A Probabilistic Framework to Detect Buildings in Aerial and Satellite Images

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

***Abstract*—Detecting buildings from very high resolution (VHR) aerial and satellite images is extremely useful in map making, urban planning, and land use analysis. Although it is possible to manually locate buildings from these VHR images, this operation may not be robust and fast. Therefore, automated systems to detect buildings from VHR aerial and satellite images are needed. Unfortunately, such systems must cope with major problems. First, buildings have diverse characteristics, and their appearance (illumination, viewing angle, etc.) is uncontrolled in these images. Second, buildings in urban areas are generally dense and complex. It is hard to detect separate buildings from them. To overcome these difficulties, we propose a novel building detection method using local feature vectors and a probabilistic framework. We first introduce four different local feature vector extraction methods. Extracted local feature vectors serve as observations of the probability density function (pdf) to be estimated. Using a variable-kernel density estimation method, we estimate the corresponding pdf. In other words, we represent building locations (to be detected) in the image as joint random variables and estimate their pdf. Using the modes of the estimated density, as well as other probabilistic properties, we detect building locations in the image. We also introduce data and decision fusion methods based on our probabilistic framework to detect building locations. We pick certain crops of VHR panchromatic aerial and Ikonos satellite images to test our method. We assume that these crops are detected using our previous urban region detection method. Our test images are acquired by two different sensors, and they have different spatial resolutions. Also, buildings in these images have diverse characteristics. Therefore, we can test our methods on a diverse data set. Extensive tests indicate that our method can be used to automatically detect buildings in a robust and fast manner in Ikonos satellite and our aerial images.**

***Index Terms*—Aerial images, building detection, data fusion, decision fusion, Ikonos satellite images, kernel density estimation, local feature vectors.**

I. Introduction

VERY high resolution (VHR) aerial and satellite images provide valuable information. In particular, detecting buildings from these images requires a specific consideration, since this information may be used in several remote sensing applications, such as automated map making, urban planning, and land use analysis. Unfortunately, it is tedious for a human expert to manually label buildings in a given aerial or satellite image for several reasons. First, buildings may be imaged from different viewpoints. They may not have a unique representation. Second, buildings may have complex interaction with the environment (such as occlusion by trees). Moreover, they may occlude each other. Third, the illumination and contrast in the image may not be sufficient to detect buildings reliably. Fourth, these images may cover large geographic areas with many buildings in them. Analyzing the image may take time. Finally, buildings do not have standard size and shape. Therefore, development of robust and fast building detection algorithms on VHR aerial and satellite images has become a necessity.

In the last two decades, researchers have developed automated building detection methods using aerial and satellite images. There are nice reviews on building detection in aerial and satellite images [23], [38]. The interested reader can get detailed information on most of the automated building detection methods from these studies. We next summarize recently introduced building detection methods focusing on the ones related to the proposed method in this paper. Kim and Muller [17] 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. Different from us, they used color aerial images and linear features. Krishnamachari and Chellappa [18] introduced a Markov-random-field (MRF)-based building detection method in aerial images. They benefit from straight line segments in the image and form their MRF-based detection method on their interactions. Compared to ours, this system is more complex. Segl and Kaufmann [29] combined supervised shape classification with unsupervised image segmentation in an iterative way. Their method allows searching small objects (like buildings) in high-resolution satellite images. Molinier *et al.* [25] considered detecting boundaries of manmade structures in satellite images by training a self-organizing map. Gamba *et al.* [11] 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 to detect urban area buildings on VHR imagery. In these studies, there is always a need for a training set. Benediktsson *et al.* [6] used mathematical morphological operations to extract structural information to detect the urban area in satellite images. This method can be used to detect buildings in the image. Ünsalan and Boyer [38] studied multispectral satellite images to detect buildings and street

Manuscript received July 29, 2009; revised February 1, 2010; accepted June 6, 2010. Date of publication July 26, 2010; date of current version December 27, 2010.

B. Sırmaçek was with the Computer Vision Research Laboratory, Department of Electrical and Electronics Engineering, Yeditepe University, 34755 Istanbul, Turkey. She is now with the Department of Photogrammetry and Image Analysis, Remote Sensing Technology Institute (IMF), German Aerospace Center (DLR), 82234 Wessling, Germany (e-mail: Beril.Sirmacek@dlr.de).

C. Ünsalan is with the Computer Vision Research Laboratory, Department of Electrical and Electronics Engineering, Yeditepe University, 34755 Istanbul, Turkey (e-mail: unsalan@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.2010.2053713

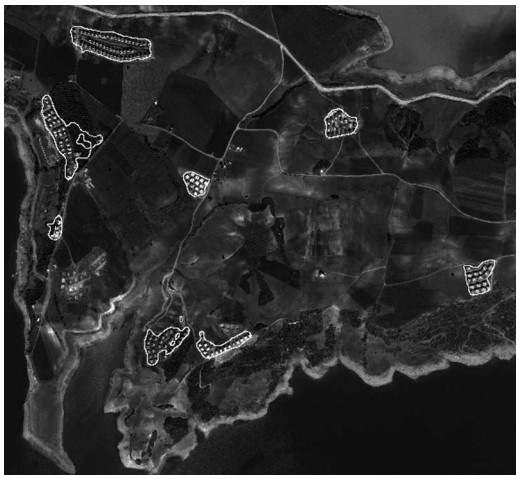

Fig. 1. Urban region detection results (detected boundaries labeled white) from the large Ikonos image acquired over Adana, Turkey.

networks in residential regions. Their method uses vegetation indices, clustering, decomposing binary images, and graph theory. Although this method is promising, multispectral information is needed to detect buildings. Akçay and Aksoy [1] also proposed a novel method for unsupervised segmentation and object detection in high-resolution satellite images. This method also needs multispectral information. Idrissa *et al.* [13] extracted the edges of man-made structures (buildings and roads) using Gabor filters together with the normalized difference vegetation index in SPOT5 images. Comparing the edges of two image sequences taken from the same region, they also detected changes. Different from us, they benefit from multispectral information. In a recent study, we introduced a method to detect buildings in panchromatic Ikonos satellite images using scale-invariant feature transform (SIFT) keypoints and graph theory formalism [32]. This method gives good results, but it has a high computation load. It also depends on template building images as a training set. In a similar framework, Xiong and Zhang [41] used interest points for satellite image matching. There are also various studies focusing on building shape extraction in aerial and satellite images [5], [11], [15], [16], [40]. This is a more complex problem compared to building detection. However, detecting building locations may help extracting building shapes from the image.

In this paper, we assume that the urban regions in the image are obtained using one of the urban region detection methods [4], [7], [9], [22], [34]–[37], [42]. Then, we focus on parts of the urban regions to detect building locations in them. In this paper, we benefit from a previous study to detect urban regions in large Ikonos satellite and aerial images [36]. We provide urban region detection results on a large Ikonos satellite image (acquired over Adana, Turkey) in Fig. 1.

Next, we focus on urban areas in this image step by step. Since our method only depends on local features, we do not need any global information. Therefore, dividing the urban region into sections and detecting buildings in them separately seems to be a good strategy. To detect building locations in these urban areas, we propose a novel probabilistic framework. For detection, we first extract local feature vectors from the given image using four different methods. We take these vectors as observations. To model the distribution of these observations, we use a variable-kernel-based density estimation method. In other words, we model the building locations in a given image as joint random variables and estimate their probability density function (pdf) using observations. The modes of the estimated pdf and their probabilities lead to building locations in the given image. We also introduce data and decision fusion methods using our probabilistic framework to detect buildings. In all these steps, we do not need any training set. We test our method on diverse aerial and Ikonos satellite image sets and provide building detection performances in Section IV.

II. Local Feature Vector Extraction

Our probabilistic building detection method depends on local feature vectors extracted from the test image. Therefore, we introduce four different local feature vector extraction methods in this section. The first method is based on Harris corner detection [12]. The second method is based on our previous study, i.e., gradient-magnitude-based support regions (GMSR) [35]. The third method is based on Gabor filtering in different orientations [39]. Finally, the fourth method is based on features from accelerated segment test (FAST) [27]. Next, we explore each method in detail.

A. Harris-Corner-Based Local Feature Vectors

Fonte *et al.* [10] considered (improved) Harris and Susan corner detectors to obtain the type of structure in a satellite image. They concluded that corner detectors are not sufficient alone to give distinctive information on the type of structure in an image. Schmid *et al.* [28], on the other hand, evaluated and compared different corner detectors for general image processing applications. They concluded that the best results are provided by the Harris corner detector [12]. Therefore, we first pick it to extract local feature vectors.

Harris and Stephens define their corner detector (generally known as the Harris corner detector) in three steps: gradient calculation, matrix formation, and eigenvalue computation. First, we should calculate smoothed (using a Gaussian function) gradients in the x- and y-directions to detect corners in a given grayscale image $I(x,y)$. We define smoothed gradient filters for the x- and y-directions as

$$g_x(x,y) = \frac{-x}{2\pi\tau_g^4}\exp\left(-\frac{x^2+y^2}{2\tau_g^2}\right) \tag{1}$$

$$g_y(x,y) = \frac{-y}{2\pi\tau_g^4}\exp\left(-\frac{x^2+y^2}{2\tau_g^2}\right) \tag{2}$$

where τ_g is the smoothing parameter. We take it as unity in this paper due to the scale of Ikonos satellite and aerial images at hand. Although our method is fairly robust to this parameter, it should be adjusted by the resolution of the image to be analyzed in future studies.

We calculate the smoothed gradients for the image $I(x,y)$ as

$$I_x = g_x(x,y) * I(x,y) \tag{3}$$

$$I_y = g_y(x,y) * I(x,y) \tag{4}$$

where $*$ stands for the 2-D convolution operation.

The Harris corner detector depends on calculating a matrix (related to the autocorrelation function) as

$$A(x,y) = \begin{pmatrix} a_{xx} & a_{xy} \\ a_{xy} & a_{yy} \end{pmatrix} \tag{5}$$

where

$$a_{xx} = \sum_{x_i \in W} \sum_{y_i \in W} I_x^2(x_i, y_i) \tag{6}$$

$$a_{xy} = \sum_{x_i \in W} \sum_{y_i \in W} I_x(x_i, y_i) I_y(x_i, y_i) \tag{7}$$

$$a_{yy} = \sum_{x_i \in W} \sum_{y_i \in W} I_y^2(x_i, y_i). \tag{8}$$

As can be seen, a_{xx}, a_{xy}, and a_{yy} are the gradient magnitudes averaged over a window W. We pick this averaging window width as seven pixels in this paper. We also tested the effect of this parameter in Section IV. For further details on this averaging operation, please see [12]. The eigenvalues of matrix A provide information about the edge in a given location. If both eigenvalues of the matrix at a given location are large, then there is a corner there. Harris and Stephens suggested that exact eigenvalue computation can be avoided by calculating the response function

$$R(A) = |A| - \kappa \, trace^2(A) \tag{9}$$

where κ is a tunable parameter, with values from 0.04 to 0.15 being reported as appropriate in the literature. We picked $\kappa = 0.06$ in this paper based on [12]. We also tested the effect of this parameter in Section IV. Harris and Stephens extract their corner points by checking the local maxima of $R(A)$. For a detailed explanation, please see [12].

As we obtain corner points with their spatial coordinates, we define local feature vectors using them. Other than spatial coordinates, we also add orientation and weight information as follows. First, we calculate the gradient orientation $[O(x,y)]$ and magnitude $[M(x,y)]$ for each image coordinate as

$$O(x,y) = \arctan\left(\frac{I_y(x,y)}{I_x(x,y)}\right) \tag{10}$$

$$M(x,y) = \sqrt{I_x^2(x,y) + I_y^2(x,y)}. \tag{11}$$

For the corner point at coordinate (x_j, y_j), the corresponding orientation is $\theta_j = O(x_j, y_j)$. To assign a weight for the local feature vector, we threshold $M(x,y)$ using Otsu's method in an adaptive manner [26]. As a result, we obtain $B(x,y)$ as a binary image. In this image, pixels having a value of one correspond to strong responses. We obtain connected pixels to (x_j, y_j) in $B(x_j, y_j)$. By definition, two pixels are connected (in a binary image) to each other if there is a path (of pixels with a value of one) connecting them [33]. As we obtain all the connected pixels to (x_j, y_j), we assign their sum as the weight w_j. Therefore, if a candidate local feature vector has more connected pixels, it has more weight. Finally, we have Harris-corner-based local feature vectors as $\vec{k}_h(j) = (x_j, y_j, \theta_j, w_j)$ for $j = 1, \ldots, K_h$. Here, K_h is the total number of detected Harris features.

B. GMSR-Based Local Feature Vectors

We next pick our previous study, i.e., GMSR, to extract local feature vectors [35]. There, we benefited from smoothed gradients to form support regions. We used these to extract structural and conditional statistical features to classify land use. In this paper, we extract support regions using smoothed gradient values, namely, I_x and I_y, given in (3) and (4). To extract support regions, we threshold $M(x,y)$ by the 10% of the maximum gradient magnitude in the considered image $I(x,y)$. The rationale here is as follows. We take the maximum gradient magnitude as a benchmark. After experiments, we observed that even 10% of this value still gives information about the structure in the image. Therefore, we have an adaptive threshold value. Similar to the Harris corner detection method, we obtain $B(x,y)$ as a binary image after thresholding. In this image, pixels having a value of one correspond to support regions. For more details, please see [35].

We define local feature vectors based on the extracted support regions. Therefore, we pick each support region pixel as a local feature vector coordinate. Assume that we have a local feature vector (x_j, y_j). By definition, $B(x_j, y_j) = 1$. We define the orientation and magnitude of the local feature vector having spatial coordinate (x_j, y_j) with the same method that we used in the previous section. As a result, we obtain local feature vectors as $\vec{k}_g(j) = (x_j, y_j, \theta_j, w_j)$ for $j = 1, \ldots, K_g$ from the GMSR. Here, K_g is the total number of detected GMSR features.

C. Gabor-Filtering-Based Local Feature Vectors

We introduce Gabor-filtering-based local feature vector extraction in this section. In this method, the first step is smoothing the image by median filtering to eliminate small noise terms [33]. Then, we apply Gabor filtering in different orientations. Based on these responses, we obtain our local feature vectors. We explore these steps in detail next.

Gabor filters are extensively used in texture segmentation and object recognition [14], [19]. They exhibit desirable characteristics, such as spatial locality and orientation selectivity [39]. Mathematically, the 2-D Gabor filter can be defined as the product of a Gaussian and a complex exponential function as

$$F_\varphi(x,y) = \frac{1}{2\pi\sigma_g^2} \exp\left(-\frac{u^2+v^2}{2\sigma_g^2}\right) \exp(j2\pi f u) \tag{12}$$

where $u = x\cos\varphi + y\sin\varphi$ and $v = -x\sin\varphi + y\cos\varphi$. f is the frequency of the complex exponential signal, φ is the orientation of the Gabor filter, and σ_g is the scale parameter. We observed that, for our test images, $\sigma_g = 1.5$ and $f = 0.65$ are suitable values after extensive testing. By fixing these two parameters, we obtain a filter shape (in the spatial domain) that is very similar to building edges. In Section IV, we experimentally justify these selections. To cover differently oriented building edges, we tested different φ values. We conclude that choosing ten different orientations for Gabor filtering ($\varphi = \{0, \pi/10, 2\pi/10, \ldots, 9\pi/10\}$ radians) is suitable for building detection. We also justify this selection in Section IV. Although we tested these parameters on two different image sets (aerial and satellite), they should be adjusted w.r.t. the image resolution at hand in future studies.

We can detect building edges and corners in a test image using Gabor filtering. For a test image $I(x,y)$ (with size $N \times M$), we benefit from the real part of the Gabor filter response as

$$G_{\varphi}(x,y) = \Re\mathfrak{e}\left\{I(x,y) * F_{\varphi}(x,y)\right\}. \tag{13}$$

$G_{\varphi}(x,y)$ is maximum for image regions having similar characteristics with that of the filter. To extract local feature vector spatial coordinates, we first search for the local maxima in $G_{\varphi}(x,y)$ for $x = 1,\ldots,N$ and $y = 1,\ldots,M$. If any pixel (x_j, y_j) in $G_{\varphi}(x,y)$ has the largest value among its eight neighbors, $G_{\varphi}(x_j,y_j) > G_{\varphi}(x_n,y_n) \forall (x_n,y_n) \in \{(x_j - 1, y_j - 1), (x_j, y_j - 1), \ldots, (x_j + 1, y_j + 1)\}$; we call it as a local maximum. It is a candidate for being a local feature vector coordinate. Next, we check the amplitude of the filter response $G_{\varphi}(x_j,y_j)$. We call our local maximum (x_j,y_j) as a candidate local feature vector coordinate if and only if $G_{\varphi}(x_j,y_j) > \alpha$. To handle different images, we obtain α using Otsu's method on $G_{\varphi}(x,y)$ in an adaptive manner for each image separately [26]. Therefore, we eliminate weak candidate local feature vectors in future calculations.

As we obtain the spatial coordinates of local feature vectors in one Gabor filter direction, we assign an orientation and weight to them. We obtain the weight for each local feature vector similar to the methods in the previous sections. Here, we obtain our binary image $B(x,y)$ by thresholding $G_{\varphi}(x,y)$ with α for weight calculations. However, we assign the orientation different than the two previous methods as follows. We check for the orientations in the eight neighbors of (x_j,y_j) and pick the orientation (θ_j) as the one having the highest magnitude. We applied this procedure to obtain robust orientation information. We apply this procedure in all φ-directions and obtain Gabor-filtering-based local feature vectors as $\vec{k}_f(j) = (x_j, y_j, \theta_j, w_j)$ for $j = 1,\ldots,K_f$. Here, K_f is the total number of detected Gabor-filtering-based features.

D. FAST-Based Local Feature Vectors

Rosten *et al.* [27] introduced the FAST method to detect corners in images in a fast and reliable manner. The method depends on wedge-model-style corner detection and machine learning techniques. This method can briefly be explained as follows. For each corner candidate pixel, its 16 neighbors are checked. If there exist nine contiguous pixels passing a set of tests, the candidate pixel is labeled as a corner. These tests are done using machine learning techniques to speed up the operation. For a detailed explanation of this method, please see [27].

We finally pick FAST to extract local feature vectors. Therefore, we pick each extracted corner pixel as a local feature vector coordinate. Assume that we have a local feature vector (x_j,y_j). We define the orientation and magnitude of the local feature vector having spatial coordinate (x_j,y_j) with the same method that we used in the Harris-corner-based local feature extraction method. As a result, we obtain local feature vectors as $\vec{k}_s(j) = (x_j, y_j, \theta_j, w_j)$ for $j = 1,\ldots,K_s$ from the FAST method. Here, K_s is the total number of detected FAST-based features.

We pick the $Adana_1$ test image shown in Fig. 2 and provide the spatial coordinates of local feature vectors extracted by the four methods mentioned earlier. As can be seen, in $\vec{k}_h$, there are only corner pixel coordinates (as expected). In addition, some corners are also missing. Therefore, we have the least number of local feature vectors with the Harris corner detection method. $\vec{k}_g$ includes both corner and building edge pixel coordinates. $\vec{k}_f$ and $\vec{k}_s$ also include both corner and building edge pixel coordinates. However, they also have some extra road segment and tree structure pixel coordinates (resembling buildings). Next, we use these local feature vectors to detect buildings.

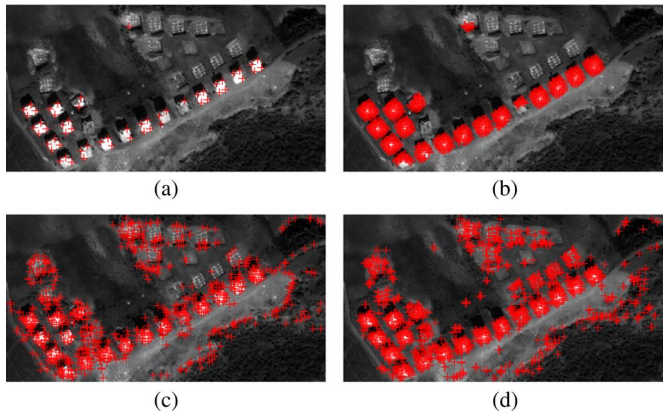

Fig. 2. $Adana_1$ test image and local feature vector coordinates extracted with four different methods. (a) $\vec{k}_h$. (b) $\vec{k}_g$. (c) $\vec{k}_f$. d) $\vec{k}_s$.

III. Building Detection

Each local feature vector indicates a building to be detected in the image. However, only one of them is not sufficient alone to detect a building. In fact, the more local feature vectors a building has, the more probable its detection becomes. On the other hand, we do not know how many buildings there are in the image. Therefore, we formulate our building detection method with a probabilistic framework. To do so, we represent possible building locations as discrete joint random variables. We then estimate their pdf by taking local feature vectors as observations. Here, we benefit from a variable-kernel density estimation method. Next, we introduce our probabilistic building detection framework using it. Finally, we introduce data and decision fusion methods based on our probabilistic framework to detect buildings. To explain our method in detail, we start with kernel-based density estimation.

A. Kernel-Based Density Estimation

Silverman [30] defines the kernel density estimator for a discrete and bivariate pdf as follows. First, the bivariate kernel function $[N(x,y)]$ should satisfy the conditions

$$\sum_x \sum_y N(x,y) = 1 \tag{14}$$

$$N(x,y) \geq 0 \qquad \forall (x,y). \tag{15}$$

The pdf estimator with kernel $N(x,y)$ is defined by

$$p(x,y) = \frac{1}{nh} \sum_{i=1}^{n} N\left(\frac{x - x_i}{h}, \frac{y - y_i}{h}\right) \tag{16}$$

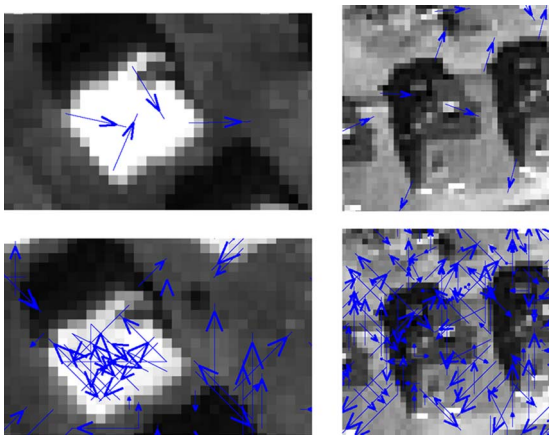

Fig. 3. (First column: bright; second column: dark) Two sample buildings and (first row: Harris; second row: Gabor) local feature vector directions on them.

where h is the window width (also called the smoothing parameter) and (x_i, y_i) for $i = 1, \ldots, n$ are observations from the pdf to be estimated.

If observations cannot be represented reliably by a fixed kernel function, then a variable kernel function can be used. This is achieved by adaptation of the amount of smoothing to the local density of the data (observation). Hence, the scale parameter is allowed to vary from one observation point to another. Furthermore, the estimate is constructed similarly to the classical kernel estimate. The pdf estimate given in (16) then becomes

$$p_v(x,y) = \frac{1}{nh}\sum_{i=1}^{n}\frac{1}{\sigma_i}N\left(\frac{x-x_i}{h\sigma_i}, \frac{y-y_i}{h\sigma_i}\right) \qquad (17)$$

where σ_i is the variable scale parameter for $i = 1, \ldots, n$.

B. Detecting Buildings Using Variable-Kernel-Based Density Estimation

Since we do not know the total number of buildings in a given test image, we use the variable-kernel-based density estimation method to detect them. As we mentioned previously, we use local feature vectors ($\vec{k}_h, \vec{k}_g, \vec{k}_f,$ and $\vec{k}_s$) as observations to estimate the pdf. Without loss of generality, we explain pdf estimation on a generic local feature vector $\vec{k} = (x_i, y_i, \theta_i, w_i)$ for $i = 1, \ldots, K_i$. These vectors provide information on buildings to be detected. However, their spatial coordinates are not sufficient enough since they represent either building corners or edges. To detect a building, we need an ensemble of edges or corners. To achieve this, we adjust the effect of local feature vectors w.r.t. their orientation and weight. In doing so, we observed that, for bright building corners and edges, gradient directions are toward the building center. For dark building edges, gradient directions are away from the building center. However, due to the nonuniformity of rooftop pixel values and shadows around buildings, some gradient directions will still be toward building centers. In Fig. 3, we show the local feature vector directions using Harris-corner- and Gabor-filtering-based local feature vectors on one bright and one dark building. As can be seen, for the bright building, almost all directions are toward the building center for both feature extraction methods.

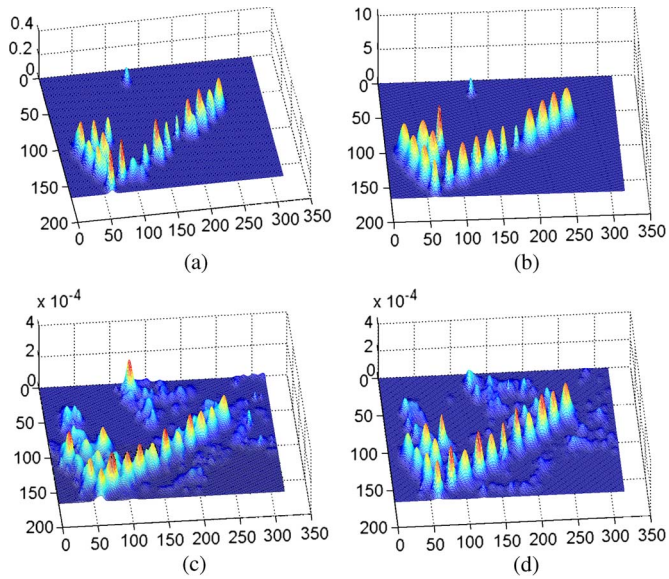

Fig. 4. $Adana_1$ test image kernel density estimation (pdf) results for four local feature vector extraction methods. (a) $\vec{k}_h$. (b) $\vec{k}_g$. (c) $\vec{k}_f$. (d) $\vec{k}_s$.

For the dark building, few Harris-corner-based local feature vectors have directions toward the building center. However, for the Gabor-filtering-based local feature vectors, we have more feature vectors toward the building center.

Based on the aforementioned observations, each local feature vector will have its effect as $\hat{x}_i = x_i + 0.5w_i\sin(\theta_i)$ and $\hat{y}_i = y_i + 0.5w_i\cos(\theta_i)$. In other words, each local feature vector is shifted in the direction of θ_i. We apply half of the weight w_i in shifting to approximately locate the building center. We also take $N(x,y)$ in (17) as a Gaussian symmetric pdf, which is used in most density estimation applications. Using these adjusted and updated observations, we form the estimated pdf as

$$p_b(x,y) = \frac{1}{R}\sum_{i=1}^{K_i}\frac{1}{\sqrt{2\pi\sigma_i}}\exp\left(-\frac{(x-\hat{x}_i)^2+(y-\hat{y}_i)^2}{2\sigma_i}\right) \qquad (18)$$

where $\sigma_i = w_i$. We will obtain the modes of $p_b(x,y)$ in detecting buildings. Therefore, we did not use a normalized kernel in this equation. However, we normalized the final estimated pdf $p_b(x,y)$ by the normalizing constant R.

Before proceeding further, we show the kernel density estimation results in Fig. 4 using the four local feature vector extraction methods introduced earlier. As can be seen, for the Harris- and GMSR-based corner detector, the estimated pdf is smooth. For the Gabor-filtering- and FAST-based methods, the estimated pdfs have more fluctuations around nonbuilding regions. This is because of the falsely extracted local features around these regions by the Gabor-filtering- and FAST-based methods.

The estimated pdf $p_b(x,y)$ will be multimodal since we have an unknown number of buildings to be detected in the image. We hypothesize that the modes of this pdf are possible building centers. However, all modes do not correspond to a building center. We assume that, for a location to be a building center, it should exceed at least a minimum probability value. We adjust this value in an adaptive manner as follows. Since we

are detecting buildings in an urban region, we assume that there is at least one building there. Remember that we applied urban region detection as a preprocessing operation. Therefore, we pick the mode having the highest probability as a building location, i.e., $(x_b, y_b) = \arg\max_{(x,y)} p_b(x, y)$. Then, we pick the remaining modes having probabilities at least $0.4 \times p_b(x_b, y_b)$ as building locations. We experimentally justified the multiplier 0.4 in Section IV. This method also automatically assigns probabilities to the detected building locations. The higher probability the location has, the more probable it represents a building. This information may be of use in some other applications.

C. Data and Decision Fusion for Building Detection

The four local feature vector extraction methods extract different information from the same image. In the previous section, we separately used these to detect building locations. Their fusion may improve our building detection performance. Fortunately, the proposed probabilistic building detection method allows fusion of information. Therefore, in this section, we introduce two fusion methods using our probabilistic framework to improve our building detection results.

Our first method is based on data fusion. This method is straightforward, such that we use all the local feature vectors extracted with different methods as one unique group. In other words, we estimate the pdf using (18) with the local feature set $\vec{k}_F = \{\vec{k}_h, \vec{k}_g, \vec{k}_f, \vec{k}_s\}$. We detect building locations from the estimated pdf with the same method in the previous section.

Our second method is based on decision fusion. Here, we mix the estimated pdfs by different methods and obtain a final pdf. While mixing the estimated pdfs, we assign a weight to each that is directly proportional to their maximum mode value. As we mentioned in the previous section, in detecting building locations from the estimated pdf, we label the mode with the maximum value as a building. By normalizing four different pdfs this way, we can mix them and obtain the final pdf estimate as

$$p_D(x, y) = \frac{1}{R} \sum_{l=\{h,g,f,s\}} \frac{p_l(x, y)}{\max_{(x,y)} p_l(x, y)} \qquad (19)$$

where $p_h(x, y)$, $p_g(x, y)$, $p_f(x, y)$, and $p_s(x, y)$ are the estimated pdfs from $\vec{k}_h$, $\vec{k}_g$, $\vec{k}_f$, and $\vec{k}_s$, respectively. R is, again, the normalizing constant. We call this method as decision fusion, since we apply the fusion operation close to the building detection step. Again, we use the building detection method in the previous section on $p_D(x, y)$ to detect building locations.

IV. Experiments

In this section, we test our novel building detection methods. Our data set consists of panchromatic Ikonos satellite and aerial image sets with 1- and 0.6-m spatial resolutions, respectively. These two image sets are naturally acquired by different sensors. Moreover, both aerial and Ikonos satellite test images are specifically selected to represent wide and diverse building characteristics. Therefore, they can provide reliable information on the performance of our methods.

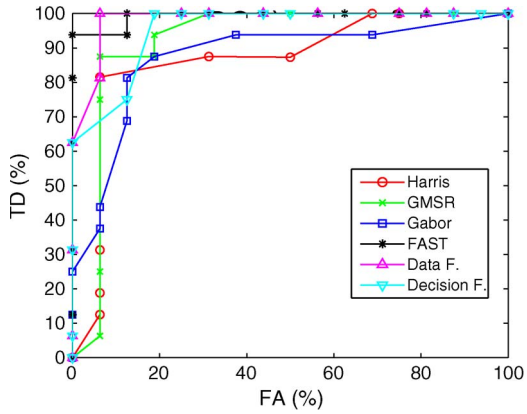

Fig. 5. ROC curves w.r.t. the downsampling ratio on the $Aerial_6$ test image. In these ROC curves, the highest TD and FA percentages occur on the rightmost corner. Each dot represents a downsampling ratio of 0.05 steps apart.

We first check the sensitivity of our building detection methods on parameter values. This way, we also experimentally justify the previously mentioned parameter selection criteria. Based on these results, we provide the building detection results on both aerial and Ikonos satellite images in a quantitative manner in terms of true detection (TD) and false alarm (FA). TD summarizes the total number of buildings correctly detected by our method. FA summarizes the corresponding total number of FAs. We also provide the percentages of TD and FA. In forming these percentages, we pick the total number of buildings (labeled by a human expert) as a benchmark. We report the performances of all four local feature vector extraction methods and data and decision fusion methods separately. In reporting these results, we follow the same strategy as that in our previous study [32]. There, if a part of a building is detected, then we assumed it to be detected correctly. If a building is detected multiple times (specifically for large buildings), then we also assumed it to be detected correctly. Different from our previous study, if a building is in construction, then we do not expect our method to detect it here. We also provide sample building detection results using decision fusion on sample satellite and aerial images.

The most important advantage of our novel building detection method is its computation time. Therefore, we examine the computation time of each building detection module in a separate section. We then compare our local feature extraction methods with the ones in the literature. We finally compare our novel building detection method in terms of building detection performances and computation times with our previous SIFT- and graph-theory-based building detection method [32].

A. Sensitivity to System Parameters

In this section, we test the sensitivity of our probabilistic building detection method to the parameters used. This way, we also experimentally justify why we picked some parameters. In the experiments, we first consider aerial images. We downsample these before detecting buildings in them. To obtain a reliable downsampling ratio, we provide the receiver-operating characteristic (ROC) curves for all our local feature extraction methods on the $Aerial_6$ image in Fig. 5. In this figure, we change the resolution of the $Aerial_6$ test image by steps of 0.05.

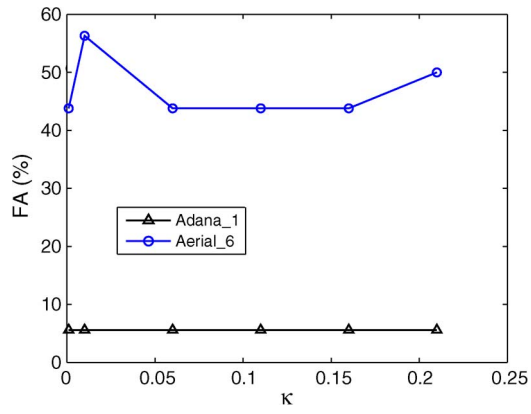

Fig. 6. Effect of κ on FA for the $Adana_1$ and $Aerial_6$ test images.

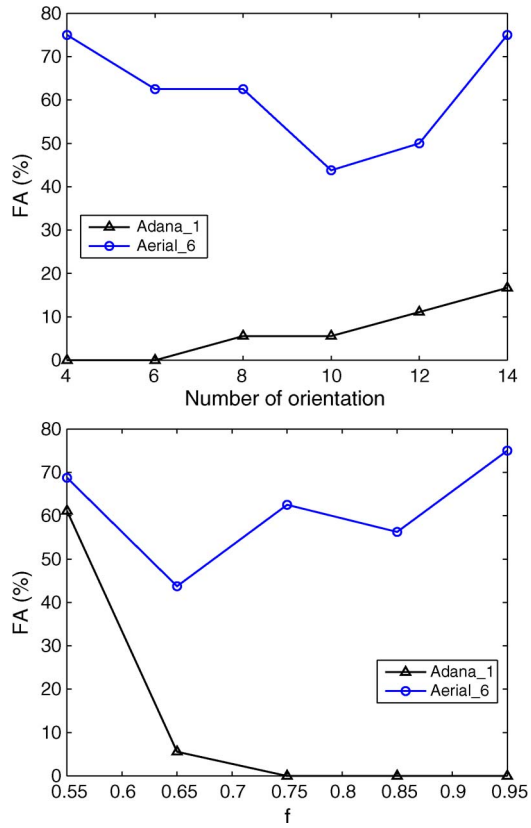

Fig. 7. Effect of φ and f on FA for the $Adana_1$ and $Aerial_6$ test images.

For each resolution, we provide the corresponding TD and FA percentages. This test shows that having a downsampling ratio of 0.5 is reasonable for our aerial images.

For the Harris-corner-based local feature extraction method, we changed the κ parameter from 0.01 to 0.21 and observed that TD did not change for both $Adana_1$ and $Aerial_6$ test images. We obtained the FA for both images as in Fig. 6. These results indicate that picking $\kappa = 0.06$ is reasonable. For the Harris-corner-based method, we also changed the averaging window width parameter from 3 to 13 and could not observe any changes. Therefore, we picked this value as seven.

For the Gabor-filtering-based local feature extraction method, we changed the total φ orientation number from 4 to 14. On both $Adana_1$ and $Aerial_6$ test images, TD rates did not change significantly. The FA rates changed as in Fig. 7.

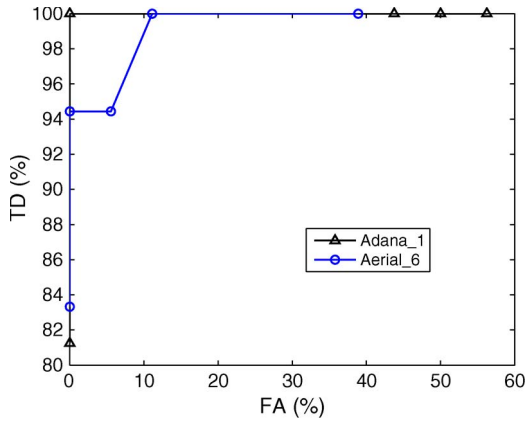

Fig. 8. ROC curves w.r.t. the building detection threshold value for the $Adana_1$ and $Aerial_6$ test images. In these ROC curves, the highest TD and FA percentages occur on the rightmost corner. Each dot represents a threshold value of 0.1 steps apart.

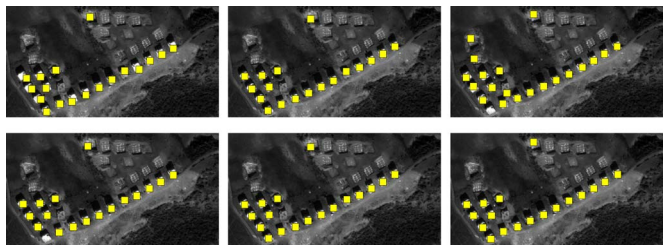

Fig. 9. Buildings detected by the proposed methods from the $Adana_1$ test image. (First row) Harris and GMSR. (Second row) Gabor and FAST. (Third row) Data fusion and decision fusion.

Based on these results, we can conclude that picking the total orientation number as ten is suitable. On the same figure, we also provide the effect of frequency on FA rate. This figure also indicates that picking $f = 0.65$ is suitable.

We finally provide the ROC curve for threshold selection for building detection decision making. As we mentioned in Section III, we picked this threshold value as 0.4 times the maximum vote value. We change this multiplier from 0.2 to 0.6 in steps of 0.1. We provide the ROC curves for the $Adana_1$ and $Aerial_6$ images in Fig. 8. Considering the TD and FA ratios on both images, choosing the multiplier value as 0.4 is reasonable.

B. Building Detection in Satellite Images

As we experimentally justified parameters in the previous section, we focus on Ikonos satellite images. We first provide the detected buildings in the $Adana_1$ test image in Fig. 9 by the four local feature vector extraction and two fusion methods mentioned previously. As can be seen, for the four local feature extraction methods, almost all of the buildings are reliably detected. There is a missing building and one FA for the Harris-corner- and GMSR-based local feature vector extraction methods. There are two missing buildings and one FA for the FAST-based local feature vector extraction method. The FA rate increases to three with the Gabor-filtering-based local feature vector extraction method. For both data and decision fusion methods, we have similar detection results with the Harris- and GMSR-based local feature extraction methods.

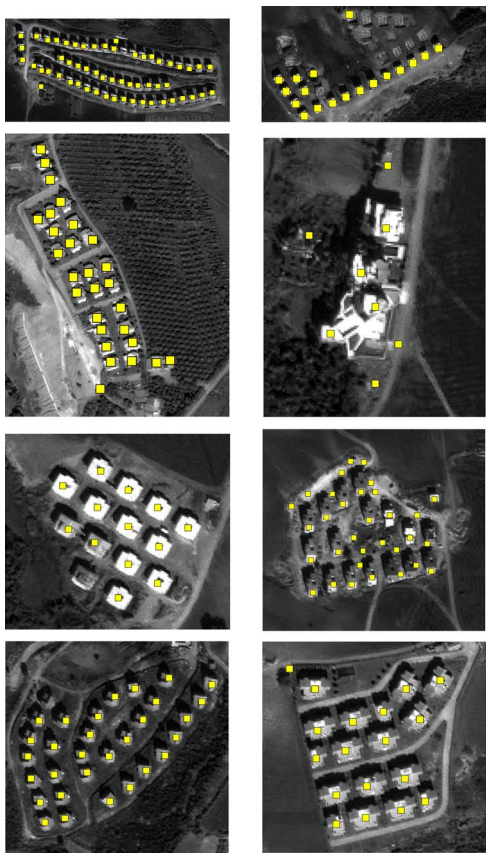

Fig. 10. Building detection results on the large Ikonos image.

TABLE I
BUILDING DETECTION PERFORMANCES FOR 32 IKONOS SATELLITE TEST IMAGES HAVING A TOTAL OF 911 BUILDINGS

Method	**TD**	**FA**	**TD** (%)	**FA** (%)
Harris	699	186	76.7	20.4
GMSR	792	215	86.9	23.6
Gabor filtering	780	187	85.6	20.5
FAST	818	181	89.8	19.9
Data fusion	831	202	91.2	22.2
Decision fusion	**851**	**163**	**93.4**	**17.9**

We then pick the test image in Fig. 1. In this large Ikonos image, our urban area detection method labeled eight different regions as urban. We applied our probabilistic building detection method with decision fusion on these regions and provide the results in Fig. 10. As can be seen, most of the buildings in these regions are correctly detected.

Next, we test our building detection methods on 32 Ikonos satellite images. From these images, 23 of them are acquired over Adana, five of them are acquired over Ankara, and four of them are acquired over Istanbul. These images cover fairly diverse geographic locations. The total number of buildings in these images is 911. To observe the performance of each local feature extraction and fusion method, we provide their building detection performances over all test images in Table I.

As can be seen in Table I, the Harris-corner-based local feature vector extraction method has the lowest detection performance. The Gabor-filtering-based local feature extraction method has the second lowest detection performance. The GMSR- and FAST-based local feature vector extraction methods have similar detection performances. Both performances are also far better than that of the Harris-corner-based method. In both data and decision fusion methods, the performance increases significantly. Using decision fusion, the correct building detection performance reaches 93.4% with 17.9% FA rate. This result is remarkable on such a diverse satellite image set.

Fig. 11. Building detection results by decision fusion in the $Adana_{18}$ and $Adana_{23}$ satellite test images.

Next, we provide sample Ikonos satellite images and building detection results on them. To show the performance of our probabilistic building detection method, we specifically selected images with complex and dense building layouts, as shown in Fig. 11.

As can be seen in Fig. 11, in both $Adana_{18}$ and $Adana_{23}$ images, the buildings are dense. Moreover, the buildings in the $Adana_{23}$ image are tall, and the image is acquired from an oblique angle. The proposed method correctly detects 66 of the 71 buildings (corresponding to TD $= 94.3\%$) in the $Adana_{18}$ image. Therefore, in this test image, almost all buildings are correctly detected. However, there are 21 FAs in this image (corresponding to FA $= 30.0\%$). Most FAs originate from man-made structures resembling buildings (like the swimming pool and road segments) in this image. For the $Adana_{23}$ image, the proposed method is able to detect 46 of the 48 buildings (corresponding to TD $= 95.8\%$). There are ten FAs (corresponding to FA $= 20.8\%$) in this image. For such a complex satellite image, this performance is fairly good.

C. Building Detection in Aerial Images

We also test our probabilistic building detection method on 22 aerial images. All these images are acquired over Istanbul and nearby villages. The total number of buildings in these images is 697. Aerial images have higher resolution. Therefore, they contain more details compared to Ikonos satellite images. To eliminate the effects of these undesired details, we first apply nonlinear bilateral filtering to aerial images [8]. Then, we downsample them by 0.5. We experimentally justified this value in the previous section. Similar to the satellite images, we provide the building detection results of our methods on aerial images in Table II.

As can be seen in Table II, for all methods, the FA rate is higher compared to the satellite image detection results. One possible reason for this poor performance is that, in aerial images, road segments and nearby land formations resemble

TABLE II
BUILDING DETECTION PERFORMANCES FOR AERIAL TEST IMAGES HAVING A TOTAL OF 697 BUILDINGS

Method	**TD**	**FA**	**TD** (%)	**FA** (%)
Harris	505	279	72.5	40.0
GMSR	569	426	81.6	61.1
Gabor filtering	591	411	84.8	59.0
FAST	532	218	76.3	31.3
Data fusion	565	297	81.1	42.6
Decision fusion	**575**	**271**	**82.5**	**38.9**

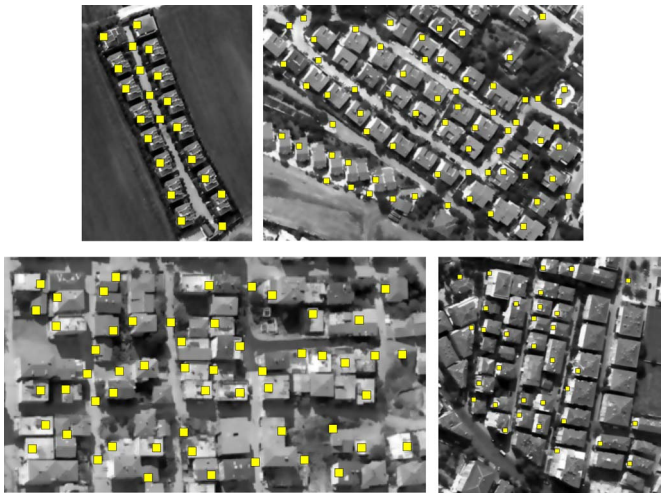

Fig. 12. Building detection results by decision fusion in the $Aerial_6$, $Aerial_{10}$, $Aerial_{12}$, and $Aerial_{16}$ test images.

buildings due to the resolution of these images. Among different local feature vector extraction methods, Harris has the lowest while Gabor filtering has the highest building detection performance. Moreover, decision fusion has a clear advantage (taking both TD and FA into account) compared to all methods in detecting buildings.

As in satellite images, we provide sample building detection results in Fig. 12 for four aerial images. These images also show fairly complex and dense building layouts. For the $Aerial_6$ test image, our method is able to detect all of the buildings with five FAs (corresponding to TD $= 100.0\%$ with FA $= 31.3\%$). For the $Aerial_{10}$ test image, our method is able to detect 44 of the 51 buildings with 20 FAs (corresponding to TD $= 86.3\%$ with FA $= 39.2\%$). Although we have a fairly high FA rate for this image, as can be seen, most FAs are close to the buildings. Therefore, they are not random. On the other hand, for such a complex scene, we have a fairly good building detection performance. For the $Aerial_{12}$ test image, our proposed method is able to detect 56 of the 73 buildings with six FAs (corresponding to TD $= 76.7\%$ with FA $= 8.2\%$). For this test image, the buildings are dense and not easily visible. We have a reasonable building detection performance. The corresponding FA rate is also lower for this image. Finally, for the $Aerial_{16}$ test image, the proposed method is able to detect 33 of the 56 buildings with only one FA (corresponding to TD $= 57.9\%$ with FA $= 1.8\%$). As in the $Adana_{23}$ satellite test image, in the $Aerial_{16}$ test image, the buildings are tall and dense. Also, the image acquisition angle is oblique. Therefore, the performance of our building detection method on this aerial image is fairly low.

TABLE III
CPU TIMES (IN SECONDS) FOR BUILDING DETECTION OPERATIONS ON THE $Adana_1$ TEST IMAGE

	Local Feature Vector			
Module	Harris	GMSR	Gabor	FAST
Preprocessing	0.08	0.09	0.08	0.08
Local feature vectors	0.20	0.80	0.33	3.21
Kernel density estimation	0.02	0.73	0.80	0.50
Building detection	0.33	0.30	0.34	0.30
Total	0.63	1.92	1.55	4.09

D. Computation Times

The most important advantage of our method is its computation time. Therefore, in this section, we consider the time needed by all building detection modules in detail. To note here, timing directly depends on the test image. As the number of buildings in a test image increases, the number of local feature vectors will also increase. Our building detection method will need more computation time. To give an idea for the possible reader, we consider the $Adana_1$ test image as a benchmark. We tabulate all CPU timings for each module in Table III. In reporting these results, we used a PC with Intel Core 2 Quad processor with 3-GHz clock speed and has 8 GB of RAM. We used Matlab as our coding platform.

As can be seen in Table III, all modules need similar CPU times. To note here, the preprocessing module summarizes image read and variable assignment operations. The total time needed to detect buildings (using any local feature vector extraction method) is fairly short. However, depending on the local feature vector extraction step, it changes slightly. If the user needs a faster building detection system, then these slight changes should be taken into account. If we consider both data and decision fusion methods, we need 7.04 and 6.46 s, respectively. Although these methods need more computation times, their building detection performances are also better.

E. Comparison With Other Keypoint and Corner Extraction Methods

Other than the local feature extraction methods used in this paper, there are several other methods in the literature [24]. In this section, we pick four of these methods and apply them to our problem. The first method we pick is Lindeberg's [20] blob detector. The second method is Lowe's [21] SIFT method. The third method is speeded up robust features (SURF) by Bay *et al.* [3]. The fourth method is chord-to-point distance accumulation technique (CPDA) by Awrangjeb and Lu [2]. We provide the coordinates of the local features extracted by these methods and the buildings detected by them using our probabilistic method on the $Adana_1$ test image in Fig. 13.

As can be seen in Fig. 13, for all the methods except CPDA, we have many FAs in building detection. For the CPDA, we miss closely located buildings (in other test images) since this method depends on edge detection as a preprocessing step. For all the methods in this section, we had to upsample the $Adana_1$ Ikonos satellite test image by six to reliably extract the keypoints. In a previous study, we had the same observation [32]. Therefore, the time needed to detect building locations increased. Based on these observations, we did not use them in our probabilistic building detection framework.

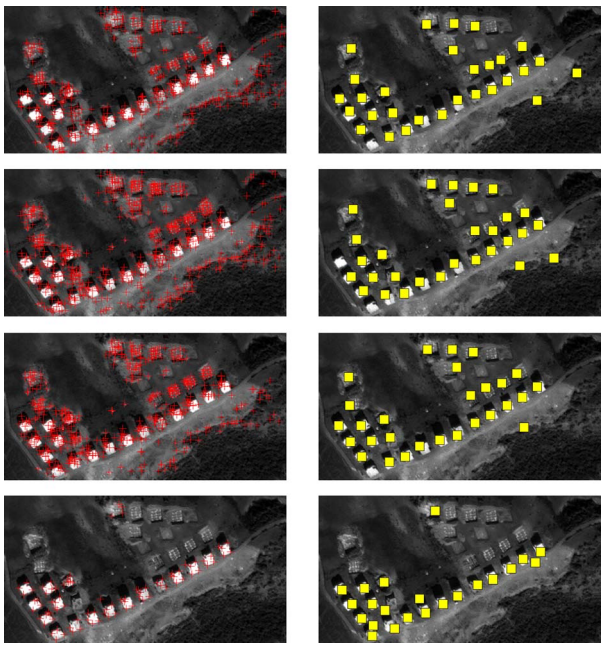

Fig. 13. Comparison of other keypoint extraction methods. (First column) Local features extracted by the blob detector, SIFT, SURF, and CPDA on the $Adana_1$ test image. (Second column) Building detection results by these features.

F. Comparison With a Previous Building Detection Method

Finally, we compare our probabilistic building detection method with our previous SIFT- and graph-theory-based method [32]. There, we used two template (representing dark and bright) building images. Using SIFT keypoints and graph matching, we were able to detect buildings and urban area in a given satellite image. However, in this paper, we do not use any template building images. Therefore, our present method is applicable to both satellite and aerial images. In our previous method, we obtained an 88.4% building detection performance with 14.4% FA rate. On the same data set, we obtained a 93.1% building detection performance with 17.7% FA rate with our new method using decision fusion. If we only use FAST-based local feature vectors, then we have an 89.5% building detection performance with 20.0% FA rate. These results indicate the correct building detection performance and FA rate of our present method is slightly higher.

In terms of CPU timing, our previous method needs 160.76 s to detect buildings from the $Adana_8$ test image (with the PC mentioned in the previous section). On the other hand, our novel building detection method needs at most 4.04 s (if FAST-based local feature vectors are used) and 6.94 s (if decision fusion is used). Therefore, the present method is extremely fast compared to our previous SIFT- and graph-theory-based method.

V. Final Remarks

In this paper, we have introduced a novel building detection method based on a probabilistic framework. To do so, we defined the spatial coordinates of buildings (to be detected) as joint random variables. We formed their pdf by the nonparametric variable-kernel density estimation method. In estimating the pdf, we used local feature vectors (extracted from the image) as observations. At this step, we used four local feature vector extraction methods. To note here, our probabilistic building detection framework is not limited to these four methods. It can be applied to other local feature extraction methods as well. Then, we detected building locations using the modes of the estimated pdf and other probabilistic constraints. We further improved our building detection method by introducing two fusion methods, i.e., one in data and the other in decision level. To obtain the performance of our building detection method, we tested it on panchromatic Ikonos satellite and aerial images. For these images, the sensor characteristics and the spatial resolution differ. After extensive testings, we observed that our method is able to detect most of the buildings (having different size, shape, and intensity values) in a correct manner on both aerial and Ikonos satellite images. However, for aerial images, the FA rate is relatively high. Therefore, a more complex system is needed to decrease the FA rate for these images. On the other hand, the time needed to detect buildings for our method is fairly short. In fact, this is one of the major advantages of the proposed method. To justify our claim, we compared our building detection method with that of our previous study using SIFT keypoints and graph theory. Although both methods have similar building detection performances, the present method is extremely fast compared to our previous method. In addition, we do not need training sets (like template images) in this paper. We can conclude that our probabilistic building detection method can be used to detect buildings in a fast and reliable manner.

Acknowledgment

The authors would like thank the anonymous reviewers for their valuable comments and corrections.

References

[1] H. Akçay and S. Aksoy, "Automatic detection of geospatial objects using multiple hierarchical segmentations," *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 7, pp. 2097–2111, Jul. 2008.

[2] M. Awrangjeb and G. Lu, "Robust image corner detection based on the chord-to-point distance accumulation technique," *IEEE Trans. Multimedia*, vol. 10, no. 6, pp. 1059–1072, Oct. 2008.

[3] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," *Comput. Vis. Image Understand.*, vol. 110, no. 3, pp. 346–359, Jun. 2008.

[4] Y. Bazi and F. Melgani, "Gaussian process approach to remote sensing image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 1, pp. 186–197, Aug. 2010.

[5] C. Benedek, X. Descombes, and J. Zerubia, "Building extraction and change detection in multitemporal aerial and satellite images in a joint stochastic approach," INRIA, Paris, France, Tech. Rep. 7143, Dec. 2009.

[6] 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.

[7] 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.

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

[9] 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.

[10] 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.

[11] 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.
[12] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proc. 4th Alvey Vis. Conf.*, 1988, pp. 147–151.
[13] M. Idrissa, V. Lacroix, A. Hincq, H. Bruynseels, and O. Swartenbroekx, "SPOT5 images for urbanization detection," in *Proc. Adv. Concepts Intell. Vis. Syst.*, 2004.
[14] A. K. Jain, N. K. Ratha, and S. Lakshmanan, "Object detection using Gabor filters," *Pattern Recognit.*, vol. 30, no. 2, pp. 295–309, Feb. 1997.
[15] K. Karantzalos and N. Paragios, "Recognition-driven two-dimensional competing priors toward automatic and accurate building detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 1, pp. 133–144, Jan. 2009.
[16] 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.
[17] T. Kim and J. P. Muller, "Development of graph-based approach for building detection," *Image Vis. Comput.*, vol. 17, no. 1, pp. 3–17, Jan. 1999.
[18] S. Krishnamarchi and R. Chellappa, "Delineating buildings by grouping lines with MRF," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 5, no. 1, pp. 164–168, Jan. 1996.
[19] V. Kyrki, J. Kamarainen, and H. Kalviainen, "Simple Gabor feature space for invariant object recognition," *Pattern Recognit. Lett.*, vol. 25, no. 3, pp. 311–318, Feb. 2004.
[20] T. Lindeberg, "Feature detection with automatic scale selection," *Int. J. Comput. Vis.*, vol. 30, no. 2, pp. 79–116, Nov. 1998.
[21] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004.
[22] 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.
[23] H. Mayer, "Automatic object extraction from aerial imagery—A survey focusing on buildings," *Comput. Vis. Image Understand.*, vol. 74, no. 2, pp. 138–149, May 1999.
[24] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 10, pp. 1615–1630, Oct. 2005.
[25] 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.
[26] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-9, no. 1, pp. 62–66, Jan. 1979.
[27] E. Rosten, R. Porter, and T. Drummond, "Faster and better: A machine learning approach to corner detection," *IEEE Trans. Pattern Anal. Mach. Learn.*, vol. 32, no. 1, pp. 105–119, Nov. 2010.
[28] C. Schmid, R. Mohr, and C. Bauckhage, "Evaluation of interest point detectors," *Int. J. Comput. Vis.*, vol. 37, no. 2, pp. 151–172, Jun. 2000.
[29] 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.
[30] B. W. Silverman, *Density Estimation for Statistics and Data Analysis*., 1st ed. London, U.K.: Chapman & Hall, 1986.
[31] B. Sırmaçek and C. Ünsalan, "Building detection using local Gabor features in very high resolution satellite images," in *Proc. RAST*, Istanbul, Turkey, 2009, pp. 283–286.
[32] B. Sırmaçek and C. Ünsalan, "Urban area and building detection using SIFT keypoints and graph theory," *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 4, pp. 1156–1167, Apr. 2009.
[33] M. Sonka, V. Hlavac, and R. Boyle, *Image Processing: Analysis and Machine Vision*, 3rd ed. Lubbock, TX: CL-Engineering, 2007.
[34] D. Tuia, F. Pacifici, M. Kanevski, and W. J. Emery, "Classification of very high spatial resolution imagery using mathematical morphology and support vector machines," *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 11, pp. 3866–3879, Nov. 2009.
[35] C. Ünsalan, "Gradient-magnitude-based support regions in structural land use classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 3, no. 4, pp. 546–550, Oct. 2006.
[36] C. Ünsalan, "Urban area detection using local features and spatial voting," *IEEE Geosci. Remote Sens. Lett.*, vol. 7, no. 1, pp. 146–150, Jan. 2010.
[37] 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.
[38] C. Ünsalan and K. L. Boyer, "A system to detect houses and residential street networks in multispectral satellite images," *Comput. Vis. Image Understand.*, vol. 98, no. 3, pp. 432–461, Jun. 2005.
[39] M. Vetterli and J. Kovacevic, *Wavelets and Subband Coding*, 1st ed. Englewood Cliffs, NJ: Prentice-Hall, 1995.
[40] L. Wei and V. Prinet, "Building detection from high-resolution satellite image using probability model," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2005, pp. 3888–3891.
[41] Z. Xiong and Y. Zhang, "A novel interest-point-matching algorithm for high-resolution satellite images," *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 12, pp. 4189–4200, Dec. 2009.
[42] 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 Yıldız Technical University, Istanbul, Turkey, in 2006 and 2007, respectively, and the Ph.D. degree from the Yeditepe University, Istanbul, in 2009.

During her Ph.D. study, she was a Research and Teaching Assistant with Yeditepe University, where she was also a member of the Computer Vision Research Laboratory. She is currently a Research Fellow in the Department of Photogrammetry and Image Analysis, Remote Sensing Technology Institute (IMF), German Aerospace Center (DLR), Wessling, Germany. Her main research interests are object detection in remotely sensed images, pattern recognition, and invariants for object recognition. She is also engaged in enhancing digital elevation models automatically to obtain very high resolution 3-D city models.

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

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