

# 확장된 작은 다이아몬드 탐색 알고리즘을 이용한 효율적인 보안 시스템

김선재, 오해석  
경원대학교 전자계산학과  
e-mail:uzics@ku.kyungwon.ac.kr

## An Expanded Small Diamond Search Algorithm for Efficient Security System

Sun-Jae Kim, Hae-Seok Oh  
Dept of Computer Science, Kyung-Won University

### 요 약

보안에 대한 관심은 정보통신 기술이 발달함에 따라 나날이 증가하고 있고 개인과 주변 사회의 환경을 급격히 변화시키는 계기가 되었다. 하지만 기존의 보안 장비들은 녹화를 끊임없이 계속해야 함에 따라 저장매체의 용량에 대한 문제와 관리에 따른 부담이 존재하였다. 이에 본 논문에서는 보안 시스템이 기본적으로 갖춰야 하는 모니터링, 녹화, 재생 등의 기능을 제공하고 대규모 저장장치에 대한 필요를 배제하기 위해서 ESDS 알고리즘을 이용하여 기준치 이상의 움직임이 발생했을 때 그 화면을 기록하며 JPEG format 방식을 채택하여 용량에 따른 부담을 줄인 보안 시스템을 설계 및 구현하였다.

### 1. 서론

정보통신 기술의 발달은 보안에 대한 의식과 개인의 환경을 급격히 변화시키는 계기가 되었다. 하지만 기존의 보안 장비들은 녹화를 끊임없이 계속해야 함에 따라 방대한 저장매체에 대한 관리의 필요성이 항시 존재하였다.

이에 본 논문에서는 위에서 언급한 기존 보안 프로그램의 한계를 극복하기 위한 방안으로 대규모의 저장장치에 대한 필요를 배제하기 위해서 확장된 작은 다이아몬드 탐색 알고리즘(expanded small diamond search, ESDS)[2][4]을 이용하여 기준치 이상의 움직임이 발생했을 때에 한해서 그 화면을 기록하였고 용량에 따른 부담을 줄이기 위해 Bitmap format 대신 JPEG format 방식을 통한 관리 방안을 제안한다.

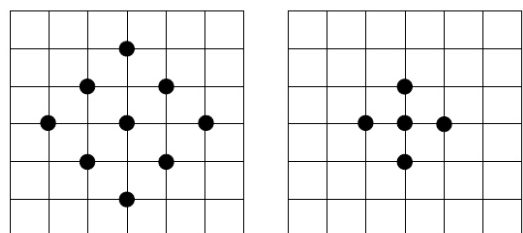
본 논문의 구성은 다음과 같다. 먼저 2장에서는 영상을 획득함에 있어 필요한 관련연구에 대해서 살펴본다. 3장에서는 본 시스템의 전체 구성과 기본 설계에 대해서 소개하고, 4장에서는 실제 구현된 본 시스템의 성능 평가 결과를 기술 하였다. 끝으로 5장에서 결론 및 향후 보안 시스템에서 개선되어야 할 사항들에 대해 논의 하였다.

### 2. 관련연구

#### 2.1 다이아몬드 탐색 알고리즘

다이아몬드 탐색(Diamond Search, DS) 알고리즘은 (그림 1)과 같이 LDSP(Large Diamond Search Pattern)와 SDSP(Small Diamond Search Pattern) 두 가지 패턴을 사용한다. LDSP는 9개의 탐색 점으로 구성되어 있으며 SDSP는 5개의 탐색 점으로 구성되어 있다.

DS 알고리즘의 탐색과정은 중심에 위치한 탐색 점에서 최소블록왜곡(Minimum Block Distortion, MBP)값이 구해질 때까지 LDSP를 반복적으로 적용하고, 만약 중심에 위치한 탐색 점에서 MBP 값이 구해질 경우 마지막 단계로 SDSP를 적용한다. SDSP의 5개 탐색 점들 중에서 MBP 값을 갖는 탐색 점을 그 블록의 움직임벡터로 고려한다.



(그림 1) DS 알고리즘에서 사용되는 LDSP와 SDSP

### 2.2 영상에서의 컬러 표현

우리가 컬러 영상정보를 인지하는 방법은 세 가지 지각 변수로 분류되는데 색상(hue), 채도(saturation), 명도(brightness)[1]가 그것이다. 또한 높은 대비를 가지는 영상들은 어두운 영역과 밝은 영역의 범위가 크고 영상이 광도를 전적으로 잘 표현한다. 대비(contrast)는 영상의 가장 어두운 영역으로부터 가장 밝은 영역의 범위를 말하며 Contrast의 범위는(식 1)과 같이 수학적으로 표현되어 나타낸다.

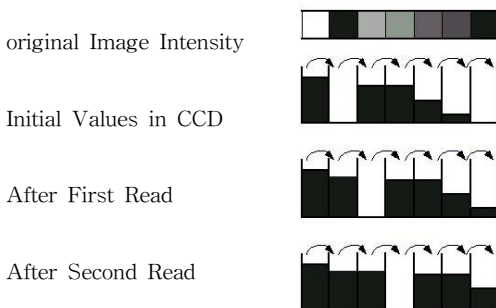
$$Contrast = \frac{I_{MAX} - I_{MIN}}{I_{MAX} + I_{MIN}} \quad (\text{식 1})$$

여기서  $I_{MAX}$  와  $I_{MIN}$  은 어떤 영역이나 영상의 명암도의 최대와 최소값이다. 영상의 대비가 증가하면, 관찰자는 더욱 상세하게 볼 수 있다. 이것은 영상에서의 정보의 총량은 전혀 증가하지 않는 순수한 지각 작용이다. 우리의 지각 작용은 순수한 광도의 강도에 민감하기 보다는 광도의 대비에 더 민감하다.

### 2.3 영상 획득과 Image Filtering

영상획득이 발전하게 된 주요 원인은 전하결합소자장치(Charge-Couple Device ; CCD)의 발명이다. 이전의 영상 획득 장치와 비교하여 CCD는 작고, 값 싸며, 빠르다. CCD는 밀폐된 공간에 있는 집광장치들의 배열로 입사되는 광자 에너지의 패턴을 아날로그 신호로 변환하도록 설계되었다. CCD는 두 가지 기능을 수행한다. 첫째는 광자 에너지를 전기 전하로 변환하는 것이고, 다음은 이 전하를 판독할 수 있도록 전하를 이리저리 이동한다.

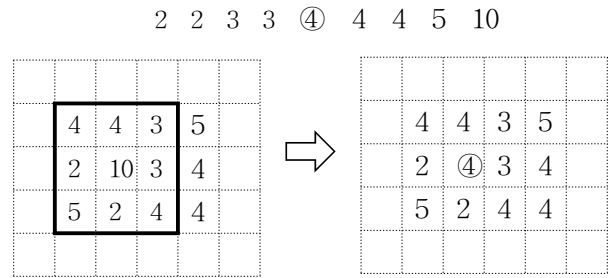
이렇게 전하를 이동시키는 처리는 버킷 브리케이드(bucket brigade, 전하를 소화하기 위한 버킷 릴레이의 줄)로 알려져 있는데, 하나의 값이 판독된 후에는 다른 모든 나머지 값들은 그에 따라서 시프트(shift)한다. CCD를 판독하는 것은 시프트 레지스터로부터 일련의 데이터 스트림을 판독하는 것과 유사하다. 충분한 판독이 이루어지면, 아래 (그림 2)과 같이 그 데이터는 다음 판독을 위해 시프트된다.



(그림 2) CCD 동작

획득한 이미지에 잡음(noise)이 있다고 할 때 잡음의 성질을 이용하여 잡음 제거를 행하는 수법인 smoothing기법으로는 잡음이나 에지를 고려하지 않고 모두 흐리게 해버리기 때문에 잡음은 제거할 수 있어도 원하는 결과 영상이 흐리게 되고 만다. 이와는 달리, 훌륭하게 영상 데이터의 에지를 흐리게 하지 않고 잡음을 제거할 수 있도록 고안된 것이 미디언(Median) 필터[1] 기법이다.

(그림 3)와 같은 농도를 가진 영상이 있다고 가정하자. 그림에서 3x3의 영역내(굵은 선)의 9개의 화소를 오름차순으로 정렬하면 다음과 같다.



(그림 3) 일반적인 화소값을 구하는 예

이 때의 중앙 값(이것을 median이라 한다)은 9개의 화소를 기준으로 하므로 왼쪽에서 5번째에 위치한 포인트점으로 구할 수 있는 4가 농도값이 된다.. 위의 예에서 10이라고 하는 원래의 화소는 잡음 성분이 있는데, 확실히 제거 되었다. 이것은 주변과 비교하여 극단적으로 농도의 차이가 있는 것은 크기의 순서로 나열했을 때, 왼쪽이나 오른쪽으로 치우치게 되어 중앙값으로 선택되지 않기 때문이다. 이와 같이 미디언 필터(Median Filter)는 어떤 화소 주변의 영역내의 화소 농도의 중간값을 구하여 원하는 화소의 농도로 처리하는 것이다.

## 3. 제안하는 시스템

### 3.1 시스템 기본 구성

시스템 설계의 기본 구조는 (그림 4)과 같으며 카메라 영상 영상이 메인 프로그램에 전송되고 메인 프로그램은 재생, 보안계획, 이미지보정, 통계, 옵션, 보안계획, 이미지 보정, 통계, 옵션 모듈로 구성되어 있다.



(그림 4) 시스템 구조 차트

재생모듈에서는 탐지영역 지정 및 감도를 설정하고 JPEG 압축으로 녹화를 하며 보안계획 모듈에서는 이전 녹화 파일의 재생과 앞으로의 침입탐지에 대한 보안설정을 할 수 있다. 이미지 보정 모듈에서는 이미지 필터링을 통하여 보다 나은 영상을 획득할 수 있고, 통계 모듈에서는 일별, 월별, 연간 통계를 보여주며 옵션 모듈에서는 비디오 설정, 파일 저장기간 설정, 사용자 계정 설정 등의 기능을 가지고 있다.

### 3.2 시스템 설계

기본적인 시스템의 설계는 먼저 특정 제어되는 주요 기능별 Class를 선정하고, 그 Class를 각 기능별 Class를 모두 포함하게 될 프로그램의 핵심 공간인 Main Application에 구성하였다. 그러나 가장 중요한 기능인 Image의 획득과 관련된 부분은 Main Application부분에서 직접 개발하였다.

#### 가) Camera → Main Application

(이미지에 대한 검사 및 저장)

설치되어진 Camera에서 받아들여진 이미지 정보는 일단 Main Application의 이미지 검사의 단계를 거쳐서 설정 감도에 따라 저장되어지거나 버려지게 된다.

일단 받아들여진 이미지의 경우 JPEG으로 압축되게 되며 이는 \*.idx 및 \*.ime의 확장자 명을 가진 본 프로그램의 기본 file 확장자로 변형된다. JPEG으로 생성되어지는 시간과 Size 및 감지도를 포함하게되는 정보의 부분은 \*.idx의 file로 실제의 Image의 경우는 \*.ime의 file로 각각 저장된다. 이 때의 정보들은 DataBase로 저장된다.

#### 나) Main Application → Play

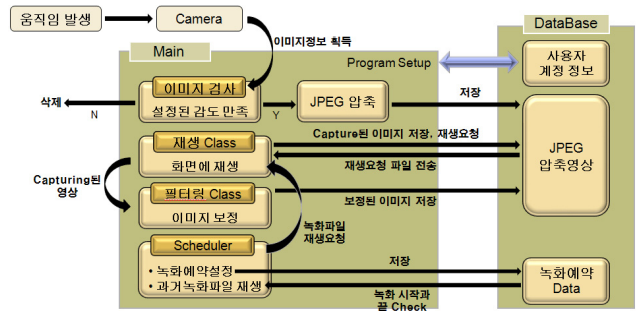
이미 저장되어지거나 생성되어지는 Image의 경우 Client가 재생을 선택하여 동영상처럼 볼려고 할 때마다 DataBase에서 해당 file을 찾고 찾아진 file은 재생 Class에서 화면에 재생하게 된다.

#### 다) Main Application → Filtering

재생 중이거나 Capturing된 영상은 Filtering Class로 가서 Image의 보정작업이 수행된다.

#### 라) Scheduler → DataBase or Play Class

Scheduler에서 설정에 관련된 작업을 하게 된다면 그 정보는 DataBase에 남겨지게 되고 이후 Main Application 부분에서 그 설정 data에 따라 녹화시작과 끝을 자동으로 check하게 된다. 또한 Scheduler에서는 과거일에 대한 녹화 file을 재생하는 기능을 하게 된다.



(그림 5) Data Flowchart

### 3.3 움직임 탐지

사용자가 지정한 수치를 벗어나면 움직임으로 간주하고 녹화한다. 이 구현 부분은 callback function내에서 있다. 메모리를 잠시 저장하고 현재의 메모리를 비교 하여 값을 비교하는 방법으로 움직임을 탐지한다. 움직임 탐지에 대한 알고리즘은 아래와 같다.

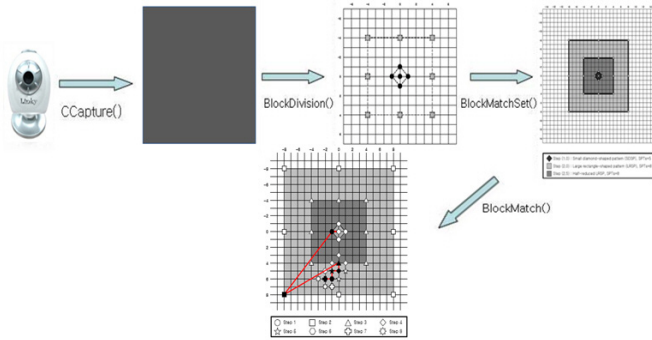
```

LRESULT CALLBACK FrameStreamCallbackProc(HWND
hwnd, LPVIDEOHDR lpVHDR) {
for (nOffset = 0 ; nOffset <(int)lpVHDR-> dwBuffer
Length; nOffset++) {
// 현재 픽셀 이미지
pixel = *(lpVHDR->lpData + nOffset);
// 픽셀을 하나 얻어와서 기존과 다를 경우
if (pixel > *(g_pOldImageBuffer + nOffset))
{ // sw2 : 민감도
if(pixel - *(g_pOldImageBuffer + nOffset) >
(int)sw2) // 민감도보다 크면 {
// 사용자 영역 감시 루틴
if(g_Recording) {
if( (x/m_fDiv >= g_UserDetectRect.left &&
x/m_fDiv <= g_UserDetectRect.right) && (
240-y >= g_UserDetectRect.top && 240-y <=
g_UserDetectRect.bottom ))
dwUserDetectCount++;
}
if(g_bMoveRect) // 감지 센서 {
if(pixel - *(g_pOldImageBuffer +
nOffset) > (int)sw2+30)
{
..... }
}
}
}
}

```

본 시스템에서 사용할 알고리즘은 3 단계 탐색 (three-step search, TSS)[3] 절차를 가지고 있으며 효율적인 3단계 탐색 (efficient three-step search, ETSS)[5]에서 더욱 발전한 ESDS[2][4] 알고리즘을 사용하며 (그림

6)의 방법으로 움직임을 탐지한다.

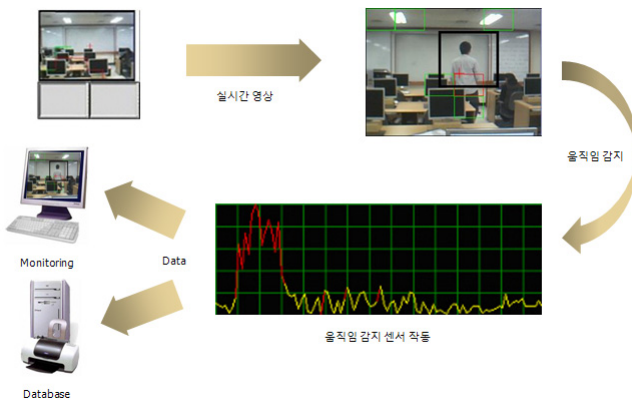


(그림 6) ESDS 알고리즘을 이용한 움직임 탐지 동작

#### 4. 구현 및 성능평가

(그림 7)은 실제 본 시스템을 구현하여 실제로 동작하는 화면을 보여준다. 본 시스템은 침입 탐지 기능외에 재생, 녹화, 사용자 영역 지정 감시등의 기능을 가지고 있다. 우선 PC 카메라에서 영상을 받아오면 블록으로 나누어 설정한 감지도 이상의 움직임이 발생할 시에 저장을 시작하는 과정을 시험하였다. 또한 자체적인 JPEG Player를 개발하여 JPEG format으로 저장된 영상을 재생할 수 있도록 하였으며 사용자 영역 지정 움직임 탐지등의 기능이 제대로 작동하는지를 시험하였다.

본 시스템을 원활히 구동하기 위해서는 윈도우 기반의 OS를 지원하는 컴퓨터와 512M 이상의 RAM이 필요하다. 그리고 감지도 설정 및 사용자 영역 지정으로 외부환경에 영향을 거의 받지 않는 편이다. 움직임의 감지는 정확하게 감지되었으며 또 영역을 설정한 후의 감지도 성공적으로 동작하였다.



(그림 7) 움직임 탐지 모니터링

#### 5. 결론

본 논문에서는 ESDS 알고리즘을 이용하여 기준치 이상의 움직임이 발생했을 때에 한해 화면을 기록함으로써 효율적인 영상 획득 방안을 제안하였고 이로써 기존의 보안

장비들이 가지는 저장매체의 용량에 대한 단점을 극복하였다. ESDS 알고리즘을 이용한 보안 시스템의 실제 구현과 동작원리 분석을 통해 영역 지정 감시 및 재생, JPEG format 압축으로 인한 저장공간의 용량을 대폭 줄이게 되었다.

향후 과제로는 움직임 감지가 발생될 때마다 증가하는 CPU 사용량에 따른 보다 효율적인 알고리즘의 설계 및 구현과 무선통신과 연동하여 거리상의 제약이 없는 시스템으로 향상 시키고자 한다.

#### 참고문헌

[1] G. J. Yang T.S Huang, "The effect of media filtering on edge location and detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-9, no. 2, Mar. 1987.

[2] S. Zhu and K. K. Ma, "A new diamond search algorithm for fast block matching motion estimation," *IEEE Trans. Image Process.*, vol. 9, no. 2, pp. 287 - 90, Feb. 2000.

[3] S.Zhu and K. K. Ma, "A new diamond search algorithm for fast block matching motion estimation," *IEEE Trans. Image Process.*, vol. 9, no.2, pp. 287-290, Feb. 2000

[4] T. Koga, K. Inuma, A. Hirano, Y. Iijima, and T. Ishiguro, "Motion compensated interframe coding for video conferencing," in *Proc. Nat. Telecommun. Conf.*, New Orleans, LA, Nov. 29 - Dec. 3 1981, pp. G5.3.1 - G5.3.5.

[5] Xuan Jing and Lap-Pui Chau, "An efficient three-step search algorithm for block motion estimation," *IEEE Trans. Multimedia.*, vol. 6, pp. 435 - 438, June. 2004.