Journal of
Applied Remote Sensing

Feature analysis for detecting people from remotely sensed images

Beril Sirmacek
Peter Reinartz

SPIE

Feature analysis for detecting people from remotely sensed images

Beril Sirmacek and Peter Reinartz
German Aerospace Center (DLR), Münchner Straße 20,
82234 Weßling, Oberpfaffenhofen, Germany
b.sirmacek@tudelft.nl

Abstract. We propose a novel approach using airborne image sequences for detecting dense crowds and individuals. Although airborne images of this resolution range are not enough to see each person in detail, we can still notice a change of color and intensity components of the acquired image in the location where a person exists. Therefore, we propose a local feature detection-based probabilistic framework to detect people automatically. Extracted local features behave as observations of the probability density function (PDF) of the people locations to be estimated. Using an adaptive kernel density estimation method, we estimate the corresponding PDF. First, we use estimated PDF to detect boundaries of dense crowds. After that, using background information of dense crowds and previously extracted local features, we detect other people in noncrowd regions automatically for each image in the sequence. To test our crowd and people detection algorithm, we use airborne images taken over Munich during the Oktoberfest event, two different open-air concerts, and an outdoor festival. In addition, we apply tests on GeoEye-1 satellite images. Our experimental results indicate possible use of the algorithm in real-life mass events. [DOI: 10.1117/1.JRS.7.073594]

Keywords: airborne images; satellite images; feature extraction; probability theory; mean shift segmentation; object detection.

Paper 12103L received Apr. 11, 2012; revised manuscript received Dec. 11, 2012; accepted for publication Dec. 14, 2012; published online Jan. 22, 2013.

1 Introduction

Recently, automatic detection of people and understanding their behavior from images became a very important research field, since it can provide crucial information, especially for police departments and crisis management teams. Detecting the amount of people and understanding their moving directions and speeds can be used for detecting or predicting abnormal situations. In addition, it can also help to estimate locations where a crowd will congregate, which gives an idea about future status of underground passages, entrances of mass events, or the density of people in streets, which can also affect traffic.

Because of the importance of the topic, many researchers tried to monitor people using street or indoor cameras, which are also known as close-range cameras. However, most of the previous studies aimed to detect boundaries of large groups and extract information about them. The early studies in this field were developed from closed-circuit television images.[1–3] These cameras can monitor only a few square meters in indoor regions, and it is not possible to adapt the developed algorithms to street or airborne cameras, since the human face and body contours will not appear as clearly owing to resolution and scale differences. To be able to monitor bigger events, researchers tried to develop algorithms that can work on outdoor camera images or video streams. Arandjelovic[4] developed a local interest point extraction–based crowd detection method to classify single terrestrial images as crowd and noncrowd regions. They observed that dense crowds produce a high number of interest points. Therefore, they used density of scale-invariant feature transform features for classification. After generating crowd and noncrowd training sets, they used support vector machine (SVM)-based classification to detect crowds. They obtained

scale-invariant and good results in terrestrial images. Unfortunately, these images do not enable monitoring of large events, and different crowd samples should be detected beforehand to train the classifier. Ge and Collins[5] proposed a Bayesian marked point process to detect and count people in single images. They used football match images and street camera images for testing their algorithm. The method requires clear detection of body boundaries, which is not possible in airborne images. In another study, Ge and Collins[6] used multiple close-range images taken at the same time from different viewing angles. They used three-dimensional heights of the objects to detect people on streets. Unfortunately, it is not always possible to obtain these multiview close-range images for the street where an event occurs. Lin et al.[7] wanted to obtain quantitative measures about crowds using single images. They used Haar wavelet transform to detect head-like contours, and then using SVM they classified detected contours as head or nonhead regions. They provided quantitative measures about number of people in crowds and sizes of crowds. Although results are promising, this method requires clear detection of human head contours and training of the classifier. In any case, street cameras have only a limited coverage area to monitor large outdoor events. In addition to that, in most cases, no cameras are installed to obtain close-range street images or video streams in the place where an event occurs. Therefore, to get image data of large groups of people in very big outdoor events, the best way is to use airborne images, which began to give more information to researchers with the development of sensor technology and better data transmission possibilities. Because most of the previous approaches in this field needed clear detection of face or body features, curves, or boundaries to detect people and crowd boundaries, which is not possible in airborne images, new approaches are needed to extract information from these images. In a previous study, Hinz et al.[8] registered airborne image sequences to estimate density and motion of people in crowded regions. For this purpose, a training background segment is first selected manually to classify image as foreground and background pixels. They used the ratio of background pixels and foreground pixels in a neighborhood to plot density. Observing change of the density map in the sequence, they estimated motion of people. This approach did not provide quantitative measures about crowds. In a following study,[9] the previous approach was used to detect individuals. Positions of detected people were linked with graphs. They also used these graphs for understanding the behavior of people. To bring automated solutions to the problem in this field, Sirmacek and Reinartz[10] proposed a dense crowd detection method based on extraction of local features from airborne images. Local features are used in a probabilistic process to identify locations of dense crowds. In a following study, Sirmacek and Reinartz[11] improved the dense crowd detection study by adding a feature selection step. By using a background comparison method, they detected individuals. In Sirmacek and Reinartz,[12] by applying Kalman[13] filtering on individual detection results (which are obtained over registered airborne image sequences), they obtained automatic tracking results. Using several measures they have extracted over automatically generated probability density functions, they also estimated the main direction of motion and abnormality level of large crowds.[12] Burkert et al.[9] and Butenuth et al.[14] used their estimations to simulate the human activity in large areas. Although the proposed approaches brought new insights to the related field, owing to the diverse appearance of the input images, obtaining robust and especially automatic solutions is still a big challenge.

Herein, we present our latest techniques for detecting dense crowds and also for detecting individuals. We test robustness of the algorithm by comparing it with different parameter selection and different feature extraction methods. For testing our algorithms, we use color airborne image sequences and Geo-Eye-1 satellite images. Quantitative results and estimations of computation times are shown, since these are crucial for future real-time application of the algorithms.

2 Detecting People from Airborne Images

Airborne image data were acquired by the Deutsches Zentrum fuer Luft und Raumfahrt (DLR, German Aerospace Center) 3K-Camera-System with a Cessna aircraft from 1000 m flight altitude and ground sampling distance (GSD) of 15 cm. The camera system, including on-board processing and downlink capabilities, is described in detail in Kurz et al.[15] and consists of three

off-the-shelf Canon Mark II cameras. Image sequences are acquired with a frequency between 2 and 1 Hz during the flight. The images are georeferenced on-board, so that the absolute coordinates are correct and objects such as cars and people can be detected at their absolute position. For each airborne image in the input sequence, we apply a dense crowd detection and people detection approach. Next, we introduce steps of the approach in detail.

2.1 *Local Feature Extraction*

To illustrate the algorithm steps, we pick Stadium_1 image from our Stadium_{1-43} test image sequence. In Fig. 1(a), we represent Stadium_1 test image, and in Fig. 1(b), we represent a subpart of the original image to give information about real resolution of the image. As can be seen, airborne image resolutions do not make it possible to see each single person with sharp details. However, we can still notice a change of color and intensity components in the place where a person exists. Therefore, our dense crowd and people detection method depends on local features extracted from the intensity band of the input test image.

For local feature extraction, we use features from accelerated segment test (FAST). FAST feature extraction method was specially developed for corner detection purposes by Rosten et al.[16] The algorithm can be briefly explained as follows:

1. Select a pixel p in the image. Assume that the intensity of this pixel is I_p.
2. Set a threshold value T (suggested to be selected as 20% of I_p intensity value).
3. Consider a circle of 16 pixels surrounding the pixel p. This is called Bresenham circle of radius 3, which is described in Ref. 17 written by Hearn and Baker.
4. N contiguous pixels of the 16 need to be either above or below I_p by the value T, if the pixel needs to be detected as an interest point. (N value is suggested to be set as 12.)
5. To make the algorithm fast, first compare the intensity of pixels 1, 5, 9, and 13 of the circle with I_p. At least three of these four pixels should satisfy the threshold criterion to detect an interest point in p location.
6. If at least three of the four pixel values are not above or below the threshold value, then p is not selected as an interest point. If at least three of the pixels are above or below the threshold value, then check for all 16 pixels and check if 12 contiguous pixels fall in the criterion.
7. Repeat the procedure for all pixels in the image.

Although the algorithm has been developed for corner detection, it also gives high responses on small regions that are significantly different from surrounding pixels. Therefore it is especially suitable if a person's top view is represented by just a few pixels, which is true for the airborne images we are working with. We assume (x_i, y_i) for $i \epsilon [1, 2, \ldots, K_i]$ as FAST local features which are extracted from intensity band of the input image, respectively. Here, K_i indicates the maximum number of features. We represent locations of detected local features for

(a) (b)

Fig. 1 (a) Stadium_1 test image from our airborne image sequence including both crowded and sparse people groups. (b) Closer view of the crowded region in Stadium_1 labeled with red square.

Stadium$_1$ test image in Fig. 2(c). Extracted FAST features will behave as observations of the probability density function (PDF) of the people to be estimated after a feature selection process, which we introduce in the next step.

2.2 *Local Feature Selection*

As can be seen in Fig. 2(c), we detected FAST features at almost each individual person's position. Unfortunately, corners of other objects also led to FAST feature detection. In this step, we apply segmentation to the input image to estimate the interest region, which helps us to eliminate redundant local features.

For segmenting the input image, we benefit from the mean shift segmentation approach, proposed by Comanicu and Meer.[18] For the mean shift segmentation process, we choose spatial bandwidth (h_s) and spectral bandwidth (h_r) parameters as 7 and 6.5 pixels, respectively, after extensive tests, and we use the same parameters for all input images. The segmentation result is a new image such as $S(x, y)$, which holds each segment labeled by a single color. We present mean shift segmentation result for our Stadium$_1$ test image in Fig. 2(b). Here, each segment is labeled with the mean of red, green, and blue band values of the original image pixels inside the segment. Although we have no idea about which segment represents which object, the segmentation result can be useful to decrease the complexity of the problem. We believe that on the interest region (generally roads), there should be many local features indicating people. Therefore, we eliminate regions having <50 local features inside. Remaining regions are assumed as interest regions, which is represented with value 1 in $M(x, y)$ binary mask. In the next steps of the algorithm, we use (x_i, y_i), $i \epsilon [1, 2, \ldots, K_i]$ local features only if they satisfy $M(x_i, y_i) = 1$ equation. In the next step, we introduce an adaptive kernel density estimation method to estimate corresponding PDF, which will help us to detect dense people groups and people in sparse groups.

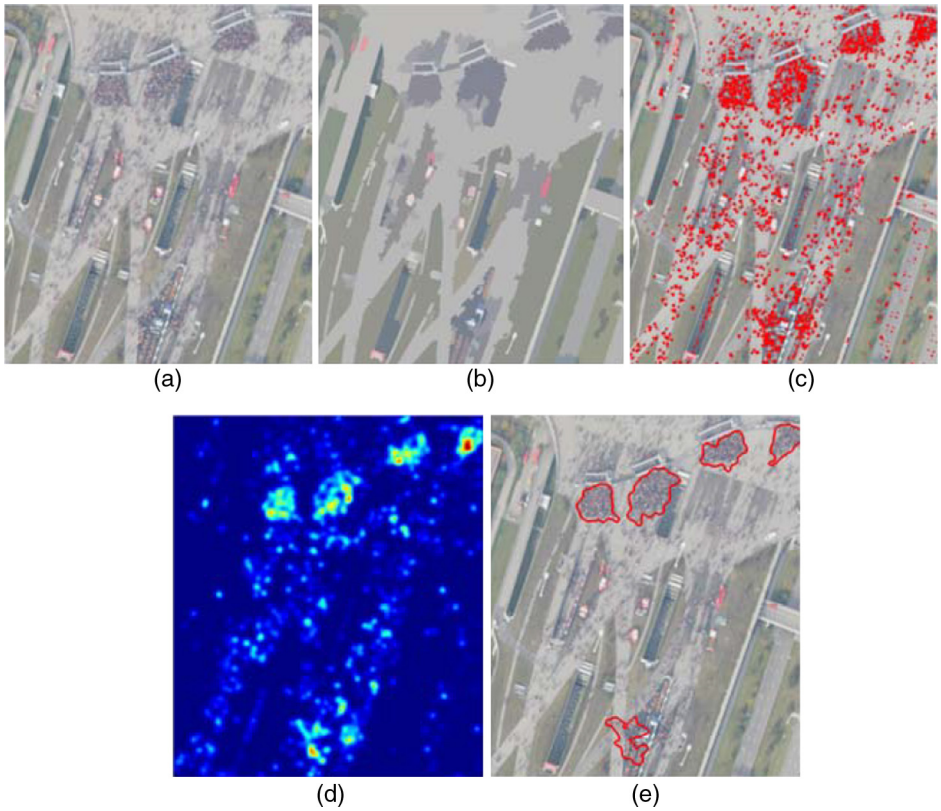

Fig. 2 (a) Original Stadium$_1$ test image. (b) Mean-shift segmentation result for Stadium$_1$ test image. (c) Detected FAST feature locations on Stadium$_1$ test image represented with red crosses. (d) Estimated PDF (color coded) for Stadium$_1$ image generated using FAST feature locations as observations. (e) Automatically detected dense crowd boundaries and detected people in sparse groups for Stadium$_1$ test image.

2.3 *Detecting Dense Crowds Based on Probability Theory*

Since we have no preinformation about the street, building, green area boundaries, and crowd locations in the image, we formulate the crowd detection method using a probabilistic framework. Assume that (x_i, y_i) is the i'th FAST feature where $i\epsilon[1, 2, \ldots, K_i]$. Each FAST feature indicates a local color change which might be a human to be detected. Therefore, we assume each FAST feature as an observation of a crowd PDF. For crowded regions, we assume that more local features should come together. Therefore, knowing the PDF will lead to detection of crowds. For PDF estimation, we benefit from a kernel-based density estimation method as Sirmacek and Unsalan[19] represented for local feature-based building detection. Silverman[20] defined the kernel density estimator for a discrete and bivariate PDF as follows. The bivariate kernel function $[N(x, y)]$ should satisfy the conditions given below:

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

and

$$N(x, y) \geq 0, \quad \forall\ (x, y) \tag{2}$$

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

$$p(x, y) = \frac{1}{nh} \sum_{i=1}^{n} N\left(\frac{x - x_i}{h}, \frac{y - y_i}{h}\right), \tag{3}$$

where h is the width of the window, which is also called smoothing parameter. In this equation, (x_i, y_i) for $i = 1, 2, \ldots, n$ are observations from PDF that we want to estimate. We take $N(x, y)$ as a Gaussian symmetric PDF, which is used in most density estimation applications. The Gaussian kernel function is easy to generate by setting only one bandwidth parameter, and the PDF, which is the sum of Gaussian kernels, gives a smooth function of probability densities with smooth transitions. Therefore, in our application the estimated PDF is formed as below:

$$p(x, y) = \frac{1}{R} \sum_{i=1}^{K_i} \frac{1}{\sqrt{2\pi\sigma}} \exp\left[-\frac{(x - x_i)^2 + (y - y_i)^2}{2\sigma}\right], \tag{4}$$

where σ is the bandwidth of Gaussian kernel (also called smoothing parameter), and R is the normalizing constant to normalize $p_n(x, y)$ values between [0, 1]. In kernel-based density estimation, the main problem is how to choose the bandwidth of Gaussian kernel for a given test image, since the estimated PDF directly depends on this value. Because we know the resolution of the image through the direct sensor orientation, we are able to adapt the bandwidth of the Gaussian kernel for any given input image. But we have to estimate this bandwidth once to achieve best results for further processing. In probability theory, there are several methods to estimate the bandwidth of kernel functions for given observations. One well-known approach is using statistical classification. This method is based on computing the PDF using different bandwidth parameters and then comparing them. Unfortunately, in our field, such a framework can be very time-consuming for large input images. The other well-known approach is called balloon estimators. This method checks k nearest neighborhoods of each observation point to understand the density in that area. If the density is high, bandwidth is reduced proportional to the detected density measure. This method is generally used for variable-kernel density estimation, where a different kernel bandwidth is used for each observation point. However, in our study, we need to compute one fixed kernel bandwidth to use at all observation points. To this end, we follow an approach slightly different from balloon estimators.

First, we pick $K_i/2$ number of random observations (FAST feature locations) to reduce the computation time. For each observation location, we compute the distance to the nearest neighbor observation point. Then, the mean of all distances gives us a number l (calculated as 105.6 for Stadium_1). We assume that variance of Gaussian kernel (σ^2) should be equal to or greater than l. To guarantee the intersection of kernels of two close observations, we assume variance of Gaussian kernel as $5l$ in our study. Consequently, bandwidth of Gaussian kernel is estimated

as $\sigma^2 = 5l$. For a given sequence, that value is computed only one time over one image. Then, the same σ value is used for all observations extracted from images of the same sequence. The introduced automatic kernel bandwidth estimation method makes the algorithm robust to scale and resolution changes. In Fig. 2(d), we represent the PDF obtained for Stadium_1 test image. The represented PDF function is color coded, which means yellow-red regions show high probability values and dark blue regions show low probability values. As can be seen in this figure, crowded areas have very high probability values, and they are highlighted in estimated PDF. We use the automatic thresholding method of Otsu[21] on this PDF to detect regions having high probability values. After thresholding our PDF function, in the binary image obtained we eliminate regions with an area <1000 pixels, since they cannot indicate large human crowds. The resulting binary image $B_c(x, y)$ holds dense crowd regions. For Stadium_1 image, boundaries of detected crowd regions are represented on original input image with blue borders in Fig. 2(e). After detecting very dense groups, in the next step we focus on detecting other people in sparse groups. After detecting dense crowds automatically, we also extract quantitative measures from detected crowds for more detailed analysis. Because they indicate local color changes, we assume that detected features can give information about number of people in crowded areas. Unfortunately, the number of features in a crowd region does not give the number of people directly. In most cases, shadows of people or small gaps between people also generate a feature; in addition, two neighbor features might come from two different chroma bands for the same person. To decrease counting errors from these features, we follow a different strategy to estimate the number of people in detected crowds. We use a binary mask $B_f(x, y)$ where the image has zero values but the (x_i, y_i) feature locations have value 1. Then, we dilate $B_f(x, y)$ using a disk-shape structuring element with a radius of 2 to connect close feature locations in binary mask.[22] Finally, we apply connected component analysis to the mask, and we assume the total number of connected components in a crowd area as the number of people (N).[22] In this process, a slight change of radius of a structuring element does not make a significant change in estimated people number N. However, an appreciable increase in radius can connect features coming from different people and that decreases N, which leads to poor estimates of number of people. Because the resolution of the input image is known, using an estimated number of people in the crowd, the density of people (d) can also be calculated. Let us assume $B_c^j(x, y)$ is the j'th connected component in $B_c(x, y)$ crowd mask. We calculate crowd density for j'th crowd as $d^j = N/[\sum_X \sum_Y B_c^j(x, y) \times a]$, where X and Y are the numbers of pixels in the image in horizontal and vertical directions, respectively, and a is the area of one pixel in square meters.

2.4 *Detecting People in Sparse Groups*

Besides detecting dense crowd regions and extracting quantitative measures on them, detecting other people in noncrowd regions is also crucial, because detecting people in noncrowd regions can help to develop people-tracking or behavior-understanding systems.

To detect people in noncrowd regions, we apply connected component analysis[22] to $B_f(x, y)$ matrix and pick mass centers of the connected components (x_p, y_p) $p\epsilon[1, 2, \ldots, K_p]$ which satisfy $B_c(x_p, y_p) = 0$ as locations of individual people in sparse groups. Unfortunately, each (x_p, y_p) $p\epsilon[1, 2, \ldots, K_p]$ location satisfying this rule does not directly indicate a person, because the location might be coming from irrelevant local features of another object such as a tree or chimney. To decide whether a (x_p, y_p) position is indicating a person or not, we apply a background comparison test. At this step, to represent a person, the background color of a connected component centered in (x_p, y_p) position should be very similar to the background color of detected dense crowds. To do a background similarity test, first we pick all border pixels of the binary objects (crowd regions) in $B_c(x, y)$ binary crowd mask. We assume L_c, a_c, and b_c as mean of L, a, and b color band values of these pixels. For each (x_p, y_p) $p\epsilon[1, 2, \ldots, K_p]$ location which satisfies $B_c(x_p, y_p) = 0$ equation, we apply the same procedure and obtain L_p, a_p, and b_p values, which indicate mean of L, a, and b color band values around connected components located at (x_p, y_p) center point. To test background similarity, we check if extracted values satisfy inequality given below:

$$\sqrt{(L_c - L_p)^2 + (a_c - a_p)^2 + (b_c - b_p)^2} < \xi. \quad (5)$$

In our study, we selected $\xi = 10$ after extensive tests. Although slight changes of ξ value do not affect the detection result, a large increase of this threshold might lead to false detections; on the other hand, a large decrease might lead to inadequate detections. We should add that it is not possible to detect individuals standing on different colored surfaces with this method.

3 Experiments

To test our method, we use airborne images obtained using the low-cost airborne frame camera system (named 3K camera system) developed at DLR. The spatial resolution (GSD) and swath width of the camera system range between 15 and 50 cm, and 2.5 to 8 km, respectively, depending on the flight altitude. Within 2 min, an area of approximately 10 by 8 km can be monitored. That high observation coverage gives great advantage to monitor large events. Image data are processed onboard by five computers using data from a real-time global positioning system/inertial measurement unit system including direct georeferencing.[15] In this study, we use data with 15-cm GSD acquired from 1000-m flight altitude. The 3K airborne camera image data set consists of a stadium entrance data set (Stadium_{1-43}), which includes 43 multitemporal images acquired with a time distance of 0.5 s or 2 Hz. We also use one-shot airborne images taken over open-air concerts, Oktoberfest, and a festival for our crowd- and people-detection tests. All images are georeferenced to get absolute coordinates of the detected objects and allow us to detect exact geographical coordinates of the objects even if the images are taken from different positions of the aircraft. That property of the developed software gives us also the opportunity to easily display our results on Google Earth. Because of focusing difficulties of the older version of the camera system, some of the images in our data set are blurred. Although this issue decreases the detection capabilities of our software system, the results can still provide important information about status of the crowds and approximate quantitative measures of crowd and noncrowd regions. Furthermore, we also represent some test results obtained by using satellite images.

3.1 *Crowd-Detection Experiments*

To obtain a measure about the performance of the crowd-analysis step of the algorithm, we have generated reference data for four dense crowds in Stadium_1, represented in Fig. 3. Because even for a human observer it is hard to count the exact number of people in crowds, we have assumed mean counts of three human observers as reference. In Table 1, we compare the automatically detected number of people (N) and density (d) with the reference data (N_{gth} and d_{gth}, respectively) for each crowd. Similarity of our measures with the reference shows the high performance of the proposed approach.

In Fig. 4, we represent a crowd-detection result for a test image that covers a large part of the Oktoberfest event region. It can be seen that mainly the densely crowded areas in the roads near the tents are detected correctly.

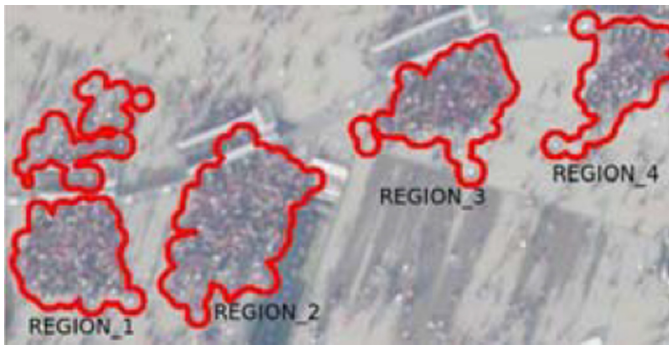

Fig. 3 A small part of Stadium_1 test image. Labels of detected crowds used for performance analysis are written on the image.

Table 1 Comparison of reference and automatically detected people number and density estimation results for test regions in $Stadium_1$ image.

	$Region_1$	$Region_2$	$Region_3$	$Region_4$
N	139	211	115	102
N_{gth}	132	180	114	98
d	0.81	0.74	0.68	0.76
d_{gth}	0.76	0.63	0.67	0.73

N, number of people; d, number of people per square meter.

In our automatic kernel-density estimation method, we assumed variance of Gaussian kernel as equal to $5l$. To prove our assumption and visualize effects of using different kernel-variance values, in this step we provide example results for different Gaussian kernel function variances. In Fig. 5, we provide crowd detection results for $Stadium_1$ test image for $1l$, $2l$, $3l$, $10l$, and $20l$ Gaussian kernel variance values. As can be seen in these results, assuming variance value equal to $1l$ leads to underestimations, because the width of the kernel function cannot be sufficient to merge probabilities of people. Assuming kernel variance as $2l$ and $3l$, we obtain results very similar to those of our previous assumption (assuming variance as $5l$). That proves robustness of the kernel density variance parameter to the chosen tolerance values. In last two images, although we use very high kernel-variance values, we could detect dense crowd regions. However, detected crowd boundaries appear larger owing to the very high width of the Gaussian probabilities.

In Fig. 6, we represent another dense crowd detection from an open-air concert in another region. The correct dense crowd boundary detection, despite the dense and diverse texture characteristics in the image, indicates the reliability of the proposed system.

In a previous study, Sirmacek et al.[23] tested FAST feature extraction-based crowd detection software on the first 16 airborne images of the Stadium image sequence. They have compared the software with other software solutions built in the same process structure but using different feature extraction methods (SPARK, ETM, and LOG features). The experiments showed that using the FAST feature extraction method, it is possible to obtain superior crowd detection performance. Using the 16 images of the test data set, the FAST feature extraction-based crowd

Fig. 4 Crowd detection result on $Munich_2$ test image.

Fig. 5 Crowd detection results for Stadium$_1$ test image for 1l, 2l, 3l, 10l, and 20l Gaussian kernel-variance values.

Fig. 6 Crowd detection and individual detection example over a complex scene.

detection software achieved performance computations of 87.21% for the l area true detection rate and 13.46% false detection rate. For comparison, manually generated dense crowd masks (binary images where dense crowd pixels are labeled with 1 and the other pixels are labeled with 0) have been used as reference.

The main advantage of the proposed system is its capability to adapt itself to spatial resolutions of the input images by detecting the l value automatically, which is important if the algorithms will be used in a real-time environment. To prove the robustness of the system to the spatial resolutions of the input images, we present crowd detection examples on a GeoEye-1 satellite image in Fig. 7. Results on satellite images of approximately 0.5 m (GSD) indicates the robustness of the automatically adapted system parameter.

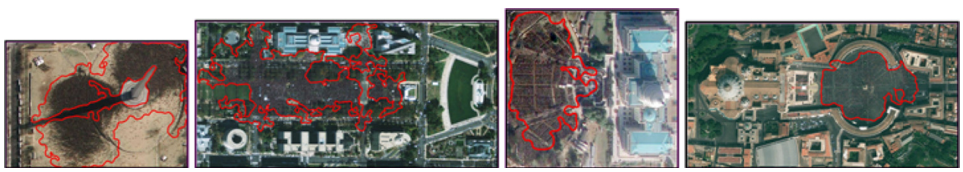

Fig. 7 Crowd detection examples on GeoEye-1 images are represented to prove robustness of the system parameters.

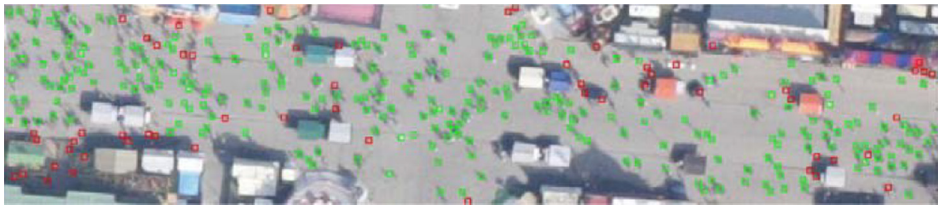

Fig. 8 Closer view of Munich$_1$ test image is presented to give information about people detection in sparse regions. True detections are labeled with green, and false detections are labeled with red.

3.2 *People-Detection Experiments*

In this section, we discuss the single people detection performance of the proposed method. We provide a detection performance illustration in Fig. 8, which is a zoomed subregion of Munich$_1$ test image. In this zoomed image part, we represent detected person locations with boxes. We have manually changed the marker box color to green for true detected persons and red for false alarms. A high number of green markers indicates high detection performance of the system. We could not prevent detection of local disturbances, which are represented with red box markers.

Unfortunately, due to the limited resolution and focusing difficulties of the airborne camera, it is difficult to generate good reference data for accurate performance calculation. To be able to discuss quantitative results, we asked three volunteers to label the individuals that they can see in the dense crowd (red-labeled) region in Fig. 6. In Fig. 9, we represent the labels generated by the volunteers. To generate reference data, we stored blue label locations in binary masks with value 1 where a person exists and value 0 in other places.

In Table 2, we tabulate the quantitative performance calculation results by using three different references given in Fig. 9. As can be seen in this table, the three volunteers labeled almost the same number of individuals. The slight difference might be because of the low sharpness of the image for displaying very small objects. We have accepted the software result as a correct detection if it is one pixel around a reference label. Our FAST feature–based software did not detect any false alarm in the dense crowd region. Unfortunately, the software could not detect each individual, especially in those regions where the object sharpness was very low. In future steps of our research, we would like to adapt feature extraction parameters depending on the input image sharpness estimation to obtain higher detection performances.

3.3 *Computation Time*

Finally, in this section we analyze computation time needed for our method. For Stadium$_1$ test image from our dataset, total dense crowd detection and individual detection modules take

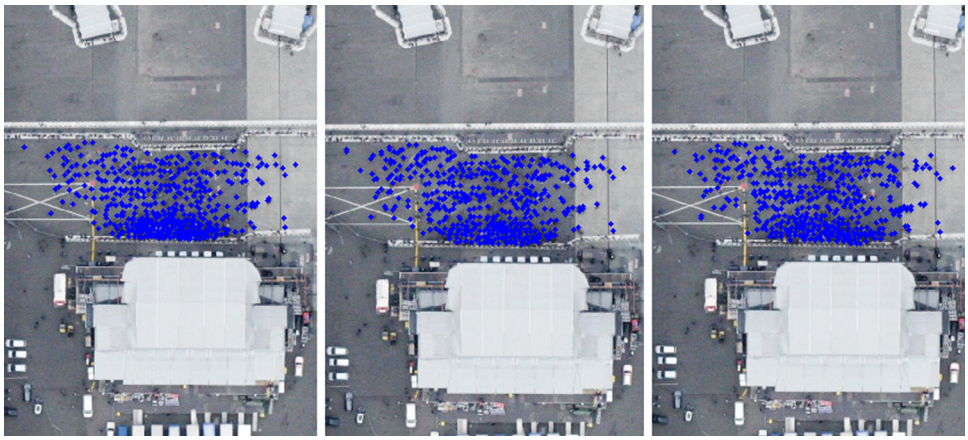

Fig. 9 Closer view of the dense crowd region in Fig. 6. Three different images show the labels generated manually by three different volunteers.

Table 2 Comparison of reference and automatically detected people number results for the dense crowd region in Fig. 6.

	Reference by Volunteer 1	Reference by Volunteer 2	Reference by Volunteer 3
Person number detected by volunteer	443	447	447
Person number detected by software	335	338	334
True detection percentage	75.62	75.61	74.72
False alarm percentage	0	0	0

107.20 s. We obtained timings using an Intel Core2Quad 2.66 GHz PC and MatLab coding environment. Total computation time for detecting dense crowds with the estimated people and people density numbers, and also for detecting individuals, show the practical usefulness of the method for on-board real-time applications. We plan to achieve higher computation time performances in a C programming environment.

4 Conclusions

To bring a novel solution for dense crowd and individual detection problem, herein we propose a fully automatic approach using remotely sensed images. Although the resolutions of airborne images are not enough to see each person with very sharp details, we can still notice a change of color components in the place where people or groups of people exist. Therefore local feature extraction-based software gave us the opportunity to develop a software system that can give high detection performances. We tested our crowd and people detection algorithm on airborne images taken from different events having diverse characteristics in their scenes. To test the robustness of the self-adaptive system parameter, we also applied dense crowd detection algorithm on GeoEye-1 satellite images. The experimental results and parameter robustness tests indicate possible use of the algorithm in real-life events, also for on-board applications.

References

1. A. Davies, J. Yin, and S. Velastin, "Crowd monitoring using image processing," *IEEE Electron. Commun. Eng. J.* **7**(1), 37–47 (1995), http://dx.doi.org/10.1049/ecej:19950106.
2. C. Regazzoni and A. Tesei, "Local density evaluation and tracking of multiple objects from complex image sequences," in *Proc. 20th Int. Conf. on Industrial Electronics, Control and Instrumentation*, Vol. 2, pp. 744–748, IEEE, Bologna, Italy (1994).
3. C. Regazzoni and A. Tesei, "Distributed data fusion for real time crowding estimation," *Signal Process.* **53**(1), 47–63 (1996), http://dx.doi.org/10.1016/0165-1684(96)00075-8.
4. O. Arandjelovic, "Crowd detection from still images," in *Proc. of the Br. Mach. Vis. Conf.* (2008).
5. W. Ge and R. Collins, "Marked point process for crowd counting," in *Proc. IEEE Comp. Vis. Pattern Recognit.*, pp. 2913–2920, IEEE, Miami, Florida (2009).
6. W. Ge and R. Collins, "Crowd detection with a multiview sampler," in *Proc. 11th Eur. Conf. Comp. Vis.*, pp. 324–337, Springer-Verlag, Berlin, Heidelberg (2010).
7. S. Lin, J. Chen, and H. Chao, "Estimation of number of people in crowded scenes using perspective transformation," *IEEE Trans. Syst. Man Cybernet. A Syst. Hum.* **31**(6), 645–654 (2001), http://dx.doi.org/10.1109/3468.983420.
8. S. Hinz, "Density and motion estimation of people in crowded environments based on aerial image sequences," in *ISPRS Hannover Workshop on High-Resolution Earth Imaging for Geospatial Information*, Hannover, Germany (2009).

9. F. Burkert et al., "People tracking and trajectory interpretation in aerial image sequences," in *Int. Arch. Photogram. Remote Sens. Spat. Inf. Sci. Commision III (Part A)*, Paris, France, Vol. XXXVIII, pp. 209–214 (2010).
10. B. Sirmacek and P. Reinartz, "Automatic crowd analysis from airborne images," in *Proc. 5th Int. Conf. Rec. Adv. Space Tech.*, pp. 116–120, IEEE, Istanbul, Turkey (2011).
11. B. Sirmacek and P. Reinartz, "Automatic crowd density and motion analysis in airborne image sequences based on a probabilistic framework," in *Proc. 2nd IEEE Int. Conf. Comp. Vis. Workshop.*, pp. 898–905, IEEE, Barcelona, Spain (2011).
12. B. Sirmacek and P. Reinartz, "Kalman filter based feature analysis for tracking people from airborne images," in *ISPRS Workshop High-Resolution Earth Imaging for Geospatial Information*, Hannover, Germany (2011).
13. R. Kalman, "A new approach to linear filtering and prediction problems," *J. Basic Eng.* **82**(1), 35–45 (1960), http://dx.doi.org/10.1115/1.3662552.
14. M. Butenuth et al., "Integrating pedestrian simulation, tracking and event detection for crowd analysis," in *Proc. of the 1st IEEE ICCV Workshop on Modeling, Simulation and Visual Analysis of Large Crowds*, pp. 150–157, IEEE, Barcelona, Spain (2011).
15. F. Kurz et al., "Low-cost optical camera system for disaster monitoring," in *Int. Archives of the Photogrammetry, Remote Sens. and Spatial Information Sci.*, Vol. XXXIX-B8, pp. 159–175, XXII ISPRS Congress, Melbourne, Australia (2012).
16. E. Rosten, R. Porter, and T. Drummond, "Faster and better: a machine learning approach to corner detection," *IEEE Trans. Pattern Anal. Mach. Learn.* **32**(1), 105–119 (2010), http://dx.doi.org/10.1109/TPAMI.2008.275.
17. D. Hearn and M. P. Baker, *Computer Graphics*, Prentice-Hall (1994).
18. D. Comanicu and P. Meer, "Mean shift: a robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Mach. Intell.* **24**(5), 603–619 (2002), http://dx.doi.org/10.1109/34.1000236.
19. B. Sirmacek and C. Unsalan, "A probabilistic framework to detect buildings in aerial and satellite images," *IEEE Trans. Geosci. Remote Sens.* **49**(1), 211–221 (2011), http://dx.doi.org/10.1109/TGRS.2010.2053713.
20. B. Silverman, *Density Estimation for Statistics and Data Analysis*, 1st ed., Chapman and Hall/CRC (1986).
21. N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Syst. Man Cybernet.* **9**(1), 62–66 (2009), http://dx.doi.org/10.1109/TSMC.1979.4310076.
22. M. Sonka, V. Hlavac, and R. Boyle, *Image Processing: Analysis and Machine Vision*, 3rd ed., CL-Engineering, Lubbock, Texas (2007).
23. B. Sirmacek et al., "Performance assessment of automatic crowd detection techniques on airborne images," in *Proc. of the IEEE International Geoscience and Remote Sensing Symposium*, Munich, Germany, pp. 2198–2201 (2012).

Beril Sirmacek received the BSc and MSc degrees from the Department of Electronics and Communication Engineering in Yildiz Technical University, Istanbul, in 2006 and 2007, respectively, and the PhD degree from the Department of Electrical and Electronics Engineering, Yeditepe University, Istanbul in 2009. During her PhD study, she was a research and teaching assistant and a member of the computer vision research laboratory at Yeditepe University. In this period, she has also made collaborations with many different universities and worked as a visiting researcher in remote sensing laboratory in the Department of Information Engineering and Computer Science in University of Trento, Italy. After receiving PhD degree, she has worked as a research fellow in the Department of Photogrammetry and Image Analysis in Remote Sensing Technology Institute of German Aerospace Center, as a guest lecturer in Institute of Computer Science at University of Augsburg, and a teaching assistant for image processing course at Technical University of Munich in Germany. Besides, she has started to pursue habilitation degree at Institute of Geoinformation at University of Osnabrueck. Recently, she is working with Delft University of Technology at Faculty of Aerospace Engineering in the Netherlands. She is engaged with developing computer vision interfaces for detailed 3D damage analysis of historical art objects.

Peter Reinartz received the diploma (Dipl.-Phys.) in theoretical physics in 1983 from the University of Munich and his PhD (Dr.-Ing) in civil engineering from the University of Hannover in 1989. His dissertation was on statistical optimization of classification methods for multispectral image data. He is department head of the Photogrammetry and Image Analysis Department at the German Aerospace Centre, Remote Sensing Technology Institute and holds a professorship for geoinformatics at the University of Osnabrück. He has more than 25 years of experience in image processing and remote sensing and over 200 publications in these fields. His main interests are in direct georeferencing, stereo-photogrammetry and data fusion of space borne and airborne data, generation of digital elevation models and interpretation of VHR data from sensors like Ikonos, Quickbird a.o. He is also engaged in using remote sensing data for disaster management and using high frequency time series of airborne image data for real time operations in case of disasters as well as for traffic monitoring.