

Real-time small target detection method Using multiple filters and IPP Libraries in Infrared Images

Chul Joong Kim*, Jae Hyup Kim**, Kyung Hyun Jang***

Abstract

In this paper, we propose a fast small target detection method using multiple filters, and describe system implementation using IPP libraries. To detect small targets in Infra-Red images, it is mandatory that you should apply a filter to eliminate a background and identify the target information. Moreover, by using a suitable algorithm for the environments and characteristics of the target, the filter must remove the background information while maintaining the target information as possible. For this reason, in the proposed method we have detected small targets by applying multi area(spatial) filters in a low luminous environment. In order to apply the multi spatial filters, the computation time can be increased exponentially in case of the sequential operation. To build this algorithm in real-time systems, we have applied IPP library to secure a software optimization and reduce the computation time. As a result of applying real environments, we have confirmed a detection rate more than 90%, also the computation time of the proposed algorithm have been improved about 90% than a typical sequential computation time.

▶ Keyword : small target detection, spatial filter, realtime detection, IPP

I. Introduction

영상 기반의 자동화 시스템에서 가장 기본이 되는 단계는 대상 물체(또는 표적)에 대한 탐지이다. 임의의 영상 위치에 나타나는 대상을 찾아냄으로써, 이후 표적의 추적, 식별 등의 단계를 수행할 수 있다. 영상 처리 분야에서 표적 탐지는 민간 분야와 군사 분야 모두에서 가장 기본적인 단계로써 연구가 진행되어 왔다. 일반적인 표적 탐지는 특정 표적이 한정되어 있는 경우가 많다. 대표적으로 Viola 등[1]은 '얼굴'이라는 특정 표적에 대해 실시간 탐지를 수행하는 기법을 연구하였다. 가장 일반적이고 특징적일 수 있는 얼굴 탐지에 대한 연구는 영상 분야에서 오래도록 수행되어온 분야이다. 이외에도 Seo 등[2]은 영상의 혈관을 탐지하는 기법을 연구하였으며, Joo 등[3]은 손이라는 특정 표적에 대한 탐지 기법을 연구하였다. 본 논문에서 목표로 하는 시스템은 앞서의 탐지 분야와는 다른 목적을 가진

다. 임의의 대공 적외선 영상에 대해 임의의 위치에서 진입하는 저조도의 소형표적을 대상으로 한다. 따라서 표적의 형태 등의 상세 정보를 활용하기 어려우며, 표적의 크기 등에도 자유로워야 한다. 또, 공중 표적의 경우 대부분 고속 이동을 하기 때문에, 고속 시스템에 적합한 실시간 처리가 가능해야 한다. 이러한 관점에서 탐지 기법을 수립할 때 고려해야 할 사항은 아래와 같다.

- 지대공 영상 환경
- 군사용 적외선 영상
- 저조도 소형 표적
- 실시간 처리

• First Author: Chul Joong Kim, Corresponding Author: Jae Hyup Kim
*Chul Joong Kim (cj.stc.kim@hanwha.com), Electro-Optics Team, Hanwha Thales Co.
**Jae Hyup Kim (jaehyubious@naver.com), Data Tech. Team, SK Co.
***Kyung Hyun Jang (kyunghyun.jang@hanwha.com), Electro-Optics Team, Hanwha Thales Co.
• Received: 2016. 06. 20, Revised: 2016. 07. 14, Accepted: 2016. 08. 21.

대부분의 군사용 적외선 시스템에서는 주야간 활용이 가능한 적외선 영상을 기반으로 한다. 따라서 영상의 특성과 물리적 계수가 큰 차이가 발생하기 때문에 탐지의 관점에서 특정한 분야로 인식된다. Rajkumar 등[4]은 중간필터(median filter)와 엔트로피(entropy)를 이용한 배경 제거 기법을 제안하였으며, 주로 지상 영상에서의 사람, 자동차 등의 일정 수준의 크기를 보장하는 표적을 대상으로 하였다. Chen 등[5]은 웨이블릿(wavelet)을 이용한 주파수 영상에서 가우시안 형태의 소형 표적 모델을 설정하여 탐지하는 기법을 제안하였으나, 소형 표적의 크기가 크고, 표적의 형태 등이 해상 환경에 적합하다. Liu 등[6]은 지역(spatial)적인 처리가 아닌 시간(temporal)적인 관점에서 배경과 표적을 구분하는 탐지 기법을 제안하였으나, 클러터(clutter)에 취약하고 다수의 프레임 정보를 누적할 수 있는 시스템에 적용이 가능하다. 국내에서도 군사 분야에서 적외선 영상 기반의 표적 탐지 연구가 진행되어 왔다. Park 등[7]은 적외선 영상의 수평선 탐지 기법을 제안하였으며, 이는 수평선이라는 특정한 표적을 대상으로 하였다. Kim 등[8]은 적외선 영상의 움직임(motion) 정보를 이용하여 영상 정렬(registration)을 수행한 후 차영상을 통해 표적을 탐지하였으며, 지상 기동 표적을 대상으로 하였다. 앞서의 연구들과 본 논문의 목표가 다른 점은 저조도 소형 표적에 대한 실시간 처리 기법이다. 이러한 관점에서, Taotao 등[9]과 Ye[10] 등의 연구에 주목하였다. Taotao 등은 모폴로지(morphology) 기반의 필터를 3프레임에 적용하여 소형 표적 탐지를 수행하였으며, FPGA 기반의 실시간 H/W 처리구조를 제안하였다. Ye 등은 LDF(linear diffusion filter), 2DLMS(2 dimensional least mean square) 필터, 평균 이동 필터(mean shift filter) 등을 이용하여 소형 표적 탐지를 수행하였다. 지대공 환경에서의 적외선 영상, 그리고 원거리 진입 표적의 경우 일반 영상에 비해 배경의 복잡도는 낮아지지만 표적의 조도가 매우 떨어질 수 있으며, 크기가 수픽셀(pixel) 수준으로 작은 표적이 대부분이다. 따라서 표적의 특성 정보를 고려할 수 없기 때문에 앞서의 연구와 같이 영상에 대한 물리적인 필터를 적용해야 한다. 그러나 Taotao 등의 탐지시스템은 FPGA H/W 기반의 시스템에서 적용이 가능하기 때문에 본 논문의 목표가 되는 시스템과는 배치된다. Ye 등의 탐지 기법은 소형 표적에 대한 방향이 동일하지만, 배경 추정을 위한 다수의 필터들이 고속 처리가 어려운 단점이 있다.

본 논문에서는 목표 시스템에 적합하도록 물리적 필터를 결합한 탐지 기법과 함께 이를 S/W 적으로 실시간 처리가 가능하도록 하는 IPP(integrated performance primitive) 적용을 설명한다. 복잡도 높은 단일 필터가 아닌 직관적인 필터들의 다중 적용을 통해 저조도 소형 표적에 적합한 탐지 알고리즘을 설계하였다. 그리고, S/W 측면의 실시간 처리를 위해서는 H/W에 구애되지 않는 최적화 모듈이 필요하다. 인텔(intel)에서 제공하는 IPP 모듈은 다양한 수치연산에 대한 CPU 기반의 최적화 모듈들을 제공하기 때문에, 위의 물리적 필터를 고속처리할 수 있

게 해준다. 본 논문에서는 2장에서 제안하는 탐지 필터에 대해 설명하고, 3장에서는 탐지 필터의 IPP 기반 S/W 구현을 설명한다. 4장에서는 탐지와 수행 시간에 대한 실험 결과를 설명하고 5장에서 결론을 설명한다.

II. Small Target Detection

본 논문에서 표적이 잘 보이지 않는 환경(영상이 흐릿하면서 잡음 성분이 많음), 즉 저조도 표적의 탐지를 위한 알고리즘은 아래 그림 1과 같다.

저대역 통과 필터(LPF, low pass filter) 연산은 영상 자체의 잡음을 제거하는 효과가 있지만 동시에 표적 성분 또한 흐리게 하는 효과를 가지고 있다. 표적이 잘 안 보이는 영상에서 LPF를 사용하면 표적이 더 흐려지는 효과도 있지만, 표적 주변과 표적 사이의 잡음을 제거하면 원래의 표적 자체의 모양이 오히려 보존되는 효과를 볼 수 있다. 적절한 표적의 보존을 위해 5x5 마스크 연산을 사용하였고, 마스크 계수는 식 (1)과 같다. 그림 2는 LPF 처리 후의 결과 영상을 나타낸다. 흐린 날씨에서 접근하는 공중 표적의 예이다.

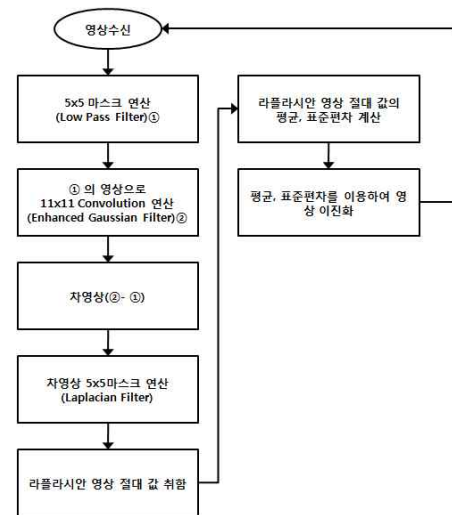


Fig. 1. The proposed detection process

$$LPF_1 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 25 & 25 & 25 & 25 & 25 \\ 1 & 1 & 1 & 1 & 1 \\ 25 & 25 & 25 & 25 & 25 \\ 1 & 1 & 1 & 1 & 1 \\ 25 & 25 & 25 & 25 & 25 \\ 1 & 1 & 1 & 1 & 1 \\ 25 & 25 & 25 & 25 & 25 \\ 1 & 1 & 1 & 1 & 1 \\ 25 & 25 & 25 & 25 & 25 \end{bmatrix} \quad (1)$$

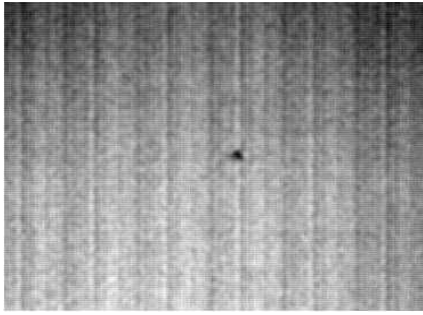


Fig. 2. The result of LPF

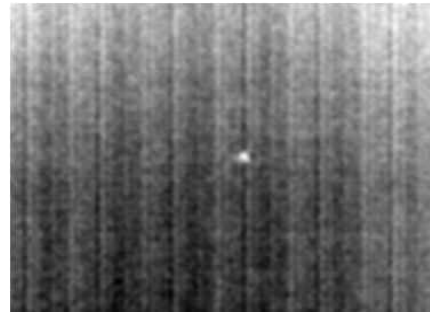


Fig. 5 The result of difference

LPF 영상에 대하여 가우시안 필터(GF, gaussian filter)를 이용하여 컨볼루션(convolution) 연산을 수행한다. 필터의 크기는 실험적으로 11x11로 설계하였으며, LPF 영상에서 GF 연산을 수행하면 영상의 타겟이 더 흐려지지만 타겟 사이 및 주변의 노이즈를 추가로 제거 해 주는 효과를 볼 수 있기 때문에 결론적으로 소형 타겟 크기를 좀 더 확장 시키는 효과가 있다. 그림 3은 11x11 GF의 형태를 나타내며, 그림 4는 GF 처리 후의 결과 영상을 나타낸다.

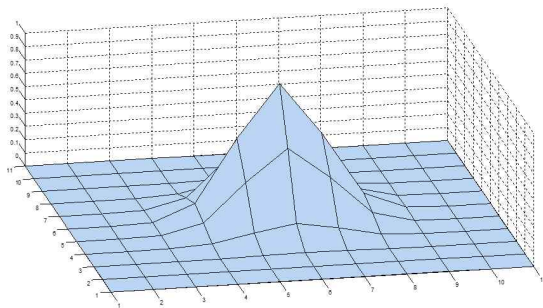


Fig. 3. The shape of GF

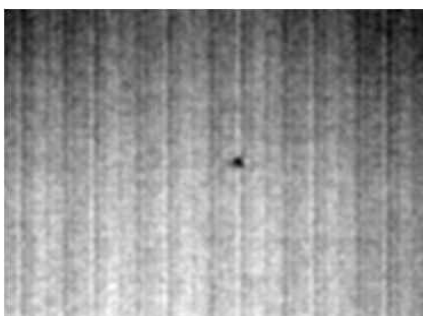


Fig. 4. The result of GF

다음 단계는 배경 및 노이즈가 감소된 영상을 대상으로 표적의 에지(edge)를 검출하기 위해 5x5 라플라시안 필터(LF, laplacian filter)를 수행한다. LF 마스크 계수는 식 (2)와 같다. 그림 6은 그림 5에 대한 LF 결과를 나타낸다.

$$LF_1 = \begin{bmatrix} -1 & -3 & -4 & -3 & -1 \\ -3 & 0 & 6 & 0 & 3 \\ -4 & 6 & 20 & 6 & -4 \\ -3 & 0 & 6 & 0 & -3 \\ -1 & -3 & -4 & -3 & -1 \end{bmatrix} \quad (2)$$

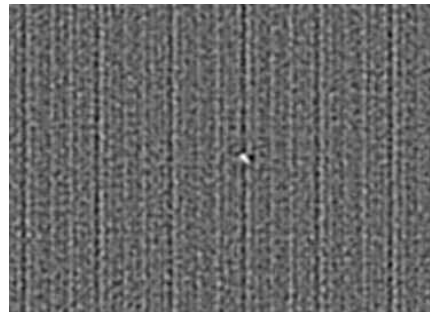


Fig. 6. The result of LF

이때 LF 결과값에서 음수가 발생할 수 있는데, 모두 절대값으로 전환한다. 이는 배경의 성분을 좀 더 제거 해 주고 표적의 에지 성분을 강화 시켜주는 효과가 있다. 또한 영상의 이진 화를 위해 평균 및 표준편차 값이 거의 일정하게 나오는 효과가 있어 이진 화 기법에 용이하게 적용 할 수 있다. 그림 7은 절대 값 영상을 나타낸다.

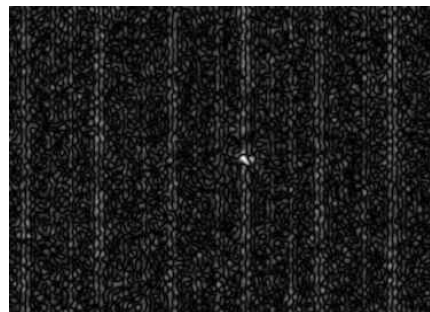


Fig. 7. The result of absolute-image

다음으로 앞서의 LPF 영상과 GF 영상의 차를 구한다. 영역 기반 필터에서 소형 표적 검출에 가장 흔히 쓰이는 기법인 평균제거필터(mean subtraction filter)의 경우 점 노이즈(spot noise)의 검출이 단점으로 지적되므로, 이를 보완하기 위하여, LPF 결과 영상과 GF 결과 영상의 차를 구하였다. 이는 배경의 잡음을 한 번 더 상쇄하는 효과를 보면서 동시에 타겟의 검출을 용이하게 만드는 효과가 있다. 그림 5 LPF와 GF의 차영상을 나타낸다.

마지막으로 영상에 대하여 평균과 표준편차를 계산하여 이진화를 진행한다. 이진화를 위한 임계값은 아래의 식 (3)을 사용한다. 식에서 f 는 이미지를 의미하며, x 는 픽셀 위치를 의미한다. μ 와 σ 는 각각 평균과 표준편차를 의미하며, α 는 사용자 계수이다. 제안하는 기법에서는 실험적으로 6으로 설정하였다. 그림8은 그림 7에 대해 이진화를 수행한 최종 결과 영상을 나타낸다.

$$THR_{img} = \begin{cases} 1 & \text{if } f(x) \geq \mu_f + \alpha \cdot \sigma_f \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

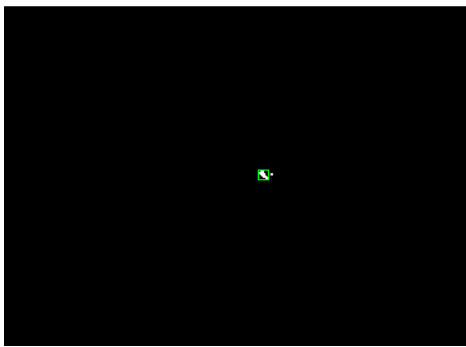


Fig. 8. The final result of detection

III. Intergrated Performance Primitives

인텔의 IPP는 S/W 성능 향상을 위해 멀티미디어, 기업 데이터 관리, 임베디드, 통신 그리고 과학/기술 어플리케이션 분야에 폭넓게 사용되고 있다. IPP의 ‘Primitive’는 기본적으로 사용하는 수천 개의 최적화된 알고리즘을 사용할 수 있게끔 인터페이스를 제공 해 주는 것을 의미한다. 따라서 IPP의 ‘Primitive’를 사용한다는 것은 어플리케이션의 최적화를 하지 않아도 자동으로 어플리케이션의 최적화를 가능하게 해 준다는 것을 의미한다. IPP 라이브러리는 인텔 CPU가 내장된 환경에서 사전에 빌드된 라이브러리 함수를 제공하며, 신호처리 및 영상처리의 높은 퍼포먼스를 제공한다. 즉, 개발자가 기능적인 구현에 초점을 맞추어 개발 할 수 있는 환경을 구성 해 준다는 것을 의미한다[11][12]. IPP의 대표적인 특징은 아래와 같다.

▪ 최적화된 성능 제공

IPP는 사용자가 인텔 프로세서의 성능을 극대화하기 위해 고도로 최적화 된 응용 프로그램을 작성할 수 있도록 프로그래밍 인터페이스를 제공하는 크로스 플랫폼 소프트웨어 라이브러리이다. 사용자는 프로세서 고유의 코드를 작성하지 않고도 고급 프로세서 기능에 접근이 가능하다. 이는 어플리케이션의 최적화를 용이하게 해 주고, 개발과 유지보수 시간을 절약 할 수 있으며, 빠른 연산시간을 수행 할 수 있게 해 준다.

▪ 호환성 및 확장성

IPP로 작성된 어플리케이션은 인텔 아키텍처 기반이면 코드 수정 없이 사용 가능하며, 코드 확장 또한 가능하다.

▪ 무상 재배포

IPP는 사용자가 설계한 어플리케이션에 있는 런타임 라이브러리의 복사본을 제한 없이 배포가 가능하고, 이는 장기적인 관점으로 보았을 때 비용 절감의 효과를 볼 수 있다.

2장에서 설명한 제안하는 탐지기법의 경우, 물리적 필터들의 연속이다. 이는 IPP기반의 S/W 최적화를 적극 활용하기 위한 것이다. 우선 LPF 의 경우 IPP 기반의 구현과 일반적인 C기반의 구현을 비교하면, 표 1과 같다. 중복 반복문을 통한 순차적 계산을 하는 C 기반 구현과는 달리 제공되는 저레벨 함수를 2 단계 적용하여 필터를 수행할 수 있다. IPP에서 제공되는 저대역 통과 필터가 실수데이터만을 대상으로 하기 때문에, 영상 데이터를 실수 데이터로 변환하는 단계가 추가된 것이다.

표 2는 GF 구현 비교를 나타낸다. 표 1의 LPF의 결과가 실수 데이터이므로, 그대로 1단계의 IPP함수로 구현이 종료된다. 표 3-7은 각각 두 결과의 차를 구하는 단계, LF, 절댓값, 문턱치(threshold)계산, 이진화(THR) 단계의 구현 비교를 나타낸다.

Table 1. The LPF pseudo-code of IPP and C

IPP	ippiConvert_16u32f_C1R(Src+Offset, ipp16ulmgStep, tmp_Dst+Offset, ipp32flmgStep, Size) ippiFilterLowpass_32f_C1R(tmp_Dst+Offset_5x5, ipp32flmgStep, Dst+Offset_5x5, ipp32flmgStep, Size_5x5, ippMskSize5x5)
C	for(i=...) for(j=...) { Dst[i*WIDTH+j] = (Src[(i-2)*WIDTH+j-2]*0.04f+ +Src[(i+2)*WIDTH+j+2] * 0.04f); }

Table 2. The GF pseudo-code of IPP and C

IPP	ippiConvFull_32f_C1R(Src+Offset_5x5, ipp32flmgStep, Size_5x5, ipp32f_krenel, 11*sizeof(float), kernelSize, Dst+Offset2_5x5, ipp32flmgStep);
C	for(i=...) for(j=...) { Dst[i*WIDTH+j] = (Src[(i-4)*WIDTH+j-1] * 0.0002f +... +Src[(i+4)*WIDTH+j+1] * 0.0002f); }

Table 3. The difference pseudo-code of IPP and C

IPP	<code>ippiSub_32f_C1R(Src1 + offset, ipp32fImgStep, Src2 + offset, ipp32fImgStep, Dst + offset, ipp32fImgStep, size);</code>
C	<pre>for(i=...) for(j=...) { Dst[i*WIDTH+j] = Src1[i*WIDTH+j] - Src2[i*WIDTH+j]; }</pre>

Table 4. The LF pseudo-code of IPP and C

IPP	<code>ippiFilterLaplace_32f_C1R(Src+Offset_5x5, ipp32fImgStep, Dst+Offset_5x5, ipp32fImgStep, Size_5x5, ippMskSize5x5);</code>
C	<pre>for(i=...) for(j=...) { Dst[i*WIDTH+j] = (Src[(i-2)*WIDTH+j-2] * (-1.0f) + ... + Src[(i+2)*WIDTH+j+2] * (-1.0f)); }</pre>

Table 5. The ABS pseudo-code of IPP and C

IPP	<code>ippiAbs_32f_C1R(Src+Offset, ipp32fImgStep, Dst+Offset, ipp32fImgStep, Size);</code>
C	<pre>for(i=...) for(j=...) { Dst[i*WIDTH+j] = fabs(Src[i*WIDTH+j]); }</pre>

Table 6. The threshold pseudo-code of IPP and C

IPP	<code>ippiMean_StdDev_32f_C1R(Src + offset, ipp32fImgStep, ipp_ROI, &Mean, &Std);</code>
C	<pre>tmp = 0.0f; tmp_std = 0.0f; for(i=...) for(j=...) { tmp += Dst[i*WIDTH+j]; } Mean = tmp/width*height; tmp = 0.0f; for(i=...) for(j=...) { tmp += (Dst[i*WIDTH+j]-Mean)* (Dst[i*WIDTH+j]-Mean); } tmp_std = tmp/width*height; Std = sqrt(tmp_std);</pre>

Table 7. The binarization pseudo-code of IPP and C

IPP	<code>ippiCompareC_32f_C1R(Src+offset, ipp32fImgStep, Threshold, Dst+offset, ipp8ulmgStep, ipp_ROI, ippCmpGreater);</code>
C	<pre>for(i=...) for(j=...) { if(Src[i*WIDTH+j] >= Threshold) Dst[i*WIDTH+j] = 255; else Dst[i*WIDTH+j] = 0; }</pre>

IV. Experimental Results

실험과 분석을 위하여 모든 모듈을 시뮬레이터로 구현하였으며, Windows 7 환경에서 IPP 라이브러리 사용이 가능한 Visual Studio 2008로 구현 하였다. 영상은 무인기 비행 영상을 사용하였고 표적간 거리는 5km, 해상도는 640x480 이다. 탐지 알고리즘의 적용은 영상 중심을 기준으로 400x300으로 지정하였으며, 고속 처리를 위한 IPP라이브러리는 7.0을 사용 하였다. 그림 9는 시뮬레이터 구현 예를 나타낸다.

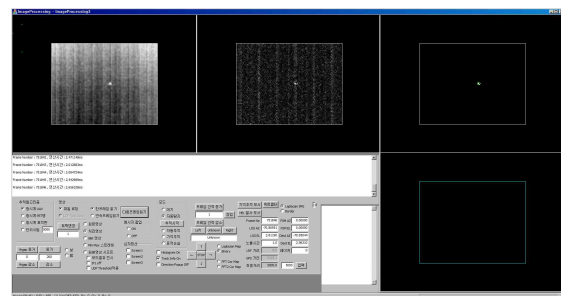


Fig. 9. Simulator

1. 탐지 성능 분석

본 논문에서는 목표가 되는 탐지 시스템에 가장 유사하다고 판단되는 Liu 등[6]의 기법과 Ye 등[10]의 기법을 선택하여 제안하는 탐지 기법과 실험하였다. 실험 영상은 목표가 되는 대공 IR 영상이며, 지대공, 공대공, 해대공 등의 환경에 대하여 315프레임의 영상을 활용하였다. 그림 10은 지대공 영상에 대한 세 탐지 기법의 실행 예를 나타낸다.



(a)



(b)



(c)



(d)

Fig. 10. The examples of detection methods, (a) original image (b) Ye's method (c) Liu's method (d) proposed method

전체 영상에 대한 수행 결과, 탐지율에서는 Liu 등의 기법이 가장 우수했으며, 오경보율, 즉 오탐지의 관점에서는 제안하는 기법이 가장 우수하였다. Liu 등의 기법은 클러스터가 상대적으로 많이 탐지 되었으며, Ye 등의 기법과 제안하는 기법은 유사한 오경보율을 나타내었다. 세 기법의 탐지 결과를 정량화 하기 위하여, 탐지 결과를 리콜(recall)과 프리시즌(precision)으로 표현하였다. 리콜은 실제 표적 중 탐지에 성공한 비율을 나타내며, 프리시즌은 탐지된 개체 중 실제 표적의 비율을 나타낸다. 그림 11은 세 기법에 대한 리콜과 프리시즌을 나타낸다.

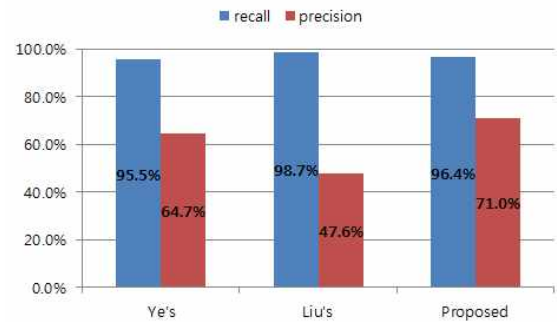


Fig. 11. The detection performance

앞서 언급한바와 같이, 리콜의 측면에서는 Liu 등의 기법이 가장 우수한 결과를 도출하였으나, 프리시즌의 측면에서는 제안하는 기법이 가장 우수한 결과를 도출하였다. 리콜과 프리시즌을 동시에 고려해야 하는 탐지 시스템의 입장에서는 Liu 등의 기법에 비해 Ye 등의 기법과 제안하는 기법이 더 적합하다고 볼 수 있다.

2. 수행 시간 분석

보유한 3155프레임에 대해 세 기법을 수행하여 평균 수행 시간을 비교하였다. 탐지 개체의 개수에 따라 연산량이 큰 차이를 보일 수 있기 때문에, 수행 시간은 전체 탐지 개체에 대한 프레임 별 평균 수행 시간, 탐지 개체 1개당 수행 시간으로 표현하였다. 세 기법을 구현하여 실험하는데 있어 S/W 역량의 개입을 제거하기 위하여, 모두 동일한 기준으로 C 기반 코드를 작성하여 실험하였다. 그림 12는 수행시간 비교 결과를 나타낸다. 단위는 ms(1/1000초) 이다.

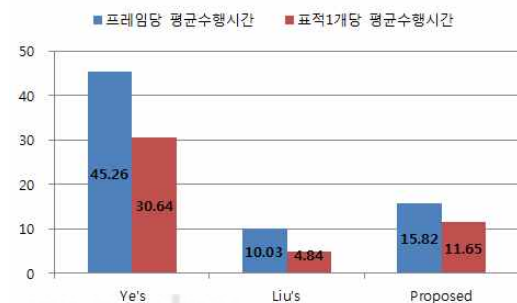


Fig. 12. The comparison of detection processing time

수행 시간 측면에서는 Liu 등의 기법이 가장 빠르게 나타났으며, 낮은 프리시즌에 비해 Ye 등의 기법은 매우 많은 수행시간을 확인할 수 있다.

본 논문에서 설명한 IPP 기반의 코드 최적화를 분석하기 위하여, 제안하는 기법의 필터 연산의 단계별 수행 시간을 측정하였다. 표 8은 IPP기반의 구현과 C기반의 구현에 대한 평균 수행 시간을 나타낸다. 전체 프레임에 대한 평균 수행 시간을 의미하며, 단위는 ms(1/000초) 이며 (평균 ± 표준편차)의 형식으로 표기하였다.

Table 8. The comparison of processing time in modules

	IPP	C
LPF	0.25 ± 0.17	4.78 ± 0.10
GF	0.65 ± 0.31	6.20 ± 0.11
DIFF	0.09 ± 0.02	0.29 ± 0.05
LF	0.29 ± 0.03	2.11 ± 0.06
ABS	0.15 ± 0.04	1.54 ± 0.06
THR	0.04 ± 0.00	0.66 ± 0.02
BIN	0.04 ± 0.00	0.25 ± 0.01

표기의 편의를 위하여 탐지 알고리즘 각각의 모듈을 약자로 표기하였다. 전체적으로 IPP를 적용했을 때 수행시간이 1/10으로 줄어들음을 확인할 수 있다. 모든 모듈의 수행시간 평균을 합했을 경우, IPP 기반 구현은 1.54ms, C 기반 구현은 15.82ms를 나타내어 IPP 기반 구현이 C 기반 구현 대비 9.7% 정도의 수행시간을 나타내고 있다. 표 9는 모듈별 수행시간의 비율을 나타내고 있다.

Table 9. The ratio of processing time in modules

	IPP time / C time
LPF	5.9%
GF	10.5%
DIFF	32.1%
LF	13.8%
ABS	9.5%
THR	5.4%
BIN	15.0%

LPF 단계와 BIN 단계는 1/20 수준으로 단축됨을 확인할 수 있다. DIFF 단계의 경우, C 기반에서도 연산이 매우 단순하여 빠른 속도로 처리되기 때문에 IPP 대비율이 높게 나타났다. 그림 9는 앞서 언급한 수행시간 데이터를 나타낸다. 총 수행 시간은 전체 모듈의 수행을 모두 합한 값이며, 각각의 프레임에 대한 시간 측정 결과를 정리하여 나타낸다.

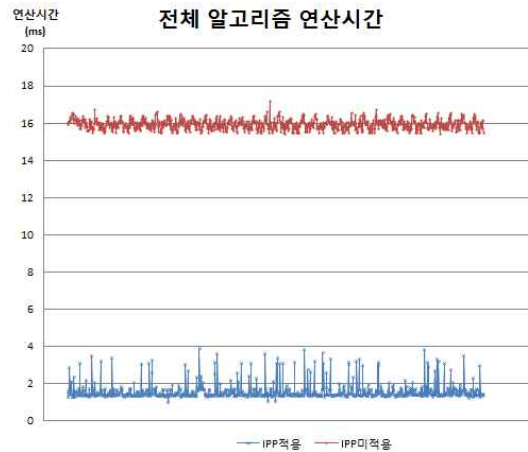


Fig. 9. The processing time per frames

V. Conclusions

본 논문에서는 다중 지역 필터를 이용한 저조도 소형 표적 탐지 기법과 IPP 기반의 실시간 처리 S/W 구축에 대해 설명하였다. IPP 기반의 구현을 통해 다중의 지역 필터를 실시간에 처리할 수있음을 확인하였으며, 90% 이상의 탐지 성능을 확인하였다. 마스크 필터 연산이 다량으로 들어간 S/W를 순차적인 C 기반으로 구현하면 최대 수백 ms 까지 나오지만 IPP기반의 구현을 적용하여 S/W 구현 및 실험 결과 연산시간이 대폭 줄어든 것을 확인 할 수 있었다. 다만, 서론에서 언급한바와 같이, 영상기반의 시스템에서는 탐지가 초기 표적 정보 획득의 시작 단계이다. 따라서 이후의 단계(예를 들어, 표적 추적)와의 연계된 연구가 필요하다. 단순한 물리적 필터기반의 경우 탐지 좌표 이외의 추가 정보를 획득할 수 없기 때문에, 후단계에서 활용가능한 정보가 거의 없다. 이는 탐지 모듈 단계에서만 고려할 문제가 아닌, 전체 시스템의 입장에서 고려되어야할 사항이다. 또, 물리적 필터 기반의 모듈들은 IPP 등의 실시간 S/W로 뛰어난 실시간 처리가 가능하지만, 배경과 표적의 변화에 매우 민감할 수밖에 없다. 시스템이 허용하는 한도 내에서, 변화에 강건한 추가적인 처리가 필요할 것으로 판단된다.

REFERENCES

[1] P. Viola and M. J. Jones, "Robust Real-time Face Detection," International Journal of Computer Vision, Vol.57, No.2, pp.137-154, March 2004.
 [2] J. J. Seo and J. W. Park, "Hepatic Vessel Segmentation using Edge Detection," Journal of the Korea Society of Computer and Information(KSCDI), Vol.17, No.3, pp.51-57,

March 2012.

- [3] S. I. Joo, S. H. Weon and H. I. Choi, "Hand Region Tracking and Fingertip Detection Based on Depth Image," Journal of the Korea Society of Computer and Information(KSCDI), Vol.18, No.8, pp.65-75, August 2013.
- [4] S. Rajkumar and P.V.S.S.R. Chandra Mouli, "Target Detection in Infrared Images Using Block-Based Approach," Proceeding of ICICTs 2014, Vol.4, pp.1-16, 2014.
- [5] H. Chen, H. Zhang, Y. Yang and D. Yuan, "Small Target detection Based on Infrared Image Adaptive," International Journal on Smart Sensing and Intelligent Systems, Vol.8, No.1, pp.497-515, March 2015.
- [6] D. Liu, J. Zhang and W. Dong, "Temporal Profile Based Small Moving Target Detection Algorithm in Infrared Image Sequences," International Journal of Infrared Military Waves, Vol.28, pp.373-391, 2007.
- [7] B. S. Park and J. H. Kim, "The horizontal line detection method using haar-like features and linear regression in infrared images," Journal of the Korea Society of Computer and Information(KSCDI), Vol.20, No.12, pp.29-36, December 2015.
- [8] J. H. Kim, B. J. Choi, S. W. Chun, J. M. Lee and Y. S. Moon, "The target detection and classification method using SURF feature points and image displacement in infrared images," Journal of the Korea Society of Computer and Information(KSCDI), Vol.19, No.11, pp.43-52, November 2014.
- [9] H. Taotao, L. Wang, J. Wang and H. Sheng, "Infrared Small Target Detecting Based on Parallel Streaming Pipeline Architecture," Proceeding of CMES 2015, pp.346-349, November 2015.
- [10] S. Y. Ye, J. H. Joo and K. G. Nam, "Small Target Detecting and Tracking Using Mean Shifter Guided Kalman Filter," Transaction on Electrical and Electronic Materials, Vol.14, No.4, pp.187-192, June 2013.
- [11] <https://software.intel.com/en-us/intel-ipp/>

Authors



Chuljoong Kim received the B.S. degree in Biomedical Engineering & Electrical Engineering from KyungHee University, Korea, in 2010.

In 2009, he joined Hanwha-Thales Co., Republic of Korea, and he is currently a researcher. He is interested in machine vision and software architecture.



Jae Hyup Kim received the B.S., M.S. and Ph.D. degrees in computer science and engineering from Hanyang University, Korea, in 2001, 2003 and 2008, respectively

From 2008 to 2009, Dr. Kim had been a post Doc. researcher at the Ambient Intelligence S/W Research Institute in Hanyang University. From 2009 to 2016, he had been a senior researcher at the image sensor team in Samsung-Thales and Hwnwha-Thales. In 2016, he joined the SK Co., Korea, as chief researcher, and is currently a researcher. He is interested in machine learning, artificial intelligence, and computer vision.



Kyung Hyun Jang received the B.S., M.S. and Ph.D. degrees in Computer Science and Engineering from Hanyang University, Korea, in 2005, 2007 and 2014, respectively

In 2014, he joined the Hanwha-Thales Co., Republic of Korea, and is currently a senior researcher. He is interested in computer vision and pattern recognition.