Development of Algorithms and Software for Automatic Pedestrian Recognition for Intelligent Situation Control Systems

N Abramov¹, V Rakova¹, D Barinov¹, A Leksashov² and E Pridanova¹

¹ Peter the Great St. Petersburg Polytechnic University St. Petersburg, Russia
nikolay.abramov@spbpu.com; valeriya.rakova@spbpu.com;
dmitriy.barinov@spbpu.com; ekaterina.pridanova@spbpu.com

²Tetracube LLC St. Petersburg, Russia
leksashov@mail.ru

Abstract. The article is devoted to the development of algorithms and software that automatically recognize road users - pedestrians, build their trajectory according to the analysis of the video stream. The architecture of the system for pedestrian recognition and inter-frame tracking is proposed. The developed algorithms can be used to estimate road situation with participation of pedestrians for the traffic monitor systems.

1. Introduction
Image processing algorithms have been in the focus of research in the field of intelligent transport systems [1, 2, 3,13] for a long time. Pedestrian detection and tracking tasks are those kinds of problems that can be solved using image processing algorithms. One of the active consumers of these algorithms are traffic monitor systems. These systems make it possible to observe whether drivers let pedestrians cross a road on regulated and unregulated pedestrian crossings equipped with such systems. Fig. 1 represents a typical image from one of those systems. However, this is not the only function of such systems, which also have to detect and track cars, determine their speed, form violation protocols, etc. That’s why in this case pedestrian detection algorithms are not only bound by strict requirements for accuracy of detection, but also have strict limitations on the quota of computing resources on the built-in computing platform, which already performs a range of tasks at the same time. Pedestrian detection is a specialized subfield of object detection, which is currently in demand in areas such as automotive, surveillance and security. Caltech Dataset [4] and Kitti [5] are up to date benchmarks for comparing the accuracy and processing time of the algorithms.

During last years diverse efforts have been made to improve the performance of pedestrian detection. Following the success of integral channel feature detector (ICF) [14], many variants [15, 16, 17] were proposed and showed significant improvement. From some recent works one can see that improved features have been driving performance and are likely to continue doing so. These features are often complemented by additional sources of information, such as optical flow [18], depth information [7], context information [19] and many others to even further boost detection accuracy.

Another class of classification algorithms – neural networks – have vastly advanced during recent years. By fine-tuning a model pre-trained on external data convolution neural networks (convolutional networks) have also reached state-of-the-art performance [20, 21].

Currently, neural networks occupy a leading position in recognition accuracy. However, the processing time of these algorithms takes from 0.1 s to several seconds [5] when executed on a graphics accelerator, which is unacceptable in our case. That’s why we focus on algorithms suitable for execution on embedded platforms with small computing capabilities.
Among such algorithms are Fastest pedestrian detection in the west [6], Pedestrian detection at 100 frames per second [7], C4: A Real-Time Object Detection Framework [8,9].

The authors of the Fastest Pedestrian Detection in the West present a method to significantly reduce execution time of multi-scale object detectors that use several types of features, including gradient histograms, with a slight decrease in the accuracy of pedestrian detection.

Two new algorithmic accelerations are presented in [7], one of which is based on improved scale processing (on monocular images), and the other on better use of depth information (on stereo images).

Article [9] makes two major contributions. First, the authors show that the outline defining the silhouette of a person is an important information for detection. Secondly, it is proposed to detect people using contour signals, using CENTRIST descriptor [10] for this purpose. Higher classification speed can be achieved due to the fact that one does not need to explicitly calculate CENTRIST descriptor, because it seamlessly embedded in the classifier estimation.

A distinctive feature of the first two approaches is that they offer more efficient methods of classifier response estimation in a multi-scale detection environment, i.e. when pedestrian size in image may vary greatly. However, in our case, the camera of a traffic monitor system is located at a distance of 10+ meters from the observed region (pedestrian crossing) and there is no need to use a large number of scales to detect pedestrians. Therefore, as a basis, we decided to use C4 pedestrian detection algorithm [9]. The following drawbacks were identified in this approach: it only uses feature that cover local region structure. In our work, we propose extensions to eliminate this drawback.

Thus, in this article, pedestrian detection algorithm for traffic monitor systems is proposed. In “Baseline classifier C4”, we describe C4 pedestrian detection algorithm, taken as a basis in our work. “Multi-scale CENTRIST descriptor” suggests classifier improvement, in the form of a multi-scale approach. Approach to training is presented in “Detection framework”. The testing conditions and the evaluation of the results are presented in “Results”.

2. Pedestrians detection algorithm for traffic monitor systems

2.1 Baseline classifier C4

C4 pedestrian detector feature calculation process is based on Census transform (CT) [11] which was originally intended to establish similarities between the local image regions. To calculate it one needs to find image gradient at each point of image first. To achieve this, Sobel operator is often used (1). Gradient values at pixel positions are calculated along the X and Y axis, and then combined together to obtain a single value. Thus, \( G_x \) is the image with gradient values along the X axis, \( G_y \) is the image with gradient values along the Y axis, \( G \) is the image with combined gradient values and \( I \) is the original image.

\[
G_x = \begin{bmatrix}
-1 & 0 & +1 \\
-2 & 0 & +2 \\
-1 & 0 & +1 \\
\end{bmatrix} \ast I
\]

\[
G_y = \begin{bmatrix}
-1 & -2 & -1 \\
0 & 0 & 0 \\
+1 & +2 & +1 \\
\end{bmatrix} \ast I
\]

\[
G = |G_x| + |G_y|
\]
Next, the gradient value of the center pixel is compared to the gradient values of eight neighboring pixels, as shown in Fig. 2. If the central pixel value is larger (or equal) to one of its neighbors, bit 1 is set at the appropriate location. Otherwise, bit 0 is set.

\[
\begin{array}{c|c|c}
32 & 64 & 96 \\
\hline
32 & 64 & 96 \\
\hline
32 & 64 & 96 \\
32 & 32 & 96 \\
\end{array}
\Rightarrow 1 \quad 0 \Rightarrow (11010110)_2 \Rightarrow \text{CT} = 214
\]

**Figure 2.** Example calculation of Census transform.

Eight bits generated from the intensity comparisons can be combined in any order to obtain the base 10 number. This is the CT value for the center pixel. The CENTRIST descriptor is a histogram of these CT values [10]. As shown in Fig. 2, the CT values briefly encode the signs of comparison between neighboring pixels. However, the CENTRIST descriptor lacks the ability to describe global (or larger) structures and contours beyond a small 3 × 3 range.

CENTRIST has some close relationship with local binary pattern, another feature used for human detection. One important difference is how the LBP values are utilized. Pedestrian detection methods use “uniform LBP”, in which certain LBP values that are called “non-uniform” are lumped together. These procedures make their descriptors to only encode a blurred version of the most important information, i.e., signs of neighboring pixel comparisons.

### 2.2 Multi-scale CENTRIST descriptor

As described above, to obtain the CENTRIST descriptor, a Census transform histogram is computed within the super block with a size of 24 x 16 pixels. Then these histograms are concatenated to form a final vector of features for the classifier. However, these histograms describe only their local area and are not able to represent a larger scale structure [28-31].

To overcome this disadvantage, we propose to use a multiscalar approach to calculate CENTRIST descriptor. To do this, one can take four neighboring superblocks, calculate the CENTRIST descriptor for each of them, and obtain a larger-scale descriptor by applying the max pooling operation. Let’s call CENTRIST of a single superblock a 1-st level descriptor. Second level CENTRIST descriptor combines adjacent lower level descriptors of the first level, and so on. Thus, the \(i\)-th element of the \(k + 1\) level CENTRIST descriptor located at the \((x, y) - C_{k+1}(x, y)\)” is obtained by applying maximum function to the \(i\)-th elements of \(k\) level descriptors.

\[
C_{k+1}(i) = \max\{C_k(x, y, 2^i(z, y)), C_k(x, y, 2^{i+1}(z, y))\}
\]  \hspace{1cm} (2)

Computation continues while candidate image region contains enough \(k\) level descriptors to form \(k + 1\) level descriptor. Visually, this principle is represented in Fig. 3. After all descriptors are calculated we concatenate them together to form a feature vector for the classifier.

It’s worth to note that Wu et al. already proposed a multi-scale approach for CENTRIST calculation [8]. However, it requires to use image pyramid to extract corresponding CENTRIST descriptors on each pyramid level. It means that we would need to calculate multiple integral images, which is acceptable when pedestrians may greatly vary in size, since we may reuse them from other detection scales. In our approach we expect to have small variation in pedestrian sizes and don’t plan to build integral images for several scales, because of complexity constraints. For the calculation of our multi-scale CENTRIST descriptor no additional image scales are required, since we can get all the required information from level 1 CENTRIST histograms.

Our approach allows to capture the structure of objects on a larger scale and improves the accuracy of pedestrian detection, which is presented in the next section [32, 33].

### 3. Detection framework

In the training phase, we have a set of 96 × 32 positive training image patches \(P\) and a set of larger negative images \(N\) that do not contain any pedestrians. We use block size of 12 x 8 pixels, which we found to show best detection results.

During training we follow the methodology of C4 original authors [9].
We first randomly choose a small set of patches from the images in \( N \) to form a negative training set \( N_1 \). Using \( P \) and \( N_1 \) we train a linear SVM classifier \( H_1 \).

![Diagram](image)

**Figure 3.** A bootstrap process is used to generate a new negative training set \( N_2 \). \( H_1 \) is applied to all patches in the images in \( N \). Multi-scale CENTRIST approach.

We then train \( H_2 \) using \( P \) and \( N_2 \). This process is repeated until all patches in \( N \) are classified as negative by at least one of previous level classifiers. We then train a linear SVM classifier using \( P \) and the combined negative set, which we call \( H_{\text{lin}} \).

To goal of the linear classifier is to preserve as much pedestrians as possible (i.e. high recall) and to filter out considerable amount image regions that don’t contain pedestrians. To obtain higher detection accuracy we train a second \( H_{\text{hik}} \) classifier. We use \( H_{\text{lin}} \) on \( N \) to bootstrap a new negative training set \( N_{\text{final}} \) and train an SVM classifier using the libHIK HIK SVM solver [8]. In the testing and detection phase, a cascade with two nodes \( H_{\text{lin}} \) and \( H_{\text{hik}} \) is used.

### 4. Results

The classifier is written in C ++ language, the learning was carried out according to the procedure described in Sect. III, using the Liblinear [12] and libHIK [8] libraries. For the learning and evaluation of results, the sampling of records from the traffic monitor system cameras was used, containing more than 200,000 positive examples with pedestrians [25-27].

The testing was performed on the Intel Nano-6060 hardware platform with a 4-core Atom E-3845 processor, clocked at 1.95 GHz. Classifier evaluation was carried out on a CPU, using singe core to simulate real-world situation, where most resources would be occupied by other tasks. As an input data we used grayscale images with a resolution of 1280x960 pixels with a region of interest specifying the position of expected pedestrian crossing.

**Table 1.** Classifier evaluation metrics.

| Algorithm   | True positive rate | False positive rate | Time per frame, ms |
|-------------|--------------------|---------------------|--------------------|
| C4 baseline | 0.974              | 0.085               | 25                 |
| C4 ours     | 0.985              | 0.054               | 28                 |

Test results show that the proposed approach slightly increases the recall of pedestrians and reduced the number of false positives by 37%. Due to the use of the hierarchical multi-scale approach, execution time slightly increased by 12%.

After analysis of Table 1 results, we came the following conclusion - the majority of false positives and false negatives in detection results related to the difficult visibility conditions such as night environment, heavy rain, fog, snow and other. In the future, we plan to adapt the algorithm to improve its efficiency in such conditions [22, 23].

### 5. Conclusion

In this article we presented a way to improve the accuracy of the C4 pedestrian recognition algorithm by using a multi-scale approach. We demonstrated that this modification allowed to increase the coverage of this classifier and reduced the number of false positives. This modification allows using this algorithm to recognize pedestrians in automatic traffic fixing systems built on the basis of low-
power embedded systems. A further development direction is to increase the accuracy of the classifier in poor visibility conditions and to develop a tracking system for pedestrians.

Acknowledgements
We are grateful for the support of the “Olvia” LLC for providing numerous pedestrian samples in real-world city environment. The research carried out with the financial support of the grant from the Program Competitiveness Enhancement of Peter the Great St.Petersburg Polytechnic University, Project 5-100-2020.

References
[1] P. Dollar, C. Wojek, B. Schiele, and P. Perona, “Pedestrian Detection: An Evaluation of the State of the Art,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 34, no. 4, pp. 743–761, Apr. 2012.
[2] R. Benenson, M. Omran, J. Hosang, and B. Schiele, “Ten Years of Pedestrian Detection, What Have We Learned?,” in Computer Vision - ECCV 2014 Workshops, vol. 8926, L. Agapito, M. M. Bronstein, and C. Rother, Eds. Cham: Springer International Publishing, 2015, pp. 613–627.
[3] S. Zhang, R. Benenson, M. Omran, J. Hosang, and B. Schiele, “How Far are We from Solving Pedestrian Detection?,” 2016, pp. 1259–1267.
[4] Caltech pedestrian dataset [Electronic resource: http://www.vision.caltech.edu/Image_Datasets/CaltechPedestrians/] (Date of the application: 24.04.2018)
[5] Kitti object detection benchmark [Electronic resource: http://www.cvlibs.net/datasets/kitti/eval_object.php] (Date of the application: 24.04.2018)
[6] P. Dollar, S. Belongie, and P. Perona, “The Fastest Pedestrian Detector in the West,” 2010, pp. 68.1–68.11.
[7] R. Benenson, M. Mathias, R. Timofte, and L. Van Gool, “Pedestrian detection at 100 frames per second,” 2012, pp. 2903–2910.
[8] Jianxin Wu, Nini Liu, C. Geyer, and J. M. Rehg, “C4: A Real-Time Object Detection Framework,” IEEE Transactions on Image Processing, vol. 22, no. 10, pp. 4096–4107, Oct. 2013.
[9] J. Wu, C. Geyer, and J. M. Rehg, “Real-time human detection using contour cues,” 2011, pp. 860–867.
[10] Jianxin Wu and J. M. Rehg, “CENTRIST: A Visual Descriptor for Scene Categorization,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 33, no. 8, pp. 1489–1501, Aug. 2011.
[11] R. Zabih and J. Woodfill, “Non-parametric local transforms for computing visual correspondence,” in Computer Vision — ECCV’94, vol. 801, J.-O. Eklundh, Ed. Berlin/Heidelberg: Springer-Verlag, 1994, pp. 151–158.
[12] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, “LIBLINEAR: A Library for Large Linear Classification,” p. 30.
[13] M. Enzweiler and D. M. Gavrila, “Monocular Pedestrian Detection: Survey and Experiments,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 31, no. 12, pp. 2179–2195, Dec. 2009.
[14] P. Dollar, Z. Tu, P. Perona, and S. Belongie, “Integral Channel Features,” 2009, pp. 91.1–91.11.
[15] W. Nam, P. Dollar, and J. H. Han, “Local Decorrelation For Improved Pedestrian Detection,” p. 9.
[16] S. Zhang, R. Benenson, and B. Schiele. Filtered channel features for pedestrian detection. In CVPR, 2015.
[17] S. Zhang, C. Bauckhage, and A. B. Cremers. Informed haar-like features improve pedestrian detection. In CVPR, 2014.
[18] D. Park, C. L. Zitnick, D. Ramanan, and P. Dollár. Exploring weak stabilization for motion feature extraction. In CVPR, 2013.
[19] W. Ouyang and X. Wang. Single-pedestrian detection aided by multi-pedestrian detection. In CVPR, 2013.
[20] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards Real-Time Object Detection
with Region Proposal Networks,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[21] F. Yang, W. Choi, and Y. Lin, “Exploit All the Layers: Fast and Accurate CNN Object Detector with Scale Dependent Pooling and Cascaded Rejection Classifiers,” 2016, pp. 2129–2137.

[22] N.A. Abramov, M.V. Bolsunovskaya, A.V. Leksashov, D.S.Barinov, “Algorithms for detecting and tracking of objects with optical markers in 3D space”, Proceedings of the 19th International Conference on Soft Computing and Measurements, SCM 2016; Saint Petersburg Electrotechnical University (SPbETU) "LETI", Saint Petersburg, Russian Federation; 25-26 May 2016; Editor Shaposhnikov S.; Institute of Electrical and Electronics Engineers Inc., 2016. Article # 7519782, PP. 362-364, DOI: 10.1109/SCM.2016.7519782)

[23] A.S. Makarov, M.V.Bolsunovskaya, “The 360° around view system for large vehicles, the methods of calibration & removal of barrel distortion for omnidirectional cameras”, CEUR Workshop Proceedings. 5th International Conference on Analysis of Images, Social Networks and Texts, AIST 2016; Yekaterinburg; Russian Federation; 6-8 April 2016,. CEUR-WS, 2016. Volume 1710. PP.182-190 (ISSN: 16130073)

[24] V.N. Volkova, A.V. Loginova, S.V. Shirokova, E.A. Kozlovskay, “Development of the innovative IT-project and managing project human resources”, Proceedings of the 19th International Conference on Soft Computing and Measurements, SCM 2016, Saint Petersburg Electrotechnical University (SPbETU) "LETI", Saint Petersburg; Russian Federation; 25-27 May 2016; 2016. Article # 7519816, PP. 470-473, DOI: 10.1109/SCM.2016.7519816)

[25] Fisher, W. D., Camp, T. K., & Krzhizhanovskaya, V. V. (2016). Crack detection in earth dam and levee passive seismic data using support vector machines. Paper presented at the Procedia Computer Science, , 80 577-586. doi:10.1016/j.procs.2016.05.339

[26] Gelgor, A., Gorlov, A., & Nguyen, V. P. (2018). Performance analysis of SEFDM with optimal subcarriers spectrum shapes. Paper presented at the 2017 IEEE International Black Sea Conference on Communications and Networking, BlackSeaCom 2017, , 2018-January 1-5. doi:10.1109/BlackSeaCom.2017.8277680

[27] Gelgor, A., Gorlov, A., & Nguyen, V. P. (2016). The design and performance of SEFDM with the sinc-to-RRC modification of subcarriers spectrums. Paper presented at the International Conference on Advanced Technologies for Communications, 65-69. doi:10.1109/ATC.2016.7764831

[28] Inochkin, F. M., Kruglov, S. K., Bronshtein, I. G., Kompan, T. A., Kondratiev, S. V., Korenev, A. S., & Puhov, N. F. (2018). Superresolution contour reconstruction approach to a linear thermal expansion measurement. Paper presented at the Proceedings - International Conference on Image Processing, ICIP, 3843-3847. doi:10.1109/ICIP.2018.8451081

[29] Liokumovich, L., Medvedev, A., Muravyov, K., Skliarov, P., & Ushakov, N. (2017). Signal detection algorithms for interferometric sensors with harmonic phase modulation: Distortion analysis and suppression. Applied Optics, 56(28), 7960-7968. doi:10.1364/AO.56.007960

[30] Lutin, V. I., Desyatirikova, E. N., Kuripta, O. V., Akamsina, N. V., & Mager, V. E. (2018). The processing of signals from sensors to observe objects in various physical fields. Paper presented at the Proceedings of the 2018 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering, EiConRus 2018, , 2018-January 1113-1116. doi:10.1109/EIConRus.2018.8317285

[31] Plotnikov, A., & Gelgor, A. (2018). Spectral efficiency comparison between FTN signaling and optimal PR signaling for low complexity detection algorithm doi:10.1007/978-3-030-01168-0_18

[32] Tsikin, I., & Melikhova, A. (2016). Angle-of-arrival GPS integrity monitoring insensitive to satellite constellation geometry doi:10.1007/978-3-319-46301-8_49