Video Processing Algorithms for Detection of Pedestrians

Karol Piniarski, Paweł Pawłowski, Adam Dąbrowski

Poznan University of Technology
Department of Computing, Division of Signal Processing and Electronic Systems
Piotrowo 3, 60-965 Poznań, Poland
E-mails: {pawel.pawlowski, adam.dabrowski}@put.poznan.pl,karol.piniarski@doctorate.put.poznan.pl

Received: 12 September 2014; revised: 26 May 2015; accepted: 08 June 2015; published online: 07 September 2015

Abstract: In this paper a video processing procedure for automatic detection of pedestrians is presented. It is planned to use it as a part of the automotive night vision system. Generally, such systems are either passive (i.e. those based on thermal vision) or active (i.e. equipped with illuminators and near infrared cameras). Passive systems provide a large range of detection, while their active counterparts, operating in a somehow smaller range, offer more readable images for car drivers. However, all images produced with both kinds of these systems are quite specific and special image processing procedures are needed for them. For this purpose the authors used modified and adapted algorithms, such as dual-threshold locally adaptive classification, connected component labeling, histogram of oriented gradients, and the support vector machine with a radial basic function kernel or with a linear kernel. Tests performed on the real night vision recordings show very high efficiency of the proposed solution with accuracy equal to 99.2% for the linear kernel and even to 99.36% for the radial basic function kernel.

Key words: video processing, night vision, object detection, classification of pedestrians, automotive systems

I. INTRODUCTION

Thanks to new achievements in the technological sciences it is possible to offer tools that can aid transportation safety. In the automotive-related areas we can find such mechanisms as roads planning, roads security, assisting of drivers and their capabilities, protection of drivers and passengers, protection of pedestrians, and many others [1]. The growth of motorization and increased traffic volume help to develop our civilization, but at the same time increase the risk of accidents. According to [2] 38% of fatal accidents in the European Union (EU) occur in darkness, despite the fact that in general the traffic at night is several times lighter than during the day. About 20% of traffic accident victims are pedestrians [3], while more than half of pedestrian deaths take place at night (51%) [3]. Pedestrian fatalities that occur at night mainly result from such factors as poor visibility, drunken pedestrians, and drunken drivers.

In view of the above problems, many organizations set up preventive measures. With the efforts undertaken by the EU (e.g. the "Road Safety Program" [1]), the total number of fatalities in car accidents is falling rapidly. It changed from 54,000 in 2001 to 31,000 in 2010 [4]. If we count accidents related to pedestrians, we get 9,100 in 2001 and 5,500 in 2010. This means a global downward trend in the average pedestrian fatalities across the EU, but we also notice some exceptions [3]. In some countries, especially those of rapid economic growth, e.g. in Poland and Romania, this trend is somehow weaker, i.e. in Poland there were 1,866 pedestrian fatalities in 2001 vs. 1,236 in 2010 [3].

Automotive companies offer more and more solutions that increase safety of the night traffic, including adaptive (intelligent) front lights, detection of the driver's weariness or intoxication, warning of lane departure, recognition of traffic signs [5], information of a vehicle blind spot, automatic braking (typically working under limited speed thus dedicated to the city limits and traffic jams).

Despite the importance explained above, up to now only few manufacturers have been offering advanced systems for night vision although they can substantially improve the driver perception, offer more time to take a decision, and protect against accidents with pedestrians, who are practically defenseless in contact with vehicles.

Herein we present the video processing algorithms which realize a pedestrian detection facility and are dedicated to the automotive night vision systems [6]. The video processing is divided into three main steps: preprocessing, object detection, and object classification. Proper and reliable detection of pedestrians depends not only on a selection of procedures, but also on their fine tuning. A crucial part of the tuning is adaptation of procedures to quite specific night vision data.

The paper is organized as follows: after an introduction we compare basic night vision systems that are typically used in the modern automotive systems. Then we describe in detail the video processing algorithms that have been used in our solution. Finally, we present results of experiments on the night vision video databases and formulate conclusions.

II. NIGHT VISION SYSTEMS

Taking image acqusition methods into account, the automotive night vision (thermo-vision) systems can be classified twofold: as passive or active systems. The passive systems capture infrared (IR) radiation which is naturally emitted by objects, while the active systems are equiped with infrared illuminators and capture the light reflected from the objects. The videos obtained by these two types of systems differ considerably and thus the video processing algorithms should be optimized separately for each of these two types of systems.

A. Passive systems

In passive systems the thermal imaging camera captures infrared radiation (heat) emitted by objects. Each object with a temperature higher than 0 K emits radiation, but in practice only the objects with temperature other than the surroundings become distinctive. The passive systems detect the electromagnetic radiation with wavelengths in the range of 3-30 μm (far infrared or FIR for short), but the cameras that are used for people detection most commonly use a narrower range, i.e. 8-14 μm [7]. The human body with the temperature of about 300 K has in this band the highest energy emission [8]. As a result, the internally heated objects such as pedestrians and cars in motion (with engines, radiators, heated reflectors) are clearly visible.

A high contrast between living beings and the surroundings is one of the major advantages of passive systems. A range of detection distances is much larger than in active systems, and for high-quality cameras it can even reach 300 m. Thermal imaging cameras are also not blinded by the lights of other vehicles. This feature is very important, because it does not distract a driver.

The main disadvantage of the passive thermo-vision comes from the physical basis of this type of imaging: the measured emission of an object strongly depends on the source material and the covering of the object. It makes the calibration of the system difficult and strongly context-dependent. Fortunately, the absolute calibration of the camera in the automotive night-vision applications is not as important as, for example, for the typical thermal imaging in the construction industry.

Among other disadvantages of passive thermal cameras are lower resolution and higher costs than for the cameras used in the active systems. Because of a specific way of image capture, they are characterized by weak representation of the textures and by low signal dynamics [8]. Additionally, the infrared spectrum is more difficult to interpret for a driver, e.g. tires are white (hot), and the rest of the car is black (cold). Other, typically high-contrast objects like horizontal lane markings, cool headlamps (LED or rear lights) are not visible in the passive thermal images. Another important disadvantage of passive thermal systems is their sensitivity to changes of thermal contrast: with season, weather, humidity, etc. Sometimes, especially in warm nights, this may lead to even practically zero contrast.

B. Active systems

Active systems use vision feedback of infrared light close to the visible range (near infrared or NIR for short), emitted by infrared illuminators and captured by cameras. In this case, the typical wavelength range is 0.8-1.1 μm [8].

The main advantage of the active systems is high resolution. The image is easy to interpret for the driver because of the proximity of NIR to the visible light. We can, for instance, see the lanes and the headlights of oncoming vehicles. A relatively low cost of the cameras and their small size make them attractive and widely available. The cameras of this type can also be used in other systems and successfully work in daylight conditions (e.g. the CCTV cameras are often equipped with the mechanically switched IR filters used as a day/night switch). The NIR cameras also have a greater development potential than their passive counterparts, mainly due to rapid technological progress in the area of automatic video processing.

The active systems have shorter detection range than the passive ones and reach about 150 m. This distance strongly depends on the power of illuminators. However, this disadvantage is typically compensated by a higher resolution of image sensors. The NIR detectors can also be dazzled by the headlights (or illuminators) of oncoming vehicles, and operate significantly worse than the FIR cameras in the fog.

Concluding, both active and passive systems used in the automotive applications, like e.g. pedestrian detection, reach a typical range of about 90-100 m [7]. Active systems are cheaper and have better resolution than the passive ones, but for the pedestrian detection they need more complicated algorithms.

III. VIDEO PROCESSING ALGORITHMS

Automatic pedestrian detection is a relatively new area of digital video processing but, as it is very important, it grows rapidly. Both passive [9] and active [10-13] systems are used for night vision solutions. Most of them use trainable algorithms, like artificial neural networks (ANNs) [13], support vector machines (SVMs) [9, 11], etc.

A general video processing procedure for the pedestrian detection is presented in Fig. 1. The first stage is the image acquisition. The respective signal processing procedures are typically built in hardware, i.e. in the camera, and do not require additional resources.

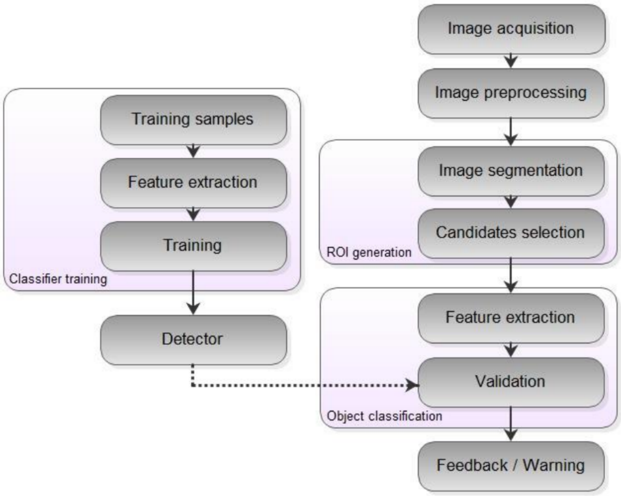


Fig. 1. Video processing for detection of pedestrians

After the image acquisition the image preprocessing stage is performed. This stage reduces noise of the image sensor and removes the interlaced scanning effect [10].

The next step shown in Fig. 1 prepares the so called region of interest (ROI), which is a selected part of the image for further processing. The properly selected ROI should consist of all objects which even potentially are in scope of the interest (the pedestrian candidates), and cannot miss important regions. In the presented case, the well generated ROI covers pedestrians to be detected, but at the same time significantly reduces the size of the analyzed image part, i.e. the amount of data which is transferred to the next stages. By this means we can substantially speed-up the data processing.

The first step of the ROI generation is image segmentation (Fig. 1). As we work with 2D images, we decided to use the threshold technique [13]. The algorithm translates the input gray scale image to the binary image, while white objects are the potential candidates to be detected as pedestrians and the background is black (cf. Fig. 3).

In order to produce uniform areas with clear edges, the actual thresholding algorithm must smoothly pass through neighboring pixels with values close to the threshold. For this purpose the hysteresis threshold technique with $T_L\left(i,j\right)$ – lower threshold and $T_H\left(i,j\right)$ – upper threshold, is used [9].

The segmentation process is defined as follows

$$S\left(i,j\right)=\begin{cases}0, & \text{if } I\left(i,j\right)<T_L\left(i,j\right) \text{ or } I\left(i,j\right)\in \\ & \in\left(T_L\left(i,j\right),T_H\left(i,j\right)\right)\cap S\left(i-1,j\right)=0 \\ 1, & \text{if } I\left(i,j\right)>T_H\left(i,j\right) \text{ or } I\left(i,j\right)\in \\ & \in\left(T_L\left(i,j\right),T_H\left(i,j\right)\right)\cap S\left(i-1,j\right)=1\end{cases} \tag{1}$$

where $S\left(i,j\right)$ is the segmented binary image after thresholding. For the pixel values greater than $T_H\left(i,j\right)$ or lower than $T_L\left(i,j\right)$ values 1 (white) and 0 (black) are assigned, respectively. If the pixel value is in the range $\left(T_L\left(i,j\right),T_H\left(i,j\right)\right)$, the output value depends on the previous sample $S\left(i-1,j\right)$.

The decision thresholds $T_L\left(i,j\right)$ and $T_H\left(i,j\right)$should be calculated for individual pixels with some knowledge of the neighborhood. Basing on our experience we decided to use a 1D horizontal neighborhood. Indeed, in the passive systems the intensity of a pedestrian object depends on the clothes (material thickness and texture). In consequence, the object is not homogeneous in the vertical axis. For this reason, a horizontal neighborhood is enough (cf. Fig. 2). Denoting the scanning width as w the analyzed neighborhood equals $2w+1$.

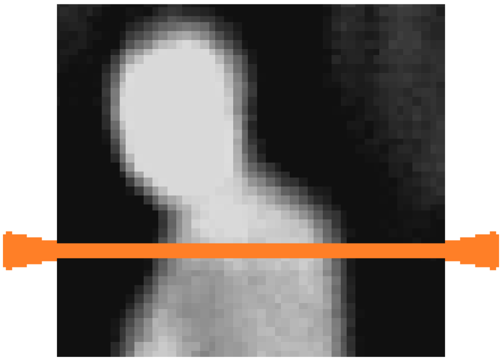


Fig. 2. Pedestrian and horizontal scanning line

Both thresholds should be defined locally and adaptively under various lighting conditions and resulting contrasts in the image. This technique guarantees reliable extraction of pedestrians. The lower threshold value $T_L\left(i,j\right)$ should be the mean of the neighborhood pixels

$$T_L\left(i,j\right)=\frac{1}{2w+1}\sum_{k=i-w}^{k=i+w} I\left(k,j\right). \tag{2a}$$

A proper value of the upper threshold $T_H\,(i,j)$ should be defined as the lower threshold corrected by a value proportional to the standard deviation $\delta(i,j)$ of the neighborhood pixels

$$T_H\,(i,j) = T_L\,(i,j) + \lambda \cdot \delta\,(i,j)\,. \tag{2b}$$

The standard deviation is calculated as follows:

$$\delta\,(i,j) = \sqrt{\frac{1}{2w{+}1}\sum_{k=i-w}^{k=i+w}\left(I\,(k,j)-\mu\right)^2} \tag{2c}$$

where: $I\,(k,j)$ is the gray-level input image (k,j)-th pixel value, w is the scanning width and μ is the mean value of the horizontal neighborhood. To control the impact of the standard deviation on the upper threshold $T_H\,(i,j)$ the weight λ is added.

The segmented image obtained with the above procedure is shown in the middle of Fig. 3. This result is far from being satisfactory because of many unwanted artifacts. In order to get rid of them a special offset β is introduced. The respective formula for the lower threshold changes to

$$T_L\,(i,j) = \frac{1}{2w+1}\sum_{k=i-w}^{k=i+w} I\,(k,j) + \beta \tag{3}$$

The offset β should generally be a function of the standard deviation $\delta\,(i,j)$

$$\beta = f\big(\delta\,(i,j)\big) \tag{4}$$

but in reality it is chosen experimentally as a constant value (Section IV). The final segmentation results are shown on the right hand side of Fig. 3.

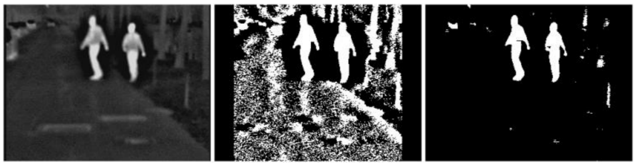


Fig. 3. Thresholding process: (from the left) original image, adaptive thresholding, the proposed adaptive thresholding with offset

After the image is segmented (cf. upper part of Fig. 5), the morphological opening removes small artifacts (bottom left of Fig. 5).

The last step in the ROI generation process is selection of candidates based on statistical features of the candidate objects. At the beginning of the analysis the algorithm looks for connected pixels which constitute objects, then these objects are separated and labeled. For this procedure the connected component labeling (CCL) [14] was adopted. Our implementation of the CCL algorithm is a one-pass linear-time version with a contour tracing technique [15] which is fast and accurate. This algorithm analyses the image line-by-line from the left to the right and from the top to the bottom starting with the upper-left image corner and ending in the lower-right corner. The procedure labels adjacent white pixels (those equal to 1) giving them the same label according to their 8-connectivity, if at least one of adjacent pixels has the same label. This main procedure is supplemented by two additional exceptions. The first one occurs if the scanning line reaches a new object, i.e. an external contour (see point A and gray area in the left part of Fig. 4). From this point the algorithm leaves the scanning line, traces the contour of the object and labels boundary pixels until it reaches the starting point A again. The second exception occurs when the scanning line finds an internal contour (see point B and the white area in the right part of Fig. 4). The contour is traced and the boundary pixels are labeled with the same label. After any of the above exceptions, the standard line scanning procedure is continued [15].

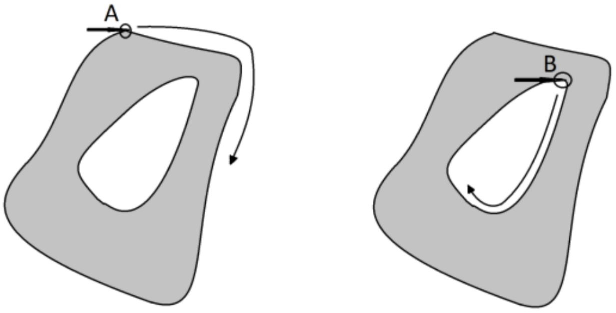


Fig. 4. Two special tracing cases in the connected component labeling: external contour (left), internal contour (right)

After the labeling is done, the objects are processed to calculate their characteristic features: width, height, area, and position. These features are inputs to the elimination algorithm which selects the most probable candidates for pedestrians and by this means constitutes the final ROI. According to [10], the object aspect ratio (height to width) is distributed for pedestrians in the range from 1:1.3 to 1:4. Thus only such objects are accepted (cf. the bottom right part of Fig. 5).

The next step after the ROI generation is the object classification (cf. Fig. 1). This stage is crucial and strongly affects quality of the pedestrian recognition [16]. The first step is the feature extraction which makes it possible to reduce the amount of data that describes the object. For the feature extraction we can use a histogram of oriented gradients (HOG) [13], shape context, and 1D/2D Haar transform. In the presented system the HOG was used. This method calculates gradients and forms histograms of gradients orientation. To improve reliability of the HOG the local normalization is used. Finally the ROI is represented by a locally normalized feature vector constructed from the histograms of orientation.

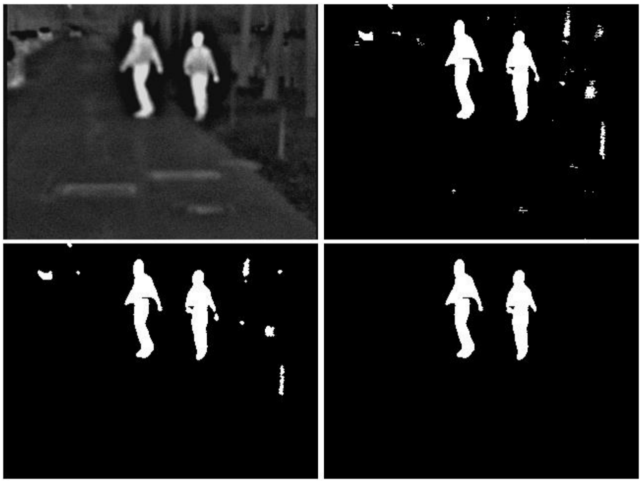


Fig. 5. Stages of processing: input picture (upper left), segmentation (upper right), morfological operations (bottom left), candidates selection (bottom right)

The first step of HOG algorithm consists in calculation of gradients G_i and G_j in the horizontal and vertical axes, respectively, with i and j treated for a moment as continuous variables

$$\nabla I\left(i,j\right)=\left(\begin{array}{c}G_i\\G_j\end{array}\right)=\left(\begin{array}{c}\dfrac{\partial I}{\partial i}\\\dfrac{\partial I}{\partial j}\end{array}\right),\tag{5}$$

where

$$\begin{aligned}&\frac{\partial I}{\partial i}\approx\frac{I\left(i{+}\Delta i,j\right)-I\left(i-\Delta i,j\right)}{2\Delta i},\\&\frac{\partial I}{\partial j}\approx\frac{I\left(i,\,j+\Delta j\right)-I\left(i,j-\Delta j\right)}{2\Delta j},\\&\Delta i,\Delta j=1.\end{aligned}$$

The above formula is equivalent with a convolution operation on the image with the filter kernels $\frac{1}{2}[-1\ 0\ 1]$ and $\frac{1}{2}[-1\ 0\ 1]^{\mathrm{T}}$ (but the factor $\frac{1}{2}$ can in fact be omitted). It is possible to use larger kernels, but for shape description it is not efficient (the proof is in [13]). After the gradients are computed, the magnitude and orientation of gradients can be obtained as

$$|\nabla I|=\sqrt{G_i^2+G_j^2}\tag{6}$$

$$\theta=\arctan\left(\frac{G_i}{G_j}\right).\tag{7}$$

The next step groups the pixels into cells (Fig. 6, left hand side), which usually have a square shape. In these cells, the orientation histograms (Fig. 6, the right hand side) are created with the use of orientation and magnitude. The histograms are divided into nine bins in the range from 0 to 360 degrees or 0 to 180 degrees. The authors of [13] claim that for nine bins the algorithm works best.

After the histograms are calculated, the four adjacent cells are grouped and create a block (Fig. 6, the left hand side). In this block a non-normalized vector $\mathbf{v}$ is created, which contains all histograms in a given block (here in four cells). Then, the vector $\mathbf{v}$ is normalized to get vector $\mathbf{v}_n$ with a formula

$$\mathbf{v}_n=\frac{\mathbf{v}}{\sqrt{\|\mathbf{v}\|_2^2+e^2}},\tag{8}$$

where e is a small constant. Finally, after normalization all these vectors in all blocks are combined into a single feature vector $\mathbf{v}_f$

$$\begin{gathered}\mathbf{v}_f=\Big[\mathbf{v}_{n[1\ 1]},\ \mathbf{v}_{n[1\ 2]},\ldots,\ \mathbf{v}_{n[1\ (m-1)]},\ \mathbf{v}_{n[1\ m]},\ldots,\\\mathbf{v}_{n[2\ 1]},\ \mathbf{v}_{n[2\ 2]},\ldots,\mathbf{v}_{n[2\ (m-1)]},\ \mathbf{v}_{n[2\ m]},\ldots\\\ldots\\\mathbf{v}_{n[(l-1)1]},\ \mathbf{v}_{n[(l-1)\ 2]},\ldots,\mathbf{v}_{n[(l-1)\ (m-1)]},\ \mathbf{v}_{n[(l-1)\ m]},\ldots,\\\ldots,\mathbf{v}_{n[l\ 1]},\ \mathbf{v}_{n[l\ 2]},\ldots,\ \mathbf{v}_{n[l\ (m-1)]},\ \mathbf{v}_{n[l\ m]}\Big]\end{gathered}\tag{9}$$

say $\mathbf{v}_{n[i\ j]}$ for all blocks $[i\ j]$, where $1\le i\le l$, $1\le j\le m$ (l,m are block indices in the analyzed image in the vertical and horizontal directions, respectively).

The last stage that finally validates the object is a classifier. The most common classifiers are: support vector machine (SVM) as an example of the supervised learning method, artificial neural networks, self-organizing maps (SOMs), and matrices of neurons [12]. A very helpful algorithm during classification is the boosting algorithm. By intensification of the most important samples it can produce one better classifier from several weaker classifiers. It also has good generalization properties. The best known implementation is AdaBoost [10].

For our application the kernel type SVM classifier has been selected. It is operating well with the antecedent stage, i.e. the HOG algorithm, and offers very good quality of detection. It is also very effective and commonly used in similar applications. The SVM is a supervised learning classification method. The goal of the SVM classifier is to separate objects $\mathbf{x}\in R^D$ being vectors in a multidimensional linear (in our case real valued) decision space R^D (D being the space dimension) into two classes labeled as $y\in\{-1,1\}$.

We assume that these two classes are linearly separable in a new space of higher dimensionality than D. This space is referred to as the feature space. It is composed of new vectors $\mathbf{\Phi}\left(\mathbf{x}\right)$, i.e., vectors $\mathbf{x}$ mapped with a mapping function $\mathbf{\Phi}\left(\mathbf{x}\right)$. The separation is done by the optimal hyperplane H in the feature space, obtained with the SVM learning process. For learning, L training samples $\mathbf{x}_i$ together with the correspond-

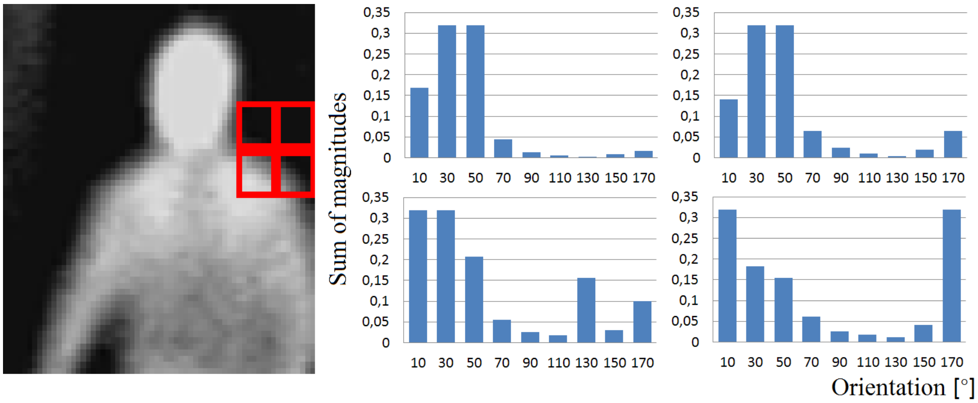


Fig. 6. Histograms of oriented gradients (HOG): image with highlighted block of four cells (left) and histograms of orientation for each cell (right)

ing labels $y_i \in \{-1,1\}, i=1,\ldots,L$, are used. The resulting hyperplane is described with the following equation

$$H=\{\boldsymbol{\Phi}(\mathbf{x}):\ g(\mathbf{x})=0\}\ \text{with}\ g(\mathbf{x})=\ \mathbf{a}^{\mathrm{T}}\boldsymbol{\Phi}(\mathbf{x})+b, \tag{10a}$$

where $\mathbf{a}$ is a vector of directional coefficients of the hyperplane and b is the appropriate offset value. The hyperplane parameters $\mathbf{a}$ and b are selected to maximize the shortest decision margin, i.e. the distance to the nearest vectors $\boldsymbol{\Phi}(\mathbf{x}_i)$ of both classes, referred to as the support vectors [17]. The optimal parameters $\mathbf{a}$ and b are normalized in such a way that the support vectors $\boldsymbol{\Phi}(\mathbf{x}_i)$ fulfill the following condition

$$\mathbf{a}^{\mathrm{T}}\boldsymbol{\Phi}(\mathbf{x}_i)+b=y_i\ \text{ or equivalently }\ y_i\left(\mathbf{a}^{\mathrm{T}}\boldsymbol{\Phi}(\mathbf{x}_i)+b\right)=1. \tag{10b}$$

The optimal hyperplane is obtained with the training procedure using L training pairs $\{\mathbf{x}_i, y_i\}$, $i=1,\ldots,L$. In order to find the optimal hyperplane, the so-called primary Lagrange function

$$L_{\mathrm{P}}=\frac{1}{2}\|\mathbf{a}\|^2-\sum_{i=1}^{L}\alpha_i\left[y_i\left(\mathbf{a}^{\mathrm{T}}\boldsymbol{\Phi}(\mathbf{x}_i)+b\right)-1\right] \tag{11a}$$

should be minimized or equivalently the so-called dual Lagrange function

$$L_{\mathrm{D}}=\sum_{i=1}^{L}\alpha_i-\frac{1}{2}\sum_{i,j=1}^{L}\alpha_i\alpha_j y_i y_j K(\mathbf{x}_i,\mathbf{x}_j) \tag{11b}$$

should be maximized with constraints

$$0\le\alpha_i\le C,\qquad \sum_{i=1}^{L}\alpha_i y_i=0 \tag{12}$$

where: α_i are the Lagrange multipliers and $K\left(\mathbf{x}_i,\mathbf{x}_j\right)$ is the so-called kernel function which is related to $\boldsymbol{\Phi}(\mathbf{x})$ by imposing: $K\left(\mathbf{x}_i,\mathbf{x}_j\right)=\boldsymbol{\Phi}(\mathbf{x}_i)^{\mathrm{T}}\boldsymbol{\Phi}(\mathbf{x}_j)$, and C is the penalty parameter which determines importance of the misclassification [18].

We have considered the following most popular kernel functions:

1. linear: in the simplest case $\boldsymbol{\Phi}(\mathbf{x})=\mathbf{x}$, then

$$K(\mathbf{x}_i,\mathbf{x}_j)=\boldsymbol{\Phi}(\mathbf{x}_i)^{\mathrm{T}}\boldsymbol{\Phi}(\mathbf{x}_j)=\mathbf{x}_i^{\mathrm{T}}\mathbf{x}_j \tag{13}$$

2. polynomial:

$$K(\mathbf{x}_i,\mathbf{x}_j)=\left(\gamma\ \mathbf{x}_i^{\mathrm{T}}\mathbf{x}_j+r\right)^d,\ \gamma>0 \tag{14}$$

3. radial basis function (RBF):

$$K(\mathbf{x}_i,\mathbf{x}_j)=\exp\left(-\gamma\,\|\mathbf{x}_i-\mathbf{x}_j\|^2\right),\ \gamma>0 \tag{15}$$

4. sigmoid:

$$K(\mathbf{x}_i,\mathbf{x}_j)=\tanh\left(\gamma\ \mathbf{x}_i^{\mathrm{T}}\mathbf{x}_j+r\right), \tag{16}$$

where: $\mathbf{x}_i$ and $\mathbf{x}_j$ are the training samples and γ, r, d are appropriate parameters [18].

We proved experimentally that the best kernel functions for our purpose are: the linear kernel (inner product) and RBF. The results are presented and discussed in Section IV.

The next step is to define a set S which contains indices of the support vectors. Then, parameters $\mathbf{a}$ and b have to be determined. Vector $\mathbf{a}$ can be calculated taking condition $(\partial L_{\mathrm{P}})/(\partial\mathbf{a})=0$ into account while offset b could directly be computed from expression (10b). However, for accuracy it is better to compute an average over all support vectors. Thus

$$\mathbf{a}=\sum_{i=1}^{L}\alpha_i y_i\boldsymbol{\Phi}(\mathbf{x}_i) \tag{17a}$$

$$b=\frac{1}{N_S}\sum_{i\in S}\left(y_i-\ \mathbf{a}^{\mathrm{T}}\boldsymbol{\Phi}(\mathbf{x}_i)\right), \tag{17b}$$

where N_S is the number of the support vectors. Finally, the trained classifier can analyze new samples $\mathbf{x}$ by evaluating for them

$$y = \operatorname{sgn}\left(\mathbf{a}^{\mathrm{T}}\mathbf{\Phi}\left(\mathbf{x}\right) + b\right). \tag{18}$$

For the RFB kernel the two parameters: γ (see (15)) and C (see (12)) should be tuned in order to find the optimal solution. The authors used a procedure described in [15], known as cross-validation. In this procedure the training set is divided into v subsets, then the SVM is trained for $v-1$ subsets and tested, each time with different parameters. Next, the search process can be repeated in a smaller set of the most efficient parameters. This method is very effective, as it is proven in the next section.

Some papers [10, 18] propose additional object tracking, e.g. with the use of the Kalman filter. It can improve the object detection quality and reduce false alarms, but it is a very computational intensive algorithm which is actually not necessary in our application.

To summarize, in the presented system the following computational techniques have been applied:

1. modified adaptive dual-threshold for the image segmentation,
2. connected component labeling(CCL) for selection of candidates,
3. histogram of oriented gradients (HOG) for feature extraction,
4. support vector machine (SVM) for training of the classifier.

IV. RESULTS OF EXPERIMENTS

In order to test the above presented video processing algorithm for the night vision system, the authors used two night vision video databases, i.e. the NTPD – "Night-time Pedestrian Dataset" [11] and the "USArmy Tetravision" [19].

The NTPD contains a set of images of pedestrians recorded by the active night vision system with the resolution of 64×128 (Fig. 7) and is divided into two sub-bases: for training and testing.

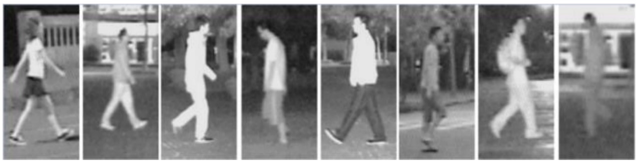


Fig. 7. Pedestrian samples

The "USArmy Tetravision" database has been prepared with the use of the passive system and it is in fact a movie recorded simultaneously with two stereo pairs of thermal cameras and NIR cameras (four video streams in total). Unfortunately, this database consists of several isolated pedestrians only. Using this database it is not possible to prepare data appropriate for both training and testing of the prepared algorithms. Thus this database was used for testing only.

A. Detector training with the NTPD

The above introduced NTPD was used to tune and test the proposed video processing algorithms for the pedestrian detection. All images used for training and testing were prepared to the feature extraction by scaling to the size of 64×128 pixels. For this size, the following values of HOG feature extractor were used: 15 blocks horizontally, 7 blocks vertically with four cells in each block. Those give $15 \cdot 7 \cdot 9 \cdot 4 = 3780$ features for each image. These features were isolated from the NTPD. After that appropriate matrices for training and testing in the classification stage were prepared. The entire set of the training data is presented in Tab. 1.

As a set of negative samples randomly selected parts of the background (with no visible pedestrians) were used (see Fig. 8).

Fig. 8. Examples of negative samples

The proposed classifier uses the radial basis function (RBF) kernel and alternatively the linear classifier. The optimal classifier was trained using the "cross-search" and "grid-search" methods using the NTPD database.

Tab. 1. Training and test samples in NTPD database

	No. of training samples	No. of test samples
positive samples	1998	2370
negative samples	8730	9000

In order to describe the effects of the classification, the following measures were used:

$$\mathrm{DR} = \frac{\text{correctly classified positive samples}}{\text{total number of the positive samples}} \tag{19}$$

$$\mathrm{FAR} = \frac{\text{falsely classified negative samples}}{\text{total number of the negative samples}} \tag{20}$$

$$\mathrm{CA} = \frac{\begin{array}{c}\text{correctly classified positive samples+}\\ \text{+ falsely classified negative samples}\end{array}}{\text{total number of all samples}}. \tag{21}$$

A special software for the tuning and testing of the proposed algorithm was prepared. It was written in the Microsoft

Visual Studio 2012 environment with the use of C# language as a part of .NET framework. For the video processing the EmguCV v. 2.4.2 library was used. This is a multi-platform library that is compatible with many programming languages and offers all functions from the commonly used OpenCV library [20].

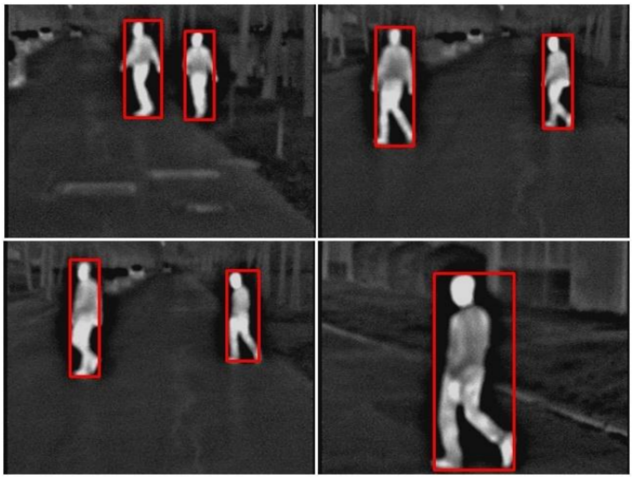


Fig. 9. Correctly detected pedestrians

The best results obtained after the optimization of the classifier, compared to the source papers, are presented in Tab. 2. As it can be seen, the classifier effectiveness is very high. The classifier with the RBF kernel is slightly better than that with the linear classifier. For comparison, the authors of the source database [11] reached DR = 94.39% only, in the same resolution and the linear SVM kernel. In another paper [13] ever worse DR = 82.9%, FAR = 6.5% are reported but they were obtained for a smaller resolution of 24×64 pixels. The algorithm presented in [9] is quite fast and works with 35 fps (frames per second). This confirms high efficiency of the linear solution for that kernel but the detection quality is worse.

Tab. 2. Classification results on a testing database for linear and RBF kernels

System (kernel)	Output parameters (γ; C)	DR [%]	FAR [%]	CA [%]
RBF	0.03375; 0.1	96.96	0.01	99.36
linear	not relevant	96.67	0.13	99.2
[9]	not relevant	82.90	6.50	88.2
[11]	not relevant	94.39	N.N.	N.N

where γ and C are parameters of the RBF kernel (see (12), (15)).

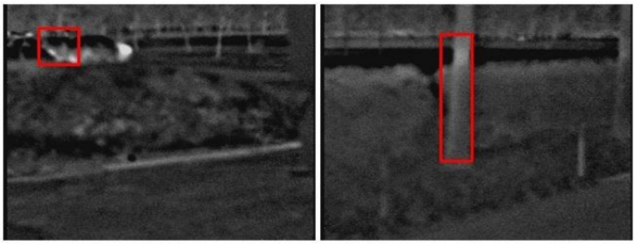


Fig. 10. False accepted objects

B. Pedestrian detecion with the USArmy Tetravision database

Global testing of the whole pedestrian detection algorithm (especially the ROI generation) with the use of the NTPD database only is not possible, because this database does not have separate testing data, i.e. the full resolution video frames assigned to training and testing. The second mentioned database, i.e. the "USArmy Tetravision", has the full resolution frames (320×240 pixels) although it is recorded in the passive system, as opposed ot the previous one. Despite basic differences in the characteristics of the image the authors performed trial tests. The classifier was trained with the NTPD database but tested with the "USArmyTetravision" movies. The DR efficiency was worse and the FAR classifier gave increased values, but for the majority of cases it was still working properly. This comes from the fact that the used HOG feature extractor is quite universal. Comparison to very low FAR in the first experiment (equal to 0.01) brings new conclusions: negative samples in the NTPD database were selected randomly and are quite far in the classification space from the positive ones. If the negative samples are more similar to the real pedestrians, e.g. telephone poles, free standing mailboxes, traffic signs, etc., the results in both cases may be closer.

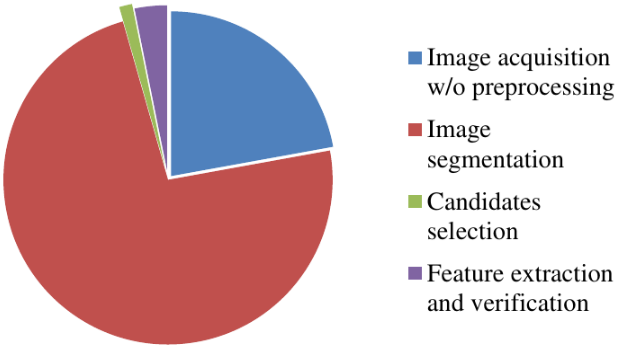


Fig. 11. Proportions of the computation time of particular image processing operations

After trial tests, the best (experimentally obtained) threshold values for the segmentation process have been set to: $\beta = 16$, $w = 20$, and $\lambda = 0.3$ (see (1),(2)). Figs. 9 and 10 present correctly detected pedestrians and cases with false classifications, respectively. The source images were taken from the "USArmy Tetravision" database, while the detection was performed by the proposed algorithm.

The computations were performed with the following hardware: CPU Intel Core 2 Duo 2.4 GHz, GPU GeForce 9600M GT, 6 GB of RAM (with no GPGPU usage).The total time of processing of one frame by the proposed algorithm implemented in C# environment (without optimization of data processing)is about 500 ms and this gives 2 fps in real time only. The biggest part of this time was consumed by the image segmentation process (see Fig. 11).

It should be mentioned that the algorithms were optimized to be as exact as possible, without any optimization of the processing speed. The processing speed optimization is planned for the final implementation on the DSP platform.

V. CONCLUSIONS

This paper presents an advanced video processing algorithm for automatic detection of pedestrians. It is thought to be a part of an automotive night vision system. Thanks to carefully selected, modified, and tuned advanced video and data processing techniques the proposed solution reaches a very good quality of detection. It was tested and proved with real night vision recordings. In the future it is planned to prepare our own night vision database with pedestrians and improve the speed of processing with the use of the GPGPU technique [20] and/or the digital signal processors [21].

Acknowledgments

The authors wish to thank M. Bertozzi, A. Broggi, M. Felisa, G. Vezzonifrom, University of Parma, Italy for sharing the "USArmy Tetravision" database of night vision movies and to Y. Zhang, Y. Zhao, G. Li, and R. Cheng from Guizhou University and Peking University Shenzhen, China for sharing the "Night-time Pedestrian Dataset" (NTPD).

This paper was financed with the 2015 DS means.

This is an extended version of the paper presented at the 18th IEEE SPA Signal Processing Conference, September 22-24, 2014, Poznan, Poland.

References

[1] European Commission, *Towards a European road safety area: policy orientations on road safety 2011-2020*, Brussels, COM(2010), 389, (2010).

[2] J. Broughton, C. Brandstaetter, G. Yannis, P. Evgenikos, et al., *Basic Fact Sheet "Seasonality"*, Deliverable D3.9 of the EC FP7 project DaCoTA, 4, (2012).

[3] J. F. Pace, J. Sanmartín, P. Thomas, A. Kirk, et al., *Basic Fact Sheet "Pedestrians"*, Deliverable D3.9 of the EC FP7 project DaCoTA, 1-12, (2012).

[4] European Commission, CARE, *Road fatalities in the EU since 2001*, EU road accidents database, (2013).

[5] P. Pawłowski, D. Prószyński, A. Dąbrowski, *Real-time procedures for automatic recognition of road signs*, Elektronika – konstrukcje, technologie, zastosowania, Sigma NOT, **3**, 57-61 (2009).

[6] K. Piniarski, P. Pawłowski, A. Dąbrowski, *Pedestrian Detection by Video Processing in Automotive Night Vision System*, Proc. of IEEE Signal Processing Algorithms, Architectures, Arrangements and Applications, SPA 2014, Poznań, Poland, 104-109, (2014).

[7] Y. Luo, J. Remillard, D. Hoetzer, *Pedestrian Detection in Near-Infrared Night Vision System*, IEEE Intelligent Vehicles Symposium, 51-58, (2010).

[8] F. Jahard, D. A. Fish, A. A. Rio, C. P. Thompson, *Far/Near Infrared Adapted Pyramid-Based Fusion for Automotive Night Vision*, International Conference on Image Processing and its Applications, Vol. 8, 886-890, (1997).

[9] Q. Liu, J. Zhuang, S. Kong, *Detection of Pedestrians at Night Time Using Learning-based Method and Head Validation*, Proc. of IEEE Conf. on Imaging Systems and Techniques (IST 2012), 398-402, (2012).

[10] J. Ge, Y. Luo, G. Tei, *Real Time Pedestrian Detection and Tracking at Night time for Driver-Assistance Systems*, IEEE Transactions on Intelligent Transportation Systems, Vol. 10, No. 2, 283-298, (2009).

[11] Y. Zhang, Y. Zhao, G. Li, R. Cheng, *Grey Self-similarity Feature for Night-time Pedestrian Detection*, Journal of Computational Information System 10: 7, 2967-2974, (2014)

[12] V. E. Neagoe, C. T. Tudoran, M. Neghina, *A neural network approach to pedestrian detection*, Proc. of ICCOMP'09, 374-379, (2009).

[13] N. Dalal, B. Triggs, *Histograms of Oriented Gradients for HumanDetection*, Proc. of IEEE Conference on Computer Vision and Pattern Recognition, Vol. 1, 886-893, (2005).

[14] R. Walczyk, A. Armitage, D. Binnie, *Comparative Study on Connected Component Labeling Algorithms for Embedded Video Processing Systems*, Proc. of Int. Conf Image Processing, Computer Vision, and Pattern Recognition, IPCV 2010, CSREA Press, Vol. 2, 853-859, (2010).

[15] F. Chang, C. J. Chen, C. J. Lu, *A linear-time component-labeling algorithm using contour tracing technique*, Computer Vision and Image Understanding, Vol. 93, No. 2, doi:10.1016/j.cviu.2003.09.002, 206-220 (2004).

[16] G. Szwoch, P. Dalka, A. Ciarkowski, P. Szczuko, A. Czyżewski, *Visual Object Tracking System Employing Fixed and PTZ Cameras*, Journal of Intelligent Decision Technologies, Vol. 5, No 2, 177-188, (2011).

[17] C. Cortes, V. Vapnik, *Support-Vector Networks*, Journal of Machine Learning archive **20**(3), 273-297 (1995).

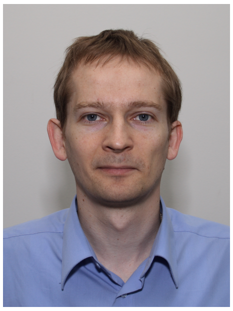
[18] T. Fletcher, *Support Vector Machines Explained*, University College London, (2009).

[19] M. Bertozzi, A. Broggi, M. Felisa, G. Vezzoni, *Low-level Pedestrian Detection by means of Visible and Far Infra-red Tetra-vision*, Proc. of IEEE Intelligent Vehicles Symposium, 231-236 (2006).

[20] A. Dąbrowski, P. Pawłowski, M. Stankiewicz, F. Misiorek, *Fast and accurate digital signal processing realized with GPGPU technology*, Electrical Review, R. 88, No 6, 47-50 (2012).

[21] T. Marciniak, D. Jackowski, P. Pawłowski, A. Dąbrowski, *Real-time people tracking using DM6437 EVM,* Proc. of IEEE Signal Processing Conference SPA 2009, 116-120, (2009).

MSc Eng. Karol Piniarski is a PhD student at the Poznan University of Technology, Faculty of Computing, Division of Signal Processing and Electronic Systems. He received MSc degree in Vision systems in the Control Engineering and Robotics field in 2014. His research interests include pedestrian detection, image processing, and machine learning.



Dr. Eng. Paweł Pawłowski is an assistant professor at the Division of Signal Processing and Electronic Systems, Poznan University of Technology, Poland. He finished this University in 2000 with MSc degree in electronics and telecommunications and in 2007 he received the PhD degree in automation and robotics. In 2001 he was guest researcher at the University Kaiserslautern, in 2008 at the University Cottbus, Germany. He is also a visiting researcher at the University of Technology and Life Sciences in Bydgoszcz, Poland. His research interests include real-time computing with exact or controlled variable arithmetic accuracy in floating-point arithmetics, microcontrollers and real-time video processing. He designed many measurement systems (e.g. ACCINO for testing road restraint systems, equipment for automatic testing of switched capacitor filters, ultrasonic measurement system for tracking cars, etc.). He is IEEE Member, reviewer of ISI journals (e.g., Journal of Circuits Systems and Computers, Applied Soft Computing, Computing and Informatics), international conferences: EUROCON, IEEE SPA, and author of over 100 scientific contributions.

Professor Dr. Habil. Eng. Adam Dąbrowski is a full professor in digital signal processing at the Faculty of Computing, Poznan University of Technology, Poland, and Chief of the Division of Signal Processing and Electronics Systems. His scientific interests concentrate on: digital signal processing (digital filters, signal separation, multidimensional systems, wavelet transformation), processing of medical images, biometrics, multimedia and vision systems, and on processor architectures. He is author or co-author of 5 books and over 300 scientific and technical publications. He was a Humboldt Foundation fellow at the Ruhr-University Bochum (Germany), visiting professor at the ETH Zurich (Switzerland), Catholic University in Leuven (Belgium), and professor of University of Kaiserslautern (Germany), and Technical University of Berlin (Germany). Currently he is Chairman of the Signal Processing (SP) and Circuits & Systems (CAS) Chapters of the Poland Section of IEEE (the Institute of Electrical and Electronics Engineers). In 1995 Professor Adam Dabrowski won for the CAS Chapter the IEEE Chapter of the Year Award, New York, USA. In 2001 he was also awarded with the diploma for outstanding position in the IEEE Chapter of the Year Contest.