Automatic Car Counting Method for Unmanned Aerial Vehicle Images

Thomas Moranduzzo, *Student Member, IEEE*, and Farid Melgani, *Senior Member, IEEE*

***Abstract*— This paper presents a solution to solve the car detection and counting problem in images acquired by means of unmanned aerial vehicles (UAVs). UAV images are characterized by a very high spatial resolution (order of few centimeters), and consequently by an extremely high level of details which calls for appropriate automatic analysis methods. The proposed method starts with a screening step of asphalted zones in order to restrict the areas where to detect cars and thus to reduce false alarms. Then, it performs a feature extraction process based on scalar invariant feature transform thanks to which a set of keypoints is identified in the considered image and opportunely described. Successively, it discriminates between keypoints assigned to cars and all the others, by means of a support vector machine classifier. The last step of our method is focused on the grouping of the keypoints belonging to the same car in order to get a "one keypoint–one car" relationship. Finally, the number of cars present in the scene is given by the number of final keypoints identified. The experimental results obtained on a real UAV scene characterized by a spatial resolution of 2 cm show that the proposed method exhibits a promising car counting accuracy.**

***Index Terms*— Car detection, feature extraction, scale invariant feature transform (SIFT), support vector machine (SVM), unmanned aerial vehicle (UAV).**

I. Introduction

AMONG fast growing remote sensing technologies, one can find unmanned aerial vehicles (UAVs) for research and investigation activities revolving around object detection problems. UAVs have been initially developed for military purposes, but thanks to their great potential they have started to be used also for civilian applications. UAVs are small aerial platforms equipped with automatic positioning and stabilization systems. These vehicles present several interesting characteristics, i.e., they are electric, ecologic, silent, safe, flexible, and customizable. Thanks to UAVs, observing and monitoring the Earth has become easier and faster because they can reach an area of interest in very short time. An UAV can be equipped with different imaging sensors depending on the desired application. Nowadays, the main areas, in which UAVs are exploited, range from environmental monitoring to land surveillance. Despite their great potential, these vehicles present some problems especially related to the control procedures and to the information acquisition. Their flight range is still quite low, in particular due to the rapid consumption of batteries. Since UAVs can flight without pilot, a correct planning of the flight path and of the trajectories is fundamental to avoid collisions with obstacles [1], [2]. In [1], an algorithm to design an offline/online path planner for UAV autonomous navigation is presented. Sharp *et al.* [2] implemented a real-time vision system to land onto a known landing target for an UAV. In addition, the knowledge of the exact position, of the velocity, and of the orientation at each moment of the flight of the UAV is of great relevance for the processing step which follows the acquisition procedure. In this context, Achtelik *et al.* [3] presented a method to control the motion of a quadcopter based on visual feedback and measurement of inertial sensors.

UAVs can acquire information from a low altitude and, therefore, with respect to standard remote acquisition systems (e.g., satellites or airborne) they allow us to collect images with very high spatial resolution. Being able to identify specific objects or a particular class of objects in an image can provide several advantages and can open the door to the development of various interesting applications. Fergus *et al.* [4] proposed a method to learn and recognize classes of objects from unlabeled and unsegmented cluttered scenes. In [5], a technique is presented based on the development of a vocabulary that performs the detection of instances of object classes in new images. Nowadays, one of the classes of objects to which the research community is paying a particular attention is the cars. The determination of the number of vehicles on roads or in parking lots represents one of the most discussed and interesting issues in the field of object detection. It opens the way to solve urban problems that could be encountered almost every day. Especially in big cities, knowing the concentration of cars in roads or in parking lots can be extremely interesting for local administrations in order to optimize the urban traffic planning and management.

In the literature, one can find several car detection techniques which exploit low-resolution images [6]–[8]. Zhao and Nevatia [6] present a system to detect passenger cars from aerial images taken along roads, where cars appear as small objects. Moon *et al.* [7] performed an end-to-end analysis of a simple model-based vehicle detection algorithm for aerial images of parking lots. In [8], an approach using 3-D model-based vehicle detection is presented. In contrast, only a few techniques which explore high-resolution images have been developed [9]–[12]. Schlosser *et al.* [9], by using an adaptive 3-D model, introduced an automatic approach to detect vehicles in monocular high-resolution aerial images. Wang [10] proposed a vehicle detection algorithm

Manuscript received May 2, 2012; revised January 24, 2013; accepted February 22, 2013. Date of publication May 7, 2013; date of current version December 17, 2013.

The authors are with the Department of Information Engineering and Computer Science, University of Trento, Trento I-38123, Italy (e-mail: moranduzzo@disi.unitn.it; melgani@disi.unitn.it).

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.2013.2253108

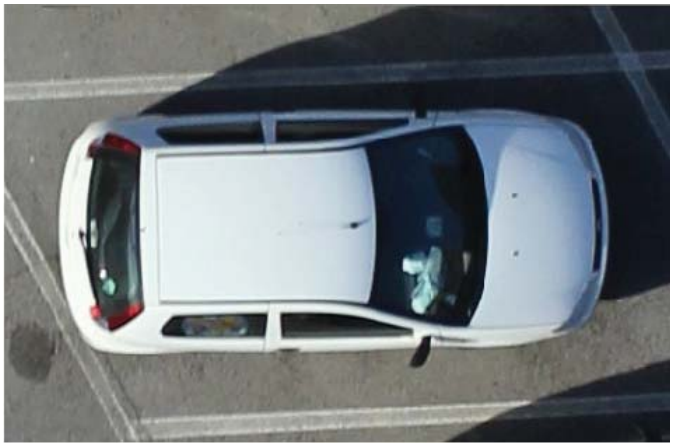

Fig. 1. Example showing the level of detail captured by a typical UAV image.

based on an improved shape matching algorithm. In [11], an object-oriented image analysis method to detect and classify road vehicles from airborne color digital orthoimagery at a ground pixel resolution of 20 cm is adopted. In [12], the problem of vehicle detection with UAV images is faced by combining a fast detection process with a classification stage.

In this paper, we propose an alternative method to detect cars for very high resolution images (2 cm) acquired by means of an UAV sensor. It begins with a screening step of asphalted zones in order to restrict the areas where detecting cars and thus to reduce the probability of false alarms. Then, we perform a feature extraction process based on scalar invariant feature transform (SIFT) thanks to which a set of consistent keypoints is identified. The algorithm then aims at the classification of these keypoints, namely at discriminating between the points which belong to cars and all the others, by means of a support vector machine (SVM) classifier. The last step of our procedure is focused on the grouping of the keypoints belonging to the same car in order to get a "one keypoint–one car" relationship. At the end of the procedure, the number of cars present inside the scene is given by the number of final keypoints identified. The main differences between our method and those available in the literature are as follows:

1) the car detection and description mechanisms;
2) our method is invariant to the car orientations;
3) it associates several pointers with the same car making the detection process more robust but requires a merging operation;
4) it allows handling occlusion problems due to shadows or trees for instance;
5) it combines the detection process with a screening operation of the asphalted areas;
6) it does not require a dictionary of precise car models.

This paper is organized as follows. Section II describes the proposed methodology in which we explain the screening step, the feature extraction and the feature classification stages, and the final merging phase. In Section III, we explain the calibration procedure and we show the experimental results. Section IV is devoted to the conclusions of this paper and to future developments.

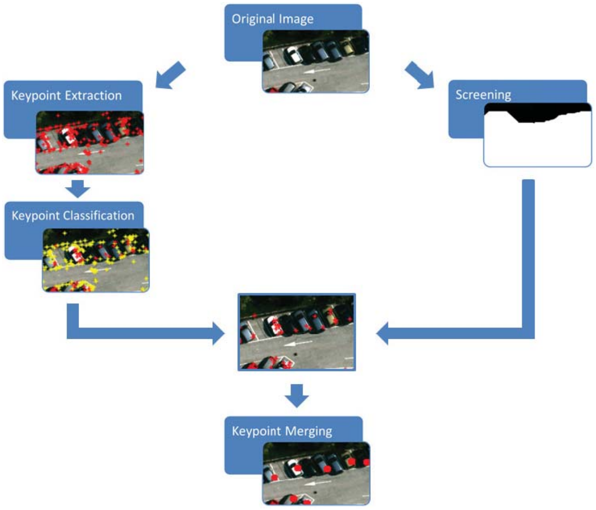

Fig. 2. Flowchart of the proposed methodology.

II. Proposed Method

A. Problem Formulation

Let us consider a very high resolution image $I(x, y)$ (where (x, y) stands for pixel coordinates in image I) acquired by means of an UAV over an urban area. UAVs typically fly at relatively low altitudes (few hundreds of meters), producing thus images with an extremely high ground resolution. The high level of details present in these kinds of images results in a large amount of potentially useful information but, at the same time, pushes us to find new processing and analysis approaches (Fig. 1). The objective of our study is to develop an automatic method for the analysis of a predetermined class of objects. In particular, we focus on the detection and counting of cars which are one of the most common objects present inside an urban area and one of the objects with particular interest.

Its underlying idea is to adopt a two-level processing scheme as described in Fig. 2. The first one works at a pixel level and is aimed at separating between asphalt and nonasphalt areas. We assume that cars lie on the sole asphalt areas. The other level consists in transforming the image into a set of keypoints found according to the SIFT transform and in detecting cars through such an image representation (Fig. 3). We believe that this representation domain is more adapted to handle the very high level of details characterizing UAV images.

B. Screening

To make the detection faster and to reduce the number of false alarms, as mentioned before, we restrict the investigated area by analyzing only regions where cars usually can be found. Assuming that cars are present only over areas covered by asphalt implies that a screening of these regions is needed. By doing so, we reduce the areas of interest to roads and parking lots.

The recognition of roads and parking lots can be envisioned in two manners. In the first one, the most accurate one, the mask to isolate the regions covered by asphalt is obtained from road maps possibly available in a geographic information system (GIS) covering the areas under analysis. In this way, no new screening is required since all asphalted areas are known *a priori*, making it easy to build the desired mask. In the second manner, screening is performed with an automatic procedure applied on the acquired image and consists of two steps: 1) a classification phase, which allows us to discriminate between asphalted and nonasphalted regions and 2) a cascade of two morphological filters applied to improve the results of the classification.

The technique used to perform the classification is based on an SVM. Let us consider a training set composed of N samples $x_i \in \Re^d$ $(i = 1, 2, \ldots, N)$ taken from the d-dimensional feature space X. With each training sample, a label $\{-1, +1\}$ is associated depending on its class (i.e., asphalt and nonasphalt). During the training phase, the SVM classifier tries to find a hyperplane in a kernel-induced feature space $(\Phi(X) \in \Re^{d'}, d' > d)$ which best separates the two groups of training samples. This hyperplane allows deriving a discriminant function $f(x)$ useful to take the membership decision

$$f(x) = w^* \Phi(x) + b^*. \tag{1}$$

The training step consists basically in the estimation of w^*, the weight vector normal, and b^*, the bias, by solving a linearly constrained quadratic optimization problem defined as

$$\min_{w,b,\varepsilon} \left(\frac{1}{2} w^T w + C \sum_{i=1}^{N} \varepsilon_i \right) \tag{2}$$

subject to the following constraints:

$$\begin{cases} y_i \left(w \cdot \Phi\left(x_i\right) + b \right) \geq 1 - \varepsilon_i, \\ \varepsilon_i \geq 0, \end{cases} \quad \text{for } i = 1, 2, \ldots, N \tag{3}$$

where the slack variable ε_{i} allows functional margin constraint violations and C is a nonnegative regularization constant, which controls the smoothness of the discriminant function in the original feature space. The final result is a discriminant function expressed as a function of the data in the original (lower) dimensional feature space X

$$f(x) = \sum_{i \in S} \alpha_i^* y_i k(x_i, x) + b^* \tag{4}$$

where $K(\cdot, \cdot)$ is a kernel function and α_i^*s are Lagrange multipliers. For more details about SVMs, we refer the reader [13]–[17].

Since pixel-based classification is typically characterized by salt-and-pepper noise and since we do not need a map of very high accuracy but just a mask where to search for cars, we will perform a refinement procedure based on the use of the mathematical morphology (MM) theory. The MM is very popular in the image processing field. Originally, it was developed for binary images but then it was adapted also to grayscale images. The main idea of MM is to investigate an image through the use of a basic element called structuring element (SE). The SE is run all over the image $I(x, y)$, and at each spatial position (x, y) the relationship between the element and the image is analyzed. An SE can assume any shape but the most common shape is the disk. The two main morphological operations, strongly related to the Minkovsky addition [18], are the dilation and the erosion.

The dilation operation on $I(x, y)$ by SE is defined by

$$I \oplus SE = \left\{ z : (SE^s)_{+z} \cap I \neq \emptyset \right\} = \bigcup_{y \in SE} I_{+y} \tag{5}$$

where $I_{+y} = \{x + y : x \in I\}$ is the translation of $I(x, y)$ along vector y and $SE^s = \{x : -x \in SE\}$ is the symmetric of SE with respect to the origin.

Similarly, the erosion operation on $I(x, y)$ by SE can be defined as

$$I \Theta SE = \{z : SE_{+z} \subseteq I\} = \bigcup_{y \in SE} I_{-y}. \tag{6}$$

In our method, we first apply an erosion operation, using a small SE, on the classification map in order to reduce the presence of noise, and then we perform a dilation operation to create homogenous asphalted regions and to recover areas such as pedestrian crossings and "holes" caused by the presence of cars on asphalt (not classified as asphalt by the classifier).

Later on in this paper, we will use again the MM theory to feed the next classification process with additional discriminating features.

C. Feature Extraction

In the previous step, we restricted the area of analysis to the regions covered by asphalt. Now, we will focus on the identification of the features which can help us in the detection of cars. Since our study is based on the recognition of a specific class of objects, we want to find features which are typical of this class. These features have to be invariant to image scale, rotation, and translation and not affected by illumination changes because we want to recognize them under all image acquisition conditions. Several object descriptors fulfilling these requirements can be found in the computer vision literature such as scale-invariant feature transform (SIFT) [19], gradient location and orientation histogram, shape context [20], spin images [21], steerable filters [22], and differential invariants [23]. Such descriptors are based on the extraction of histograms which describe the local proprieties of the points of interest. The main differences between them lie in the kind of information conveyed by the local histograms (e.g., intensity, intensity gradients, edge point locations, and orientations) and the dimension of the descriptor. An interesting comparative study was proposed in [24], where it was shown that SIFT descriptors perform better than other typical local descriptors. In this paper, we will also rely on the SIFT algorithm proposed by Lowe [19] in order to localize and characterize the keypoints in a given image.

The SIFT algorithm is widely used in the computer vision community for its effectiveness in describing high-resolution objects in complex scenes. It thus potentially fits well our scope, i.e., the detection of cars in UAV images typically characterized by extremely high spatial resolution. Moreover, the intrinsic variability of cars in the shape, the color, and the variable position conditions (rotation and scale problems) make Lowe's method a good solution candidate for our problem, which also can potentially overcome the problem of partial occlusions (i.e., cars partially hidden by trees or shadows) common in urban environments.

The algorithm starts with the extraction of SIFT keypoints which are highly distinctive. Each identified keypoint is then characterized by a feature vector which describes the area surrounding such keypoint. Since these keypoints are invariant to many transformations of the images, we think that they can potentially be appropriate features for the characterization of the vehicles in an image.

The process used to produce the SIFT features consists mainly of four steps.

The first of the four steps is devoted to the identification of possible locations which are invariant to scale changes. This objective is carried out by searching for stable points across various possible scales of a scale space properly created by convolving I with a variable scale Gaussian filter

$$L(x,y,\sigma) = I(x,y) * \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) \tag{7}$$

where "*" is the convolution operator and σ is a scale factor.

The detection of stable locations is done by identifying scale-space extrema in the difference-of-Gaussian (DoG) function convolved with the original image

$$D(x,y,\sigma) = L(x,y,\sigma) - L(x,y,k\sigma) \tag{8}$$

where k is a constant multiplicative factor which separates the new image scale from the original image. To identify which points will become possible keypoints, each pixel in the DoG is compared with the 8 neighbors at the same scale and with the other 18 neighbors of the two neighbor scales. A pixel is called keypoint if it is larger or smaller than all the other 26 neighbors. The points getting extremum in the DoG are then classified as candidate locations. The DoG function is sensitive to noise and edges; hence, a careful procedure to reject points with low contrast and poorly localized along the edges is necessary. This improvement is done considering the Taylor expansion of the scale-space function and shifting the $DoG(x,y,\sigma)$ so that the origin is at the sample point

$$D(x) = D + \frac{\partial^2 \Omega}{\partial u^2} x + \frac{1}{2} x^T \frac{\partial^2 D}{\partial x^2} x \tag{9}$$

where D and its derivatives are evaluated at the sample point and $X = (x,y,\sigma)^T$ is the offset from this point. The location of the extremum $\hat{X}$ is determined by taking the derivative of this function with respect to X and setting it to zero, giving

$$\hat{X} = -\left(\frac{\partial^2 D}{\partial X^2}\right)^{-1} \frac{\partial D}{\partial X}. \tag{10}$$

If $\hat{X} > 0.5$, then it means that the extremum lies closer to a different sample point. In this case, the interpolation is performed. If we substitute (10) into (9), we obtain a function useful to determine the points with low contrast and reject them

$$D(\widehat{x}) = D + \frac{1}{2}\frac{\partial D^T}{\partial x}\widehat{x}. \tag{11}$$

The locations with $|D(\hat{x})|$ smaller than a predefined threshold are discarded.

The DoG produces a strong response along the edges, but the locations along the edges are poorly determined and could be unstable even with small amount of noise. So, a threshold to discard the points poorly defined is essential. Usually a poorly defined peak in the DoG has a large principal curvature across the edge and a small curvature in the perpendicular direction. The principal curvatures are computed from a 2 × 2 Hessian matrix H estimated at the location and scale of the keypoint

$$H = \begin{pmatrix} D_{xx} & D_{yx} \\ D_{xy} & D_{yy} \end{pmatrix}. \tag{12}$$

The derivatives are estimated by taking differences of neighboring sample points. The eigenvalues of H are proportional to the principal curvatures of D. Let α be the eigenvalue with the largest magnitude and β be the smallest one. We can compute the sum and the product of the eigenvalues from the trace and from the determinant of H

$$\mathrm{Tr}(H) = D_{xx} + D_{yyv} = \alpha + \beta \tag{13}$$

$$\mathrm{Det}(H) = D_{xx}D_{yy} - (D_{xy})^2 = \alpha\beta. \tag{14}$$

Let r be the ratio between the largest eigenvalue and the smallest one; then

$$\frac{\mathrm{Tr}(H)^2}{\mathrm{Det}(H)} = \frac{(\alpha+\beta)^2}{\alpha\beta} = \frac{(r\beta+\beta)^2}{r\beta^2} = \frac{(r+1)^2}{r}. \tag{15}$$

To check that the ratio of principal curvatures is below some threshold, we need to check whether

$$\frac{\mathrm{Tr}(H)^2}{\mathrm{Det}(H)} < \frac{(r+1)^2}{r}. \tag{16}$$

A set of scale-invariant points is now detected, but as we stated before we need locations invariant also to the rotation point of view and this goal is reached by assigning to each point a consistent local orientation. The scale of the keypoint is used to select the Gaussian smoothed image L with the closest scale, so that all computations are performed in a scale-invariant manner. For each image sample $L(x,y)$ at this scale, the gradient magnitude $m(x,y)$ and the orientation $\theta(x,y)$ are evaluated using pixel differences

$$\begin{aligned} &m(x,y) \\ &= \sqrt{(L(x+1,y)-L(x-1,y))^2+(L(x,y+1)-L(x,y-1))^2} \end{aligned} \tag{17}$$

and

$$\theta(x,y) = \tan^{-1}\frac{(L(x,y+1)-L(x,y-1))}{(L(x+1,y)-L(x-1,y))}. \tag{18}$$

A region around a sample point is considered and an orientation histogram is created. This histogram comprises 36 bins in order to cover all the 360° of orientation (each bin holds 10°). Each sample added to the histogram is weighted

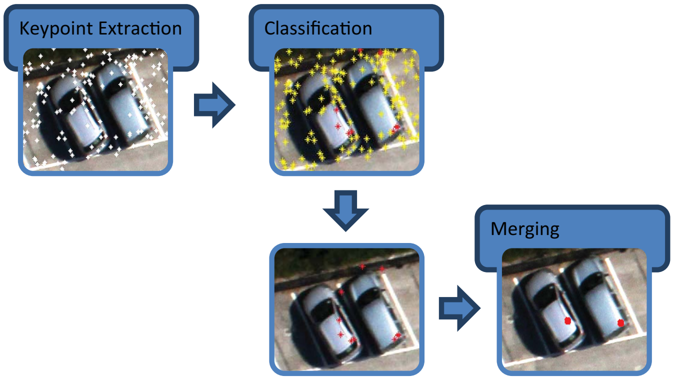

Fig. 3. Keypoints extraction, classification, and merging stages.

by its gradient magnitude and by a Gaussian-weighted circular window. The highest peak of the histogram is detected and together with the peaks within the 80% of the main peak is used to create a keypoint with that orientation.

In the last step of the method proposed by Lowe, at each keypoint, a vector is assigned which contains image gradients to give further invariance, especially with respect to the remaining variations (i.e., change in illumination and 3-D viewpoint), at the selected locations. The gradient magnitude and the orientation at each location are computed in a region around the keypoint location to create the keypoint descriptor. These computed values are weighted by a Gaussian window. They are then accumulated into orientation histograms summarizing the contents over 4 × 4 subregions, with the length of each arrow corresponding to the sum of the gradient magnitudes near that direction within the region. The descriptor is formed as a vector, which consists of values of all the orientation histogram entries.

We will adopt the common 4 × 4 array of histograms with eight orientation bins, which means that the feature descriptor will be composed of $4 \times 4 \times 8 = 128$ features. Finally, the descriptor is normalized to unit length to reduce the effects of illumination change. Any change in contrast in a pixel value multiplied by a constant will multiply gradients by the same constant, so this contrast change is canceled by vector normalization. At the end of the feature extraction procedure, we are able to associate at each keypoint two vectors: the first one contains four values, two spatial positions, orientation, and scale value, and the second vector conveys the 128 features of the detectors. We will use the first vector to determine the spatial position of the keypoints inside the image and the second one to perform the classification of such keypoints.

D. Keypoint Classification

Once the set of keypoints, with their respective descriptors, is extracted, the goal of the next stage of the process is the discrimination between keypoints which belong to cars and keypoints which represent all other objects ("background"). Since the dimension of the extracted features is relatively large, it is recommended to adopt a suitable classification method such as the SVM classifier. Before applying a classification based on an SVM classifier, we will add further information to the keypoint descriptor in order to potentially improve its discrimination power.

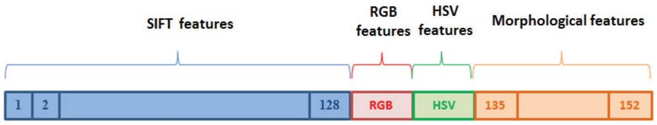

Fig. 4. Structure of the SIFT descriptor.

The first six features we will add are related to color information. Indeed, we think that the addition of some proprieties strongly associated with the object itself can lead to a better discrimination. Even if car colors can be very heterogeneous, in numerous cases, their colors appear dissimilar to the appearance of dominant objects in the contextual environment (e.g., asphalt, houses, and green areas). For this reason, we think that the use of features linked to colors spaces can help in the discrimination. In particular, we will add information related to the most common color representation system, namely the RGB system, and information from a representation system commonly used in the computer vision field, i.e., the hue, saturation, value (HSV) system. Differently from the standard RGB system, HSV has a cylindrical representation and was introduced to yield a more intuitive and perceptually relevant representation than RGB. In HSV, instead of using the three primary colors (red, green, and blue) for the representation as in RGB, a color is represented by three parameters h (hue), s (saturation), and v (value). These three parameters can be simply computed starting from the three main colors and using three basic transformations [25]. Hence, the first six features we select to add to the descriptor of each keypoint are the three primary colors and the three components of the HSV system.

Recently, in high-resolution satellite imagery, morphological operators have shown particularly effective in boosting the classification accuracy [26], [27]. We think that such operators could be even more useful in UAV images, where object geometry plays a more significant role (compared to satellite images). Consequently, in addition to the six color features, we will include also morphological features. We first apply a cascade of dilation filters and then a cascade of erosion filters on image $I(x, y)$. In both operations, at each step of the cascade, the *SE* used becomes bigger to produce two sets of images having different resolutions. These two sets give us the opportunity to characterize each keypoint at different resolutions obtaining more information about the same spatial position. The cascade of erosion and dilation filters is characterized by three steps. Accordingly, at the end of the filtering steps, the two sets of images are composed as follows:

$$S_d = \{Id_{\mathrm{SE1}}, Id_{\mathrm{SE2}}, Id_{\mathrm{SE3}}\} \tag{19}$$

$$S_e = \{Ie_{\mathrm{SE1}}, Ie_{\mathrm{SE2}}, Ie_{\mathrm{SE3}}\} \tag{20}$$

where S_d and S_e contain the dilated and eroded images, respectively, with growing sizes of SE.

Considering these two sets, 18 new features (the RGB values at each scale) are added to the descriptor of each keypoint forming final descriptors composed of 152 features (Fig. 4).

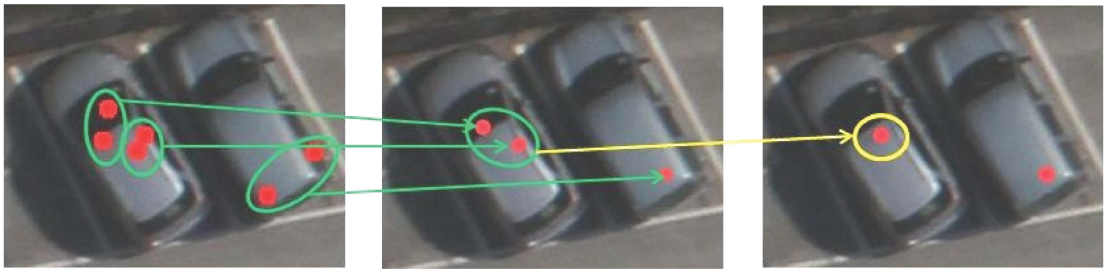

Fig. 5. Merging process.

Once the features are determined, the "car" versus "background" discrimination problem can be faced. As mentioned before, this problem will be handled by means of another SVM classifier, which will produce two sets of keypoints (i.e., car and background keypoints). Since we are interested only in one set (the set composed of car keypoints), the background keypoints are discarded.

E. Car Keypoint Merging

At the end of the keypoint classification procedure, the number of keypoints associated with the car class can be larger than the number of cars itself. The reason is that it is likely that a single car is marked by more than one keypoint. Let $K_c = \{k_1, k_2, \ldots, k_N\}$ be the set of N keypoints found for the car class in the considered image $I(x, y)$; the goal is to estimate the number of cars present in $I(x, y)$ and to identify them in a univocal manner. To pursue this scope, we will develop an algorithm to group the keypoints which belong to the same car. Since the merging is performed in the spatial image domain, it will rely on a merging criterion based on a spatial distance between the keypoints in order to identify neighboring keypoints and possibly merge them into a unique keypoint representing the car on which they lie (see Fig. 5). The main steps of our merging algorithm are summarized.

Step 1: The spatial coordinates of the keypoints contained in the set K_c are used as input of the algorithm.

Step 2: To the vector of parameters, a further parameter m is added and initialized to 1. It will act as a counter to keep trace of the number of "merging operations" done with that keypoint.

Step 3: A matrix $N \times N$ containing the Euclidean distances in the spatial domain between all keypoints is computed.

Step 4: The two keypoints (k_i, k_j) with the smallest distance $d_{\min}$ are selected.

Step 5: If $d_{\min} < T_m$ (threshold) → k_i and k_j are merged into a new point k_t which will replace the two keypoints in the set K_c.

Step 6: The matrix containing the distances is then recomputed with the new point.
Steps 3–6 are repeated until $d_{\min} > T_m$.

Step 7: Assuming that the points with a value of m smaller than 2 are isolated points only the points with $m > 1$ are kept. The number of resulting merged keypoints represents finally the estimation of the number of cars present in the scene. This step is useful to detect isolated keypoints and discard them since viewed as false alarms.

Fig. 6. UAV used for the acquisition of the images.

The value of the threshold T_m needs to be estimated by relating the expected car dimensions with the actual spatial resolution of the considered images. For instance, if the expected car width is around 180 cm and the sensor resolution is of 2 cm, the best threshold value could be empirically searched for around 90.

In order to take into account the number of times each keypoint has contributed in the merging up to a current iteration of the algorithm, a weighted merging is implemented. Let k_i and k_j be the two keypoints with the smallest distance and k_t the new point coming from the merging of the two keypoints; the new parameters of k_t will be

$$X_t = \frac{X_i \cdot m_i + X_j \cdot m_j}{m_i + m_j} \tag{21}$$

$$Y_t = \frac{Y_i \cdot m_i + Y_j \cdot m_j}{m_i + m_j} \tag{22}$$

$$\theta_t = \frac{\theta_i \cdot m_i + \theta_j \cdot m_j}{m_i + m_j} \tag{23}$$

$$s_t = \frac{s_i \cdot m_i + s_j \cdot m_j}{m_i + \ m_j} \tag{24}$$

$$m_t = m_i + m_j + 1. \tag{25}$$

III. Experimental Results

A. Dataset Description and Experimental Setup

In order to validate our methodology, we tested the whole car detection process on a real scene. We acquired such scene

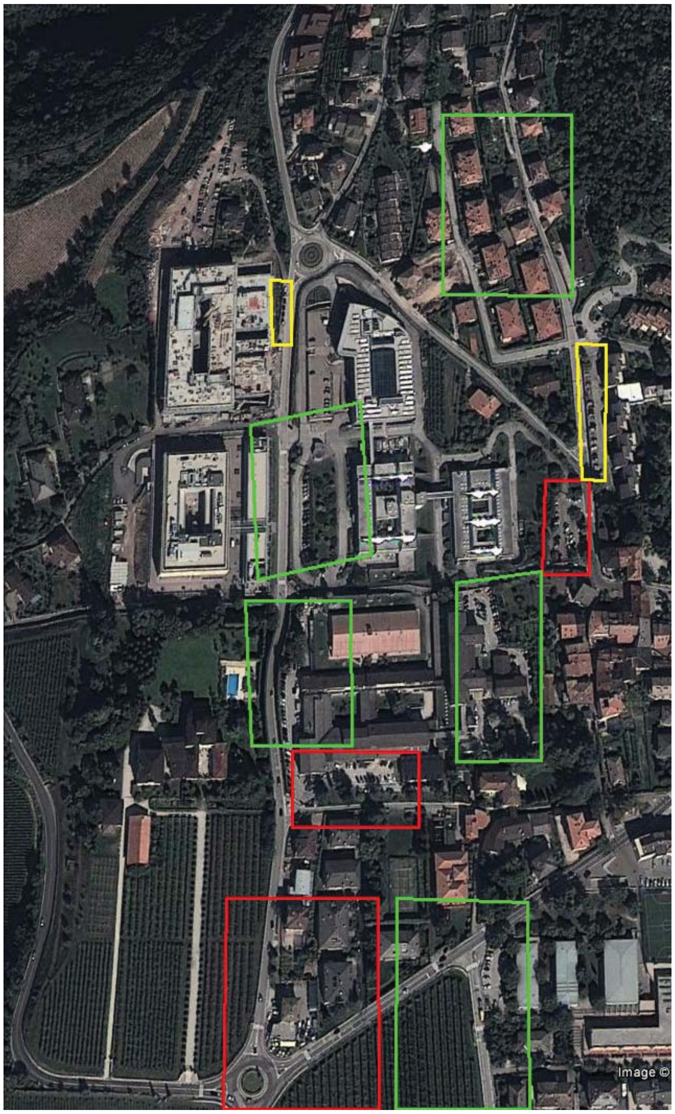

Fig. 7. View of the whole scene. Red rectangles represent the training images, rectangles in yellow are the validation areas, and green rectangles are the test images.

using an UAV equipped with imaging sensors spanning the visible range (see Fig. 6). Nadir acquisitions were performed with a picture camera Canon EOS 550D characterized by a CMOS APS-C sensor with 18 megapixels. The images were acquired over the Faculty of Science, University of Trento, Trento, Italy, on October 3, 2011, at 12:00 A.M. The images are characterized by three channels (RGB) and by a spatial resolution of 2 cm. All the acquired images have sizes of 5184 × 3456 pixels with 8 bits of radiometric resolution. From the whole scene, we selected ten images (see Fig. 7) and we divided them into three groups.

1) Training Group: It is composed of three images (i.e., red rectangles in Fig. 7). It will be used for the training of the SVM classifiers for both the screening step and the classification of the keypoints. Inside the images, it is possible to recognize 80 cars (26 in the first image, 21 in the second, and 30 in the third image). For the training of the SVM classifier devoted to the screening operations, we collected 30 807 samples which are divided into 12 463 samples of asphalt and 18 344 samples of background. This classifier is fed with three features, namely the three original color components (red, green, and blue) of the acquired images. On its side, the SVM classifier devoted to the classification of keypoints was trained with 28 632 keypoints, of which 1171 represent car keypoints and 27 461 represent the background keypoints.

2) Validation Group: It consists of two images (i.e., yellow rectangles in Fig. 7). This group will be used to calibrate the free parameters (i.e., the parameters of the SVM classifiers and of the feature extraction procedure as well as the sizes of the SE) and the threshold (i.e., the value of T_m used in the merging stage). The images belonging to this group represent two parking lots, in which the cars (6 in the first image and 17 in the second one) are easily recognizable.

3) Test Group: It is composed of five images (i.e., green rectangles in Fig. 7). This is the group on which we assessed the accuracy of the developed methodology. It includes different images in order to test the method under different conditions. We selected two images representing two big parking lots full of cars (51 and 31 cars, respectively), two images representing two standard urban areas with a medium density of cars (19 and 15 cars), and an image with only 3 cars to verify our technique in a situation where the presence of cars is rare and the presence of other objects (e.g., solar panels) could affect the automatic analysis.

We divided the experimental procedure into two main phases. In the first one, we worked only with the training and the validation groups. In this part of the work, we trained the SVMs and we calibrated the parameters. The two SVMs were trained using samples extracted from the training datasets thanks to the use of some masks expressly created in which we manually selected the cars and the asphalted areas. For the validation step, the masks were used to test how our algorithm works and to calibrate the parameters. Once the two SVM classifiers were trained, we moved to the test phase to collect the final results.

In greater detail, coming back to the training and validation steps, for the screening operations, we had to set the parameters of the SVM classifier, i.e., the regularization parameter C and the kernel parameter γ. We found, by fivefold cross-validation, that the best values for these parameters are 500 and 2, respectively. Another issue in the screening operation is the correct choice of the shape and the sizes of the SE used for the morphological filters. We observed that the disk is the best SE that could be used for both erosion and dilation operations. However, the size (the radius) of the disk used in the two operations is different. For the erosion phase, we decided to use a disk with dimension 15 (30 cm) in order to remove most of the noise. In contrast, the SE involved in dilation step has a dimension of 150 (300 cm). This specific choice was made by observing the original mask obtained before the morphological refinement. The screening aims at the identification of all the asphalted areas; thus, without the morphological improvements there could be misclassification problems especially due to cars present on the sides of the roads or to holes left on the asphalted areas by the screened cars.

For the classification procedure, additional parameters are involved: those of the second SVM classifier (C and γ)

TABLE I
SCREENING ACCURACIES IN PERCENT BEFORE AND AFTER THE APPLICATION OF THE MORPHOLOGICAL OPERATIONS

	Before Morphology		After Morphology	
	Asphalt Accuracy	**Background Accuracy**	**Asphalt Accuracy**	**Background Accuracy**
Image Test 1	33.07	90.91	74.59	84.77
Image Test 2	26.38	90.97	64.21	69.95
Image Test 3	46.61	85.64	83.76	48.86
Image Test 4	58.37	93.13	88.33	74.30
Image Test 5	62.57	95.94	99.83	85.33

TABLE II
(a) KEYPOINT CLASSIFICATION ACCURACY IN PERCENT AND
(b) CAR KEYPOINT DETECTION AND FALSE ALARMS

(a)

Features	Car	Background	Total
SIFT	27.72	98.49	**98.15**
SIFT + Color	50.07	98.61	**98.37**
SIFT + Morphology	52.75	98.67	**98.40**
SIFT + Color + Morphology	48.81	98.7	**98.34**

(b)

Features	Detection	False Alarms
SIFT	178	464
SIFT + Color	331	330
SIFT + Morphology	412	369
SIFT + Color + Morphology	495	519

TABLE III
(a) KEYPOINT CLASSIFICATION AND (b) CAR DETECTION ACCURACIES
IN PERCENT OBTAINED ON THE VALIDATION IMAGES

(a)

	Car Point Accuracy	Background Point Accuracy	Total Accuracy
Validation	42	98.03	90.96

(b)

	Producer's Accuracy	User's Accuracy	Accuracy
Validation	82.61	86.36	84.49

estimated by cross-validation, those of the feature extraction procedure (the peak threshold $\left|D(\hat{x})\right|$, and the value r of the edge threshold $(r+1)^2/r$), and the threshold T_m needed by the merging algorithm determined by maximizing the accuracy measure, *Acc,* on the validation images. For the SVM parameters, the values of C and γ are 25 and 0.25, respectively. For $\left|D(\hat{x})\right|$, the parameter which filters small peaks in the DoG scale space, the best extracted value is equal to 1. For r, the best value which eliminates peaks of the DoG scale space with a small curvature is equal to 9. To determine the parameter T_m, we did a simple assumption: generally a car has a width of about 1.80 m and it has a length of about 4.50 m. Accordingly, we tested T_m choosing the value from a range from 70 (1.40 m) to 90 (1.80 m) and we found that the best value is 80 (1.60 m).

To assess the capability of our methodology to correctly identify and count the number of cars in UAV images, we adopted a measure of accuracy based on two already existing measures [28].

The first one is the producer's accuracy. It shows the percentage of a particular class correctly classified. It is computed by dividing the number of correct samples for a class (TP = true positives = number of cars correctly identified) by the total number of samples, N (real number of cars present in the scene), of that class

$$P\text{acc} = \frac{\text{TP}}{N}. \tag{26}$$

The second one is the user's accuracy, which is a measure of the reliability of an output map generated from a classification scheme. It tells the percentage of a class that corresponds to the ground-truth class. It is calculated by dividing the number of correct samples for a class by the total number of samples assigned to that class

$$U\text{acc} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{27}$$

where FP (total number of cars incorrectly identified in the image) stands for false positives, that is the samples assigned to a class that do not have the correspondence in the ground-truth map.

The final accuracy, which we adopted to quantify our results, is the average of the two previous accuracies

$$\text{Acc} = \frac{P\text{acc} + U\text{acc}}{2}. \tag{28}$$

Such accuracy has the advantage of taking into account both the number of cars correctly classified and the number of false alarms.

B. Final Results

The obtained results are summarized in the following paragraphs. In particular, we will report the results of all the stages of the methodology starting from the validation step and concluding with the final results. For the first part, we will show the results for the feature extraction and for the classification step to allow us to evaluate each single step. For the test step, we will show only the final results of the procedure because they are the ones on which we are really interested. We will also report the final results considering different kinds of screening: without screening, with automatic screening, and with GIS-based screening. This comparison is useful to understand the importance of this step, by analyzing its impact on the other stages.

TABLE IV
ACCURACIES IN PERCENT ACHIEVED ON THE TEST IMAGES WITHOUT SCREENING

	TP	FP	N (Cars Present)	Producer's Accuracy	User's Accuracy	Total Accuracy
Image Test 1	43	32	51	84.31	57.33	**70.82**
Image Test 2	19	20	31	61.29	52.78	**57.03**
Image Test 3	16	60	19	84.21	21.05	**52.63**
Image Test 4	9	10	15	60.00	53.68	**47.37**
Image Test 5	1	21	3	33.33	4.55	**8.94**
TOTAL	88	143	119	73.95	38.10	**56.02**

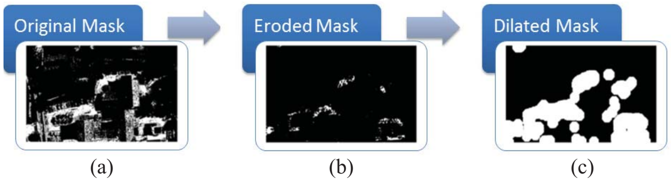

Fig. 8. Block scheme of the screening step. (a) Original mask obtained with the SVM classifier. (b) Eroded mask. (c) Dilated the final mask.

Before analyzing the final results, we will first underline the importance of the morphological operations performed on the masks obtained at the end of the screening phase and the usefulness of the addition of color and morphological features to the original SIFT descriptors. By observing the results reported in Table I, we can notice how the accuracies after the application of the morphological filters are substantially improved. The average asphalt accuracy obtained before the morphological stage is about 45.4%. It becomes 82.1% after the morphological operation. This substantial gain of accuracy compensates widely the drop of accuracy incurred for the background class (from 91.26% to 72.64%). This will be further supported by the final results discussed later.

The reason behind our decision to integrate the SIFT description vector, obtained at the end of the feature extraction procedure, takes origin from the results achieved with extra color and morphological features. Indeed, the addition of 24 features (i.e., RGB, HSV, and 3 × 6 morphological features) allows us to improve the results of the keypoint classification as can be seen in Table II. One moves from a correct detection of 178 car keypoints, by using only the 128 original SIFT features, to a right detection of 495 car keypoints by integrating the original descriptors with the aforementioned 24 features. This feature integration conveys also the advantage of making the detection process more robust since it increases the number of car keypoints found for each car, and thus renders more likely that isolated car keypoints represent false alarms.

The results of the keypoint classification are shown in greater detail in Table III (a). Generally, for a car, it is possible to identify tens of keypoints but the classifier will not associate all of them with the car class and accordingly the car keypoint classification accuracy is not very high. Nonetheless, what it is really important is that all cars are characterized by at least some correct car keypoints. Indeed, to reach our goal, we are not particularly interested in the correct identification of all the keypoints belonging to the car class. To confirm this, we can look at the results reported in Table III(b). Observing these results, we notice that we were able to detect 19 cars over the 23 cars present in the two validation images. This result confirms that a perfect discrimination of the keypoints is not mandatory to solve our problem. By tuning better the values of the parameters of the SVM classifier used for the discrimination of the keypoints, since background keypoints are dominant in the image, we can obtain better results in terms of total classification accuracy of background and car keypoints, but at the end of the whole detection process we may obtain worse results in terms of car detection. For this reason, we opted for parameter values which allow us to find the highest number of cars despite a poorer result in terms of keypoint classification accuracy. Note that single noncar keypoints can be misclassified. If they are isolated, they will be anyway considered as false alarms by our method. If they are close to car keypoints, they will be merged with car keypoints to correctly represent cars. In contrast, if misclassified noncar keypoints are spatially close to each other, after merging, they can incur in wrong car detections.

At the end of the validation step, we moved to the test stage. First, we tested our procedure on the five test images without applying the procedure of screening in order to understand how well the classification stage works. The results of these first tests are resumed in Table IV. Analyzing these results, we can notice that despite a good capability of detection (88 cars over 119) there are a large number of false alarms (143). We can note also that the merging algorithm works satisfactorily because we are able to detect 74% of cars present in the images. The main problem regards the fact that we need to reduce the number of false alarms. We notice that most of the false alarms are concentrated:

1) on the solar panels present on the roof of the buildings—due to the very high resolution of the images, keypoints found on the solar panels are associated by the classifier with the windscreen washer of cars;
2) on the fences which present a complex geometry similar to the geometry of cars.

This makes us think that a screening of the images could produce better results because we should be able to eliminate the nonasphalted regions from our analysis and consequently to reduce the number of false alarms.

As expected, performing the automatic screening of the images (Fig. 8) allows us to improve the results as reported

TABLE V
ACCURACIES IN PERCENT ACHIEVED ON THE TEST IMAGES WITH THE PROPOSED AUTOMATIC SCREENING

	TP	FP	N (Cars Present)	Producer's Accuracy	User's Accuracy	Total Accuracy
Image Test 1	40	9	51	78.43	81.63	**80.03**
Image Test 2	15	6	31	48.39	71.43	**59.91**
Image Test 3	13	27	19	68.42	32.50	**50.45**
Image Test 4	9	8	15	60.00	52.94	**56.47**
Image Test 5	1	1	3	33.33	50.00	**41.67**
TOTAL	78	51	119	65.55	60.47	**63.01**

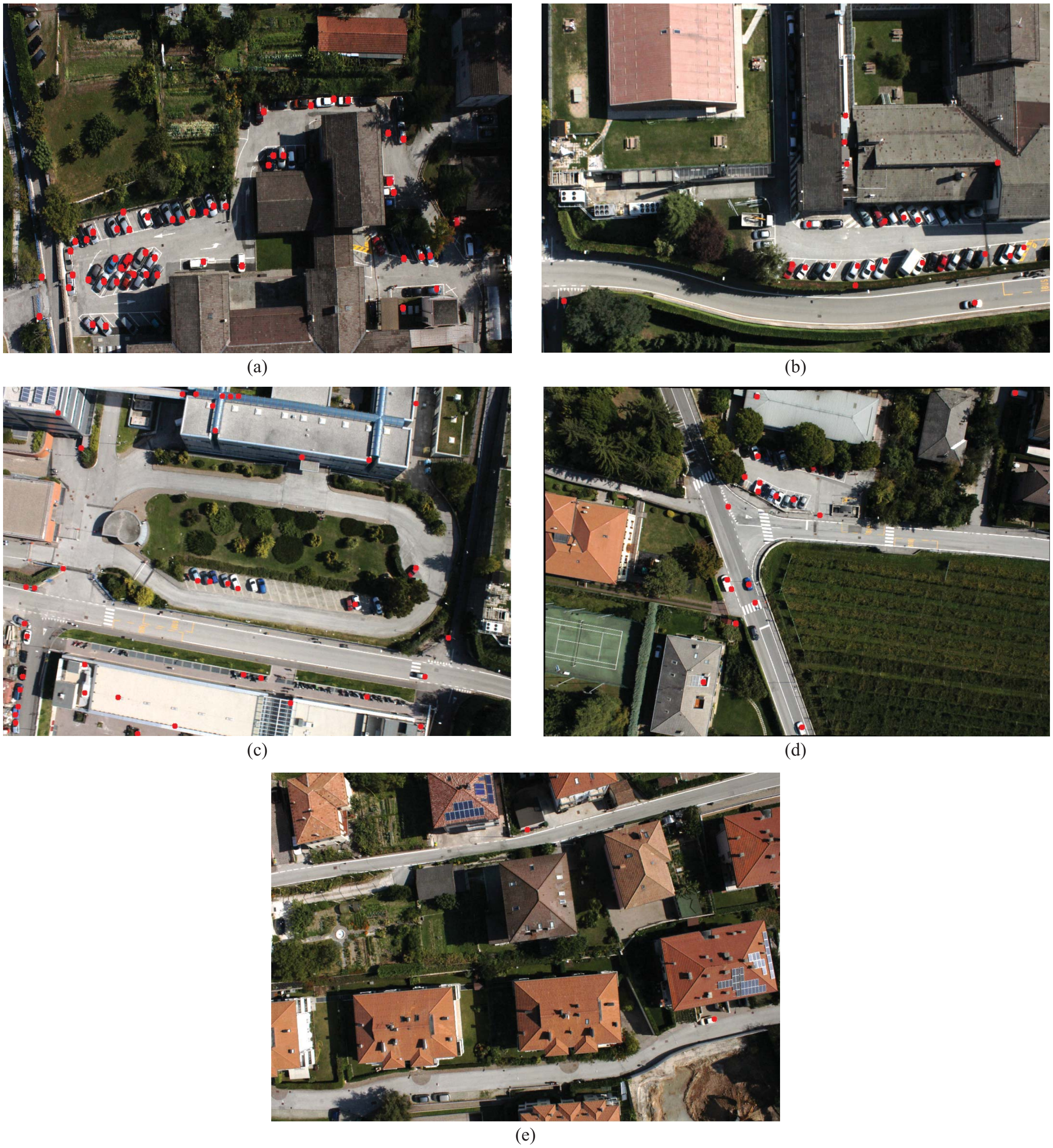

Fig. 9. Final results obtained on the test images. (a) First image. (b) Second image. (c) Third image. (d) Fourth image. (e) Fifth image.

in Table V and Fig. 9. With respect to the previous case, we "lost" ten cars but reduced the number of false alarms by one-third. We eliminated most of the keypoints localized on the roofs and on the fences; the majority of the remaining false alarms (27 over a total of 51) are in the third test image. In this particular situation, the building roofs have a color similar

TABLE VI
ACCURACIES IN PERCENT ACHIEVED ON THE TEST IMAGES WITH IDEAL SCREENING

	TP	FP	N (Cars Present)	Producer's Accuracy	User's Accuracy	Total Accuracy
Image Test 1	43	12	51	84.35	78.18	**81.25**
Image Test 2	19	3	31	61.29	86.36	**73.83**
Image Test 3	16	3	19	84.21	84.21	**84.21**
Image Test 4	9	5	15	60.00	64.29	**62.14**
Image Test 5	1	0	3	33.33	100	**66.67**
TOTAL	88	23	119	73.95	79.28	**76.61**

to the asphalt and, at first glance, they really look like parking lots. The SVM classifies these areas as asphalt since it bases the classification on the sole three color components of the image. In Table I, for the third test image, one can notice how the classifier provides poor results in terms of background classification accuracy (i.e., 48.86%) because it confuses the roofs with asphalted areas due to their very similar natural appearances. The user's accuracy obtained on this test image is poor (i.e., 32.5%) and it strongly affects the final accuracy. An accurate GIS-based masking could solve drastically these problems. Still considering these last results, we can observe that the lost cars are in particular those hidden under the shadows. These regions are not classified as asphalted regions and consequently are not considered in the search for car keypoints. It is possible to solve this problem still using a GIS-based screening.

In the last part of our analysis, we assessed the proposed method using a GIS-based screening obtained with the use of road maps. Analyzing the results reported in Table VI, we can observe a substantial improvement, as expected. The false alarms are further reduced and the number of detected cars is increased. Considering this ideal situation, the nonidentified cars are especially those present over nonasphalted areas, those strongly covered by the shadows, and those partially covered by branches of trees. The cars, which are not covered by any kind of obstacles (e.g., shadows) and have not been detected, are 16, of which 11 are completely black. The black cars are not recognized because they are assimilated by the classifier to shadows. The false alarms present in this last situation are only 23 and most of them, precisely 14 false alarms, are caused by the double recognition of the same car (two keypoints characterizing the same car). Among the 31 missed alarms, only 4 cars (about 3% of the total number of cars) have not been detected because they are too near to other cars and their keypoints have been consequently merged. This result suggests that the value adopted for the merging threshold T_m is adequate since just very few cars have been lost. Reducing further its value would solve this problem (close cars) but at the same time increase the acuity of the aforementioned double recognition issue. It is worth noting that we were able to detect 11 cars partially covered by shadows or trees. This was possible thanks to the SIFT capability for retrieving occluded objects.

To sum up, we can observe from Fig. 10 that our merging algorithm works well and that the introduction of the screening step leads to detection improvements. Without any kind of screening, we can obtain good results in terms of producer's accuracy, which are however mitigated by a low user's accuracy.

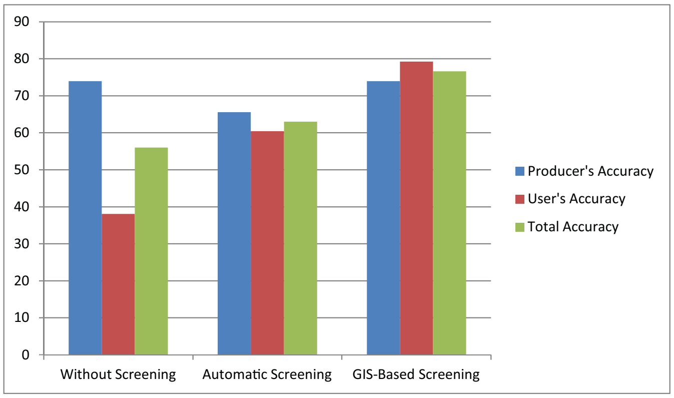

Fig. 10. Accuracies achieved with the different screening scenarios.

C. Comparison With Other Works

For the sake of comparison, we implemented the method proposed by Gleason *et al.* [12]. It is based on a two-stage process, namely detection and classification steps. In their paper, several classifiers (k-nearest neighbor, SVM, decision trees, and random tree classifiers) were compared. In our case, we will adopt the SVM classifier for uniformity with our work. Also, two feature extraction techniques (histogram of gradients and Gabor coefficients) were proposed. We opted for the histogram of gradients since it appeared the most efficient when combined with the SVM classifier. Training of this comparison method was performed with the same training images used for our method. To obtain a direct comparison with our method, we integrated the technique proposed in [12] with the automatic screening of the asphalted areas. The final results obtained on the test images are reported in Table VII. Our method (implemented with the automatic screening) correctly detects 33 cars more than what detected by the comparison method. Thanks to the use of the screening operation, the number of false alarms is limited and is similar to that produced by our method (i.e., 51 with our method versus 45). An example of detection is provided in Fig. 11. This lower performance of the method introduced in [12] is mainly due to the fact that they make use of an edge detection

TABLE VII
COMPARISON OF THE DETECTION PERFORMANCES OBTAINED BY THE PROPOSED METHOD AND THE ONE INTRODUCED IN [12]

	N (Cars Present)	Our Method		Method [12]	
		TP	FP	TP	FP
Image Test 1	51	40	9	24	5
Image Test 2	31	15	6	12	7
Image Test 3	19	13	27	2	13
Image Test 4	15	9	8	6	12
Image Test 5	3	1	1	1	8
TOTAL	119	78	51	45	45

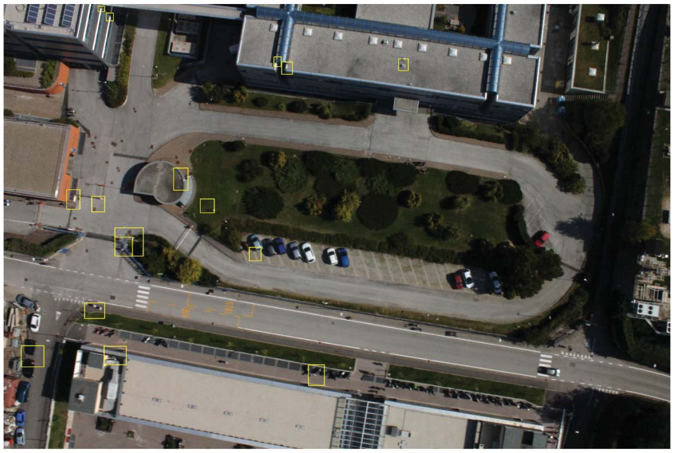

Fig. 11. Example of a final result obtained with the method proposed by Gleason *et al*. [12].

step based on the Sobel detector known to be responsive in particular in urban environments.

IV. CONCLUSION

In this paper, we developed a four-stage procedure for the automatic detection and counting of cars present in images collected by means of an UAV. Our work starts with a screening step in which through the use of a supervised classifier we detect the regions covered by asphalt assuming that usually cars in an urban scenario lie over asphalted regions (e.g., roads and parking lots). This procedure permits us to reduce the areas of investigation making the algorithm faster and with fewer false alarms. The second step is focalized on the extraction of a set of points that are invariant to affine transformations. All these points are characterized by a vector which represents some proprieties of the surrounding area around each point. In order to give more information to each keypoint, we added other spectral and morphological features in the keypoint descriptor. Next, a properly trained classifier allowed us to discriminate the points corresponding to the car class. In the last part of this paper, an algorithm was implemented to merge the car keypoints belonging to the same car. This step is necessary because, at the end of the keypoint classification, a car is typically identified by more than one keypoint.

We showed results without considering the screening step, with an automatic screening and with a GIS-based screening. This analysis was important to assess the impact of the screening step and to understand where it is possible to find potential improvements. By analyzing the obtained accuracies which are in general encouraging, we were able to conclude that the screening step is fundamental especially to reduce the number of false alarms. The effective number of detected cars is more or less constant in the three situations, so it makes us think that the merging step is correct and it works well independently from the screening step. This is confirmed by the three producer's accuracies that are all over 65% (Fig. 10). An improvement of the proposed methodology could be achieved by developing a technique able to reduce the number of multiple keypoints recognized on the same car.

Finally, as future developments, other color-based SIFT methods could be envisioned in order to better exploit the discrimination potential conveyed by the original image color space. Furthermore, we think efforts should be made in the direction of assessing other kinds of descriptors and detectors (e.g., Harris and Gabor filters, local binary patterns) as potentially alternatives to SIFT in terms of complexity and discrimination power. The car keypoint merging was performed by means of a simple solution based on spatial clustering. This step of the proposed procedure could be potentially improved by adopting more sophisticated solutions involving for instance spatiospectral clustering. Finally, after passing from a standard pixel-based to a keypoint-based analysis of the image as done in this paper, it would be potentially interesting to perform a further jump by facing the car detection and counting problem under an object-based perspective.

ACKNOWLEDGMENT

The authors would like to thank C.-C. Chang and C.-J. Lin for supplying the software LIBSVM [29] and A. Vedaldi and B. Fulkerson for making the software VLFeat [30] available for use in the context of this paper.

REFERENCES

[1] I. K. Nikolos, K. P. Valavanis, N. C. Tsourveloudis, and A. N. Kostaras, "Evolutionary algorithm based offline/online path planner for UAV navigation," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 33, no. 6, pp. 898–912, Dec. 2003.

[2] C. Sharp, O. Shakernia, and S. Sastry, "A vision system for landing an unmanned aerial vehicle," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2001, pp. 1720–1727.

[3] M. Achtelik, T. Zhang, K. Kuhnlenz, and M. Buss, "Visual tracking and control of a quadcopter using a stereo camera system and inertial sensors," in *Proc. IEEE Int. Conf. Mech. Autom.*, Aug. 2009, pp. 2863–2869.

[4] R. Fergus, P. Perona, and A. Zisserman, "Object class recognition by unsupervised scale-invariant learning," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2003, pp. 264–271.

[5] S. Agarwal, A. Awan, and D. Roth, "Learning to detect objects in images via a sparse, part-based representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 11, pp. 1475–1490, Nov. 2004.

[6] T. Zhao and R. Nevatia, "Car detection in low resolution aerial images," in *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 1. Jul. 2001, pp. 710–717.

[7] H. Moon, A. Rosenfeld, and R. Chellappa, "Performance analysis of a simple vehicle detection algorithm," *Image Vis. Comput.*, vol. 20, no. 1, pp. 1–13, 2002.

[8] Z. W. Kim and J. Malik, "Fast vehicle detection with probabilistic feature grouping and its application to vehicle tracking," in *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2. Oct. 2003, pp. 524–531.

[9] C. Schlosser, I. Reitherger, and S. Hinz, "Automatic car detection in high resolution urban scenes based on an adaptive 3D-model," in *Proc. 2nd IEEE/ISPRS Joint Workshop Remote Sens. Data Fusion Urban Areas*, May 2003, pp. 167–171.

[10] S. Wang, "Vehicle detection on aerial images by extracting corner features for rotational invariant shape matching," in *Proc. IEEE Int. Conf. Comput. Inf. Technol.*, Aug.–Sep. 2011, pp. 171–175.

[11] Q. Tan, J. Wang, and D. A. Aldred, "Road vehicle detection and classification from very-high-resolution color digital orthoimagery based on object-oriented method," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, vol. 4. Jul. 2008, pp. 475–478.

[12] J. Gleason, A. V. Nefian, X. Bouyssounousse, T. Fong, and G. Bebis, "Vehicle detection from aerial imagery," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2011, pp. 2065–2070.

[13] V. Vapnik, *Statistical Learning Theory*. New York, NY, USA: Wiley, 1998.

[14] N. Cristianini and J. S. Taylor, *An Introduction to Support Vector Machines*. Cambridge, U.K.: Cambridge Univ. Press, 2000.

[15] N. Ghoggali, F. Melgani, and Y. Bazi, "A multiobjective genetic SVM approach for classification problems with limited training samples," *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 6, pp. 1707–1718, Jun. 2009.

[16] E. Pasolli, F. Melgani, and M. Donelli, "Automatic analysis of GPR images: A pattern-recognition approach," *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 7, pp. 2206–2217, Jul. 2009.

[17] N. Ghoggali and F. Melgani, "Genetic SVM approach to semisupervised multitemporal classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 5, no. 2, pp. 212–216, Apr. 2008.

[18] P. Maragos and R. W. Schafer, "Morphological filters–part I: Their set-theoretic analysis and relations to linear shift-invariant filters," *IEEE Trans. Acoust., Speech Signal Process.*, vol. 35, no. 8, pp. 1153–1169, Aug. 1987.

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

[20] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 4, pp. 509–522, Apr. 2002.

[21] S. Lazebnik, C. Schmid, and J. Ponce, "Sparse texture representation using affine-invariant neighborhoods," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2003, pp. 319–324.

[22] W. Freeman and E. Adelson, "The design and use of steerable filters," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 9, pp. 891–906, Sep. 1991.

[23] J. Koenderink and A. van Doorn, "Representation of local geometry in the visual system," *Biol. Cybern.*, vol. 55, no. 6, pp. 367–375, 1987.

[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] W. Burger and M. Burge, *Digital Image Processing—An Algorithmic Introduction Using Java*, 1st ed. New York, NY, USA: Springer-Verlag, 2007.

[26] J. A. Benediktsson, M. Pesaresi, and K. Amason, "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.

[27] 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.

[28] N. Lavra, P. Flach, and B. Zupan, "Rule evaluation measures: A unifying view," in *Proc. 9th Int. Workshop Induct. Logic Program.*, 1999, pp. 174–185.

[29] C.-C. Chang and C.-J. Lin. (2013). *LIBSVM-A Library for Support Vector Machines* [Online]. Available: http://www.csie.ntu.edu.tw/~cjlin/libsvm

[30] A. Vedaldi and B. Fulkerson. (2008). *VLFeat Platform* [Online]. Available: http://www.vlfeat.org/index.html

Thomas Moranduzzo (S'12) received the M.Sc. degree in telecommunication engineering from the University of Trento, Trento, Italy, in 2011, where he is currently pursuing the Ph.D. degree in information and communication technologies.

He is with the Signal Processing and Recognition Laboratory, Department of Information Engineering and Computer Science, University of Trento. His current research interests include image processing and computer vision techniques applied to unmanned aerial vehicle imagery.

Farid Melgani (M'04–SM'06) received the State Engineer degree in electronics from the University of Batna, Batna, Algeria, in 1994, the M.Sc. degree in electrical engineering from the University of Baghdad, Baghdad, Iraq, in 1999, and the Ph.D. degree in electronic and computer engineering from the University of Genoa, Genoa, Italy, in 2003.

He cooperated with the Signal Processing and Telecommunications Group, Department of Biophysical and Electronic Engineering, University of Genoa, from 1999 to 2002. Since 2002, he has been an Assistant Professor and then an Associate Professor of telecommunications with the University of Trento, Trento, Italy, where he has taught pattern recognition, machine learning, radar remote-sensing systems, and digital transmission. He is the Head of the Signal Processing and Recognition Laboratory, Department of Information Engineering and Computer Science, University of Trento. His current research interests include processing, pattern recognition and machine learning techniques applied to remote sensing and biomedical signals/images (classification, regression, multitemporal analysis, and data fusion). He has co-authored more than 130 scientific publications and is a referee for numerous international journals.

Dr. Melgani has served on the scientific committees of several international conferences and is an Associate Editor of the IEEE GEOSCIENCE AND REMOTE SENSING LETTERS.