임베디드 제어에 의한 무인 영상 감시시스템 구현

Implementation of An Unmanned Visual Surveillance System with Embedded Control

김동진*, 정용배*, 박영석**, 김태효**

Dong-jin Kim*, Yong-bae Jung*, Young-seak Park**, Tae-hyo Kim**

요 약

본 논문에서는 SOPC 기반 NIOS II임베디드 프로세서와 C2H 컴파일러를 적용하여 영상 감시 시스템을 구현하였다. 카메라의 영상신호 출력, 영상처리, 시리얼 통신 및 네트워크 통신의 제어를 위해 C2H 컴파일러에 의한 IP를 구성하였고, SOPC 및 NIOS II 임베디드 프로세서에 기반한 각각의 IP를 효과적으로 제어할 수 있도록 구현하였다.

그리고, 보다 빠르고 환경에 강인한 이동 물체 검출을 위한 방법으로 배경영상을 갱신하는 알고리듬을 적응 가우시안 혼합 모델(AGMM)을 제안하였다. 그 결과 주간 및 야간에서도 이동 물체를 잘 검출할 수 있었다. 실험을 통해 제안된 AGMM 알고리듬이 적응 임계치법(ATM)과 가우시안 혼합모델(GMM)보다 이동하는 보행자 및 차량의 검출에서 우수함을 확인하였다.

Abstract

In this paper, a visual surveillance system using SOPC based NIOS II embedded processor and C2H compiler was implemented. In this system, the IP is constructed by C2H compiler for the output of the camera images, image processing, serial communication and network communication, then, it is implemented to effectively control each IP based on the SOPC and the NIOS II embedded processor.

And, an algorithm which updates the background images for high speed and robust detection of the moving objects is proposed using the Adaptive Gaussian Mixture Model(AGMM). In results, it can detecte the moving objects(pedestrians and vehicles) under day-time and night-time. It is confirmed that the proposed AGMM algorithm has better performance than the Adaptive Threshold Method(ATM) and the Gaussian Mixture Model(GMM) from our experiments.

Keywords : NIOS II embedded processor, C2H compiler, visual surveillance, background image update, Gaussian mixture model

Ⅰ. 서 론

현대사회에서 전 세계적으로 테러 및 범죄 사고가 증가함에 따라 무인 영상 감시기술의 요구가 더욱 확대되고 있다. 많은 공공장소에서 안전과 보안 시설이 확대 설치되고 있고, 사고 발생 시에 자동으로 운영자에게 그 상황을 전송하여 원격으로 모니터링할 수 있도록 하고 있다. 현재의 영상기반 감시시스템은 효과적인 전송, 컬러영상 해석, 사건

기반 및 모델기반 인식을 위한 실시간 영상해석을 적용하고 있다[1]. 효과적이고 다양한 센서들을 융합하여 저가격 고속 계산이 가능하며, 멀티미디어시스템에 결합하는 형태로 발전되었다[2].

첨단 영상감시시스템이 성공적으로 작동하는 상품을 개발하기 위해 많은 핵심적인 문제점을 극복해야 한다. 다양한 외부 환경의 변화, 즉 변하는 장면, 조명광 및 일기 조건 등에서도 대상 물체를 분류 또는 인식률이 높은 알고리듬이 요구된다. 이들 알고리듬은 패턴인식 및 통계적 방법을 적용하여 구현하고 있다[2-3].

한편, 현재까지의 영상 감시용 CCTV 시스템은 PC기반으로 구현되었으며, 그 가격이 비싼 편이었다. 이러한 기존 CCTV 시스템의 단점을 보완하기 위해 더 많은 인력을 충원하거나 감시 카메라의 수를 늘리기 보다는 지능적으로 판단하고 추적할 수 있는 임베디드 시스템을 기반으로 하는 영상 감시시스템의 필요성이 더욱 커지고 있다[4-5].

따라서 본 논문에서는 Nios II 임베디드 프로세서 시스템의

* 경남대학교

투고 일자 : 2010. 12. 20 수정완료일자 : 2011. 1. 17
게재확정일자 : 2011. 2. 2
* 본 논문은 2009년도 경남대학교 교내학술연구비 지원에 의하여 연구된 결과임을 밝힙니다.

Linux 디바이스 드라이버 및 영상감시용 하드웨어 플랫폼을 구성하여 pc기반 영상감시시스템의 단점을 보완하고, SOPC형 NIOS II 임베디드 프로세서와 영상처리 알고리듬 구현에서 소프트웨어 중심 프로그래밍 기법의 주요 문제점인 고속처리를 위하여 소프트웨어 프로그래밍과 C-To-Hardware(C2H) 컴파일러를 사용하는 하드웨어 프로그래밍을 병합하여 시스템의 성능을 향상시키고자한다. NIOS II 임베디드 프로세서 플랫폼 설계 중심으로 각각의 디바이스 인터페이스를 통합 서버시스템을 구축하고 사용자의 접근 효율을 높이기 위한 네트워크상에서 제어하는 기능을 제공하는 시스템을 구현한다[6-9].

구현된 임베디드 시스템을 이용하여 본 논문에서는 영상감시 알고리듬을 제안한다. 옥외의 환경변화가 큰 점을 감안하여, 즉 주간, 야간, 주위의 불필요한 배경성분의 변화등에 강인한 적응 가우시안 혼합 모델(AGMM)을 제안하고, 보행자 및 이동차량을 대상으로 본 알고리듬이 다른 적응 임계치 기법(ATM)이나 가우시안 혼합 모델(GMM)과 비교하여 환경 변화에 강인함을 확인하고자 한다.

II. Nios II 임베디드 프로세서 코어 기 반 IP 설계

2.1 Nios II 임베디드 프로세서 플랫폼 구현

하드웨어 설계의 주요 도구로는 Quartus II 설계 소프트웨어와 SOPC Builder 시스템 개발도구, Modelsim 시뮬레이션 소프트웨어 그리고 구현회로 검증을 위해서는 SignalTap II 임베디드 논리 해석기를 이용하였다[5-6]. 그림 1은 본 연구가 목적하는 시스템의 하드웨어 구성이다.

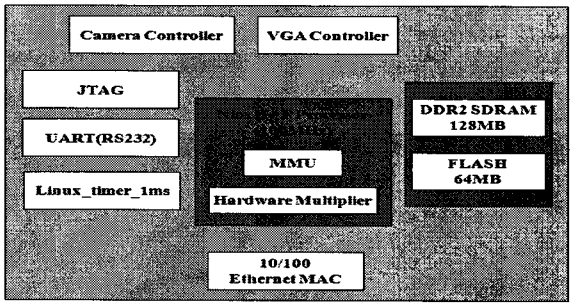


그림 1 시스템 하드웨어 블록도
Fig. 1. Block diagram of system hardware

그리고 Linux를 위한 최소 시스템으로 프로세서의 경우 Nios II f 코어와 MMU, Hardware multiplier가 요구된다. 특히 MMU 설정에서 512 또는 1KB의 On-Chip Memory가 필요하다. MMU를 포함한 Nios II CPU를 디자인하기 위해서 On-Chip Memory를 추가한다.

그림 2는 CPU와 On-Chip Memory 관계를 설정하는 모습을 보여주고 있다. 여기서 CPU의 jtag_debug_module, MMU를 위한 On-Chip Memory등의 어드레스를 설정한다[7-8].

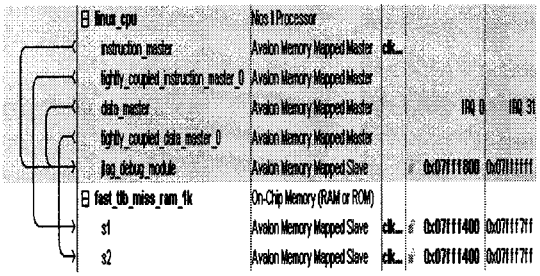


그림 2. Base Address 설정
Fig. 2. Base Address Configuration

또한 사용자 IP나 다른 주변장치를 추가할 경우, 0x0~0x1FFFFFFF의 주소를 가지는 주변장치는 주변장치의 레지스터와 메모리가 직접 맵핑되며, 0x20000000 이상의 주소를 가지는 주변장치의 경우는 MMU에 의해 주변장치의 레지스터와 메모리가 맵핑된다. SDRAM의 경우 최소 8MB의 용량이 요구되며 최대 128MB까지 가능하다. 사용자 콘솔 장치로 JTAG UART 또는 Serial UART가 사용되며, 한 개의 Full Featured Timer가 필요하다. 리눅스의 경우 IRQ0은 Auto-detected를 의미하므로 Timer를 제외한 나머지 디바이스는 절대 IRQ0을 사용할 수 없다.

Nios II 임베디드 프로세서의 플랫폼을 구성하기 위해서는 목적 시스템을 설정하고 이를 SOPC Builder 개발도구를 사용하여 생성한다. 이는 하드웨어 구조를 생성하는 Quartus II 설계도구 환경에서 구동된다[7]. 또한 영상 입출력 및 주변 장치를 포함한 Nios II 임베디드 프로세서 코어 기반 임베디드 프로세서의 하드웨어 구성은 그림 3과 같이 CPU(Nios II Processor), DDR2 SDRAM, FLASH Memory, Camera Controller(TVP5145),VGA Controller(CHRO

NTEL CH7010B), I2C Controller, Triple-Speed Ethernet, JTAG UART, Timer 등의 컴포넌트가 포함된다.

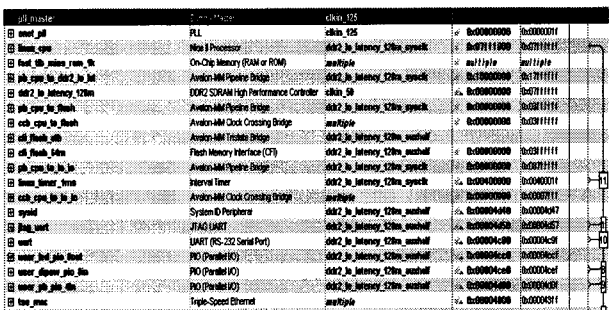


그림 3. 임베디드프로세서 하드웨어 플랫폼 구성
Fig. 3. The hardware platform configuration for an embedded processor

그림 4는 Quartus II 소프트웨어로 구성한 시스템 전체의 RTL 회로도이다.

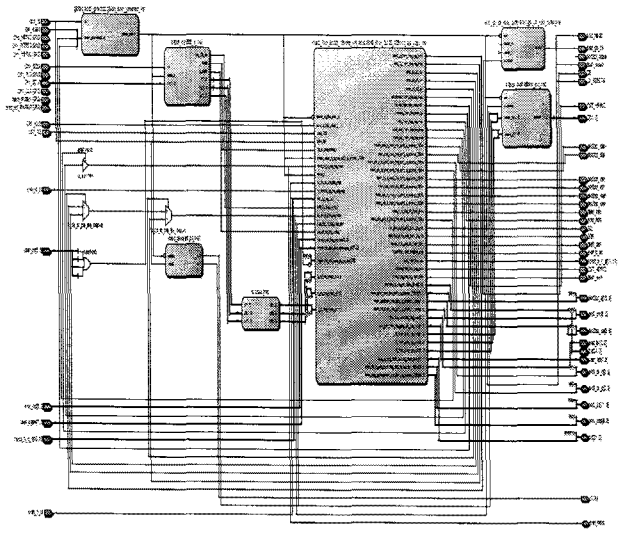


그림 4. 시스템 전체의 RTL 회로도
Fig. 4. RTL circuit diagram of the full System

그리고 Nios II 임베디드 프로세서 기반 리눅스 운영체제를 탑재하기 위해 임베디드 프로세서의 기본 정보가 포함된 헤드 파일을 커맨드 셸에 의해 생성하여 리눅스에서 컴파일 할 때 사용한다.

2.2 C2H(C-To-Hardware) 컴파일러

C2H는 소프트프로그램인 C기반 프로그램 구조를 IDE(NIOSII Integrated Development Environment)에서 연동하여 하드웨어 구조로 변경하는 컴파일러이다. 그림 5는 소프트웨어 빌드와 C2H간 통합 관계를 보이며, C2H를 이용하여 일부분을 컴파일 했을 경우 하드웨어와 소프트웨어 간 최적의 동작 흐름을 나타내고 있다[6-7].

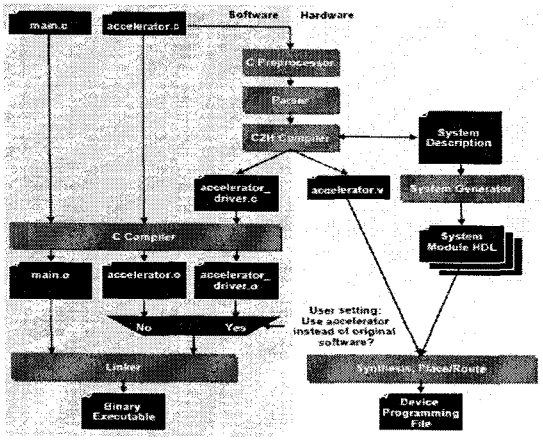


그림 5. 소프트웨어 빌드에 의한 C2H 통합 및 처리 흐름도.
Fig. 5. C2H integration and processing flow diagram by the software build

그림 6은 비전 기반 감시 알고리듬의 흐름에 대한 블록다이어그램을 보였다. 감시 알고리듬의 처리 과정은 입력받은 영상 맵에서 전처리 과정을 거쳐 객체의 후보와 위치를 인지하고 분류하는 형태로 구성되어 있다[8-10].

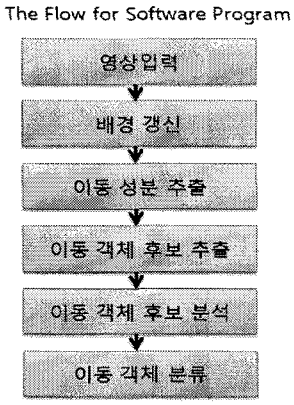


그림 6. 영상 처리 흐름도
Fig. 6. Block diagram of image processing

여기서 C2H를 사용할 경우, C2H를 적용할 부분을 분류하는 것이 가장 중요하다. 따라서 감시 알고리듬 구성은 C2H를 적용하는 부분은 반복학습이 많은 부분을 중심으로 하드웨어 구조를 적용하도록 그림 7과 같이 구성하였다.

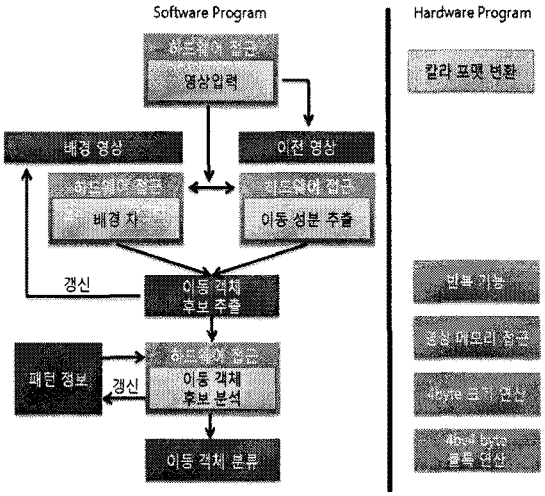


그림 7. C2H에 의한 영상 처리 흐름도
Fig. 7. Block diagram of image processing by C2H

그리고 그림 7은 C2H 컴파일러를 적용하는 영상처리 동작에 대한 처리과정을 보였는데, 이는 그림 8과 같이 IDE 개발 도구를 사용하여 C2H를 설정 및 컴파일한다. 여기서 C2H 컴파일러의 수행에 의해 생성된 하드웨어 구조인 Verilog 코드는 SOPC Builder에 추가 가능한 IP형태의 옵션을 생성된다.

그림 9는 SOPC Bulider 도구에서 C2H에 의해 생성된 감시 IP를 NIOS II 임베디드 프로세서 내부에 추가 및 재구성한 결과이며, Quartus II 도구에 컴파일 함으로서 최종적으로 감시 솔루션에 대한 플랫폼을 완성한다.

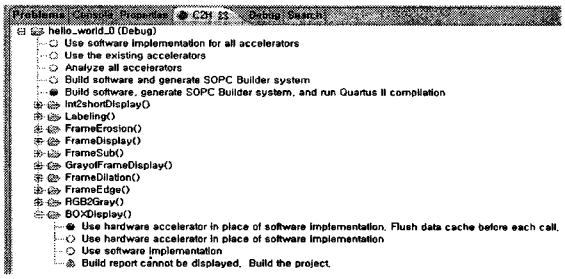


그림 8. IDE tool에 의한 C2H 설정
Fig. 8. The set-up of C2H by IDE tool

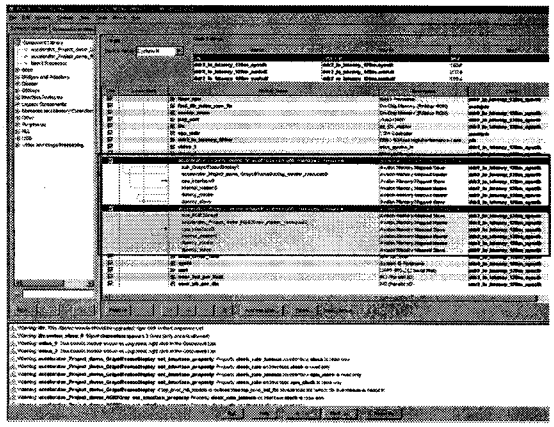


그림 9. SOPC Builder에서의 NOIS II 임베디드 프로세서 구조 재설정
Fig. 9. Re-establishment for the Nios II embedded processor by SOPC Builder Tool

III. Linux 운영체제 기반 감시 솔루션의 디바이스 드라이버 구현

3.1 영상 입출력 디바이스의 특성

영상을 획득하기 위해서 영상 입출력 확장 모듈을 사용하였으며 모듈의 동작 흐름은 그림 10과 같다.

영상 입출력 확장 모듈은 4채널 Composite Video Input을 지원하는 TI사의 TVP5154 Device를 사용하였다. 최대 해상도는 720×480이며, ITU-R BT.656 Standard Sampling을 지원한다. 그리고 I2C 통신으로 Camera의 Scaler 레지스터를 설정하여 320×240 해상도의 영상을 Camera Controller로 전달한다.

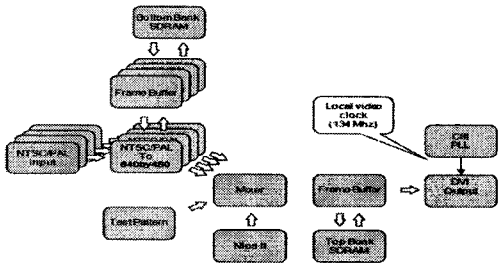


그림 10. 영상 획득 모듈의 내부 흐름도
Fig. 10. The internal flow of image acquisition module

또한 카메라로부터 받은 영상을 BT656 모듈에서는 YCbCr로 변환하고, CSC 모듈에서는 RGB로 변환하여 Nios II 프로세서로 전송한다[8-9]. 마지막으로 입력 받은 RGB 영상을 처리한 후 VGA 제어 장치를 이용해서 영상을 출력한다.

3.2 감시 솔루션의 디바이스 드라이버 설계

그림 11은 영상을 입력 받아 프로세싱 후 출력 과정까지의 흐름을 간략히 보여주고 있다. 여기서 Image Processing 블록의 동작을 C2H에 의해 재구성한 하드웨어 구조로 구성되어 있다.

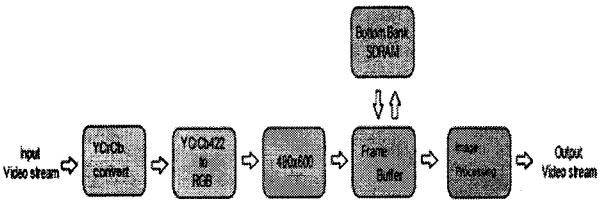


그림 11. 감시 솔루션의 영상처리 과정
Fig. 11. Image processing procedure for surveillance solution

그림 12는 그림 11에서 나타내고 있는 감시 솔루션의 흐름에 대하여 구현된 디바이스 드라이버의 동작 순서도이다. 위와 같이 II절 및 III절에서 구현된 임베디드 제어장치는 그림 13과 같다

IV. 영상감시 알고리듬 구현

4.1 배경영상 갱신 알고리듬

본 논문에서는 이동객체를 추출하기 위해서 II, III절에서 구현한 시스템 특성을 이용하여 관심 영역 범위 내에서 그림 14와 같이 이동 객체추출을 위해 적응적 가우시안 혼합 모델과 프레임 간의 이동 성분 분석의 두 가지 방법을 적용하였다.

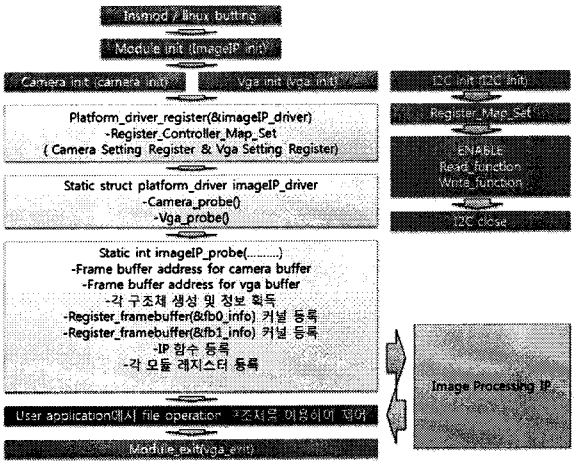


그림 12. 영상처리 디바이스 드라이버 동작
Fig. 12. The flow of Image Processing device driver

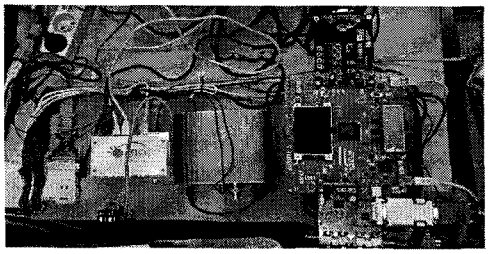


그림 13 구현된 임베디드 시스템
Fig. 13. Implemented embedded system

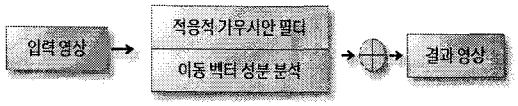


그림 14 제안된 이동 객체 추출 방법
Fig. 14. The proposed method for moving object extraction

여기서 제시한 다중 가우시안 혼합모델은 배경성분을 다중 가우시안 혼합 모델로 분석하여 새로 생성된 배경과 현재 영상의 차를 이용한 배경 차분 기법이다[10]. 그리고 이동성분 분석은 이전 영상과 현재 영상의 차에 의해 생성된 영상을 블록단위로 분석하여 이동성분의 밀도가 높으면 블록을 활성화하는 방법이다[11-12].

(1) 가우시안 혼합 모델

다중 가우시안 혼합 모델의 성분들은 가우시안 파라미터 집합을 이루는 각 성분의 평균과 분산 그리고 가중치를 통해 추정할 수 있다. 그러므로 개별 가우시안 확률 밀도 함수를 이루는 파라미터에 따르는 다음 식(1)과 같이 표현 할 수 있다.

$$p(x|\theta)=\sum_{i=1}^{N} p(x|\mu_i,\sigma_i^2)\alpha_i \tag{1}$$

여기서, μ_i와 σ_i^2는 배경의 i번째 가우시안 확률 밀도 함수에서 평균과 분산이며, α_i는 혼합 가중치(mixture weight)이다. 그리고 각 파라메타는 다음 식(2), 식(3)과 같이 갱신된다.

$$\mu_{i,t}=(1-\varphi)\mu_{i,t-1}+\varphi x_t \tag{2}$$

$$\sigma_{i,t}^2=(1-\varphi)\sigma_{i,t-1}^2+\varphi(x_t-\mu_{i,t})^T(x_t-\mu_{i,t}) \tag{3}$$

그리고 가중치는 다음 식 (4)와 같이 갱신한다.

$$\alpha_{i,t}=(1-\phi)\alpha_{i,t-1}+\phi \tag{4}$$

여기서, ϕ 그리고 φ는 갱신 율을 나타내며, t는 연속적으로 입력하는 시각을 의미한다. 만약 관찰된 화소가 i번째 가우시안 분포의 현재 화소 값을 따르지 않으면,

$\mu_{i,t}$ 그리고 $\sigma_{i,t}$는 동일하게 남기고 가중치만을 다음 식(5)와 같이 갱신 한다.

$$\alpha_{j,t}=(1-\phi)\alpha_{j,t-1} \tag{5}$$

(2) 이동 성분 추출 과정

이동 성분 추출과정은 배경모델과 입력되는 영상으로부터 관심영역을 설정한다. 관심영역은 다시 관심영역 내에서 블록단위로 나눈다. 여기서 관심영역 내에서 블록의 크기는 기본 8*8과 상태 및 상황 조건에 의한 4*4이다. 기본 블록은 입력되는 영상에서 잡음 성분이 많이 검출 될 때와 관찰 영역이 근거리일 경우 적용되며, 상태에 의한 블록은 입력되는 영상에서 잡음 성분이 적을 경우 선택되어진다.

관심 블록의 활성화 상태를 확인하기 위해 식(6)과 같이 모델링한다.

$$ROI=\begin{cases} Enable & \text{if } B_i > TH \\ Disable & \text{if } B_i \le TH \end{cases} \tag{6}$$

여기서 ROI는 관심 블록내의 픽셀단위에서 임계값 TH에 의존한 픽셀 수량의 상태를 의미한다. 그리고 B_i는 식(7)에 의해 표현되어진다.

$$B_i=\begin{cases} B_i+1 & \text{if } ID_t(x,y)>0 \\ B_i & \text{if } ID_t(x,y)=0 \end{cases} \tag{7}$$

영상에서 $D_t(x,y)$의 값을 이진화하기 위해 식(8)에 의해 결정된다.

$$ID_t(x,y)=D_t(x,y)>T_t(x,y) \tag{8}$$

여기서, $ID_t(x,y)$는 임계값 $T_t(x,y)$에 의해 결정되어진 이진화 처리된 영상의 값이다. 그리고 $T_t(x,y)$는 영상의 이진화를 위한 픽셀 단위의 적응 임계값이다. 따라서 임계값 $T_t(x,y)$보다 크면 이동 성분이 발생함을 의미한다.

그리고 입력 영상마다 적응적으로 임계값을 갱신하기 위해 다음 식 (9)와 같이 나타낸다.

$$T_{t+1}(x,y)=\begin{cases} \alpha T_t(x,y)+(1-\alpha)(c|I_t(x,y)-S_t(x,y)| \\ \text{if } no-moving object \\ T_t(x,y) \\ \text{if } moving object \end{cases} \tag{9}$$

여기서, $I_t(x,y)$는 시각 t의 영상에서 관심영역의 블록단위 정보를 나타내고 $S_t(x,y)$는 배경영상으로 모델링된 동일한 블록 정보를 나타낸다. 또 α, c는 실험에 의해 결정된 값이며, α는 밝기(intensity) 오차로서, 0~1사이의 값으로 결정되고 c는 칼러성분의 오차로서, 1보다 큰 값으로 결정된다.

4.2 이동물체 검출

본 논문에서는 외부환경에서 조명이 변화하는 문제점을 해결하기 위하여 국부적 구간에 대한 영상의 화소 단위에서 배경성분 분석[13]을 위한 적응 가우시안 혼합 모델의 적용과 프레임간의 이동성분의 비교로 최적의 이동물체를 검출하는 알고리듬을 제안한다. 그림 15는 이동객체 추출을 위한 제안된 알고리듬의 흐름을 보여주고 있다. 여기서, 관심영역 분할단계는 다중 가우시안 모델을 적용했을 때 속도향상을 위해서 모든 영상에서 처리되는 것이 아니라 국부적 영역에서 수행을 하게 된다. 또한 이동벡터 성분추출의 가중치에 대한 비교에 의해 다중 가우시안 모델의

적용범위 및 활성화 임계치를 조정한다.

다음 그림 16은 블록 단위에서 이동객체가 발생했을 때 최상위의 이동 성분과 중간의 적응 가우시안 모델을 이용하여 제시한 배경 예측 모델을 구성한 결과이며,

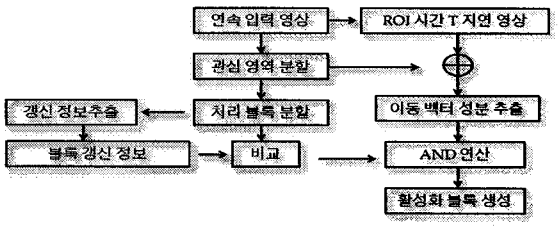


그림 15. 움직임 정보를 이용한 이동 지점 예측

Fig. 15. Moved point prediction using moved data

최하단의 배경 예측 모델은 이동 객체가 출현할 때 비교적 안정적으로 검출되는 모습을 나타내고 있다.

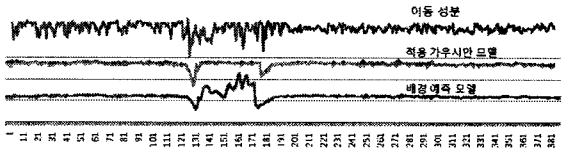


그림 16. 제시한 모델의 블록단위 성분 분석

Fig. 16. Component analysis of block unit for the proposed model

다음 그림 17은 제시한 배경 예측 모델의 특성을 나타내고 있으며 이동 객체가 출현하는 경우 상위의 GMM과 중간 라인의 AGMM에 비교하여 더욱 강인한 구분 능력을 보여주고 있다.

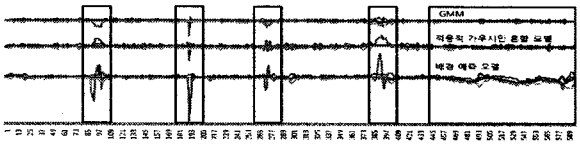


그림 17. 전역 구간에 적용한 GMM과 제시한 AGMM 배경모델의 특성 비교

Fig. 17. Quality comparision of background model for GMM and AGMM

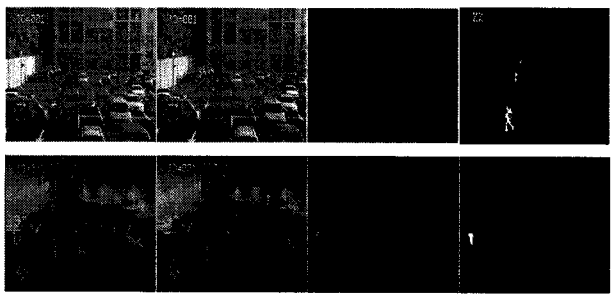


그림 18. 주간 및 야간에서 이동물체 검출 결과

Fig. 18 Detection results under day-time and night-time

위 그림 18은 주간의 복잡한 환경에서의 보행자 검출(상위영상)과 야간의 환경에 대한 보행자 검출 결과를 보였다. 좌측부터 현재 영상, 배경 예측 영상, 적응 임계치 기법(ATM) 적용 영상 그리고 마지막이 AGMM을 적용한 결과이다. 또한 상단의 낮의 환경과 하의 야간 환경에서의 실험 결과 영상이다. 그리고 그림 19에는 제안한 AGMM을 적용하여, 일반 도로에서의 보행자(왼쪽) 및 차량(오른쪽)을 검출한 결과를 보였다. 위의 왼쪽은 입력 영상, 오른쪽은 검출결과를 보였고, 아래 왼쪽은 물체인식 결과이며, 오른쪽은 그 배경영상을 보였다.

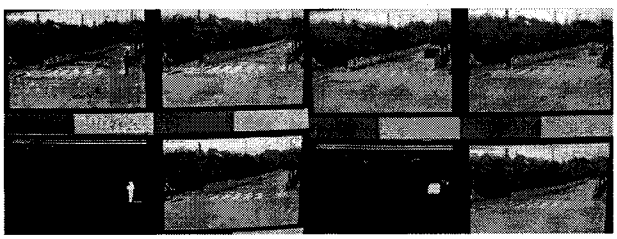


그림 19 일반 도로에서의 보행자 및 차량 검출결과

Fig. 19. Detection results for a pedestrian and a vehicle

여기서 제안한 AGMM의 성능이 효과적인 것을 알 수 있었다. 최종적으로 실험 결과를 다른 방법들과의 비교한 것을 표 1에 보였다. 실험 환경은 조명광의 상태로 분류하여 관심의 이동 물체를 인식하는 형태로 진행하였다. 각 조명광에 따라서 20회씩 실험을 진행한 결과이다. 여기서, 제안된 AGMM방법은 일반 전역 GMM 방법 및 적응 임계치법(ATM)보다 빠른 속도로 처리되었고, 보다 높은 인식률로 전경과 배경을 분리 할 수 있음을 알 수 있었다. 그림 20은 각 알고리듬의 성능을 비교한 대표적 영상을 보였다.

표 1. 실험 결과

Table 1. Results of experiments

조명광	적응 임계치법		전역 GMM		제안된방법	
	Speed	인식율 (%)	Speed	인식율 (%)	Speed	인식율 (%)
안정	30f/s	100	13.3f/s	100	30f/s	100
높음	30f/s	91	15.3f/s	92	30f/s	92
낮음	30f/s	94	14.1f/s	98	30f/s	98
인공조명	30f/s	97	14.6f/s	99	30f/s	99
어두움	30f/s	95	15.2f/s	97	30f/s	98
반사광	30f/s	83	15.5f/s	90	30f/s	92

(a) 배경영상 (b) 현재 영상 (c) ATM결과

(d) GMM 결과 (e) AGMM 결과 (f) 물체인식 결과

그림 20. 비교 실험에 사용된 영상

Fig. 20. A sample images for experiments

V. 결론

본 본문에서는 SOPC 기반 NIOS II임베디드 프로세서와 C2H 컴파일러를 적용하여 영상 감시 시스템을 구현하였다. 카메라의 영상신호 출력, 영상처리, 시리얼 통신 및 네트워크 통신의 제어를 C2H 컴파일러에 의한 IP 구성과 SOPC 기반 NIOS II 임베디드 프로세서에 기반한 각각의 IP를 효과적으로 제어할 수 있는 시스템을 구현하였다.

그리고, 보다 빠르고 환경에 강인한 영상 감시를 위한 방법으로 배경영상을 갱신하는 알고리듬을 적응 가우시안 혼합 모델(AGMM)을 제안하였다. 그 결과 주간 및 야간에서도 이동 물체를 잘 검출할 수 있었다. 이동하는 보행자 및 차량의 실험을 통해 확인하였으며, 본 알고리듬은 일반 가우시안 혼합 모델(GMM)보다 빠르고 정확하게 검출할 수 있음을 확인할 수 있었다.

참고문헌

[1] Carlo S. Regazzoni, Andrea Cavallaro, Ying Wu, "Video analytics for surveillance : Theory and Practice,", IEEE Signal Processing Magazine, pp.16-17, Sep, 2010

[2] Venkatesh Saligrama, Janusz Konrad, "Video Anormaly Identification : a Statistical Approach", IEEE Signal Processing Magazine, pp.18-34, 2010

[3] W. Hu, T. Tab, L. Wang and S. Maybank, "A Survey on visual surveillance of object motion and behaviors," IEEE Trans. Syst. Man Cybern., vol.34 no.3, pp334-352, 2004

[4] P. van der Wolf, E. de Kock, T, Henrikson, "Design and programming of embedded multiprocessors: An interface-centric approach,''Proc. Int. Conf. ardware/Software Codesign and System Synthesis, Stockholm, Sweden, Sept., pp.206-217, 2004

[5] S. Edward, L. Lavagno, E. A. Lee, "Design of embedded systems : Formal models, validation, and synthesis," Proc. IEEE, vol.85, no.3, pp.366-390, March, 1997

[6] Altera Corp, http://www.altera.com

[7] Altera Corp., "Nios II: World's Most Versatile Embedded Processor," Oct. 2004.

[8] Altera Corp., "Using C-To-Hardware Acceleration In FPGA for Waveform Baseband Processing" Nov. 2006.

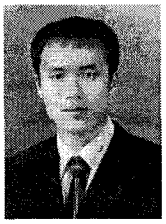
[9] Altera Corp., "Nios II C2H Compiler" Nov. 2009.

[10] S. Araki, T. Matsuoka, N. Yokogawa, "Real-time tracking of multiple object contours in moving camera image sequences," IEICE Trans. Inform. Syst., vol.E83-D, no. 7, pp.1583-1591, July 2000.

[11] J. Black, T. Ellis, and P. Rosin, "Multi view image surveillance and tracking," in Proc. IEEE Workshop Motion and Video Computing, Orlando, Fl., 5-6 Dec. 2002.

[12] A. Hampapur, L. Brown, J. Connell, et al, "Multi-scale tracking for smart video surveillance," IEEE Signal Processing Mag., vol.22, pp.38-51, March, 2005.

[13] Grimson, W. E. L., Stauffer, C. Romano, R. Lee, "Using adaptive tracking to classify and monitor activities in a site," Proc. of IEEE Computer Society Conf. on Computer Vision and Pattern Recognition. 1998.



김 동진(Dong-jin Kim)

2007년 경남대 정보통신공학과 학사 졸업.
2009년 경남대 정보통신공학과 석사 졸업
2009~현재 : 경남대 정보통신공학과 박사 과정
※ 관심분야 : FPGA설계, 임베디드 시스템

정 용배(Yong-bae Jung)

2003년 경남대학교 정보통신 공학과 졸업(공학사)
2006년 경남대학교 정보통신공학과졸업(공학석사)
2010년 동 대학원 정보통신공학과졸업(공학박사)
※ 관심분야: 영상처리, 컴퓨터비젼,
신호처리시스템 설계

박 영석(Young-seak Park)

1979년 영남대 전자공학과 학사 졸업.
1981년 한양대 전자공학과 석사 졸업
1985년 한양대 전자공학과 박사 졸업
1990~1991년 : 일본 우정성 통신총합연구소 (관서선단연구센터) 초빙과학자
1990~1991년 : 일본 긴끼이동통신센터 객원연구원
2001년~2002년 : 미국 North Carolina 주립대학(NCSU) 교환교수
2001년~현재 : 경남대 정보통신공학과 교수
※ 관심분야: Software Engineering, Web-based Software Design & Development, Pattern Recognition, Image Processing, Computer Network & Network Computing, Embedded Processor System HW/SW

김 태효(Tae-hyo Kim)

1977년 영남대학교(공학사)
1980년 동 대학원(공학석사)
1988년 동 대학원(공학박사)
1990. 12-1991. 12 미국 펜실베니아대학 Electrical Eng. 박사후과정
2007. 7-2008.7 미국 하와이대학 Electrical Eng. 해외파견 연구