within any segmented orthophoto that has ground-truth building footprints. This assumption stipulates that the segmented regions will have the label of the ground truth label associated with the largest part of it.

This scheme is illustrated in Fig. 7. Fig. 7(a) illustrates the ground-truth building footprint in a given orthophoto together with four segmented regions. We can observe that regions R_1 and R_2 are pure regions, and regions R_3 and R_4 are mixed regions, i.e., their pixels belong to two different classes (building and background). Fig. 7(b) depicts the misclassified pixels (shown in blue) assuming that the classifier was able to correctly classify the pure regions and label the mixed regions with the ground-truth label associated with the largest part of it. In that example, the misclassification is given by the percentage of the misclassified pixels in the size of the orthophoto.

Fig. 8 illustrates the percentage of misclassified pixels as a function of the segmentation algorithm. The same figure illustrates the average number of segmented regions. We can draw the following observations. First, there is a high correlation between the number of segmented regions and the misclassification resulting from mixed regions. In other words, by increasing the number of regions the misclassification will increase. The segmentation algorithms that provide the lowest misclassification are SRM-1, SRM-2, and MS-2. This tends to confirm that a high level of over-segmentation will lead to a decrease in performance. This can be explained by the fact that whenever a large number of segmented regions is used, the delineation of these regions is not coinciding with the real delineation of buildings. Although the misclassification depicted in Fig. 8 was based only on ground-truth building maps and the segmented regions, it gives a hint on the behavior of segmentation algorithm when the descriptors are used. This will be confirmed in the sequel.

4.4. Segmentation algorithms and descriptors

In this section, we study the performance of different segmentation algorithms and different descriptors. As segmentation methods, we considered the eight segmentation algorithms shown in Table 1 and we use four descriptors: (i) color histogram (RGB), (ii) color histogram and LBP (hybrid descriptor=RGB+LBP), (ii) covariance matrix (COV), (iv) color histogram and covariance matrix (hybrid descriptor=RGB+COV).

Tables 2–5 illustrate the performance of the eight segmentation algorithms obtained with RGB, RGB+LBP, COV and RGB+COV,

Table 2
Average and 95% confidence interval of recall, precision, F1, accuracy and Matthews correlation coefficient (MCC) corresponding to a binary classification (pixel level) using color histograms. The averages correspond to 20 random splits training/test. The results are obtained with SVM classifier.

	Color histogram (RGB)				
Segmentation	Recall (%)	Precision (%)	F1 (%)	Accuracy (%)	MCC
SRM-1	82.72 ± 2.88	84.58 ± 2.13	83.50 ± 2.17	93.59 ± 1.41	0.80 ± 0.03
SRM-2	90.06 ± 1.90	79.87 ± 2.33	84.40 ± 1.15	93.86 ± 0.59	0.81 ± 0.01
SRM-3	89.22 ± 1.81	81.18 ± 2.85	84.76 ± 1.35	93.91 ± 0.92	0.81 ± 0.02
SRM-4	88.67 ± 1.65	81.84 ± 1.76	84.94 ± 1.36	94.30 ± 0.57	0.82 ± 0.02
SRM-5	90.22 ± 2.19	80.41 ± 2.01	84.77 ± 1.02	94.01 ± 0.68	0.81 ± 0.01
MS-1	90.46 ± 1.25	78.16 ± 2.86	83.57 ± 1.61	93.67 ± 0.89	0.80 ± 0.02
MS-2	90.76 ± 2.10	79.46 ± 3.63	84.32 ± 1.54	93.73 ± 0.82	0.81 ± 0.02
TURBO	89.53 ± 2.18	72.23 ± 3.07	79.63 ± 1.71	90.81 ± 1.60	0.75 ± 0.02

Table 3
Average and 95% confidence interval of recall, precision, F1, accuracy and Matthews correlation coefficient (MCC) corresponding to a binary classification (pixel level) using both color and LBP descriptors (color histograms with LBPs). The averages correspond to 20 random splits training/test. The results are obtained with the SVM classifier.

	Hybrid descriptor (RGB+LBP)				
Segmentation	Recall (%)	Precision (%)	F1 (%)	Accuracy (%)	MCC
SRM-1	87.07 ± 2.98	90.66 ± 1.49	88.65 ± 1.77	95.85 ± 0.66	0.86 ± 0.02
SRM-2	86.83 ± 1.30	89.25 ± 1.17	87.83 ± 0.55	95.56 ± 0.20	0.85 ± 0.01
SRM-3	86.42 ± 1.12	87.97 ± 1.23	87.00 ± 0.43	95.38 ± 0.27	0.84 ± 0.01
SRM-4	85.59 ± 1.25	88.73 ± 0.79	86.96 ± 0.60	95.40 ± 0.23	0.84 ± 0.01
SRM-5	87.02 ± 1.34	87.92 ± 1.18	87.24 ± 0.35	95.55 ± 0.21	0.85 ± 0.00
MS-1	86.00 ± 0.99	86.28 ± 1.03	85.99 ± 0.54	94.81 ± 0.24	0.83 ± 0.01
MS-2	87.80 ± 1.14	86.56 ± 1.82	86.81 ± 0.88	95.37 ± 0.42	0.84 ± 0.01
TURBO	84.17 ± 3.41	85.31 ± 1.99	84.41 ± 1.64	94.37 ± 0.33	0.81 ± 0.02

Table 4
Average and 95% confidence interval of recall, precision, F1, accuracy and Matthews correlation coefficient (MCC) corresponding to a binary classification (pixel level) using covariance descriptors. The averages correspond to 20 random splits training/test. The results were obtained with the SVM classifier.

	Covariance matrix descriptor (COV)				
Segmentation	Recall (%)	Precision (%)	F1 (%)	Accuracy (%)	MCC
SRM-1	87.31 ± 1.57	87.23 ± 1.51	87.11 ± 0.79	94.79 ± 0.58	0.84 ± 0.01
SRM-2	85.65 ± 1.06	86.12 ± 1.08	85.73 ± 0.54	94.92 ± 0.20	0.83 ± 0.01
SRM-3	86.36 ± 0.74	86.13 ± 0.90	86.13 ± 0.41	94.93 ± 0.25	0.83 ± 0.01
SRM-4	86.52 ± 1.05	85.57 ± 1.34	85.88 ± 0.50	94.71 ± 0.24	0.83 ± 0.01
SRM-5	85.35 ± 1.41	86.78 ± 1.33	85.86 ± 0.47	94.80 ± 0.33	0.83 ± 0.01
MS-1	85.44 ± 1.31	85.73 ± 1.55	85.34 ± 0.43	94.61 ± 0.20	0.82 ± 0.00
MS-2	86.73 ± 0.92	88.24 ± 1.17	87.31 ± 0.60	95.28 ± 0.39	0.84 ± 0.01
TURBO	88.12 ± 1.62	83.45 ± 1.45	85.34 ± 0.61	94.27 ± 0.47	0.82 ± 0.01

Table 5
Average and 95% confidence interval of recall, precision, F1, accuracy and Matthews correlation coefficient (MCC) corresponding to a binary classification (pixel level) using hybrid descriptors (color histograms and covariance descriptors). The averages correspond to 20 random splits training/test. The results were obtained with the SVM classifier.

	Hybrid descriptor (RGB+COV)				
Segmentation	Recall (%)	Precision (%)	F1 (%)	Accuracy (%)	MCC
SRM-1	87.90 ± 2.09	91.57 ± 1.07	89.60 ± 1.38	96.03 ± 0.40	0.87 ± 0.01
SRM-2	88.12 ± 0.90	89.30 ± 0.90	88.60 ± 0.47	95.96 ± 0.22	0.86 ± 0.01
SRM-3	88.42 ± 1.15	88.58 ± 0.85	88.38 ± 0.54	96.04 ± 0.15	0.86 ± 0.01
SRM-4	88.43 ± 0.92	89.00 ± 0.93	88.62 ± 0.57	95.85 ± 0.19	0.86 ± 0.01
SRM-5	88.44 ± 1.18	88.62 ± 1.04	88.42 ± 0.70	95.81 ± 0.20	0.86 ± 0.01
MS-1	88.31 ± 0.96	87.64 ± 1.30	87.84 ± 0.57	95.53 ± 0.22	0.85 ± 0.01
MS-2	88.64 ± 0.75	88.66 ± 1.79	88.42 ± 0.98	95.84 ± 0.47	0.86 ± 0.01
TURBO	87.31 ± 3.97	86.19 ± 1.43	86.37 ± 1.80	94.82 ± 0.31	0.83 ± 0.02

respectively. The classifier used is a Support Vector Machine (SVM). In each table, the results correspond to 20 random splits training/test subsets. These tables depict the average and confidence interval of the recall, precision, F1 measure, accuracy and MCC. We stress the fact that the hybrid descriptors are obtained by simple concatenation of the feature vectors. The feature vector lengths of RGB, RGB+LBP, COV and RGB+COV, are respectively 4096, 4411, 276, and 4372. It is worth noticing that our evaluation is performed without any postprocessing of the detection process. In other words, we evaluate the classification results at pixel level without adding ad hoc post-processing schemes that can reduce the false positive rate.

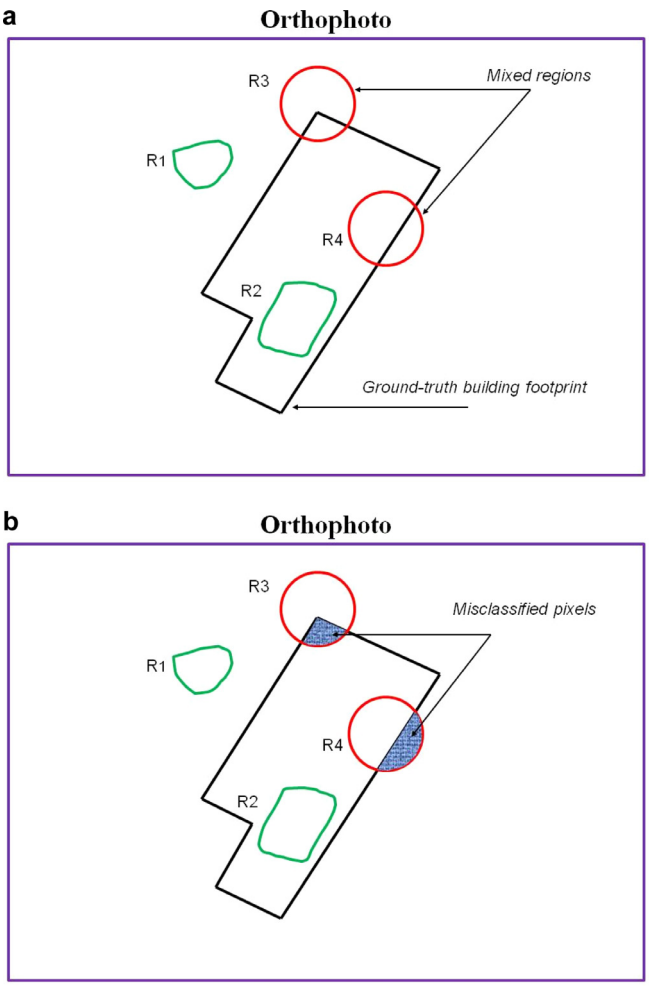


Fig. 7. Pixels misclassification due to the presence of mixed regions that are provided by a segmentation algorithm. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

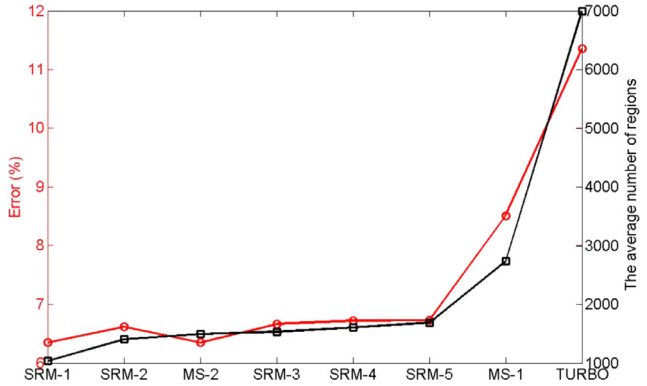


Fig. 8. Pixels misclassification of different segmentation algorithms due to the presence of mixed regions.

As can be seen, based on the five statistical scores, the best performance over the 32 combinations segmentation algorithm-descriptor is the one obtained with SRM-1 and the hybrid descriptor RGB+COV descriptor. Regarding the segmentation algorithms, the best ones are SRM-1, SRM-2 and MS-2. We recall that these algorithms are also found to be the best in the experiment shown in Fig. 8.

We can also observe that the best performances are obtained with the hybrid descriptors. It should be noticed that the worst performance is achieved with the color descriptor alone. For example, the 93.16% accuracy obtained with the color descriptor becomes 95.85% (color+LBP), 95.10% (covariance), and 96.03% (color+covariance), which correspond to an absolute improvement of about 2-3%. A 3% of an orthophoto may correspond to several tens of squared meters.

We stress the fact that the covariance descriptor produces very good results despite its low number of features (276), which is about 15 times less than the color descriptor (4096). For classifier training, it will be advantageous to use compact predictive variables. Thus, in practice the covariance descriptor can be a good trade-off for accuracy and training efficiency.

4.5. Classifiers performance

In this section, we study the performance of classifiers used to classify the segmented regions. We have used five classifiers: K Nearest Neighbor (K-NN) with ($K = 1$ and $K = 3$), SVM, linear Partial Least Square and non-linear Partial Least Square. A brief description of all of them is included below.

4.5.1. Classifiers

Instance Based Learning. Instance Based Learning relies on the K-NN paradigm (Aha, Kibler, & Albert, 1991; Mena-Torres, Aguilar-Ruiz, & Rodriguez, 2012). This is a distance based classifier that calculates a certain distance between the test sample to be classified and the samples in the training dataset. It then decides the class of the test sample based on the K nearest samples in the dataset it uses as a model. In our experiments we employ a KNN classifier with $K = 1$ and $K = 3$.

Support Vector Machines (SVMs). are supervised learning techniques deployed for classification and regression (Meyer, Leisch, & Hornik, 2003). For a binary classification problem, sample data are viewed as two sets of points in an N-dimensional space. The SVM will build a separating hyperplane in that space, one which maximizes the margin between the two sets of points. To calculate the margin, two parallel hyperplanes are estimated, one on each side of the separating hyperplane, which are "pushed up against" the two data sets. Intuitively, a good delineation of the two sets is reached by the hyperplane that has the largest distance to the close data points of both classes, since in general a large margin can lower the generalization error of the classifier (Meyer et al., 2003). SVMs were also extended to deal with datasets that are not linearly separable via the use of non-linear kernels. In our work, we use Gaussian kernels.

Partial Least Square (PLS). The Partial Least Squares (PLS) classifier or regressor (Rosipal & Kramer, 2006) is a statistical method that retrieves relations between groups of observed variables X and Y through the use of latent variables. It is a powerful statistical tool which can simultaneously perform dimensionality reduction and classification/regression. It estimates new predictor variables, known as components, as linear combinations of the original variables, with consideration of the observed output values. PLS is also extended to deal with non-linear datasets (Kramer & Braun, 2007). In our work, we use both types of PLS, with the number of latent components fixed to 50.

4.5.2. Results

Table 6 summarizes the average and confidence interval of the recall, precision, F1 measure, accuracy and MCC obtained by the five classifiers. The segmentation algorithm is the SRM-1 and the

Table 6
Average and 95% confidence interval of recall, precision, F1, accuracy and Matthews correlation coefficient (MCC) corresponding to a binary classification (pixel level) based on covariance descriptor. The segmentation method is the SRM-1.

	Covariance descriptor				
Classification	Recall (%)	Precision (%)	F1 (%)	Accuracy (%)	MCC
1-NN	81.45 ± 1.83	84.20 ± 1.29	82.61 ± 0.85	92.99 ± 0.66	0.78 ± 0.01
3-NN	85.15 ± 1.38	84.77 ± 1.86	84.78 ± 0.67	93.49 ± 0.55	0.81 ± 0.01
Linear PLS	82.55 ± 1.69	87.34 ± 1.12	84.77 ± 0.92	93.52 ± 0.57	0.81 ± 0.01
SVM	87.31 ± 1.57	87.23 ± 1.51	87.11 ± 0.79	94.79 ± 0.58	0.84 ± 0.01
Non-linear PLS	87.40 ± 2.15	91.70 ± 0.88	89.28 ± 1.09	96.29 ± 0.30	0.87 ± 0.01

Table 7
Average and 95% confidence interval of recall, precision, F1, accuracy and Matthews correlation coefficient (MCC) corresponding to a binary classification (pixel level) based on covariance descriptor. The results were obtained using a reduced training set (20% of the original training set). The segmentation method is the SRM-1.

	Covariance descriptor				
Classifier	Recall (%)	Precision (%)	F1 (%)	Accuracy (%)	MCC
1-NN	70.40 ± 4.46	81.41 ± 1.87	74.98 ± 2.82	91.30 ± 0.69	0.70 ± 0.03
3-NN	71.92 ± 5.87	78.31 ± 3.01	74.11 ± 3.68	90.47 ± 0.98	0.69 ± 0.04
Linear PLS	79.26 ± 3.83	84.11 ± 2.56	80.95 ± 1.70	92.02 ± 0.91	0.76 ± 0.02
SVM	74.33 ± 8.23	89.79 ± 1.71	79.39 ± 5.29	93.14 ± 1.21	0.77 ± 0.05
Non-linear PLS	83.81 ± 3.52	83.28 ± 1.97	83.09 ± 1.89	93.33 ± 0.69	0.79 ± 0.02

Table 8
Average and 95% confidence interval of recall, precision, F1, accuracy and Matthews correlation coefficient (MCC) corresponding to a binary classification (pixel level) using the covariance descriptor. The results were obtained using linear PLS with a reduced training set (20% of the original training set).

	Covariance descriptor				
Segmentation	Recall (%)	Precision (%)	F1 (%)	Accuracy (%)	MCC
SRM-1	76.66 ± 4.44	83.72 ± 1.77	79.27 ± 2.50	92.26 ± 0.74	0.75 ± 0.03
SRM-2	74.25 ± 4.68	86.11 ± 1.91	78.79 ± 2.31	93.19 ± 0.53	0.76 ± 0.02
SRM-3	76.89 ± 3.47	85.87 ± 1.49	80.54 ± 1.80	93.35 ± 0.47	0.77 ± 0.02
SRM-4	78.68 ± 3.50	84.60 ± 1.64	80.82 ± 1.38	93.38 ± 0.32	0.77 ± 0.01
SRM-5	77.98 ± 4.22	83.68 ± 2.12	79.84 ± 1.81	93.09 ± 0.49	0.76 ± 0.02
MS-1	79.65 ± 5.61	82.73 ± 2.54	79.98 ± 2.70	93.08 ± 0.55	0.77 ± 0.02
MS-2	74.44 ± 6.28	86.24 ± 2.41	78.50 ± 3.50	92.63 ± 0.70	0.75 ± 0.03
TURBO	78.71 ± 4.63	80.30 ± 2.40	78.41 ± 1.96	92.27 ± 0.54	0.74 ± 0.02

descriptor was provided by the covariance matrix. For each classifier, the results correspond to 20 random splits training/test.

Table 7 is similar to Table 6, however this time the training examples (the set of labeled descriptors) is reduced by 80 %, and the test set is kept fixed. We can observe that although the training set is now very reduced, the decrease in performance of all classifiers is small. This is a very desirable behavior from a machine learning point of view since training and testing can be more efficient. Table 8 depicts the performance of the linear PLS classifier for all segmentation algorithms using the reduced training data. The descriptor used here is the covariance matrix.

As can be seen, among all classifiers the non-linear PLS provided the best performances, followed by the SVM. This can be explained by the fact that both classifiers are non-linear.

Figs. 9 and 10 illustrate the automatic building detection obtained with the proposed framework after applying it on two orthophotos. The image segmentation is obtained with the SRM-1 algorithm. The region descriptor is given by RGB+COV. The classifier used is the SVM. The lower part of each figure illustrates the automatic building delineation (delimited by closed yellow contours) overlaid with the ground-truth delineation (shown in dark red). One can also appreciate the false positive and false negative regions. The former ones are blue pixels within a detected region, and the latter ones are red pixels outside of any detected region.

As can be seen, the developed framework demonstrates excellent accuracy in terms of building boundary extraction, i.e., the majority of the building roofs present in the image are detected with good boundary delineation. Indeed, the proposed framework gives reliable results for complex environments having buildings with red and non-red rooftop buildings and/or buildings having very complex shapes.

5. Discussions

In this paper, we have addressed building detection in orthophotos. The detection is achieved through the use of image segmentation and descriptor classification. As a case study, we have considered 200 buildings in aerial orthophotos. Evaluating the detection performance over the segmentation algorithm-descriptor pair showed that the SRM segmentation algorithm used with hybrid descriptors provided the best results. Nevertheless, good performances can also be obtained with other combination such as the ones using the covariance matrix descriptor. To the best of our knowledge, the presented work is the first one that uses the covariance matrix descriptor in order to detect and classify buildings. The advantage of that descriptor is twofold. First, it provides high detection performance with relatively few features. Second, it can be considered as a compact feature that combines several cues including color and texture. The proposed framework has several strengths that cannot be found jointly in other frameworks. These are as follows. First, the proposed framework releases the use of active sensors in the sense that only optical aerial images are used.

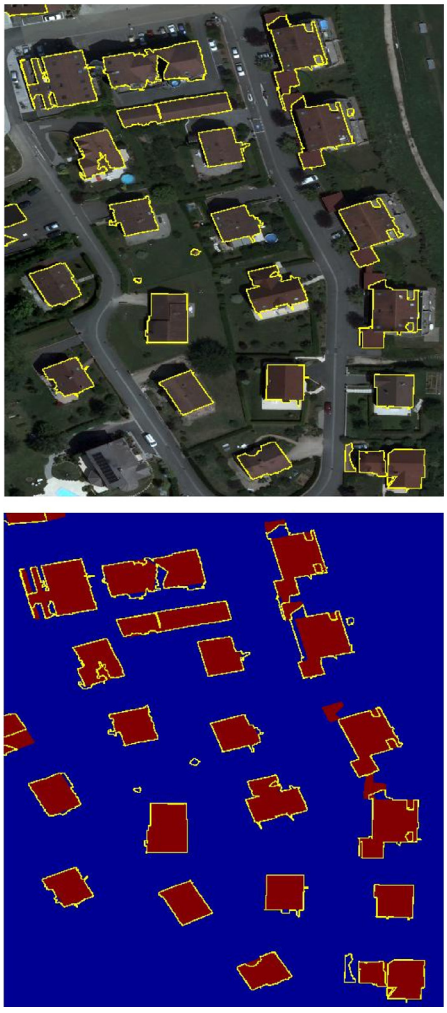


Fig. 9. Building detection results with SRM-1 segmentation algorithm and RGB+COV descriptor. The classifier is SVM. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

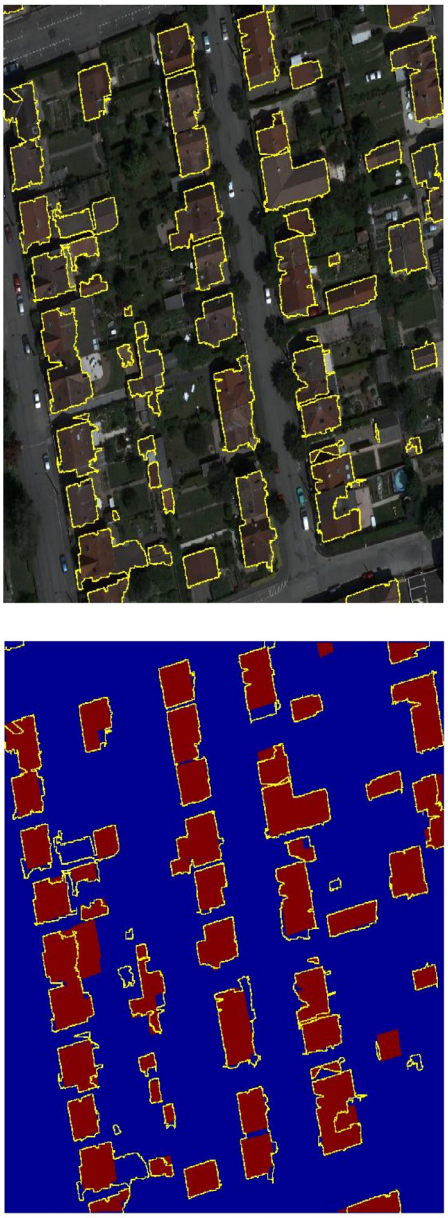


Fig. 10. Building detection results with SRM-1 segmentation algorithm and RGB+COV descriptor. The classifier is SVM. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

Furthermore, due to its flexibility, the proposed framework lends itself easily to small scale detection tasks as well as to large scale detection tasks. The framework can also be exploited by inexpensive acquisition systems such as drones. Secondly, the framework does not need a priori assumptions on the shape of building footprints, so all building shapes can be handled. Thirdly, despite the fact that the proposed method relies on a supervised scheme, at running time, there is no user interaction. Fourthly, thanks to the use of image over-segmentation, the proposed framework does not need to perform feature extraction nor decision at pixel level. The proposed framework (detection chain and performance study) is generic in the sense that it can be easily extended to the automatic detection of other categories of objects such as roads, vehicles, and vegetation. The main limitation of the proposed method is its confidence in the image segmentation algorithm. Indeed, the framework assumes that each provided segmented regions contains either a building part or a non-building part. In some segmented regions where building and surrounding regions have very similar appearances, the segmented patches/regions may contains pixels belonging to both categories. Therefore, since the classifier will assign a single label to the whole patch some pixels will be misclassified. It should be noted that this phenomenon is also affecting any other image-based building detection framework.

6. Conclusion

The paper introduced a generic framework (detection chain and performance study) that can be easily deployed in order to detect other types of objects in orthophotos and images. Unlike methods that rely on interactive image segmentation, our framework does not require, at running time, any user interaction or setting of initial algorithm parameters (a threshold of similarity for example). The proposed framework involved a supervised scheme in which off-line manual delineation and automatic segmentation are carried out to set trained models (labeled descriptors and classifiers). At running time, after an over-segmentation of the image, one can classify the segmented regions as object parts or background using region classification. In order to show its performance, the proposed method was applied to extract building roofs from orthophotos. This is a very challenging problem given the complexity of the objects contained in the orthophotos.

One main application of the method is the computation of building footprints in order to get full 3D modeling of buildings using Digital Surface Maps. Indeed, the use of 3D modeling to represent cities is growing in different applications such as urbanism and architectural conception, natural catastrophe management, traffic simulation, etc. The 3D models generation, which is achieved generally thanks to manual operations within specialized computer environments like 3D Studio Max, is now tending more and more to be automatized by jointly using optical images and 3D data. Indeed, automatic 3D model generation has become indispensable to take into account large geographical areas, while maintaining a significant level of detail and ensuring completion deadlines and acceptable costs. In this context, building detection plays a crucial role in automatic 3D building generation, especially for 3D cities modeling. Indeed, the extracted information (related to buildings) from aerial images allows building recognition using knowledge databases. Then, by refining/deforming 3D building models generated automatically from geographical data, we can obtain the desired 3D models. Moreover, the proposed framework can be very useful for building change detection, intelligent unmanned aerial vehicle navigation, and scene parsing in intelligent vehicles.

Future insights for further investigation are as follows. First, we envision the used of deep Neural Networks. This use can be carried in two ways: (i) as a classifier providing the label of the region, and (ii) as a descriptor extractor providing robust image descriptors about segmented regions. Second, we envision exploiting confidence score of the classifiers in order to predict possible regions that contains object of interest and its surrounding areas. The choice of more adapted color space could be also an interesting way to improve the results. Finally, the dimensionality reduction paradigms could be applied on the descriptors in order to improve both efficiency and accuracy.

References

Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., & Süsstrunk, S. (2012). SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 34*(11), 2274–2282.

Adams, R., & Bischof, L. (1994). Seeded region growing. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 16*, 641–647.

Aha, D., Kibler, D., & Albert, M. (1991). Instance-based learning algorithms. *Machine Learning, 6*, 37–66.

Ahonen, T., Hadid, A., & Pietikäinen, M. (2006). Face description with local binary patterns: Application to face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 28*(12), 2037–2041.

Arbelaez, P., Maire, M., Fowlkes, C., & Malik, J. (2011). Contour detection and hierarchical image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 33*, 898–916.

Bereta, M., Karczmarek, P., Pedrycz, W., & Reformat, M. (2013). Local descriptors in application to the aging problem in face recognition. *Pattern Recognition, 46*, 2634–2646.

Boykov, Y., & Jolly, M. (2001). Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images. In *Proceedings of IEEE international conference on computer vision*.

Comaniciu, D., & Meer, P. (2002). Mean shift: A robust approach toward feature space analysis. *IEEE Transaction on Pattern Analysis and Machine Intelligence, 24*, 603–619.

Di-Zenzo, R. (1986). A note on the gradient of a multiimage. *Computer Vision, Graphics and Image Processing, 33*, 116–125.

Friedland, G., Jantz, K., & Rojas, R. (2005). SIOX: simple interactive object extraction in still images. In *Proceedings of IEEE international symposium on multimedia*.

Gevers, T., & Smeulder, A. (2000). PicToSeek: Combining color and shape invariant features for image retrieval. *IEEE Transaction on ImageProcessing, 9*, 102–119.

Huang, D., Shan, C., Ardabilian, M., & Wang, Y. (2011). Adaptive particle sampling and adaptive appearance for multiple video object tracking. *IEEE Transaction on Systems, Man, and Cybernetics, Part C: Applications and reviews, 41*(6), 765–781.

Kramer, N., & Braun, M. (2007). Kernelizing PLS, degrees of freedom, and efficient model selection. In *Proceedings of international conference on machine learning* (pp. 441–448).

Kumar, S., & Hebert, M. (2003). Man-made structure detection in natural images using a causal multiscale random field. In *Proceedings of IEEE international conference on computer vision and pattern recognition* (pp. 119–126).

Levinshtein, A., Stere, A., Kutulakos, K., Fleet, D., Dickinson, S., & Siddiqi, K. (2009). Turbopixels: Fast superpixels using geometric flows. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 31*(12), 2290–2297.

Liasis, G., & Stavrou, S. (2016). Building extraction in satellite images using active contours and colour features. *International Journal of Remote Sensing, 37*(5), 1127–1153. doi:10.1080/01431161.2016.1148283.

McGuinness, K., & O'Connor, N. E. (2010). A comparative evaluation of interactive segmentation algorithms. *Pattern Recognition, 43*, 434–444.

Mena-Torres, D., Aguilar-Ruiz, J., & Rodriguez, Y. (2012). An instance based learning model for classification in data streams with concept change. In *Proceedings of the 11th Mexican international conference on artificial intelligence (MICAI), 2012* (pp. 58–62). doi:10.1109/MICAI.2012.22.

Meyer, D., Leisch, F., & Hornik, K. (2003). The support vector machine under test. *Neurocomputing, 55*, 169–186.

Ngo, T.-T., Collet, C., & Mazet, V. (2015). Automatic rectangular building detection from VHR aerial imagery using shadow and image segmentation. In *Proceedings of 2015 IEEE international conference on image processing (ICIP)* (pp. 1483–1487). doi:10.1109/ICIP.2015.7351047.

Nock, R., & Nielsen, F. (2004). Statistical region merging. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 26*(11), 1452–1458.

Ojala, T., Pietikäinen, M., & Maenpaa, T. (2002). Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 24*, 971–987.

Ok, A., Senaras, C., & Yuksel, B. (2013). Automated detection of arbitrarily shaped buildings in complex environments from monocular VHR optical satellite imagery. *IEEE Transactions on Geoscience and Remote Sensing, 51*(3), 1701–1717.

Quang, N. T., Thuy, N. T., Sang, D. V., & Binh, H. T. T. (2015). An efficient framework for pixel-wise building segmentation from aerial images. In *Proceedings of the sixth international symposium on information and communication technology, SoICT 2015* (pp. 282–287). New York, NY, USA: ACM. doi:10.1145/2833258.2833272.

Ren, X., & Malik, J. (2003). Learning a classification model for segmentation. In *Proceedings of IEEE international conference on computer vision* (pp. 10–17).

Roerdink, J. B. T. M., & Meijster, A. (2001). The watershed transform: Definitions, algorithms and parallelization strategies. *Fundamenta Informaticae, 41*(1–2), 187–228.

Rosipal, R., & Kramer, N. (2006). *Subspace, latent structure and feature selection techniques* (pp. 34–51). Springer.

Rother, C., Kolmogorov, V., & Blake, A. (2004). Grabcut – Interactive foreground extraction using iterated graph cuts. In *Proceedings of ACM transactions on graphics (SIGGRAPH'04), New York, NY, USA*.

Saito, S., Yamashita, T., & Aoki, Y. (2016). Multiple object extraction from aerial imagery with convolutional neural networks. *Journal of Imaging Science and Technology, 60*(1), 010402-1–010402-9.

Shorter, N., & Kasparis, T. (2009). Automatic vegetation identification and building detection from a single nadir aerial image. *Remote Sensing, 1*(4), 731–757.

Sirmacek, B., & Unsalan, C. (2008). Building detection from aerial images using invariant color features and shadow information. In *Proceedings of 23rd international symposium on computer and information sciences, 2008. ISCIS'08* (pp. 1–5).

Sirmacek, B., & Unsalan, C. (2009). Urban area and building detection using sift keypoints and graph theory. *Computer Vision and Image Understanding, 47*(4), 1156–1167.

Sun, M., Pang, L., Liu, H., Zhang, X., Ai, L., & He, S. (2016). *Geo-informatics in resource management and sustainable ecosystem*. Springer.

Takala, V., Ahonen, T., & Pietikäinen, M. (2005). Block-based methods for image retrieval using local binary patterns. In *Proceedings of Scandinavian conference on image analysis, SCIA*. In *LNCS: Vol. 3540*.

Tournaire, O., Brédif, M., Boldo, D., & Durupt, M. (2010). An efficient stochastic approach for buildings footprint extraction from digital elevation models. *ISPRS Journal of Photogrammetry and Remote Sensing, 65*, 317–327.

Tuzel, F. P., & Meer, P. (2006). A fast descriptor for detection and classification. In *Proceedings of European conference on computer vision* (pp. 589–600).

Unsalan, C., & Boyer, K. (2005). A system to detect houses andresidential street networks in multispectral satellite images. *Computer Vision and Image Understanding, 98*(3), 423–461.

Vakalopoulou, M., Karantzalos, K., Komodakis, N., & Paragios, N. (2016). Building detection in very high resolution multispectral data with deep learning features. *Technical Report, hal 01264084*. HAL.hal.archives-ouvertes.fr

Vincent, L., & Soille, P. (1991). Watersheds in digital spaces. An efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 13*(6), 583–598.

Wang, O., Lodha, S. K., & Helmbold, D. P. (2006). A bayesian approach to building footprint extraction from aerial lidar data. In *Proceedings of international symposium on 3d data processing, visualization, and transmission*.

Wolf, L., Hassner, T., & Taigman, Y. (2008). Descriptor based methods in the wild. In *Faces in real-life images workshop in ECCV*.

Wu, J., Xu, J., Zhao, J., Li, N., & Xiang, S. (2015). Comparison of several features of building detection in remote sensing image. In *Proceedings of international conference on mechatronics and industrial informatics*.