

Article

# Autonomous Building Detection Using Edge Properties and Image Color Invariants

Ali J. Ghandour <sup>1,\*</sup>  and Abdelkarim A. Jezzini <sup>2</sup>

<sup>1</sup> National Council for Scientific Research, CNRS, Zahia Salmame street, Jnah, P.O. Box 11-8281 Beirut, Lebanon

<sup>2</sup> Department of Computer and Communication Engineering, Islamic University of Lebanon, P.O. Box 30014 Beirut, Lebanon; abedelkarim.j@gmail.com

\* Correspondence: aghandour@cnrs.edu.lb

Received: 20 February 2018; Accepted: 18 April 2018; Published: 1 May 2018



**Abstract:** Automated building extraction from high-resolution satellite imagery is a challenging research problem, and several issues remain with respect to the variety of variables to be accounted for. In this paper we present an approach for building detection using multiple cues. We use the shadow, shape, and color features of buildings to propose our approach, known as Building Detection with Shadow Verification (*BDSV*). *BDSV* has three main pillars, which are: (1) tile building detection (*TBD*) to detect roof tile buildings; (2) flat building detection (*FBD*) to detect non-tile flat buildings according to shape features; and (3) *results fusion* used to fuse and aggregate results from previous blocks. Analyses performed over different study areas reveal high quality percentage and precision metrics, exceeding 95%. Performance analysis over the Sztaki–Inria and Istanbul datasets shows that *BDSV* outperforms benchmark algorithms.

**Keywords:** building extraction; remote sensing; shadow detection; hypothesis and validation; object identification

## 1. Introduction

Remote sensing is the science and art of obtaining information about an object, area, or phenomenon through the analysis of data acquired by a device that is not in contact with the subject under investigation. Object extraction from remote sensing acquired data has been an important research topic in computer vision for many years. Some useful applications of this subject include map creation, urban city planning, and land use analysis. For instance, building extraction and delineation is one of the most important tasks in map feature extraction. Obviously, the automation of such tasks can lead to greater productivity, resulting in reduced timelines for map production.

High-resolution satellite imaging sensors provide a new data source for building extraction that is of low cost or even free of charge. The high spatial resolution of the imagery reveals very fine details in urban areas and greatly facilitates the classification and extraction of urban-related features such as roads.

Many techniques have been developed to extract buildings from high-resolution satellite imagery. These techniques can be classified, according to the need for involvement of human experts, into three main categories as follows: (1) manual; (2) semi-autonomous; and (3) autonomous.

Manual techniques for the extraction of buildings from high-resolution satellite imagery are slow and require huge amounts of time. Thus, semi-autonomous and autonomous techniques have been proposed to improve the speed and utility of various applications such as urban map production.

One can easily observe that high-resolution satellite imagery contains two categories of buildings: (1) buildings with sloped roof tiles; and (2) non-tile flat roof buildings. Most existing studies and

proposed techniques in the literature focus on the detection of either roof tile buildings, or non-tile flat buildings which have standard rectangular structures. It is rare to find studies on both types.

The contributions of this paper are as follows. Firstly, we present a thorough review and classification of all available techniques for building detection using satellite images. Secondly, we introduce an autonomous and highly accurate approach for roof tile and flat building detection, known as Building Detection with Shadow Verification (*BDSV*). This approach depends on both color and shape information extracted from satellite/aerial images. Color information is used to detect roof tile buildings using our defined color invariant threshold.

Shape information is based mainly on edge detection followed by line generation technique where generated lines are grouped into: (1) parallel and (2) perpendicular line sets. Non-tile flat building candidates are identified from the treatment of these two line sets. In addition, shadow information is used to verify the generated building hypotheses. Our proposed approach to detect and verify the presence of both roof tile buildings and non-tile flat buildings is robust and accurate due to the fusion of results obtained from both algorithms.

Moreover, we benchmarked our proposed approach with state-of-the-art building detection algorithms such as Markov random field model segmentation [1], and the multi temporal marked point processes proposed in [2]. Results reveal that *BDSV* records the highest overall quality percentage (92%), in addition to the lowest execution time.

The remainder of this paper is organized as follows. Section 2 presents a review of published work on building detection, including: (1) edge-based detection techniques; (2) object-based detection techniques; and (3) thresholding-based detection techniques. Section 3 introduces the framework of our proposed (*BDSV*) approach and discusses in depth both the roof tile and non-tile flat building detection algorithms that constitute the main components in *BDSV*. The fine-tuning of various parameters is shown in Section 3. Section 4 presents performance analysis with benchmarking against state-of-the-art existing algorithms over the SztaKi-Inria and Istanbul building datasets using the RGB color format with 1.5 m/pixel and 0.3 m/pixel resolution, respectively. Finally, conclusions are given in Section 5.

## 2. Literature Review

Building extraction from high-resolution satellite imagery is an active research topic in the photogrammetry and computer vision fields. Comprehensive surveys of research in this area can be found in [3–5].

Considering both radiometry and geometry, many of these algorithms are edge-based detection techniques [6–8] consisting of linear feature detection and grouping for parallelogram structure hypotheses extraction [9]. In addition to linear feature detection approaches, other edge-based detection approaches rely on building polygons verification using the geometric structure of buildings [7], shadow [6,10–12], and illuminating angles [6], among others.

Park et al. [13] use a rectangular building model to search and find missing lines relying on detected lines by creating pairs of antiparallel lines over the roof using a line-rolling algorithm. A rectangular shape model can also be estimated using shadow information [14], implementing a novel Markov random field (MRF)-based region-growing segmentation technique. An iterative classification is applied, and the final buildings are determined using a recursive minimum bounding rectangle.

The dominant orientation angle in a building cluster is studied in [15], where local Fourier analysis is used to generate a building unit shape (BUS) space by recursive partitioning of regions using a hyperline in two-dimensional (2D) image space. A seeded BUS then searches for its neighbors and is grown until the predefined homogeneous criteria are satisfied.

Providing an approximate position and shape for candidate building objects is explored by Lee et al. [16]. This method is based on classification on multispectral IKONOS imagery. Fine extraction is then carried out in the corresponding panchromatic image through ISODATA segmentation and squaring based on the Hough transform.

Segl and Kaufmann [17] combined both supervised and unsupervised image segmentation for detection of buildings in suburban areas. A series of image classification results was generated by applying the thresholding technique with different values within a pre-defined range. The buildings were classified by shape matching with the database model. If the correct shape parameter is greater than an optimal threshold, then the object is classified as a building unit.

A region-growing algorithm on gray-scale images followed by using the edge map for the building presence verification is implemented in [18]. Another approach is discussed in [19], where the authors propose a low-level solution that effectively merges bi-dimensional and three-dimensional information to detect and analyze edges in satellite images.

A different group of building detection techniques is based on the combination of imagery and three-dimensional data. The use of two complementary sources of data usually improves the results. Most of these studies are based on two main approaches: object-based classification techniques and thresholding-based detection techniques.

In object-based classification techniques, image objects are created using automatic segmentation techniques. Then, the objects are characterized by means of spectral and height features, among others [20,21]. Approaches based on thresholding consist of the application of a threshold value to the normalized digital surface model (*nDSM*) to discriminate buildings and vegetation, combined with a threshold applied to the normalized difference vegetation index (*NDVI*) image in order to mask the vegetation. Then, buildings can be accepted or rejected according to different conditions, such as size and shape [18,22], spectral values [23], or texture features [24,25].

A thresholding algorithm with the application of the *nDSM* is implemented in [26], with a subsequent correction of the shape of the detected building based on alignments extracted from the imagery. Moreover, height data, active contours, and spectral information to extract the buildings are discussed in [27].

In [28], the authors propose a novel approach which depends on shadow information and color features to detect buildings. Finally, building shapes are determined using a novel box-fitting algorithm. Moreover, color information is also introduced in [29] where the author apply a segmentation technique on the HSV color space in addition to edge features in order to extract buildings.

Besides the discussed building detection techniques, another approach based on a novel probabilistic model is presented in [2]. The author introduces a new method which integrates building extraction with change detection in remotely-sensed image pairs. The method incorporates object recognition and low-level change information in a joint probabilistic approach. The accuracy is ensured by Bayesian object model verification, showing high accuracy percentages over several datasets.

The authors in [1] propose a novel segmentation algorithm based on a Markov random field model. In addition, they present a novel feature-selection approach useful for detecting both roads and buildings. Experimental results reveal that the proposed framework is solid, robust, and unsupervised, with promising quantitative and qualitative results compared to other building detection methods.

On the other hand, it is observed that principal component analysis (PCA) is also utilized in building detection tasks. The authors in [30] propose a building detection algorithm from pan-sharpened satellite color images, through eliminating non-building objects. First of all, vegetation and shadow regions are eliminated using the normalized difference vegetation index (*NDVI*) and the YIQ color space intensity ratio, respectively. Secondly, roads are masked out using a thinning morphological operation. Finally, PCA is implemented, resulting in false object removal, and thus buildings are retained. The proposed algorithm performance analysis is conducted over several study areas with different properties.

Dong [31] presents an approach for building detection in high-resolution satellite images. This approach utilizes the automatic histogram-based fuzzy C-means (AHFCM) technique as an image segmentation task. Vegetation and shadow regions are masked out using *NDVI* and a ratio map in the HIS model, respectively. Finally, buildings are detected and classified through PCA.

### 3. BDSV Approach

Based on what we have reviewed in the previous section, we note that integrating multiple features such as color, geometrical structure, and reflectance shadow can improve the detection process reliability and robustness, especially in complex real scenes. In this section, we will discuss in depth our proposed approach for building detection known as Building Detection with Shadow Verification (*BDSV*), which is useful for extracting both roof tile and non-tile flat buildings.

Extraction of buildings with sloped roof tiles mainly depends on the color information contained in the image. Based on a color invariant defined as a color ratio later in Section 3.1.1, the roof tile building identification task can be accomplished. Non-tile flat building extraction is more complex since it depends on the geometrical shape of the buildings. This task can be done using the edge and line features of the rectangular structural formation of buildings in the image.

Finally, in order to verify candidate detected buildings, shadow information is integrated. Candidate buildings with close shadows will be considered as actual buildings, otherwise they will be neglected.

Figure 1 illustrates the overall block diagram of our proposed Building Detection with Shadow Verification (*BDSV*) approach, which is composed of three main pillars as follows:

- The tile building detection (*TBD*) algorithm: A proposed algorithm for the detection of roof tile buildings using color invariants.
- The flat building detection (*FBD*) algorithm: A proposed algorithm for the detection of non-tile flat buildings based on edge and line features.
- Results fusion: A fusion algorithm to integrate the results obtained by the algorithms above and select candidate buildings.

Figure 1 shows the result of applying *TBD* and *FBD* to the Jiyeh area in Mount Lebanon where both roof tile and non-tile buildings exist. *TBD* was able to successfully detect roof tile buildings marked with a blue circle with the green dot referring to the detected building centroid and the red dot representing the shadow centroid. A red rectangle is used to illustrate buildings detected by *FBD* which are mainly non-tile buildings and some of the roof tile ones.

Although *FBD* by design focuses on the detection of flat buildings, in the previous study area it was able to detect some roof tile buildings with rectangular shapes. Moreover, although *TBD* is focused by design on detecting roof tile buildings, theoretically it would be able to detect a flat building with a red-colored roof. The choice of algorithm names (*TBD* and *FBD*) is only meant to reflect the type of buildings upon which each algorithm is focused. Further details about both algorithms will be presented in the following sections.

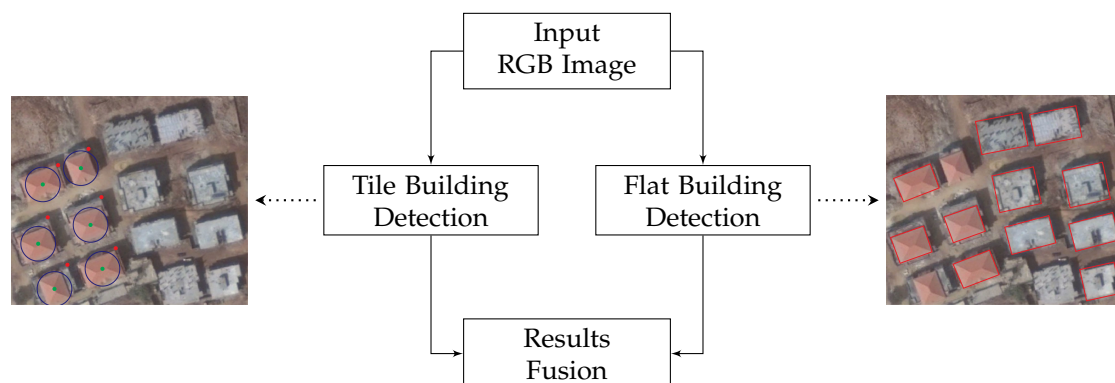


Figure 1. A Building Detection with Shadow Verification (*BDSV*) block diagram.

### 3.1. The Tile Building Detection (TBD) Algorithm

Roof tile building detection algorithm is mainly based on the color features extracted from satellite/aerial image. For the scope of this work, we focus on red roof tiles, which are commonly available in our country and the region. The proposed technique can be easily extended to account for other tile colors.

A new color invariant to detect vegetation areas has been suggested in [32]. In this work, we define a color ratio ( $CR$ ) related to red roof tile building, and then a thresholding technique is applied using the calculated  $CR$ . After that, different morphological operations are performed in order to verify the presence of buildings. Figure 2 illustrates the block diagram of this algorithm to be discussed thoroughly in the coming subsections.

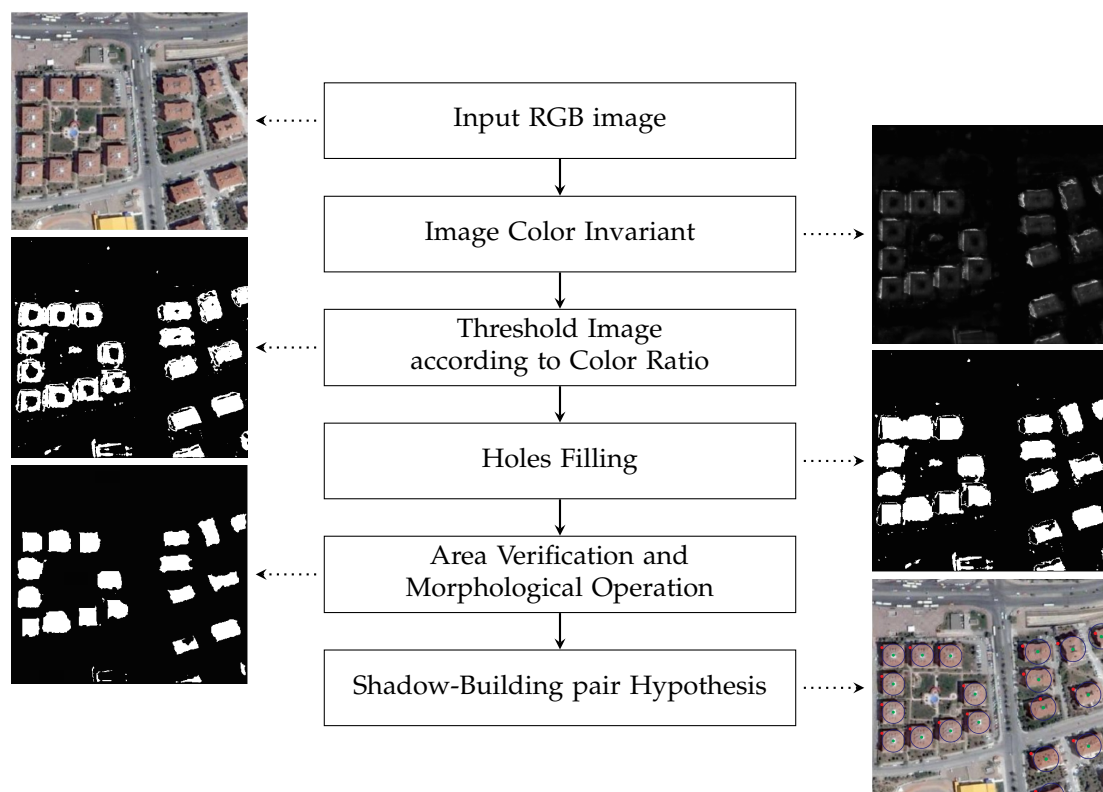


Figure 2. The tile building detection (TBD) algorithm.

#### 3.1.1. Image Color Invariant

Color invariants help to extract color properties of objects without being affected by imaging conditions. We extract the color information in the aerial image using a specific color index based on a study [33] where red and near-infrared bands are used. We follow here the same strategy to define a color invariant, using red and green bands of the color aerial image as shown in Equation (1) [28]:

$$ColorInvariantMaskImage = \frac{4}{\pi} \arctan \left( \frac{R - G}{R + G} \right) \quad (1)$$

where  $R$  stands for the red band and  $G$  stands for the green band of the color image. This color invariant has a value of unity for red-colored objects independent of their intensity values. Similarly, it has a value of minus unity for green-colored objects in the image. The fraction aims to increase the contrast of red objects within the image. As discussed in [33], the  $\arctan$  is a nonlinear transformation and the normalization by  $\pi/4$  is used to give values between  $-1$  and  $1$ .

Generally, the most popular roof tiles used in our country and the region are red, so it can be easily segmented using the proposed color ratio with great accuracy. *TBD* can be easily extended to detect other roof tile colors like blue and green.

### 3.1.2. Image Thresholding and Hole Filling

Image thresholding is the principle of applying a useful operation in order to separate out the regions of the image corresponding to objects in which we are interested from the regions of the image that correspond to background. Thresholding often provides an easy and convenient way to perform this segmentation on the basis of the different intensities or colors in the foreground and background regions of an image.

Image thresholding results in a binary image that consists mainly of two regions: objects of interest and the background region. There are a lot of techniques used in image processing, but the common thread among these different techniques is that all of them use a threshold as an input for the thresholding operation. For the sake of tile building detection (*TBD*), we use the color ratio image generated in the previous step as a input image to the local thresholding technique, which transforms the color ratio image into a binary image using a *color invariant* threshold as depicted in Figure 2. The obtained binary image contains the regions of interest (red roof tile buildings), but these regions are not closed regions, i.e., there are some holes in each object. Hence, in order to eliminate these holes and get enclosed building candidates, we must apply traditional hole-filling technique.

### 3.1.3. The Area Verification and Shadow–Building Pair Hypothesis

After applying the holes filing step, we get a resultant image that contains building candidates and some non-building objects that have the same color as buildings rooftops. In order to eliminate non-building objects, we apply a disc erosion morphological operation. The structuring element (SE) disc size is calculated based on the average area of building candidates since in general, buildings within the same region usually have a similar size.

Moreover, building candidates are verified according to the area criterion (*building area*), which is represented by the number of pixels per building. This area is defined as the average area of a building in the study area.

The final operation in the *TBD* algorithm is to eliminate falsely detected building candidates. This is achieved by calculating the distance between each building candidate centroid represented by a green dot, and its relative shadows centroid represented by a red dot. This distance must be below a pre-defined *building–shadow distance* threshold. Any building candidate with no relative shadow that adheres to this hypothesis is eliminated. For illustration purposes, *TBD*-detected buildings are delineated by a blue circle and *FBD*-detected buildings will be highlighted using a red rectangle.

It is worth mentioning that shadows can be segmented using any robust algorithm from the literature. For the scope of this work, we relied on HSV shadow detection technique proposed in [34]. Since we rely on the shadow centroid, the time of day where the image is taken is not of high impact. However, if the building shadow is missing, then it will result in a false positive.

## 3.2. The Flat Building Detection (*FBD*) Algorithm

In this section, we will discuss our proposed algorithm, known as flat building detection (*FBD*), used to detect all types of buildings that have the regular rectangular shape, but also buildings that have at least one right angle. *FBD* depends mainly on edge detection and line generation techniques. Resultant lines need to be filtered and processed to eliminate non-relevant lines, only keeping those representing building boundaries.

In our line-merging routine, we first divide detected lines into two classes: (1) perpendicular lines and (2) parallel lines. In the first class, for treatment we consider that every right angle is a building candidate, while in the second one we consider that every two parallel lines separated by a predefined

distance are building boundaries. Finally, the results obtained by the two classes are merged, and thus building candidates are obtained. Figure 3 shows the block diagram for this algorithm.

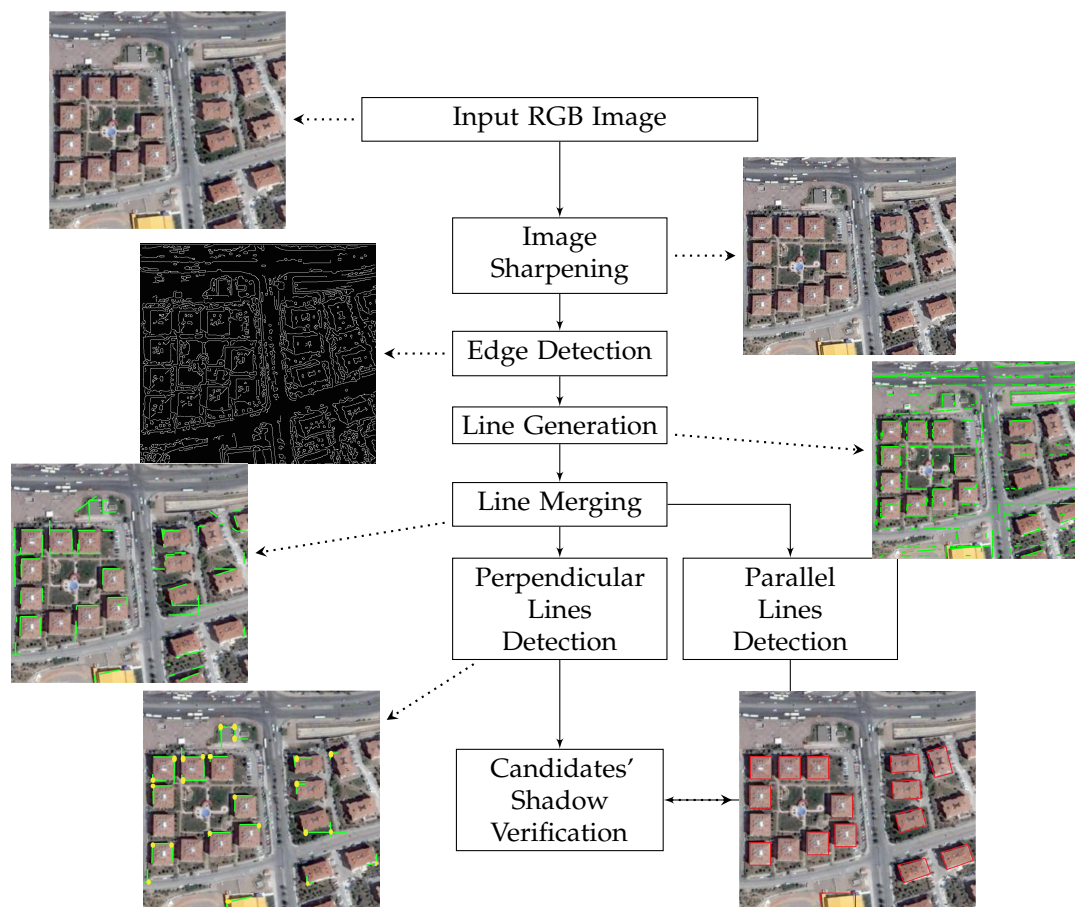


Figure 3. The flat building detection (FBD) algorithm.

### 3.2.1. Image Sharpening and Edge Detection

*FBD* mainly relies on the edge properties of buildings. That is why in the preprocessing stage of the proposed algorithm the sharpening filter is applied to show fine details and improve the edge detection performance.

Of the several known edge detection algorithms (Sobel, Canny, LoG, Roberts) [35], Canny edge detection is used in this study. The Canny algorithm provides several advantages over other edge detection techniques where it produces one pixel wide edges and connects broken lines. Those lines are crucial for the remaining blocks in this algorithm.

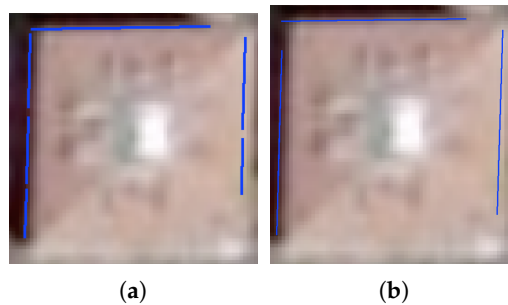
### 3.2.2. Line Generation and Merging

Detected edges in the previous block are the points with a zero second derivative in the input image. In order to detect buildings, these discrete points must be grouped using the Hough transform to form straight lines [36]. For each edge pixel, the Hough transform searches for other nearby pixels that will form straight lines taking into account a threshold distance so that pixels with a distance larger than the threshold value are discarded.

As shown in Figure 4a, generated lines are not connected and suffer from gaps and discontinuities. In our line-merging block, if the gap between the lines' segment is below a threshold (the *line-merging distance*), then these lines are merged. The line-merging procedure calculates the Euclidean distance between each line starting/ending point and all other line starting/ending points. Therefore, in every

iteration, we get a pair of two nearest lines, and we obtain also a new line which represents the two merged lines. Figure 4b shows a sample output of the introduced line-merging technique. By the end of the iterations, non-merged lines are eliminated, reducing the search space and improving the runtime of subsequent blocks.

It is worth noting that previous blocks were implemented using existing libraries in Matlab (off-the-shelf), however the line-merging block and all remaining ones were designed and implemented by the authors.



**Figure 4.** Figure 4a,b show the resultant output of the generated lines before and after the merging process, respectively.

### 3.2.3. Perpendicular Line Detection

Buildings usually have at least one right-angle, and thus we assume that every two perpendicular lines form a building's corner. In this block, we iterate over merged lines and calculate the angle between each pair (*perpendicular line angle*). If the calculated angle is between 80 and 100 degrees, then we consider that these two lines are perpendicular. Perpendicular line pairs are merged to form a right-angle. Each right-angle is considered as a seed point for a building candidate.

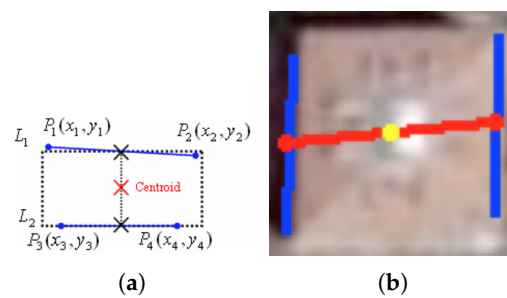
In order to calculate the angle between two lines, first we obtain the vector representation of each line. Using the vector representation, we calculate the norm of each vector. The last step before calculating the angle between the two vectors is to calculate their dot product. Finally, the angle between vectors  $\vec{V}_1$  and  $\vec{V}_2$  is given by :

$$\cos(\alpha) = \frac{\vec{V}_1 \cdot \vec{V}_2}{\|\vec{V}_1\| \cdot \|\vec{V}_2\|}. \quad (2)$$

### 3.2.4. Parallel Line Detection

As part of the parallel line pairs detection, two conditions must be verified: (1) the angle between the two lines must be less than 10 degrees, (*parallel line angle*) and (2) the distance between those two lines must be bounded within certain thresholds that represent the distance of building's parallel sides (*parallel line distance*). For each pair of detected parallel lines, a new virtual line connecting their midpoints is generated. The midpoint of the newly generated line is considered as a seed point for building candidate. Figure 5a,b show an illustration of the theoretical and real case implementation of generating building candidate seed point from a pair of parallel lines.





**Figure 5.** Figure 5a,b show the theoretical and real treatment of detected parallel lines, respectively.

### 3.2.5. Candidates' Shadow Verification

In this final step, we consider all the resulting right angles from the perpendicular line treatment, and all the resulting midpoints from the parallel lines treatment as seeds points for building candidates. It is worth mentioning here that a building structure could be detected via both parallel and perpendicular verification conditions where parallel line detection and perpendicular line detection complement each other.

For parallel line building candidates, we apply the same verification condition as for *TBD* buildings candidates through calculating the *building-shadow distance* distance, and any parallel lines building candidate with no relative shadow will be eliminated.

On the other hand, the distance between buildings candidates right corner and shadows centroid is referred to as the *right corner-shadow distance*. If this distance is less than a specific threshold, then the presence of a building is verified.

This verification block is important to eliminate non-building candidates such as parked cars or swimming pools. These structures could be detected as buildings due to their standard rectangular shape, however they usually do not have a corresponding shadow centroid.

### 3.3. Results Fusion

Integrating multiple cues and characteristics of buildings can leads to better results in building detection operation. The results obtained by both building detection algorithms (*TBD* and *FBD*) are combined using a simple OR operation. Detected buildings are marked by a tick that can be either red, green, or black, as illustrated in the following.

In the last *ResultsFusion* block, we define the following five states:

- State 1: true buildings detected by both *TBD*, and *FBD*. These buildings will be highlighted by red and green ticks in the results section.
- State 2: true buildings detected by *TBD*, but not by *FBD*. These buildings will be highlighted by red ticks only in the results section.
- State 3: true buildings detected by *FBD*, but not by *TBD*. These buildings will be highlighted by green ticks only in the results section.
- State 4: true buildings not detected by both *TBD* and *FBD*. These buildings will be highlighted by black crosses in the results section.
- State 5: false buildings detected by either *TBD* or *FBD*. These buildings will be highlighted by black ticks in the results section.

We use a simple logical OR operator for *ResultsFusion*, where candidates belonging to States 1, 2 and 3 are considered real buildings.

#### 4. Parameter Tuning

In this section, the numerical values of different thresholds used in *TBD* and *FBD* will be presented and discussed.

The *TBD* algorithm contains three thresholds: (1) the area condition threshold *building area* to eliminate falsely-detected objects; (2) the *color invariant* threshold by which the color invariant image is transformed into a binary image, where all pixels with values greater than the threshold are rounded up to one; and (3) the *building–shadow distance* threshold used for building candidate verification. This threshold represents the distance between building centroid and corresponding shadow centroid.

With respect to the *FBD* algorithm, five thresholds were defined in this manuscript. Firstly, in the line-merging routine, the minimum distance between two lines to be merged must be less than a pre-defined *line-merging distance* threshold. Secondly, *parallel line angle* represents the value of the angle between two lines in order to be considered as a pair of parallel lines. Thirdly, the distance that separates a pair of parallel lines as discussed in Figure 5 must belong to the *parallel line distance* threshold interval. Fourthly, the angle between two perpendicular lines must be bounded within the *perpendicular line angle*. Fifthly, the *right corner–shadow distance* represents the distance threshold between right corner building candidate and its relative shadow centroid. *TBD* and *FBD* sub-routines and relative thresholds are summarized in Table 1.

**Table 1.** BDSV operations.

<i>Algorithm</i>	<i>Operation</i>	<i>ThresholdName</i>
<i>TBD</i>	Image thresholding	Color invariant
	Area verification	Building area
	Shadow–building pair hypothesis	Building–shadow distance
<i>FBD</i>	Line merging	Line-merging distance
	Perpendicular line detection	Perpendicular line angle
	Parallel line detection	Parallel line angle
	Candidates' shadow verification	Parallel line distance
		Right corner–shadow distance

Two approaches can be followed to assign values for various thresholds: (1) a static approach where these values are manually assigned and tuned according to the environmental conditions of the study area; and (2) a dynamic unsupervised approach that relies on the existing study area to learn how to set these thresholds. For instance, the average area of all detected building candidates under consideration in the study area can be used to set the value of *building area*.

For the scope of this work, we will stick to the static approach where we manually assign the thresholds' values based on the study area environmental conditions. Future work will focus on devising unsupervised approach to learn how to set these thresholds. Table 2 presents a description of all used thresholds throughout this manuscript and corresponding values used over Istanbul dataset in Section 5.

Table 2. BDSV thresholds.

Algorithm	Threshold Name	Threshold Value	Threshold Description
<i>TBD</i>	Building area	400	Average candidate building area
	Color invariant	0	Threshold used with image color invariant mask
	Building–shadow distance	70	Distance used to verify building–shadow pair
<i>FBD</i>	Line-merging distance	10	Distance threshold used to merge a pair of lines
	Parallel lines angle	10	Angle threshold to consider a pair of lines as parallel
	Parallel line distance	>50 and <150	Distance threshold range between parallel lines pair forming a building
	Perpendicular line angle	>80 and <100	Angle threshold to consider a pair of lines as perpendicular
	Right corner–shadow distance	100	Distance threshold used between shadow centroid and right corner forming a building

## 5. Result Analysis

In this section, we will benchmark *BDSV* results with several state-of-the-art buildings datasets used in the literature. Performance metrics used in the Istanbul Dataset benchmark will be based on a set of parameters including the miss factor (MF), branching factor (BF), and the quality percentage (QP), for reporting algorithm performances. Many researchers use these definitions to report the efficiency of their systems. Based on these definitions, all objects in the image are categorized into four types.

- True positive (TP). Both manual and automated methods label the object belonging to the buildings regions.
- True negative (TN). Both manual and automated methods label the object belonging to the background.
- False positive (FP). The automated method incorrectly labels the object as belonging to the building regions.
- False negative (FN). The automated method does not correctly label a pixel truly belonging to the building regions.

To evaluate performance, the numbers of objects that fall into each of the four categories TP, TN, FP, and FN are determined, and the following measures are computed:

$$MissFactor(MF) = \frac{FN}{TP} \quad (3)$$

$$BranchingFactor(BF) = \frac{FP}{TP} \quad (4)$$

$$QualityPercentage(QP) = 100 \times \frac{TP}{TP + FP + FN} \quad (5)$$

The interpretation of the above measures is as follows. The branching factor is a measure of the commission error where the system incorrectly labels background objects as buildings. The miss factor measures the omission error where the system incorrectly labels buildings objects as background. Among these statistics, the quality percentage measures the absolute quality of the extraction and is the most important measure. To obtain 100% quality, the extraction algorithm must correctly label every building (FN = 0) without mislabeling any background objects (FP = 0).

Moreover, the following performance parameters will be used in evaluating the Sztaki-Inria dataset benchmark, in accordance with other works.

$$Recall = \frac{TP}{TP + FN} \quad (6)$$

$$F - Score = \frac{2 \cdot TP}{2 \cdot TP + FN + FP} \quad (7)$$

$$F1 - Score = 2 \cdot \frac{Pr \cdot Rc}{Pr + Rc} \quad (8)$$

$$Precision = \frac{TP}{TP + FP} \quad (9)$$

### 5.1. The Sztaki-Inria Dataset

In this section, *BDSV* will be benchmarked with various building detection algorithms studied in [2], where the authors propose a new framework for building detection and introduce a comprehensive analysis between their algorithm, and other state-of-the-art algorithms over the *Sztaki-Inria* dataset which includes eight study areas satellite images in RGB color format with a resolution of 1.5 m/pixel.

The benchmarked algorithms used are defined as follows:

- *SIFT* key point-based method [37]: a supervised building detection algorithm, where the input image must be characterized by a couple of template buildings which are used for training. Its main drawback is that for images containing a high variety of buildings may need a huge template library, which is hard to control.
- *Gabor* method [38]: detects buildings according to joint probability density functions of four different local feature vectors, and performs data and decision fusion in a probabilistic framework to detect building locations.
- *Edge-verification (EV)* method [28]: depends on edge information, where an edge map is produced. Fitting the building rectangles and verification of the proposals are based purely on the Canny edge map of the obtained candidate regions. For dense urban areas the processing time and the appearance of false edge patterns.
- *Segment-Merge (SM)* method [39]: considers building detection as a regional-level or image segmentation problem. This method assumes that buildings are homogeneous areas w.r.t. either color or texture. It can fail if the background and building areas cannot be efficiently separated with the chosen color or texture descriptors.
- *MPP* method [2]: is an object-change modeling approach based on multi temporal marked point processes, which simultaneously exploit low-level change information between the time layers and object level building descriptions to recognize and separate changed and unaltered buildings.
- *MRF* method [1]: is based on Markov random field model segmentation. This segmentation algorithm is based on class-driven vector data quantization and clustering and the estimation of the likelihoods given the resulting clusters. The main contribution depends on the estimation of likelihoods in order to reach higher accuracy percentages in the building detection operation.

The benchmarking process will be applied at three levels: (1) object-level performance analysis; (2) as a pixel-level performance benchmark; and (3) for computational time analysis. The results for SZADA1, SZADA2 and SZADA3 will be combined and reported in the following tables as SZADA. Similarly, results for COTE D'AZUR1 and COTE D'AZUR2 study areas will be merged and reported as COTE D'AZUR.

*BDSV* resultant output on Sztaki-Inria dataset is shown in Figure 6 where subfigures Figure 6a–h show the resultant output of applying *BDSV*, respectively, on the SZADA1, SZADA2,

SZADA3, Budapest, Bodensee, Normandy, COTE D'AZUR1, and COTE D'AZUR2 study areas. The resultant output shows the efficiency of *TBD* in roof tile building detection. Even though *FBD* records some false positives among different study areas, it succeeds in improving the overall accuracy by detecting several non-tile flat buildings.



**Figure 6.** Figure 6a–h show the resultant output of applying DBSV, respectively, on SZADA1, SZADA2, SZADA3, Budapest, Bodensee, Normandy, COTE D'AZUR1, and COTE D'AZUR2 study areas from the Sztaki–Inria dataset.

Object-level performance analysis is done over Sztaki–Inria dataset using missing objects (*MO*) and false objects (*FO*) performance metrics, in addition to the *F-Score* metric defined in Equation (7) which represents the overall accuracy of the algorithm.

Table 3 shows the object-level accuracy assessment results of benchmarking *BDSV* against the *SIFT*, *Gabor*, *EV*, *SM*, and *MPP* algorithms. It is worth mentioning that *BDSV* records the lowest *FO* and *MO* values, thus outperforming all the benchmarked algorithms with an overall F-score of 0.925.

**Table 3.** Object-level accuracy assessment of benchmarked algorithms over the Sztaki–Inria dataset. MO: missing objects; FO: false objects.

Study Area		Object-Level Performance											
Name	#obj	SIFT [37]		Gabor [38]		EV [28]		SM [39]		MPP [2]		BDSV	
		MO	FO	MO	FO	MO	FO	MO	FO	MO	FO	MO	FO
BUDAPEST	41	20	10	8	17	11	5	9	1	2	4	7	2
SZADA	57	17	26	17	23	10	18	11	5	4	1	4	0
COTE D’AZUR	123	55	9	12	24	14	20	20	25	5	4	5	5
BODENSEE	80	34	9	32	8	11	13	18	15	7	6	2	1
NORMANDY	152	69	14	24	14	18	32	30	58	18	1	17	5
Overall F-score		0.662		0.667		0.719		0.655		0.894		<b>0.925</b>	

The authors of [1] use pixel-level accuracy assessment to compare their unsupervised building and road classification algorithm against existing ones in the literature. Pixel-level performance benchmark and accuracy assessment rely on the *precision* ( $Pr$ ) metric defined earlier in Equation (9), *recall* ( $Rc$ ) shown in Equation (6), and the *F1-score* defined in Equation (8).

In Table 4, we benchmarked *BDSV* in addition to several algorithms. It is clearly shown that *BDSV* records the highest *precision* and *recall* values for all study areas, indicating the high efficiency of our proposed building detection approach. As a result, *BDSV* ranked first in terms of overall *F1-score* with a value equal to (0.913).

**Table 4.** Pixel-level accuracy assessment of benchmarked algorithms over the Sztaki–Inria dataset.

Study Area		EV [28]		SM [39]		MPP [2]		MRF [1]		BDSV	
Name	#obj	Pr	Rc	Pr	Rc	Pr	Rc	Pr	Rc	Pr	Rc
BUDAPEST	41	0.73	0.46	0.84	0.61	0.82	0.71	0.79	0.61	0.94	0.82
SZADA	57	0.61	0.62	0.79	0.71	0.93	0.75	0.74	0.67	1	0.80
COTE D’AZUR	123	0.73	0.51	0.75	0.61	0.83	0.69	0.76	0.67	0.95	0.91
BODENSEE	80	0.56	0.30	0.59	0.41	0.73	0.51	0.71	0.68	0.98	0.92
NORMANDY	152	0.60	0.32	0.62	0.55	0.78	0.60	0.72	0.58	0.96	0.88
Overall F1-score		0.524		0.640		0.725		0.718		<b>0.913</b>	

Finally, Table 5 shows the computational time of applying several benchmarked algorithms over each study area separately. *BDSV* outperforms all the benchmarked algorithms in recording the lowest computational time, which makes it more faster in execution even in study areas containing large number of buildings as in the case in the *NORMANDY* and *COTED’AZUR*.

**Table 5.** Computational time of the benchmarked algorithms over the Sztaki–Inria dataset.

Study Area	Computational Time (seconds)						
	SIFT [37]	Gabor [38]	EV [28]	SM [39]	MPP [2]	MRF [1]	BDSV
BUDAPEST	197.3	14.5	120.4	18.4	25.2	10.4	11.23
SZADA	200.9	49.8	30.3	89.1	31.5	42.7	15.23
COTE D’AZUR	416.0	57.7	324.3	47.2	68.5	23	13.66
BODENSEE	90.0	27.9	30.4	35.1	66.2	26.4	10.13
NORMANDY	236.7	67.2	109.6	72.3	46.3	36.2	16.76
Average	228.18	43.42	123	52.42	47.54	27.74	<b>15.97</b>

### 5.2. Istanbul Dataset

The authors of [28] propose a novel method for building detection by defining a set of parameters used in shadow detection, in addition to determining shapes of the buildings using edge information. They present the Istanbul dataset which includes 12 study areas: five of them contain only red roof tile buildings, while the remaining seven study areas contain both roof tile and non-tile flat buildings.

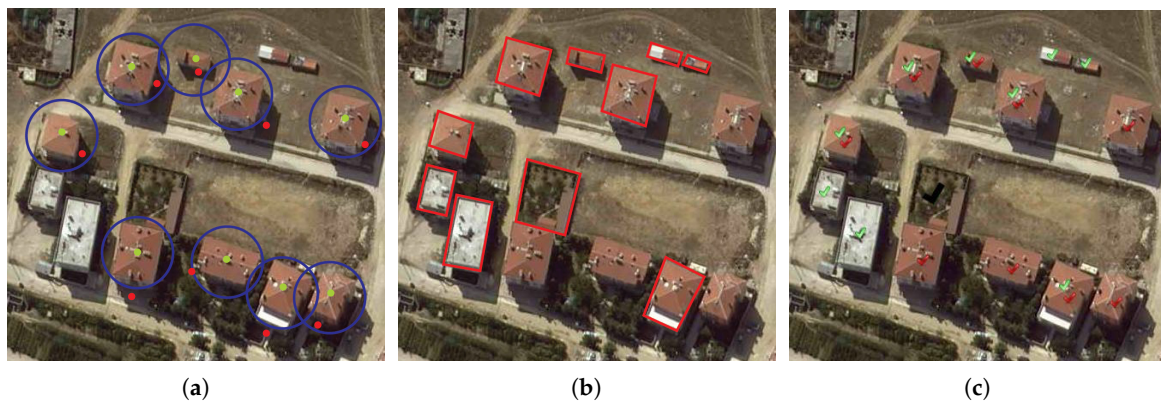
For the scope of this work, we will focus in particular on three study areas that contain both roof tile and non-tile buildings, since the remaining study areas have similar environmental characteristics. The chosen images are: (1) *Ist7* which includes buildings having various sizes; (2) *Ist8*, a suburban study area including vegetation zones; and finally, (3) *Ist10*, a study area that contains highway, cars, and trees, in addition to target buildings. The Istanbul dataset is described in Table 6. The aerial images used are in *RGB* color format with a 0.3 m/pixel resolution.

**Table 6.** Relevant image ground truth details of the Istanbul dataset .

Study Area	Roof Tile Building Count	Non-tile Flat Building Count	Total Count
Ist1	14	0	14
Ist2	8	0	8
Ist3	3	0	3
Ist4	28	3	31
Ist5	22	4	26
Ist6	7	0	7
<b><i>Ist7</i></b>	<b>11</b>	<b>2</b>	<b>13</b>
<b><i>Ist8</i></b>	<b>17</b>	<b>3</b>	<b>20</b>
Ist9	10	0	10
<b><i>Ist10</i></b>	<b>20</b>	<b>4</b>	<b>24</b>
Ist11	4	3	7
Ist12	1	5	6

The *Ist7* study area contains two non-tile flat buildings and the resultant images of applying *BDSV* are shown in Figure 7. As can be noted from Figure 7a, *TBD* detected 9 roof tile buildings, whereas *Ist7* Study Area contains 11 roof tile buildings. *TBD* did not detect the two buildings located on the right top of the *Ist7* study area, due to their small area as compared to the other buildings.

*FBD* detected 10 of 13 buildings, and it classified a garden wall as a building due to its rectangular shape as shown in Figure 7b. The *results fusion* in Figure 7c ensures that all buildings in the *Ist7* study area are detected with only one *FP* classified object. Therefore, we can observe that *BDSV* outperforms algorithm suggested in [28] for the *Ist7* study area.



**Figure 7.** Figure 7a–c show the resultant output of applying TBD, FBD, and results fusion, respectively, on the Ist7 study area.

Table 7 shows a detailed accuracy assessment of applying *BDSV* on the *Ist7* study area. Analysis of *Ist7* results reveal two important issues. First, for study areas which contains buildings with great variation in sizes, *TBD* will fail in detecting all colored rooftop buildings. Future work will focus on handling this drawback. Second, the garden detected as a building by *FBD* is due to the enclosure wall shadow. Usually, rectangular-shaped structures such as parking lots, swimming pools, tennis courts, and gardens will have no relative shadows in most cases.

**Table 7.** Accuracy assessment of applying *BDSV* on Ist7 that includes 13 building objects.

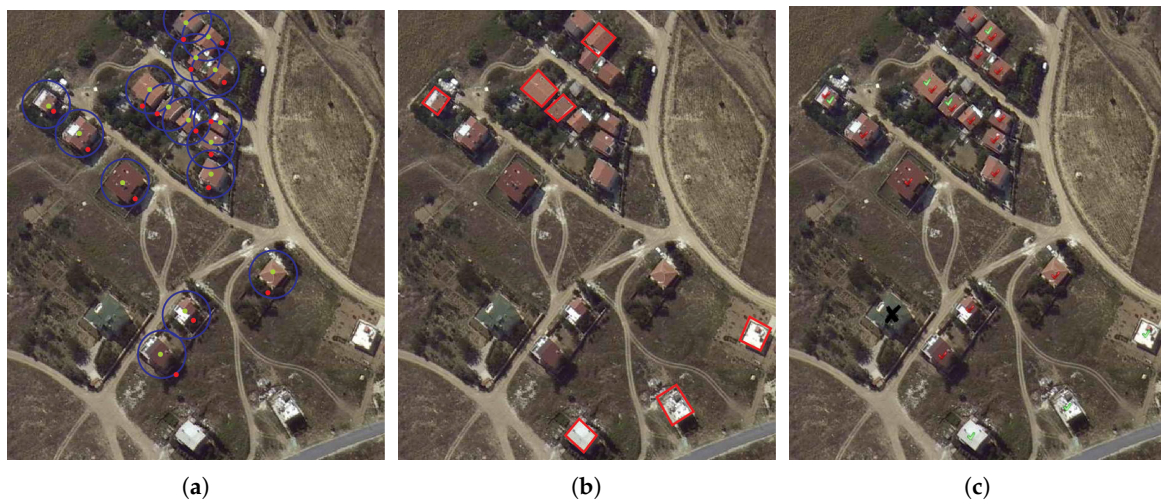
Algorithm	TP	FP	FN	BF	MF	QP
TBD	9	0	4	0	0.44	69.2%
FBD	9	1	4	0.11	0.44	64.2%
Results Fusion	13	1	0	0.07	0	92.8%

The *Ist8* study area has the same environmental conditions as the *Ist7* study area with much more complexity due to the non-uniform distribution of buildings. The results of applying *BDSV* on the *Ist8* study area are shown in Figure 8. *TBD* missed only one green-tiled roof building located in left-bottom corner of the study area, as seen in Figure 8a. This issue will be addressed in the future work since currently *TBD* focuses on red-rooftop buildings only. Output results reported in Table 8 reveals that *BDSV* has zero *FP* objects, while algorithm suggested in [28] scored six *FP* objects, as we will see later.

**Table 8.** Accuracy assessment of applying *BDSV* on Ist8, which includes 20 building objects.

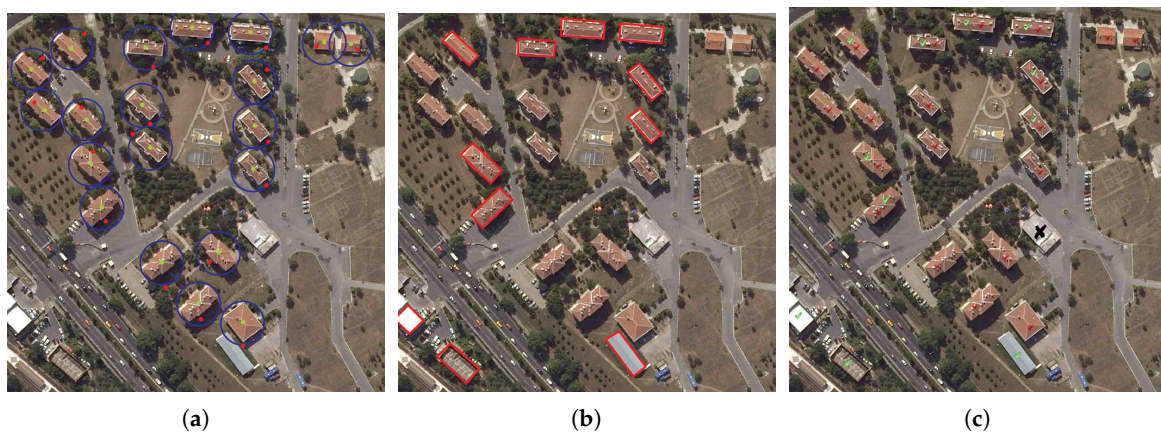
Algorithm	TP	FP	FN	BF	MF	QP
TBD	16	0	4	0	0.25	80%
FBD	7	0	13	0	1.8	35%
Results fusion	19	0	1	0	0.05	95%





**Figure 8.** Figure 8a–c show the resultant output of applying TBD, FBD, and results fusion, respectively, on the Ist8 study area.

*Ist10* is an urban area that contains additional objects classes such as cars and roads. *TBD* in Figure 9a succeeded in detecting all 16 red roof tile buildings, while *FBD* detected 11 buildings, as shown in Figure 9b. With results fusion in Figure 9c, *BDSV* records only one false negative,  $FN = 1$ . The accuracy results of the *Ist10* study area are tabulated in Table 9 where *BDSV* detected 23 buildings out of 24 with  $FP = 0$ , and the algorithm suggested in [28] detected 20 buildings with  $FP = 3$  as we will see next.



**Figure 9.** Figure 9a–c show the resultant output of applying TBD, FBD, and results fusion, respectively, on the Ist10 study area.

**Table 9.** Accuracy assessment of applying *BDSV* on Ist10, including 24 building objects.

Algorithm	TP	FP	FN	BF	MF	QP
TBD	20	0	4	0	0.2	83.3%
FBD	11	0	13	0	1.8	45.8%
Results fusion	23	0	1	0	0.04	95.8%

Table 10 shows a detailed comparison of *BDSV* and the algorithm suggested in [28] for the three images from Istanbul dataset. The *precision* metric defined in Equation (9) is utilized for this

comparison as reported in [28]. It is obvious that *BDSV* records higher precision for the three areas, which reveals the superiority of our proposed approach in detecting both roof tile and non-tile buildings. Moreover *BDSV* records lower *FP*, which indicates that *BDSV* have less commission error in labeling background objects as buildings.

Finally, Table 11 shows the frequency of occurrence of various states in the *results fusion* block for all five study areas considered so far. The recorded states for the various areas used in the results analysis clearly show the importance of using multiple criterion, i.e., the color and shape properties of a building unit.

State 5 shows the high efficiency of *BDSV* in detecting true buildings, where it records only one false building in the *Ist7* study area. As is clear from the State 4 column, *BDSV* rarely fails in detecting target objects. It missed two buildings in Study Area 2 and only one in both *Ist8* and *Ist9* due to building closeness and overlapping. The *results fusion* between *TBD*, and *FBD* increase the robustness of our building detection technique, which is clear from State 1 being mostly lower than the sum of both State 2 and State 3. This results in more accurate detection, especially in mixed urban areas that contain both roof tile and non-tile flat buildings.

**Table 10.** Precision comparison of *BDSV* and Beril’s algorithm [28].

Study Area	BDSV			Beril’s Algorithm [28]		
	TP	FP	Precision	TP	FP	Precision
Ist7	14	1	0.93	9	4	0.69
Ist8	19	0	1	17	6	0.73
Ist10	23	0	1	20	3	0.86

**Table 11.** Results fusion states of all five study areas.

Study Area	State 1	State 2	State 3	State 4	State 5
Ist7	5	4	4	0	1
Ist8	4	12	3	1	0
Ist10	8	12	3	1	0

## 6. Conclusions

In this paper we present an approach known as Building Detection with Shadow Verification *BDSV* to detect buildings from RGB aerial images. We suggest *TBD* to extract roof tile buildings based on color features. We also present *FBD* to detect non-tile flat buildings with rectangular shapes based on edge information. The *TBD* algorithm shows a great efficiency in detecting colored rooftop buildings. *FBD* has some limitations in complex environments where we have a non-uniform distribution of buildings, since right corner estimation becomes a difficult task.

Experimental results reveal the high performance of the proposed method on a 11 different study areas belonging to the Sztaki–Inria dataset and the Istanbul dataset, where *BDSV* outperforms state-of-the-art algorithms in terms of the overall quality percentage (92.25%), the overall executional time (15.97 s), and the precision metric. The integration of machine learning techniques will be applied in future work, leading to more accurate results, especially for improving *FBD* performance in detecting and validating different building shapes, thus improving overall *BDSV* performance.

**Author Contributions:** A.G. and A.J. conceived and designed the experiments; A.J. performed the experiments; A.G. analyzed the data; A.G. and A.J. wrote the paper; A.G. acquired the funding.

**Acknowledgments:** This project has been funded with support from the National Council for Scientific Research in Lebanon.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Grinias, I.; Panagiotakis, C.; Tziritas, G. MRF-based Segmentation and Unsupervised Classification for Building and Road Detection in Peri-urban Areas of High-resolution. *ISPRS J. Photogramm. Remote Sens.* **2016**, *122*, 145–166. [[CrossRef](#)]
2. Benedek, C.; Descombes, X.; Zerubia, J. Building development monitoring in multitemporal remotely sensed image pairs with stochastic birth-death dynamics. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *34*, 33–50. [[CrossRef](#)] [[PubMed](#)]
3. Guindon, B. Computer-based aerial image understanding: A review and assessment of its application to planimetric information extraction from very high resolution satellite images. *Can. J. Remote Sens.* **1997**, *23*, 38–47. [[CrossRef](#)]
4. Mayer, H. Automatic object extraction from aerial imagery a survey focusing on buildings. *Comput. Vis. Image Understand.* **1999**, *74*, 138–149. [[CrossRef](#)]
5. Shufelt, J.A. Performance evaluation and analysis of monocular building extraction from aerial imagery. *IEEE Trans. Pattern Anal. Mach. Intell.* **1999**, *21*, 311–326. [[CrossRef](#)]
6. Lin, C.; Nevatia, R. Building detection and description from a single intensity image. *Comput. Vis. Image Understand.* **1998**, *72*, 101–121. [[CrossRef](#)]
7. Jaynes, C.; Stolle, F.; Collins, R. Task driven perceptual organization for extraction of rooftop polygons. In Proceedings of the 1994 IEEE Workshop on Applications of Computer Vision, Sarasota, FL, USA, 5–7 December 1994; pp. 152–159.
8. Kim, Z.; Nevatia, R. Uncertain reasoning and learning for feature grouping. *Comput. Vis. Image Understand.* **1999**, *76*, 278–288. [[CrossRef](#)]
9. Ghaffarian, S.; Ghaffarian, S. Automatic Building Detection Based On Supervised Classification Using High Resolution Google Earth Images. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2014**, *40*, 101. [[CrossRef](#)]
10. Ozgun, A.; Senaras, C.; Yuksel, B. Automated Detection of Arbitrarily-Shaped Buildings in Complex Environments from Monocular VHR Optical Satellite Imagery. *IEEE Trans. Geosci. Remote Sens.* **2013**, *51*, 1701–1717.
11. Ozgun, A. Automated detection of buildings from single VHR multispectral images using shadow information and graph cuts. *ISPRS J. Photogramm. Remote Sens.* **2013**, *86*, 21–40.
12. Irvin, R.B.; McKeown, D.M. Methods for exploiting the relationship between buildings and their shadows in aerial imagery. *IEEE Trans. Syst. Man Cybern.* **1989**, *19*, 1564–1575. [[CrossRef](#)]
13. Park, W.; Kwak, S.; Kim, T. Line-rolling algorithm for automated building extraction from 1-meter resolution satellite images. *Proc. Int. Symp. Remote Sens.* **2000**, 31–36.
14. Ngo, T.T.; Mazet, V.; Collet, C.; de Fraipont, P. Shape-based Building Detection in Visible Band Images using Shadow Information. *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.* **2017**, *10*, 920–932. [[CrossRef](#)]
15. Sohn, G.; Dowman, I.J. Extraction of buildings from high-resolution satellite data. *Autom. Extr. Man-Made Objects Aerial Space Images* **2001**, *3*, 345–355.
16. Lee, D.S.; Shan, J.; Bethel, J.S. Class-guided building extraction from Ikonos imagery. *Photogramm. Eng. Remote Sens.* **2003**, *69*, 143–150. [[CrossRef](#)]
17. Segl, K.; Kaufmann, H. Detection of small objects from high-resolution panchromatic satellite imagery based on supervised image segmentation. *IEEE Trans. Geosci. Remote Sens.* **2001**, *39*, 2080–2083. [[CrossRef](#)]
18. Levitt, S.; Aghdasi, F. Fuzzy representation and grouping in building detection. *IEEE Int. Conf. Image Process.* **2000**, *3*, 324–327.
19. Paparoditis, N. Building detection and reconstruction from mid- and high-resolution aerial imagery. *Comput. Vis. Image Understand.* **1998**, *72*, 122–142. [[CrossRef](#)]

20. Lafarge, F.; Descombes, X.; Zerubia, J.; Pierrot-Deseilligny, M. Automatic building extraction from DEMs using an object approach and application to the 3D-city modeling. *ISPRS J. Photogramm. Remote Sens.* **2008**, *63*, 365–381. [[CrossRef](#)]
21. Oczipka, M.; Bucher, T.; Trosset, A.M. Mapping and Updating Maps in Dense Urban Regions Using High Resolution Digital Airborne Imagery, Surface Models and Object-Based Classification. *IAPRS* **2008**, *37*, 551–555.
22. Teo, T.A.; Chen, L.C. Object-Based Building Detection from LiDAR Data and High Resolution Satellite Imagery. In Proceedings of the 25th Asian Conference on Remote Sensing, Chiang Mai, Thailand, 22–26 November 2004; p. 6.
23. Vu, T.T.; Matsuoka, M.; Yamazaki, F. LiDAR Signature to Update Japanese Building Inventory Database. In Proceedings of the 25th Asian Conference on Remote Sensing, Chiang Mai, Thailand, 22–26 November 2004; pp. 624–629.
24. Rottensteiner, F.; Trinder, J.; Clode, S.; Kubic, K. Building Detection Using LIDAR Data and Multispectral Images. In Proceedings of the Digital Image Computing: Techniques and Applications, Sydney, Australia, 10–12 December 2003; pp. 673–682.
25. Rottensteiner, F.; Trinder, J.; Clode, S.; Kubic, K. Building Detection by Dempster-Shafer Fusion of LIDAR Data and Multispectral Aerial Imagery. In Proceedings of the 17th International Conference on Pattern Recognition, Cambridge, UK, 23–26 August 2004; pp. 339–342.
26. Cheng, L.; Gong, J.; Chen, X.; Han, P. Building Boundary Extraction from High Resolution Imagery and LiDAR Data. In Proceedings of the XXI-st ISPRS Congress, Beijing, China, 3–11 July 2008; pp. 693–698.
27. Guo, T.; Yasuoka, Y. Snake-Based Approach for Building Extraction from High-Resolution Satellite Images and Height Data in Urban Areas. In Proceedings of the 23rd Asian Conference on Remote Sensing, Kathmandu, Nepal, 25–29 November 2002; p. 6.
28. Sirmacek, B.; Unsalan, C. Building Detection from Aerial Images using Invariant Color Features and Shadow Information. In Proceedings of the 23rd International Symposium on Computer and Information Sciences, Istanbul, Turkey, 27–29 October 2008; pp. 1–5.
29. Guducu, H.V. Building Detection from Satellite Images Using Shadow and Color Information. Ph.D. Thesis, Middle East Technical University, Ankara, Turkey, 2008.
30. Aytekn, O.; Erener, A.; Ulusoy, I.; Duzgun, S. Unsupervised building detection in complex urban environments from multispectral satellite imagery. *Int. J. Remote Sens.* **2012**, *33*, 2152–2177. [[CrossRef](#)]
31. Kim, D.J.; Manjusha, P.L. Building Detection in High Resolution Remotely Sensed Images based on Automatic Histogram-Based Fuzzy C-Means Algorithm. *Asia-Pac. J. Converge.Res. Interchange* **2017**, *3*, 57–62.
32. Shorter, N.; Kasparis, T. Automatic Vegetation Identification and Building Detection from a Single Nadir Aerial Image. *Remote Sens.* **2009**, *1*, 731–757. [[CrossRef](#)]
33. Unsalan, C.; Boyer, K.L. Linearized vegetation indices based on a formal statistical framework. *IEEE Trans. Geosci. Remote Sens.* **2004**, *42*, 1575–1585. [[CrossRef](#)]
34. Seref, A. *Shadow Detection and Compensation in Aerial Images with an Application to Building Height Estimation*; Middle East Technical University: Ankara, Turkey, 2010.
35. Ziou, D.; Tabbone, S. Edge detection techniques: An overview. *Int. J. Pattern Recognit. Image Anal.* **1998**, *8*, 537–559.
36. Hamarneh, G.; Althoff, K.; Gharbieh, R. *Automatic Line Detection*; Image Analysis Group, Department of Signals and Systems, Chalmers University of Technology: Gothenburg, Sweden, September 1999.
37. Sirmacek, B.; Unsalan, C. Urban-area and building detection using SIFT key points and graph theory. *IEEE Trans. Geosci. Remote Sens.* **2009**, *47*, 1156–1167. [[CrossRef](#)]
38. Sirmacek, B.; Unsalan, C. A probabilistic framework to detect buildings in aerial and satellite images. *IEEE Trans. Geosci. Remote Sens.* **2011**, *49*, 211–221. [[CrossRef](#)]
39. Muller, S.; Zaum, D. Robust building detection in aerial images. *Int. Arch. Photogramm. Remote Sens.* **2005**, *36*, 143–148.

