

부분 재구성을 이용한 노이즈 영상의 경계선 검출 시스템

윤일중¹, 정희원¹, 김승종², 민병석³, 이주흥^{*}

¹홍익대학교 전자전산공학과, ²한양여자대학교 컴퓨터정보과, ³충청대학교 전자컴퓨터학부

Edge Detection System for Noisy Video Sequences Using Partial Reconfiguration

Il-Jung Yoon¹, Hee-Won Jung¹, Seung-Jong Kim², Byong-Seok Min³, Joo-Heung Lee^{*}

¹Department of Electronics and Computer Engineering, Hongik University

²Department of Computer Science & Information System, Hanyang Women's University

³Department of Digital Electronic Communications, Chung Cheong University

요약 본 논문에서는 Zynq SoC 플랫폼을 사용하여 노이즈 영상의 경계선 검출 및 노이즈 감소를 위한 부분 재구성 시스템을 설계한다. 실시간 1080p 영상 시퀀스의 처리를 위한 높은 연산량을 제공하기 위해 재구성이 가능한 Programmable Logic 영역을 사용하고 하드웨어 필터를 구현한다. 또한 하드웨어 필터들은 부분 재구성 가능한 영역을 활용한 자동 재구성 기능을 통해 제한된 환경의 임베디드 시스템에서 더욱 더 효과적으로 하드웨어 자원 활용을 가능하게 한다. 주어진 한계점을 넘는 잡음을 포함한 입력 영상의 경우 적응적 노이즈 제거를 위한 필터링 연산을 하드웨어에 자동 재구성하여 수행함으로써 제안된 시스템은 향상된 경계선 검출 결과를 보여 주고 있다. 제안 하는 시스템을 사용하여 영상 시퀀스의 잡음 밀도에 따라 영상 처리 필터의 bitstream이 스스로 재구성 되었을 때 경계선 검출의 정확도에 대한 결과가 향상된 것을 (14~20배 PFOM) 구현 결과에서 보여 준다. 또한, ZyCAP을 사용하여 구현 한 경우 2.1배 빠르게 부분 재구성을 확인하였다.

Abstract In this paper, the Zynq system-on-chip (SoC) platform is used to design an adaptive noise reduction and edge-detection system using partial reconfiguration. Filters are implemented in a partially reconfigurable (PR) region to provide high computational complexity in real-time, 1080p video processing. In addition, partial reconfiguration enables better utilization of hardware resources in the embedded system from autonomous replacement of filters in the same PR region. The proposed edge-detection system performs adaptive noise reduction if the noise density level in the incoming video sequences exceeds a given threshold value. Results of implementation show that the proposed system improves the accuracy of edge-detection results (14~20 times in Pratt's Figure of Merit) through self-reconfiguration of filter bitstreams triggered by noise density level in the video sequences. In addition, the ZyCAP controller implemented in this paper enables about 2.1 times faster reconfiguration when compared to a PCAP controller.

Keywords : Edge Detection, Partial Reconfiguration, Pratt's FOM, Salt-and-pepper noise, Zynq SoC

1. 서론

영상의 경계선 검출은 처리해야 할 정보의 양을 줄여 주면서 중요한 구조적 정보를 유지 시켜주기 때문에 물체 검출, 얼굴 인식, 패턴 인식 등 여러 분야에서 적용되고 있으며, 인식 및 검출에 관련한 대부분의 영상 전처리

과정으로 활용되고 있다. 그러나 영상이 salt-pepper 잡음에 훼손되었을 경우 경계선 검출의 성능이 급격하게 저하된다. 이러한 문제점을 방지하기 위하여 미디언 필터 (median filter)와 같은 잡음제거 필터들이 경계선 검출 전 전처리 과정으로 활용된다. 최근의 영상처리 시스템의 기반이 고속, 고성능의 컴퓨터에서 휴대폰이나

이 논문은 2016학년도 홍익대학교 학술연구진흥비에 의하여 지원되었음.

*Corresponding Author : Joo-Heung Lee(Hongik Univ.)

Tel: +82-44-860-2544 email: joolee@hongik.ac.kr

Received October 5, 2016

Revised November 11, 2016

Accepted January 6, 2017

Published January 31, 2017

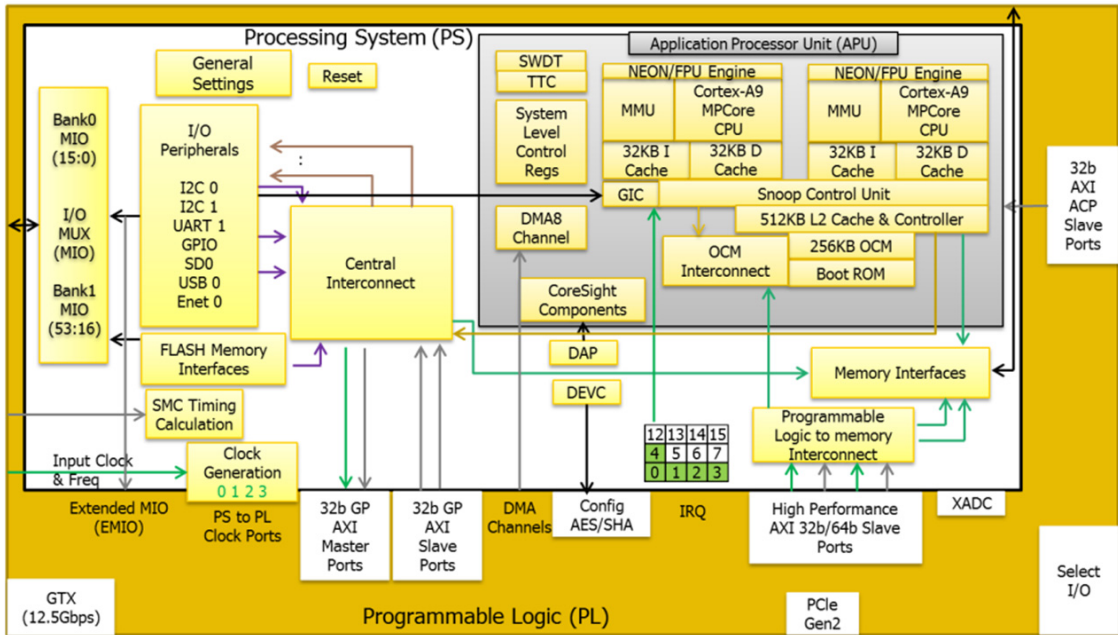


Fig. 1. Zynq SoC Architecture

SoC (System on Chip)를 탑재한 소형 임베디드 환경으로 이동함에 따라 많은 제약사항을 고려한 설계가 이루어지고 있다. 특히 많은 데이터를 실시간 처리해야 하는 영상정보의 경우, CPU를 활용한 소프트웨어 처리는 매우 많은 클럭을 소모하기 때문에, 이의 효과적인 연산을 위하여 FPGA (Field- Programmable Gate Array)와 같은 하드웨어를 이용하여 처리함으로써 각 응용에 최적화된 병렬처리 구조와 파이프 라이닝을 통해 저속의 동작 주파수에서도 영상정보의 실시간 처리가 가능하도록 시스템 레벨에서의 설계가 이루어지고 있다[1]. 이전에 FPGA와 고성능의 ARM Dual-core Cortex-A9 프로세서를 탑재한 Zynq SoC 플랫폼을 이용하여 영상의 고주파 에지 성분을 추출하기 위한 경계선 검출 필터를 구현하였고 [2], 이를 바탕으로 적응적 신호처리를 위한 복수의 영상처리 필터들이 floorplanning을 통해 지정된 하드웨어 영역을 함께 사용하도록 설계함으로써, 공통된 하드웨어 자원을 활용하여 여러 개의 영상처리 알고리즘들을 필요에 따라 실시간 선택하여 사용할 수 있는 PR (Partial Reconfiguration) 시스템을 구현하였다.

본 논문에서는 Zynq SoC 플랫폼을 이용하여 영상의 잡음을 제거한 후 경계선 성분을 추출하는 필터를 구현한다. 또한, 영상에 존재하는 잡음의 정도에 따라 노이즈

제거를 위한 필터를 자동 재구성하여 연산하는 시스템을 구현한다. 이를 통하여 임베디드 환경에서 제한적인 하드웨어 자원을 더욱 효과적으로 활용할 수 있으며, 지금까지의 연구 방향은 사용자의 선택에 따른 수동적 재구성을 전제로 이루어져 왔지만 본 논문에서 제안한 방법은 필터를 입력신호의 특성을 고려하여 스스로 재구성함으로써 잡음에 의해 훼손된 영상이 입력될 경우 이에 따른 실시간 대응이 가능하여 효과적인 경계선 검출을 수행할 수 있다.

2장에서는 Zynq SoC Platform에 대한 설명과 필터의 자동 재구성을 위해 제안된 알고리즘을 기술한다. 3장에서는 실험 결과를 기술하고, 4장에서 결론을 서술한다.

2. 본론

2.1 Zynq SoC

2.1.1 Zynq SoC Platform

성능이 낮은 프로세서를 지원하는 기존의 FPGA와는 다르게 reconfigurable SoC platform을 대표하는 Zynq-7000 AP SoC는 ARM 프로세서와 Xilinx

7-Series 의 FPGA가 결합된 hybrid FPGA platform 이다[3]. Fig. 1에서 보여주고 있는 Zynq 내부 구조는 PS 와 PL로 구성되어 있다. PS는 UART, USB, CAN, SPI, I2C 와 같은 다양한 built-in peripherals 과 ARM dual-core Cortex-A9 MP core, caches, DMA Controller 로 구성되어 있고 PL은 다음과 같은 자원을 가지고 있다 : Configurable Logic Blocks (CLB), Digital Signal Processing (DSP) Blocks, Analog-to-Digital converters, serial transceivers. Advanced Microcontroller Bus Architecture (AMBA)는 PL과 PS 그리고 IP (Intellectual Property)의 원활한 연결을 위하여 SoC 설계에 사용되는 bus protocol이다[4]. AMBA의 대표적인 bus interface는 다음과 같다 : Advanced Peripheral Bus (APB), Advanced High Performance Bus (AHB) and Advanced eXtensible Interface (AXI). 특히 AMBA 3.0 Spec 에 속해있는 AXI는 write response channel 이 추가되어 있고, read/write가 동시에 가능하여 SoC의 고속 동작을 지원한다. 다양한 종류의 AXI port가 PS와 PL의 통신에 관여하고 종류는 다음과 같다 : 32bit General-Purpose (GP) master/slave interface, 32bit/64bit High-Performance interface, 64bit Application Processor Unit interface. AMBA 기반의 마이크로 컨트롤러는 일반적으로 AXI나 AHB를 시스템 버스로 사용하여 ARM CPU나 DMA 등을 연결하여 사용한다. 또한 저속이면서 저전력 주변장치와의 통신을 위하여 APB를 사용하며, AXI나 AHB에 브리지(Bridge)를 사용하여 연결한다.

2.1.2 Partial Reconfiguration (PR)

Xilinx FPGA에서 지원되는 PR은 재구성이 가능한 하드웨어 자원에서 특정부분을 Partially Reconfigurable Region (PRR) 으로 재정의 하여 미리 설계된 하드웨어 비트스트림을 다운로드하여 사용할 수 있도록 지원한다. 다양한 기능의 비트스트림들을 정의된 PRR에 실시간으로 재사용하여 하드웨어 자원을 효과적으로 활용할 수 있으며 이를 통한 전력 및 비용 감소를 얻을 수 있다 [5-9]. 기존의 full configuration의 주된 문제점인 configuration overhead는 full bitstream 보다 용량이 작은 partial bitstream을 PRR에 다운로드 함으로써 재구성에 필요한 시간을 단축시킬 수 있다. 하드웨어 Bitstream을 PL 영역으로 전달하기 위해 Zynq device는 두 타입

의 interface를 지원한다 : PL 스스로 재구성하는 방식의 Internal Configuration Access Port (ICAP), PS 영역에서 재구성 하는 Processor Configuration Access Port (PCAP)[10]. ICAP과 PCAP은 32-bits의 data width를 가지고 이론적으로 최대 400MB/s의 reconfiguration throughput을 지원한다. 본 논문에서는 PCAP을 사용하여 full bitstream과 partial bitstream 파일을 PL로 전송하며 그 과정을 Fig. 2에서 보여주고 있다. 외부 메모리 저장소인 SD 카드에서 받은 First Stage Boot Loader (FSBL)는 PS를 boot 시키고 full bitstream을 읽어서 PL의 Static Logic에 다운로드한다. 그리고 SD 카드에 바이너리 파일 형식으로 저장되어 있는 partial bitstream은 PS가 접근 가능한 DDR Memory로 옮겨지고 PR이 진행되면 필요한 기능에 해당하는 partial bitstream이 PL의 재구성 가능한 영역으로 다운로드 된다. 다운로드 되는 동안 하드웨어의 Static Logic은 동작을 멈추지 않는다[11].

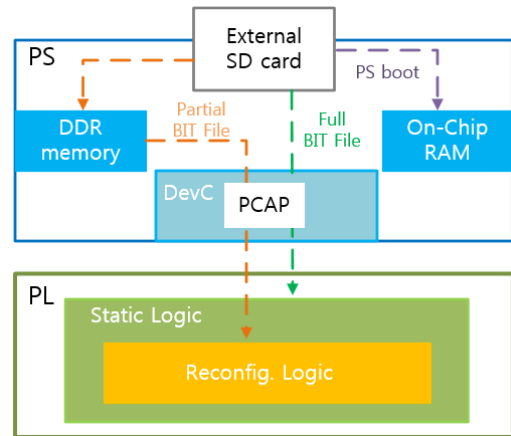


Fig. 2. PR interface using PCAP

2.2 제안 알고리즘

1080p 해상도의 비디오 영상에서 물체의 경계선을 실시간으로 추출하기 위해 PL 영역에서 Sobel filter를 구현하였다. Fig. 3은 각각 2차원 형태의 수직, 수평 Sobel 연산자를 보여주며, 입력된 영상의 모든 픽셀에 컨볼루션되어 픽셀 값을 계산한다. $f(x,y)$ 는 (x,y) 지점에서의 픽셀 값을 의미한다[12].

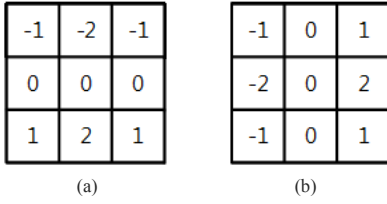


Fig. 3. Sobel Operator (a) Vertical Operator $S_1(k,l)$ (b) Horizontal Operator $S_2(k,l)$

$$G_1(x,y) = \sum_{k=-1}^1 \sum_{l=-1}^1 S_1(k,l) f(x+k, y+l) \quad (1)$$

$$G_2(x,y) = \sum_{k=-1}^1 \sum_{l=-1}^1 S_2(k,l) f(x+k, y+l) \quad (2)$$

식 (1)과(2)는 (x,y) 지점에서의 수직 연산자와 수평 연산자의 컨벌루션을 나타내며, 기울기의 크기와 방향은 아래의 식과 같이 나타낸다.

$$G(x,y) = \sqrt{G_1^2(x,y) + G_2^2(x,y)} \quad (3)$$

$$\theta_G(x,y) = \tan^{-1} \frac{G_2(x,y)}{G_1(x,y)} \quad (4)$$

경계선 검출은 위에서 정의한 Sobel 연산을 통하여 검출되지만 만약 영상을 촬영하는 센서에 결함이 발생하거나 영상을 전송하는 과정에서 bit error가 발생하여 Salt-and-Pepper와 같은 noise가 영상에 손상을 입히게 되면 경계선 검출 필터의 성능이 크게 감소하게 된다. 본 논문에서는 [13]에서 제안된 noise 검출 알고리즘을 구현하였으며, noise 검출 알고리즘은 다음과 같다.

$p(x,y)$ 는 (x,y) 지점의 픽셀 값이며, 3x3 window에서 중앙에 위치하게 된다. P_{\min} 과 P_{\max} 는 각각 3x3 window에서의 최소값과 최대값이며, threshold 값의 최소값인 T_{\min} 과 최대값인 T_{\max} 는 다음과 같이 정의된다.

$$T_{\min} = \begin{cases} 2P_{\min} - 255, & 2P_{\min} > 255 \\ 0, & otherwise \end{cases} \quad (5)$$

$$T_{\max} = \begin{cases} 2P_{\max}, & 2P_{\max} < 255 \\ 255, & otherwise \end{cases} \quad (6)$$

식(7)은 $p(x,y)$ 가 noise에 의해 손상된 픽셀인지 아닌지를 판별하는 기준으로 사용된다.

$$noise_{det} = \begin{cases} 1, & p(x,y) \leq T_{\min} \text{ or } p(x,y) \geq T_{\max} \\ 0, & otherwise \end{cases} \quad (7)$$

검출된 noise 픽셀의 총 수를 영상의 전체 픽셀수로 나눈 값이 주어진 영상 프레임의 noise 밀도가 된다. 영상이 Salt-and-pepper noise에 의해 손상되면 경계선 검출 성능이 많이 감소하게 되며, 경계선 검출 성능의 감소를 막기 위해서 noise를 제거하는 Median filter를 구현하였다. Median filter는 식 (8)의 정의와 같다.

$$M(x,y) = Median\{M_{x',y'} : (x',y') \in W(x,y)\} \quad (8)$$

window W안의 픽셀들의 median 값은 출력 값으로 선택된다[14].

Salt-and-pepper noise에 의해 손상된 영상에서 효과적인 경계선 검출을 실행하기 위해서 본 논문에서는 Median filter를 전처리 과정으로 사용하여 noise를 감소시킨 후 Sobel filter를 실행 (이후에는 Median+Sobel filter로 표기)하는 self-reconfiguration 방법을 제안하여 경계선 검출의 효율성을 높이고자 한다.

경계선 검출의 성능을 평가하기 위해 제안된 Pratt's Figure of Merit (FOM) 을 사용하여 경계선 검출에 방해가 되는 noise 밀도의 기준점을 결정하는데 사용하였다. Noise에 의해 경계선 검출의 정확도가 감소하게 되면, 필터영역이 재구성 되어 noise를 감소시킨 후에 경계선 검출 필터가 다시 실행되게 된다. Pratt's FOM은 다음과 같이 정의된다.

$$R = \frac{1}{I_N} \sum_{i=1}^{I_D} \frac{1}{1 + \alpha d^2} \quad (9)$$

식 (9)에서 $I_N = MAX(I_I, I_D)$ 이며, I_I 는 경계선이 라고 정의되는 픽셀의 총 수이며, I_D 는 실제로 검출된 경계선 픽셀의 총 수이다. α 는 1/9로 정의된 값이며, d 는 검출된 경계선과 정의된 경계선 사이의 거리 값이다 [15].

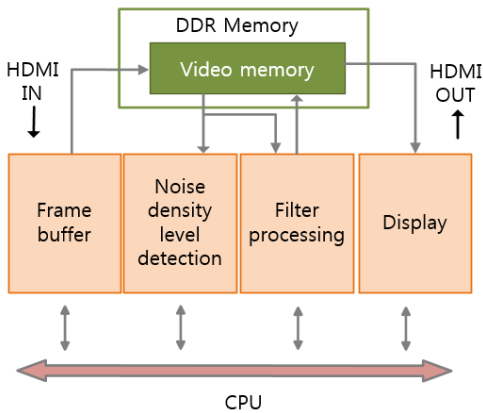


Fig. 4. Video pipeline and noise detection task

Fig. 4는 video pipeline 과 noise detection task를 보여준다. HDMI-IN으로 1080p 해상도의 영상이 들어오면 DDR memory에 저장된다. Noise density level detection 블록의 결과에 따라 Median+Sobel filter가 부분 재구성되며, 부분 재구성 과정은 PS에 의해 실행된다. 만약 noise density가 한계점보다 높게 측정되면 Median+Sobel bitstream이 Partially Reconfigurable Region (PRR)에 다운로드 되어 기존의 Sobel bitstream을 교체하게 된다. 필터링을 거친 영상은 display controller를 통해 HDMI-OUT으로 출력된다.

3. 실험 결과

실험에 사용되는 장비는 XC7Z020 CLG484-1 AP SoC를 장착한 ZC702 Evaluation board와 ADV7611/ADV7511 기반의 HDMI Input/Output을 지원하는 FMC Module, 그리고 1920X1080 해상도를 지원하는 monitor가 있다. UART Terminal Emulator를 사용하여 Board를 제어한다[16].

Fig. 5는 본 논문에서 제안하는 재구성 가능한 경계선 검출 시스템의 실험 환경이다. 실험에 사용하는 보드를 부팅하기 위해서 다음과 같은 파일들을 SD 카드에 포함한다. Software Development Kit (SDK) tool을 사용하여 생성된 FSBL, Vivado에서 생성한 full bit file 그리고 U-boot가 결합된 BOOT 바이너리 파일을 생성하고 리눅스 운영체제를 보드에 지원해 주는 압축된 커널 이미지인 uImage 또한 생성한다[17]. 리눅스 시스템을 사용하기 위한 uRamDisk와 PL영역을 재구성하기 위

한 Partial bit file을 생성한다. PCAP 인터페이스를 통한 PR을 수행하기 위해 partial bit file 또한 SD card에 저장한다.

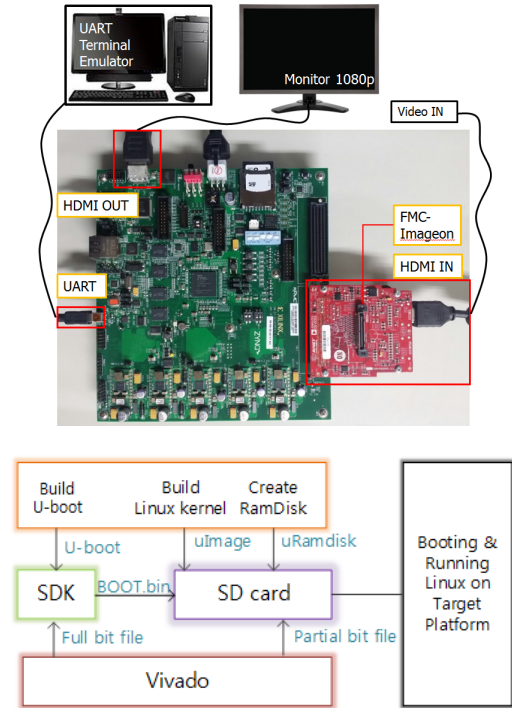


Fig. 5. Experimental environment

실험에 사용하는 Sobel 필터와 Median+Sobel 필터는 High-Level Synthesis (HLS) tool을 사용하여 생성된다 [18][19]. HLS는 C언어 또는 C++ 언어를 Register Transfer Level (RTL)로 구성된 Verilog 또는 VHDL 언어로 변경하고 IP core를 생성한다. 또한 tool의 GUI 인터페이스를 통하여 pipeline 기능을 제공한다. HLS를 사용하여 생성된 필터들은 필터 processing, 경계성 검출 processing, 그리고 비디오 영상의 입력력 제어를 위한 버스 인터페이스 세 종류의 기능으로 구성되어 있다. IP core 합성 결과, pipeline이 적용된 경계선 검출 기능의 latency는 9 clock cycle이 소모되었고 모든 기능을 사용하기 위한 processing time은 2,059 clock cycle이다.

Vivado Integrated Design Environment (IDE)는 Xilinx Integrated Synthesis Environment (ISE)와 Xilinx Platform Studio (XPS)을 지원하는 개발 tool이며 HDL 디자인을 분석하거나 합성하는 기능을 지원한다.

Fig. 6은 bitstream 생성의 전체적인 과정이다. HLS에 의해 생성된 필터 IP core와 전체 시스템의 HDL 디자인이 합성된다. 이 과정에서 재구성 가능한 영역인 PRR의 하드웨어 자원이 90%이하로 사용되도록 설정한다.

Table 1. Comparison of PRR and PRMs resources

Resources	PRR	PRMs	
	Available	SOBEL	M+S
LUT	7800	2871(36.8%)	3478(44.6%)
SLICE	1400	1054(75.3%)	1220(87.1%)
RAMB18	40	4(7.5%)	6(15.0%)
DSPs	40	0(0%)	0(0%)

Table 2. Bitstream configuration time

	Full Bitstream	Partial Bitstream
Bitstream Size	4,045,564 Byte	460,544 Byte
PCAP Configuration Time	83ms	10ms
ZyCAP Configuration Time	-	4.7ms

Fig. 7과 같이 Median+SOBEL의 PRM은 기존의 Sobel 필터에서 median 필터의 기능이 추가되어 Sobel의 PRM보다 많은 하드웨어 자원을 사용하고 있음을 Table 1에서 보여주고 있다.

시스템의 Implementation과 bitstream의 생성을 마치게 되면 결과적으로 full bitstream 하나와 partial bitstream 두 개가 생성된다. PL 영역에 처음으로 다운로드 되는 full bitstream은 static logic과 Sobel filter를 포함하고 있다. 비디오 영상에서 일정 수치 이상의 salt-pepper 노이즈가 발생할 경우 Median+SOBEL filter의 기능을 가진 partial bitstream 파일이 PCAP 인터페이스를 통하여 PRR에 다운로드 된다. 만약 일정 수치보다 낮은 salt-pepper 노이즈 밀도로 줄어들 경우 Sobel filter의 기능을 수행하는 partial bitstream으로 하드웨어가 재구성된다.

PR의 사용으로 인해서 bitstream의 메모리 용량과 PCAP configuration에 소모되는 시간 또한 감소되었음을 Table 2에서 보여주고 있다. 실시간 영상처리에서 중요한 부분인 configuration time은 full bitstream의 configuration에 사용되는 시간의 12%만을 사용하여

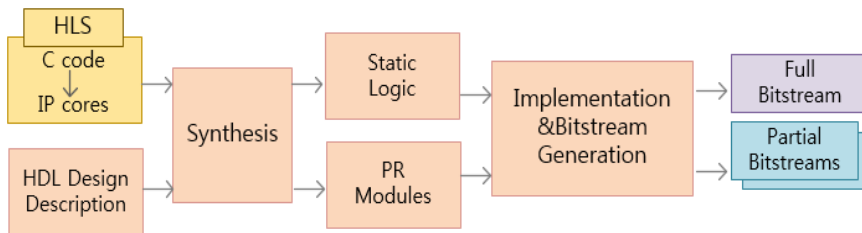


Fig. 6. The procedure of bitstream generation

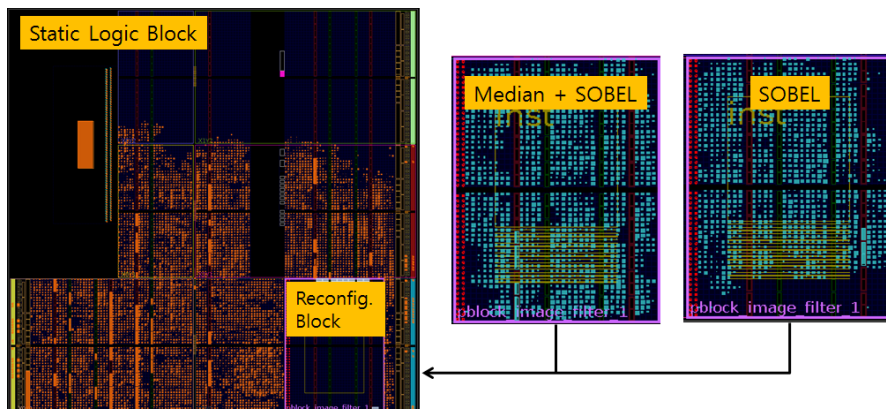


Fig. 7. Static logic and PR modules

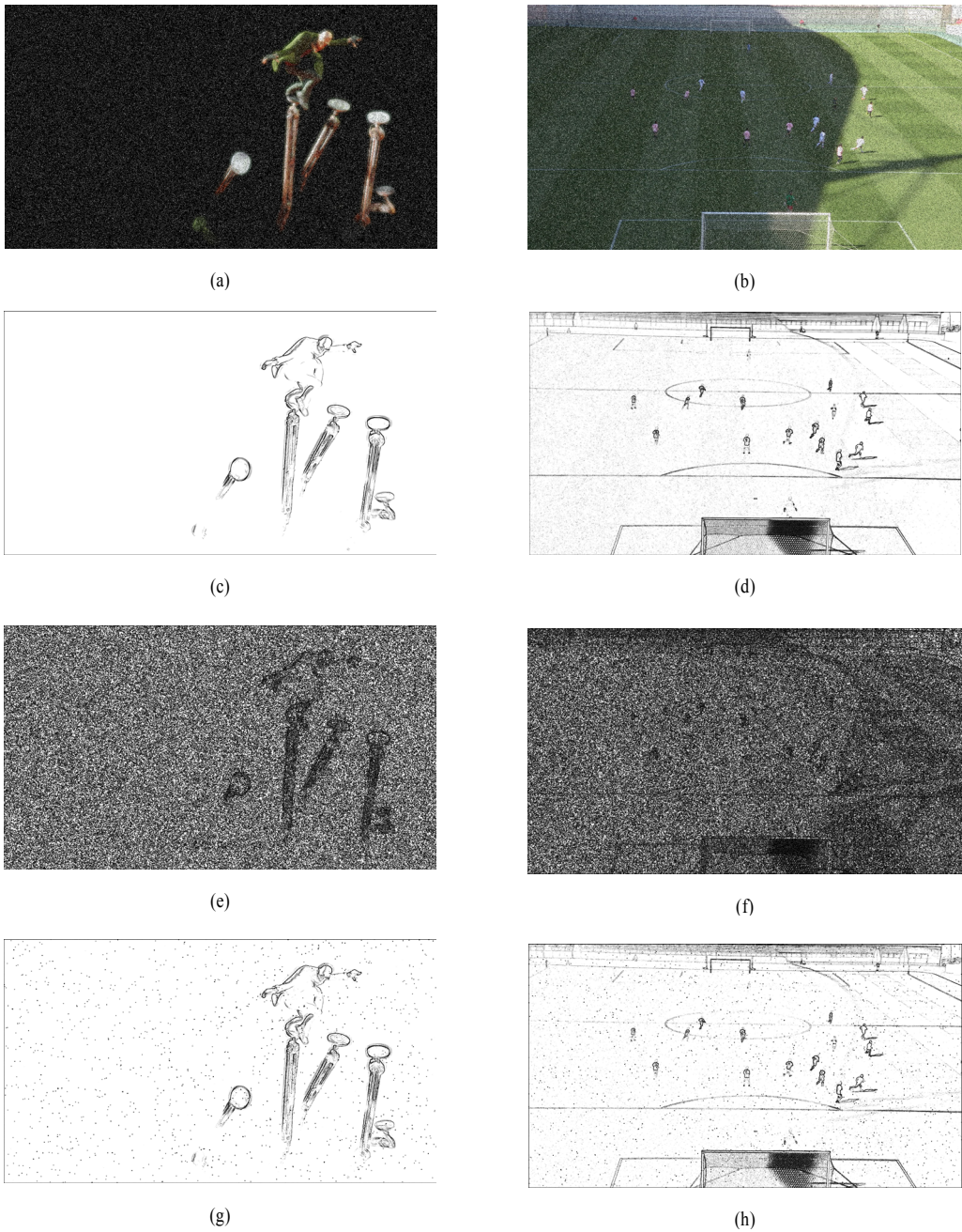


Fig. 8. Comparison of edge detection results for the noisy video sequences (a) Noisy video #1 (Salt-and-Pepper noise density : 20%) (b) Noisy video #2 (Salt-and-Pepper noise density : 20%) (c) Output of Sobel filter - original video #1 (d) Output of Sobel filter - original video #2 (e) Output of Sobel filter (noisy video #1) (f) Output of Sobel filter (noisy video #2) (g) Output of Median+Sobel filter (noisy video #1) (h) Output of Median+Sobel filter (noisy video #2)

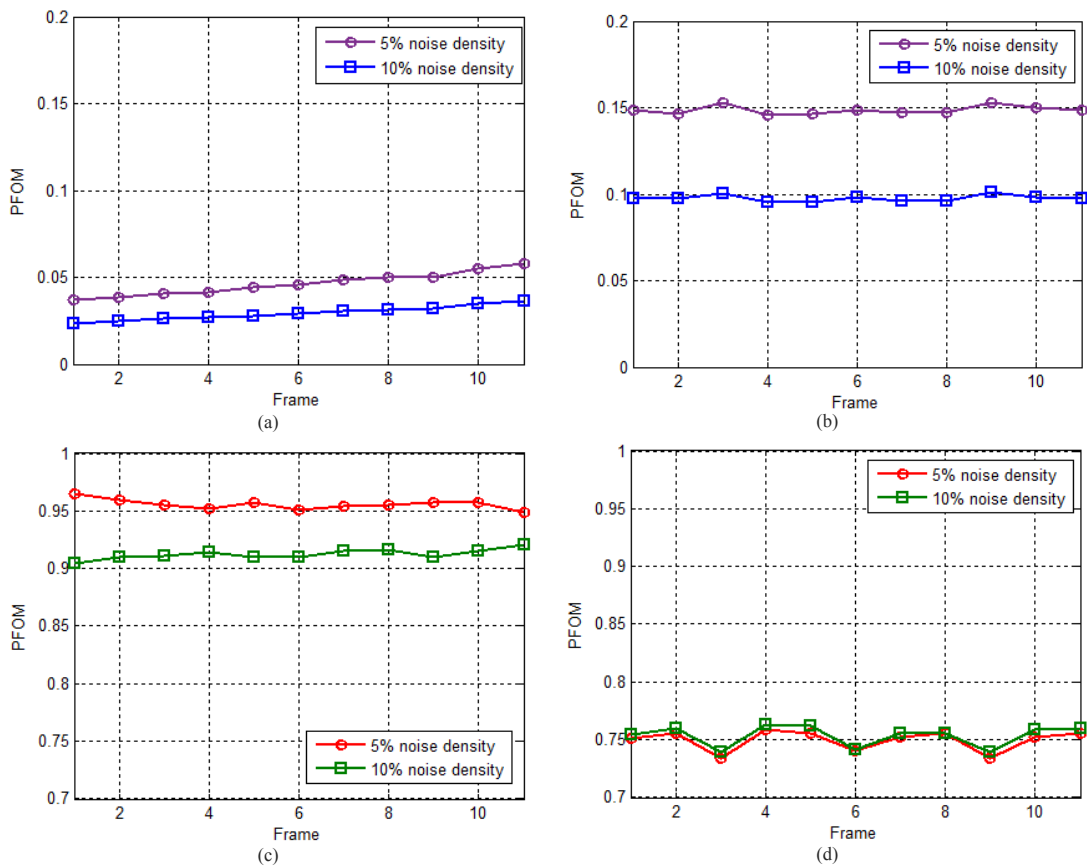


Fig. 9. PFOMs for Sobel and Median+Sobel filters (a) Sobel filter with Video #1 (b) Sobel filter with Video #2 (c) Median+Sobel filter with Video #1 (d) Median+Sobel filter with Video #2

partial bitstream의 configuraiton이 수행되었음을 확인하였다. 기존 Xilinx에서 제공하는 PR방식들 보다 좀 더 빠르게 부분 재구성을 수행하는 ZyCAP은 AMBA Bus의 HP Port를 효과적으로 사용하고 기존 재구성 방식에서 소프트웨어에 의한 overhead를 최소화 하여 PR을 수행하는 방법이며, 오픈소스로 제공된다[20]. 현재 플랫폼에 적용하여 재구성 시간을 측정한 결과, 4.7ms 로서 PCAP보다 2.1배 빠르게 재구성함을 확인하였다.

실험에 사용한 영상은 1920x1080의 해상도를 가지는 두 종류의 Full HD 비디오 영상이며 Fig. 8의 Video #1은 Elephants dream 이고 Video #2는 Soccer Game Goal 이다. Fig. 8-(a)와 Fig. 8-(b)는 기존 Video #1과 #2가 20%의 salt-and-pepper 노이즈 밀도에 의해 훼손된 영상이다. 훼손되지 않은 기존 영상에 대한 Sobel filter의 경계선 검출 결과 영상이 Fig. 8-(c)와 Fig. 8-(d)이다. 경계선 검출의 결과가 매우 깨끗함을 알 수 있다.

반면에 Fig. 8-(a)와 Fig. 8-(b)의 Sobel filter의 결과 영상인 Fig. 8-(e)와 Fig. 8-(f)는 경계선 검출의 성능이 매우 낮음을 PFOM을 통해 확인 할 수 있다.

Fig. 8-(g)와 Fig. 8-(h)는 Median+Sobel filter를 거친 결과 영상이고 Sobel filter 만을 사용한 결과영상 보다 노이즈 제거에 탁월하고 경계선 검출 성능도 효과적인 것을 확인 할 수 있다.

Sobel filter와 Median+Sobel filter의 경계선 검출 능력의 객관적인 비교의 지표로서 PFOM을 사용하여 판단한 그래프가 Fig. 9이다. 5%와 10%의 밀도를 가지는 salt-and-pepper 노이즈에 의해 훼손된 비디오 영상인 Video #1과 #2를 측정 대상으로 하고 frame 진행에 따른 Sobel filter와 Median+Sobel filter의 결과 영상에 따른 PFOM 수치를 측정하였다[21][22]. Median+Sobel filter의 PFOM 수치가 Sobel filter 보다 매우 높게 측정되어 잡음에 의해 훼손된 영상의 경우 보다 효과적으로

경계선 검출을 수행함을 확인할 수 있다.

Fig. 10의 CPU 사용량을 보면 Sobel 필터의 SW 구현 시 CPU를 100% 사용하고 있으나 Full HD급의 영상 프레임이 실제 끊기는 현상이 발생하며, HW 구현 시 CPU 로드를 최소화하며 실시간 연산 처리되는 결과를 확인할 수 있다.

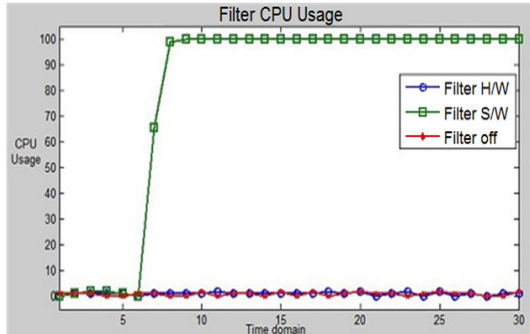


Fig. 10. CPU Usage

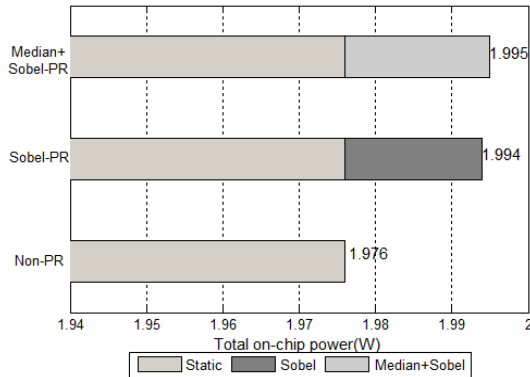


Fig. 11. Comparison of power consumption

Xilinx Zynq SoC 하드웨어 플랫폼의 전력소모량을 Vivado Design Suite에서 제공하는 Power Report를 사용하여 측정하였다[23]. Fig. 11은 각 모드 별 소모되는 전력량을 측정한 그래프이다. 하드웨어가 재구성되기 전의 non-PR 모드 일 때 전력 소모량은 재구성이 된 후의 전력 소모량보다 적다. 그리고 Sobel 모듈은 Median+Sobel 모듈 보다 더 적은 자원을 사용하기 때문에 상대적으로 적은 전력을 소모함을 알 수 있다.

4. 결론

본 논문에서 우리는 적응적 부분 재구성 시스템을 이용하여 경계선 검출 필터의 성능을 증가시키는 방법을 제안하였다. 필터 영역을 FPGA에서 하드웨어로 구현하여 1080p 해상도 영상의 경계선 검출에 필요한 연산 능력을 향상 시켰다. 영상에 noise가 검출되면 이를 처리하기 위한 다른 하드웨어 bitstream이 자동 재구성 되어 손상된 영상에서도 경계선이 효과적으로 검출됨을 확인할 수 있었다. 실험 결과를 보면 부분적 재구성이 전체 재구성에 걸리는 시간 보다 약 12% 감소함을 확인할 수 있으며, Median+Sobel filter를 사용한 경우 PFOM 수치가 Sobel filter만 사용한 경우보다 14 ~ 20배 정도 향상됨을 확인하였다. 이후 연구 진행 방향은, 하드웨어의 적응적 재구성을 더욱 효과적으로 지원할 수 있는 시스템 소프트웨어의 설계와 영상의 높은 잡음 밀도에 더욱 강한 필터를 설계 구현하고자 한다.

References

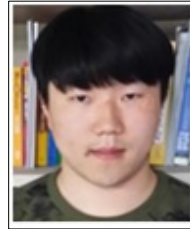
- [1] P. Greisen, M. Runo, P. Guillet, S. Heinzle, A. Smolic, H. Kaeslin and M. Gross, "Evaluation and FPGE Implementation of Sparse Linear Solvers for Video Processing Applications", Circuits and Systems for Video Technology, IEEE, vol. 23, Issue: 8, pp. 1402-1407, Feb. 2013. DOI: <https://doi.org/10.1109/TCSVT.2013.2244797>
- [2] UG1165 (v2015.3), "Zynq-7000 All Programmable SoC: Embedded Design Tutorial", Xilinx, Nov. 2015.
- [3] DS190(v1.8), "Zynq-7000 All Programmable SoC Overview", Xilinx, May 2015.
- [4] UG585(v1.10), "Zynq-7000 All Programmable SoC Technical Reference Manual", Xilinx, Feb. 2015.
- [5] UG909(v2014.4), "Vivado Design Suite User Guide Partial Reconfiguration", Xilinx, Nov. 2014.
- [6] E. Stott, P. Sedcole, P. Y. K. Cheung, "Fault tolerant methods for reliability in FPGAs", International Conference on Field Programmable Logic and Applications, pp. 415-420, Sept. 2008. DOI: <https://doi.org/10.1109/fpl.2008.4629973>
- [7] Naveed Imran, Ronald F. DeMara, Jooheung Lee, Jian Huang, "Self-Adapting Resource Escalation for Resilient Signal Processing Architectures", Journal of Signal Processing Systems, vol. 77, no. 3, pp. 257-280, Dec. 2014. DOI: <https://doi.org/10.1007/s11265-013-0811-x>
- [8] C. Insaurralde, "Reconfigurable computer architectures for dynamically adaptable avionics systems", IEEE Aerospace and Electronic Systems Magazine, vol. 30, pp. 46-53, Sept. 2015.

DOI: <https://doi.org/10.1109/MAES.2015.140077>

- [9] UG909(v2014.4), "Vivado Design Suite User Guide Partial Reconfiguration", Xilinx, Nov. 2014.
- [10] Kizheppatt Vipin and Suhaib A. Fahmy, "ZyCAP: Efficient Partial Reconfiguration Management on the Xilinx Zynq", Embedded Systems Letters, IEEE, vol. 6, Issue: 3, pp. 41-44, Sep. 2014.
DOI: <https://doi.org/10.1109/LES.2014.2314390>
- [11] XAPP1159(v1.0), Christian Kohn, "Partial Reconfiguration of a Hardware Accelerator on Zynq-7000 All Programmable SoC Devices", Xilinx, Jan. 2013.
- [12] Soonjong Jin, Wonki Kim and Jechang Jeong, "Fine Directional De-interlacing Algorithm Using Modified Sobel operation", Consumer Electronics, IEEE, vol.54, Issue: 2, pp. 587-862, May. 2008.
DOI: <http://dx.doi.org/10.1109/TCE.2008.4560171>
- [13] Pei-Yin Chen, Chih-Yuan Lien, and Yi-Ming Lin, "A Real-time Image Denoising Chip", Circuits and Systems, ISCAS IEEE International Symposium on, pp. 3390-3393, May. 2008.
DOI: <http://dx.doi.org/10.1109/ISCAS.2008.4542186>
- [14] Chenglong Chen, Jiangqun Ni and Jiwu Huang, "Blind Detection of Median Filtering in Digital Images: A Difference Domain based Approach", Image Processing, IEEE, vol. 22, Issue: 12, pp. 4699-4710, Aug. 2013.
DOI: <https://doi.org/10.1109/TIP.2013.2277814>
- [15] William K. Pratt, "Digital Image Processing", PIKS Inside, Third Edition, pp. 490-494, Aug. 2000.
- [16] XAPP1231 (v1.1), Christian Kohn, "Partial Reconfiguration of a Hardware Accelerator with Vivado Design Suite for Zynq-7000 Ap SoC Processor", Xilinx, Mar. 2015.
- [17] UG821 (v12.0), "Zynq7000 All Programmable SoC Software Developers Guide", Xilinx, Sep. 2015.
- [18] XAPP890 (v1.0), Fernando Martinez Vallina, Christian Kohn, and Pallav Joshi, "Zynq All Programmable SoC Sobel Filter Implementation Using the Vivado HLS Tool", Xilinx, Sep. 2012.
- [19] UG902 (v2015.4), "Vivado Design Suite User Guide : High-Level Synthesis", Xilinx, Nov. 2015.
- [20] Kizheppatt Vipin and Suhaib A. Fahmy, "ZyCAP: Efficient Partial Reconfiguration Management on the Xilinx Zynq", IEEE Embedded Systems Letters, vol. 6, Issue : 3, Mar. 2014.
DOI: <http://dx.doi.org/10.1109/LES.2014.2314390>
- [21] Ikram E. Abdou and William K. Pratt, "Quantitative design and evaluation of enhancement/thresholding edge detectors", Proceedings of the IEEE , vol. 67 , no. 5, pp. 753-763, May 1979.
DOI: <https://doi.org/10.1109/PROC.1979.11325>
- [22] J.-A. Jiang, C.-L. Chuang, Y.-L. Lu and C.-S. Fahn, "Mathematical-morphology-based edge detectors for detection of thin edges in low-contrast regions", Image Processing, IET, vol. 1, no.3 pp. 269-277, Sep. 2007.
DOI: <https://doi.org/10.1049/iet-ipr:20060273>
- [23] UG907(v2015.4), "Vivado Design Suite User Guide: Power Analysis and Optimization", Xilinx, Nov. 2015.

윤 일 중(II-Jung Yoon)

[준회원]



- 2014년 2월 : 홍익대학교 전자전 기공학과 졸업
- 2014년 3월 ~ 현재 : 홍익대학교 일반대학원 전자전산공학과 석사 과정

<관심분야>
영상처리, 임베디드 시스템

정 희 원(Hee-Won Joung)

[준회원]



- 2014년 2월 : 홍익대학교 전자전 기공학과 졸업
- 2014년 3월 ~ 현재 : 홍익대학교 일반대학원 전자전산공학과 석사 과정

<관심분야>
임베디드 시스템, IoT

김 승 중(Seung-Jong Kim)

[종신회원]



- 1994년 2월 : 한양대학교 대학원 전자통신공학과 (공학석사)
- 2000년 8월 : 한양대학교 대학원 전자통신공학과 (공학박사)
- 2000년 3월 ~ 2000년 8월 : (주)VI 선임연구원
- 2000년 9월 ~ 현재 : 한양여자대학교 컴퓨터정보과 교수

<관심분야>
멀티미디어 영상처리, 디지털 통신 및 신호처리

민 병 석(Byong-Seok Min)

[정회원]



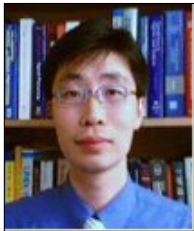
- 1990년 2월 : 한양대학교 전자통신공학과 졸업 (공학사)
- 1992년 6월 : 한양대학교 대학원 전자통신공학과 졸업 (공학석사)
- 2002년 6월 : 한양대학교 대학원 전자통신공학과 졸업 (공학박사)
- 1995년 3월 ~ 현재 : 충청대학교 전자통신전공 교수

<관심분야>

영상처리, 임베디드 시스템

이 주 흥(Joo-Heung Lee)

[정회원]



- 1998년 2월 : 한양대학교 대학원 전자통신공학과 (공학석사)
- 2006년 8월 : 펜실베이니아 주립대학교 전자공학과 (공학박사)
- 2006년 9월 ~ 2011년 8월 : 센트럴 플로리다 주립대학교 전자전산공학과 조교수
- 2011년 9월 ~ 현재 : 홍익대학교 전자전기공학과 부교수

<관심분야>

영상처리, 임베디드 시스템, SoC 설계(System on a Chip)