FPGA-accelerated sliding window classifier with structured features

Ondřej Sychrovský, Martin Matoušek, Radim Šára
{sychro1,xmatousm,sara}@cmp.felk.cvut.cz

Center for Machine Perception, Department of Cybernetics,
Faculty of Electrical Engineering, Czech Technical University in Prague

CENTER FOR MACHINE PERCEPTION

Abstract

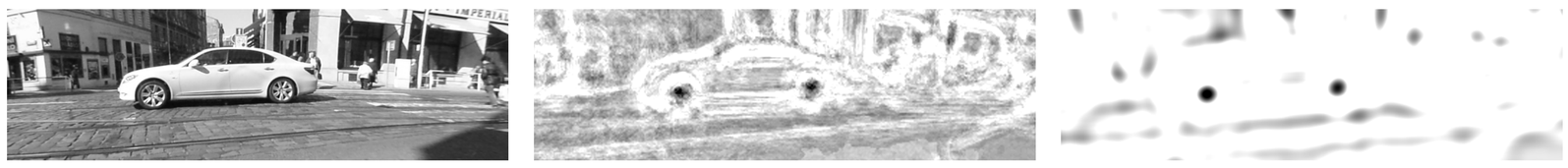
Certain classification tasks in computer vision require the classifier response to be computed in every pixel of an image. When combined with large, complex features, it becomes challenging to build such a classifier on a standard PC architecture and achieve real-time performance.

We present an FPGA implementation of a car wheel classifier response computation, built as an instantiation of a generic classification system. An interesting optimization problem concerning performance and speed is addressed. Our implementation is running in real-time as a part of a more complex collision mitigation system based on car detection in video data.

Hardware Platform

- Xilinx Virtex-6 ML605 evaluation board
- Xillybus PCI Express wrapper

Motivation: The Wheel Detector



The input image, FPGA output – the classifier response map, response after spatial post-processing.

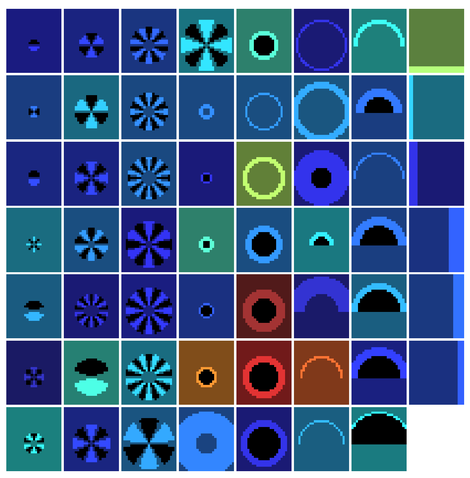
Classifier with Structured Features

1. image patch normalization
   $\mathbf{S}_{x,y} = n(x,y) \cdot \mathbf{I}_{x,y}$
2. dot product
   $d_i(x,y) = \langle \mathbf{S}_{x,y}, \mathbf{M}_i \rangle$
3. weak classifier

$$v_i = \begin{cases} d_i & \text{if } \mathbf{M}_i \in \mathbb{R}^{w \times h} \\ |d_i| & \text{if } \mathbf{M}_i \in \mathbb{C}^{w \times h} \end{cases}$$

$$y_i = \begin{cases} +1 & \text{if } g_i v_i > t_i \\ -1 & \text{if } g_i v_i \leq t_i \end{cases}$$

4. overall classifier response
   $r(\mathbf{S}_{x,y}) = \sum_{i=1}^{n} \alpha_i y_i$

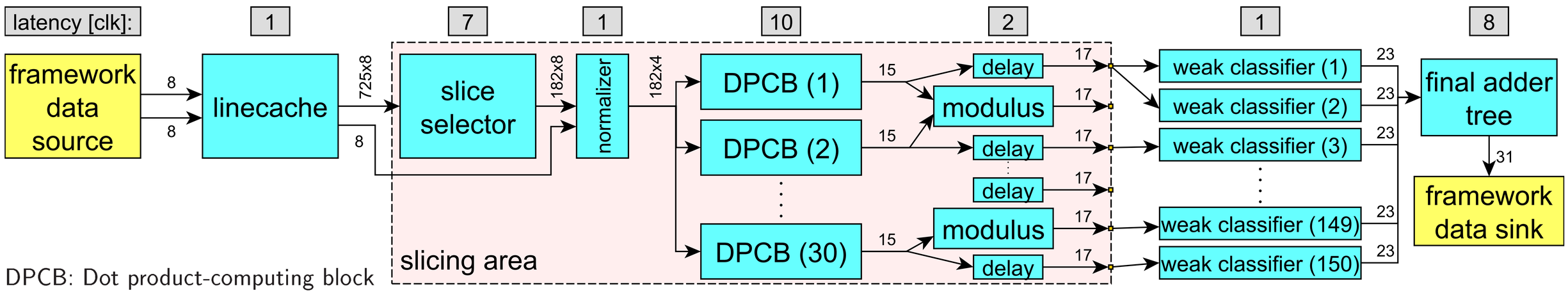


The selected kernels $\mathbf{M}_i$, quantized to $\{-1, 0, +1\}$.

- AdaBoost learned
- 150 weak classifiers defined by $\mathbf{M}_i$ (kernel), t_i (threshold), g_i (sign) and α_i (weight)
- 55 unique kernels (22 complex, 33 real) $\rightarrow$ 77 dot products
- kernel (patch) size 25×29 pixels

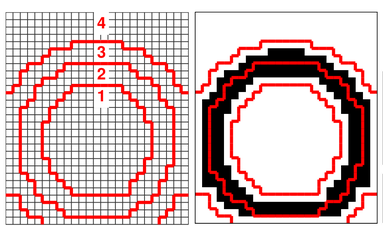
Positive example:

FPGA Implementation



DPCB: Dot product-computing block

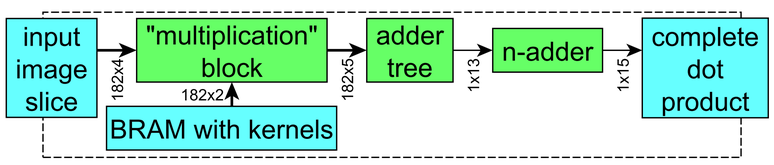
Slice Selector



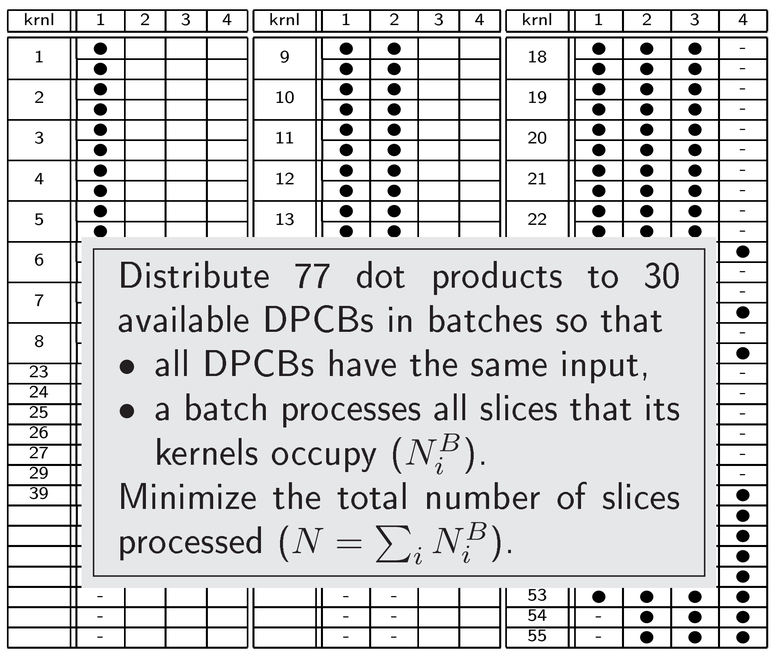
	slice number			
	1	2	3	4
krnl k		•	•	

The 25×29 patch and kernels are divided into four parts (slices). Example kernel and its corresponding *Incidence table* (occupying slices 2 and 3). Each slice takes one clock cycle to process.

Dot product-computing Block



Slicing Scheme Optimization



Distribute 77 dot products to 30 available DPCBs in batches so that

- all DPCBs have the same input,
- a batch processes all slices that its kernels occupy (N_i^B).

Minimize the total number of slices processed ($N = \sum_i N_i^B$).

Resource Consumption

num. blocks	clk cycles	LUTs	BRAMs	DSPs
30	7	73202	371	364
20	10	53445	261	364
13	15	39577	184	364
10	19	32456	151	364

Resource consumption for different number of DPCBs for the image patch of size 25×29. Each block adds 11 BRAMs and about 2000 LUTs.

	overall			per DPCB	
patch size	LUTs	BRAMs	DSPs	LUTs	BRAMs
9×15	16803	67	68	369	1
15×19	26506	111	144	731	2
19×25	38506	177	238	1297	7
25×29	53445	261	364	1952	11
29×35	76541	347	508	2720	15

Resource consumption for different image patch sizes with fixed number of 20 DPCBs.

Conclusions

We have proposed an approach for implementing a dense linear classifier on an FPGA. The proposed architecture is scalable and quite generic, it is up to the designer how many blocks are needed to place in the FPGA. The proposed wheel classifier response map computation is used in a car detector running in an intelligent vehicle as a part of a more complicated collision mitigation system [1], that requires processing cycle of 20–30 fps and a maximum latency of 200 ms.

Acknowledgement

This work was supported by the European Commission under interactIVe, a large scale integrated project, part of the FP7-ICT-246587 for Safety and Energy Efficiency in Mobility. The authors would like to thank all partners within interactIVe for their support.

References

[1] P. Heck, J. Bellin, M. Matoušek, S. Wonneberger, O. Sychrovský, R. Šára, and M. Maurer, "Collision mitigation for crossing traffic in urban scenarios," in *Proc IEEE Intelligent Vehicles Symposium*, 2013.

[2] O. Sychrovský, M. Matoušek, and R. Šára, "FPGA-accelerated sliding window classifier with structured features," Center for Machine Perception, Czech Technical University, Prague, Research Report CTU–CMP–2013–16, June 2013.

FPL2013, Sept. 2–4, Porto, Portugal