High Performance Architecture for Object Detection in Streamed Video

Pavel Zemčík, Roman Juránek, Petr Musil, Martin Musil, Michal Hradiš
Department of Computer Graphics and Multimedia
IT4Innovations Excellence Center, Graph@FIT
Faculty of Information Technology,
Brno University of Technology, Brno, Czech Republic

Abstract

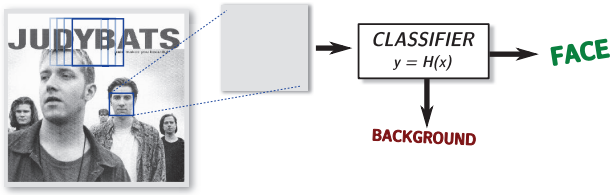
Object detection is one of the key tasks in computer vision. It is computationally intensive and it is reasonable to accelerate it in hardware. The possible benefit of the acceleration is reduction of the computational load of the host computer system, increase of the overall performance of the applications, and reduction of the power consumption. In this paper, we shortly review the WaldBoost based object detection algorithm and introduce a novel architecture of engine for high performance multi-scale detection of objects in video. We implemented the engine in FPGA and we show that it can process 640x480 pixel video streams at over 160 fps without the need of external memory. We evaluate the design, compare it to state of the art designs, and discuss its features and limitations.

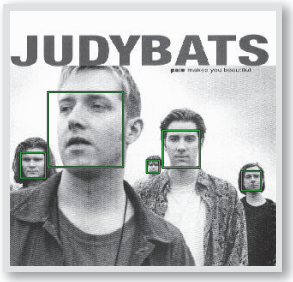
Object Detection with Classifiers

1. Input image is scanned with a sliding window with fixed size. Image is scaled to scan areas of different size

2. Every subwindow is classified with previously learned classifier as an object or background

3. Positive responses of the classifier are detected objects



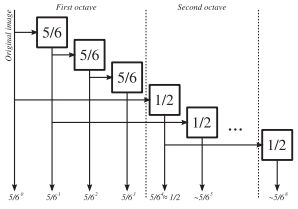


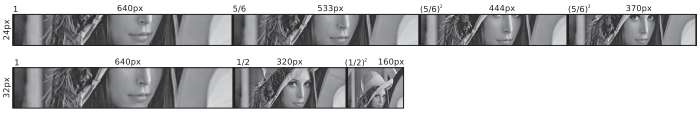
We use WaldBoost algorithm for classifier training and LRD or LBP image features. This combination is hardware-friendly is it requires only addition and comparison operations.

Proposed Architecture

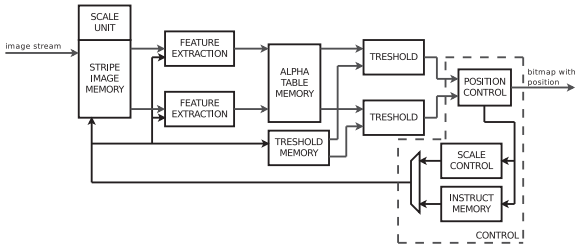
4. Input is image stream from a camera

5. Image is buffered using internal memory. Different scales of the image are created on the fly. Only a narrow image stripe containing all scales is stored (e.g. 24 or 32 rows). We explored 5/6 fine scaling and 1/2 coarse scaling.

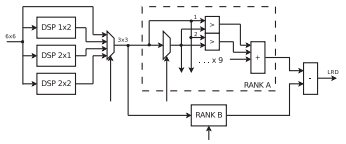




6. On every horizontal position of the stripe, the classifier is evaluated. The engine is controlled by a microprogram which defines feature parameters, and by an automata controlling classification execution and image scaling. The engine is realized as 9 stage pipeline and therefore 9 positions of the stripe are evaluated in parallel.

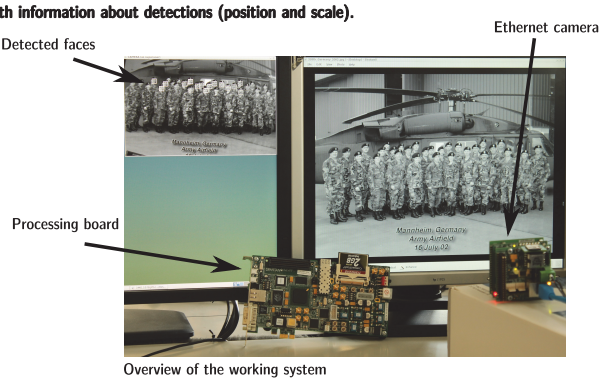


7. We use simple features - Local Binary Patterns and Local Rank Differences (see image below). They are evaluated from 6x6 pixel blocks, and they are based only on comparison and addition operations.



Circuit for evaluation of LRD image features

8. **Classification results are sent out along with the input data either as another bitmap or a stream with information about detections (position and scale).**



Overview of the working system

Results

9. The resuts achieved in the experiments are summarized in Table 1 and Table 2 below.

	FPGA Resources				Performance	Frames
	Registers	LUTs	BRAMs	DSPs	[MHz]	per Second
LBP 1/2	1678 (3%)	7098 (26%)	77 (66%)	0	163	87
LBP 5/6	1737 (3%)	7405 (27%)	43 (37%)	0	152	131
LRD 1/2	1673 (3%)	7014 (26%)	29 (25%)	0	163	103
LRD 5/6	1732 (3%)	7373 (27%)	31 (27%)	0	152	164

Table 1: Resource consumption of different versions of our design.

	FPS	Features	Scaling method	Scale factor	Freq. [MHz]	BRAMs	LUTs	Regs.	FPGA
Huang [10]	—	Haar	Img. scaling	1.2	65	—	80000	—	Virtex5 LX155T
Cho [2]	7	Haar	Img. scaling	1.2	—	41	66900	21900	Virtex5 LX110T
Kim [12]	50	MCT	Img. scaling	—	106	18	133000	45700	Virtex5 LX330T
Kyrkou [13]	40	Haar	Img./feature scaling	1.33	100	24	25800	23800	Virtex2 XC2VP30
Lai [14]	143	Haar	Img. scaling	1.25	126	44	20900	7800	Virtex2 XC2VP30
Zemcik [19]	22	LRD	None	None	—	14	2980	—	Virtex2 250
Ours LRD 5/6	164	LRD	Img. scaling	1.2	152	31	7373	1732	Spartan6 LX45T
Ours LRD 1/2	103	LRD	Img. scaling + multiple classifiers	1.2	163	29	7014	1673	Spartan6 LX45T

Table 2: Comparison of our design to state-of-the-art.

Conclusion

We proposed and implemented the engine in Xilinx Spartan 6 LX45T FPGA. It takes only a fraction of its resources. When compared to state-of-the-art designs, our engine is smaller, it operates on higher frequency, and it outperforms others in terms of frames per second. This was achieved mainly thanks to the unique buffering and scaling, and exploitation of simple, hardware-friendly image features.

Acknowledgments

This work has been supported by the Czech government project Tools and Methods for Video and Image Processing for the Fight against Terrorism, VG20102015006, the European Regional Development Fund through the *IT4Innovations Centre of Excellence* project, CZ.1.05/1.1.00/02.0070, and the Technology Agency of the Czech Republic project *V3C - Visual Computing Competence Center*, TE01010415.