

다채널 동작이 가능한 단일-스캔 연결 성분 라벨러

정재원*, 신윤철*, 박주현*, 이명진*

한국항공대학교 정보통신공학과

mrjeongjw@gmail.com

Multi-Channel Single-Pass Connected Components Labeler

Jae-won Jeong*, Youn-chul Shin*, Joo-heon Park* and Myeong-jin Lee*

Department of Telecommunication and Information Engineering, Korea Aerospace University

요약

기존의 영상 라벨러는 이중 스캔 연결 성분 라벨링 방식이 주로 사용되어 왔다. 그러나, 이중 스캔 연결 성분 라벨링 방식은 별도의 라벨 병합 구간 때문에 실시간 처리와 다채널 처리가 어렵다. 최근 단일 스캔 연결 성분 라벨링 방법들이 제안되었지만, 이들 역시 스캔 횟수는 줄일 수 있으나 별도의 라벨 병합 구간을 제거하지는 못하였다. 본 논문에서는 별도의 라벨 병합 구간을 갖지 않는 단일 스캔 연결 성분 라벨러 구조를 제안한다. 제안하는 구조는 라인 스캔과 동시에 라벨 병합이 이루어지기 때문에 고속 동작하고, 예상 가능한 프레임 처리율을 가지며, 다채널 처리가 가능하다. 제안하는 라벨러 구조는 FPGA로 설계되었고 동작이 검증되었으며, 성능 실험을 통해 고속 동작과 다채널 영상 처리에 적합함을 확인하였다.

1. 서론

2000년대에 접어들면서 보안, 치안 그리고 범죄 예방 등에 관련된 비디오 보안 산업이 이슈화 되고 있다. 비디오 보안 산업은 급속하게 성장하고 있으며, 디지털 영상을 실시간으로 분석하여 판단 할 수 있는 지능형 영상감시 시스템이 세계 각지에 설치되어 있다. 일반적으로 지능형 영상감시 시스템은 그림 1과 같은 영상처리를 통하여 분석된다.



그림 1. 지능형 영상 감시시스템에서의 영상처리 과정

기존의 지능형 영상 감시 시스템에서는 이중 스캔 연결 성분 라벨러(TPL; two-pass connected component labeler)를 사용하였다[1]. TPL은 두 번째 영상 스캔에서 사용될 불필요한 메모리를 요구하고 두 번의 영상 스캔을 통하여 라벨링을 처리하기 때문에 실시간 영상처리에 어려운 문제점이 있다[2]. 이러한 문제점을 개선하기 위해서 최근의 지능형 영상감시 시스템은 단일 스캔 연결 성분 라벨러(SPL; single-pass connected component labeler)를 사용한다 [3], [4].

Bailey는 SPL은 입력 스트림에 대하여 한 번의 스캔과 이전 한라인의 라벨 정보만을 갖고 라벨링을 처리하기 때문에 기존의 TPL의 문제점을 개선하였다[3]. 하지만 라인단위의 라벨링을 처리한 후 추가적인 라벨 병합 구간을 필요로 하는 문제점이 있다. 이는 입력 스트림에 대해서 영상의 패턴에 따라 유동적인 처리시간을 갖으며, 최악의 경우에는 실시간으로 영상처리가 불가능한 경우가 생긴다. 한 라인의 라벨링이 처리된 후에 필요한 추가적인 라벨 병합기간은 [3]은 $O(N)$, [4]는 $O(N^2)$ 의 복잡도를 가진다. N 은 한 라인에서 발생하는 라벨 병합

수를 나타낸다.

본 논문에서는 추가적인 라벨 병합 기간에 대해서 $O(1)$ 의 복잡도를 갖는 SPL의 구조와 방법에 대해서 제안한다. 본 논문의 구성은 다음과 같다. 제2장에서는 제안하는 SPL의 구조를 설명한다. 제3장에서는 제안하는 SPL을 FPGA로 설계하고, 검증한 결과를 토대로 성능을 평가한다. 제4장에서는 본 논문의 결론을 맺는다.

2. 제안하는 SPL의 구조와 방법

본 절에서는 그림 2의 SPL 구조를 제안한다. 제안 구조는 라벨 병합 구간 없이 실시간으로 다채널 처리가 가능한 구조이며, Connected Components Checker (CCC), Label Shift Register (LSR), Label Stack (LS) 그리고 Information Extractor (IE)로 구성된다.

CCC는 입력 픽셀 P 와 이웃 픽셀 A, B, C, D 에 대하여 8방향 인접성을 검사한다. 또한, 8방향 인접성 검사를 통해 P 에 할당된 checked label($L_{checked}$)과 라벨 병합이 발생될 때 얻어지는 merged label (L_{merged}), 그리고 하나의 Event를 생성하여 그 정보를 이웃 모듈에게 전달한다. Event는 P 가 전경이고, 이웃픽셀이 모두 배경일 때 발생하는 New label allocation event (E_{NEW}), P 가 전경이고 이웃 픽셀이 하나의 라벨을 가질 때 발생하는 Neighboring label allocation event ($E_{NEIGHBOR}$), P 가 전경이고 이웃 픽셀이 서로 다른 두 개의 라벨을 가질 때 발생하는 Label merging event ($E_{MERGING}$) 그리고 P 가 배경일 때 발생하는 No event (E_{NO})로 구성된다.

LSR는 $Frame_{width} + 1$ 만큼의 연속된 label register (R_N)들로 구성되며, P 로부터 이전 한 라인의 라벨 정보를 저장하고 있다. CCC에서 생성된 $L_{checked}$ 가 R_0 에 저장되며, 이를 이웃 R_N 에게 전달한다. 또한 이웃하는 R_N 사이에는 라벨 병합을 실시간으로 처리하는

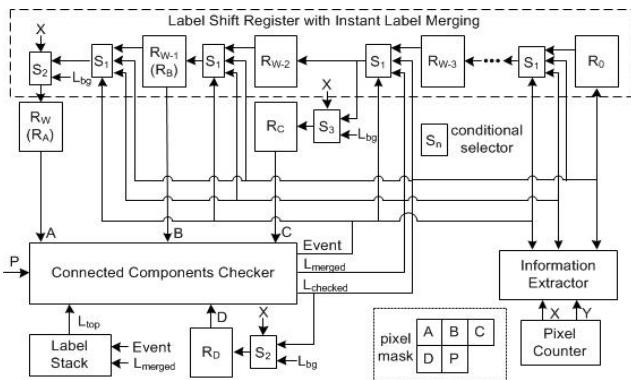


그림 2. 제안하는 SPL의 구조

conditional selector S_1 이 존재한다. S_1 는 $E_{MERGING}$ 이 발생했을 경우 이웃 R_N 에게 전달 받는 라벨정보와 L_{merged} 가 같으면 $L_{checked}$ 를 R_N 에게 전달하고, 그 이외의 경우에는 이웃 R_N 에게 전달 받는 라벨정보를 R_N 에게 전달한다. S_1 은 라벨 병합이 발생할 때 마다 실시간으로 LSR에 존재하는 모든 R_N 에게 라벨 병합을 적용시키기 때문에 제안하는 SPL는 추가적인 라벨 병합 기간이 없이 라벨링 처리가 가능하게 된다. R_N 이외에 LSR에는 P에 대한 이웃 픽셀의 정보를 저장하고 있는 neighboring label registers (R_A, R_B, R_C 그리고 R_D)가 있다. R_A, R_C 그리고 R_D 는 conditional selector S_2, S_3 에 의해서 영상의 경계 조건을 검사받는다. 만약 영상의 경계를 벗어날 경우에 background label L_{BG} 를 제공 받으며, 경계를 벗어나지 않을 경우에는 이웃하는 R_N 에 저장된 라벨정보를 제공 받는다.

LS는 앞으로 사용될 라벨 정보들을 관리하고, 효율적인 메모리 사용을 위해 스택 구조로 설계되었다. LS는 CCC로부터 Event를 받아 동작하며, E_{NEW} 발생시 LS의 top 위치에 저장된 라벨 L_{top} 을 pop하여 P에 할당한다. 그리고, $E_{MERGING}$ 발생시 CCC로부터 받은 L_{merged} 를 push하여 L_{top} 에 저장한다. LS는 사용했던 라벨을 다시 L_{top} 에 저장함으로써 추후에 E_{NEW} 발생 시 재사용이 가능하다.

IE는 CCC와 pixel counter로부터 받은 위치정보(X, Y)를 이용하여 각 물체들의 바운딩 박스와 무게중심을 계산하고 이를 관리한다.

3. 실험 결과 및 평가

제안하는 SPL 구조는 Verilog HDL을 이용하여 설계되었으며, 성능을 평가하기 위하여 Quartus II를 소프트웨어로 시뮬레이션 되었고, Altera Cyclone IV FPGA device에 퓨징되어 그 동작과 성능이 검증되었다. 본 평가에서는 물체의 크기를 영상 전체 해상도를 기준으로 가로 1/8, 그리고 세로 1/3로 제한하였고, 이를 초과하는 검출 물체는 오류로 판단하여 삭제처리 하였다.

표 1은 제안하는 SPL의 성능 평가표이다. 우리는 해상도, 최대 수용 라벨 수 그리고 오브젝트 정보 추출 기능을 서로 다르게 함으로써 성능을 평가하였다.

표 1. 제안하는 SPL의 시뮬레이션 환경의 성능

최대 라벨 수	오브젝트 정보	성능지표	QVGA	VGA	D1
63	바운딩 박스	LUT(ea)	8,435	12,073	12,856
		F_{max} (MHz)	89.79	92.37	90.07
127	바운딩 박스	LUT(ea)	14,223	18,990	20,036
		F_{max} (MHz)	78.94	77.86	78.64
63	바운딩 박스	LUT(ea)	15,791	20,510	21,416
		F_{max} (MHz)	70.48	68.33	67.81
127	무게 중심	LUT(ea)	29,136	36,478	37,984
		F_{max} (MHz)	61.43	60.58	60.43

그림 3은 동일한 해상도에서 서로 다른 패턴의 영상에 대해 식 (1)의 고정적인 clock 수 ($Clk_{RequiredLabeling}$)를 가지고 라벨링을 처리하는 제안하는 SPL의 시뮬레이션 결과 화면이다.



그림 3. 제안하는 SPL의 시뮬레이션 결과 화면 (model-sim)

$$Clk_{RequiredLabeling} = Frame_{width} + Frame_{height} \quad (1)$$

제안하는 SPL의 fps는 라벨러가 동작하는 clock 수 ($Clk_{Labeler}$)와 $Clk_{RequiredLabeling}$ 의 관계에 따라 식 (2)와 같이 수행된다.

$$FPS_{ProposedSPL} = Clk_{Labeler} \div Clk_{RequiredLabeling} \quad (2)$$

표 2는 식 (1), (2)와 표1의 F_{max} ($Clk_{Labeler}$)를 이용하여 얻은 $FPS_{ProposedSPL}$ 를 토대로 수용 가능한 채널수를 나타낸 표이다. 영상

표 2. 제안하는 SPL의 프레임 처리율 및 수용 가능한 채널 수

최대 라벨 수	오브젝트 정보	성능지표	QVGA	VGA	D1
63	무게 중심	프레임 처리율	1,169	300	260
		수용 가능 채널 수	38	10	8
127	무게 중심	프레임 처리율	1,027	253	227
		수용 가능 채널 수	34	8	7
63	바운딩 박스 + 무게 중심	프레임 처리율	917	222	196
		수용 가능 채널 수	30	7	6
127	바운딩 박스 + 무게 중심	프레임 처리율	799	197	174
		수용 가능 채널 수	26	6	5

의 해상도가 D1인 경우에 대해 최대 8 채널까지 수용이 가능한 것을 확인 할 수 있으며, VGA의 경우에는 최대 10 채널까지 수용이 가능한 것을 확인 할 수 있다.

4. 결론

본 논문은 기존의 단일 스캔 연결성분 라벨러의 별도의 라벨 병합 구간을 제거하고, 다채널 처리가 가능한 단일 스캔 연결성분 라벨러 구조를 제안하였다. 제안된 라벨러 구조는 FPGA로 설계되어 기능이 검증되었으며, VGA 최대 10채널, D1 최대 8채널까지 실시간으로 다채널 라벨링 처리가 가능하다. 제안된 라벨러 구조는 다채널 지능형 영상 분석이 필요한 임베디드 시스템에 탑재가 가능하다.

감사의 말

본 연구는 경기도지역협력연구센터(GRRC) 프로그램에 따른 한국항공대학교 차세대방송미디어기술 연구센터(GRRC항공2012-B04) 지원으로 수행되었음

참고문헌

- [1] A. Rosenfeld and J. Pfaltz, "Sequential operations in digital picture processing," *Journal of the ACM*, vo. 13, no. 4, pp. 471-494, 1966.
- [2] M. Jablonski and M. Gorgon, "Handel-C implementation of classical component labeling algorithm", *Proc. Euromicro Symp. Digital System Design*, pp. 387-393, 2004, Rennes, France.
- [3] C. Johnston and D. Bailey, "FPGA implementation of a single pass connected components algorithm," *Proc. IEEE Int. Symp. Electronics, Design, Test and Applications*, pp. 228 - 231, 2008.
- [4] J. Trein, A. Th. Schwarzbacher, and B. Hoppe, "FPGA implementation of a single pass real-time blob analysis using run length encoding," *Proc. MPC-Workshop*, Feb 2008, Ravensburg-Weingarten, Germany.