실시간 IR 소형표적 탐지를 위한 라인버퍼·팽창(dilation) 고속화 설계

김정석*, 김대연
*국방과학연구소
*bluesky@add.re.kr, daeyeon@add.re.kr

Line-Buffering and Dilation Acceleration for Real-Time Infrared Small-Target Detection

Kim Jeong Seok*, Kim Dae Yeon
Agency for Defense Development

요 약

적외선(IR) 검출기는 일반적으로 프레임을 한 줄(line)씩 신호처리 보드로 순차 전송하므로, 프레임 단위 영상 전처리를 CPU에서 일괄 수행 할 경우 탐지 시작 시점이 지연되는 문제가 있다. 본 연구는 신호처리 보드의 FPGA가 입력 라인을 라인 버퍼에 저장하면서, 구조요소 반경만큼의 줄이 확보되는 즉시 형태학적 팽창(dilate)을 스트리밍 방식으로 수행하여 dilate 완료 프레임을 생성하고, 이를 곧바로 CPU로 전달해 CPU가 프레임 수신 직후 탐지 알고리즘을 실행하도록 하는 저지연 파이프라인을 제안한다.

제안 방식의 FPGA에서 dilate는 윈도우 의존 연산이지만, 최소 라인버퍼 깊이 $B = 2r + 1$을 갖는 간단한 구조로 실현되며, 원본−dilate 잔차 기반으로 일관되게 처리된다. CPU 단계는 잔차 영상을 기반으로 탐지를 수행한다. 이때 전체 지연은 “프레임 수신 완료+ 꼬리 r줄 처리”만큼으로 상한이 정해지며, 소프트웨어 전용 경로에서 요구되던 CPU 측 dilate, 잔차영상 생성 시간만큼의 지연을 제거한다. 제안 방식이 라인 스캔 기반 IR 시스템에서 하드웨어 자원, 구현 복잡도를 크게 증가시키지 않으면서 탐지 개시 지연을 유의미하게 단축함을 보인다.

Ⅰ. 서 론

적외선(IR) 소형표적 탐지는 낮은 SNR과 강한 배경 클러터(구름 경계, 해면 패턴, 지형 텍스처, 센서 잡음 등) 환경에서 신뢰성 있게 표적을 부각해야 한다는 점에서 본질적으로 까다롭다. 이 때문에 전처리가 얼마나 배경 에너지를 억제하고 표적의 국소 대비를 보존하느냐가 이후 단계의 성패를 좌우한다. 최근에는 딥러닝 기반 방법도 활발하지만, 온보드/임베디드 환경에선 지연(latency), 전력, 메모리 제약 때문에 경량 신호처리 기반 전처리가 여전히 핵심 역할을 한다[1],[2].

많은 IR 검출기는 라인 스캔(line-scan)방식으로 프레임을 한 줄(line)씩 신호처리기로 전송한다. 전통적 구성에서는 CPU가 프레임 전체 수신을 기다린 뒤 형태학 연산(예: dilate/erode)이나 대형 커널 전처리를 일괄 수행하므로, (i) 프레임 대기 시간과 (ii) CPU 전처리 시간이 누적되어 탐지 개시 시점이 지연된다.

본 논문은 이러한 병목을 해소하기 위해, 신호처리기 보드의 FPGA가 라인 수신 중 라인버퍼(깊이 $B = 2r + 1$)로 윈도우를 완성하는 즉시 형태학적 팽창(dilation)을 스트리밍으로 계산하여 dilate 완료 프레임 D을 구성하고, 라인별 잔차 연산(예: $S = I - D$) 을 수행하여 이를 CPU로 전달하는 저지연 파이프라인을 제안한다[4]. 이렇게 하면 탐지 시작 시각이 $H \cdot T_L + T_{dil}$에서 $H \cdot T_L + r \cdot T_L$로 앞당겨지며, 특히 T_{dil}이 ms 단위로 큰 시스템에서 프레임당 수 ms 수준의 지연 절감을 기대할 수 있다. 구현은 경계 복제(Replicate)와 희소 구조요소(SE; 상·하·좌·우·모서리 8점)로 단순화하여 비교 트리의 폭과 BRAM 사용량을 줄이고, 라인-당 1출력 속도를 달성한다[3]. 형태학 연산 자체의 계산 복잡도는 SE 크기에 무관한 O(1)/픽셀 비교 횟수로 낮출 수 있다. 그림1은 본 프레임워크를 통해 얻은 잔차 영상의 결과이다.

다만 본 논문의 기여 초점은 알고리즘 SOTA 갱신이 아니라, 라인 스캔 입출력의 시간 구조에 맞춘 하드웨어 - 소프트웨어 설계로 프레임-도착 즉시 탐지를 가능하게 하는 시스템 지연 상한의 근본적 단축에 있다.

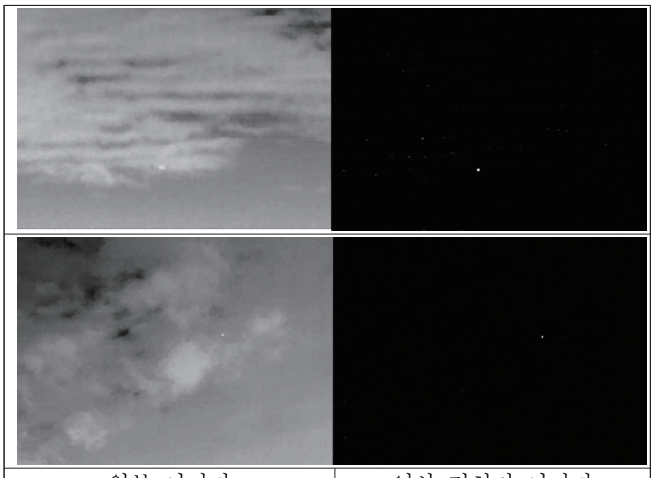

그림 1 영상 전처리를 적용한 결과 비교

Ⅱ. Framework 설계

본 프레임워크는 그림 2와 같이 구성된다. 프레임 해상도는 $H \times W = 256 \times 256$을 가정하되, 제안 프레임워크는 해상도에 무관하게 동작한다. 입력 단계에서 검출기는 1라인씩 스트리밍하며, FPGA는 이를 라인버퍼에 순차 저장한다. SE 반경을 r라 하면(예: $r = 3$이면 7×7 윈도우), dilate는 해당 라인의 상·하 r줄과 좌·우 r픽셀 범위의 최대 연산으로 정의된다. 스트리밍 구현에서는 최소 라인버퍼 깊이 $B = 2r + 1$로 창(window)을 완성할 수 있으며, 입력 라인 인덱스를 i라 할 때 $i \geq 2r$이면 중앙 라인 $i - r$에 대해 dilate 출력을 계산할 수

있다. 이렇게 하면 첫 유효 출력은 $2r$줄이 수신된 직후 발생하며, 이후에는 "한 줄 입력 → 한 줄 출력"의 파이프라인이 형성된다. 경계 처리는 Replicate(인덱스 클램프) 정책으로 구현하여 이미지 가장자리에서도 일관된 결과를 보장한다. 또한 잔차 영상도 FPGA에서 실시간으로 만들도록 하였다. 원본 라인 I와 dilate 라인 D의 차를 계산하여 잔차 라인 맵 $S = I - D$를 얻는다.

FPGA가 프레임 끝까지 스트리밍 dilate를 수행하면[4], 마지막 줄 수신 이후 꼬리 r줄의 출력 라인만 마무리하면 된다. 결과적으로 FPGA는 완성된 dilate 프레임 $D \in R^{H \times W}$과 잔차 맵 S를 산출하여 DMA 등으로 CPU에 전달한다.

이어서 정규화와 임계화를 수행하고, 필요시 형태학적 잡음 억제나 연결요소 분석으로 후보를 정제한다. 그 후 CPU에서 해당 잔차맵을 이용하여 탐지 알고리즘을 수행한다. 또한 본 설계는 딥러닝 기반이나 추가 전처리와도 쉽게 결합되며, S를 후보 맵 또는 가중 맵으로 활용할 수 있다.

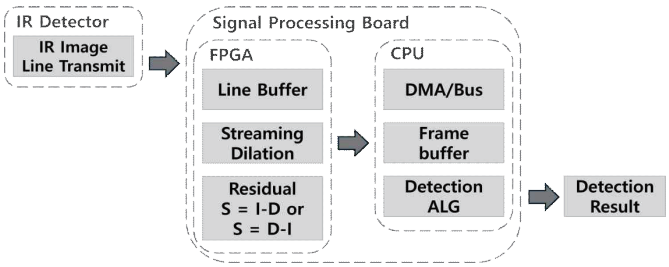

그림 2 소형표적 탐지를 위한 영상 전처리 프레임워크

Ⅲ. 실험

본 실험은 그림 2의 파이프라인-검출기의 라인 스트리밍을 받으면서 FPGA가 라인버퍼 $B = 2r + 1$로 창을 완성하는 즉시 스트리밍 dilation을 수행해 프레임 D를 만들고, CPU가 원본 I와 D로 잔차 $S = I - D$ 기반 탐지를 실행-이 소프트웨어 전용 경로에 비해 탐지 개시 지연을 얼마나 줄이는지를 정량화하는 데 목적이 있다. 해상도는 256×256, 비트심도는 14bit로 고정했고, 라인레이트 f_L는 20 kHz와 40 kHz, 구조요소 반경 r은 3(7×7)과 9(19×19)로 바꿔가며 평가했다. 경계는 Replicate(클램프) 정책, SE는 상·하·좌·우와 모서리의 8점 희소 마스크를 사용했다.

지연 모델은 단순하다. 라인 주기 $T_L = 1/f_L$일 때, 소프트웨어 전용 경로의 탐지 시작 시점은 프레임 전체 수신 후 CPU가 dilate를 끝내야 하므로 $T_{start,SW} \approx HT_L + T_{dil,CPU}$로 근사된다. 제안 방식은 FPGA가 수신 중에 dilate를 끝내므로 CPU는 프레임 수신과 거의 동시에(정확히는 꼬리 r라인 산출 직후) 탐지를 시작하게 되어 $T_{start,\mathrm{Prop}} \approx HT_L + rT_L$가 된다. 따라서 절감량은 $\Delta T \approx T_{dil,CPU} - rT_L$로 표현된다[5].

예를 들어 $W = 256, H = 256, r = 3(7\times7), f_L = 20kHz$이면 $T_L = 50us$이고 꼬리 시간 $rT_L = 150us$에 불과해, CPU에서 dilate가 수 ms만 걸려도 프레임당 수 ms 수준의 지연을 바로 줄일 수 있다. $r = 9(19\times19)$에서도 $rT_L = 450us$로 여전히 작게 유지된다.

측정은 동일 입력 프레임에 대해 두 경로를 각각 실행하며 진행했다. dilate 구간의 CPU 시간 $T_{dil,CPU}$는 고해상도 타이머로 직접 계측했고, 프레임 수신 완료, FPGA-DMA 완료, CPU-탐지 시작 시각을 타임스탬프로 기록하여 $T_{start,SW}$와 $T_{start,\mathrm{Prop}}$를 산출했다. 각 설정당 200프레임 이상 반복해 평균과 표준편차를 확인했으며, FPGA가 산출한 D와 소프트웨어 기준 D_{SW}의 픽셀 불일치율도 함께 계산해 99.99% 이상(불일치율 $<10^{-4}$) 일치를 목표로 검증했다.

결과는 표 1에 요약했다. Table 1은 각 (f_L, r) 조합에서 T_L, HT_L, rT_L, $T_{dil,CPU}$, 두 경로의 탐지 시작 시각, 그리고 제안 방식을 통해 단축된 지연 시간 ΔT(ms)를 정리해 이론 모델과 실측이 잘 일치함을 보여준다(일반적으로 $T_{dil,CPU} \gg rT_L$이므로 $\Delta T > 0$).

f_L (kHz)	r	TL (us)	rTL (us)	$T_{start,SW}$ (ms)	$T_{start,\mathrm{Prop}}$ (ms)	ΔT (ms)
20	3	50	12.8	14.62	12.95	**1.67**
20	9	50	12.8	17.35	13.25	**4.1**
40	3	25	6.4	1.82	6.48	**1.74**
40	9	25	6.4	4.55	6.63	**4.32**

표 1 탐지까지 단축된 지연 시간 측정 결과

제안 방식이 의도대로 탐지 시작 시점을 앞당겼음을 확인하였다.

Ⅳ. 결론

본 논문은 라인 스캔 기반 IR 영상에서 FPGA 형태학 선처리(dilate) 스트리밍을 통해 dilate 완료 프레임을 생성·전달하고, CPU가 프레임 수신 직후 원본-dilate를 한 차분영상에 탐지를 수행하도록 하는 저지연 파이프라인을 제안하였다. 최소 라인버퍼 $B = 2r + 1$과 경계 복제 정책으로 구현 난이도와 자원 소모를 억제하면서, CPU 전처리 시간을 프레임 수신 동안에 완전히 은닉하여 탐지 개시 지연을 실질적으로 제거한다. 제안 방식은 기존 소프트웨어 전용 경로 대비 탐지 시작 시점을 1.67 ~ 4.32ms까지 앞당기며, 다양한 SE 크기·형태에도 쉽게 확장 가능하다. 향후에는 딥러닝 모델의 입력 이미지에 적용할 영상 전처리로 활용하여 테스트 할 예정이다.

ACKNOWLEDGMENT

This work was supported by the Agency for Defense Development Grant Funded by the Korean Government(912A16701).

참 고 문 헌

[1] R. Kou, J. Liu, and X. Sun, "Infrared small target segmentation networks: A survey," Pattern Recognition, 2023, vol. 143 pp. 109788.

[2] Y. Cheng, Z. Zhang, and G. Wang, "Infrared dim small target detection networks: A review," Sensors, 2024, vol. 24, no. 12, pp. 3885.

[3] S. Moradi, M. Oskouei, and H. Rabiee, "Fast and robust small infrared target detection using ADMD," Signal Processing, 2020, vol. 177, 107727, pp. 1 - 12.

[4] M. van Herk, "A fast algorithm for local minimum and maximum filters on rectangular and octagonal kernels," Pattern Recognition Letters, 1992, vol. 13, no. 7, pp. 517 - 521.

[5] H. Usamentiaga, D. F. Garcia, J. L. Rendueles, and L. Molleda, "Real-time adaptive method for noise filtering of a stream of data from a line-scan camera," Journal of Electronic Imaging, 2008, vol. 17, no. 3, 033012.