

---

# 이동 가능한 윈도우를 사용한 효율적인 SAD 설계

이재동\* · 김준섭\* · 이종훈\*\* · 권순\*\* · 문병인\*\*\* · 이용환\*

\*금오공과대학교 전자공학부 · \*\*대구경북과학기술원

\*\*\*경북대학교 전기전자컴퓨터공학부

## Design of Sum of Absolute Differences Based on Shifting Window

Jae-Dong Lee\* · Jun-Sub Kim\* · Jong-Hun Lee\*\* · Soon Kwon\*\*

Byung-In Moon\*\*\* · Yong-Hwan Lee\*

\*School of Electronic Engineering., Kumoh National Institute of Technology

\*\*Daegu Gyeongbuk Institute of Science & Technology

\*\*\* School of Electrical Engineering and Computer Science., Kyungpook National University

E-mail : yofule@kumoh.ac.kr · jskim@kumoh.ac.kr · jhlee@dgist.ac.kr

soonyk@dgist.ac.kr · bihmoon@knu.ac.kr · yhlee@kumoh.ac.kr

### 요 약

본 논문에서는 여러 스테레오 매칭 알고리즘에서 차이값 생성에 사용되는 SAD(Sum of Absolute Differences)의 윈도우 기반 하에서 효율적인 수행에 관해 제시한다. 본 8x8 윈도우 기반의 SAD는 데이터 입력 상태와 데이터 쉬프트 상태로 나뉜다. 데이터 쉬프트 상태에서 디스패리티가 8x8 개의 데이터가 한 클럭에 한번에 생성이 되며 쉬프트 동작으로 데이터 코스트의 연속적인 생성이 가능하다. 본 논문에서는 8x8 윈도우 기반의 SAD를 설계하고 검증한다.

### I. 서 론

비전 시스템의 제작에 있어서 필요한 요소 중 하나는 주변 환경에 대한 거리정보를 수집하는 것이다. 거리정보를 구축하기 위해서 사람의 두 눈과 유사한 형태의 스테레오 카메라 비전 시스템을 사용하게 된다. 스테레오 비전 시스템은 일반적으로 2대의 카메라를 통해 서로 다른 관점에서 찍은 한 쌍의 2D 이미지를 바탕으로 이미지 사이의 대응점 변위를 계산하여 거리 정보를 추출하게 된다. 하지만 깊이 맵 생성에 있어서 연산의 높은 시간 복잡도로 처리속도가 매우 느려 실시간 시스템에 사용하기에는 무리가 있어 보다 비용 효율적인 설계로 실시간성을 만족시켜야 하겠다[1].

본 논문에서는 SAD(Sum of Absolute Differences)를 사용하여 하드웨어 구현에 적합한 최소한의 연산 과정을 거친 비용 효율적인 스테레오 매칭방법을 설계하며, 효율적인 설계를 위해 8x8 윈도우를 사용하여 처리 하도록 하였다. 또한 윈도우의 시프팅 만으로 다음 차이값 생성이 가능하도록 하여 실시간으로 처리되도록 하였다. 마지막으로 윈도우 사이즈에 따른 깊이 맵 생성에 대해 논의 하며, 윈도우 사이즈에 따른 BER(Bit Error Rate)을 비교 분석 하였다.

### II. 스테레오 매칭

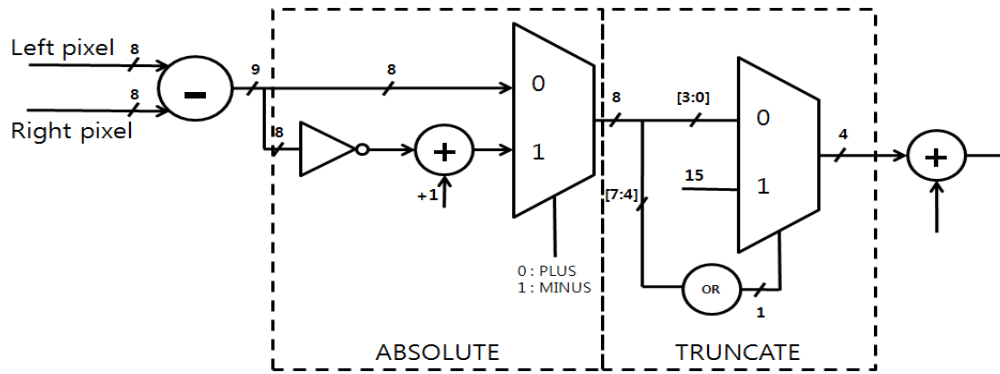


그림 1. SAD 블록

## 2.1 SAD Matching Algorithm

SAD 매칭 알고리즘은 이미지간의 픽셀 간 차이값에서 최소점인 위치로 정합점을 결정한다. 본 논문에서는 0~15의 차이값에 최적화된 스테레오 정합을 설계 하였다. SAD 연산은 식(1)과 같이 표현된다[2].

$$SAD(x, y, d) = |I_1(x, y) - I_2(x + d, y)| \quad (1)$$

이 연산에서 입력값은 8비트 픽셀값을 가지게 되고 차이의 절대값을 취해주게 된다. 그림 1의 ABSOLUTE 블록 부분이 여기에 해당된다.

$$SAD_{trunc}(x, y, d) = \min(SAD(x, y, d), 15) \quad (2)$$

식(2)는 두 영상간의 차이값은 최대 범위가 15이므로 15를 초과하는 값은 카메라의 epipolar 라인의 오류이거나 노이즈에 해당하므로 15를 초과하는 값은 절삭하여 15로 만든다[3]. 15를 넘어가는 값을 찾기 위해서는 뺄셈연산이 필요하지만 절대값 연산에 의해 항상 양수이므로 MSB 4비트중에 1이 하나라도 존재하게 되면 15를 초과하는 점을 고려하면 그림 1의 TRUNCATE 블록처럼 비트4번부터 비트7번까지를 OR 연산하여 최적화를 시킬 수 있다. 이렇게 연산된 결과의 최소값의 위치가 실질적인 disparity map이다[4].

이 map의 disparity 값은 영상에서 밝기 정보로 매핑되어 카메라로부터 가까운 물체는 밝게 나타나고, 멀어질수록 어둡게 표시된다.

## 2.2 Window-Based Processing

SAD 스테레오 매칭 알고리즘의 연산을 픽셀 단위로 처리하게 되면 경계 주변의 오차로 인해 차이값의 오차

가 발생하게 된다. 그래서 경계부분을 포함하여 연산할 수 있도록 식(3)과 같이  $2^n \times 2^n$  크기의 윈도우를 씌우게 되면 경계부분의 오차가 줄어들게 만들 수 있다.

$$D_{\lfloor x/2^n \rfloor, \lfloor y/2^n \rfloor} = D_{\lfloor x/2^n \rfloor, \lfloor y/2^n \rfloor} +$$

$$\sum_x \sum_y \sum_d^5 SAD_{trunc}(x, y, d) \quad (3)$$

하지만 그림2에서 볼 수 있듯이 n의 값이 커질수록 오차가 비례적으로 줄어들지 않음을 볼 수 있다. 또한 n의 값이 커질수록 하드웨어적 소프트웨어적 비용소모가 발생하기 때문에 적절한 값의 선택이 필요하게 된다. n이 3이상 일 때 에러를 감소가 급격히 적어지므로 본 논문에서는 n이 3인 8x8윈도우를 사용하였다. 또한 윈도우의 쉬프트 동작만으로 다음 8x8개의 데이터 재분배 동작없이 8개의 새로운 데이터값의 갱신만으로 다음 깊이 값을 얻을 수 있게 된다.

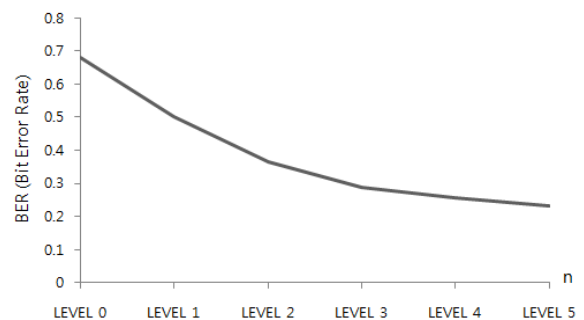


그림 2. 윈도우 크기에 따른 BER



(a) 왼쪽 이미지      (b) 오른쪽 이미지



(c) True Disparity

그림 3. 사용된 이미지 [5]

그림 2의 그래프는 그림 3의 (a)와 (b)의 이미지 한 쌍을 사용하여 생성된 실제 disparity map (c)와 제안한 알고리즘으로 생성된 disparity map인 그림 4의 이미지와 비교한 그래프이다. 여기서 생성된 disparity map은 C언어 기반하에서 제작된 시뮬레이터에 의해 생성된 것이다.

### III. 결 론

본 논문에서는 비용 효율적인 연산을 위해 이동 가능한 윈도우를 사용한 SAD를 설계하고 검증하였다. 그림 4의 disparity map 이미지에서 볼 수 있듯이 윈도우가 커짐에 따라 좀 더 잡음이 적은 disparity map 이미지로 바뀔 수 있다.

앞으로는 FPGA 기반하에서 알고리즘의 실시간성이 만족하도록 설계와 검증을 할 것이다.

### 참고문헌

- [1] 조재일, 최승민, 장지호, 황대환, “지능형 로봇 스테레오 비전 처리 기술”, 전자통신동향분석, 제22권 제2호, 2007.
- [2] Stefania Perri, Daniela Colonna, Paolo Zicari, and Pasquale Corsonello, "SAD-Based Stereo Matching Circuit for FPGAs", IEEE, 2006
- [3] Luigi Di Stefano, Massimiliano Marchionni, Stefano Mattocchia, "A fast area-based stereo matching algorithm", ELSEVIER, VOL 22, page 984-1005, 2004
- [4] S. Vassiliadis, E.A. Hakkennes, J.S.S.M Wong,

G.G. Pechanek, " The Sum-Absolute-Difference Motion Estimation Accelerator, IEEE, 1998

[5] <http://vision.middlebury.edu/stereo/>



(a) 1x1



(b) 2x2



(c) 4x4



(d) 8x8

그림 4. 제안한 알고리즘으로 생성된 disparity map 이미지