Pre-attention Cues for Person Detection

Karel Paleček[1], David Gerónimo[2], and Frédéric Lerasle[3,4]

[1] Institute of Information Technology and Electronics,
Technical University of Liberec, Czech Republic
[2] Computer Vision Center, Autonomous University of Barcelona, Spain
[3] CNRS: LAAS, 7 Avenue Colonel Roche F-31077 Toulouse, France
[4] Université de Toulouse, UPS, INSA, INP, ISAE, LAAS-CNRS, Toulouse, France
karel.palecek@tul.cz, dgeronimo@cvc.uab.es, lerasle@laas.fr

Abstract. Current state-of-the-art person detectors have been proven reliable and achieve very good detection rates. However, the performance is often far from real time, which limits their use to low resolution images only. In this paper, we deal with candidate window generation problem for person detection, i.e. we want to reduce the computational complexity of a person detector by reducing the number of regions that has to be evaluated. We base our work on Alexe's paper [1], which introduced several pre-attention cues for generic object detection. We evaluate these cues in the context of person detection and show that their performance degrades rapidly for scenes containing multiple objects of interest such as pictures from urban environment. We extend this set by new cues, which better suits our class-specific task. The cues are designed to be simple and efficient, so that they can be used in the pre-attention phase of a more complex sliding window based person detector.

Keywords: person detection, candidate window generation, pre-attention.

1 Introduction

In the last two decades, human detection has been an active research area of computer vision. The algorithms for human detection are applicable in various tasks, e.g. surveillance systems [15], driver assistance [7,11] or human-machine interaction [10].

However, fast and robust human detection is still a challenging task for several reasons. The main problem is large variability of appearance, i.e. people can wear different clothes or take various poses. Typically, there are also other common factors such as variability in camera poses, lighting conditions or overall image quality.

Although several approaches for human detection in still images have been adopted over the years, some of the most successful detectors are based on a sliding window technique. This class of algorithms treats human detection task as a classification problem, i.e. input image is sequentially scanned and each sub-window is classified as containing or not containing a human. Since the size of the human figure is not known a priori, this procedure is repeated for different

A. Esposito et al. (Eds.): Cognitive Behavioural Systems 2011, LNCS 7403, pp. 225–235, 2012.

scales. State-of-the-art examples of such classifiers are Histogram of Oriented Gradients (HoG) [6] or part-based models [8]. These classifiers are based on low level gradient features, similar to ones introduced in [14]. They are computed over small blocks of 16×16 pixels and classified using linear Support Vector Machine (SVM).

However, due to complexity of gradient features and large number of windows that needs to be evaluated, one of the main problems of these classifiers is their computational complexity. For example, processing a 640×425 grayscale image with Felzenswalb's detector [8] takes roughly 7 seconds on a 3 GHz Core Duo machine with 8 GB RAM.

We aim our work at reducing the search space that needs to be evaluated by sliding window based classifiers and therefore speeding-up the detection process. In other words, we deal with the problem of candidate window generation for person detection, i.e. we want to discard as many false positive windows and keep as many true positive ones as possible by utlizing less costly algorithms than the ones used during the final classification.

We base our work on the paper on candidate window generation by Alexe et al. [1], which deals with generic object detection. Alexe et al. propose a framework with a set of four cues and a window sampling procedure, which serves as a preprocessing step for the robust state-of-the-art classifiers. Cues of the framework are designed such that they favor regions that are likely to contain an object of any class. They claim their framework to speed-up several state-of-the-art object detectors [6,8] by up to 20-40 times by reducing the total number of windows that needs to be evaluated. In this paper, we evalute this framework in the context of person detection, modify if and extend it by considering another set of cues, specifically suited for person detection task.

The structure of this report is as follows. Sect. 2 describes the pre-attention cues and their fusion, Sect. 3 then evaluates and discusses achieved results. Finally, in Sect. 4 we present conclusions and propose future extensions.

2 Description of the Pre-attention Cues

We first review the cues proposed in [1] and then we propose additional cues suited for person detection task.

2.1 Objectness Cues

Multiscale Saliency. Multiscale Saliency (MS) is based on spectral residual approach of Hou et al. [12], which has high response for regions that are unique in terms of appearance within an image. Typical examples of such regions are object on uniform background. The image f is first down-sampled to some predefined size $s_0 \times s_0$. The saliency map annotated $I^{s_0}_{MS}$ is computed by inverse FFT of the residual of original and smoothed log-spectrums of the image. In order to extract objects at different scales MS repeats this procedure for several image

sizes s and for each of them a saliency map I^s_{MS} is obtained. MS score of window r is then computed as

$$MS(r, \theta^s_{MS}) = \sum_{\mathcal{P}} I^s_{MS}(p) \times \frac{|p \in r \mid I^s_{MS}(p) \geq \theta_s|}{|r|} \tag{1}$$

where θ^s_{MS} are free threshold parameters for each of the scale s and $\mathcal{P} = \{p \in r \mid I^s_{MS}(p) \geq \theta_s\}$. In order to extract regions of different sizes, MS is computed for every $m' \times n'$ sub-window r of the resulting saliency maps, where $m', n' = 1, \ldots, s$.

Color Contrast. Color Contrast (CC) measures color dissimilarity between a region and its surroundings. Surroundings $\mathrm{Surr}(r, k_{CC})$ of a window r is defined as a rectangular ring obtained by scaling the window r proportionally by factor k_{CC} in and subtracting the area of window r. The dissimilarity is computed as Chi-square distance of color histograms of the window and its surroundings, i.e.

$$CC(r, k_{CC}) = \chi^2 \left(h(r), h\left(\mathrm{Surr}(r, k_{CC})\right)\right) \tag{2}$$

To compute the histograms $h(r)$ and $h\left(\mathrm{Surr}(r, k_{CC})\right)$ of the window r and its surroundings $\mathrm{Surr}(r, k_{CC})$, image is converted to Lab space and then quantized into l color levels. Note that with relatively small number of quantized colors the histograms can be computed in constant time by a table look-up using the summed area tables (integral images) trick.

Edge Density. Edge Density (ED) captures the fact that images of objects usually have well defined borders while they do not have many edge pixels inside. The ED is computed as a density of edge pixels near the window borders, i.e.

$$ED(r, k_{ED}) = \frac{\sum_{p \in \mathrm{Inn}(r, k_{ED})} I_{ED}(p)}{\mathrm{Len}(\mathrm{Inn}(r, k_{ED}))}, \tag{3}$$

where $\mathrm{Inn}(r, k_{ED})$ is the inner ring of window r obtained by shrinking it by factor k_{ED} (similarly to CC) and $\mathrm{Len}(\cdot)$ is its perimeter. Edge pixels are obtained by Canny edge detector [3].

Superpixels Straddling. Similarly to ED, Superpixels Straddling (SS) cue captures the closed boundary characteristics of an object. It is based on superpixel segmentation [9], which segments the image into small regions of uniform color. After segmentation, surfaces of objects consist of several superpixels, which preserve their boundaries. The SS measures the extent to which the superpixels straddle the test window r. A superpixel is straddling a window r if it contains at least one pixel inside and one pixel outside r. The degree, by which a superpixel straddles window r, is defined as the minimum of the number of its pixels inside r and the number of its pixels outside r. SS score of window is then computed as a sum of degrees of straddling for all superpixels contained in r, i.e.

$$SS(r, \theta_{SS}) = 1 - \sum_{s \in Sl(\theta_{SS})} \frac{\min\left(|s \backslash r|, |s \cap r|\right)}{|r|}, \tag{4}$$

where θ_{SS} is a segmentation scale.

2.2 Proposed Cues

Since in our task we want to find candidate windows which might contain persons rather than just generic objects, we might take advantage of some specific features. As persons in images from video surveillance or driver assistance systems are usually in upright positions, we mainly try to explore the vertical symmetry and edge properties of candidate regions.

Color Symmetry. Color Symmetry (CS) cue is based on comparing the color distributions of the inner and outer parts of a test window. Ideally, the windows containing persons should be vertically symmetrical in terms of color, i.e. the left and right part of the bounding box should have similar color distribution, whereas the person itself should be distinctive from its local neighborhood. Each window r of size $m \times n$ is divided into four parts in the direction of x-axis: left half $r_L(r)$, right half $r_R(r)$, inner rectangle $r_I(r)$ and outer part $r_O(r)$. The inner rectangle has size $m \times n/2$ pixels and is positioned in the center of the window r. The outer rectangle covers the rest of the window r. The CS score is then computed as

$$CS(r) = \frac{\chi^2\left(h_I(r), h_O(r)\right)}{\chi^2\left(h_L(r), h_R(r)\right) + \epsilon}, \tag{5}$$

where $\chi^2\left(h_x(r), h_y(r)\right)$ is the Chi-squared distance of color histogram of the rectangles $r_x(r)$ and $r_y(r)$ and ϵ is a smoothing parameter. In order to compute the histograms efficiently and avoid additional computation demands, we use the same approach for quantization of the colors as in the case of Color Contrast cue, that is the image is first converted into Lab color space and then quantized into l color levels.

Edge Symmetry. Edge Symmetry (ES) is another symmetry-based cue. Similarly to [16], it exploits the fact that person often appears in an image as either bright or dark blob, which is roughly symmetrical. As mentioned in [2], the left and right parts of the window should contain similar amount of edges, but their sign in the direction of x-axis should be opposite. We therefore vertically divide the window r into k rectangular areas $r_{1\ldots i\ldots k}(r)$ of size $\left\lfloor \frac{n}{k} \right\rfloor \times m$ and on each of them we compute the total sum $s_i^L(r)$ and $s_i^R(r)$ of their left and right edge pixels. We call an edge pixel $p = (x, y)$ left if $I(x-1, y) < I(x+1, y)$, i.e. it lies on a transition from dark to bright region. ES score of window r is computed as

$$ES(r) = \sum_{i=1}^{k} \left(s_i^L(r) - \overline{s^L(r)}\right) \cdot \left(s_{k-i+1}^R(r) - \overline{s^R(r)}\right), \tag{6}$$

We subtract the mean values $\overline{s^L(r)}$ and $\overline{s^R(r)}$ from $s_i^L(r)$ and $s_i^R(r)$ in order to suppress the score of windows with many randomly distributed edges. Similarly to ED, edges are obtained using Canny edge detector.

Verticality. Usually images of persons contain much more vertical edges than edges of other directions. Verticality (VE) tries to use this property by computing the relative amount of vertically oriented edge pixels in a window r. It is similar to Edge orientation histogram descriptor (EOH) [13], but it differs in several aspects. The number of orientation bins b_i is fixed to 4, for better robustness the edges are detected using Canny edge detector rather than simple thresholding of the convolution of the image with Sobel kernel and it considers only the vertical orientation bin value. It is computed as

$$VE(r) = \frac{E_3(r)}{\sum_i E_i(r)}, \tag{7}$$

where

$$E_i(r) = \sum_{p \in r} \psi_i(p) \tag{8}$$

are sums of edge pixels p, which have orientation ψ_i, i.e.

$$\psi_i(p) = \begin{cases} 1 & \text{if} \theta(p) \in b_i \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

Thus, $E_3(r)$ corresponds to the vertical bin. Note that similarly to the CC and CS cues the values of $E_i(r)$ can be computed rapidly by computing summed area table for each of the four orientations.

Dominant Orientation. Since VE considers only the value of the vertically oriented bin, the natural question that arises is whether it would be beneficial to utilize the values of the other bins too. We construct a normalized window descriptor $E^n(r) = [E_1^n, \ldots, E_4^n]^\top$, where

$$E_i^n = \frac{E_i(r)}{\sum_j E_j(r)}, \tag{10}$$

and classify the window using linear classifier w_{DO}. The score is then computed as

$$DO(r) = w^\top E(r) + \beta \tag{11}$$

In the case of VE score computation 7 the classifier corresponds to $w = [0, 0, 1, 0]^\top$ and $\beta = 0$, i.e. only the relative value of the vertically oriented bin E_3 8 is considered. In DO 2.2 the coefficients of the classifier w and β are found by training a linear SVM [5] on a set of positive and negative window samples. We use libSVM [4] for SVM classification.

2.3 Cue Combination

In order to take advantage of all information available, cues can be combined into a single classifier. The score of each cue is computed for every position of the sliding window, which has a fixed size. Different scales are processed by recursively reducing the size of the image. Note that this approach is different from that in [1], where the cues are only computed for windows sampled from MS score distribution.

Naïve Bayesian classifier is used to compute the final score of a test window. The score $p(\mathrm{p}|\mathcal{A})$ of set of cues $\mathcal{A}$ is computed as

$$p\left(\mathrm{p}|\mathcal{A}\right) = \frac{p\left(\mathrm{p}\right)\prod_{c\in\mathcal{A}} p_c\left(x_c|\mathrm{p}\right)}{p\left(\mathrm{p}\right)\prod_{c\in\mathcal{A}} p_c\left(x_c|\mathrm{p}\right) + p\left(\bar{\mathrm{p}}\right)\prod_{c\in\mathcal{A}} p_c\left(x_c|\bar{\mathrm{p}}\right)}, \tag{12}$$

where $p(\mathrm{p})$ and $p(\bar{\mathrm{p}})$ are prior probabilities of finding a person and background, respectively, $p_c\left(x_c|\mathrm{p}\right)$ is the probability of the score x_c respective to the score distribution $p_c\left(\mathrm{p}\right)$ of the cue c. Rather than assuming a continuous probability density such as normal distribution, the probability distributions p_{p} and $p_{\bar{\mathrm{p}}}$ are modeled as histograms, which are computed over a training dataset.

3 Experiments

We evaluate the cues on INRIA person dataset[1] as it is standardized dataset containing persons in different contexts, scales and image quality. The free parameters of the cues were trained on the training subset of the INRIA person database containing 2416 images. The test subset comprised of 288 annotated images containing total of 589 persons (number of persons in one image varies from 1 to 16).

Fig. 1. Example images from INRIA person dataset

[1] `http://pascal.inrialpes.fr/data/human/`

3.1 Evaluation of Cues

Cues are evaluated based on the pyramid scheme. Each image is first proportionally resized to the maximum allowed size of 640×480 pixels for efficiency reasons. Then, an image pyramid is built by recursively reducing the size of the image using bilinear interpolation. We use $L = 18$ pyramid levels with scale factor $\kappa = 2^{-1/6}$. Search window size is fixed to 20×50 pixels for all pyramid layers.

For every cue true positive ratio (TPR) vs. false positive ratio (FPR) curve (ROC) is plotted, see Fig. 2 and Fig. 3. We consider a test window positive if it is covering any of the ground truth windows of the image. The window r is covering the window o if the Pascal criterion

$$PC(r, o) = |r \cap o| \,/\, |r \cup o| \tag{13}$$

is larger than some threshold t_{PC}, i.e. $PC(r, o) > t_{PC}$. In our case, we use $t_{PC} = 0.7$. Thus, TPR is a ratio of correctly classified positive windows and the total number of all positive windows. Note that using this definition of positive and negative samples has two advantages. The first advantage is the ability to generate nearly arbitrary number of training samples. The second advantage is that the training samples are similar to the input in the testing phase, i.e. due to sparse sampling, the sliding window is often poorly aligned to ideal position over an image of person.

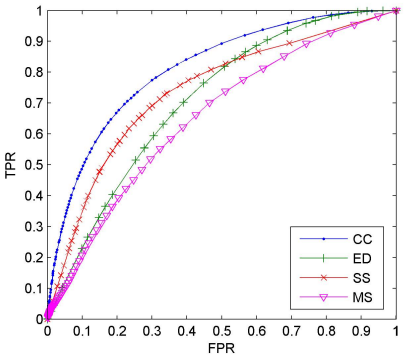


Fig. 2. ROC of the cues proposed in [1]

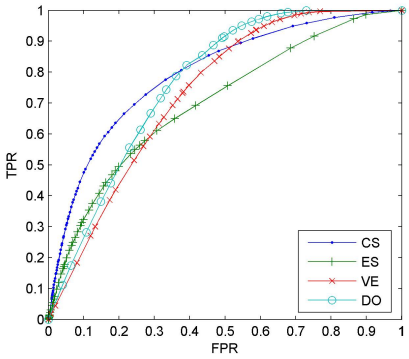
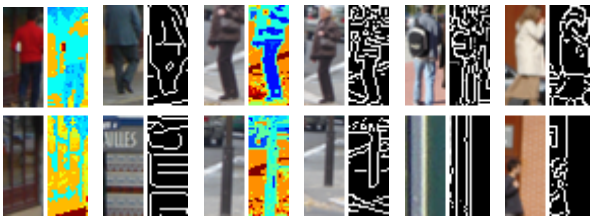


Fig. 3. ROC of our proposed cues

The obtained results for individual cues are shown in Tab. 1. For each cue, the FPR is false positive ratio when 95% of true positive windows are preserved.

The edge-based cues, mainly VE and DO, performed very well in our tests. Their performance depends on the ability to extract edges from an image reliably. They are both global features in terms of the test window, which makes them robust to noise and missing edge information. Also, they are both very fast to evaluate, since they can be computed using the summed area table trick [17].



(a) (b) (c) (d) (e) (f)

Fig. 4. True positive (top) and false positive (bottom) examples: a) CC, b) ED, c) CS, d) ES, e) VE, f) DO

Symmetry-based cues work best for test windows that are precisely aligned to some ground truth window. Both ES and CS are sensitive to misalignments of the test and ground truth windows. This can be overcome by using finer scale-factor κ in order to cover the ground truth windows by more percent in terms of the Pascal criterion. However, the obvious problem with this approach is an increase of computational costs.

As opposed to [1], SS cue did not perform best in our test. The main problem of the superpixel segmentation is its sensitivity to blur and other image quality degradations, which results in violation of the key assumption of SS that superpixels preserve object boundaries. While decreasing the value of segmentation scale θ_{SS} increases the overall recognition rate, it also increases the computational complexity because of the large number of extracted regions that needs to be evaluated for every window.

We also found that the Multiscale Saliency is not well suited for detecting multiple objects within a single image. This is due to its spectral residual approach, which only favors regions with unique appearance, not repetitive patterns.

The examples of false and true positive windows as classified by the cues are shown in Fig. 4.

Table 1. Results for Pascal criterion threshold 0.7 and TPR = 0.95

Cue	MS	CC	ED	SS	CS	ES	VE	DO
FPR	0.89	0.66	0.72	0.85	0.69	0.82	0.61	0.55

3.2 Cue Combination

We evaluated all 127 possible cue combinations. The results for selected combinations as well as for original objectness measure from [1] are shown in Fig. 5. One of the best result was obtained using only three cues: Color Contrast, Color Symmetry and Dominant Orientation. The achieved false positive rate at 0.95 true positive rate was 0.33. The false positive rate at the same true positive rate

for general objectness measure was 0.50. When the computationally inefficient SS cue was not taken into account, the objectness false positive rate was 0.63, almost twice as large as the best case. The improvement of our extended set over the general objectness measure is caused by two reasons: utilization of person specific charateritics and poor performance of Multiscale Saliency for cluttered scenes or images with multiple persons.

Adding another cues did not lead to significantly better performance in our experiments. This can be explained by the fact that cues that are based on the same type of information (e.g. color) are quite correlated. As one can expect, the most correlated pair of cues with normalized correlation coefficient $\rho = 0.76$ are VE and DO, which differ only in classification of the edge histogram.

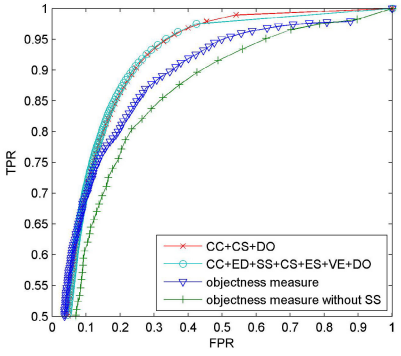


Fig. 5. ROC curves for selected cue combinations

3.3 Time Consumption

We also evaluated time consumption of the cues. All experiments were performed on PC with Core 2 Duo @ 3GHz processor and 8 GB RAM. We implemented the algorithms in Matlab and C++ using OpenCV[2] library for image processing tasks and the HoG detector. We use Felnzenswalb's code[3] for the segmentation for SS cue. Since each group of cues (e.g. color-based) performs similar preprocessing steps such as color quantization or edge detecting, we measure the time cost of the preprocessing and the window score computation separately. The results are shown in Tab. 2 and Tab. 3

Longer computation times of color-based compared to edge-based cues are caused by relatively large number of integral histograms, which have to be computed for every quantized color. Similar problem causes extremely long computation times for SS cue, where the integral histogram is computed for each superpixel. Depending on the segmentation scale θ_{SS}, there can be up to hundreds of superpixels in a cluttered scene. We also evaluated our candidate window

[2] `http://opencv.willowgarage.com/wiki/`
[3] `http://www.cs.brown.edu/~pff/segment/`

Table 2. Preprocessing

Color-based cues	331 ms
Edge-based cues	46 ms
Superpixels	7617 ms

Table 3. Score computation

MS	35 ms	CS	461 ms
CC	640 ms	ES	113 ms
ED	133 ms	VE	82 ms
SS	11255 ms	DO	82 ms

generation algorithm together with the HoG detector [6]. With exhaustive search on a dense grid, 83% detection rate was obtained with the HoG detector, while processing the cca 0.8 MPix images took 46 seconds on average. When using the pre-attention phase with CC, CS and DO cues and thresholding 12, the overall achieved detection rate was 69% with the average of 4 seconds per image. In other words, we achieved a speed-up factor of 11.5 while preserving 84% of the true positives.

4 Conclusion

We have evaluated several pre-attention cues for person detection that can be used to reduce the search space of arbitrary sliding window based detector. We have shown that cues proposed in [1] do not perform well in the task of person detection, especially in the cases where image contains cluttered background, multiple objects of interest or blur. In order to solve these problems, we have proposed additional cues specifically suited for person detection. Also, Tab. 1 shows better efficiency of the proposed cues. In our experiments, cues utilizing edge orientation properties achieved the lowest false positive rate. They also are more efficient than color-based cues, which rely on computing color histograms. Symmetry-based cues performed well only when combined with other type of features. Superpixel Straddling cue did not perform well in our experiments, both in false positive rates and efficiency. As a next development we will investigate how to optimally combine the cues into more efficient classifier, possibly using multiple stages of classification.

Acknowledgments. The research reported in this paper was partly supported by Student Grant Scheme at Technical University of Liberec.

References

1. Alexe, B., Deselaers, T., Ferrari, V.: What is an object? In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2010), pp. 73–80 (2010)
2. Bertozzi, M., Broggi, A., Del Rose, M., Felisa, M.: A symmetry-based validator and refinement system for pedestrian detection in far infrared images. In: Intelligent Transportation Systems Conference, pp. 155–160. IEEE (2007)
3. Canny, J.: A computational approach to edge detection. IEEE Trans. Pattern Anal. Mach. Intell. 8, 679–698 (1986)

4. Chang, C.C., Lin, C.J.: LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology 2, 27:1–27:27 (2011), software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`
5. Cortes, C., Vapnik, V.: Support-vector networks. Machine Learning 20, 273–297 (1995), doi:10.1007/BF00994018
6. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), vol. 1, pp. 886–893. IEEE Computer Society (2005)
7. Enzweiler, M., Gavrila, D.M.: Monocular Pedestrian Detection: Survey and Experiments. IEEE Transactions on Pattern Analysis and Machine Intelligence 31(12), 2179–2195 (2009)
8. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part-based models. IEEE Transactions on Pattern Analysis and Machine Intelligence 32, 1627–1645 (2010)
9. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient graph-based image segmentation. Int. J. Comput. Vision 59, 167–181 (2004)
10. Fong, T., Nourbakhsh, I., Dautenhahn, K.: A survey of socially interactive robots (2003)
11. Gerónimo, D., López, A.M., Sappa, A.D., Graf, T.: Survey of Pedestrian Detection for Advanced Driver Assistance Systems. IEEE Transactions on Pattern Analysis and Machine Intelligence 32(7), 1239–1258 (2010)
12. Hou, X., Zhang, L.: Saliency detection: A spectral residual approach. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2007, pp. 1–8 (June 2007)
13. Levi, K., Weiss, Y.: Learning object detection from a small number of examples: the importance of good features. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2004), pp. 53–60. IEEE Computer Society (2004)
14. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60, 91–110 (2004)
15. Ogale, N.A.: A survey of techniques for human detection from video. Survey, University of Maryland (2006)
16. Schauland, S., Kummert, A., Park, S.B., Iurgel, U., Zhang, Y.: Vision-based pedestrian detection – improvement and verification of feature extraction methods and svm-based classification. In: Intelligent Transportation Systems Conference, ITSC 2006, pp. 97–102. IEEE (September 2006)
17. Viola, P.A., Jones, M.J.: Rapid object detection using a boosted cascade of simple features. In: CVPR, pp. 511–518 (2001)