

EGML 기반 이동객체 검출 프로세서의 저면적 하드웨어 구현

성미지 · 신경욱*

A Small-area Hardware Implementation of EGML-based Moving Object Detection Processor

Mi-ji Sung · Kyung-wook Shin*

School of Electronic Engineering, Kumoh National Institute of Technology, Gumi, Kyungbuk 39177, Korea

요 약

EGML (Effective Gaussian Mixture Learning) 기반의 배경차분 기법을 이용한 이동객체 검출 (Moving Object Detection; MOD) 프로세서의 효율적인 하드웨어 구현 방식을 제안한다. 하드웨어 복잡도를 감소시키기 위해 배경 생성에 사용되는 일부 연산을 근사화하여 구현하였으며, 배경차분과 가우시안 계산의 나눗셈 연산에 사용되는 하드웨어 자원이 공유되도록 설계하였다. 설계한 MOD 프로세서는 MATLAB/Simulink를 이용한 HDL-netlist 시뮬레이션과 FPGA-in-the-loop 방식을 통해 기능을 검증하였다. IEEE CDW-2014 데이터 세트의 6가지 영상을 입력으로 사용하여 MOD 성능을 평가한 결과, 평균 재현율(recall)은 0.7700, 평균 정밀도(precision)는 0.7170, F-measure가 0.7293으로 평가되었다. Xilinx ISE를 이용하여 FPGA 합성한 결과, Virtex5 XC5VSX95T 디바이스에서 총 882 슬라이스와 146×36 kbit의 블록 램으로 구현되었으며, 동일한 알고리즘을 적용한 기존의 구현 사례에 비해 약 60%의 하드웨어를 감소시켰다. MOD 프로세서는 최대 75 MHz의 클럭 주파수로 동작하여 800×600 해상도의 영상에 대해 39 fps의 성능으로 실시간 처리가 가능한 것으로 평가되었다.

ABSTRACT

This paper proposes an efficient approach for hardware implementation of moving object detection (MOD) processor using effective Gaussian mixture learning (EGML)-based background subtraction method. Arithmetic units used in background generation were implemented using LUT-based approximation to reduce hardware complexity. Hardware resources used for both background subtraction and Gaussian probability density calculation were shared. The MOD processor was verified by FPGA-in-the-loop simulation using MATLAB/Simulink. The MOD performance was evaluated by using six types of video defined in IEEE CDW-2014 dataset, which resulted the average of recall value of 0.7700, the average of precision value of 0.7170, and the average of F-measure value of 0.7293. The MOD processor was implemented with 882 slices and block RAM of 146×36 kbits on Virtex5 FPGA, resulting in 60% hardware reduction compared to conventional design based on EGML. It was estimated that the MOD processor could operate with 75 MHz clock, resulting in real-time processing of 800×600 video with a frame rate of 39 fps.

키워드 : MOD, 이동객체 검출, EGML, 배경차분, GMM

Key word : MOD, moving object detection, EGML, background subtraction, GMM

Received 09 August 2017, Revised 17 August 2017, Accepted 16 November 2017

* Corresponding Author Kyung-wook Shin(E-mail:kwshin@kumoh.ac.kr, Tel:+82-54-478-7427)

School of Electronic Engineering, Kumoh National Institute of Technology, Gumi, Kyungbuk 39177, Korea

Open Access <https://doi.org/10.6109/jkiice.2017.21.12.2213>

print ISSN: 2234-4772 online ISSN: 2288-4165

©This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.
Copyright © The Korea Institute of Information and Communication Engineering.

I. 서 론

영상보안은 우리 주변에서 흔히 찾아볼 수 있을 정도로 필수적인 기술로 자리 잡고 있으며, 안전 및 보안뿐만 아니라 교통관리 시스템, 비즈니스 분야 등으로 영역이 확장되고 있다. 영상보안 기술의 보편화에 따라 효율적인 감시를 위한 방법들이 연구되고 있으며, 이동객체의 검출은 지속적인 감시, 신속한 조치, 효율적인 메모리 사용 등을 위한 핵심 요소기술로 인식되고 있다. 이동객체 검출은 입력된 영상으로부터 배경과 객체를 분류하는 것을 의미하며 이동객체 검출을 위한 MOD 알고리즘 또한 다양하게 존재한다.

MOD 알고리즘은 객체검출 방식에 따라 연속한 두 프레임의 차를 이용하는 차영상 기법, 화소 또는 영역간의 움직임 벡터를 이용한 광류 흐름 기법 (optical flow), 배경 모델과 현재 프레임의 화소 값 차이를 이용한 배경차분 기법 (background subtraction)으로 분류될 수 있으며, 다양한 방법들이 제안되고 있다[1]. 이 중에서 배경차분 기법은 복잡한 배경을 가지는 환경에서 차영상 기법보다 정확도가 높고, 광류 흐름 기법에 비해 연산량이 비교적 적다는 장점이 있다.

본 논문에서는 배경차분 기법의 GMM (Gaussian Mixture Model)[2] 알고리즘에 기반한 EGML[3] 알고리즘을 하드웨어로 구현하였다. GMM 알고리즘은 다중 가우시안 분포를 이용하여 혼합 배경 모델을 생성하는 방법으로, 여러 객체가 등장하거나 카메라가 흔들리는 환경에서도 우수한 이동객체 검출 성능을 보이지만 초기 배경학습 속도가 느리다는 단점이 있다. EGML 알고리즘은 매칭 횟수를 학습율에 반영하여 배경 학습이 효율적으로 이루어지도록 고안된 알고리즘이다. 경량화 구현을 위해 일부 연산기의 근사화 및 하드웨어 자원공유 기법을 적용하여 설계하였으며, MATLAB/Simulink를 이용하여 HDL-netlist 시뮬레이션을 통해 논리 기능을 검증하고, FPGA-in-the-loop 방식을 이용한 시뮬레이션을 통해 하드웨어 동작을 검증하였다. II장에서는 GMM 알고리즘 기반 EGML 알고리즘에 대해 간략히 설명하고, III장에서는 EGML 알고리즘 기반 MOD 프로세서의 경량화 설계에 대해 설명한다. 설계한 회로의 기능검증 및 성능평가에 대하여 IV장에서 기술하고, 마지막으로 V장에서 결론을 맺는다.

II. EGML 알고리즘

EGML 알고리즘은 가우시안 혼합 모델에 기반한 배경학습 기법 중 하나로, 배경학습 속도에 따라 성능의 차이가 발생하는 AGMM (Adaptive Gaussian Mixture Model)[4] 알고리즘의 단점을 개선하기 위해 제안된 알고리즘이다. 각 가우시안 별로 가중치를 업데이트하며, 입력 화소의 가우시안 매칭 빈도를 학습을 업데이트에 반영함으로써 효율적으로 배경을 학습하도록 한다.

EGML 알고리즘의 슈도 코드는 표 1과 같다[3]. p_j 는 입력 화소 x 에 대한 j 번째 가우시안 확률을 의미하며, w, g, μ, σ, T_o 는 각각 가중치, 가우시안 확률밀도함수, 평균, 표준편차, 임계 값을 나타낸다. 매칭 확률이 가장 높은 가우시안에 대해 $q_k=1$, 그렇지 않은 가우시안에 대해 $q_k=0$ 의 값을 가지게 되며, 승자독식 방법 (winner-take-all)에 의해 결정된 q_k 값은 가중치, 평균, 분산 값의 업데이트에 영향을 미친다.

배경모델 생성에 사용되는 파라미터는 식 (1) ~ 식 (5)에 의해 업데이트 된다. 여기서 α, η 는 학습율, c 는 입력 화소의 가우시안 매칭 횟수, σ^2 는 분산 값을 의미한다.

$$w_k(t) = (1 - \alpha) \cdot w_k(t-1) + \alpha \cdot q_k \quad (1)$$

$$c_k(t) = c_k(t-1) + q_k \quad (2)$$

$$\eta_k = q_k \cdot \left(\frac{1 - \alpha}{c_k} + \alpha \right) \quad (3)$$

$$\mu_k(t) = (1 - \eta_k) \cdot \mu_k(t-1) + \eta_k \cdot x \quad (4)$$

$$\sigma_k^2(t) = (1 - \eta_k) \cdot \sigma_k^2(t-1) + \eta_k \cdot (x - \mu_k(t-1))^2 \quad (5)$$

α 는 0 ~ 1 사이의 값을 가지는 고정된 학습율이며, α 가 클수록 파라미터의 업데이트 변화량이 높아져 학습 속도가 상승하고 α 가 작을수록 업데이트에 따른 파라미터의 변화량이 작아 노이즈가 입력되더라도 배경 학습이 비교적 안정적으로 이루어진다. 따라서 학습율 α 의 값은 영상의 환경에 따라 반복적인 실험을 통해 결정될 수 있다. η 는 가우시안 매칭 횟수 c 가 반영된 학습율이며, 평균과 분산 값의 업데이트에 영향을 미친다. 식 (3)에 따라 학습율 η 는 매칭 횟수가 적은 가우시안 생성 초기에 가장 큰 값을 가지며, 매칭 횟수가 많아질

Table. 1 Pseudo code of EGML algorithm

<p>Control variables : K, V_0, α, T_σ</p> <p>Initialization : $\forall_{j=1..K} \omega_j = 0, \mu_j = \infty,$ $\sigma_j = V_0, c_j = 0$</p> <p>While new data $x(t)$</p> <p>$\forall_{j=1..K}$</p> $p_j = \begin{cases} \omega_j \cdot g_j(x; \mu_j, \sigma_j) & \text{if } \frac{ x - \mu_j }{\sigma_j} < T_\sigma \\ 0 & \text{otherwise} \end{cases}$ <p>If $\sum_{j=1}^K p_j > 0$ Then</p> <p>For ($k = 0; k < K; k++$)</p> $q_k = p_k / \sum_{j=1}^K p_j$ <p>If <i>Winner-Take-All</i> Then</p> $q_k = \begin{cases} 1 & \text{if } k = \text{argmax}_j(p_j) \\ 0 & \text{otherwise} \end{cases}$ <p>End If</p> $w_k(t) = (1 - \alpha) \cdot w_k(t-1) + \alpha \cdot q_k$ <p>If $q_k > 0$ Then</p> $c_k(t) = c_k(t-1) + q_k$ $\eta_k = q_k \cdot \left(\frac{1 - \alpha}{c_k} + \alpha \right)$ $\mu_k(t) = (1 - \eta_k) \cdot \mu_k(t-1) + \eta_k \cdot x$ $\sigma_j^2(t) = (1 - \eta_k) \cdot \sigma_j^2(t-1) + \eta_k \cdot (x - \mu_k(t-1))^2$ <p>End If</p> <p>End For</p> <p>Else</p> $\forall_{j=1..K} w_j(t) = (1 - \alpha) \cdot w_j(t-1)$ $k = \text{argmin}_j(w_j)$ $w_j = \alpha, \mu_j = x, \sigma_j = V_0, c_j = 1$ <p>End If</p> <p>normalize w</p> <p>End while</p>

수록 작은 값을 가진다. 매칭 횟수 c 의 증가에 따른 학습을 η 의 변화량은 점점 감소하게 되며, 식 (4) ~ 식(5)에 따라 가우시안 생성 초기에는 파라미터 업데이트 변화량이 높아져 배경 학습 속도가 비교적 높아지고, 매칭 횟수가 많을수록 파라미터 업데이트 변화량이 작아져 배경학습의 안정도가 유지되도록 한다.

입력 화소 x 가 어느 가우시안에도 매칭 되지 않는 경우에는 가중치가 가장 낮은 가우시안의 파라미터 값이 초기화되어 새로운 가우시안을 형성하게 된다.

III. MOD 프로세서 설계

그레이 스케일의 8비트 화소 값을 입력받아 해당 화소가 배경이면 0, 객체면 1을 출력하는 MOD 프로세서를 경량화 설계하였다. 설계한 MOD 프로세서의 전체 구조는 그림 1과 같으며, MOD 연산 블록 (MOD_oper) 과 메모리, 제어 블록으로 구성된다.

MOD 연산 블록은 배경생성 블록 (BG_generator)과 객체검출 블록 (FG_detector)으로 구성되며, 배경생성 블록은 배경차분, 가우시안 확률밀도함수 계산, 가우시안 매칭, 파라미터 업데이트 연산을 수행한다. 객체검출 블록은 입력 화소의 가우시안 매칭 여부를 판단하고, 배경과 객체를 구분하는 기능을 수행한다. 제어 블록은 가변 파라미터 선택 신호를 입력받아 각 파라미터의 값을 설정하고, MOD 연산 블록이 4 클럭 주기에 걸쳐 연산을 수행하도록 제어 신호를 출력한다. 또한 입력 영상 해상도에 따라 메모리 주소를 제어하여 파라미터 값들을 올바른 주소에 읽고 쓸 수 있도록 한다. 메모리는 FPGA 내부의 블록 램을 사용하였으며, 제어신호에 따라 MOD 연산블록에서 업데이트된 파라미터 값을 저장한다.

저면적 MOD 프로세서 구현을 위해 다음과 같은 설계기법을 적용하였다. 첫째, 큰 하드웨어를 필요로 하는 일부 연산 (나눗셈 연산, 지수연산을 위한 상수 곱셈)을 근사화하여 구현하였다. 둘째, 배경차분과 가우시안 확률밀도함수 계산 단계의 일부 하드웨어 자원이 공유 되도록 설계하였으며, 이를 위해 하나의 화소가 4클럭 주기에 처리되도록 하였다.

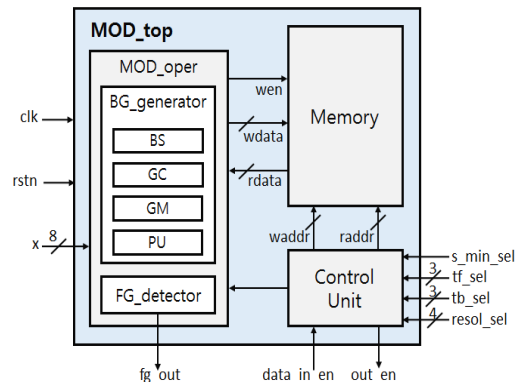


Fig. 1 Structure of MOD processor

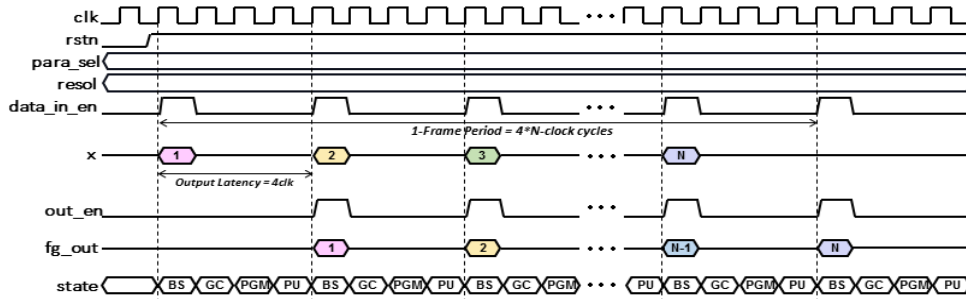


Fig. 2 Timing diagram of MOD processor

그림 2는 설계한 MOD 프로세서의 동작 타이밍 도이다. 그레이 스케일의 8비트 화소 값을 입력받아 첫 번째 클럭에서 배경 차분 (BS)이 계산되고, 두 번째 클럭에서 각 가우시안 확률밀도함수 계산 (GC)이 이루어지며, 이 결과에 각 가중치를 적용하여 입력 x 에 대한 가우시안 매칭 확률 계산과 매칭 여부를 판별하는 과정 (PGM)이 세 번째 클럭에서 수행되고, 네 번째 클럭에서는 파라미터 업데이트 및 객체 검출 (PU)이 이루어진다. 이와 같이 하나의 화소를 처리하는데 4 클럭주기가 소요되며, 이 과정이 완료되면 out_en 신호와 함께 입력 화소에 대한 배경 또는 객체판별 결과가 외부로 출력된다. N개의 화소로 구성되는 한 프레임을 처리하는데 총 4N 클럭 주기가 소요된다.

배경생성 블록의 구조는 그림 3과 같다. 3개의 BS-GC 블록은 배경차분과 가우시안 확률 계산을 병렬로 처리하며, 그 결과에 따라 가우시안 매칭 블록을 통해 매칭 여부가 판단된다. 가우시안 매칭 결과에 따라 3개의 PU 블록에서 파라미터가 업데이트되고, 그 결과가 메모리에 저장된다. 배경생성 블록 내부의 연산 복잡도에 따라 세부 연산에 소요되는 시간이 다르므로, 효율적인 동작을 위해 동작을 최적화 하였다. 연산량이 많은 가우시안 매칭확률 계산 (P_1, P_2, P_3)을 확률밀도함수 계산 (g_1, g_2, g_3)과 가중치 곱셈을 분리하여 두 클럭 주기에 연산되도록 하였으며, 연산량이 적은 가우시안 매칭이 가중치 곱셈과 동일 클럭에 처리되도록 하였다.

8비트 화소 값 x 가 BS-GC 블록으로 입력되면, 배경 차분을 수행하여 임계값과 비교한다. 두 번째 클럭과 세 번째 클럭에 가우시안 매칭 확률이 계산되며, 배경 차분 결과 값이 임계값 보다 작은 경우에만 가우시안 매칭확률 계산 값이 출력되고, 임계값보다 크면 0이 출

력된다. 이 출력 값에 의해 가우시안 매칭 연산을 수행되며, 매칭이 되었다면 네 번째 클럭에서 식 (1) - 식 (5)에 따라 파라미터가 업데이트되고, 매칭이 되지 않았다면 가중치가 가장 작은 가우시안의 파라미터 값이 초기화되어 메모리에 저장된다.

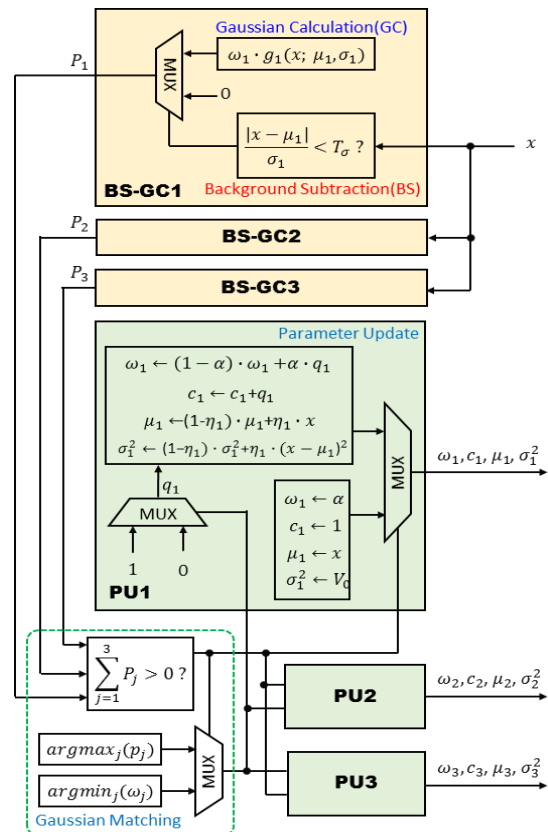


Fig. 3 Background generation block

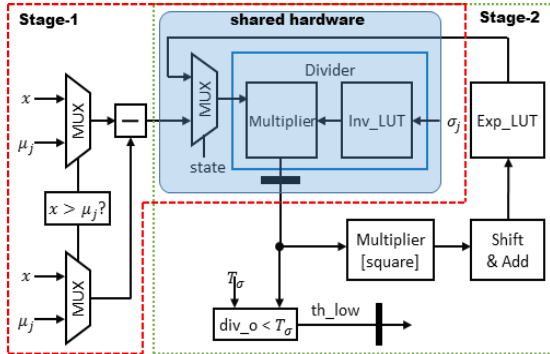


Fig. 4 Hardware sharing between BS and GPD calculation and LUT-based arithmetic approximation

하드웨어 공유 기법과 산술연산 근사화를 적용하여 설계한 BS-GC 블록의 일부는 그림 4와 같다. 배경차분 연산과 가우시안 확률밀도함수 계산에 필요한 나눗셈 연산을 직접 하드웨어로 구현하는 대신에 표준 편차의 역수로 이루어진 룩업 테이블과 곱셈기로 구현함으로써 회로의 동작속도가 개선되도록 하였다.

또한, 가우시안 확률밀도함수 계산에 사용되는 지수연산을 위한 상수 곱셈기는 2진 시프트와 덧셈으로 표현할 수 있는 가장 가까운 수로 근사화하는 방법을 적용하였으며, 이를 통해 시프트와 가산기만으로 구현 되도록 회로를 단순화하였다. 나눗셈 연산은 배경차분 연산 (stage-1)과 가우시안 확률밀도함수 계산 (stage-2)에 동일하게 사용되므로, 하나의 두 연산 단계에서 하나의 나눗셈기를 공유해서 사용하도록 설계하였으며, 이와 같은 자원공유 기법을 통해 하드웨어 복잡도를 줄였다.

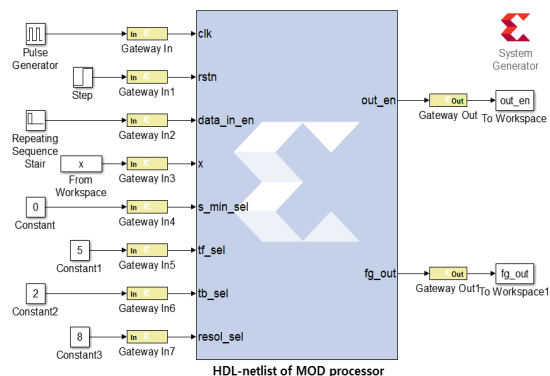
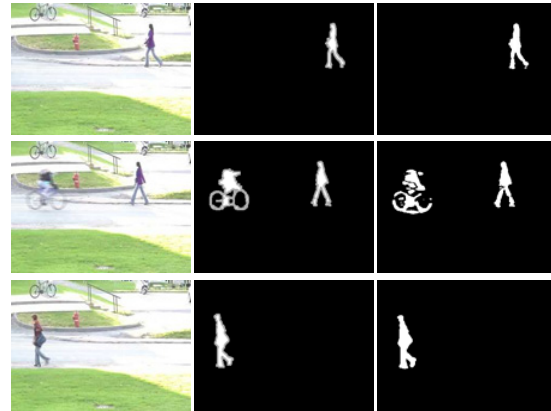


Fig. 5 HDL-netlist simulation using MATLAB/Simulink



(a) input (b) ground truth (c) output

Fig. 6 HDL-netlist simulation results

IV. 기능검증 및 성능평가

Verilog HDL을 이용하여 설계한 MOD 프로세서는 MATLAB/Simulink의 HDL-netlist 시뮬레이션을 통해 기능을 검증하였으며, 그림 5는 시뮬레이션 설정 화면을 보이고 있다. 펄스 생성기를 이용하여 클럭을 생성하고, 스텝을 이용하여 리셋을 인가하였다. 4 클럭마다 데이터가 입력되어야 하므로 반복 시퀀스를 이용하여 data_in_en 신호를 인가하였으며, 파라미터 선택 신호는 상수 값이 입력되도록 하였다. 8비트 입력 x 는 MATLAB을 통해 전처리를 거친 이미지의 화소 값을 워크스페이스로부터 순서대로 읽어 들여 인가하였다. 출력 데이터 fg_out은 워크스페이스에 시간 직렬 형태의 행렬로 출력된다.

HDL-netlist 시뮬레이션을 통한 MOD 기능 동작은 IEEE CDW-2014[5]의 데이터 셋을 이용하여 검증하였으며, 그림 6은 baseline 범주의 pedestrians 영상을 이용한 검증 결과이다. 입력 영상에 대하여 해당 화소가 배경인지 객체인지에 따라 0 또는 1이 출력되며, 배경은 검은 색, 객체는 흰 색으로 표시되는 흑백 영상이 출력되는 것을 확인하였다.

기능검증을 완료한 MOD 프로세서는 그림 7과 같이 FPGA-in-the-loop를 이용한 검증 환경을 통해 하드웨어 동작을 검증하였다. 입력 값은 HDL-netlist 시뮬레이션 검증과 동일하게 생성되어 JTAG 포트를 통해 FPGA

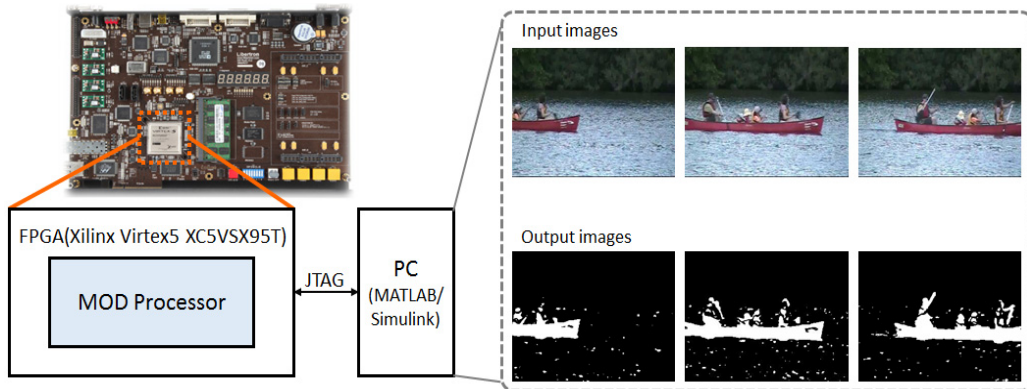


Fig. 7 FPGA verification results of MOD processor

로 입력된다. FPGA에서 수행된 MOD 연산결과는 다시 JTAG를 통해 PC로 전송되어 MATLAB의 워크스페이스로 출력된다.

CDW-2014 데이터 세트의 기준값 (ground truth)과 MOD 출력 결과를 비교하여 재현율 (recall), 정밀도 (precision), F-measure 값을 도출하는 방법을 통해 설계한 MOD 프로세서의 성능을 평가하였다. 성능 지표는 식 (6), (7), (8)에 의해 계산되며, TP (True Positive)는 객체에 해당하는 화소 중에서 객체로 올바르게 검출된 화소의 수, TN (True Negative)은 배경에 해당하는 화소 중에서 배경으로 올바르게 분류된 화소의 수, FP (False Positive)는 배경에 해당하는 화소를 객체로 오검출한 화소의 수, FN (False Negative)은 객체에 해당하는 화소를 배경으로 잘못 분류한 화소의 수를 나타낸다. 재현율과 정밀도가 1에 가까울수록 기준값과 유사성을 가지며, 두 값의 조화 평균인 F-measure가 1에 가까울수록 MOD 성능이 우수한 것으로 평가할 수 있다.

$$recall = TP / (TP + FN) \quad (6)$$

$$precision = TP / (TP + FP) \quad (7)$$

$$F\text{-measure} = (2 \times precision \times recall) / (precision + recall) \quad (8)$$

그림 8은 6가지 영상에 대해 알고리즘 종류에 따른 MOD 성능을 비교한 결과이다. CDW-2014 상위 등수에 해당하는 FTSG[6], CwisarDH[7] 알고리즘과 GMM 기반의 GMM-Stauffer & Grimson[2], GMM-Zivkovic [8] 알고리즘, CDW-2012[9]의 GMM-KaewTraKul Pong[4], GMM-RECTGAUSS-TEX[10] 알고리즘을 비교 하였다. 본 논문에서 구현한 EGML 기반 MOD 프로세서의 성능은 6가지 유형의 입력 영상에 대해 평균 재현율 0.7700, 평균 정밀도 0.7170, 그리고 F-measure는 0.7293으로 평가되었으며, 영상 유형에 따른 F-measure 값의 증감 추세가 다른 알고리즘과 유사한 양상을 갖는 것으로 확인되었다.

설계한 MOD 프로세서는 Xilinx ISE로 합성한 결과, 최대 75 MHz의 주파수로 동작이 가능하며, 800×600 해상도의 영상에 대해 39 fps의 성능으로 실시간 처리

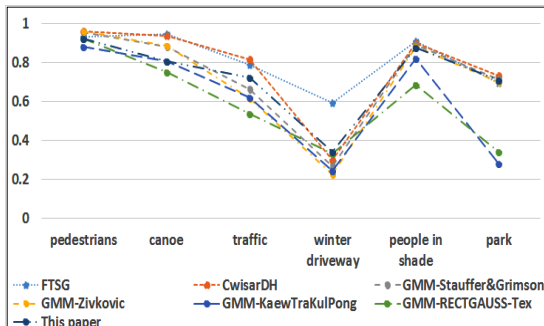


Fig. 8 Comparison of MOD performances(F-measure)

Table. 2 FPGA synthesis results

	[11, 12]	This paper	Reduction ratio
Registers	1,501	370	75.3%
LUTs	5,633	2,155	61.7%
Block RAM [36Kb]	151	146	3.3%
Total slices	2,218	882	60.2%
Max. Freq. [MHz]	102	75	-
Processing time [fps]	3,935	723	-

가 가능한 것으로 평가되었다.

표 2는 본 논문의 MOD 프로세서와 문헌[11, 12]에 발표된 설계 사례를 비교한 결과이다. 문헌 [11, 12]의 MOD 프로세서는 배경생성 블록이 총 13단계 파이프라인으로 동작하도록 설계되어 일정 레이턴시가 지나면 1 클럭당 1 화소씩 처리되므로 처리율이 높으나, 많은 하드웨어 자원을 필요로 한다. 본 논문에서는 4 클럭당 1 화소를 처리하도록 설계하여 처리율은 작으나, 총 882 슬라이스의 적은 하드웨어 자원으로 구현되었다. 본 논문에서 제안한 MOD 프로세서는 문헌 [11, 12]과 비교하여 약 40%의 하드웨어만 사용하므로, 저면적/저전력 구현이 필요한 분야에 적합하다[13].

V. 결 론

EGML 알고리즘 기반의 MOD 프로세서를 저면적 하드웨어로 구현하기 위한 설계방법을 제안하였다. 경량화 구현을 위해 일부 연산을 근사화하여 적용하였으며, 배경차분 단계와 가우시안 확률밀도함수 계산 단계에서 하드웨어 자원을 공유하도록 설계하여 하드웨어 복잡도를 크게 감소시켰다. 본 논문의 MOD 프로세서를 FPGA로 합성한 결과, 최대 동작주파수 75 MHz에서 총 882 슬라이스와 146×36 kbit의 메모리로 구현되었다. 본 논문의 MOD 프로세서는 지능형 감시 시스템과 같은 고정된 카메라 환경에서 저가형 이동 객체 검출 시스템에 적용될 수 있을 것이다. HD급의 1280×720 해상도의 영상에 대한 실시간 MOD를 위해서는 약 110 MHz 이상의 주파수로 동작해야 하며, 이를 위해 적합한 성능을 제공하는 FPGA 디바이스를 선택하여 구현하거나 또는 전용 칩으로 구현할 필요가 있다.

ACKNOWLEDGMENTS

- This paper was supported by Kumoh National Institute of Technology
- Authors are thankful to IDEC for supporting EDA softwares

REFERENCES

- [1] W. Hu, T. Tan, L. Wang and S. Maybank, "A Survey on Visual Surveillance of Object Motion and Behaviors," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 34, no. 3, pp. 334-352, Aug. 2004.
- [2] C. Stauffer and W. E. L. Grimson, "Adaptive Background Mixture Models for Real-time Tracking," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Fort Collins: CO, vol. 2, pp. 246-252, 1999.
- [3] D. S. Lee, "Effective Gaussian Mixture Learning for Video Background Subtraction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 5, pp. 827-832, Jun. 2005.
- [4] P. KaewTraKulPong and R. Bowden, "An Improved Adaptive Background Mixture Model for Real-time Tracking with Shadow Detection," in *Video Based Surveillance Systems*, Kluwer Academic Publishers, pp. 135-144, 2001.
- [5] Y. Wang, P. M. Jodoin, F. Porikli, J. Konrad, Y. Benezeth and P. Ishwar, "CDnet 2014: An Expanded Change Detection Benchmark Dataset," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, Columbus: OH, pp. 387-394, 2014.
- [6] R. Wang, F. Bunyak, G. Seetharaman and K. Palaniappan, "Static and Moving Object Detection Using Flux Tensor with Split Gaussian Models," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, Columbus: OH, pp. 414-418, 2014.
- [7] M. D. Gregorio and M. Giordano, "Change Detection with Weightless Neural Networks," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, Columbus: OH, pp. 409-413, 2014.
- [8] Z. Zivkovic, "Improved Adaptive Gaussian Mixture Model for Background Subtraction," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, Washington, DC, pp. 28-31, 2004.
- [9] N. Goyette, P. M. Jodoin, F. Porikli, J. Konrad and P. Ishwar, "Changetection.net: A New Change Detection Benchmark Dataset," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, Rhode Island: RI, pp. 1-8, 2012.
- [10] D. Riahi, P. L. St-Onge and G. A. Bilodeau, "RECTGAUSS -Tex : Block-based Background Subtraction," *Ecole Poly*

- technique de Montreal, Montréal: QC, Technical Report EPM-RT-2012-03, 2012.
- [11] G. H. Kim, "A Hardware Implementation of Moving Object Detection Algorithm using Background Subtraction based on EGML," M. S. dissertation, Kumoh National Institute of Technology, Gumi, Kyungbuk, 2015.
- [12] G. H. Kim, H. S. An and K. W. Shin, "A Hardware Implementation of EGML-based Moving Object Detection Algorithm," *Journal of the Korea Institute of Information and Communication Engineering*, vol. 19, no. 10, pp. 2380-2388, Oct. 2015.
- [13] M. J. Sung, "A Small-area Hardware Design of EGML-based Moving Object Detection(MOD) Algorithm for Intelligent Surveillance Systems," M.S. dissertation, Kumoh National Institute of Technology, Gumi, Kyungbuk, 2017.



성미지(Mi-Ji Sung)

2015년 2월 금오공과대학교 전자공학부(공학사)
2017년 8월 금오공과대학교 전자공학과(공학석사)
2017년 9월~현재 넥스트칩(주) ISP2팀 주임연구원
※관심분야 : 통신 및 신호처리용 반도체 IP 설계, 정보보호용 반도체 IP 설계



신경욱(Kyung-Wook Shin)

1984년 2월 한국항공대학교 전자공학과(공학사)
1986년 2월 연세대학교대학원 전자공학과(공학석사)
1990년 8월 연세대학교대학원(공학박사)
1990년 9월~1991년 6월 한국전자통신연구소 반도체연구단(선임연구원)
1991년 7월~현재 금오공과대학교 전자공학부(교수)
1995년 8월~1996년 7월 University of Illinois at Urbana-Champaign(방문교수)
2003년 1월~2004년 1월 University of California at San Diego(방문교수)
2013년 2월~2014년 2월 Georgia Institute of Technology(방문교수)
※관심분야 : 통신 및 신호처리용 SoC 설계, 정보보호 SoC 설계, 반도체 IP 설계