http://dx.doi.org/10.7236/JIIBC.2014.14.1.169
JIIBC 2014-1-23

움직임 감지를 이용한 네트워크 카메라 기반 영상보안 시스템 구현

Implementation of Video Surveillance System with Motion Detection based on Network Camera Facilities

이규웅*

Kyu-Woong Lee*

요 약 DVR 및 NVR을 이용한 디지털 저장매체를 영상감시 시스템에서 활용하게 되면서 영상처리 모듈의 개발은 영상 보안 시장의 필수적인 요소이다. 특히 네트워크 카메라의 등장은 기존 아날로그 방식의 CCTV를 대체하면서 영상처리 모듈 개발의 필요성을 더욱 부각시키고 있다. 본 논문에서는 움직임 감지 기법을 이용한 영상 감시 서버를 설계 및 구축하고 서버에서 처리되는 영상처리 결과를 실시간으로 모바일 디바이스에서 확인 가능한 영상감시 시스템을 개발하였다. 영상처리를 위해 리눅스 기반의 서버에 오픈소스 OpenCV를 활용한 영상처리 모듈을 개발하였고, 네트워크 카메라로부터 전송되는 실시간 비디오 데이터를 저장 및 가공하여 안드로이드기반 모바일 기기에서 검색 가능한 영상감시 시스템을 구축하였다.

Abstract It is essential to support the image and video analysis technology such as motion detection since the DVR and NVR storage were adopted in the real time visual surveillance system. Especially the network camera would be popular as a video input device. The traditional CCTV that supports analog video data get be replaced by the network camera. In this paper, we present the design and implementation of video surveillance system that provides the real time motion detection by the video storage server. The mobile application also has been implemented in order to provides the retrieval functionality of image analysis results. We develop the video analysis server with open source library OpenCV and implement the daemon process for video input processing and real-time image analysis in our video surveillance system.

Key Words : Video Surveillance, Image Analysis, Motion Detection, Network Camera, OpenCV, DVR, NVR

Ⅰ. 서 론

안전의식 강화 및 국가기반시설 방재요구가 사회 전반적으로 증가함에 따라 지능형 영상보안 시스템의 구축 개발이 활발하게 진행되고 있다. 최근 개발되고 있는 영상보안 시스템은 다양한 사용자 요구사항과 시스템 장비 개선으로 인해 단순 감시가 아닌 통합 영상보안 시스템의 개발을 필요로 하고 있는 실정이다. 또한 기존의 단순한 CCTV 방식의 아날로그 영상장비에서 디지털 보안 장비로 확장되고 있으며, 특히 인코더 등의 지능형 영상

*정회원, 상지대학교 컴퓨터정보공학부 교수(교신저자)
접수일자 : 2013년 12월 23일 , 수정완료 : 2014년 1월 29일
게재확정일자 : 2014년 2월 7일
Received: 23 December, 2013 / Revised: 29 January, 2014
Accepted: 7 February, 2014
*Corresponding Author: leekw@sangji.ac.kr
Dept. of Computer Science, Sangji University, Korea

기술을 부가한 장비 및 영상인식 기술개발과 함께 지속적인 성장 산업으로 확대되고 있으며 시장규모 또한 급속히 증가하고 있다. 아날로그 영상장비는 동영상 품질 면에서 디지털 장비보다 우세한 장점을 갖고 있지만, 실시간 영상 분석 측면에서 심각한 단점을 갖고 있다[1,2]. 기존 영상보안 시스템에서 가장 많이 활용되고 있는 아날로그 방식의 CCTV는 그림 1과 같이 VTR과 같은 아날로그 저장매체에 저장되며, 실시간으로 사용자가 요구하는 영상분석이 불가능하다. VTR처럼 아날로그 매체를 사용하지 않고 디지털 컨버터를 이용하여 DVR(Digital Video Recorder) 디지털 저장매체에 저장한 후, 영상가공 및 처리를 할 수 있는 시스템을 구축하기도 한다.

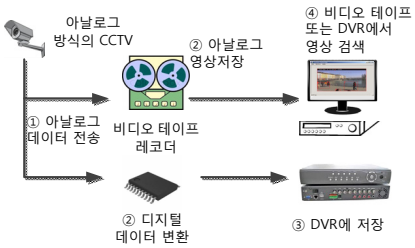


그림 1. 아날로그 방식의 영상보안 구조
Fig. 1. Video Surveillance System based on Analog Devices

기존 CCTV 방식의 영상보안 시스템의 가장 큰 단점은 실시간으로 영상 분석을 할 수 없다는 것이다. 아날로그 데이터의 특성상 이미지 분석 서버에서 데이터를 직접 활용할 수 없다. 아날로그 데이터를 가공하여 디지털 매체로 저장하기 전 까지는 실시간 분석이 불가능하다. 최근 HD급 네트워크 카메라 또는 IP 카메라로 일컫는 디지털 감시카메라 장비들이 영상보안 시장에 널리 활용되고 있다. 기존 아날로그 영상장비들의 화질보다는 다소 미흡하지만 영상보안의 목적에 충분히 만족할 만한 고화질 동영상을 제공하면서 동시에 디지털 데이터를 실시간으로 제공해 줄 수 있는 산업적 장점을 지니고 있다. 따라서 네트워크 카메라를 이용하여 영상보안 시스템의 감시장비를 교체하고 이를 저장하기 위한 영상서버 및 영상처리 서버를 구축하면 실시간 영상분석이 가능한 영상보안 시스템을 구축할 수 있다[1,2].

본 논문에서는 네트워크 카메라를 활용한 영상보안 시스템을 설계 및 구현한다. 네트워크 카메라에서 전송되는 디지털 영상을 영상 저장서버에 보내고 이를 실시간으로 분석하여 움직임 감지가 있는 영상을 추출하여 데이터베이스로 구축하고 그 결과를 웹 인터페이스 및 모바일 인터페이스를 통하여 제공해주는 영상보안 시스템을 설계 및 구현한다.

본 논문의 구성은 다음과 같다. 제2장에서 본 논문에서 제안하는 영상보안 시스템의 구조에 대하여 설명하고 제3장에서 움직임 감지 기반의 영상처리 서버 구현내용에 대하여 설명한다. 제4장에서 움직임 처리 기반 영상처리 결과를 실시간으로 조회할 수 있는 안드로이드 기반 모바일 응용프로그램 구현내용에 설명하고 끝으로 제5장에서 결론을 맺는다.

II. 영상보안 시스템 개발 환경

1. 네트워크 카메라 기반 영상보안 시스템 구조

본 논문에서 제안하는 네트워크 카메라 기반의 영상보안 시스템 구조는 그림 2와 같다. 영상감시 장비는 기존의 아날로그 데이터를 산출하는 CCTV대신 네트워크 카메라를 활용한다. 네트워크 카메라에서 전송되는 영상데이터는 실시간으로 영상조회가 가능하도록 사전처리 없이 직접 모바일 장비로 전송되어 질수 있도록 구현되었다(그림 2의 ⑤). 카메라의 줌 및 틸트 기능 및 데이터 전송 등의 제어는 네트워크 카메라 자체적으로 지원하는 웹서버를 이용하여 컨트롤된다. 움직임 감지와 같은 고급 영상처리를 위해서 영상데이터는 우선 대용량 영상 스토리지 서버로 전송되어 저장 관리된다(그림 2의 ②). 영상 스토리지 서버는 Xeon 칩의 2.4Ghz CPU와 8GB 메인 메모리를 탑재한 Fedora 계열 리눅스 서버를 이용하여 구축하였다. 저장장치로는 NAS 스토리지를 클러스터화 하여 4 테라바이트 이상 저장 가능하도록 구축하였다. 저장된 영상데이터는 영상처리를 위해 그림 2의 ③과 같이 영상처리 모듈 입력데이터로서 전송된다. 영상처리 모듈에서는 움직임 감지와 같은 영상인식 처리를 한 후 그 결과를 별도로 가공저장 하여 영상보안 시스템 관리 서버 및 모바일 장치에서 조회가 가능하도록 웹 서비스 준비를 하게 된다(그림 2의 ④). 최종적으로 그림 2의 ⑤와 같은 모바일 단말기에서 영상처리 결과를 조회하여 확인할 수 있도록 한다. 영상처리 결과 조회 인터페이스는 웹 서비스 기반으로 구축하였으므로, 모바일 장비 뿐

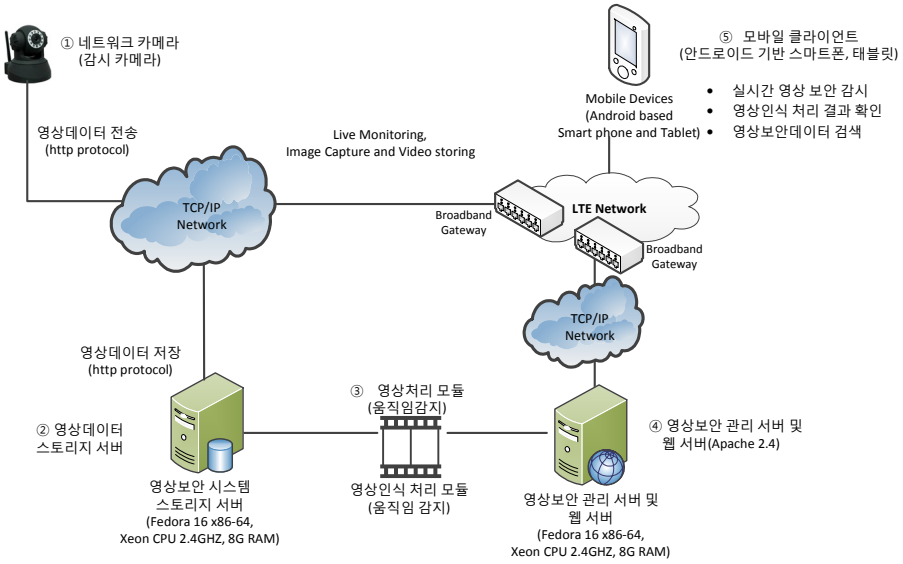


그림 2. 영상보안 시스템 구조도 및 데이터 처리 흐름
Fig. 2. Video Surveillance System Architecture and Data Flow

만이 아니라 데스크탑 컴퓨터 등 웹 접속을 지원하는 모든 장비에서 결과확인이 가능하다. 본 논문에서 구현한 영상보안 시스템을 통하여 그림 2의 ①에서 전송되는 실시간 데이터를 조회가능하고, ③에서 처리되는 영상처리 결과를 ④에서 제공하는 웹 서버를 통하여 임의의 장비에서 즉시 조회가능 하도록 구현되었다.

2. 네트워크 카메라의 영상전송 및 제어 방식

최근 카메라 기술의 발전에 따라 네트워크 카메라가 고화질 영상을 제공함과 동시에 디지털 데이터를 직접 전송할 수 있다는 장점을 보이면서 영상보안 시스템 시장에서 CCTV를 대체하고 있을 만큼 많은 관심을 받는 장비이다. 본 논문에서 사용하는 네트워크 카메라는 IEEE 802.11 b/g/n을 이용한 무선 접속 액세스 포인트를 제공하므로 클라이언트에서 직접 카메라에 대한 접근이 가능하다. 또한 웹 서버를 내장하여 HTTP 프로토콜을 이용하여 네트워크 카메라 PTZ(Pan, Tilt Zoom)에 대한 제어 모듈을 호출하여 제어할 수 있게 한다. 또한 PTZ 제어뿐만이 아니라 카메라의 MJPEG, H264, MPEG 등의 비디오 스트림에 대한 전송 제어와 간단한 이미지 관리 제어를 HTTP 프로토콜을 이용하여 할 수 있다. 네트워크 카메라에서 전송하는 데이터를 수신하고 제어할 수 있는 일반적인 네트워크 카메라 제어의 구조도는 그림 3과 같다.

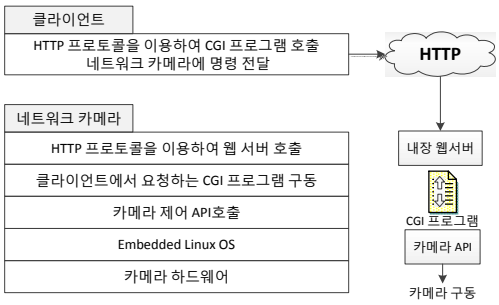


그림 3. 일반적인 네트워크 카메라 제어 구조도
Fig. 3. Control Flow of General Network Camera Devices

또한 네트워크 카메라가 HTTP 프로토콜에 의해 일반적으로 지원하는 제어 명령어는 다음 표 1과 같이 정리할 수 있다[3]. 네트워크 카메라를 HTTP 프로토콜로 제어하기 위하여 제공되는 웹 CGI 프로그램들은 대부분 표 1의 제공되는 기능을 기본적으로 포함한다. 기본 제공 CGI들은 크게 정지 영상을 캡쳐하는 기능과 실시간으로 영상을 전송하는 비디오 스트리밍 기능, 카메라의 줌인 및 줌아웃 그리고 화면을 하는 틸트 등의 제어를 위한 기능으로 구분되어 제공된다. CGI 프로그램들은 보안의 목

적으로 카메라 설치 시 기본적으로 설정된 user, password 매개변수를 포함한다. 영상 캡쳐를 위해서는 파일명을 제공하기 위해서 next_url 매개변수를 이용하고, 비디오 스트리밍에서는 화면의 해상도와 데이터 전송속도를 조정하기 위하여 resolution과 rate라는 매개변수를 이용한다. 카메라 제어에서는 command라는 매개변수를 사용하여 매개변수 값에 따라 다양한 명령어를 수행할 수 있도록 한다.

표 1. HTTP를 이용한 네트워크 카메라 제어 CGI 목록
Table 1. CGI List for control of Network Camera using HTTP protocol

카메라 기능	CGI 프로그램	매개변수
정지 영상 캡쳐	snapshot.cgi	user, password next_URL
비디오 스트리밍	videostream.cgi	user, password resolution, rate
카메라 컨트롤 및 틸트 제어	decoder_control.cgi	user, password onestep, degree
사용자권한 정보	check_user.cgi	user, password
카메라 센어제어	camera_control.cgi	uuser, password next_URL
장치 재시작	reboot.cgi	user, password
공장 초기값 설정	restore_factory.cgi	user, password netx_URL

III. 영상보안 시스템의 설계 및 구현

1. 움직임 감지 영상처리 모듈의 설계 및 구현

본 논문에서 구현한 영상보안 시스템은 움직임 감지 기능을 갖는 영상처리모듈을 개발하였다. 감시카메라로부터 실시간 전송되는 영상 데이터를 저장함과 동시에 영상처리 모듈로 전송한다. 영상처리 모듈에서는 움직임이 감지되면 움직임 부분에 대한 이동물체의 윤곽부분영상을 캡쳐하여 정지영상을 저장하고, 움직임 감지 시점의 전후 일정시간에 대한 동영상을 별도 저장하는 시스템을 구현하였다. 움직임 감지를 위한 기본적인 프리미티브 연산들은 오픈소스 OpenCV[8,9] 라이브러리를 이용하여 구현하였다.

이동 물체를 탐지하고 탐지된 이동 물체의 동작 정보를 추출하는 작업은 많은 연구가 진행되어 왔다[4, 5,6,7,8]. 대부분의 연구들은 첫째 이동물체의 정보를 추출하는 방법과 둘째 차영상(difference image)을 분석하여 이동물체에 대한 형태정보를 추론하는 방법으로 분류된다[6,10]. 이동물체의 정보를 추출하는 방법은 유사한 속도 벡터를 갖는 점 또는 영역의 그룹을 감지하여 움직임 객체에 대한 정보를 추출하는 방법이다. 차영상 기법은 서로 인접한 (i-1)번째 영상과 i번째 영상간의 명암차에 의하여 구축된 차영상들을 분석하여 움직임 물체에 대한 이동형태를 추론하는 방법[10, 11]이다.

본 논문에서 구현한 영상보안 시스템의 움직임 감지 영상처리 모듈은 차영상 기반의 움직임 감지 기법을 이용하여 구현하였다. 움직임 감지를 위한 영상처리 모듈의 개괄적인 알고리즘은 그림 4와 같다. 먼저 차영상 기반의 움직임 감지를 수행한 후, 이전 영상과 현재 영상의 차이가 주어진 임계값이상인 경우, 이동 물체에 대한 정지영상 탐지 및 저장을 위한 영상처리와 동영상 저장을 위한 영상처리로 구분되어 수행된다.

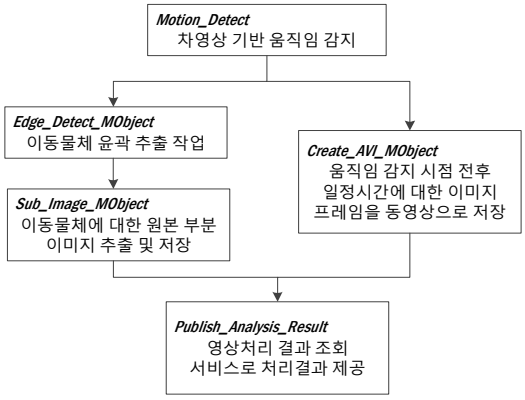


그림 4. 움직임 감지 영상처리 모듈의 개괄적 알고리즘
Fig. 4. Motion Detection Algorithm in Video Analysis Module

그림 4의 차영상 기반의 움직임 감지 함수인 Motion_Detect는 두 개의 이미지를 기반으로 각 픽셀의 RGB 바이트 값을 모두 조사하여 움직임을 감지하도록 구현하였다. 두 개의 이미지에 대한 차이를 판별하는 알고리즘은 그림 5와 같다. 제1단계에서 각 픽셀을 구성하는 RGB의 바이트마다 하위 4비트 이상이 다른 픽셀은 상이 픽셀로 판정한다. 세 개의 색상바이트에서 모두 하위 4비트가 다른 값을 갖게 되면 상이 픽셀로 판정한다. 전체 이미지의 10%이상의 픽셀에 대해 상이 판정이 발생하면 두 개의 이미지는 서로 다른 이미지로 판정하고 움직임 감지를 결정한다. 본 논문에서는 RGB 비교와 픽셀 수 비교를 위해 두 개의 임계값 Color_Threshold(하위 4비트 비교)와 MDetect_Threshold (10%이상 픽셀수)를

사용하여 응용에 적합하게 적용가능하도록 구현하였다. 본 영상보안 시스템에서 구현한 네트워크 카메라의 기본 전송 크기 영상은 640x480 프레임 크기를 기본값으로 구현하여 약 300개 이상의 픽셀이 다른 경우 움직임을 감지하도록 구현하였다. 보안이 강화되는 민감한 시스템에서는 임계값을 완화하여 세밀한 움직임 감지가 가능하다.

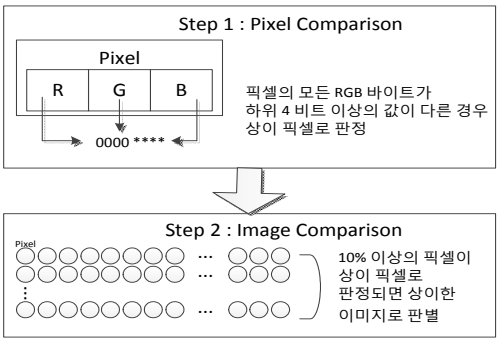


그림 5.픽셀 비교를 이용한 차영상 움직임 감지 기법
Fig. 5. Motion Detect based on Image Difference with Pixel Comparisons

움직임 감지가 판정되면 함수 Edge_Detect_MObject와 Sub_Image_MObject를 이용하여 이동객체에 대한 움직임 영역 추출 작업을 그림 6과 같이 수행된다. Edge_Detect_MObjec 함수와 Sub_Image_MObject 함수는 그림 6에 보인 바와 같이 움직임 영역 추출을 위해 크게 7단계의 과정을 갖는다. 움직임 영역 추출과정은 기본적으로 OpenCV[8,9]에서 제공하는 cvAbsDiff 및 cvSub 라이브러리를 이용하여 1단계 1차 움직임 영역추출과정과 6단계 2차 움직임 영역 추출과정으로 구성한다. 1차 움직임 영역 추출과정에서 움직임이 미약한 경우 윤곽선 추출이 모호한 경우가 있어 영역 추출에서 객체 경계의 오류를 범하는 경우가 많다. 본 논문에서는 이러한 미약한 움직임 객체의 윤곽선을 보다 강화하고 효율적으로 탐지하기 위해 단계3부터 단계5를 통해 움직임 영역 암흑화 과정, 백색화 과정 및 보정팽창 과정을 반복적으로 적용한다.

이전 이미지와 현재 이미지의 픽셀간 차이에 대한 절대값을 계산하여 1차 움직임 영역 추출과정(단계1)의 결과를 생성한다. 결과 이미지에서 움직임 없는 부분은 픽셀값이 0에 수렴하고 움직임이 있는 부분만 미약한 색상을 갖게 된다. 단계 2, 5에서 수행하는 보정 및 팽창작업은 cvThreshold 함수를 이용하여 흑백화 작업을 진행하고 cvDilate 함수를 이용하여 윤곽선 테두리에 대한 이미지 강조작업을 통하여 움직임 영역을 보정하는 작업을 수행한다.

단계 3의 움직임 영역 암흑화 작업은 원본 이미지에서 단계 2의 결과 영상의 픽셀차를 통해 얻는다. 결과 이미지에서 움직임 없는 영역은 원본 색상을 갖고 움직임 영역은 음수의 값을 갖게 되어 흑색을 갖게 된다. 이 결과 이미지를 다시 반복적으로 원본 이미지와 절대값 차를 계산하여 단계 4의 움직임 영역 백색화 작업을 수행한다.

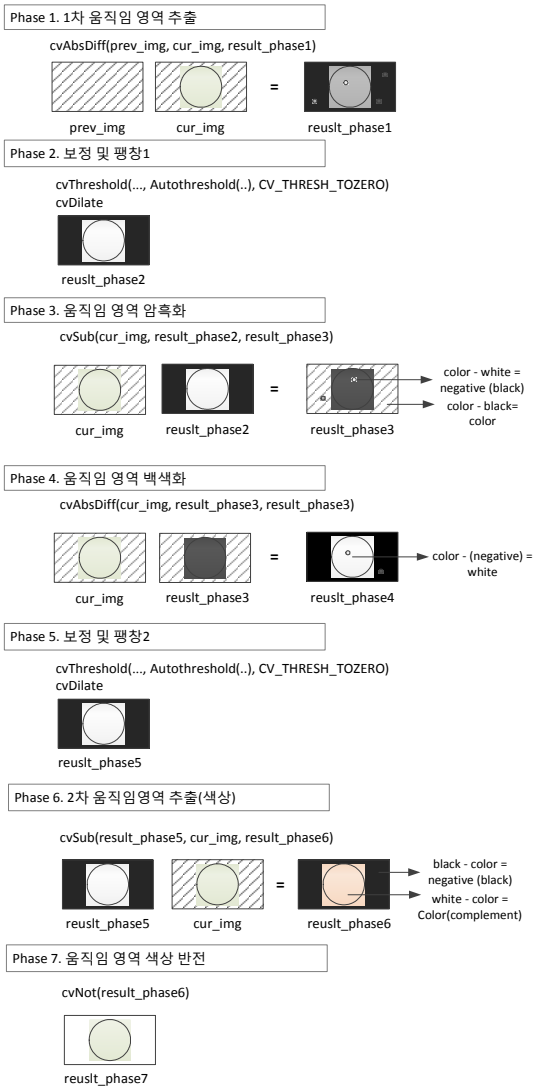


그림 6. 이미지 차 연산을 반복 적용하는 움직임영역 추출 알고리즘
Fig. 6. Extraction of Motion Region with Recursive Image Comparisons

단계 4의 결과이미지는 움직임이 없는 영역은 흑색, 움직임 영역은 백색으로 변환하려는 시도이다. 이 결과는 단계 5에서 보정 및 팽창작업을 통해 결과이미지 정제 작업을 수행한다. 이 결과의 형태는 단계 2의 형태와 같은 의미의 결과이다. 즉 단계 1의 결과는 단계 4의 결과와 같은 의미로서 움직임 영역만 특정색으로 보이게 하려는 의도이다. 본 논문에서는 단계 3과 4에서 반복적으로 픽셀 차 연산을 수행하게 하여 암흑화와 백색화 작업을 함으로서 움직임이 미약한 영상에 대해 뚜렷한 윤곽을 얻을 수 있도록 제안하였다.

2차 움직임 영상 추출과정(단계 6)은 단계 5의 결과이미지에서 현재 원본 영상의 차이를 계산하여 수행한다. 단계 5의 결과이미지는 픽셀값 255와 0으로 구성하도록 의도한 이미지이므로 원본 이미지와 cvSub 연산을 수행하면 움직임 없는 영역은 음수값을 띄어 흑색을 갖게 되고 움직임 영역은 255에 대한 보수값을 갖는 결과를 갖게 된다. 최종적으로 단계 6의 결과이미지를 반전 시키면 움직임 영역에 대한 원본 색 이미지를 얻을 수 있게 된다. 본 논문에서 제안한 픽셀 차연산을 반복적으로 적용한 움직임영역 추출 알고리즘에 의해 움직임객체가 탐지되는 과정을 그림 7에 기술하였다.

그림 7. 픽셀차 반복 적용을 통한 움직임 객체 추출 예
Fig. 7. Example of Motion Region Extraction with Pixel Comparison

각 단계별로 수행되는 과정에서 움직임 객체에 대한 윤곽을 반복적으로 픽셀차를 구해 객체의 윤곽선을 명확하게 하는 중간결과 영상을 확인 할 수 있다. 그림 7의 ①과 ⑤에 대한 결과 이미지는 움직임 영역 추출에 대한 같은 의미의 결과 작업이며 그 중간 단계의 작업은 픽셀차를 구하기 위한 반복적 단계이다. 최종적으로 ⑦의 단계에서 움직임 영역을 추출한 최종 영상을 구할 수 있다. 본 논문에서 제안한 방법은 반복적 픽셀차 연산을 수행하므로 미세한 픽셀차이에 대한 움직임 객체를 추출할 수 있다.

2. 영상처리 결과 검색 인터페이스 구현

본 절에서는 영상처리 모듈에서 분석된 영상 처리 결과를 사용자에게 제공하는 결과 조회 인터페이스의 구현에 대한 내용을 기술한다. 제2장의 설명한 바와 같이 영상처리는 그림 2(②, ③)의 영상처리 스토리지 서버와 영상인식 처리 모듈에서 수행된다. 움직임 감지와 움직임 영역 이미지 추출등의 결과가 생성되면 그림 2(④)의 웹 서버에 그 결과를 전송하고 웹 인터페이스를 통해 영상인식 결과를 접근할 수 있도록 구현하였다. 또한 웹 인터페이스는 안드로이드 기반 모바일 기기에서 쉽게 접근할 수 있도록 안드로이드 응용 프로그램으로도 구현하였다.

모바일 응용 프로그램의 주요 기능은 네트워크 카메라로부터 전송되는 실시간 영상을 직접 조회할 수 있는 기능, 영상인식 처리결과를 조회하고 검색할 수 있는 기능, 네트워크 카메라의 하드웨어 정보 설정 및 PTZ(pan-tilt-zoom) 조정을 설정하는 기능을 포함하여 구현되었다. 모바일 응용 프로그램 클래스 설계도 및 각 클래스 XML목록은 그림 8과 같다. IPcam_InputStream 클래스는 수신 받은 패킷에서 영상 데이터를 추출하는 작업을 한다. 영상 데이터를 저장할 버퍼 공간을 생성하고 영상 데이터의 시작을 나타내는 SOI(Start of Image)를 이용하여 패킷의 포인터를 이동한다. 이동된 포인터가 가리키는 영상데이터를 미리 생성한 버퍼 공간에 저장한다. 저장된 영상 데이터는 비트맵 형식으로 변환하여 반환(return)한다. IPcam_view 클래스는 IPcam_InputStream 클래스의 반환 값을 사용하여 영상 데이터를 비트맵형식으로 제공받는다. 비트맵을 프레임 레이아웃에 삽입하여 이미지 영상을 화면에 출력하고, 영상을 실시간으로 받기 위해 스레드 객체를 사용하여 백그라운드 작업으로 카메라 컨트롤 CGI 프로그램 명령어를 송신하여도 화면 출력은 지속적으로 이루어지면서 카메라 컨트롤이 가능하게 된다. 또한 현재 영상을 캡처하여 모바일 어플리케이션 내에 특정 폴더를 생성 후 JPG형식으로 이미지를 저장한다.

클래스 View_Mdtect_Result는 영상처리 결과목록을 조회하기 위한 클래스 설계이다. 이미지 목록과 동영상 목록 검색을 위해 View_Mdtect_Imglist와 View_Mdtect_avilist 클래스를 하위클래스로 설계하여 이미지와 동영상을 조회할 수 있도록 한다.

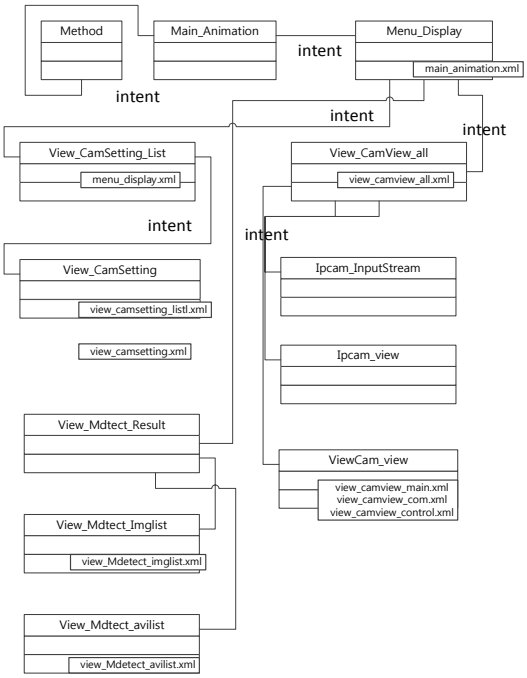
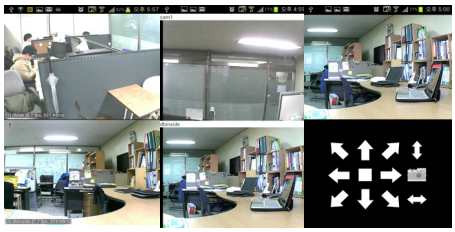
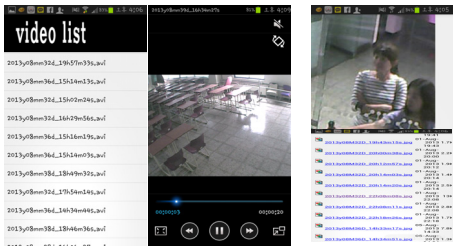


그림 8. 모바일 인터페이스를 위한 클래스 설계도
Fig. 8. Class Design of Mobile Applicatio Interface



(1) 모든 카메라 뷰 스크린(Thumbnail Screen) (2) 개별 카메라 스크린 및 PTZ 컨트롤



(3) 영상처리 결과 조회 화면(움직임 탐지 동영상 및 이미지)

그림 9. 모바일 인터페이스 화면 예
Fig. 9. Mobile Interface Screen Example

그림 9는 영상처리 결과 조회를 위한 모바일 응용 프로그램 화면 예이다. (1)은 다수의 네트워크 카메라로부터 실시간 전송되는 동영상을 조회하는 화면이고 (2)는 개별 카메라에 대한 PTZ 컨트롤을 하는 화면이다. (3)화면은 본 논문에서 개발한 움직임 감지 영상처리 모듈의 처리결과를 검색하고 조회하는 화면이다.

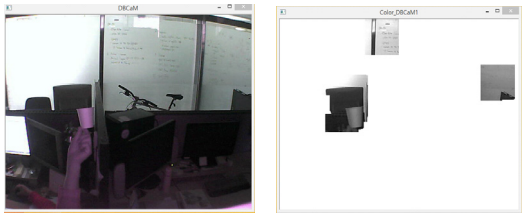
IV. 영상처리 인식 결과 분석

본 논문은 픽셀차의 반복 연산을 수행하여 미세한 움직임 영역을 검출하는 움직임 감지 기반 영상보안 시스템을 구현하였다. 영상보안 시스템의 목적은 침입탐지를 기본으로 수행하는 것이므로 미약한 움직임에 대해서 민감하게 동작해야 한다. 따라서 움직임 감지를 위한 기본 절차를 크게 세 가지로 구성하여 영상보안의 목적에 부합하도록 하였다. 첫째 차영상 기반의 움직임 감지 단계에서는 하나의 픽셀에 대한 RGB 한 바이트의 하위 일정 비트들을 비교하여 미세한 이미지 차이에 대하여 감지할 수 있도록 제안하고 구현하였다. 한 픽셀의 미세한 차이를 감지한 후, 기본 이미지 픽셀 수의 일정 비율을 넘는 픽셀수가 서로 다르게 판정되면 움직임으로 판단하여 침입감지를 판정할 수 있도록 하였다. 한 픽셀에 대한 비교 연산에 사용하는 Color_Threshold(하위 4비트 비교)와 일정 픽셀수가 다른 픽셀로 판단되는지 검사하는 MDetect_Threshold (10% 이상 픽셀)값은 시스템의 파라미터로 정의하여 응용에 적합하도록 구축할 수 있다.

둘째, 움직임 감지가 되면 움직임 객체에 대한 영역 추출작업을 수행하는 단계이다. 첫 번째 단계에서 미약한 움직임에 대해 움직임 감지를 한 경우 움직임 객체 영영 추출은 윤곽선 탐지 오류율이 높다. 본 논문에서 움직임 영역 추출을 위해 움직임 객체 부분에 대한 암흑화와 백색화 단계를 통해 픽셀차 연산을 반복적으로 수행하고 보정 및 팽창작업을 통한 정제작업을 하여 미세 움직임 영역에 대한 검출 확률을 높였다. 셋째, 움직임 영역 추출이 완료되면 영역에 대한 이미지를 추출하여 저장하고, 움직임 감지 시간을 기준으로 앞뒤 일정 프레임을 동영상을 저장하도록 하였다. 저장된 영상처리 결과는 웹 인터페이스를 통해 모바일 및 데스크 탑에서 쉽게 접근 가능하도록 구현하였다.

본 논문은 미세 움직임을 탐지하고 영역을 추출하는

방법을 제안했으나 네트워크 카메라로부터 전송되는원본 입력영상의 조명이 너무 밝거나 낮은 경우 추가적으로 밝은 영역에 대해 움직임으로 탐지되는 경우가 있다. 그림 10은 조명이 너무 높은 영역과 낮은 영역이 있는 원본 이미지에 대해서 미세한 움직임을 주고 영역을 탐지했을 때 움직임 이외의 영역이 탐지되는 상황을 보이고 있다. 특정한 일부 영역에 너무 밝은 조명을 주는 경우 움직임으로 판단하는 오차가 있다. 그러나 조명이 일정한 경우 본 논문에서 제안한 움직임 감지 및 영역 추출기법은 95%이상의 성공률을 보였다.



조명이 너무 낮은 영역과 너무 높은 영역이 있는 이미지 | 반복 연산을 통해 특정 영역이 검출되는 오류

그림 10. 원본 입력영상의 조명이 일정하지 않은 경우 움직임 영역 탐지

Fig. 10. Motion Detection with Poor Quality-Light Image

V. 결론

영상보안 시스템의 감시 카메라 장비가 네트워크 카메라와 같은 디지털 장비로 전환되면서 고급 영상처리 기능에 대한 요구사항이 증가하고 있다. 본 논문에서는 네트워크 카메라를 이용한 움직임 감지 기반 영상보안 시스템을 구현하였다. 영상 스토리지 서버 및 영상 처리 모듈에서는 네트워크 카메라와 같은 디지털 감시장비로부터 전송되는 이미지를 실시간으로 분석하여 움직임 탐지 및 움직임 객체의 영역 추출을 수행하고 그 결과를 웹서버에 전송하여 사용자가 손쉽게 접근할 수 있도록 구현하였다. 특히 영상처리 모듈에서 제안한 움직임 탐지 기법은 미세 움직임을 탐지 할 수 있도록 픽셀내 RGB바이트 차연산을 수행하였고, 미세 움직임에 대한 영역 추출을 위해 암흑화 및 백색화 연산을 반복 수행하여 움직임 윤곽을 확대할 수 있는 방안을 제안하였다. 추가적으로 조명이 높거나 해상도가 낮은 경우 특정영역이 검출되는 오차를 개선하는 연구를 진행중이다.

References

[1] Yang, Ming-Jiang, et al. "Cost effective ip camera for video surveillance." Industrial Electronics and Applications, 2009. ICIEA 2009. 4th IEEE Conference on. IEEE, 2009.

[2] XU, Ning. A survey of sensor network applications. IEEE Communications Magazine, 2002, 40.8: 102-114.

[3] FOSCAM CGI/SDK Library Manual http://www.foscam.us

[4] MIGLIORE, Davide A.; MATTEUCCI, Matteo; NACCARI, Matteo. A revaluation of frame difference in fast and robust motion detection. In: Proceedings of the 4th ACM international workshop on Video surveillance and sensor networks. ACM, 2006. p. 215-218.

[5] JING, Guo; SIONG, Chng Eng; RAJAN, Deepu. Foreground motion detection by difference-based spatial temporal entropy image. In: TENCON 2004. 2004 IEEE Region 10 Conference. IEEE, 2004. p. 379-382.

[6] CUTLER, Ross; DAVIS, Larry S.. . Robust real-time periodic motion detection, analysis, and applications. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2000, 22.8: 781-796.

[7] Wang, Yiwei, John F. Doherty, and Robert E. Van Dyck. "Moving object tracking in video." the Proceedings of 29th Applied Imagery Pattern Recognition Workshop, pp.95-101, IEEE, 2000.

[8] AmmarAnuar, KhairulMuzzammilSaipullah, NurulAtiqah Ismail, and Soo Yew Guan. "OpenCV Based Real-Time Video Processing Using Android Smartphone." International Journal of Computer Technology and Electronics Engineering (IJCTEE) Vol 1, No. 3, pp58-63, 2011

[9] OpenCV, Open source Computer Vision library. In http://opencv.willowgarage.com/wiki/, 2009.

[10] Zhong-Yong Che, Sangchul Kim, “A Surveillance System Using Images and Movement Detection

Sensors" Journal of the Institute of Internet, Broadcasting and Communication, VOL. 13 No. 1, February 2013

[11] Hye-Youn Lim, Dae-Seong Kang, The Moving Object estimation Using an Efficient Background Extraction in the Outdoor Environment, Journal of Korean Institute of Information Technology, Vol 7, No. 3, 2014. 226-231.

저자 소개

이 규 웅(정회원)

• 1986년 2월 : 한국외국대학교 전자계산학과(공학사)
• 1992년 2월 : 서강대학교 컴퓨터학과(공학석사)
• 1998년 2월 : 서강대학교 컴퓨터학과(공학박사)
• 1998년 3월 ~ 2000년 8월 : 한국전자통신연구원(ETRI) 선임연구원
• 2000년 9월 ~ 현재 : 상지대학교 컴퓨터정보공학부 교수
<주관심분야 : 데이터베이스 시스템, 클러스터 파일 시스템>
• Email : leekw@sangji.ac.kr

※ 이 논문은 2011학년도 상지대학교 교내연구와 2012년 중소기업청 기업부설연구소 지원사업의 지원을 받아 수행된 것임