

환자움직임 감지를 위한 효율적인 하드웨어 및 소프트웨어 혼성 모드 영상처리시스템설계에 관한 연구[☆]

A study on the design of an efficient hardware and software mixed-mode image processing system for detecting patient movement

정 승 민¹ 정 의 성² 김 명 환^{3*}
Seungmin Jung Euisung Jung Myeonghwan Kim

요 약

본 논문에서는 환자와 같은 특정 객체의 움직임을 감지하고 추적하기 위한 효율적인 영상처리 시스템을 제안한다. 이진화된 차 영상에서 객체의 윤곽선추출을 위하여 기존 알고리즘대비 대비 정밀한 감지가 가능하고 혼성모드설계에 용이한 세선화 알고리즘을 적용하여 영역을 추출한다. 연산량이 많은 이진화와 세선화 단계를 RTL(Register Transfer Level) 기반으로 설계하여 논리회로 합성을 거쳐 최적화된 하드웨어 블록으로 대체된다. 설계된 이진화 및 세선화 블록은 표준 180nm CMOS 라이브러리를 이용하여 논리회로로 합성한 후 시뮬레이션을 통하여 동작을 검증하였다. 소프트웨어기반의 성능비교를 위해 32bit FPGA 임베디드시스템 환경에서 640 x 360 해상도의 샘플 영상을 적용하여 이진 및 세선화 연산에 대한 성능분석도 실시하였다. 검증결과 혼성모드 설계가 이전의 소프트웨어로만 이루어지는 처리속도에서 이진 및 세선화 단계에서 93.8% 향상될 수 있음을 확인하였다. 제안된 객체인식을 위한 혼성모드 시스템은 인공지능 네트워크가 적용되지 않는 엣지 컴퓨팅 환경에서도 환자의 움직임을 효율적으로 감지할 수 있을 것으로 기대된다.

☞ 주제어 : 혼성모드설계, 움직임 감지, 실시간 추적, RTL 설계, 논리회로 합성

ABSTRACT

In this paper, we propose an efficient image processing system to detect and track the movement of specific objects such as patients. The proposed system extracts the outline area of an object from a binarized difference image by applying a thinning algorithm that enables more precise detection compared to previous algorithms and is advantageous for mixed-mode design. The binarization and thinning steps, which require a lot of computation, are designed based on RTL (Register Transfer Level) and replaced with optimized hardware blocks through logic circuit synthesis. The designed binarization and thinning block was synthesized into a logic circuit using the standard 180nm CMOS library and its operation was verified through simulation. To compare software-based performance, performance analysis of binary and thinning operations was also performed by applying sample images with 640x 360 resolution in a 32-bit FPGA embedded system environment. As a result of verification, it was confirmed that the mixed-mode design can improve the processing speed by 93.8% in the binary and thinning stages compared to the previous software-only processing speed. The proposed mixed-mode system for object recognition is expected to be able to efficiently monitor patient movements even in an edge computing environment where artificial intelligence networks are not applied.

☞ keyword : Mixed mode design, Motion detection, Real-time tracking, RTL(Register Transfer Level), Logic synthesis

1. 서 론

실시간 영상에서 배경 영상과 입력 영상을 구분하여 움직이는 객체를 검출하는 고전적인 방법으로는 차영상을 이용한 방법, 블록 정합기법, 배경 영상을 이용하는 방법 등이 있다[1-4]. 최근 인공지능 기술의 도입으로 움

¹ Division of Software Convergence, Hanshin University, Osan city, Korea (18101)
² MetaReal Co., Ltd. 137 Hanshindaegil, Osan-si, Gyeonggi-do, Korea (18101)
³ SIMSREALITY Co.,Ltd. 4th floor, 74 Technojungang-ro, Yuseong-gu, Daejeon(34018)

*Corresponding author: jasmin@hs.ac.kr

[Received 01 December 2023, Reviewed 17 January 2024, Accepted 22 January 2024]

[☆] This study was supported by research funding from the Purchase Conditional Project [Project Number S3302465] from the Technology Information Promotion Agency for Small and

Medium Enterprises (TIPA), Ministry of SMEs and Startups. And also, this work was supported by Hanshin University Research Grant.

직접 검출은 객체인식 알고리즘인 CNN과 YOLOv3 와 같은 인공지능 기반의 객체 탐지를 위한 고도의 지능형 네트워크 적용으로 구현된다[5-12]. 그러나 이 방법은 IoT 및 엣지컴퓨팅 기반시스템의 환자움직임 감지를 위한 환경에서와 같이 영상변화가 심하지 않은 경우, 객체인식을 위한 최신의 인공지능 네트워크는 물리적 프로세서와 메모리와 같은 시스템 자원 사용과 처리시간의 측면에서 비효율적일 수 있다.

배경 영상을 이용한 객체의 움직임검출 단계는 그림 1 과 같다. 현재 프레임과 기준이 되는 배경 영상의 차이를 구하는 방법으로서, 차영상 방법과 같이 인접한 두 프레임을 비교하는 것이 아니라 이전 프레임들로부터 배경이 되는 영상을 추출하고 이 영상과 현재 프레임을 비교하여 움직임 객체를 검출하는 방법이다. 이미지처리 각 단계에서 이진화 영상으로부터 객체의 윤곽선을 추출하는 것은 객체영역의 중심좌표를 구하는 마지막 단계로서 가장 중요한 단계라고 할 수 있다. 윤곽선을 검출하는 일반적인 연산은 Sobel 필터 혹은 유사한 계열의 기법을 적용하는 것이다. 반면 세션화 알고리즘은 윤곽선 검출 이미지와 달리 오브젝트의 윤곽을 하나의 픽셀 라인으로 줄이는 영상골격화를 의미한다. 환자움직임과 같은 정밀한 움직임을 감지하기 위해서는 기존 윤곽선추출방식 보다는 세션화 알고리즘이 미세한 움직임의 객체이동 추적에 적합하다. 세션화 알고리즘은 영상프레임의 정보를 최소화하여 줄일 수 있다. 또한 병렬처리가 가능하므로 처리 속도 또한 빠른 편이다. 무엇보다 이진화와 세션화 알고리즘의 연산은 RTL 설계에 의한 합성된 하드웨어 논리 회로로 구현이 용이하여 별도의 프로세서로 설계하여 수행할 경우, 작은 규모의 논리회로를 이용해도 일반적인 고성능 CPU와 운영체제를 탑재한 소프트웨어적인 처리에 비하여 처리속도가 매우 빠르고 전력소모도 작아 효율적이다[13].

본 논문에서는 영상에서 환자와 같은 특수한 환경에 존재하는 객체의 움직임을 감지하고 추적하기 위한 효율적인 하드웨어 및 소프트웨어 혼성모드 시스템을 제안한다. 움직이는 객체영역 감지는 기존의 방식대비 정밀제어가 가능하고 하드웨어 구현에 더 효율적인 세션화 알고리즘을 이용하여 윤곽선을 추출하고 영역의 중심 좌표를 매 프레임마다 모니터링 하여 추적한다. 이진화와 세션화 단계는 RTL(Register Transfer Level) 기반으로 설계하여 논리회로 합성을 거쳐 하드웨어 블록으로 대체된다. 640 x 360 nHD (ninth High Definition)급 해상도의 샘플 영상을 적용하여 기존방식 대비 하드웨어 논리회로로

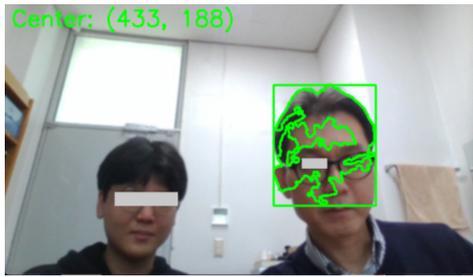
구현된 윤곽선 추출 단계의 세션화 연산에 대한 성능분석을 확인한다. 기존 소프트웨어 기반 알고리즘의 성능 분석에는 32 bit RISC MCU 를 합성하고 Embedded FPGA 시스템에 이용하여 CPU 소요 사이클 수를 분석한다.



(그림 1) 소프트웨어기반 객체 움직임 검출 알고리즘 (Figure 1) Software-based object motion detection algorithm

2. 소프트웨어기반 객체 움직임 검출 알고리즘의 구현

동영상처리에서 움직이는 객체를 탐지하는 기술로서 가장 많이 시도되는 방식이 배경분리 BGS(Background Subtraction) 영상에 의한 객체 영역 추출 방식이다[14-15]. BGS 방식 중에서 배경모델이 BGS의 핵심기술로서 배경 모델을 어떻게 구성하고 갱신해 가는가 하는 것이 중요한 관건이 되며, 이에 따라 여러 가지 방법들이 제안되고 있다. 그림 2는 본 연구에서 구현한 BGS 방식의 알고리즘 처리 결과이다. 그림 1 에서 언급된 바와 같이 기존의 제안된 여러 알고리즘들을 분석하여 최적화를 거쳐 파이썬 3.11.4 버전 환경에서 구현되었다. 카메라로 입력된 640 x 360 해상도의 동영상은 프레임에 따라 초당 30 장의 이미지로 캡처되며 분석을 통하여 배경영상을 만든다. 배경영상은 픽셀의 전체평균 값의 변화 또는 주변 이미지의 밝기변화, 그리고 개별 픽셀 내 미세한 변화 등의



(a) 원 이미지
(a) Original image



(c) 가우시안필터 및 이진화
(c) Gaussian filter and binarization



(b) 배경영상과 움직임영상과의 차영상
(b) Difference image between background image and motion image



(d) 윤곽선추출 및 객체영역검출
(d) Contour extraction and object area detection

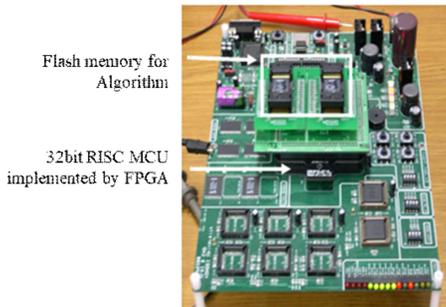
(그림 2) 소프트웨어기반 객체 움직임 검출 알고리즘 처리 결과 (640 x 360 (ninth HD))
(Figure 2) Software-based object motion detection algorithm processing results (640 x 360 nHD)

변화를 감지하고 여러 가지 변수를 반영하여 비 주기적으로 업데이트를 결정한다. 객체의 움직임 감지는 등록된 배경영상과 현재 영상의 차영상을 추출하여 이루어진다. 그림 2(b)는 사람의 움직임이 없는 배경 영상에 대하여 시간이 지남에 따라 움직이고 있는 사람이 있는 경우의 프레임들을 보여주며 이는 차영상에 해당된다. 영상에는 두 사람의 얼굴이 존재하나 오른쪽 객체의 움직임만 있는 것으로 실험하였으며 이는 차영상에 해당된다. 그림의 좌측에 보여지듯이 좌측 객체의 미세한 움직임에도 움직이는 객체로 간주될 수 있다. 특히 이러한 특성은 랜덤하게 발생하여 가우시안 잡음 형태로 나타나게 된다. 잡음으로 간주되는 부분을 없애기 위해 형태학적 필터링 기법인 침식을 적용한다. 가우시안 필터와 함께 적응형으로 적용하였다. 노이즈가 제거된 후 그림 2(c)와 같이 이미지 이진화가 진행된다. 이진화된 이미지는 그림 2(d)와 같이 윤곽선 추출 알고리즘을 거쳐 움직임이 발생한 객체영역을 사각형 블록으로 표시한다. 윤곽선 추출 알고리즘에는 일반적으로 Sobel 연산자 혹은 유사한 형태의 다른 알고리즘이 적용되나 본 논문에서는 환자와 같은 특수한 환경에 존재하는 객체의 움직임을 감지하기

위하여 기존의 방식대비 정밀제어가 가능하고 하드웨어 구현에 더 효율적인 세션화 알고리즘을 이용하여 윤곽선을 추출하였다. 움직이는 객체의 중심은 추출된 객체영역의 중심좌표로 결정되며 매 프레임마다 변화를 추적하여 다양한 방식으로 객체의 상태를 판단하는데 응용될 수 있다. 그림 2(d)는 설정된 관심 영역 내에서 이진화되어 있는 값들을 이용하여 수평, 수직 투영 기법으로 움직임 객체 영역을 재검출하게 된다. 이를 움직임 객체가 발생한 원 영상에 객체 표시를 하게 되면 그림 2(a)와 같이 같이 최종적으로 움직임 객체가 검출되게 된다.

구현된 소프트웨어 기반 알고리즘의 성능 분석에는 그림 3과 같이 Embedded FPGA 시스템을 이용하여 32 bit RISC MCU를 합성하고 알고리즘의 단계별 CPU 소요 사이클 수는 ARM 에뮬레이터를 동시에 이용하여 분석하였다. 알고리즘 명령어의 모든 결과는 ARM 프로세서의 소프트웨어 에뮬레이터인 ARM 에뮬레이터의 결과로 분석될 수 있다. 일반적으로 RISC 기반 ARM 계열 프로세서를 탑재한 임베디드시스템의 CPU 명령어는 3 종류로 분류된다. 표 1은 ARM 에뮬레이터 시뮬레이션 결과를 보여준다. 여기서 S 사이클은 순차적 사이클이다. 이 단

계에서 마이크로프로세서는 이전 주소 뒤에 단어 또는 빈 단어가 있는 주소나 동일한 주소로 요청을 보낸다. N 사이클은 비순차적 사이클로서 마이크로프로세서는 이전 주기에 사용된 주소와는 별개의 주소로 요청을 보낸다. I 사이클은 내부 사이클로서 마이크로프로세서는 내부 명령을 실행 중이므로 전송을 요청하지 않는다. 640 x 360 해상도의 현재 프레임에 대한 분석결과에 따르면 영상의 이진화와 세션화에 소요된 CPU 사이클 수는 총 42,584,665 사이클 이었으며 혼성모드설계 결과와 그 성능이 비교된다.



(그림 3) 소프트웨어기반 알고리즘의 CPU 점유를 분석하기 위한 FPGA 보드 알고리즘 처리 결과 (640 x 360 (ninth HD))

(Figure 3) FPGA board for analyzing CPU occupancy rate of software-based algorithms

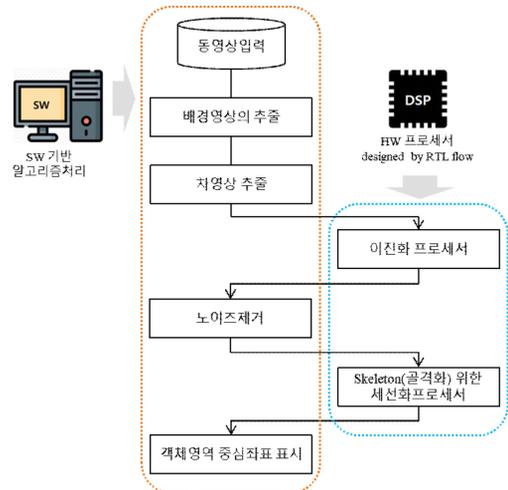
3. 혼성모드기반 객체 움직임 검출 알고리즘의 구현

3.1 제안된 알고리즘

본 논문에서는 그림 4 에서와 같이 환자와 같은 특정 객체의 움직임을 감지하고 추적하기 위한 효율적인 소프트웨어 및 하드웨어 블록이 혼재된 혼성모드 영상처리 시스템을 제안한다. 연산량이 많은 이진화와 세션화 단계를 RTL(Register Transfer Level) 기반으로 설계하여 논리회로 합성을 거쳐 최적화된 하드웨어 블록으로 대체한다. 설계된 이진화 및 세션화 블록은 표준 180nm CMOS 라이브러리를 이용하여 논리회로로 합성한 후 시뮬레이션을 통하여 동작을 검증한다. 소프트웨어기반의 성능비교를 위해 32bit FPGA 임베디드시스템 환경에서 640 x 360 해상도의 샘플 영상을 적용하여 이진 및 세션화 연산에 대한 결과를 바탕으로 비교분석을 실시한다.

3.2 세션화 알고리즘의 RTL 구현

이미지 세션화는 두꺼운 영상의 선을 가늘게 하여 한 줄의 Pixel(픽셀) 값으로 표시한다.

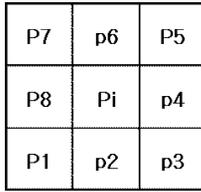


(그림 4) 제안된 혼성모드 객체 움직임 검출 시스템 (Figure 4) Proposed mixed-mode object motion detection system

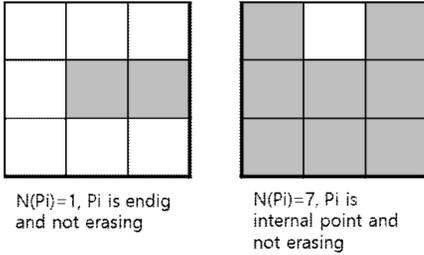
그림 5(a)에서 P_i 와 $P1\sim 8$ 은 이진 픽셀 이미지 값이다. 세션화 알고리즘에서 중심 이미지의 삭제에 위한 조건은 다음과 같다. 본 연구에서 이미지 값 1은 검정색을 의미하고 0은 흰색을 의미한다. 세션화는 중앙 이미지 P_i 가 1의 값 즉, 검은색일 때만 수행된다. 첫 번째 단계로서 다음 1~4 의 조건을 만족하는 픽셀이 삭제된다. 조건 1 에서, 중심 픽셀을 기준으로 주위 픽셀의 검은 픽셀의 개수가 2~6개인 픽셀을 검사한다. $N(P_i)$ 는 그림 5(b)의 예와 같이 8개 이웃 픽셀 중 값 1의 개수이며 다음의 수학적 1로 표현된다.

$$N(P_i) = P1 + P2 + \dots + P8 \quad (1)$$

조건 2번에서 주위 8개의 픽셀의 connectivity가 1인 픽셀을 검사한다. connectivity란 주위 픽셀에 연결되어 있는지를 알 수 있는 숫자로 주변 픽셀이 검정->흰색 아니면 흰색->검정색으로 변하는 개수를 말한다. 그림 6에서 볼 수 있듯이 $S(P_i)$ 는 8개의 이웃 픽셀에 1~0(1->0) 패턴의 연결(connectivity) 개수를 의미한다. 3번은 $p2, p4, p6$ 중의 적어도 하나는 흰색 픽셀일 조건이고 4번은 $p4, p6$,

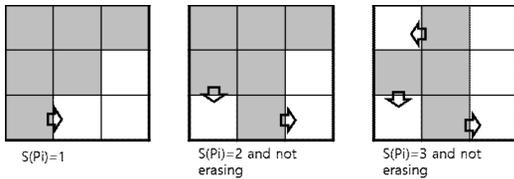


(a) 3x3 세선화 윈도우의 픽셀배치
(a) Pixel placement in a 3x3 window



(b) P(i) 1의 개수의 예
(b) Example of number of P(i) 1
(그림 5) 픽셀 인덱스와 P(i)의 계산예

(Figure 5) Calculation example of pixel index and P(i)



(그림 6) P(i)의 connectivity 계산예
(Figure 6) Example of connectivity calculation of P(i)

p8 중의 적어도 하나는 흰색 픽셀일 경우이다. 이 네 가지 조건에 모두 만족하는 픽셀을 모두 삭제한다. 조건 중에서 하나라도 만족하지 못한 경우가 있으면 삭제되지 않는다.

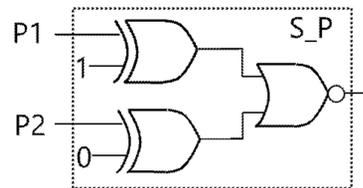
1. $2 \leq N(P_i) \leq 6$
2. $S(P_i) = 1$
3. $P_2 * P_6 * P_8 = 0$
4. $P_4 * P_6 * P_8 = 0$

두 번째 단계에서는 첫 번째 단계의 결과 이미지를 그대로 적용하여 다음 조건에서 다시 실행된다. 1 단계에서

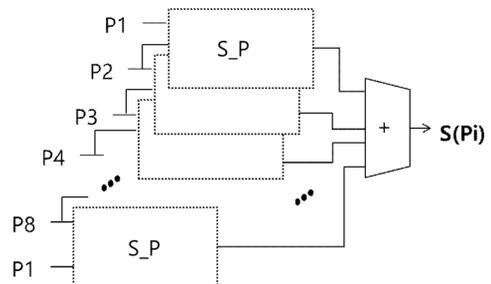
나온 이미지 전체에 대하여 조건 1 과 2 는 동일하며 조건 3 과 4 만 다음과 같이 변경된다. 역시 조건 중에서 하나라도 만족하지 못한 경우가 있으면 삭제되지 않는다.

3. $P_2 * P_4 * P_8 = 0$
4. $P_2 * P_4 * P_6 = 0$

그림 7은 S(Pi)를 계산하기 위한 논리회로이다. 두 개의 연속적인 P(n), P(n+1) 및 (1→0)의 일치 여부를 확인하기 위해 회로는 EXOR 게이트를 사용하여 연산을 수행한 다음 NOR을 수행한다. 모든 값을 합산한다. 이 연산의 경우 하나의 클럭으로도 가능하기 때문에 이전 SW 연산보다 빠른 결과를 얻을 수 있다. 세선화를 위한 논리 블록 다이어그램과 타이밍은 그림 8에 나타나 있다. 3x3 픽셀 윈도우는 P(i,j)의 8개 이웃 픽셀을 생성한다. 세선화 알고리즘에서는 중앙 이미지 P(i,j) 인접 3x3 행렬의 8개 픽셀에 대해 2단계 처리 프로세스가 수행된다. 설계된 회로는 블록 중심 이미지 P(i, j)의 3개의 수직 이미지와 일치하는 메모리 주소를 생성한다. 즉, 3클럭마다 3픽셀의 이미지가 3x3 2D 시프트 레지스터로 전송되면서 동시에 윈도우의 왼쪽으로 시프트 된다. 제안된 회로는 소프트웨어

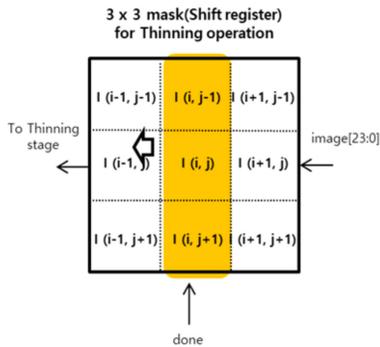


(a) S_P circuit

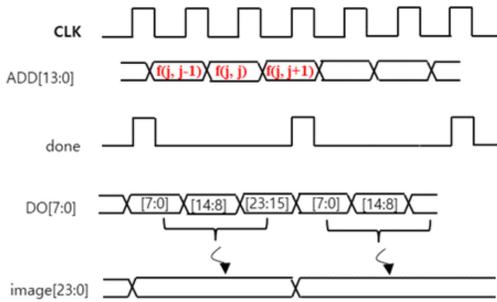


(b) S_P를 이용한 S(Pi) 추출 회로
(b) S(Pi) extraction circuit using S_P

(그림 7) S(Pi) 계산을 위한 제안된 논리회로
(Figure 7) Proposed logic circuit for calculating S(Pi)

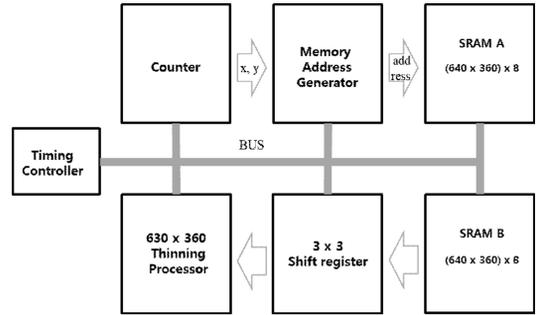


(a) 3x3 쉬프트 레지스터의 동작



(b) 쉬프트 레지스터 제어신호

(그림 8) 세선화를 위한 3x3 쉬프트 레지스터 동작
(Figure 8) 3x3 shift register operation for thinning

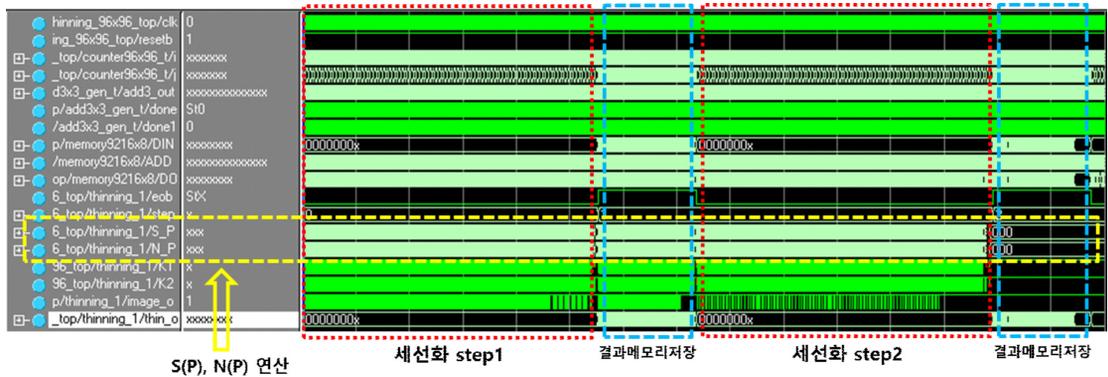


(그림 9) 세선화 처리기의 RTL 블록도

(Figure 9) RTL block diagram of thinning processor

전용 방식으로 이루어지는 간단한 루틴의 반복으로 인한 불필요한 처리 시간을 줄이고 빠른 시간에 세선화 단계를 처리한다. 간단한 회로로 3클럭에서 $P(i,j)$ 의 세선화 결과를 얻을 수 있다. 3개의 클럭으로 메모리에서 읽은 24비트 이미지는 'done' 신호에 의해 쉬프트 레지스터에 저장되고 3x3 값과 세선화 작업이 수행된다.

그림 9는 제안된 세선화 프로세서의 전체 논리 블록 다이어그램을 나타낸다. 메모리를 위한 주소 생성기, 카운터 및 메모리모듈로 구성되어있다. 하드웨어의 각 블록은 RTL로 설계되고 시뮬레이션 되었다. 로직은 또한 180nm 표준 CMOS 프로세스를 사용하여 Synopsys Design Compiler에 의해 전체 논리회로가 합성되었다. 그림 10은 합성된 논리 회로 시뮬레이션을 보여준다. 그림 11(a)는 이진화된 640 x 360 해상도의 프레임 입력이미지를 나타내며 그림 11(b)는 세선화 프로세서의 동작결과를 보여준다. 이진화 및 세선화 동작에는 50MHz 클럭주파수에



(그림 10) 논리회로 시뮬레이션 결과
(Figure 10) Logic simulation results

(표 1) 이진화 및 세선화 CPU 점유율 비교 (640x320 이미지 샘플)

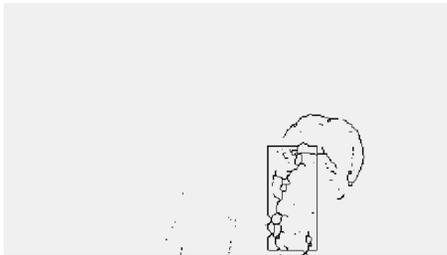
(Table 1) Comparison of binarization and thinning CPU occupancy (640x320 image samples)

Process step 640 x 320	Binarization	Thinning	S/W total	Proposed H/W total (*Improvement %)
S-cycles	1,268,166	21,850,651	23,118,818	
N-cycles	375,695	12,824,269	13,199,963	2,541,885
I-cycles	136,226	5,909,745	6,045,971	(94%)
Total	1,780,087	40,584,665	42,364,753	

* Improvement = (S/W total - H/W total) / (S/W total) %



(a) 이진화 입력영상



(b) 노이즈필터 이전 세선화 처리기 결과영상

(그림 11) 세선화처리기 동작결과 (640 x 360)

(Figure 11) Thinning processor operation result (640 x 360 resolution sample image)

서 2,541,885 사이클이 소요되었다. 표 1은 ARM 애플레터에 의하여 분석된 소프트웨어기반 동작의 CPU 점유율과 RTL 방식의 하드웨어 기반에서 이진화 및 세선화 동작의 점유율을 나타낸다. 검증결과 RTL 방식의 하드웨어 기반에서 이진화 및 세선화 동작에서 94%의 CPU 점유율 향상도를 나타내었다.

4. 결 론

영상에서 환자와 같은 특수한 환경에 존재하는 객체

의 움직임을 감지하고 추적하기 위한 효율적인 하드웨어 및 소프트웨어 혼성모드 시스템을 제안한다. 환자움직임과 같은 정밀한 움직임을 감지하기 위해서는 기존 윤곽선추출방식 보다는 세선화 알고리즘이 미세한 움직임의 객체이동 추적에 적합하다. 세선화 알고리즘은 영상프레임의 정보를 최소화하여 줄일 수 있다. 또한 병렬처리가 가능하므로 처리속도 또한 빠르다. 이진화와 세선화 알고리즘의 연산은 RTL 설계에 의한 합성된 하드웨어 논리회로로 구현이 용이하여 별도의 프로세서로 설계하여 수행할 경우, 작은 규모의 논리회로를 이용해도 일반적인 고성능 CPU와 운영체제를 탑재한 소프트웨어적인 처리에 비하여 처리속도가 매우 빠르고 전력소모도 작아 효율적이다. 본 논문에서는 움직이는 객체영역 감지를 위해 기존의 방식대비 정밀제어가 가능하고 하드웨어 구현에 더 효율적인 세선화 알고리즘을 적용하여 윤곽선을 추출하고 영역의 중심 좌표를 매 프레임마다 모니터링하여 추적하였다. 이진화와 및 세선화 단계는 RTL 기반으로 설계하여 논리회로 합성을 거쳐 하드웨어 블록으로 대체하고자 한다. 640 x 360 nHD 급 해상도의 샘플 영상을 적용하여 기존방식 대비 하드웨어 논리회로로 구현된 윤곽선 추출 단계의 세선화 연산에 대한 성능분석을 통해서 그 성능을 비교분석 하였다. 기존 소프트웨어 기반 알고리즘의 성능 분석에는 32 bit RISC MCU 를 합성하고 Embedded FPGA 시스템을 이용하여 CPU 소요 사이클 수를 분석하였다. 설계된 이진화 및 세선화 블록은 표준 180nm CMOS 라이브러리를 이용하여 논리회로로 합성된 후 시뮬레이션을 통하여 동작을 검증하였다. RTL 방식의 하드웨어 기반의 혼성모드 설계가 윤곽선 추출단계에서 이전의 소프트웨어로만 이루어지는 처리속도에서 93.8% 향상될 수 있음을 확인하였다. 제안된 객체인식을 위한 혼성모드 시스템은 인공지능 네트워크가 적용되지 않는

엣지 컴퓨팅 환경에서도 환자의 움직임을 효율적으로 감지할 수 있을 것으로 기대된다.

References

- [1] Yegi Lee, Shin Kim, Hanshin Lim, Hee Kyung Lee, Hyon-Gon Choo, Jeongil Seo, and Kyoungro Yoon, "Performance Analysis of Object Detection Neural Network According to Compression Ratio of RGB and IR Images," *JBE*, Vol. 26, No. 2, March 2021. <https://doi.org/10.5909/JBE.2021.26.2.155>
- [2] Seulkee Jung, Jong-Bae Lee, Seongsoo Lee, "Implementation of Real-Time Multi-Camera Video Surveillance System with Automatic Resolution Control Using Motion Detection," *Journal of IKEEE*, Vol. 18, No. 4, pp. 612-619, Dec. 2014. <https://doi.org/10.7471/ikeee.2014.18.4.612>
- [3] Lin, Tsung-Yi, et al., "Focal loss for dense object detection," *Proceedings of the IEEE international conference on computer vision*, pp. 2980-2988, 2017. <https://doi.org/10.48550/arXiv.1708.02002>
- [4] Girshick, Ross, et al., "Rich feature hierarchies for accurate object detection and semantic segmentation," *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580-587, 2014. <https://doi.org/10.48550/arXiv.1311.2524>
- [5] Girshick Ross, "Fast rcnn," *Proceedings of the IEEE international conference on computer vision*, pp. 1440-1448, 2015. <https://doi.org/10.1109/ICCV.2015.169>
- [6] Ren, Shaoqing, et al., "Faster rcnn: Towards real-time object detection with region proposal networks," *IEEE transactions on pattern analysis and machine intelligence*, Vol. 39, No. 6, pp. 1137-1149, 2016. <https://doi.org/10.48550/arXiv.1506.01497>
- [7] Dong-Hoon Shin, Ji-Won Baek, Roy C. Park, Kyungyong Chung, "Deep Learning-based Vehicle Anomaly Detection using Road CCTV Data," *Journal of the Korea Convergence Society*, Vol. 12. No. 2, pp. 1-6, 2021. <https://www.dbpia.co.kr/journal/voisDetail?voisId=VOIS00645034>
- [8] Redmon, Joseph, et al., "You only look once: Unified, real-time object detection," *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779-788, 2016. <https://doi.org/10.48550/arXiv.1506.02640>
- [9] Farahnakian, Fahimeh, et al., "Deep Convolutional Neural Network based Fusion of RGB and IR Images in Marine Environment," *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pp. 21-26, 2019. <https://doi.org/10.1109/ITSC.2019.8917332>
- [10] Liu, Shuo, and Zheng Liu, "Multi-channel CNN-based object detection for enhanced situation awareness," *arXiv preprint arXiv :1712. 00075*, 2017. <https://doi.org/10.48550/arXiv.1712.00075>
- [11] Ha-Rin Lee, Yoo-Jin Kim, Min-Ah Lee, Jae-Hyun Moon, "Motion Recognition of SmartPhone using Sensor Data," *Proceedings of the Korean Information Processing Society 2020 Online Fall Conference*, Vol.27, No. 2, Dec. 2020.
- [12] Y. C. Lee, and C. W. Lee, "Danger Alert Surveillance Camera Service using AI Image Recognition technology," *Journal of Korea Multimedia Society*, vol. 17, no. 12, pp. 1437-1445, 2014. <https://doi.org/10.3745/PKIPS.y2020m11a.814>
- [13] Seungmin Jung, "VLSI Implementation of Algorithm for Performance Improvement of Minutiae Based Fingerprint Recognition System," *Journal of Computing Science and Engineering* Vol. 14, No. 4, pp. 146-153, Dec. 2020. <https://doi.org/10.5626/JCSE.2020.14.4.146>
- [14] Kim dong-woo, Song young-jun, Kim ae-kyeong, Ahn jae-hyeong, "The moving object detection for moving picture with gaussian noise," *Korea Contents Association 2009 Spring Conference*, 2009. <https://doi.org/10.1109/MECO.2019.8760066>
- [15] Jong-beom Park, "A Study on the Background Image Updating Algorithm for Detecting Fast Moving Objects," *J. Korea Inst. Intell. Transp. Syst.* Vol. 5, No. 41, pp. 53~160, 2016. <https://doi.org/10.12815/kits.2016.15.4.153>

◎ 저 자 소 개 ◎



정 승 민 (Seungmin Jung)

1990년 연세대학교 전자공학과 (공학사)
1992년 연세대학교 대학원 전자공학과 (공학석사)
2006년 연세대학교 대학원 전기전자공학과 (공학박사)
2024년 현재 한신대학교 소프트웨어융합학부 교수
관심분야 : SoC(System on Chip), 생체인식 및 이미지처리, VLSI 설계
E-mail : jasmin@hs.ac.kr



정 의 성 (Eu-Sung Jung)

1990년 연세대학교 전자공학과(공학사)
1992년 연세대학교 대학원 전자공학과(공학석사)
1992년~1998년 효성T&C 선임연구원
1998년~2000년 DSPD&C 선임연구원
2000년~2004년 텔슨정보통신 책임연구원
2004년~2005년 레인콤 책임연구원
2006년~2007년 맥시안 책임연구원
2007년~2010년 엠에스웨이 기술이사
2010년~2017년 셀앤틱 연구소장
2016년~2021년 제스엔지니어링 연구소장
2022년~현재 메타리얼 주식회사 대표
관심분야 : 메타버스, UWB 측위기술, 정보통신, 사물인터넷, etc.
E-mail : jes559@daum.net



김 명 환(Kim Myung Hwan)

2003년 경상대학교 컴퓨터과학과(이학사)
2007년 매크로미디어 연구소장
2016년 솔트웍스 사업기획팀 팀장
2018년~현재 (주)심스리얼리티 대표이사
관심분야 : 메타버스, XR, 소프트웨어 아키텍처, 분산객체 etc
E-Mail : oops@simsreality.com