A Highly Integrated Hardware-Software Co-design and Co-verification Platform

Shufan Yang, *Member, IEEE,* Zheqi Yu

***Abstract*—The demand for low-cost, high-volume consumer multimedia products has led to the need of faster methodologies of hardware-software co-design and co-verification, which enables projects to be completed in a shorter time and with greater confidence in integrated systems. We present here a new method to significantly shorten system verification time and have applied it as a pedestrian tracking application as a case study based on a system-on-chip design device (Xilinx ZYNQ-7000 hardware platform ZC702). Three open access image data sets with four common video codecs (H.263/MPEG-4 Part2, H.264/MPEG-4 AVC, Microsoft codec, Google (on2) codec) have been used to test precession rates of tracking.**

***Index Terms*—Hybrid FPGA-SoCs, ZYNQ, Hardware-Software co-design, Co-verification**

I. Introduction

CURRENT evolution in semiconductor industry brings a possibility of introducing significant complex modules onto system-on-chip design platforms. Traditional verification processes are software-based simulations or physical verifications. They have limited expansions and reuse capabilities [1]. However, an architecture-level design space exploration in hardware/software co-design and co-evaluation based on FPGA-SoCs offers sufficient performance to run application software on hardware system with orders of magnitude faster than software simulators. The FPGA-SoCs allow engineers to explore the high level of integration with a quick and convenient co-verification, which helps engineers to determine whether software and hardware both are executed correctly on microcontroller processors and customised hardware.

The FPGA-SoCs platforms have still in the early stage of development[1]. The difficulty of integrating software and hardware is how to create a flexible interface between hardware implementation and software algorithms. Since those integrations take a long learning curve for engineering, it causes less design productivity even using a advanced design tool.

In this paper, we present a buffering scheme and memory mapping interfaces for hardware/software co-design and co-verification platform. A verification process for image and video multimedia products with a universal scalable codec platform has been evaluated based on this platform. It also supports an architecture level verification for software/hardware co-design, allowing design optimisation before an embedded design is committed for chip fabrication.

Shufan Yang is at James Watt Building, School of Engineering, University of Glasgow, G12 8QQ, Glasgow, United Kingdom. Email:shufan.yang@glasgow.ac.uk

Zheqi Yu is with the School of Computer Science, University of Wolverhampton, WV1 2SB, Wolverhampton, United Kingdom.

II. Relate Work

FPGAs are commonly used in image and video multimedia applications to ensure vision systems being capable to run at real-time [1]. Those applications have been reported with carefully considered hardware complexities and limited hardware-software co-verification methods. Table I summarises features of recent hybrid FPGA-SoCs design for image and video multimedia processing applications using those "on-shelf" chips.

Mefenza et al (2015) proposed a Components Interconnect and Data Access (CIDA) interface mode to meet the requirement of image data transferring among individual image processing modules. Image segmentation applications have been implemented using a ZYNQ chip. However, their work is directly coded using VHDL with limited configurable features to wrap a user logic function; these configurations of CIDA also needs experienced hardware designers to carry out. A software and hardware integrated design has been demonstrated in Crezuela-Mora et al (2015)'s work, but their work has limitations as it relies on an AXI-Stream bus interface to transmit image data in DDR system memory. Similar to Mefenza's work, this work might be incompatible with different DDR memory settings among various boards.

Although several image processing systems can be used for real-time image processing, most current work uses a highly optimised hardware implementation with less modification flexibilities to add additional features. In addition, no works have considered providing a platform for hardware-software co-verifications, since hardware and software are only built independently with plug-and-debug methods. We present a novel framework to integrate universal video codecs with Linux-based drivers for communicating with FPGA logic. Taking the advantage of the Xilinx Vivado high-level synthesis flow, an architecture-level test bench can be integrated with embedded software and customised hardware. Any developers can mitigate our framework into a sophisticate image and video multimedia processing system to accelerate time-to-market and improve product readabilities.

III. Principles and System Architectures

In this work, we developed an AXI wrapped buffer driver and memory mapping scheme to allow simultaneous designing processes with integrated hardware and software designs. This method also allows designers to obtain a flexibility of integrate hard IP blocks into systems using a video streaming method. A integrated co-evaluation platform is also introduced to help our multimedia system to run on the real-time hardware platform.

TABLE I
COMPARISON OF RECENT WORK ON IMAGE AND VIDEO MULTIMEDIA PROCESSING BASED ON HYBRID FPGA-SOCS SYSTEMS

Researchers	Application	H/S Buffer management	Video format	FPGA Board	Verification method
Mefenza et al (20015) [2]	Segmentation	CIDA	Not supported	ZYNQ XC7	Hardware prototype only
Hsiao, et al (2015) [3]	Pedestrian detection	Write/Read buffers	MPEG2	Xilinx Spartan-6	Separate HW/SW testing
Cerezuela-Mora, et al (2015) [4]	Sobel Filter	Frame buffer	Raw data only	ZYNQ 7000	Hardware prototype only
Bieszczad, et al (2016) [5]	Thermal Image Processing	Frame buffer	Raw data	Cyclone V	Hardware prototype only
Altuncu, et al (2015) [6]	Edge Detection	24bit convolution buffer	Not support	Zed-Board	Separate HW/SW testing

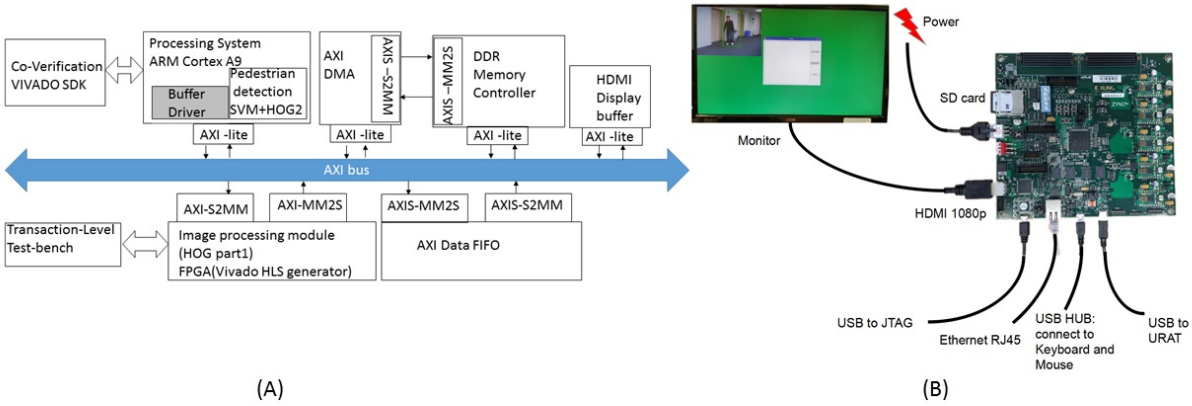

Fig. 1. (A) The block diagram of image and video multimedia application platform (B) The Prototype platform using the Xilinx ZYNQ-ZC702 board

A. System Overview

Figure 1 shows our system overview and prototyping environment (Xilinx 702C board). The electric I/O circuit interconnections between FPGA IP modules and AXI system buses for both the processing system (PS) and the programmable logic portion (PL) in the system platform are shown in Figure 1(A). As illustrated in Figure 1(A), a dedicated DMA controller (one of Xilinx IP cores provide by Xilinx [7]) provides memory-mapped access to the DDR memory through the AXI_MM2S and AXI_S2MM bus, where MM2S stands for "memory-mapped access to streaming access"and S2MM stands for "streaming access to memory-mapped access". This DMA controller also transfers data from the FPGA side to one of the ARM cortex A9 cores through the AXI-lite bus. The AXI streaming buses, AXIS_MM2S and AXIS_S2MM, can source continuous stream of image data into Cortex A9 with a configurable block size [8]. Extra AXI bus between the AXI data buffer and cortex A9 is used for transferring data from cortex A9 to a HDMI display module for playing back videos. Our device drive allows interrupts handling for the commence of VDMA to transfer data between FPGA buffers to user space, which enables the hardware IP runs as a peripheral within an software operating system.

The version of Linux kernel used for our published source code is 3.10.0. A multi-platform supported and open-sourced U-boot software [14] is used to load operating systems for ARM cortex-A9 processor with a C++ enabled tool chain and virtual memory management support. A HDMI monitor is used to playback the post-processed video. As shown in Figure1(B), it shows a person walking through a corridor with a green bounding box for pedestrian detection in current frame.

Compared to traditional bare-metal hardware system designs, this architecture integrates a software development cycle, including kernel configuration and compilation, boot loader, and finally the generation of the hardware description file using high-level synthesis (HLS) steps with architecture-level co-verifications.

B. Universal Codec for Reconfigurable SoC Systems

To enable a co-design system between hardware and software, a particular memory mapping scheme is also developed to enable soft multimedia decoding. V4L2 framework and FFmpeg library, both actively maintained by a software developing community [9], are used to build a universal codec platform on reconfigurable SoCs. In this work, we use FFmpeg 3.1 to support more than 90 encoders. A cross compiler tool to transplant FFmpeg from X86 architecture into ARM architecture has been used based on GUN automakefile. All I/O routines and coding/decoding is handled very efficiently by using off-the-shelf components from the V4L2 API [9]. The left panel of Figure 2 shows how to use memory mapping to direct access Video Direct Memory Access (DMA). System call *mmap()* is used to map the video encode applications into video DMA physical location. It has the benefit of increasing computational efficiency since ARM processors do not need to copy data into kernel spaces. For other control signals, the system call *ioctal()* is used for control signalling configurations. Linux system function call *dma_alloc_coherent* binds the physical address of buffers and its virtual kernel space address to allocate chunk of memory, which provides data to user space encoder running at a cortex A9 processor. The memory mapping is applied to the user program to correspond

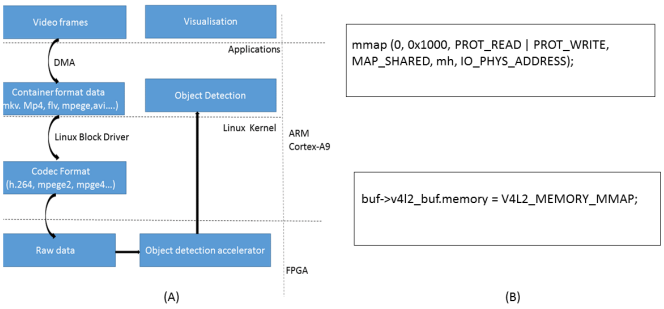

Fig. 2. Universal Codec implementation for hardware-software co-design based on Reconfigurable Hardware

with the AXI streaming bus in the DMA with transmission rates range from 4096 by 16 bits to 4096 by 64 bits at a time.

Processes in application user spaces write data into the write buffers at the source end, calling the kernel space driver to start DMA, and then successfully transmits the data into FPGA connected AXI FIFO. For the reverse behaviour of the data transmission, i.e., to read data from the FPGA, the driver is first called to start DMA so as to read the FPGA data and write them into the read buffer. Using this strategy, the main program can evaluate the memory requirement and dynamically allocate buffer spaces to facilitate data transferring between processing logic (FPGA) to processing system (Cortex A9).

C. Hardware-Software co-evaluation platform

In addition to our new driver buffering scheme and memory mapping approach in soft multimedia decoding, we also develop a verification flow for the integrated multimedia processing system (as shown in Figure3). After the pedestrian detection (HOG) module was ran through a high-level synthesise tool to generate a hardware description with Xilinx Vivado HLS (2015.4) design tools [7]. The behavioural Verilog test bench that wraps around the HOG design top level is also provided by Vivado. This test bench provides clocking and reset stimulus to the HOG design top level to run simulations, which is useful for getting familiar with the signalling on the FPGA core modules by observing simulation waveforms. After the behaviour of the HOG design is satisfied, a transistor level hardware description is generated. At next stage, a behaviour model with C-wrapper interface is automatically generated for fast system verification to explore the feasibility of the HOG algorithm in terms of throughput and latency, which we refer to as an architectural level verification (as shown in Figure 1). This C-wrapped test bench can work with all simulation outputs from a behavioural RTL through post-implementation timing. After the HOG based pedestrian detection module has been solidified through the Vivado physical-implementation flow, it can be integrated into the design using IP integrator (available from Xilinx Vivado Tools). Finally the image processing results are observed using a monitor connected with HDMI interface connected to the ZYNQ 702c board. Finally the image processing results are observed using monitor connected with HDMI interface in the ZYNQ 702c board.

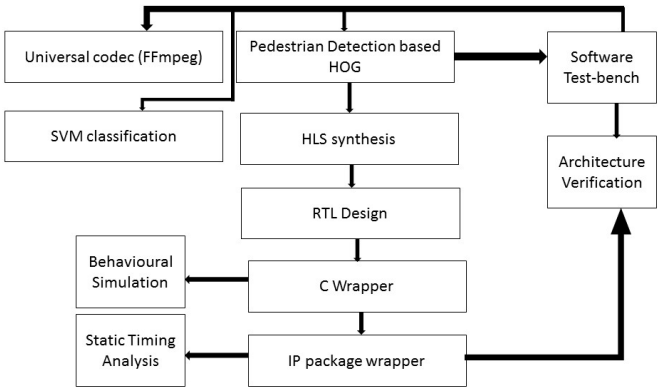

Fig. 3. Co-verification flow for developing hardware and software and integrating both design into an embedded system based on a hybrid FPGA-system-on-chip architecture

IV. A Case Study of Pedestrian Tracking

A. Pedestrian Tracking algorithms

We implement a video pedestrian tracking based on a HOG algorithm to demonstrate the functionality of our verification platform. A feature descriptor generated by the histogram of oriented gradients is used to describe the local objects appearance and shape [10]. Following the calculation of the local histograms of the image, a normalization process is used to reduce the impacts of changes that were introduced by various illuminations and shadowing in the image. Once another histogram of oriented gradient descriptors is calculated, the descriptors are fed into a classification system based on a supervised learning algorithm.

In this work we use a support vector machine (SVM) as a baseline classifier. The SVM is trained with the INRIA training set and the classifier is based on AdaBoots. The value of the regularisation coefficient in SVM is an important value since this parameter controls the degree of over learning. Thus, a small value of coefficient allows a large separation margin between classes, which reduces over learning and improves generalisation. In our experiments, we chose 1.5 as the coefficient value.

B. Hardware Implementation platform

Pedestrian tracking algorithms which are implemented in Zynq-7000 contain an Artix-7 FPGA and an ARM Cortex-A9 processor on the same chip together. In our hardware-software co-design platform, a small part of computations (colour-to-grey) is performed on the FPGA while pedestrian detection and SVM based classification are ran on one of the ARM processor cores; the other processor core is used for universal codec functions.

The HOG related processing module (as shown in Figure 2(A)) is generated by Vivado HLS software. Although this is a relatively less computational expensive function, we aim to demonstrate that the facility of exploring design space for any given image-processing applications and increasing re-usability. The hardware resources required by Xilinx ZC-702 chip for HOG and SVM based pedestrian detection only

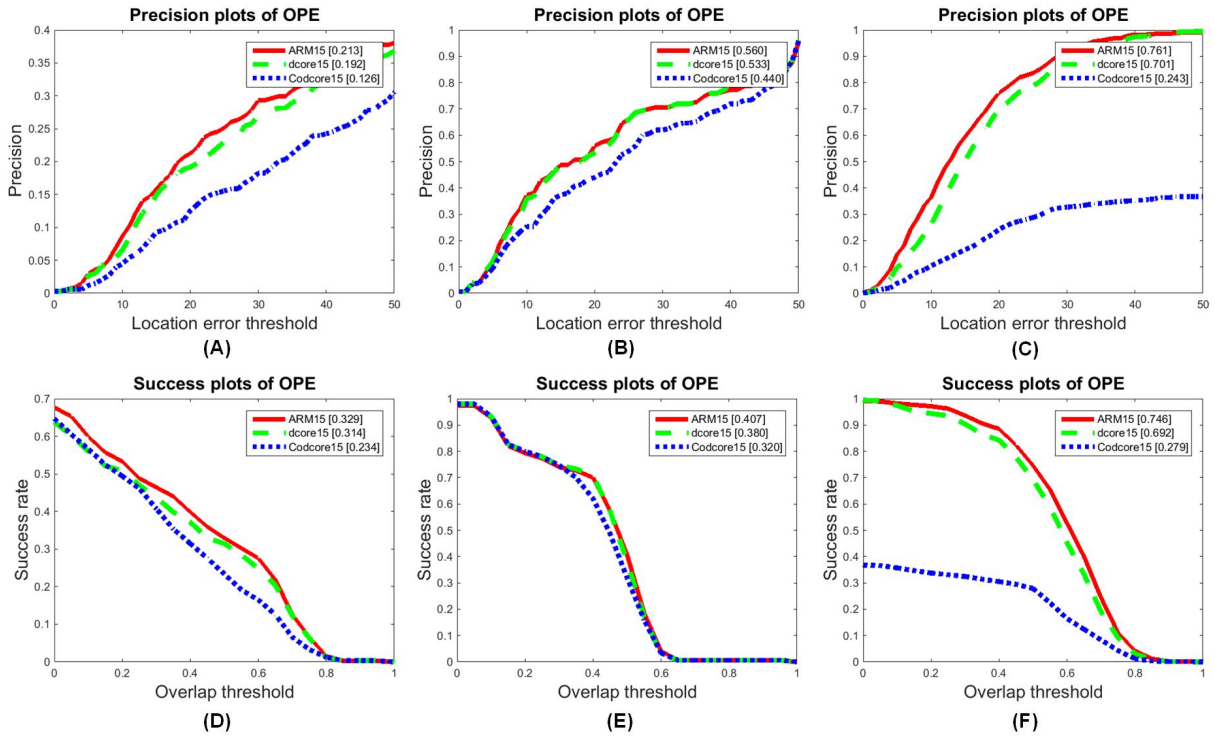

Fig. 4. Precession rates of four image datasets (A) precision rates of blur_body scene (B) precision rates of dance2 scene (C) precision rates of gym scene (D) success rates of blur_body scene (E) success rates of dance2 scene (F) success rates of gym scene

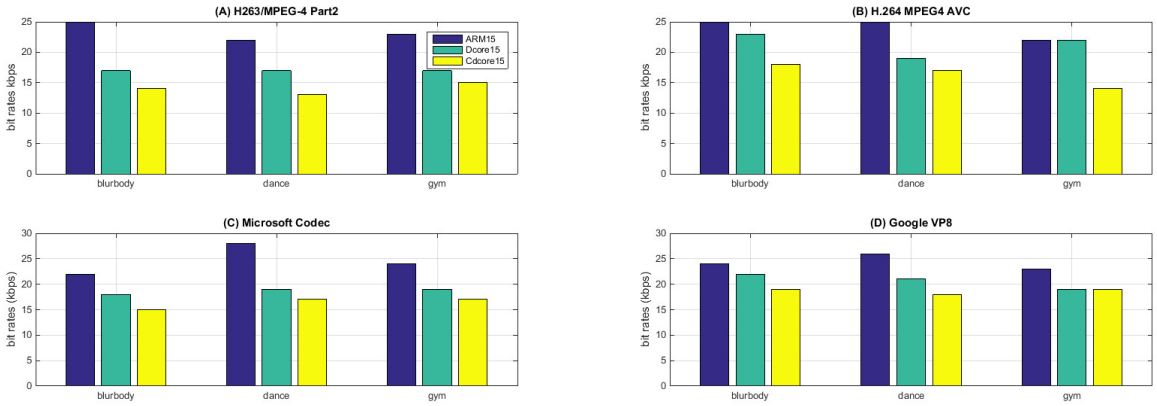

Fig. 5. Processing speed of four video decode with four video codec formatse (A)The decoding speed using H.263 codec (B)The decoding speed using H.264 MPEG4 codec (C)The decoding speed using Microsoft codec (D)The decoding speed using Google VP8 codec

used 29% of LUT and 14% of BRAM. It is noted that the hardware utilization forecast from Vivado HLS will be changed after RTL (generated hardware description language) level generation; hence it is necessary that we keep a margin for hardware device choice for a design optimisation.

C. Experiment Setup and Evaluation

The performance of our system is evaluated by measuring the encoding rates and the precession rates of pedestrian tracking tasks. All timing and profiling procedures for measurements are implemented with the ARM cortex A9 clock frequency, 800MHz. A histogram of oriented gradients (HOG) algorithm based pedestrian tracking system was implemented using a Xilinx ZYNQ-7000 chip. For evaluation processes, we adapted the most popular pedestrian tracking data sets to evaluate our hardware-software co-design platform [11]. Three typical real-life scene sequences with four video codec is used to evaluate the speed of encode and detection precision for pedestrian tracking. The detail of the image data sets can be found in Appendix. The three video sequences with four video formats were chosen to test real-life scenario with video resolution's range from QVGA (320 x 240) to VGA (640 x 480).

Although Wu, et al (2013) proposed a benchmark for testing visual pedestrian tracking: precision rates, success rates and

robustness. In this work, we focus on testing the real-time image processing performance instead of testing the tracker performance. Therefore, we only analysed precision rates and success rates. The precision rates indicate the ratio of frames with an average centre location error below a threshold to the ground truth from a visual tracker benchmark [11]. The success rates indicate the ratio of successful frames at the thresholds varied from 0 to 1 compared to the ground truth.

In these experiments, we compared three approaches: the results from software only single ARM cortex-A9 processor, the dual ARM cortex-A9 processors approach and, finally, FPGA co-design approach. In Figure 4, the ARM15 curves indicate the experiment using duel ARM Cortex A9 processors. The Dcore curves indicates the experiments of using single ARM cortex core and the Codcore curves indicates the hardware-software co-design approach for pedestrian tracking. Overall, the best result was from the gym sequence (Figure 4 (C)) in those three experiments. Since the size of pedestrian objects in the video sequence blur_body, dance2 and gym scene are 10% smaller than the size of detection windows (64 pixels by 128 pixels). It is noted that the results of video sequence of gym and dance2 scene have the same increasing rates for all three platforms. The precession rates of FPGA Co-design platforms (with legend Codcore15 in figure 4) are better than single ARM cortex-A9 system (With legend ARM15 in figure 4) and dual ARM cortex A9 processors (with legend dcore15 in figure4) due to the FPGA hardware acceleration advantage. Resultant success plots for those three scenes are shown in Fig.4 (D-F), which shows our implementation is in both scale and aspect ratio adaptability.

Taking advantage of HLS implementation a further optimise can be explored on memory utilisation. The performance of different video encoders are presented in figure 5 with DAM block transferring provides 2x further speed up compared to initial version of our FPGA co-design approach. Our measurements were carried out with four popular video codec formats: H.263/MPEG-4 Part2, H.264/MPEG-4 AVC, Microsoft codec and Google (on2) VP8 codec. The average decoding frame FPS_avg is computed as $FPS\_avg(n) = f\_cpu/c\_frame(n)$. Here the f_cpu is the CPU frequency, and $c\_frame(n)$ denotes average decoding cycles per VGA frame. The bit rate reported in Figure 5 is relatively stable on 30 frames/second, which meets the real-time image processing need.

A comparison for time-to-market (TTM) with event-based simulator to our co-design and co-evaluation platform can demonstrate a factor 5 times faster. Since our frame work is able to offer sufficient performance to run complete application software on top of the targeted real-time operation system based on ARM cortexA9 and FPGA programmable logic directly.

V. Conclusion

With a rapidly changing system-on-chip design industry, fast verification is needed to enable a higher productivity for embedded system products. Hybrid FPGA-SoCs co-verification architecture is an effective solution, providing designers the benefit of early stage hardware-software co-verification and integration between hardware and software, which in addition can be used as a direct test in the real environment.

Since the main concern of this work was the feasibility and scalability of verification platform for image and video multimedia processing, our C-like program was not compatible for standard C compilers. Since C language is not intrinsically concurrent, a special design high-level-synthesis compiler is required. We also did not use a hard wired codec for maximum performance, as it is important that our platform efficiently elevates the design cost of video codecs function using an interface to elegantly sharing video processing needs. Based on our current work, any new codec can be added into the platform with minimal efforts.

Nevertheless, this project presents a universal codec enabled verification platform with a software and hardware co-designed framework. We used a real-time case study to demonstrate the capability of evaluating real-time image processing algorithms using open access image datasets and, while doing so, we also overcame the problem of memory mapping from ARM cortx-A9 processor cores to FPGA buffers and successfully transplanted video streaming drivers into Linux V4L2 multimedia framework based on a Xilinx open source Linux operating system. Although we only demonstrated pedestrian tracking as a case study, any other embedded vision algorithm can be ran on this platform since the interfaces between general POSIX software interfaces and hardware accelerator are well defined. In this project, we also demonstrated how to take advantage of POSIX standard Linux software with vast amount of software libraries and traditional hardware to speed up time-to-market for system-on-chip designs.

Appendix A
Video Datasets

(1) blur_body: width:320 height:240 Features:SV, DEF, MB, FM, IPR.
link:
http://cvlab.hanyang.ac.kr/tracker_benchmark/seq/blur_body.zip
(2) dance2: width:624 height:352 Features:DEF.
link:
http://cvlab.hanyang.ac.kr/tracker_benchmark/seq/dance2.zip
(3) gym: width:352 height:288 Features:SV, DEF, IPR, OPR.
link:
http://cvlab.hanyang.ac.kr/tracker_benchmark/gym.zip

References

[1] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.
[2] M. Mefenza, N. Edwards, and C. Bobda, "Interface based memory synthesis of image processing applications in fpga," *ACM SIGARCH Computer Architecture News*, vol. 43, no. 4, pp. 64–69, 2016.
[3] P.-Y. Hsiao, S.-Y. Lin, and S.-S. Huang, "An fpga based human detection system with embedded platform," *Microelectronic Engineering*, vol. 138, pp. 42–46, 2015.
[4] J. Cerezuela-Mora, E. Calvo-Gallego, and S. Snchez-Solano, "Hardware/software co-design of video processing applications on a reconfigurable platform," in *Industrial Technology (ICIT), 2015 IEEE International Conference on*. IEEE, 2015, pp. 1694–1699.
[5] G. Bieszczad, T. Sosnowski, H. Madura, M. Kastek, and J. Barela, "Adaptable infrared image processing module implemented in fpga," in *SPIE Defense, Security, and Sensing*. International Society for Optics and Photonics, 2010, pp. 76 603Z–76 603Z–10.

[6] A. Menon, J. R. Santos, Y. Turner, G. J. Janakiraman, and W. Zwaenepoel, "Diagnosing performance overheads in the xen virtual machine environment," in *Proceedings of the 1st ACM/USENIX international conference on Virtual execution environments*. ACM, 2005, pp. 13–23.
[7] X. Zynq, *Zynq-7000 All Programmable SoC ZC702 Base Targeted Reference Design. v9.0*, 2015.
[8] I. LogiCORE, "Axi memory mapped to stream mapper v1.10."
[9] A. Cox, "Video4linux programming," *alan@ redhat.com*, 2000.
[10] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1. IEEE, 2005, pp. 886–893.
[11] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 2411–2418.

Shufan Yang is a Lecturer at University of Glasgow, United Kingdom. She was awarded PhD degree from the University of Manchester, United Kingdom. Her main research interestes include artificial neural nework, embedded and intelligent systems, programmable architectures and image processings. She has co-authoured over 25 peer-reviewed papers.

Zheqi Yu is a PhD candidate at University of Wolverhampton, United Kingdom. He was awared M.S.c in University of Wolverhampton in 2014. His research interestes include embedded systems, medical image analysis and programmable devices.