FPGA-GPU Architecture for Kernel SVM Pedestrian Detection

Sebastian Bauer[1], Sebastian Köhler[2], Konrad Doll[3], Ulrich Brunsmann[2]

[1]Pattern Recognition Lab, Department of Computer Science
University Erlangen-Nuremberg, Germany
[2]Laboratory for Pattern Recognition and Computational Intelligence
[3]Laboratory for Computer-Aided Circuit Design
University of Applied Sciences Aschaffenburg, Germany
sebastian.bauer@informatik.uni-erlangen.de
{sebastian.koehler, konrad.doll, ulrich.brunsmann}@h-ab.de

Abstract

We present a real-time multi-sensor architecture for video-based pedestrian detection used within a road side unit for intersection assistance. The entire system is implemented on available PC hardware, combining a frame grabber board with embedded FPGA and a graphics card into a powerful processing network. Giving classification performance top priority, we use HOG descriptors with a Gaussian kernel support vector machine. In order to achieve real-time performance, we propose a hardware architecture that incorporates FPGA-based feature extraction and GPU-based classification. The FPGA-GPU pipeline is managed by a multi-core CPU that further performs sensor data fusion. Evaluation on the INRIA benchmark database and an experimental study on a real-world intersection using multi-spectral hypothesis generation confirm state-of-the-art classification and real-time performance.

1. Introduction

Statistics reveal that on a global scale the majority of road traffic related deaths are among vulnerable road users (VRUs) covering pedestrians, bicyclists and motorcyclists. Even in Europe and the U.S., which rank first in road infrastructure safety, this group of traffic participants account for 43% and 28%, respectively, of all road accident fatalities [2, 3]. Hence, VRU detection is becoming an integral part of future advanced driver assistance systems (ADAS). Based on a prediction of collision risk, such a system could issue a warning to the driver and perform autonomous braking or maneuvering for collision avoidance or activate VRU impact mitigation devices to reduce severe injuries in case of an imminent collision.

Vehicle-based embedded detection systems [31] are one component of the solution, but visibility from the driver's perspective is limited. Infrastructure-based road side units (RSUs) could complement vehicle-based sensors via wireless communication in order to increase the virtual perception diversity and range and thus provide a better foundation for improved driver reaction. In particular, intersections demand a high level of concentration. Within a fraction of a second, the driver needs to filter out irrelevant information, draw conclusions and react in time. Several research programs address these topics. For example, the U.S. IntelliDrive program [7], the Japanese Assistance For Safe Driving Development area of ITS [5], the European INTERSAFE projects [4], the German Ko-PER sub-program of the Ko-FAS research initiative [6] and the French PUVAME project [8], all include detection of VRUs with infrastructure-based units that support the driver in coping with line-of-sight obstructions and stimulus overflow when approaching an intersection or precarious traffic locations.

2. Related Work

Human detection is an inherently complex problem due to the variability in appearance (body articulation, clothing, occlusions, environmental conditions). At the same time, it is a key component in many applications like surveillance, robotics and intelligent vehicles. Hence, in the past decade, a variety of approaches in terms of system architecture, image descriptors and classification schemes have been proposed in this domain, e.g. [20, 27]. Recent comparative studies like [18] indicate that appearance-based methods seem more promising. Experimental studies have shown that histograms of oriented gradients (HOG) descriptors are the leading edge in terms of classification performance. Dollar *et al.* benchmarked sliding-window based pedes-

978-1-4244-7030-3/10/$26.00 ©2010 IEEE

trian detectors [15], while Enzweiler and Gavrila compared appearance-based approaches with their shape-texture detection system [16, 17]. Both groups of authors conclude that HOG outperforms other features in most ADAS-related scenarios. Furthermore, the basic descriptor introduced by Dalal and Triggs [14] evokes moderate computational costs compared to other features surveyed in the benchmarks. Real-time requirements have spurred the development of different hardware accelerated HOG implementations, e.g. [9, 10, 23]. Regarding the classifier itself, Dalal and Triggs propose support vector machines for HOG classification. Their experiments show that using a Gaussian kernel SVM instead of a linear one increases performance significantly at the cost of much higher run time [14]. As a result, authors that have proposed real-time HOG detection systems chose linear SVMs on FPGA [19] or GPU hardware [25, 28, 29] or different classifiers such as AdaBoost [32]. However, early rejection by one of the weak classifiers often results in a classification performance loss. Intersection kernel SVMs have shown to outperform linear SVMs in terms of classification performance at the same order of computational cost for evaluation of the similarity between histograms describing the features [21].

It is generally expected that only a combination of complementary sensors will meet the requirements of real-world applications. Best performance can be obtained when using the top-of-the-line sensors in combination with the top-of-the-line features and classifier.

In this paper, we present a real-time pedestrian detection system based on a multi-sensor platform for intersection assistance. Fusion of imaging data in the visible and far infrared range (VR, FIR) is employed for hypothesis generation. Then based on the HOG descriptor, a sliding-window framework evaluates pedestrian candidates with a support vector machine (SVM) classifier. In contrast to previous hardware-accelerated approaches, we employ a Gaussian kernel SVM, placing great importance on classification performance and flexibility with regard to descriptor representations. In order to achieve real-time processing while maintaining the superior classification performance with the combination of HOG descriptors and a kernel SVM, we introduce a hardware architecture based on FPGA, CPU and GPU that is implemented on commercially available standard PC hardware components. Arranged in a pipeline, feature extraction is performed on a low-cost FPGA of the frame grabber, classification on the GPU of the graphics card. A multi-core CPU builds the core of the processor network.

We consider the proposed system design of a distributed network of parallel hardware architectures as a generic and flexible framework for a variety of applications in the domain of computer vision and pattern recognition beyond pedestrian detection. Depending on the individual method to be incorporated into the system, the appropriate device can be chosen with respect to the implementation strategy and hardware constraints. The fact, that all system components are available as PC hardware including a graphics-oriented software tool that facilitates FPGA design for image processing, makes the framework attractive for rapid prototyping and for researchers joining the field of embedded computer vision.

3. Method

The proposed detection scheme is composed of two major stages. In the first stage, pedestrian hypotheses are generated. Currently, we use background subtraction and foreground analysis in this stage. Candidate regions of interest (ROIs) are selected with respect to geometric features. Subsequently, in the verification stage hypotheses are validated based on the HOG descriptor and kernel SVM classification.

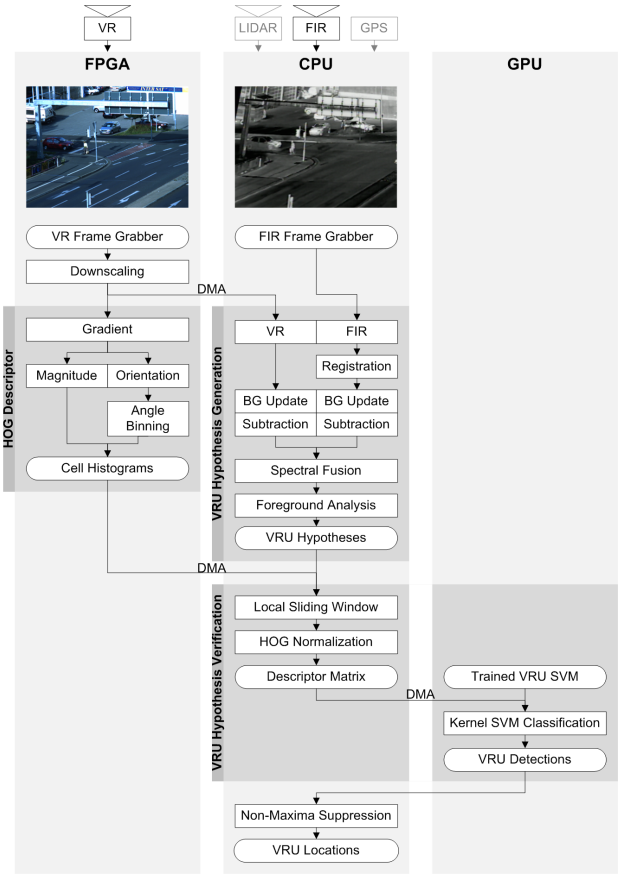

Figure 1. Multi-sensor hardware architecture and assignment of tasks. Currently, the system is equipped with a CCD color camera (VR) and a microbolometer FIR camera; integration of a LIDAR system and of GPS data of traffic participants is prepared. HOG descriptors are generated on the FPGA, multi-spectral candidate selection on the CPU, and kernel SVM classification on the GPU.

3.1. Multi-Spectral Hypothesis Generation

Based on our general purpose multi-sensor platform, see Fig. 1, our RSU merges imaging data in the visible and far-infrared range in a sequential fusion for pedestrian candidate generation. This multi-spectral fusion compensates individual weaknesses of VR (shadow artefacts) and FIR imaging (thermal reflections) and improves the robustness of hypothesis selection. Infrastructure-based systems that rely on stationary mounted sensors simplify the hypothesis generation stage. As the global region of interest remains constant, foreground objects can be extracted by background subtraction. Our system performs two individual background subtractions for each spectral range and merges the resulting foreground maps. In terms of registration, the FIR image is aligned with the synchronously captured VR image based on planar homography, mapping the planar road surface in both images by means of a projective transformation. The transformation matrix $H^{3\times3}$ was determined in an off-line calibration procedure, solving a system of equations of manually labeled corresponding points $(\boldsymbol{p}_{i,vr}, \boldsymbol{p}_{i,fir}), i \geq 4$ in the VR and FIR image, respectively.

As our hypothesis selection method relies on object silhouettes, a robust foreground segmentation is essential. For outdoor scenes, the background must be updated recursively during run-time as changes in the scene and illumination due to varying weather conditions have to be taken into consideration. We employ the widely-used mixture of Gaussians (MOG) model [24]. Each VR and FIR frame is then subtracted from its respective MOG background model, yielding two binary foreground maps. After polishing each map using basic morphology, the maps are merged via pixel-wise application of the logical *and* operation. Objects in the resulting foreground mask are analyzed iteratively in terms of simple geometrical features (position, size, aspect ratio). If a contour fulfills coarse target characteristics, it is passed to the verification module.

3.2. HOG-based Hypothesis Verification

Hypothesis verification i.e. separating human detections from false hypotheses is performed according to the HOG descriptor and SVM classification scheme proposed by Dalal and Triggs [14]. Instead of using an exhaustive scan, shifting a detection window on a regular lattice over the image at potentially all positions and scales, we perform a local spatial sliding-window search at the hypothesis ROI. For each window, HOG features are computed and evaluated with a kernel SVM classifier that was trained by supervised learning. Assuming that this procedure provides a peak at the correct object position and weaker responses around it, multiple nearby detections are merged to one ROI that best adjusts to the pedestrian using mean shift [12] as non-maxima suppression approach.

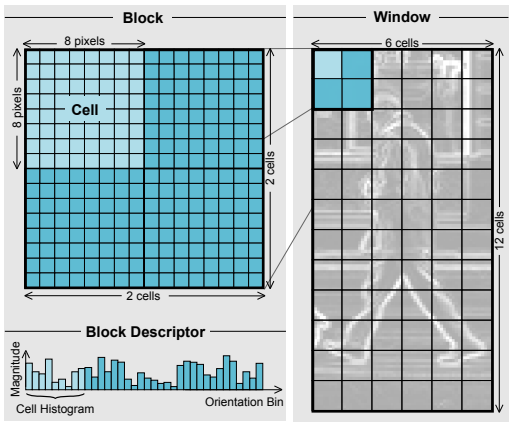

Figure 2. HOG descriptor scheme. A concatenation of cell orientation histograms weighted by the gradient magnitude generates the set of block descriptors.

3.2.1 Descriptor Structure

The basic idea of HOG is that local object appearance and shape is characterized by the distribution of intensity gradient directions. As the descriptor is well described in literature, we summarize here the structure of our implementation, which closely follows the original default detector [14]. Our descriptor operates on the grayscale VR images and evaluates windows of 48×96 pixels, see Fig. 2. First, the gradient magnitudes and directions are computed for the image patch. Then, in order to measure local distributions of gradient values, the window is divided into 6×12 cells covering 8×8 pixels each. For each cell, the pixels are discretized according to its gradient direction into 9 evenly spaced angular bins of an orientation histogram. The contribution depends on the gradient magnitude at the respective pixel. Because different background scenes, the presence of shadows or changes in illumination can cause significant variations in the gradient magnitudes, local contrast normalization is essential for superior performance. Hence, sets of 2×2 neighboring cells are grouped into overlapping blocks. The $4 \times 9 = 36$-dimensional block descriptor concatenates the corresponding four cell histograms, normalized to unit length. Finally, the HOG descriptor is represented by a concatenation of the entirety of block descriptors, yielding a $(5 \times 11) \cdot (2 \times 2) \cdot 9 = 1980$-dimensional feature space. Implementation differences with respect to [14] are discussed in section 5.1.

3.2.2 Kernel SVM Structure

Based on the HOG descriptor, machine learning techniques such as SVMs can learn an implicit representation of the classification object from examples and categorize unseen image patches into one of the predefined classes, pedestrian or non-pedestrian in our case. Part of the appeal for kernel SVMs is that non-linear decision boundaries can be learnt

by performing a linear separation in a high-dimensional feature space. We use a 2-norm soft margin kernel SVM with classification function $f(\boldsymbol{x})$,

$$f(\boldsymbol{x}) = sgn\left(\sum_{i=1}^{n_S} \alpha_i y_i K(\boldsymbol{s_i}, \boldsymbol{x}) + b_0\right) \quad (1)$$

where $\alpha_i \geq 0$ denote the positive Lagrange multipliers, b_0 the bias, y_i the class label, $\boldsymbol{s_i}$ the n_S support vectors, $\boldsymbol{x}$ a HOG instance and $K(\boldsymbol{s}, \boldsymbol{x}) = e^{(-\gamma\|\boldsymbol{s}-\boldsymbol{x}\|^2)}$ the Gaussian kernel function . The parameters γ (kernel) and C (weighting factor of slack variables) are determined by grid search. A description of the general procedure is given e.g. in [13].

4. Implementation

The road side unit is equipped with a SenTech STC-CL232A[1] progressive scan CCD color camera (1600 $\times$ 1200, 30 fps) and a FLIR SR-50[2] microbolometer camera (320 $\times$ 240, 25 fps) operating in the far infrared range ($7.5 - 13\mu m$). The hardware architecture and assignment of tasks for data processing is illustrated in Fig. 1. FPGA processing is performed on a microEnable IV-FULLx4 PC frame grabber board[3] being equipped with two devices: a Xilinx Spartan 3 XC3S 2000 provides data transfer interfaces to the camera (CameraLink) and to the host (PCIe). The HOG descriptor computation is implemented on a Spartan 3 XC3S 4000 using the graphics-oriented hardware development software VisualApplets[3]. Hypothesis generation and descriptor normalization is performed on an Intel Core i7 CPU 920 @ 2.66 GHz, SVM classification on a NVIDIA Geforce GTX 295 GPU. Inter-device transfers are managed via direct memory access (DMA).

[1] http://www.sentechamerica.com
[2] http://www.flir.com
[3] http://www.silicon-software.com

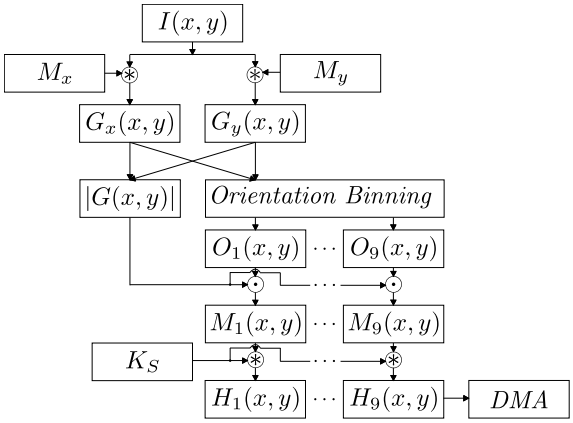

Figure 3. Flowchart of the FPGA-based convolution scheme for HOG descriptor generation.

4.1. Convolution Scheme for HOG Computation

The FPGA-based HOG scheme computes the descriptors for all potential window positions of the frame (pixel-wise) in one run. We outsourced the corpus of the computation, leaving only the descriptor normalization to the CPU. An overview of the scheme is given in Fig. 3. In the next subsections, we present our solutions of tackling these issues with low hardware resources and we describe the implementation of individual system components. For the experimental RSU setup using a long focal length lens (see Fig. 5), variations in pedestrian size are negligible and HOG descriptors are computed for one single scale level.

4.1.1 Magnitude-weighted Orientation Binning

We compute the 1-D spatial derivatives G_x, G_y in x- and y-direction by convolving the gradient masks M_x, M_y with the VR image I,

$$G_x = M_x * I \qquad M_x = \begin{pmatrix} -1 & 0 & 1 \end{pmatrix} \quad (2)$$

$$G_y = M_y * I \qquad M_y = M_x^T \quad (3)$$

The gradient magnitude $|G(x,y)|$ and orientation angle $\phi(x,y)$ are then computed for each pixel,

$$|G(x,y)| = \sqrt{G_x(x,y)^2 + G_y(x,y)^2} \quad (4)$$

$$tan\left(\phi\left(x,y\right)\right) = \frac{G_y(x,y)}{G_x(x,y)} \quad (5)$$

We insist on using the square root formulation as the best performance is reported for the Euclidean metric [14]. Note, however, that decimal places are cut off in our implementation. Calculating the $arctan()$ to get ϕ on FPGA hardware is expensive. Hardware friendly approximation algorithms are available but generally iterative and slow. Similar to [10], we adopted an orientation binning scheme that determines the angular bin (1-9, evenly spaced over $[0, \pi]$, unsigned gradient) without computing the orientation angle explicitly. Our implementation however, introduces two improvements. First, the number of bins is increased from 4 to 9 aiming for superior classification performance [14].

Second, the gradient angle is discretized with a scheme that avoids the use of signs, reducing the bit width for the required relational operators. From the signs of G_x, G_y we know the angle's respective quadrant ($I - IV$). The quadrant-respective orientation bin is determined with an integer multiplication scheme illustrated for bin 1, $[0, \frac{\pi}{9}]$,

$$0 < \quad tan(\alpha) \quad < tan\left(\frac{\pi}{9}\right) \quad (6)$$

$$0 < \quad \frac{|G_y(x,y)|}{|G_x(x,y)|} \quad < tan\left(\frac{\pi}{9}\right) \quad (7)$$

$$0 < \quad |G_y(x,y)| \quad < tan\left(\frac{\pi}{9}\right) \cdot |G_x(x,y)| \quad (8)$$

More specifically, the quadrant-respective angular binning $[0, \frac{\pi}{2}]$ is performed w.r.t. the horizontal principal axis of the unit circle in case the angle α to the horizontal axis is less than $\frac{2}{9}\pi$. Otherwise it is performed w.r.t. the vertical axis. Hence, the comparison range for orientation binning is reduced significantly, resulting in an equivalent reduction of bit width for the relational operators. The floating point multiplication in the right part of inequality (8) is replaced by fixed point operations by multiplying the terms by a scalar $\gg 1$ using bit shifts.

4.1.2 Histogram Generation

At this stage, information is held in 9 binary images O_i where the value 1 denotes that the pixel's gradient orientation lies within the angular range of bin i, 0 denoting the opposite. Now we multiply each $O_i(x, y)$ with the gradient magnitude $|G(x, y)|$ providing 9 non-binary magnitude-weighted bin images M_i. The histogram entry of a specific bin i in a particular cell can be computed by accumulating the pixel intensity values over the cell region within M_i. In order to calculate these bin entries for all cells over the entire image efficiently, integral maps (IMAPs) are a popular technique [10, 32]. We, instead, convolve a quadratic sum filter kernel K_S with M_i,

$$H_i = K_S * M_i \qquad K_S = \begin{pmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{pmatrix} \tag{9}$$

being implemented as a consecutive convolution of M_i with a vector of ones $\boldsymbol{k_S}$ and its transposed,

$$H_i = \boldsymbol{k_S^T} * (\boldsymbol{k_S} * M_i) \qquad \boldsymbol{k_S} = \begin{pmatrix} 1 & \cdots & 1 \end{pmatrix} \tag{10}$$

because K_S is separable. The HOG cell histograms H_i are transferred into the main memory of the host PC via DMA using a quad lane PCIe interface (760 MByte/s). In addition, a reference copy of the VR frame needed for hypothesis generation is transferred through an individual DMA channel. The resource usage level of the Spartan 3 XC3S 4000 is presented in Table 1.

Type of Resource	Usage	
4-input LUTs	28,616	46%
Internal Block RAM (18 kbit)	100	61%
Embedded Multipliers (18×18)	18	18%

Table 1. Xilinx Spartan 3 XC3S 4000 resource usage level for a processing resolution of 800×600 pixels.

4.2. GPU-based kernel SVM

We perform the kernel SVM classification on the GPU as the runtime complexity and memory requirements of non-linear SVMs is high. Our classifier is based on an available GPGPU implementation [11]. Compared to a CPU, a much larger portion of GPU resources is devoted to data processing than to caching or flow control, increasing throughput and reducing computation time. In our case, the classifier input is a matrix storing the HOG descriptors of all the hypotheses. Hence, the device can parallelize the classification over the entirety of instances.

For training the SVM, we use 2,416 positive examples of people in upright poses and 12,180 negative examples, both generated from the INRIA dataset [1]. Descriptors of the training samples are extracted with the FPGA implementation that was modified in a way that the frame grabber module is replaced by a buffer that loads INRIA samples from the PC to the board's memory. Training for one SVM took a few seconds on the GPU. By grid search we generated roughly one terabyte of data of different SVMs to finally select the parameters $C = 2^3$ and $\gamma = 2^{-7}$ for optimum performance.

4.3. System Core

Since the CPU is the system core, its major task is the management and flow control of the FPGA-GPU pipeline. In addition, it is employed in the hypothesis generation, the normalization of the HOG descriptors that are to be classified, and the tracking of detected pedestrians. Normalization is performed by dividing the feature vector $\boldsymbol{v}$ by the L2-norm,

$$\boldsymbol{v} \longrightarrow \frac{\boldsymbol{v}}{\sqrt{\|\boldsymbol{v}\|_2^2 + \epsilon^2}} \tag{11}$$

where ϵ is a small constant inhibiting divisions by zero. Interfaces for infrastructure-to-vehicle communication can be integrated straightforward as the system is running on a standard PC.

5. Experiments and Results

First, as a baseline, we evaluate our implementation with the INRIA benchmark dataset [1]. Then a case study on detection of pedestrians at the target intersection is presented in order to show the real-world performance of our RSU.

5.1. INRIA Benchmark

For comparing human detection algorithms, the INRIA dataset [1] has established as a de facto baseline [18]. Evaluation is performed on a per-window basis and results are illustrated as detection error trade-off (DET) curves plotting miss rate versus false positives on a log-log scale. We have generated a CPU-based reference implementation with our HOG scheme and we compare the Gaussian kernel SVM classification performance of our FPGA-based HOG scheme to this reference implementation and to results from [14].

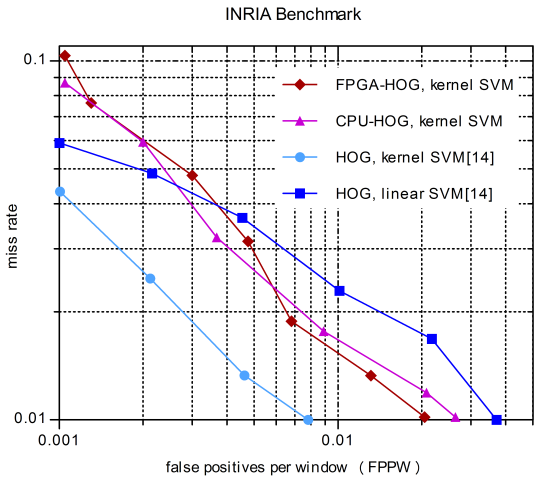

Figure 4. Gaussian kernel SVM classification performance of our FPGA- and CPU-based HOG implementations. The data for kernel SVM R-HOG and linear SVM R-HOG is extracted from [14] and represent classification performance after retraining.

The classification results plotted in Fig. 4 show that our FPGA- and CPU-based HOG implementations are consistent. In contrast to the curves extracted from [14], we have not yet applied classifier retraining techniques that have the potential to improve performance by an order of magnitude [22]. In fact, the deviation of miss rate (+6% at 10^{-4} FPPW) lies within this margin for improvement, indicating that the performance of our kernel SVM HOG system will meet the state-of-the-art set by Dalal and Triggs. The tendency to inferior values for smaller FPPWs results from quantification effects due to the size of the testing set (4,530 samples).

We expect minor effects on classification performance due to the following implementation differences compared with [14]. First, we work on grayscale images. Second, gradient magnitudes are computed with fixed point accuracy. Third, votes are not interpolated between neighboring histogram bin centers in both orientation and position (anti-aliasing). Finally, our implementation does not perform a Gaussian spatial down-weighting of values with respect to their location in the respective block.

5.2. Case Study

Outdoor experiments show the potential for deploying the road side unit in practice. The experimental sensor setup and view on the target intersection is depicted in Fig. 5. As expected, multi-spectral sensor fusion reduces artefacts in the hypothesis generation stage. We observe, that both shadows in the visible range and reflections in the far-infrared range are eliminated. Results of the HOG-based hypothesis verification are shown in Fig. 6, 7, the classifier trained on pedestrians does detect humans on bicycles and motorcycles as well. Visual inspection indicates that the

Figure 5. Experimental setup of the road side unit used for evaluation. In the background: target intersection.

detections from the hypothesis verification module are often discriminative enough to handle occluded humans and to separate groups of pedestrians that had been detected as one common hypothesis object.

Our hypothesis generation is not yet optimized, it detects about 80 per cent of pedestrians. For quantitative evaluation, we limited the ROI to a crosswalk of our target intersection in order to mainly detect pedestrians and we captured 5,000 frames of synchronized VR/FIR data in the course of the day and at different weather conditions. To ensure unbiased results, we divide the dataset into temporal windows of 5 frames each and evaluate only one randomly chosen frame per window. The evaluation is based on 1,000 manually annotated regions, that have been identified by hypothesis generation[4]. Detection rate and false alarm rate per frame are 95.4% and 0.1%, respectively.

5.3. Computation Time

The total latency of our convolution scheme for FPGA-based HOG descriptor computation is 312 μs demonstrating the efficiency. It is obtained with 800×600 pixels at a design frequency of 63 MHz. The latency distribution of the consecutive HOG computation steps is presented in Table 2.

Due to our candidate selection, the number of windows to be validated is decreased significantly compared to an exhaustive scan approach. Another benefit is the inherent reduction of false alarms while maintaining a constant detection rate and speeding up the entire system as classification tends to be the most time-consuming task. The multi-spectral hypothesis generation takes 25 ms, depending on the number of contours. The computation time for hypothesis verification depends on the number of candidates that are to be validated. Per candidate, we perform a local 2-dimensional spatial search classifying $10 \times 10 = 100$ windows. GPU-based Gaussian kernel SVM prediction takes about 65 μs per window including transfer times from and back to the CPU. In total, the entire pipeline including descriptor computation and SVM evaluation takes less than

[4]Data of the FIR sequences is available from the authors for non-commercial research purposes.

Figure 6. Pedestrian detection results of the HOG-based hypothesis verification from different sequences. Note the detected bicyclist in the first image row, second from left.

100 ms ($>$ 10 fps) when a maximum number of 1000 windows are to be classified. Being the bottleneck of the current implementation, ongoing work focuses on outsourcing the descriptor normalization from the CPU onto the GPU.

6. Discussion and Conclusions

In this paper we have introduced a processing pipeline of FPGA, CPU and GPU architectures used in a multi-sensor approach for pedestrian detection to improve intersection safety. Experimental results indicate the capability of the approach to achieve state-of-the-art classification performance in real time by applying a hardware-accelerated Gaussian kernel SVM. For real-time processing, we propose a flexible hardware architecture that can be implemented on available standard PC components. The HOG descriptor generation is outsourced to a low-cost FPGA device that performs feature extraction on the fly with a novel convolution scheme evoking a latency of 312 μs. Kernel SVM classification is calculated in parallel on the GPU. In this road side unit setup, the evaluation of 1000 windows takes less than 100 ms by pre-selecting candidates based on multi-spectral image fusion. A promising direction to speed up the hardware-based classification is to prefilter hypotheses with a linear SVM [26] or using a multi-resolution approach [30] in order to restrict the use of the kernel SVM to verification purposes. Classification performance can be improved by evaluating a joint HOG descriptor of both the VR and FIR domain, candidate selection by integrating a time-of-flight (lidar) device. The FIR sensor will then be used mainly for hypothesis verification and for detection of hot spots, e.g. to discriminate between pedal cyclists and motorcyclists.

HOG computation step	Latency in [μs]	
Image buffer	26.2	8.4%
Scaling	26.0	8.3%
Gradients, magnitudes, orientations	52.2	16.8%
Histograms	207.2	66.5%
Total	311.6	

Table 2. Latencies of the consecutive HOG computation steps for a UXGA camera input (1600 $\times$ 1200 pixels) downscaled to a processing resolution of 800 $\times$ 600 pixels and a design frequency of 63 MHz.

Acknowledgements

The authors at Aschaffenburg University of Applied Sciences acknowledge the support of the Bayerisches Staatsministerium für Wissenschaft, Forschung und Kunst, in the context of the Forschungsschwerpunkt Intelligente Verkehrssicherheits- und Verkehrsinformationssysteme.

A reference implementation of our FPGA-based HOG descriptor on the microEnable IV-FULLx4 board (addressable via C++ SDK) is available from the authors for non-commercial research purposes.

References

[1] INRIA person dataset, 2005. http://lear.inrialpes.fr/data/human.

[2] European Road Safety Observatory (ERSO), annual statistical report, 2008. http://www.erso.eu.

[3] National Highway Traffic Safety Administration (NHTSA), Traffic safety facts, 2008 data, dot hs 811 162, 2008. http://www-nrd.nhtsa.dot.gov.

[4] INTERSAFE-2, cooperative intersection safety, 2009. http://www.intersafe-2.eu.

[5] ITS, Japan, assistance for safe driving, 2009. http://www.its-jp.org.

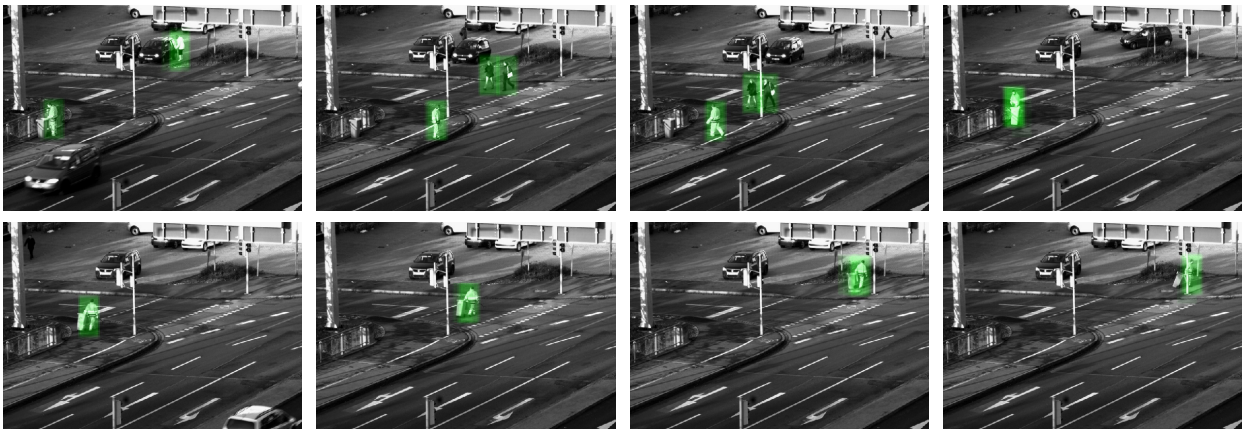

Figure 7. Pedestrian detection results of the HOG-based hypothesis verification from a sequence. Note that the garbage can does not affect the detection of the garbageman.

[6] Ko-FAS, Kooperative Sensorik und kooperative Perzeption für die präventive Sicherheit im Straßenverkehr, 2009. http://www.kofas.de.

[7] U.S. Department of Transportation's (DOT's) IntelliDrive program, 2009. http://www.intellidriveusa.org.

[8] O. Aycard, A. Spalanzani, M. Yguel, J. Burlet, N. D. Lac, A. D. L. Fortelle, T. Fraichard, H. Ghorayeb, M. Kais, C. Laugier, C. Laurgeau, G. Michel, D. Raulo, and B. Steux. PUVAME - new french approach for vulnerable road users safety. In *IVS*, pages 2–7, 2006.

[9] S. Bauer, U. Brunsmann, and S. Schlotterbeck-Macht. FPGA implementation of a HOG-based pedestrian recognition system. In *MPC Workshop*, pages 49–58, 2009.

[10] T. P. Cao and G. Deng. Real-time vision-based stop sign detection system on FPGA. In *DICTA*, pages 465–471, 2008.

[11] A. Carpenter. cuSVM, a CUDA implementation of support vector classification and regression. http://patternsonascreen.net/cuSVM.html.

[12] D. Comaniciu. An algorithm for data-driven bandwidth selection. *TPAMI*, 25(2):281–288, 2003.

[13] N. Cristianini and J. Shawe-Taylor. *Support Vector Machines and other kernel based learning methods*. Cambridge University Press, 2006.

[14] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, pages 886–893, 2005.

[15] P. Dollar, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: A benchmark. In *CVPR*, pages 304–311, 2009.

[16] M. Enzweiler and D. M. Gavrila. Monocular pedestrian detection: Survey and experiments. *TPAMI*, 31(12):2179–2195, 2009.

[17] D. M. Gavrila and S. Munder. Multi-cue pedestrian detection and tracking from a moving vehicle. *IJCV*, 73(1):41–59, 2007.

[18] D. Gerónimo, A. M. López, A. D. Sappa, and T. Graf. Survey on pedestrian detection for advanced driver assistance systems. *TPAMI, to appear*.

[19] M. Hiromoto and R. Miyamoto. Hardware architecture for high-accuracy real-time pedestrian detection with CoHOG features. In *ECVW*, pages 894–899, 2009.

[20] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[21] S. Maji, A. Berg, and J. Malik. Classification using intersection kernel support vector machines is efficient. In *CVPR*, pages 1–8, 2008.

[22] S. Munder and D. M. Gavrila. An experimental study on pedestrian classification. *TPAMI*, 28(11):1863–1868, 2006.

[23] R. M. R. Kadota, Y. Nakamura. Hardware implementation of HOG feature extraction for real-time pedestrian recognition. Technical report. In *IEICE Smart Info-Media System*, volume 109, pages 43–48, 2009.

[24] C. Stauffer and W. E. L. Grimson. Adaptive background mixture models for real-time tracking. In *CVPR*, pages 2246–2252, 1999.

[25] H. Sugano and R. Miyamoto. Parallel implementation of pedestrian tracking using multiple cues on GPGPU. In *ECVW*, pages 900–906, 2009.

[26] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman. Multiple kernels for object detection. In *ICCV*, pages 606–613, 2009.

[27] P. A. Viola, M. J. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. *IJCV*, 63(2):153–161, 2005.

[28] C. Wojek, G. Dorkó, A. Schulz, and B. Schiele. Sliding-windows for rapid object class localization: A parallel technique. In *DAGM*, pages 71–81, 2008.

[29] L. Zhang and R. Nevatia. Efficient scan-window based object detection using GPGPU. In *CVPR*, pages 1–7, june 2008.

[30] W. Zhang, G. Zelinsky, and D. Samaras. Real-time accurate object detection using multiple resolutions. In *ICCV*, pages 1–8, 2007.

[31] Y. Zhang, A. S. Dhua, S. J. Kiselewich, and W. A. Bauson. *Challenges of Embedded Computer Vision in Automotive Safety Systems*, pages 257–279. Advances in Pattern Recognition. Springer, 2009.

[32] Q. Zhu, M.-C. Yeh, K.-T. Cheng, and S. Avidan. Fast human detection using a cascade of histograms of oriented gradients. In *CVPR*, pages 1491–1498, 2006.