Urban-Area and Building Detection Using SIFT Keypoints and Graph Theory

Beril Sırmaçek, *Student Member, IEEE*, and Cem Ünsalan, *Member, IEEE*

***Abstract*—Very high resolution satellite images provide valuable information to researchers. Among these, urban-area boundaries and building locations play crucial roles. For a human expert, manually extracting this valuable information is tedious. One possible solution to extract this information is using automated techniques. Unfortunately, the solution is not straightforward if standard image processing and pattern recognition techniques are used. Therefore, to detect the urban area and buildings in satellite images, we propose the use of scale invariant feature transform (SIFT) and graph theoretical tools. SIFT keypoints are powerful in detecting objects under various imaging conditions. However, SIFT is not sufficient for detecting urban areas and buildings alone. Therefore, we formalize the problem in terms of graph theory. In forming the graph, we represent each keypoint as a vertex of the graph. The unary and binary relationships between these vertices (such as spatial distance and intensity values) lead to the edges of the graph. Based on this formalism, we extract the urban area using a novel multiple subgraph matching method. Then, we extract separate buildings in the urban area using a novel graph cut method. We form a diverse and representative test set using panchromatic 1-m-resolution Ikonos imagery. By extensive testings, we report very promising results on automatically detecting urban areas and buildings.**

***Index Terms*—Building detection, graph cut, multiple subgraph matching, scale invariant feature transform (SIFT), urban-area detection.**

I. Introduction

SATELLITE images offer valuable information to researchers. The resolution of earlier satellite imagery (such as Landsat) would not allow detecting separate man-made or natural objects. Therefore, researchers mostly focused on extracting the region properties from these imagery. As the advent of very high resolution (VHR) satellite imagery (such as Ikonos and Quickbird), it became possible to observe these objects. Aside from region properties, extracting man-made objects in VHR satellite images may help researchers in various ways, such as automated map making.

Among different man-made objects, buildings play an important role. Therefore, the robust detection of buildings in VHR satellite images requires a specific consideration. Unfortunately, it is still tedious for a human expert to manually label buildings in a given satellite image. One main reason is the total number of objects in the scene. The other reason is the resolution of the satellite image. Although the resolution of the satellite imagery has reached an acceptable level, it is still not possible for a human expert to extract information from it in a robust manner. To solve this problem, researchers introduced automated urban-area- and building-detection methods using VHR satellite and aerial images.

Zerubia *et al.* [18] used texture information to detect urban areas in both optical and radar images. They extracted texture parameters using chain-based Gaussian models. Then, they used clustering with a Markovian segmentation step. Their system is robust to sensor changes, but not very robust to resolution changes in images. Rokos *et al.* [12] used building density information to classify residential regions. Benediktsson *et al.* [1] used mathematical morphological operations to extract structural information to detect the urban area in satellite images. Ünsalan and Boyer [30], [32] extracted linear features from satellite images to detect residential regions. They benefit from spatial coherence and graph theory for labeling urban or residential regions. Ünsalan [29], in a related study, used graph theory to grade changes in urban regions. Fonte *et al.* [7] considered corner detectors to obtain the type of structure in a satellite image. They concluded that corner detectors may give distinctive information on the type of structure in an image. Bhagavathy and Manjunath [2] used texture motifs for modeling and detecting regions (such as golf parks and harbors) in satellite images. Bruzzone and Carlin [3] proposed a pixel-based system to classify VHR satellite images. They used support vector machines fed with a novel feature extractor. Zhong and Wang [35] introduced conditional random fields to learn dependencies in the image. They fuse the multilevel structural information to obtain urban areas in satellite images. A related problem in the literature is the satellite image classification. The literature is vast on this topic. Some related papers can be counted as [4], [6], [10], [15]–[17], and [20].

Kim and Muller [14] used graph theory to detect buildings in aerial images. They extracted linear features in the given image and used them as vertices of a graph. Then, they extracted buildings by applying subgraph matching with their model building graph. Finally, they used intensity and shadow information to verify the building appearance. This study follows a similar strategy as we did. Different from us, they used colored aerial images and linear features. Segl and Kaufmann [26] combined supervised shape classification with unsupervised image segmentation in an iterative way. Their method allows searching objects (like buildings) in high-resolution satellite images. Ünsalan and Boyer [31] studied multispectral satellite

Manuscript received July 2, 2008; revised September 15, 2008 and October 19, 2008. Current version published March 27, 2009.

The authors are with the Department of Electrical and Electronics Engineering and the Computer Vision Research Laboratory, Yeditepe University, Istanbul 34755, Turkey (e-mail: unsalan@yeditepe.edu.tr; bsirmacek@yeditepe.edu.tr).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TGRS.2008.2008440

images to detect buildings and street networks in residential regions. Their method depends on vegetation indexes, clustering, decomposing binary images, and graph theory. They also offer a nice review on building-detection algorithms in the literature. Mayunga *et al.* [21] used polygons, formed by edge information, to detect buildings. They proposed a novel snake algorithm starting from an approximate polygon center. The snake grows radially until it fits a closed polygon shape. Then, they used linear features to verify the building appearance. Peng *et al.* [25] also proposed an improved snake model to detect buildings in colored aerial images. They report good detection results with their system. Huang *et al.* [11] considered fusion of multispectral Ikonos imagery to classify objects (including buildings) in urban regions. Molinier *et al.* [23] considered detecting man-made structures in satellite images using PicSOM. Gamba *et al.* [8] used boundary information to extract the map of an urban area. They fed the boundary and nonboundary data to two different classifiers. Then, they combined the results of the two classifiers. They obtained satisfactory results to detect urban-area buildings on VHR imagery. Wei and Xin [33] introduced a method based on level sets to segment out man-made objects in aerial images. Katartzis and Sahli [13] used a hierarchical stochastic model based on perceptual organization to detect building rooftops in colored satellite images. Zhang *et al.* [34] used airborne light detection and ranging data to detect buildings.

Detecting buildings in satellite images is a difficult task for several reasons. Buildings may be imaged from different viewpoints. The illumination and contrast in the image may not be sufficient for detecting buildings. There may be several other structures, such as nearby trees and street segments, making the building-detection problem harder. In addition to these difficulties, buildings do not have a standard size and shape. All these issues make building detection a hard problem.

Scale invariant feature transform (SIFT) is a good candidate for urban-area and building detection in satellite images. SIFT is a local descriptor extraction method having valuable properties such as invariance to illumination and viewpoint [19]. In the literature, it is extensively used to match objects represented by template images in a given image. Unfortunately, the standard SIFT implementation is not sufficient for urban-area and building detection from satellite images alone. There are many similar and nearby buildings in the satellite image. Therefore, standard feature matching does not work properly.

In this study, we propose novel methods to detect the urban areas and buildings from panchromatic VHR Ikonos images. Our detection methods are based on SIFT keypoints and graph theoretical tools. We first upsample the image by six to help SIFT keypoint extraction. Then, we apply a nonlinear bilateral filtering (BF) operation to smooth out unwanted noise terms in the image [28]. Afterwards, we extract local SIFT keypoints (and associated features) from the processed satellite image. Different from the original SIFT, we cast the urban-region-detection problem as one of multiple subgraph matching. In forming the graph, each SIFT keypoint is taken as a vertex. The neighborhood between different vertices is summarized as edges of the graph. In the same way, we formulate the building-detection problem in terms of graph cut. Gautama *et al.* [9], in a related study, used error-tolerant graph matching to find correspondences between the detected image features and the geospatial vector data.

We test our urban-area- and building-detection methods on a fairly diverse and representative image set formed from panchromatic Ikonos images of 28 different urban sites in Adana, Ankara, and Istanbul, Turkey. They contain various building types, as well as various urban-area characteristics. Tests indicate the potential of our urban-area- and building-detection methods. To explain our novel detection methods in detail, we start with the preprocessing of our satellite images.

II. Bilateral Filtering for Preprocessing

Ikonos satellite images are not directly suitable for robust SIFT keypoint extraction due to their resolution. The main problem here is the minimum size of the filter scale used in standard SIFT keypoint extraction. Unfortunately, this filter scale is fairly large for detecting building properties. Therefore, we first upsample the image by six in each coordinate axis using bilinear interpolation. To note here, nearest neighbor interpolation may also be used with a slight performance change. Moreover, upsampling itself is not sufficient. We should also eliminate noise in the satellite image. Unfortunately, objects are so small in these images that it is almost impossible to discard noise terms without disturbing object boundaries using standard linear filters. Therefore, we propose the use of BF proposed by Tomasi and Manduchi [28]. It performs nonlinear smoothing on images by keeping the edge information. Nonlinear smoothing is performed by combining the geometric and intensity similarity of pixels. Elad [5] shows that BF is closely related to edge preserving smoothing operations such as anisotropic diffusion. We next explain the BF operation using Elad's notation.

Let $I_g(x,y)$ be a grayscale image having values in the range [0, 1]. Let $I_b(x,y)$ be the bilateral filtered version of $I_g(x,y)$. The filtering operation can be represented as

$$I_b(x,y)=\frac{\sum_{n=-N}^{N}\sum_{m=-N}^{N}W(x,y,n,m)I_g(x-n,y-m)}{\sum_{n=-N}^{N}\sum_{m=-N}^{N}W(x,y,n,m)}. \tag{1}$$

This equation is simply a normalized weighted average of a neighborhood of $2N+1$ by $2N+1$ pixels around the pixel location (x,y). The weight $W(x,y,n,m)$ is computed by multiplying the following two factors:

$$W(x,y,n,m)=W_s(x,y,n,m)\times W_r(x,y,n,m) \tag{2}$$

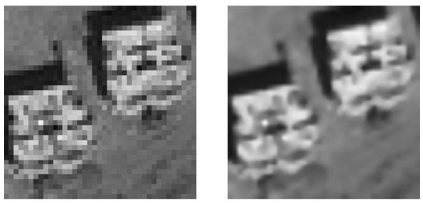
where $W_s(x,y,n,m)$ stands for the geometric weight factor. It is based on the Euclidean distance between the center pixel (x,y) and the $(x-n,y-m)$ pixel as

$$W_s(x,y,n,m)=\exp\left(-\frac{(x-n)^2+(y-m)^2}{2\sigma_s^2}\right). \tag{3}$$

This way, nearby samples influence the final result more than distant ones.

The second weight $W_r(x,y,n,m)$ is based on the grayscale intensity distance between the values of the center pixel (x,y)

Fig. 1. Adana$_8$ test image.



(a) (b)

Fig. 2. Subpart of the Adana$_8$ test image and its BF result. (a) Original. (b) Bilaterally filtered.

and the $(x-n, y-m)$ pixel. Again, it is based on the Euclidean distance between intensity values as

$$W_r(x,y,n,m) = \exp\left(-\frac{(I_g(x,y) - I_g(x-n,y-m))^2}{2\sigma_r^2}\right). \tag{4}$$

Thus, pixels with close grayscale intensity values tend to influence the final result more than those having distant values.

The bilateral filter is controlled by three parameters. N dictates the support of the filter. A larger support gives a stronger smoothing. The parameters σ_s and σ_r control the decay of the two weight factors. We pick $N = 5$, $\sigma_s = 3$, and $\sigma_r = 0.1$ for implementation. These values are picked keeping in mind the minimum geometric size and intensity variations of buildings in the Ikonos image to be detected. For other satellite or aerial image types, these parameters should be adjusted accordingly.

In this study, we use the Adana$_8$ image shown in Fig. 1 to illustrate our method step by step. In this test image, it is hard to detect the urban area and the buildings due to the low contrast between building rooftops and the background. Moreover, this is a typical test site.

To provide a detailed explanation of our system, we focus on a subpart of the Adana$_8$ image as in Fig. 2. In this paper, we only consider two buildings. We first provide the BF results of this subpart image in the same figure. As can be seen, the bilateral filtered image is fairly smooth, with the edge information of buildings kept fairly well.

III. SIFT

Lowe [19] proposed the SIFT for object detection based on its template image. SIFT leads to local image descriptors, invariant to translation, scaling, and rotation. These descriptors are also partially invariant to illumination changes and 3-D projection. Mikolajczyk and Schmid [22] compared SIFT descriptors with other invariant feature descriptors. They concluded that SIFT performs best under changes in scale, rotation, and illumination. Lowe proposed the detection of an object in an image by descriptor matching. To do so, first, a template image is obtained for the object to be detected in the test image. Its descriptors are extracted. Then, descriptors for the test image are extracted. A one-to-one matching between the template and test image descriptors leads to detecting the object in the test image.

Buildings can be considered as objects to be detected in the satellite image. In satellite images, buildings can have different illumination conditions, structural differences, and size changes. More importantly, they can be imaged from different viewpoints. The invariance properties of SIFT can handle most of these variations. Therefore, SIFT is suitable for building detection from satellite images. Next, we briefly explain SIFT and keypoint extraction. More details on SIFT can be found in [19].

The first stage of keypoint detection is to identify locations and scales that can be repeatedly assigned under differing views of the same object. Detecting locations that are invariant to the scale change of the image can be accomplished by searching for stable features across all possible scales, using a continuous function of scale known as scale space. The scale space of an image is defined as a function $L(x,y,\sigma)$ that is produced from the convolution of a variable scale Gaussian, with the input image $I_b(x,y)$ as

$$L(x,y,\sigma) = \frac{1}{2\pi\sigma^2}\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) * I_b(x,y) \tag{5}$$

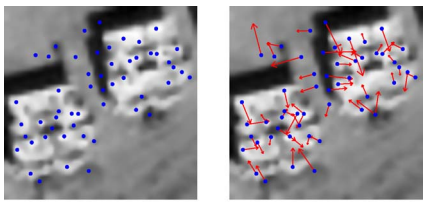
where $*$ is the convolution operation in x, y and σ is the Gaussian scale.

To efficiently detect stable keypoint locations in scale space, Lowe proposed the use of the scale-space extrema. It is calculated from the difference of Gaussian images computed from the two nearby scales separated by a constant multiplicative factor k as

$$D(x,y,\sigma) = L(x,y,k\sigma) - L(x,y,\sigma). \tag{6}$$

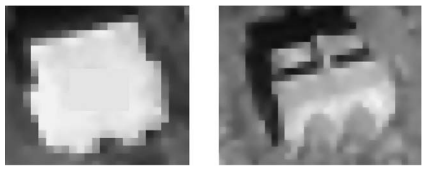
In order to detect the extrema of $D(x,y,\sigma)$, each sample point is compared to its eight neighbors in the current image and nine neighbors in the scales above and below. The sample point is selected only if it is larger than all of these neighbors or smaller than all of them. Once a keypoint candidate has been obtained by comparing a pixel to its neighbors, the next step is to perform a detailed fit to the nearby data for the location, scale, and ratio of the principal curvatures. This information allows points that have low contrast (and are therefore sensitive to noise) or are poorly localized along the edge to be rejected. We label the keypoint by its spatial coordinates as (x_i, y_i).

One or more orientations are assigned to each keypoint location based on local image gradient directions. By assigning a consistent orientation to each keypoint based on local image properties, the keypoint descriptor can be represented relative to this orientation. Therefore, it has an invariance to image rotation.



(a) (b)

Fig. 3. SIFT keypoints and their vector representation on the Adana_8 subpart image. (a) Keypoints. (b) Vectors.



(a) (b)

Fig. 4. Two building (bright and dark) templates used in this paper. (a) Bright building. (b) Dark building.

The keypoint descriptor generated by the SIFT algorithm is created by sampling the magnitudes and orientations of the image gradient in the patch around the keypoint. Finally, a 128-element vector (represented as $\mathbf{f_i}$) is formed as a descriptor. This vector is then normalized to unity magnitude to have an invariance to illumination.

We extract keypoints and vector representations from the Adana_8 subpart image as in Fig. 3. In this figure, vectors originate from keypoint locations. They represent the dominant direction of the SIFT descriptor for each keypoint as well as their strength. As can be seen, keypoints in this image are located around building corners and other significant intensity changes.

IV. Detecting the Urban Area and Buildings

In the original SIFT, the object to be detected is represented by one or several template images. Then, keypoint descriptors are obtained for each template. Keypoint descriptors for the test image are also obtained. A one-to-one matching between template and test image keypoints is performed using the Euclidean distance between keypoint descriptors. SIFT descriptors are highly discriminative. Therefore, each template keypoint matches with the closest (in the Euclidean distance sense) test image keypoint. This is the strength of the original SIFT-based object detection. However, this property of the SIFT is not suitable for our problem. First, we have many buildings in the satellite image to be detected. Second, it is not feasible to have templates for all types of buildings to be detected. Therefore, we propose novel graph theoretical methods to detect the urban area and buildings.

To detect buildings and the urban area, we use two template building images as shown in Fig. 4. The first template represents a bright building, having a high contrast between the background and its rooftop. The second template represents a dark building, having relatively low contrast between the background and its rooftop. These two templates cover a wide range of building characteristics.

As aforementioned, each SIFT keypoint is described by a vector $\mathbf{v}_i = (x_i, y_i, \mathbf{f_i})$. Here, (x_i, y_i) represents the spatial coordinate of the keypoint. $\mathbf{f_i}$ is the 128-element feature vector for that keypoint. We first extract keypoints for the two templates and the test image. Then, we represent them as $\mathbf{v}_i^a,\ i = 1, \ldots, I$, for the dark building template, $\mathbf{v}_j^r,\ j = 1, \ldots, J$, for the bright building template, and $\mathbf{v}_m^t,\ m = 1, \ldots, M$, for the test image.

A. Graph Representation

To detect the urban area and buildings, we cast the problem in terms of graph theory. A graph G is represented as $G = (V, E)$, where V is the vertex set and E is the edge matrix showing the relations between these vertices. For the urban-area and building detection, we represent the keypoints extracted from the dark building template $(\mathbf{v}^a)$, bright building template $(\mathbf{v}^r)$, and test image $(\mathbf{v}^t)$ in a graph formation as $G^a(V^a, E^a)$, $G^r(V^r, E^r)$, and $G^t(V^t, E^t)$, respectively. Let us consider G^a. $V^a = \{\mathbf{v}_i^a\},\ i = 1, \ldots, I$. E^a is an $I \times I$ matrix defined as

$$E^a(i,j) = \begin{cases} d_{ij}, & \text{if } d_{ij} < \epsilon_1 \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

where $d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$. We take $\epsilon_1 = 30$ due to the size of buildings that we are detecting in the Ikonos imagery. If this method is applied to higher resolution images, then the value of ϵ_1 should also be increased accordingly. $E^a(i,j) = 0$ means that there is no edge between vertices $\mathbf{v}_i$ and $\mathbf{v}_j$. We form G^r and G^t in a similar way.

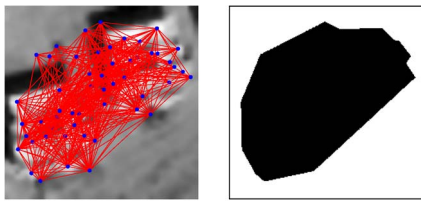
B. Multiple Subgraph Matching to Detect the Urban Area

Buildings in a region indicate the existence of an urban area. Therefore, in order to detect the urban area, it is sufficient to detect buildings. To detect all buildings in an image (without selecting a special one), we apply multiple subgraph matching between G^a, G^t and G^r, G^t separately. Applying multiple subgraph matching between the template and test images is different from the original SIFT. As aforementioned, we have many buildings in the same region, and we want to detect all at once without selecting a specific one. This formalism simplifies our urban-area-detection problem since nearby buildings affect each other's detection probability in multiple subgraph matching. Also, using this formalism, we relax the graph matching condition.

From now on, we will explain our multiple subgraph matching method on the G^a, G^t pair. The method is the same for the G^r, G^t pair. Our multiple subgraph matching method can be rephrased as a one-to-many matching between two graph vertices both in unary and binary terms. For the unary matching between G^a and G^t, we define multiple vertex matching between two graphs $G^a = (V^a, E^a)$ and $G^t = (V^t, E^t)$ as

$$\mathbf{M_1}\left(\mathbf{v}_i^a, \mathbf{v}_j^t\right) = \begin{cases} 1, & \text{if } \left\|\mathbf{f}_i^a - \mathbf{f}_j^t\right\| < \epsilon_2 \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

$\mathbf{M_1}(\cdot,\cdot)$ will be an $I \times M$ matrix. We check matching $\forall \mathbf{v}_i^a \in V^a$ and $\forall \mathbf{v}_j^t \in V^t$. In (8), ϵ_2 stands for the adaptive tolerance value for unary matching. Since we have multiple subgraph



(a) (b)

Fig. 5. Graph and its region obtained from the Adana$_8$ subpart image. (a) Graph G^d. (b) Region $A(G^d)$.

matching, such an adaptive parameter adjustment strategy is necessary. We calculate ϵ_2 as follows. We first calculate the best match between V^a and V^t. We assume this to be a valid match. In other words, we assume that the matched vertex in the test image belongs to a building. Then, we calculate the next best match by comparing the match distance with the first best match. We repeat this process as long as the match distance is comparable with the best match case. If there is a larger distance (such as two times the best match), then we assume that the match is not valid. We pick this distance value as ϵ_2.

For urban-area and building detection, matched keypoints should also keep their structure. Therefore, we define binary vertex matching between the two weighted graphs $G^a = (V^a, E^a)$ and $G^t = (V^t, E^t)$ as

$$\mathbf{M_2}\left(E^a(i,j), E^t(k,l)\right) = \begin{cases} 1, & \text{if } (\mathbf{M_1}\left(v_i^a, v_k^t\right) = 1) \wedge \\ & \left(\mathbf{M_1}\left(v_j^a, v_l^t\right) = 1\right) \wedge (\gamma < \epsilon_3) \\ 0, & \text{otherwise} \end{cases} \quad (9)$$

where $\gamma = |E^a(i,j) - E^t(k,l)|$. Here, ϵ_3 is the tolerance value for binary matching. Based on the dimensions of the buildings in our building template images, we set $\epsilon_3 = 4$. For other building template images, ϵ_3 should be adjusted accordingly. For example, if a larger building template is to be used, then ϵ_3 should be increased. Similarly, if a smaller building template is to be used, then ϵ_3 should be decreased.

We form a new graph, based on unary and binary matching (matched vertices and edges of G^t). We call it as $G^d = (V^d, E^d)$ to indicate that it contains vertices and edges of the test image graph matched with the dark building template graph. $\mathbf{v}_m^t \in V^d$ iff $\exists \mathbf{v}_i^a \in V^a$ such that $\mathbf{M_1}(\mathbf{v}_i^a, \mathbf{v}_m^t) = 1$. We form the edge matrix E^d as

$$E^d(k,l) = \begin{cases} E^t(k,l), & \text{if } \exists\, i,j \text{ s.t.} \\ & \mathbf{M_2}\left(E^a(i,j), E^t(k,l)\right) = 1 \\ 0, & \text{otherwise.} \end{cases} \quad (10)$$

We provide the constructed graph G^d on the Adana$_8$ subpart image in Fig. 5(a). As can be seen, all the matched vertices of G^t (now the vertices of G^d) lie on the buildings in the image. Due to multiple subgraph matching, most vertices on different buildings also have edges connecting them.

We apply the same procedure to form $G^b = (V^b, E^b)$ in the same way using the G^r, G^t pair. This graph indicates the unary and binary matching between the bright building template and the test image.

Fig. 6. Detected urban area from the Adana$_8$ test image.

To locate the urban area containing buildings, we define the *region of a graph*. For a graph $G(V,E)$ with vertices $V = \{v_i\}$ having spatial coordinates $v_i = (x_i, y_i)$, we define its region $A(G)$ as follows.

1) $v_i \in V \Rightarrow (x_i, y_i) \in A(G)$.
2) Let l_{ij} be the line segment joining v_i, v_j where $E(i,j) \neq 0$; $(x_a, y_a) \in l_{ij} \Rightarrow (x_a, y_a) \in A(G)$.
3) Let t_{ijk} be the triangle with corners $v_i, v_j, v_k \in V$ where $E(i,j) \neq 0$, $E(i,k) \neq 0$, and $E(j,k) \neq 0$; $(x_a, y_a) \in t_{ijk} \Rightarrow (x_a, y_a) \in A(G)$.

This region is as small as possible. It includes all vertices and line segments joining them. To clarify this operation, we formed the region of graph G^d on the Adana$_8$ subpart image. We provide $A(G^d)$ for this image in Fig. 5(b).

There may be both bright and dark buildings in a given test site. Therefore, we detect the final urban area R using both $A(G^d)$ and $A(G^b)$ as

$$R = A(G^d) \bigcup A(G^b). \quad (11)$$

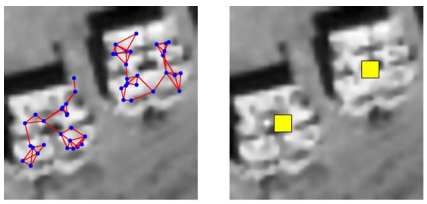
We provide the detected urban area from the Adana$_8$ test image in Fig. 6. As can be seen, the urban area in this image is correctly detected. We provide more urban-area-detection examples, as well as the qualitative results in the Section V.

C. Graph Cut Method to Detect Separate Buildings

Up to now, we benefit from the close proximity of buildings in detecting the urban area. Now, the second step is detecting each building alone. For separate building detection, we need extra information. Therefore, we cut some edges of G^d and G^b based on the intensity criteria. We hypothesize that vertices (keypoints) on the same building have similar intensity values due to the building color. Therefore, to locate separate buildings, we cut edges between vertices having different intensity values. We expect the cut edges to be the ones between vertices on different buildings. We form two new graphs as $G^p = (V^p, E^p)$ and $G^q = (V^q, E^q)$. Here, $V^p = V^d$ and $V^q = V^b$. Let us consider G^p. We assign weights to its edges as

$$E^p(k,l) = \begin{cases} 1, & \text{if } \left(E^d(k,l) \neq 0\right) \wedge (w_{kl} < \epsilon_4) \\ 0, & \text{otherwise} \end{cases} \quad (12)$$

where $w_{kl} = |I_b(x_k, y_k) - I_b(x_l, y_l)|$. (x_k, y_k) and (x_l, y_l) stand for the spatial coordinates of $\mathbf{v}_k$, $\mathbf{v}_l \in V^p$. $I_b(x,y)$ is our



(a) (b)

Fig. 7. Graph cut on the Adana$_8$ subpart image. (a) Graph cut. (b) Buildings.

bilateral filtered image. $\epsilon_4 = 0.1$ is the tolerance value (remember $I_b(x, y) \in [0, 1]$). As long as the grayscale satellite image is scaled between zero and one, fixing ϵ_4 to 0.1 seems reasonable. Therefore, this parameter is insensitive to the resolution of the satellite image used. It may only be adjusted by the dynamic grayscale range of the image. We apply the same procedure to obtain E^q as

$$E^q(k,l) = \begin{cases} 1, & \text{if } \left(E^b(k,l) \neq 0\right) \wedge (w_{kl} < \epsilon_4) \\ 0, & \text{otherwise.} \end{cases} \quad (13)$$

The weight-assignment step may not work properly if keypoints are located outside the building. To overcome this problem, keypoint vectors may be used to shift each keypoint location by an offset. For buildings brighter than the background (and the bright building template is matched with), keypoint vectors are directed outside the building center. For buildings darker than the background (and the dark building template is matched with), keypoint vectors are directed toward the building center (as can be seen in Fig. 3). This information may be used to shift each keypoint location inside the building.

Equations (12) and (13) lead to disconnected subgraphs. Each disjoint and connected subgraph possibly represents a building. Therefore, we detect disjoint and connected subgraphs from G^p and G^q. A graph $G(V, E)$ can be decomposed into disjoint and connected subgraphs $G_l(V_l, E_l),\ l = 1, \ldots, L$. Each subgraph satisfies the following conditions.

1) $V_l \subseteq V$.
2) $\bigcup_{l=1}^{L} V_l = V$.
3) $\forall v_i,\ v_j \in V_l \exists$ a path between v_i and v_j. A *path* in G is a finite alternating sequence of vertices and edges.
4)

$$E_l(i,j) = \begin{cases} 1, & \text{if } (E(i,j) = 1) \wedge (v_i, v_j \in V_l) \\ 0, & \text{otherwise.} \end{cases} \quad (14)$$

We obtain disjoint and connected subgraphs for G^p using the aforementioned definition as $G_i^p,\ i = 1, \ldots, I$. However, there may be many noise terms. To discard them, we select subgraphs having at least two vertices. Similarly, we obtain disjoint and connected subgraphs for G^q as $G_j^q,\ j = 1, \ldots, J$. For the Adana$_8$ subpart image, we provide the obtained disjoint and connected subgraphs $G_i^p,\ i = 1, 2$, in Fig. 7(a). As can be seen, each disjoint and connected subgraph lies on a different building in this test image.

As we hypothesized, each subgraph $G_i^p,\ i = 1, \ldots, I$, and $G_j^q,\ j = 1, \ldots, J$, represents a candidate building. To locate buildings, we obtain the region of each subgraph as $A(G_i^p)$ and $A(G_j^q)$ separately. Some subgraphs in G_i^p and G_j^q may represent the same building. In other words, the building may be detected by both dark and bright building templates. To eliminate double counts, we obtain the union of regions as

$$F = A\left(G_i^p\right) \bigcup A\left(G_j^q\right) \quad \forall i, j. \quad (15)$$

We apply binary labeling on F and obtain its connected components [27]. Each connected component represents a building. If the size of a connected component is less than 1000 pixels, we take it to be noise and discard it. This corresponds to approximately a building with a size of 6×5 pixels in the original image (remember, we upsample the test image by six in each coordinate at the beginning). We obtain the center of mass of each connected component and take it as the location of the building it represents.

Fig. 8. Detected buildings in the Adana$_8$ test image.

We provide the building-detection result on the Adana$_8$ subpart image in Fig. 7(b). As can be seen, the two buildings in this image are correctly detected. We also provide the building-detection results on the Adana$_8$ test image in Fig. 8. Again, most of the buildings are correctly detected in this test image. Since the contrast between the background and the building rooftops are very low for this image, it is really hard to detect these buildings even for a human observer. In the next section, we quantify our building-detection results on a diverse test set.

V. Experiments

We test our urban-area- and building-detection methods on 28 panchromatic Ikonos images. Of these images, 19 are acquired from different sites in Adana, Turkey. Five of these images are taken from Ankara, Turkey, and four of these images are taken from Istanbul, Turkey. In these test images, the sizes and shapes of buildings, their proximity, environment, and contrast w.r.t. background all differ. These test images are specifically selected to represent a wide and diverse urban area and building characteristics. We provide our test images (in the first columns) in Figs. 9–11. In these figures, we provide the detected urban area for each test image in the second columns. We also provide the detected buildings for each test image in the third columns. In the following sections, we analyze these detection results quantitatively.

A. Urban-Area Detection

We start analyzing the urban-area-detection results. As can be seen in Figs. 9–11 (second columns), our method labels the

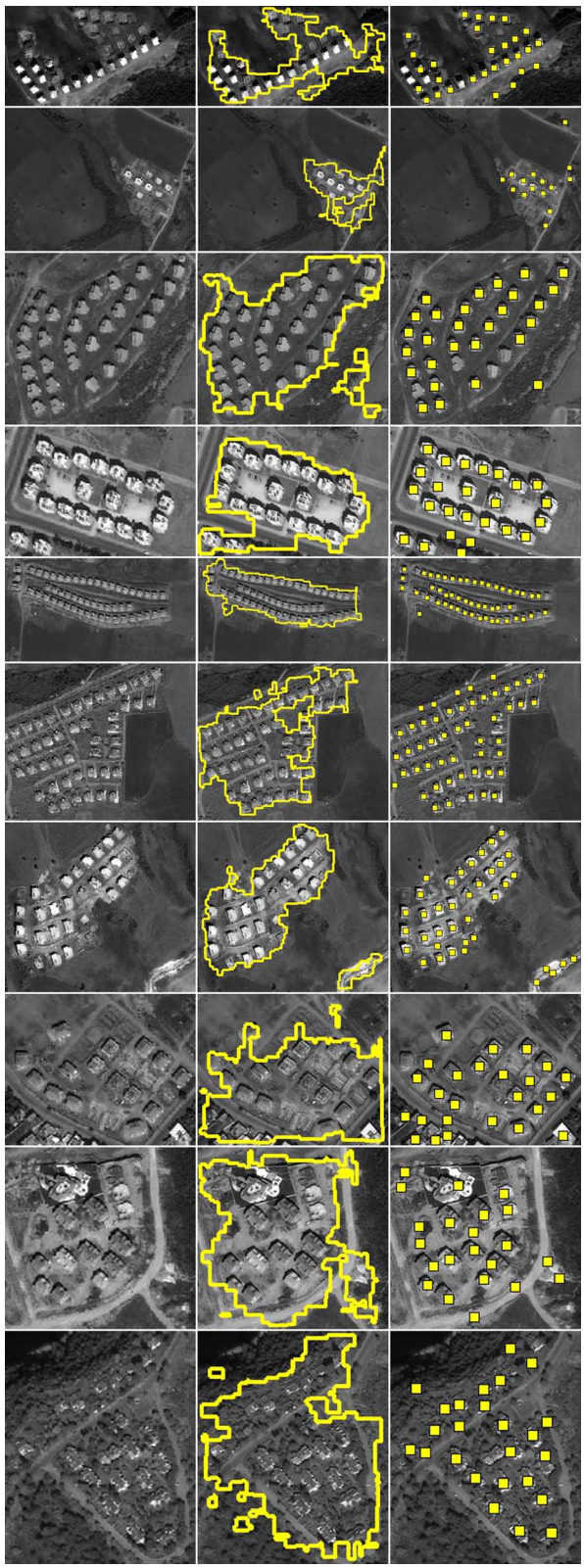


Fig. 9. Test results for Adana$_{\{1,2,3,4,5,6,7,9,10,11\}}$ images for each row separately. (First column) Test images. (Second column) Detected urban areas. (Third column) Detected buildings.

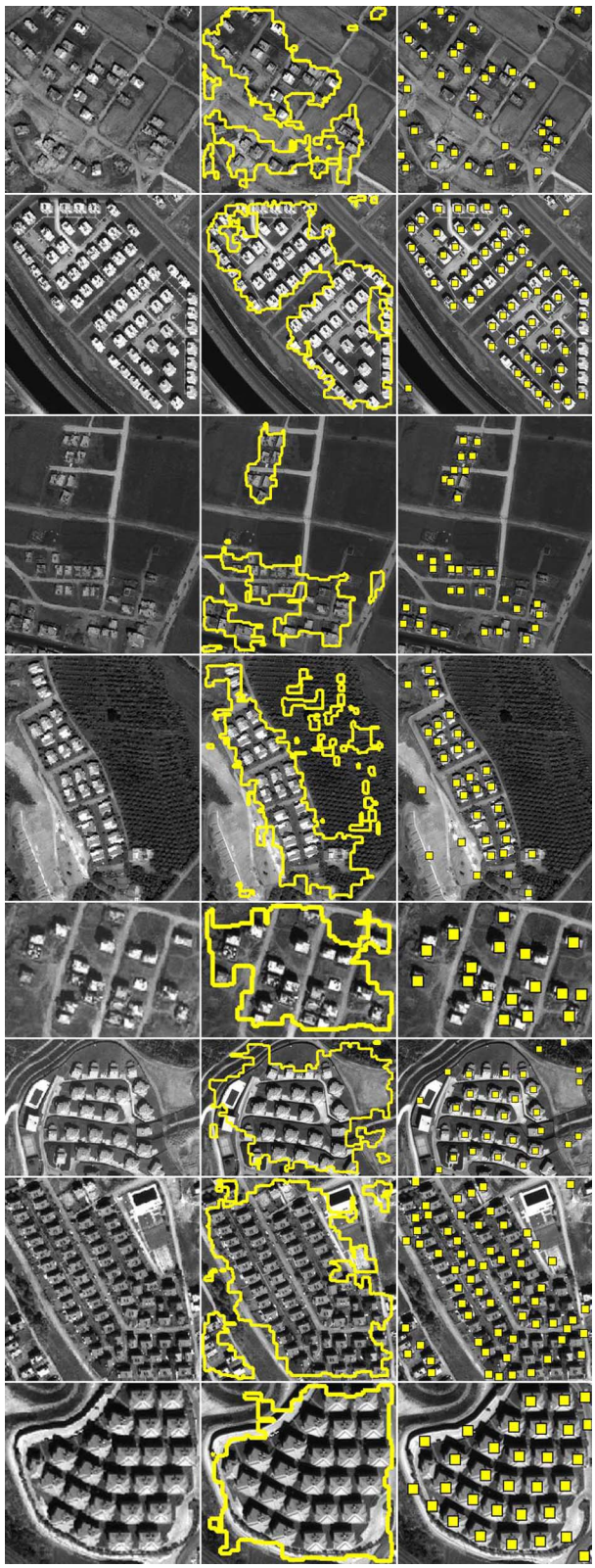


Fig. 10. Test results for Adana$_{\{12,13,14,15,16,17,18,19\}}$ images for each row separately. (First column) Test images. (Second column) Detected urban areas. (Third column) Detected buildings.

urban area for each test image fairly well. To quantify these detection results, we formed the ground truth for each test image to the best of our knowledge. In forming the ground truth, we label a region as urban if it contains building clusters, gardens around them, and nearby street segments joining them. We provide the urban-area-detection performances (in percentages) for all test images in Table I. In this table, P_d stands for probability of detection (correct urban-area-detection

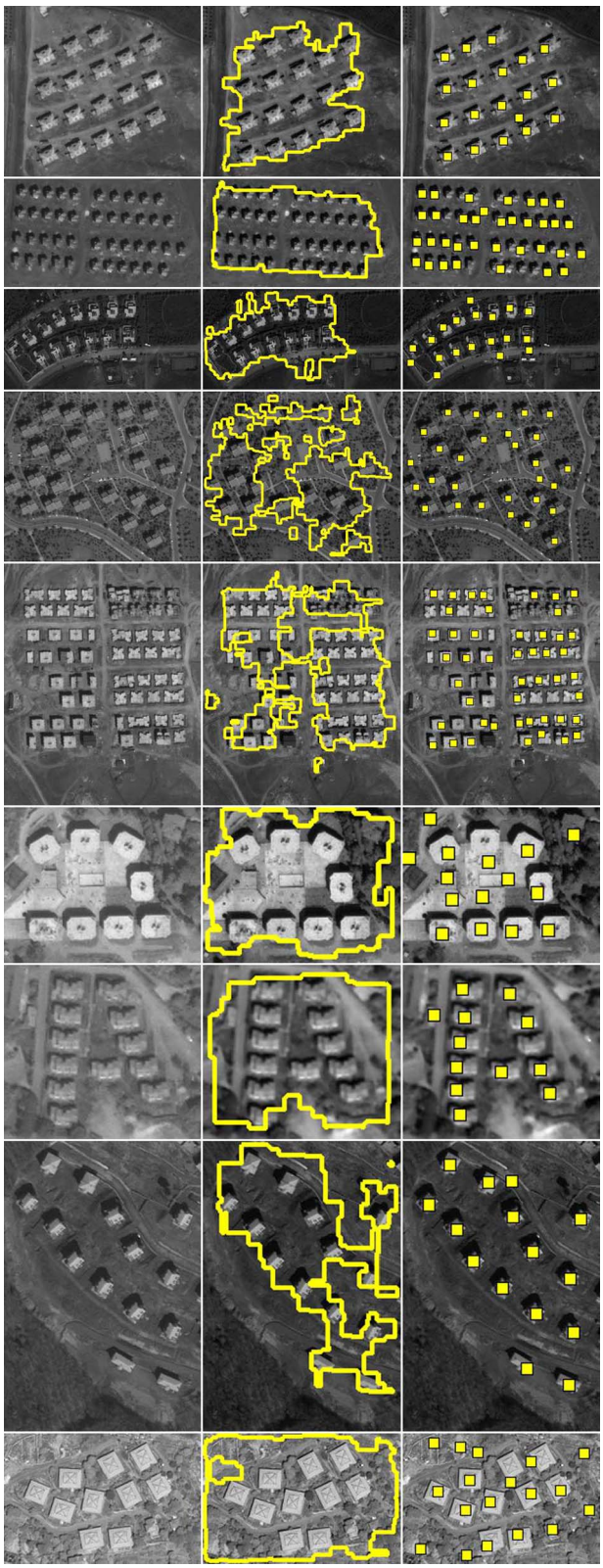


Fig. 11. Test results for Ankara$_{\{1,2,3,4,5\}}$ and Istanbul$_{\{1,2,3,4\}}$ images for each row separately. (First column) Test images. (Second column) Detected urban areas. (Third column) Detected buildings.

ratio), and P_f stands for probability of false alarm (false urban-area-detection ratio). The size of each image and the number of urban-area pixels (in the ground truth image), labeled as "UA," are also given in this table.

TABLE I
URBAN-AREA-DETECTION PERFORMANCES (IN PERCENTAGES) FOR TEST IMAGES

Image Name	Size (pixels)	UA (pixels)	$\mathbf{P_d}(\%)$	$\mathbf{P_f}(\%)$
$Adana_1$	166×318	17848	98.06	22.42
$Adana_2$	302×401	9735	90.96	39.48
$Adana_3$	192×213	23283	94.87	5.62
$Adana_4$	143×210	15035	99.26	9.86
$Adana_5$	242×450	36058	91.84	1.55
$Adana_6$	293×354	35805	94.83	8.77
$Adana_7$	289×324	31276	95.40	7.64
$Adana_8$	235×265	28747	84.26	1.62
$Adana_9$	166×208	25094	89.40	1.05
$Adana_{10}$	192×200	22016	94.78	13.77
$Adana_{11}$	228×186	25332	95.39	6.06
$Adana_{12}$	257×267	37810	76.88	1.53
$Adana_{13}$	325×289	53482	85.30	0.54
$Adana_{14}$	336×275	26033	84.37	3.30
$Adana_{15}$	334×264	26288	86.81	15.30
$Adana_{16}$	119×173	14517	97.04	2.41
$Adana_{17}$	216×306	29242	87.62	12.69
$Adana_{18}$	258×247	47031	92.42	5.22
$Adana_{19}$	171×185	22382	98.05	8.13
$Ankara_1$	208×240	25226	90.36	3.92
$Ankara_2$	130×239	20037	94.08	1.07
$Ankara_3$	145×285	16986	95.13	15.61
$Ankara_4$	271×315	40513	84.60	13.16
$Ankara_5$	340×278	48769	81.69	1.36
$Istanbul_1$	128×162	13266	97.49	17.84
$Istanbul_2$	98×111	6141	99.69	31.95
$Istanbul_3$	248×169	21241	73.70	4.42
$Istanbul_4$	127×193	12725	99.49	57.82
Total		**739263**	**89.62**	**8.03**

We obtain 89.62% correct urban-area detection and 8.03% false-alarm rates over 739 263 pixels of total urban area labeled from 28 test images. For our diverse test set, this result is very promising. Table I can give us further information on our urban-area-detection method. The lowest urban-area-detection rate is obtained on the Istanbul$_3$ image. Here, buildings are sparse. Therefore, the region does not show strong urban characteristics. In other words, there are many nonurban pixels surrounding building clusters. Since our urban-area-detection method depends on building-cluster detection, this poor performance is reasonable. However, the false-alarm rate for this image is fairly low. The highest false-alarm rate is obtained on the Istanbul$_4$ image. The reason for this high false-alarm rate is that some nearby regions are also taken as urban. The best urban-area-detection rate is obtained on the Istanbul$_2$ image.

B. Building Detection

Having detected urban areas, we next concentrate on detecting buildings. As can be seen in Figs. 9–11 (third columns), our method detects buildings in each urban area fairly well. To quantify building-detection results, we apply the following methodology. If a part of a building is detected, we assume it to be detected correctly. If our method detects a building multiple times (particularly for large buildings), we assume it to be detected correctly. If a building is in construction (showing building characteristics), we expect our method to detect it. Based on these assumptions, we provide the building-detection performance for all test images in Table II. We tabulate the total number of buildings in each test image under the column "Buildings." In this table, "TD" stands for the total number of buildings correctly detected in the image. "FA" stands for the

TABLE II
BUILDING-DETECTION PERFORMANCES FOR TEST IMAGES

Image Name	Buildings	TD	FA	TD (%)	FA (%)
$Adana_1$	29	25	4	86.2	13.8
$Adana_2$	9	9	6	100.0	66.7
$Adana_3$	31	27	1	87.1	3.2
$Adana_4$	21	21	4	100.0	19.0
$Adana_5$	54	49	1	90.7	1.9
$Adana_6$	47	45	6	95.7	12.8
$Adana_7$	28	28	9	100.0	32.1
$Adana_8$	24	21	2	87.5	8.3
$Adana_9$	24	23	5	95.8	20.8
$Adana_{10}$	14	13	9	92.9	64.3
$Adana_{11}$	21	19	5	90.5	23.8
$Adana_{12}$	32	29	4	90.6	12.5
$Adana_{13}$	67	60	2	89.6	3.0
$Adana_{14}$	33	29	0	87.9	0.0
$Adana_{15}$	28	27	6	96.4	21.4
$Adana_{16}$	23	17	1	73.9	4.3
$Adana_{17}$	24	21	8	87.5	33.3
$Adana_{18}$	70	48	13	68.6	18.6
$Adana_{19}$	24	23	7	95.8	29.2
$Ankara_1$	18	18	1	100.0	5.6
$Ankara_2$	44	34	1	77.3	2.3
$Ankara_3$	14	14	5	100.0	35.7
$Ankara_4$	23	23	5	100.0	21.7
$Ankara_5$	61	47	0	77.0	0.0
$Istanbul_1$	11	11	4	100.0	36.4
$Istanbul_2$	13	12	0	92.3	0.0
$Istanbul_3$	14	14	1	100.0	7.1
$Istanbul_4$	11	11	7	100.0	63.6
Total	**812**	**718**	**117**	**88.4**	**14.4**

total number of false alarms in the image. We also provide the percentages of TD and FA for each test image in the last two columns of the table.

In 28 different test images, our method detected 718 of 812 buildings correctly. This corresponds to an 88.4% correct detection rate. There are also 117 false alarms in building detection, and this corresponds to a 14.4% false-alarm rate. On such a diverse test set, these results are very promising.

Next, we consider some test images in detail. The lowest building detection rate is obtained on the Adana_{16} image. The main reason for this poor performance is that some buildings are closely spaced, and they are small in this test site. Therefore, some of the buildings could not be detected. The highest false-alarm rate is obtained on the Adana_2 image. Some road segments are also labeled as buildings in this test image. This is the main reason for such a high false-alarm rate. The highest building detection rate is obtained on the $\text{Adana}_{\{2,4,7\}}$, $\text{Ankara}_{\{1,3,4\}}$, and $\text{Istanbul}_{\{1,3,4\}}$ images. In these test images, buildings are well separated and distinctive. Therefore, our method works fairly well on them. The lowest false-alarm rate is obtained on the Adana_{14}, Ankara_5, and Istanbul_2 images. In these test images, there are no buildinglike structures besides the actual buildings. This leads to a low false-alarm rate.

Aside from these test images, there are other noteworthy test sites. For the Adana_5 test image, the buildings are closely spaced. However, most of the buildings are correctly detected. For the Adana_{11} test image, the buildings are occluded by trees. It is even hard for a human expert to detect buildings in this image. However, again, most of the buildings are correctly detected. For the Adana_{15} test image, there are regularly spaced trees near buildings. One may think that, our building-detection method may fail. However, for this test image, there are only six false alarms, and almost all of the buildings are correctly detected. In the Adana_8 and Adana_9 test images, the contrast between buildings and background is fairly low. However, correct building-detection rates for these images are fairly well. For the $Ankara_3$ test image, there are complex buildings having different parts with different intensity values, such as with two different kinds of rooftops. This is also the case in the Istanbul_3 image. Our method can detect all of these buildings in these images without any problem. Based on these experimental results, we can conclude that our building-detection method works fairly well on a diverse test set.

There are also some minor shortcomings for our building-detection method. First, very closely spaced buildings cannot be separated. They are detected as a single building. Second, the contrast between the background and the building is very important for detection. If the contrast is low, the building may be missed. Third, some ground formations resembling buildings (such as sand hillocks) are major reasons for false detections. Finally, our building-detection method is limited with the templates used. If one needs to detect more complex buildings, then the solution is to use the corresponding template.

TABLE III
URBAN-AREA-DETECTION RESULTS ON THE Adana_8 IMAGE. DIFFERENT VARIATIONS

	Normal BF		Fast BF	
Template	$P_d(\%)$	$P_f(\%)$	$P_d(\%)$	$P_f(\%)$
Bright	77.82	1.63	74.05	0.61
Dark	70.25	0.01	64.93	0.34
Both	84.26	1.62	80.85	0.82

C. Tests on Different Modules

As mentioned in previous sections, our urban-area- and building-detection methods are composed of many submodules (such as BF, SIFT keypoint extraction, and building template matching). Therefore, in this section, we test the effect of these different submodules on the final detection results. We pick the Adana_8 test image for this purpose. First, we test the effect of the bilateral filter. There is also a faster version of BF proposed by Paris and Durand [24]. We call this filtering as "fast BF." Instead of the normal bilateral filtering (we call it as "normal BF"), we use this fast implementation. Second, we test the effect of the building templates for detection. Instead of using two building templates (as dark and bright), we test using each template alone. Based on these variations, we provide the urban-area-detection results in Table III.

As can be seen in Table III, in detecting the urban area, the dark building template has the lowest performance. Using both building templates drastically improves the performance. Using normal or fast BF has a significant effect in urban-area-detection performance. We also provide the building-detection results based on the same variations in Table IV.

As can be seen in Table IV, in detecting buildings, the bright building template has the lowest performance. Using the dark building template or both templates improves the building-detection performance. The type of the bilateral filter does not affect the building-detection performance much. However,

TABLE IV
BUILDING-DETECTION RESULTS ON THE Adana$_8$ IMAGE. DIFFERENT VARIATIONS

	Normal BF		Fast BF	
Template	TD	FA	TD	FA
Bright	13	1	11	1
Dark	20	2	19	1
Both	21	2	19	1

using the fast bilateral filter implementation slightly decreases the building-detection performance in all template settings.

D. Tests on Parameter Values

In order to validate our parameter settings, we explore them in detail here. As a benchmark, we pick the building-detection results [in terms of true detection (TD)] on the Adana$_8$ test image. We change the value of each parameter and plot the TD values in Fig. 12. As can be seen, for ϵ_1, the optimal parameter is around 30. As we increase ϵ_1, the TD performance does not decrease drastically. For ϵ_3, the acceptable value is around 4. Increasing ϵ_3 further does not change the result further. We obtain a similar result for ϵ_4. The optimal value is around 0.1, and increasing ϵ_4 does not affect the result much. These experiments indicate the fairly robust characteristics of our parameter adjustment methods. Further details on the physical meanings of these parameters can be found in the previous sections.

E. Comparison With Derivative of Morphological Profiles

We also compare our building-detection method with the well-known derivative morphological profiles (DMP) method [1]. To note here, DMP is not introduced for building detection alone. However, because of its strength, it can also be used for building detection in panchromatic images. Therefore, we pick three test images and provide their segmentation results using DMP in Fig. 13. In testing DMP, we applied its principal component-analysis-based implementation. In segmentation, we picked the best threshold value for the Adana$_8$ test image. As can be seen in Fig. 13, detected buildings using DMP is not as good as our method. The main reason for this difference is that we designed our system to building and urban-area detection alone. On the other hand, the time needed for a DMP operation on the Adana$_8$ test image is 19.19 s. This timing is much less than the time needed for our method. Next, we discuss our method's timing requirements in detail.

F. Computation Times

We finally tabulate the time needed to the detect urban areas and buildings. To note here, timing directly depends on the test image. As the number of buildings in a test image increases, the number of local features will also increase. Therefore, the graph matching and graph cut algorithms will need more computation times. To give an idea for the possible reader, we consider the Adana$_8$ test image as a benchmark. We tabulate all CPU timings for each module in Table V. In reporting these results, we used a PC with an Intel Core2Due processor with a 2.13-GHz clock speed and having 4 GB of RAM. We used Matlab as our coding platform except the original SIFT implementation. The SIFT package is taken from D. Lowe's web site, and it is written in C. If we were to use the C platform for coding all modules, we would have a much faster system.

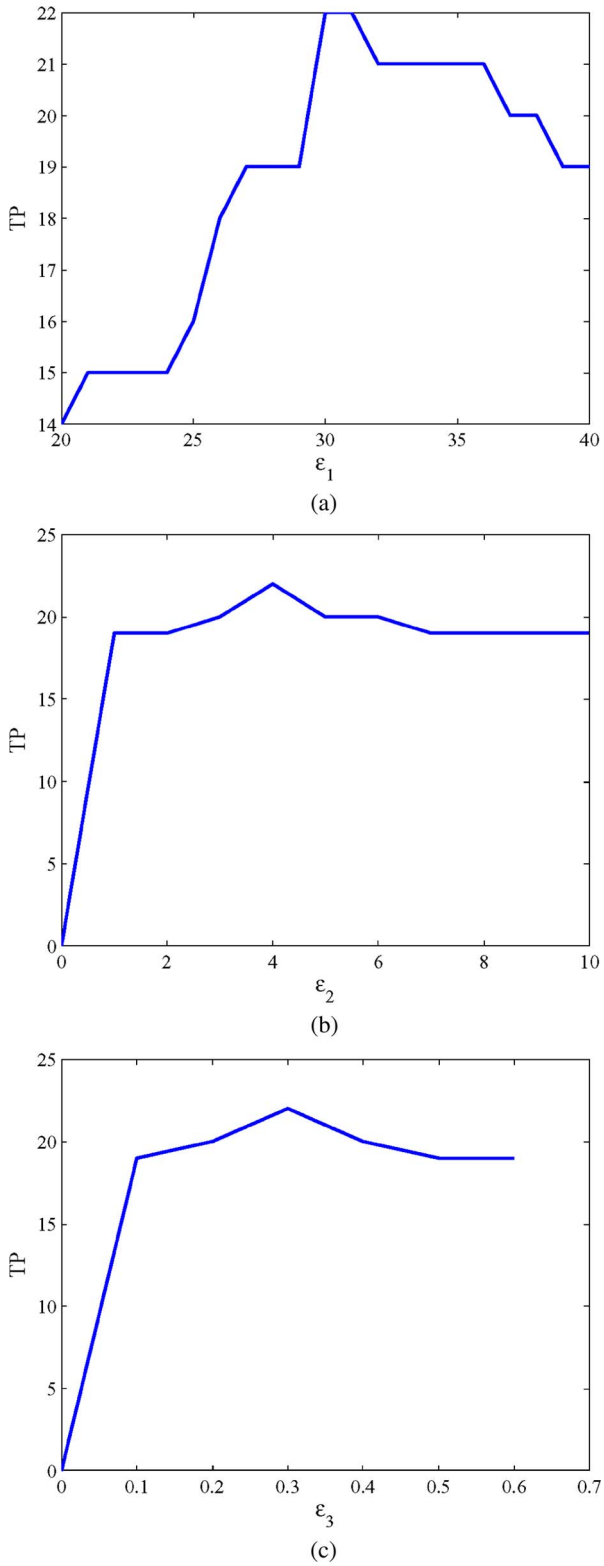
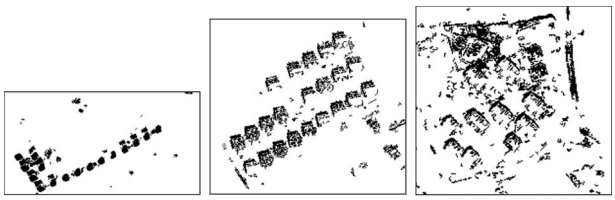


Fig. 12. TD versus ϵ values for the building-detection performance on the Adana$_8$ test image. (a) TD versus ϵ_1. (b) TD versus ϵ_3. (c) TD versus ϵ_4.



(a) (b) (c)

Fig. 13. DMP test results on the Adana_1, Adana_8, and Adana_{10} images. (a) Adana_1. (b) Adana_8. (c) Adana_{10}.

TABLE V
CPU TIMES (IN SECONDS) FOR URBAN-AREA- AND BUILDING-DETECTION OPERATIONS ON THE Adana_8 TEST IMAGE

Module / Template	Dark	Both
Upsampling	0.19	0.19
Normal BF	62.67	62.67
Fast BF	6.51	6.51
SIFT keypoints	0.19	0.38
Graph matching	12.42	18.73
Graph cut	117.06	187.47

In Table V, we provide both normal and fast BF implementations. Similarly, we provide the computation times for using only the dark building template and both templates. We can summarize the different scenarios as follows. Using normal BF and both templates, urban-area-detection operation requires 81.97 s. In the same setting, building detection requires 269.44 s. This scenario is for obtaining the best performances for both urban-area and building detection. If we can tolerate slightly lower detection performances, then we can use fast BF and only the dark template. In this scenario, urban-area detection requires only 19.31 s. Here, building detection only requires 136.37 s. The possible reader should select the suitable scenario (both in terms of detection performance and CPU time needed) for his or her needs.

VI. FINAL REMARKS

This study focuses on urban-area and building detection on VHR satellite images. To do so, we introduce novel methods based on SIFT keypoints, multiple subgraph matching, and graph cut methods. We picked two template building images, one representing dark buildings and the other representing bright buildings. We obtain their SIFT keypoints. We also obtain the SIFT keypoints for the test image. Then, by applying multiple subgraph matching between template and test image SIFT keypoints, we detect the urban area in the test image. From the detected urban area, we detect separate buildings using a novel graph cut method. We test both urban-area- and building-detection methods on a diverse and representative image set. We obtain an 89.62% correct urban-area-detection performance with an 8.03% false-alarm rate. This performance on such a diverse test set is noteworthy. False alarms in urban-area detection mostly occur because of terrain formations resembling buildings (such as sand hillocks) in satellite images. The building-detection performance on our test image set is 88.4% with a 14.4% false-alarm rate. This performance on 28 test images having different building characteristics is very promising. Our building-detection method may not detect buildings if the contrast between their rooftop and the background is low. Since we are only using the grayscale information in the satellite image, this is reasonable. Moreover, some closely spaced buildings may not be detected correctly. This is the major drawback of our method. In a future study, we plan to embed the multispectral information to detect closely spaced buildings. Finally, we should mention that the proposed method can be generalized to detect any kind of object in satellite images.

REFERENCES

[1] J. A. Benediktsson, M. Pesaresi, and K. Arnason, "Classification and feature extraction for remote sensing images from urban areas based on morphological transformations," *IEEE Trans. Geosci. Remote Sens.*, vol. 41, no. 9, pp. 1940–1949, Sep. 2003.

[2] S. Bhagavathy and B. S. Manjunath, "Modeling and detection of geospatial objects using texture motifs," *IEEE Trans. Geosci. Remote Sens.*, vol. 44, no. 12, pp. 3706–3715, Dec. 2006.

[3] L. Bruzzone and L. Carlin, "A multilevel context-based system for classification of very high spatial resolution images," *IEEE Trans. Geosci. Remote Sens.*, vol. 44, no. 9, pp. 2587–2600, Sep. 2006.

[4] L. Bruzzone, M. Chi, and M. Marconcini, "A novel transductive SVM for semisupervised classification of remote-sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 44, no. 11, pp. 3363–3373, Nov. 2006.

[5] M. Elad, "On the origin of the bilateral filter and ways to improve it," *IEEE Trans. Image Process.*, vol. 11, no. 10, pp. 1141–1151, Oct. 2002.

[6] M. Fauvel, J. Chanussot, and J. A. Benediktsson, "Decision fusion for the classification of urban remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 44, no. 10, pp. 2828–2838, Oct. 2006.

[7] L. M. Fonte, S. Gautama, W. Philips, and W. Goeman, "Evaluating corner detectors for the extraction of man made structures in urban areas," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2005, pp. 237–240.

[8] P. Gamba, F. D. Acqua, G. Lisini, and G. Trianni, "Improved VHR urban area mapping exploiting object boundaries," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 8, pp. 2676–2682, Aug. 2007.

[9] S. Gautama, R. Bellens, G. D. Tre, and W. Philips, "Relevance criteria for spatial information retrieval using error-tolerant graph matching," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 4, pp. 810–817, Apr. 2007.

[10] C. S. Hernandez, D. S. Boyd, and G. M. Foody, "One-class classification for mapping a specific land-cover class: SVDD classification of Fenland," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 4, pp. 1061–1073, Apr. 2007.

[11] X. Huang, L. Zhang, and P. Li, "An adaptive multiscale information fusion approach for feature extraction and classification of IKONOS multispectral imagery over urban areas," *IEEE Geosci. Remote Sens. Lett.*, vol. 4, no. 4, pp. 654–658, Oct. 2007.

[12] V. Karathanassi, C. Iossifidis, and D. Rokos, "A texture-based classification method for classifying built areas according to their density," *Int. J. Remote Sens.*, vol. 21, no. 9, pp. 1807–1823, Jun. 2000.

[13] A. Katartzis and H. Sahli, "A stochastic framework for the identification of building rooftops using a single remote sensing image," *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 1, pp. 259–271, Jan. 2008.

[14] T. Kim and J. P. Muller, "Development of a graph-based approach for building detection," *Image Vis. Comput.*, vol. 17, no. 1, pp. 3–17, Jan. 1999.

[15] A. Laha, N. R. Pal, and J. Das, "Land cover classification using fuzzy rules and aggregation of contextual information through evidence theory," *IEEE Trans. Geosci. Remote Sens.*, vol. 44, no. 6, pp. 1633–1641, Jun. 2006.

[16] J. Lee and O. K. Ersoy, "Consensual and hierarchical classification of remotely sensed multispectral images," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 9, pp. 2953–2963, Sep. 2007.

[17] S. Lee and R. G. Lathrop, "Subpixel analysis of Landsat ETM^+ using self-organizing map (SOM) neural networks for urban land cover characterization," *IEEE Trans. Geosci. Remote Sens.*, vol. 44, no. 6, pp. 1642–1654, Jun. 2006.

[18] A. Lorette, X. Descombes, and J. Zerubia, "Texture analysis through a Markovian modelling and fuzzy classification: Application to urban area extraction from satellite images," *Int. J. Comput. Vis.*, vol. 36, no. 3, pp. 221–236, Feb. 2000.

[19] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[20] J. M. Mari, L. Bruzzone, and G. C. Valls, "A support vector domain description approach to supervised classification of remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 8, pp. 2683–2692, Aug. 2007.
[21] S. D. Mayunga, Y. Zhang, and D. J. Coleman, "Semi-automatic building extraction utilizing Quickbird imagery," in *Proc. ISPRS Workshop CMRT*, 2005, pp. 131–136.
[22] K. Mikolojczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 10, pp. 1615–1630, Oct. 2005.
[23] M. Molinier, J. Laaksonen, and T. Hame, "Detecting man-made structures and changes in satellite imagery with a content-based information retrieval system built on self-organizing maps," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 4, pp. 861–874, Apr. 2007.
[24] S. Paris and F. Durand, "A fast approximation of the bilateral filter using a signal processing approach," *Int. J. Comput. Vis*. [Online]. Available: http://www.springerlink.com/content/l2262j3814733q56/
[25] J. Peng, D. Zhang, and Y. Liu, "An improved snake model for building detection from urban aerial images," *Pattern Recognit. Lett.*, vol. 26, no. 5, pp. 587–595, Apr. 2005.
[26] K. Segl and H. Kaufmann, "Detection of small objects from high-resolution panchromatic satellite imagery based on supervised image segmentation," *IEEE Trans. Geosci. Remote Sens.*, vol. 39, no. 9, pp. 2080–2083, Sep. 2001.
[27] M. Sonka, V. Hlavac, and R. Boyle, *Image Processing, Analysis and Machine Vision*, 2nd ed. Pacific Grove, CA: PWS Publ., 1999.
[28] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proc. Int. Conf. Comput. Vis.*, 1998, pp. 839–846.
[29] C. Ünsalan, "Measuring land development in urban regions using graph theoretical and conditional statistical features," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 12, pp. 3989–3999, Dec. 2007.
[30] C. Ünsalan and K. L. Boyer, "Classifying land development in high-resolution panchromatic satellite images using straight-line statistics," *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 4, pp. 907–919, Apr. 2004.
[31] C. Ünsalan and K. L. Boyer, "A system to detect houses and residential street networks in multispectral satellite images," *Comput. Vis. Image Underst.*, vol. 98, no. 3, pp. 432–461, Jun. 2005.
[32] C. Ünsalan and K. L. Boyer, "A theoretical and experimental investigation of graph theoretical measures for land development in satellite imagery," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 4, pp. 575–589, Apr. 2005.
[33] W. Wei and Y. Xin, "Feature extraction for man-made objects segmentation in aerial images," *Mach. Vis. Appl.*, vol. 19, no. 1, pp. 57–64, Jan. 2008.
[34] K. Zhang, J. Yan, and S. C. Chen, "Automatic construction of building footprints from airborne LIDAR data," *IEEE Trans. Geosci. Remote Sens.*, vol. 44, no. 9, pp. 2523–2533, Sep. 2006.
[35] P. Zhong and R. Wang, "A multiple conditional random fields ensemble model for urban area detection in remote sensing optical images," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 12, pp. 3978–3988, Dec. 2007.

Beril Sırmaçek (S'07) received the B.Sc. and M.S. degrees from the Department of Electronics and Communication Engineering, Yildiz Technical University, Istanbul, Turkey, in 2006 and 2007, respectively. She has been working toward the Ph.D. degree in the Department of Electrical and Electronics Engineering, Yeditepe University, Istanbul, since 2007.

She is a Research Assistant and member of the Computer Vision Research Laboratory, Yeditepe University. Her research interests are object detection in remotely sensed imagery, pattern recognition, and invariants for object recognition.

Cem Ünsalan (S'92–M'03) received the B.Sc. degree from Hacettepe University, Ankara, Turkey, in 1995, the M.Sc. degree from Boğaziçi University, Istanbul, Turkey, in 1998, and the Ph.D. degree from The Ohio State University, Columbus, in 2003, all in electrical and electronics engineering.

Since 2003, he has been with the Department of Electrical and Electronics Engineering, Yeditepe University, Istanbul, Turkey, where he is also the Director of the Computer Vision Research Laboratory. His research interests include computer vision and pattern recognition, and he has published extensively on these topics.