

논문 2004-41SD-4-12

# 수직축 Strip 구조를 이용한 실시간 Disparity 시스템의 설계

## (Design of the Real Time Disparity System using Vertical Strip Structure)

강 봉 순\*, 양 훈 기\*\*

(Bongsoon Kang and Hoongee Yang)

### 요 약

본 논문은 2개의 영상을 이용, disparity algorithm을 바탕으로 영상에 나타난 물체의 깊이 정보를 영상의 밝고 어두운 정도로 나타낼 수 있는 방법을 제안한다. 또한 제안된 방법을 이용, 실시간 동작이 가능한 하드웨어 설계 및 구현을 실현한다. 실시간 동작을 위해 본 논문에서는 수직축 strip 구조를 이용하여 영상들의 유사 정도를 계산하였다. 물체의 거리 정보를 흑백 영상으로 변환하여 가까이 있는 물체는 밝게, 멀리 있는 물체는 어둡게 나타낼 수 있도록 하여 각종 영상장치에서 확인할 수 있도록 하였다. 본 논문에서 제안한 방법의 하드웨어는 30 frame/sec의 동작을 하도록 설계하여 Altera APEX20K1000EBC652-3의 device를 사용하여 검증하였으며, Hynix 0.35um CB35 ASIC library와 256PQFP package를 이용하여 IC로 제작하였다.

### Abstract

In this paper, we propose the method that analyzes the depth of object using 2 images in the disparity algorithm. It also presents the design and implementation of the proposed method for a real time processing. The proposed system uses the vertical strip structure for calculating similar pixel numbers for the processing and converts the depth of object into gray scale images in order to be displayed on various display devices. The hardware using the proposed method is operating with 30 frames/sec and verified by using the Altera APEX 20K1000EBC652-3. The proposed method is also implemented into IC by using the Hynix 0.35um CB35 ASIC library and 256PQFP package.

**Keywords :** Disparity, WMC, Vertical Strip, Stereo Image

## I. 서 론

일반적으로 물체의 위치를 3 차원적으로 파악할 경우 2 곳의 지점에서 물체의 위치를 파악하고 삼각 측량법을 이용하여 물체의 거리를 측정하게 된다<sup>[1]</sup>. 이를 바탕으로 2 개의 2 차원 영상을 이용, disparity algorithm을 이용하여 2차원 영상 내에 있는 물체의 깊이 정보를 파악할 수 있다. 이는 2 개의 2 차원 영상에 일정한 크기의 정합창(matching window)을 씌우고 정합창 내의 영상이 같을 때, 정합창 내의 영상이 얼마나 떨어져 있

음을 판단하여 물체의 거리 정보를 파악할 수 있게 된다. 이를 이용하는 분야는 대체적으로 물체 인식, 움직임 추출, 얼굴 영역 추출, 제스처 인식 기술 등에 사용되게 된다. 그러나 이들 분야에 적용하기 위해서는 반드시 실시간 동작이 되어야 하나, 일반적인 disparity algorithm을 이용 할 경우 정합창 내의 영상이 어느 정도 유사한가를 판단하는 과정에서 상당히 많은 연산 시간이 소요된다<sup>[2-4]</sup>.

이에 이 문제점을 해결한 여러 가지 방법이 제안되어졌다. 첫 번째로 웨이블릿 변환과 기준점을 이용하여 변위를 추정하는 방법이 있다. 이 방법은 기존 방법에서 문제였던 연산 시간이 많이 소요된다는 단점을 보완하기 위한 방법이다. 그러나 이 방법은 하나의 변위를 얻기 위하여 영상을 웨이블릿 변환을 해야 하고, 변환된 영상을 이용, 2 번의 탐색을 통한 기준점 추출, 기준

\* 정희원, 동아대학교 전기전자컴퓨터공학부  
(School of Electrical, Electronic and Computer Eng., Dong-A Univ.)

\*\* 정희원, 광운대학교 전파공학과  
(Dept. of Radio Science and Eng., Kwangwoon Univ.)

접수일자: 2003년1월6일, 수정완료일: 2004년4월1일

점의 edge detecting. 기준점 이외의 변위추출, 역 웨이 블릿 변환을 취해야 하는 많은 과정을 거치기 때문에 하드웨어 구현 시 실시간 동작 속도 및 하드웨어 크기가 매우 커지게 된다는 단점이 있다<sup>[4]</sup>. 이에 두 번째 방법으로 입력되는 영상을 직접 사용하는 방법이 있다. 이 방법은 정합창 내의 영상의 유사도를 계산하는 부분에서 중복으로 연산하는 부분을 줄여 실시간 동작을 가능하게 하는 하드웨어 구조이다. 이는 30 frames/sec으로 실시간 동작이 가능하나, 중복으로 연산하는 부분을 없애기 위하여 상당한 크기의 메모리를 사용하게 된다. 이 기능을 구현하기 위한 필요한 메모리의 용량만 계산해도 307,200 bits(=320 pixels×11 정합창 크기×64 탐색범위 + 4 bits×320 pixels×64 탐색범위)로 하드웨어 크기가 매우 증가된다는 단점이 있다<sup>[2]</sup>.

이에, 본 논문에서는 정합창 내에서 영상의 유사도를 파악할 때, 정합창을 세로로 분할시킨 세로축 strip 구조를 이용, 정합창 내에서 영상의 유사도를 파악하는 방법을 제안한다. 이 방법은 세로축 strip 내에서 영상의 유사도를 일정시간 계속 사용하기 때문에 중복 연산 대신 간단한 크기의 buffer를 사용하여 연산된 값을 계속 사용하게 된다. 이 경우 메모리의 크기는 2,816 bits(=64 탐색범위×11 정합창 크기×4 bits)로 줄게 되고, 하드웨어 크기 역시 줄일 수 있게 된다. 또한 제안한 방법을 이용하면 한번에 탐색 범위만큼의 정합창 내에서 영상의 유사도를 파악할 수 있게 된다. 또한 연산을 하지 않은 시간을 이용하기 때문에 하드웨어 크기도 줄일 수 있을 뿐만 아니라, 이로 인하여 연산 시간을 15 frames/sec 이내로 줄일 수 있고 각 frame을 두 번 출력하여 실시간 동작을 가능하게 한다<sup>[5]</sup>.

본 논문의 II장에서는 disparity algorithm에 대해 간단히 알아보고, 기존에 사용한 방법 및 제안한 수직축 strip 구조를 이용하여 거리 정보를 계산하는 방법을 제안한다. III장에서는 수직축 strip 구조를 이용하여 구성한 본 하드웨어의 블록 구성도 및 기존에 사용되었던 구조와의 차이점, 각 블록의 하드웨어 구조에 대하여 다루고 IV장에서는 시뮬레이션 결과 비교와 FPGA로 구현한 하드웨어 검증을 논하면서 본 연구에서 제안한 방법의 타당성을 다루고 마지막으로 V장에서 결론을 맺게 된다.

## II. 알고리즘

### 1. 일반적인 disparity algorithm

일반적인 disparity algorithm은 2곳의 지점에서 얻은 영상을 이용하여 물체의 깊이 정보를 얻을 수 있는 삼각 측량법에 기초를 두고 있다. 삼각 측량법의 기본은 다음과 같다. 우선 그림 1과 같이 2개의 카메라와 물체를 위치시킨다. x축 상에 광축을 서로 평행하게 하고, 카메라 2대의 거리 차이를  $b$ 가 되도록 배치한 후, 초점 거리를  $f$ , 3차원 공간상의 한 점을  $P(x_w, y_w, z_w)$ ,  $P(x_w, y_w, z_w)$ 가 좌우 카메라의 영상면에 투영된 점의 좌표를  $(x_L, y_L)$ ,  $(x_R, y_R)$ 이라 한다면 기준 좌표계에 대한 카메라 지표계의 원점이 같은 거리에 존재하고 높이에 해당하는 좌표가 같은 상태라는 에피폴라 구속 조건<sup>[6]</sup>에 의해  $y_R = y_L$ 이며 x축 좌표는 삼각형의 비례 관계식에 따라 수식 (1)과 수식 (2)와 같이 나타낼 수 있다.

$$x_L = \frac{fx_w}{z_w}, \quad x_R = \frac{f(x_w - b)}{z_w} \tag{1}$$

$$z_w = \frac{bf}{x_L - x_R} = \frac{bf}{d} \tag{2}$$

이때 수식 (2)의  $d$ 를 변위라 하고 이는 깊이 정보에 반비례함을 알 수 있다. 즉 카메라와 물체의 거리가 가까우면  $d$ 의 값은 커지게 되어 이를 이용하게 되면 기

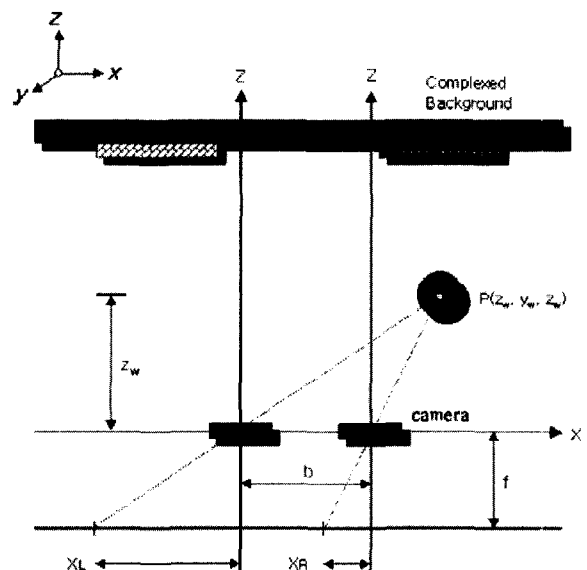


그림 1. 삼각 측량법을 이용하기 위한 카메라 및 물체 위치도  
Fig. 1. Position diagram of the cameras and objects for triangulation method.

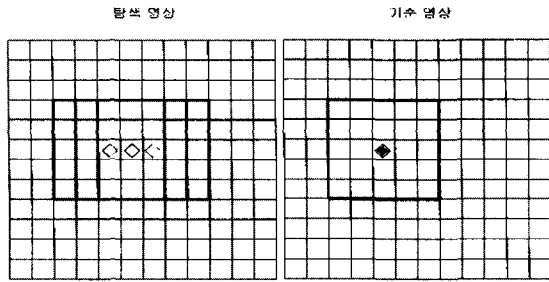


그림 2. 2개의 영상에 정합창 (matching window)을 씌우는 방법

Fig. 2. Method of matching window inside two images.

본적인 물체의 깊이 정보를 얻을 수 있다. 2 개의 2 차원 영상에 적용할 경우, 영상에 일정한 크기의 정합창을 씌운 후 정합창 내의 영상이 유사할 때, 각각의 영상에 나타난 물체의 거리 차이(수식 (2)의  $d$ )를 파악하게 된다. 이 거리 차이는 수식 (2)와 같이 카메라와 물체의 거리가 가까우면  $d$ 의 값이 커지게 되고 반대로 카메라와 물체의 거리가 멀게 되면  $d$ 의 값이 작아지게 되어 우리는  $d$  값에 의하여 물체의 거리 정보를 파악할 수 있게 된다. 이것이 disparity algorithm의 기본 흐름이 된다<sup>[7]</sup>.

그림 2는 2 개의 영상에 각각 5x5 크기의 정합창을 씌운 형태를 나타낸다. 2 개의 영상은 각각 기준 영상과 탐색 영상으로 나누어지는데, 기준 영상의 정합창 내의 영상과 유사한 영상을 탐색 영상의 정합창 내에서 찾아야 한다. 즉, 그림 2와 같이 기준 영상과 탐색 영상의 일정한 지점에 일정한 크기의 정합창을 씌우고, 탐색 영상에서의 정합창을 오른쪽으로 이동하면서 유사 영상을 찾게 된다. 여기서 탐색 영상에서 정합창이 옮겨지는 범위를 탐색 범위라 하는데, 이는 우리가 구하려는 물체의 깊이 정보로 수식 (2)의  $d$ 값과 같다고 할 수 있다.

정합창 내에서 영상의 유사함을 판별할 수 있는 방법으로 4가지 방법이 있다. 첫 번째로 SSD(Sum of Squared Distance) 방법으로 이는 정합창 내의 화소들 간의 밝기 차이를 이용하여 영상의 유사도를 계산하는 방법이다<sup>[3]</sup>. 그러나 이 방법은 한 화소가 유사도에 미치는 영향이 화소의 밝기 값에 따라 다르기 때문에 화소간의 밝기 변화가 큰 영역이 작은 영역으로 변위의 경계선이 연장되는 경계의 지나침(Boundary Overreach) 현상이 생겨 경계 부분에서 잘못된 결과를 초래하게 된다. 두 번째 SAD(Sum of Absolute Distance) 방법은 SSD 방법과 매우 유사하지만 SSD 방법과 똑 같은 문제점을

안고 있어 사용하기가 힘들다<sup>[3]</sup>. 세 번째 방법으로 NC C(Normalized Correlation Coefficient) 방법으로, 화소들 간의 상관관계를 이용하는 방법이다. 이 방법은 카메라에 의한 잡음 등에 강한 효과를 가지고 있지만, 경계의 지나침 현상 및 하드웨어 구현 시 문제가 발생하게 된다<sup>[5]</sup>. 마지막 방법으로 WMC(Window Matching Counts) 방법이 있게 된다. 이 방법은 정합창 내의 밝기가 비슷한 화소의 개수를 계산하는 방법으로 수식 (3)과 같이 나타낼 수 있다. 여기서  $Th$ 는 화소의 유사 정도를 나타내는 값으로 두 화소의 차이가  $Th$ 보다 작을 경우 같은 화소로 인식되게 하는 것이다<sup>[3]</sup>.

$$WMC(x, y, d) = \sum_w P(x, y, d)$$

$$P(x, y, d) = \begin{cases} 1, & \text{if } |R_{xy} - L_{x+d, y}| \leq Th \\ 0, & \text{others} \end{cases} \quad (3)$$

이 방법은 한 화소가 유사도에 미치는 영향이 밝기 값에 관계없이 동일하므로 경계의 지나침 현상이 발생하지 않고 다른 방법보다 수식이 간단하기 때문에 하드웨어 구현이 용이하게 된다. 이에 본 하드웨어에서는 WMC 방법을 사용하여 정합창 내의 영상의 유사도를 판별하게 된다.

유사도 판별법은 1개의 기준 영상의 정합창 내의 영상과 비교한 탐색 범위만큼의 정합창 내에서 영상의 유사도를 수식 (3)을 통하여 얻을 수 있게 된다. 이렇게 구한 값들 중 최대 값을 가지는 값이 있게 되는데, 이때 최대 값의 의미는 기준 영상과 탐색 영상의 정합창 내의 영상이 상당히 유사함을 나타내고, 이는 두 영상의 정합창 내의 영상이 같음을 의미한다. 이때, 즉  $WMC(x, y, d)$ 가 최대 값을 가질 때의 탐색 범위값  $d$ 는 두 영상에 나타난 물체의 떨어진 거리를 나타내고 이는 물체의 깊이 정보를 의미하게 된다. 이렇게 구한  $d$ 의 값을 '스테레오 변위'라 하고 전체 영상에서 구한 스테레오 변위를 흑백 영상화하여 물체의 거리 정보를 영상의 밝고 어두운 정도로 파악할 수 있게 된다. 이렇게 영상화 한 이미지를 우리는 '변위 지도 (disparity map)'라 부른다. 변위 지도는 가까운 물체는 밝게 나타나고 반대로 카메라에서 멀리 떨어진 물체는 어둡게 나타나는 특성을 가지고 있다.

2. 하드웨어 구현이 용이하도록 제안된 algorithm 위에서 보았듯이 이 알고리즘을 직접 하드웨어에 적용하게 될 경우 각 영상의 정합창 내에서 영상의 유사

도를 판별하는 부분에서 많은 연산 시간이 소요되게 된다. 예를 들어 2개의 320×240 영상과 11×11인 정합창의 크기, 63의 탐색 범위를 가진다면 약 450만개(=(320 pixels-10)×(240 lines-10)×64 탐색범위), 10은 정합창 크기에서 1을 뺀 것)의 정합창 내의 영상의 유사도를 파악해야 하는데 약 0.67 ms 정도의 시간이 소요되어, 0.067 ms(=15 frames/sec)가 요구되는 실시간 동작에 무리가 발생하게 된다<sup>[7]</sup>.

이에 본 논문에서는 이러한 문제점을 해결할 수 있는 방법을 제안하고자 한다. 제안된 방법은 중복 연산되는 부분 중 필요한 부분만 메모리를 사용하고, 나머지 부분은 중복 계산을 허용하는 방법을 사용하였다. 그림 3에 제안된 방법을 도시하였다. 영상의 유사도 계산 시 그림 3과 같이 수직축 strip을 각 영상에다가 씌운 후 수직축 strip내의 영상의 유사도를 수식 (4)을 이용하여 계산한다. 여기서 기준 영상의 한 개의 수직축 strip내의 영상과 비교하는 탐색영상의 수직축 strip내의 영상이 64개로 되어 있는데, 이것은 한꺼번에 64개에 해당하는 영상의 유사도를 계산하게 된다. 여기서 64는 '탐

색 범위'를 의미하는 것으로 기준영상과 탐색영상의 수직축 strip내의 영상의 유사도를 탐색 범위만큼 계산하게 되는 것이다. 이렇게 계산된 수직축 strip내의 영상의 유사도를 수식 (5), 그림 4와 같이 모두 더하여 정합창 내에서 영상의 유사도를 계산하게 된다.

$$S(x, y, d) = \sum_{i=-xy}^{xy} P(x, y + i, d) \tag{4}$$

$$WMC(x, y, d) = \sum_{i=-xy}^{xy} S(x + i, y, d) \tag{5}$$

위의 방법을 기본으로 하여 탐색 범위에 해당하는 정합창 내에서 영상의 유사도를 계산할 수 있는데, 그림 5에서 각 strip내에서 영상의 유사도인 S0\_0~S10\_0을 더하면 탐색범위가 0일 때의 정합창(11×11) 내에서 영상의 유사도를 계산할 수 있음을 보여준다. 이를 이용하면 전체 탐색범위(0~63)에서의 영상의 유사도를 한꺼번에 계산할 수 있어 실시간 동작에 유리하다. 그림을 보면 S1\_0~S11\_0을 더하여 두 번째에 해당하는 정합창 내의 유사도를 계산할 때 S1\_0~S10\_0은 중복으로 계산되는 부분임을 알 수 있다. 중복연산을 피하기 위하여, 처음 연산시에 이 값들을 메모리를 이용하여 저장하고 있다면 두 번째 연산부터는 S11\_0에 해당하는 값만 계산하면 되기에 작은 하드웨어로 구현이 가능함을 알 수 있다.

위와 같은 방법으로 한 화소에 해당하는 탐색 범위 내에서의 정합창 내에서 영상의 유사도를 계산할 수 있다. 이렇게 탐색 범위 내에서 계산한 값들 중 영상의 유

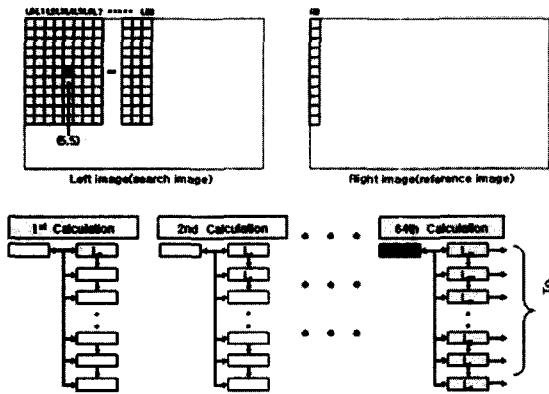


그림 3. 수직축 strip 내에서 영상의 유사도를 계산하는 방법

Fig. 3 Calculating the similar pixel numbers of image inside the vertical strip structure.

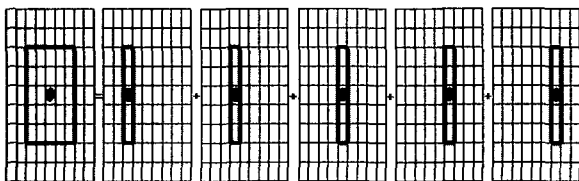


그림 4. 수직축 strip 구조를 이용하여 정합창 내에서 영상의 유사도를 계산하는 방법

Fig. 4. Calculating the similar pixel numbers of image inside the matching window using the vertical strip structure.

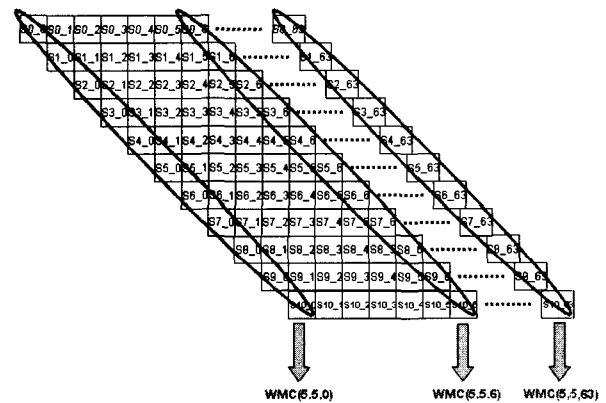


그림 5. 수직축 strip 내에서 영상의 유사도를 이용, 정합창 내에서 영상의 유사도를 계산하는 방법

Fig. 5. Calculating the similar pixel numbers inside the matching window using the one inside the vertical strip structure.

사도가 최대 값을 가질 때의 탐색 범위 값을 계산하게 된다. 이를 나타낸 것이 수식(6)이다.

$$DIS(x, y) = d, \text{ if } [MAX\{WMC(x, y, d)\}] \quad (6)$$

여기서 탐색 범위 값  $d$  는 각 영상에 나타나 있는 물체의 거리 정보가 되고 우리는 이 값을 '스테레오 변위'라 부른다. 즉 스테레오 변위 값이 클 경우 물체와 카메라와의 거리가 가깝다는 것을 의미하고 반대로 작을 경우에는 물체와 카메라와의 거리가 멀리 떨어져 있다는 것을 의미한다. 이렇게 구한 스테레오 변위는 흑백 영상화하여 변위 지도로 만들 수 있고 이는 사용자가 눈으로 직접 물체의 거리를 변위 지도의 밝기로 판별할 수 있게 된다.

### III. 하드웨어 아키텍처

그림 6은 본 논문에서 제안한 알고리즘을 이용한 실시간 disparity 시스템의 내부 블록도를 나타낸다. 제안한 하드웨어는 30 frames/sec의 640×480 영상을 입력받아 15 frames/sec의 320×240의 영상으로 변환 후 스테레오 변위를 추출하여 영상장치에 15 frames/sec으로 출력하게 된다. 본 논문에 제안한 하드웨어는 7개의 기능 블록으로 구성되어 있는데, (i) 입력되는 영상의 동기화 및 SRAM에 데이터를 저장 및 읽어내기 위한 제어 신호를 생성하는 Core\_pre 블록, (ii) 탐색 범위만큼의 수직축 strip 내의 영상의 유사도를 계산하는 Strip\_Calc 블록, (iii) Strip\_Calc 블록에서 계산된 수직축 strip내의 영상의 유사도를 저장하는 역할을 하는 Strip\_buffer 블록, (iv) 저장된 수직축 strip내의 영상의 유사도를 이용하여 전 탐색 범위에서의 정합창 내의 영상의 유사도를 계산하는 WMC\_Calc 블록, (v) 탐색 범위만큼 구한 정합창 내의 유사도들 중에서 최대 값을 가질 때의 탐색 범위값, 즉 스테레오 변위 값을 계산하는 Max\_Sel 블록, (vi) 계산된 스테레오 변위를 영상장치인 TV에 출력하기 위한 제어 신호 생성 및 SRAM에 저장 및 읽어내기 위한 제어 신호를 생성하는 Core\_post 블록, (vii) 마지막으로 입력되는 영상을 저장하기 위한 4개의 SRAM, 계산된 스테레오 변위를 디스플레이 하기 위해 데이터를 저장하는 2개의 SRAM으로 구성된 MACRO 블록으로 구성되어 있다.

Core\_pre 블록은 입력되는 2개의 영상의 동기를 맞추는 역할과 30 frames/sec, 640×480 크기로 입력되는

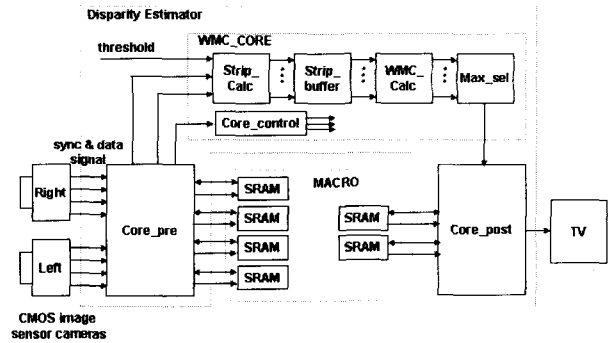


그림 6. 제안된 방법을 이용한 실시간 disparity 시스템의 블록도  
Fig. 6. Block diagram of the real time disparity system using proposal method.

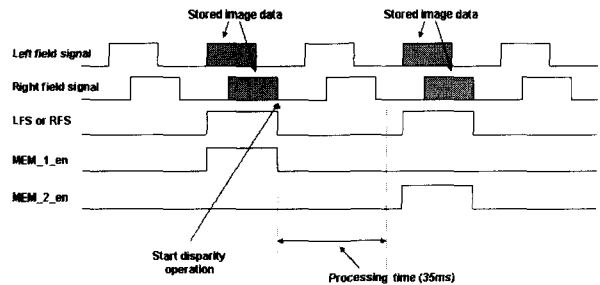


그림 7. Core\_pre의 타이밍도  
Fig. 7. Timing diagram of the Core\_pre.

영상을 15 frames/sec, 320×240 크기의 영상으로 만들고 이렇게 만들어진 영상을 SRAM에 저장하는 역할을 하는 블록이다. 그림 7은 Core\_pre 블록에서의 타이밍도를 나타낸 것으로 이는 입력되는 영상이 동기가 맞지 않기 때문에 동기를 맞추는 방법을 나타낸 것이다. 그림에서 왼쪽과 오른쪽 카메라가 동기가 맞지 않게 입력된다고 할 때 각 카메라에서 입력되는 필드 신호를 이용하여, 2개의 필드 신호를 OR 연산한 신호와 입력되는 필드 신호가 '1'일 때 데이터를 SRAM으로 보내게 된다. 여기서 OR 연산한 신호는 한 frame을 건너 뛰어 생성되게 된다. 이렇게 동기를 맞춘 후 변위 연산을 위해 데이터를 읽어내게 되는데, 이때 소요되는 연산 시간은 약 35 ms로 1 frame이 넘게 된다. 이로 인하여 한 개의 카메라에 한 개의 SRAM을 사용할 경우 OR 연산한 신호가 31 ms보다 커질 때 SRAM에서는 읽기 신호와 쓰기 신호가 동시에 발생되게 된다. 이에 본 논문에서는 하나의 카메라에 2개의 SRAM을 할당하여 그림 7과 같이 교번 하는 형태로 사용하여 이와 같은 문제점을 해결하였다.

Strip\_Calc 블록은 전 탐색 영역에 해당하는 수직축 strip 내에서 영상의 유사도를 계산하는 블록으로 그림

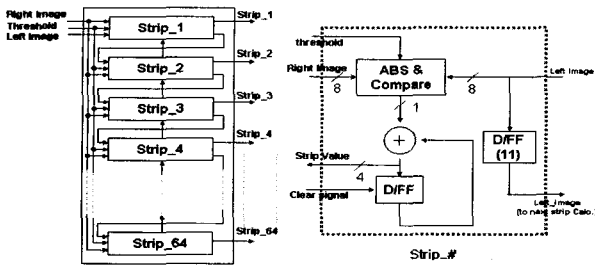


그림 8. Strip\_Calc의 블록도  
Fig. 8. Block diagram of the Strip\_Calc.

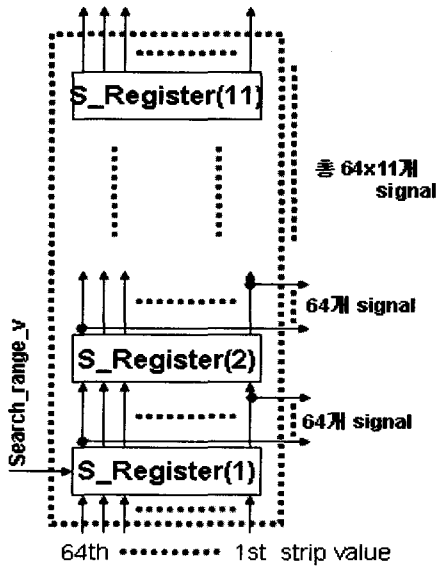


그림 9. Strip\_Buffer의 블록도  
Fig. 9. Block diagram of the Strip\_Buffer.

8과 같이 구성되어 있다. 여기서 사용된 수직축 strip의 크기는 11로 정합창의 크기는 11×11이라 보면 된다. 총 64개의 Sub\_strip으로 구성되어 있고, 이는 탐색 범위에 해당하는 값이다. 즉 이 블록은 64개의 수직축 strip내에서 영상의 유사도를 한번에 계산하게 된다. 그림 8의 오른쪽 그림은 각 sub\_strip의 내부 구성도를 보여준다. 2개의 카메라에서 입력되는 데이터를 이용하여 수식 (4)을 수행하는 블록이다.

Strip\_Buffer 블록은 Strip\_Calc 블록에서 구한 영상의 유사도를 저장하는 역할을 하는 것으로 그림 9와 같이 구성되어 있다. 이는 수식 (5)과 그림 4의 방법으로 전체 탐색 범위에 해당하는 정합창 내에서 영상의 유사도를 계산할 경우 수직축 strip 내에서 영상의 유사도를 저장해서 계속 사용하기 때문이다. 유사도를 저장하기 위해서는 메모리가 필요한데, 필요한 메모리 용량이 2816 bits의 작은 용량이기때, 이 블록은 11개의 세부 S\_Register로 구성된 flip-flop으로 구성하여 탐색 범위 내의 수직축 strip 내에서 영상의 유사도를 저장하게 하

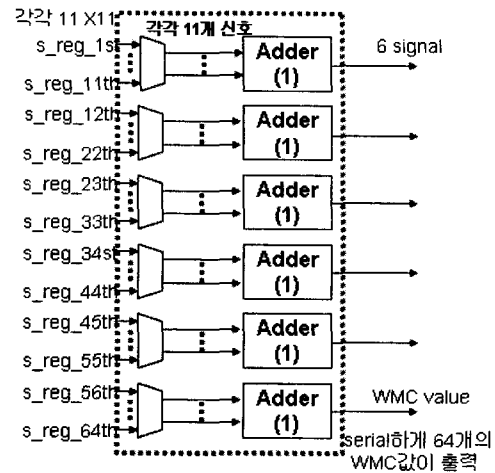


그림 10. WMC\_Calc의 블록도  
Fig. 10. Block diagram of the WMC\_Calc.

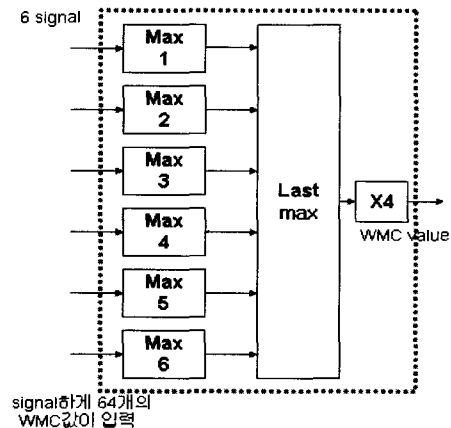


그림 11. Max\_Sel의 내부 블록도  
Fig. 11. Block diagram of the Max\_Sel.

였다. 본 하드웨어에서는 64개의 탐색 범위를 가지므로 64개의 수직축 strip 내에서 영상의 유사도 값을 저장하게 된다. S\_Register 내의 데이터는 11 클럭마다 다음 단으로 데이터를 전달하게 된다. 각 S\_Register는 수직축 strip내에서 영상의 유사도 값이 출력하게 되는데 이는 다음 블록에서 수식 (5)을 계산하기 위해 출력된다.

WMC\_Calc 블록은 Strip\_Buffer 블록에 저장된 수직축 strip 내에서 영상의 유사도 값을 이용하여 정합창 내에서 영상의 유사도를 계산하는 블록으로 그림 10과 같이 구성되어 있다. 이 블록은 한 화소의 탐색 범위만큼, 즉 여기서는 64개의 정합창 내의 유사도를 계산하게 된다. 입력 데이터는 11클럭마다 입력되어지고 입력되는 값들은 64개이기 때문에 11 클럭만에 64개의 정합창 내의 영상의 유사도를 계산해야 한다. 그러나 기존에 사용되었던 하드웨어 구조는 Adder기를 64개를 사

용하여 64개의 정합창 내의 영상의 유사도를 계산하는 구조이어서 동작 시간 면에서도 비효율적이고 하드웨어 복잡도 역시 늘어나게 된다. 이에 본 블록에서는 입력되는 값 64개를 크게 6개로 나누어 첫 번째 세부 블록에는 1~11번째 데이터, 두 번째는 12~22번째 데이터를, 이런 방법으로 총 6개로 나누어 입력하는 방법을 채택하였다. 각 세부 블록에는 한 클럭마다 하나의 데이터를 입력받아서 수식 (5)을 실행, 매 클럭마다 정합창 내의 영상의 유사도를 계산하여 출력하게 된다. 최종적으로 11 클럭마다 한 화소에 해당하는 64개의 정합창 내에서 영상의 유사도를 계산할 수 있게 된다<sup>[3]</sup>.

Max\_Sel 블록은 WMC\_Calc 블록에서 구한 64개의 정합창 내에서 영상의 유사도 값을 이용하여 수식 (5)를 사용, 최대 값을 찾아 스테레오 변위 값을 얻게 되는 블록으로 그림 11과 같이 구성되어 있다. 기존의 방법은 64개의 데이터 값이 순차적으로 입력되기 때문에 데이터를 한꺼번에 비교하여 최대 값을 구하는 방법을 사용하였지만 이는 하드웨어 복잡도에서 부담을 느낄 수 있게 된다. 이에 64개의 정합창 내의 영상의 유사도 값이 WMC\_Calc 블록에서 출력되는 형태로 이용한다. 이렇게 입력되는 데이터는 첫 단의 Max\_# 블록에서 최대 값을 구하게 된다. 이 블록은 데이터를 매 클럭마다 입력받아 매번 비교하여 최대 값을 11 클럭마다 출력하게 된다. 이렇게 하여 출력된 값은 6개이고 Last\_max블록에서 6개중 가장 최대 값을 가질 때의 탐색 범위 값, 즉 스테레오 변위를 출력하게 된다. 이 값은 11 클럭마다 출력되고 마지막에 스테레오 변위에 4배를 하게 되는데, 이는 영상 매체에 스테레오 변위를 출력하기 위해서이다. 스테레오 변위값 자체는 0~63까지의 값만을 가지게 되는데, 이를 그대로 사용하여 영상 매체에 출력하게 되면 화면이 대체적으로 어둡게 보이기 때문에 4배를 하여 0~252까지의 값을 가지도록 하였다<sup>[7]</sup>.

Core\_post 블록은 영상 매체인 TV Monitor에 스테레오 변위를 출력하기 위해 이전 블록에서 계산된 스테레오 변위를 일시 저장함에 필요한 SRAM의 제어 신호를 생성하는 역할 및 TV Monitor에 필요한 제어 신호를 생성 시켜주는 역할을 하는 블록이다. TV Monitor에 출력하기 위해서 KS0119 Encoder를 사용하게 되는데 이 블록에서 Encoder에 필요한 제어 신호인 horizontal sync, vertical sync 및 기준 클럭을 생성시켜 준다<sup>[8 10]</sup>.

MACRO 블록은 6개의 SRAM으로 구성되어 있다. 사용한 SRAM은 Samsung에서 생산된 256K×18bit의

크기를 가지는 Synchronous SRAM KM718V895를 사용하였다<sup>[11]</sup>. 이 블록은 입력되는 영상을 저장하기 위해, 계산된 스테레오 변위를 출력하기 위해 데이터를 저장하는 역할을 하게 된다.

본 하드웨어에서 사용한 카메라는 Omnivision사에서 생산된 CMOS Sensor Camera로 OV7620A 2개를 이용하였다. 사용된 카메라는 640×480, 30 frame/sec, interlace 방식으로 영상이 입력된다<sup>[12]</sup>.

#### IV. 실험 결과

그림 12는 본 논문에서 제안된 방법을 이용하여 구한 스테레오 변위 값을 흑백 영상으로 변환한 영상인 변위 지도를 보여준다. 정합창의 크기는 11×11, 탐색 범위는 63, threshold 값은 15로 하여 제안된 알고리즘을 이용 시뮬레이션 하였다. 그림에 나타난 것과 같이 2 개의 영상에서 기둥 부분이 뒷배경보다 앞에 있음을 알 수 있는데 시뮬레이션 결과를 보게 되면 기둥 부분이 다른 부분들 보다 좀 더 밝게 나타난 것을 알 수 있다.

본 논문에서 제안한 시스템은 하드웨어 언어인 VHDL을 이용하여 설계하였으며, 실시간 검증을 위해서 Altera APEX20K1000EBC652-3 FPGA를 사용하여 구현하였다<sup>[13]</sup>. Altera FPGA에 사용된 Logic Cell의 크기는 약 47% 이었고, 동작 주파수는 43.08 MHz이다. 이는 하드웨어 동작 주파수를 만족하는 것이다. 그림 13은 본 논문에서 제안한 방법을 실시간 검증을 위해 제작된 검증용 데모 보드이며, 그림 14는 데모 보드를 이용하여 검증한 TV set이다. 검증용 데모 보드는 2개

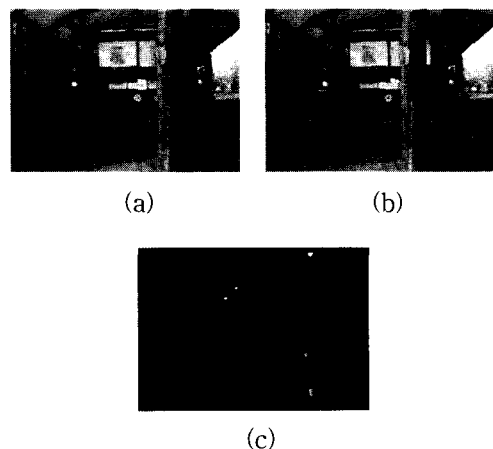


그림 12. 2개의 영상을 이용하여 구한 변위 지도  
(a)왼쪽 영상 (b)오른쪽 영상 (c)변위 지도  
Fig. 12. Disparity map obtained by using two images  
(a)Left Image (b)Right Image (c)Disparity map

의 CMOS Sensor Camera(Omnivision사의 OV7620A)에서 입력되는 Y값(Luminance값)과 필드 신호, Horizontal sync신호를 FPGA로 받아들인다. 그 후 영상 데이터를 저장하기 위해 4개의 SRAM(Samsung Electronic사의 KM718V895 SRAM)에 데이터를 저장하게 된다. 저장된 데이터는 disparity연산을 하기 위해 다시 FPGA로 데이터를 받아 들여 연산을 하게 된다. 연산을 하여 구한 스테레오 변위는 다시 2개의 SRAM에 저장되고 최종 영상 매체에 출력하기 위해 FPGA에 데이터를 거친 후 Video Encoder에 스테레오 변위 데이터를 보내게 된다. 그림 13은 제안한 실시간 disparity 시스템을 데모보드와 TV set을 이용하여 검증한 화면이다. 그림에 나타난 것과 같이 2명의 사람을 거리를 두고 나타내었다. 최종 결과에 나타난 것과 같이 앞에 있는 사람이 밝게 나타남을 알 수 있고 뒤에 있는 사람은 앞에 있는 사람보다 보다 어둡게 나타남을 알 수 있다. 이를 통하여 본 논문에 제안한 방법을 이용하여 15 frames을 두 번씩 출력하여 30 frames/sec의 실시간 동작이 가능함을 나타낼 수 있었다.

제안한 시스템은 Synopsys simulator를 사용하여 검

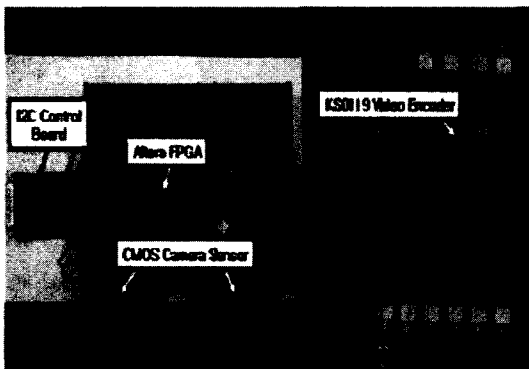


그림 13. 실시간 disparity 시스템의 데모 보드 구성도  
Fig. 13. Demonstration board of the real time disparity system.



그림 14. 실시간 disparity 시스템의 데모 TV set  
(a)입력 영상 (b)출력 영상  
Fig. 14. Demonstration TV set of the real time disparity system. (a)Input Image (b)Output Image

증하였고, 검증된 VHDL 소스 코드의 하드웨어 복잡도를 살펴보기 위하여 Hynix 0.35um CB35 library를 이용하여 합성하였다<sup>[14]</sup>. 설계된 하드웨어의 동작 주파수는 27MHz이고, 악조건 제약은 3.0V공정과 온도 100°C 조건을 사용하였다. 표 1은 본 논문에서 제안한 알고리즘을 이용한 실시간 disparity 시스템의 합성된 gate counts와 각 블록에서 critical path을 보여주는 maximum-timing을 나타낸다. 하드웨어의 복잡도는 일반적으로 2-input NAND gate 기준으로 표시되는데, 본 논문에서 제안한 방법은 117,023 gate counts가 필요하였다. 또한 설계된 하드웨어의 최대 critical path는 Core\_pre 블록이었으며, 최소 연산 주파수는 28.75MHz로 하드웨어의 요구 동작 속도인 27MHz를 만족함을 알 수 있다.

그림 15는 제안된 시스템의 레이아웃을 수행한 결과이다. 그림 15(a)는 레이아웃의 물리적 패턴을 나타내며, 그림 15(b)는 실제 레이아웃 영역만을 확대한 그림이다.

표 1. 실시간 disparity 시스템의 gate counts와 maximum timing.

Table 1. Gate counts and maximum timing of the real time disparity system.

Module Name	Gate Counts	Maximum Timing (MHz)
Core_pre	12,884	29.85
Strip_Calc	55,808	60.06
Strip_Buffer	27,690	34.45
WMC_Calc	8,313	74.12
Max_sel	2,743	65.23
Core_control	685	200
Core_post	6,221	39.90
Top_i2c	2,291	173
Total Gate Counts and Max. Timing	117,023	28.75

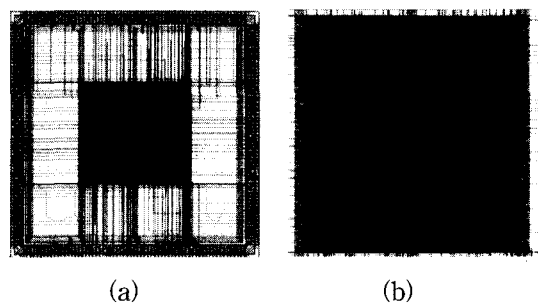


그림 15. 제안된 시스템의 레이아웃  
(a)레이아웃 (b)확대한 실제 레이아웃 영역  
Fig. 15. Layout of the proposal system. (a)Layout (b)Zoom in the active layout



본 논문에서 제안한 disparity 시스템은 두 대의 카메라에서 입력을 받아들이고, 연산한 결과를 외부 메모리에 저장하며 연산된 결과를 출력하기 위하여 총 227개의 데이터 pin이 필요하여, 필요한 레이아웃 면적보다도 많은 pin 수가 요구되어 그림 15(a)와 같이 큰 lead frame을 선택하였기에 빈 공간이 많이 발생하였다. 그림 15(a)의 전체 layout 크기는  $7,320\mu\text{m} \times 7,320\mu\text{m}$ 이고, 그림 15(b)의 본 논문에서 제안한 disparity 시스템의 실제 레이아웃 크기는  $3,310\mu\text{m} \times 3,250\mu\text{m}$ 이며, 제작된 package는 256PQFP를 사용하였다.

## V. 결론

본 논문에서 제안한 실시간 disparity 시스템은 하드웨어 크기를 작게 하기 위하여 수직축 strip 구조를 제안하여 30 frames/sec(15 frames의 변위 지도를 두 번 출력)의 실시간 동작이 가능함을 보여 주었다. 또한 하드웨어의 용량을 감소하기 위해 정합창 내의 영상의 유사도를 계산할 때 순차적으로 데이터를 입력받아 계산하는 방법을 사용하였고, 수식 (5)의 방법을 사용하여 소요된 하드웨어 크기를 작게 하였다. 실시간 검증을 위하여 Altera APEX20K1000EBC652-3 FPGA를 사용하여 제안한 방법의 타당성을 검증하였으며, 두 대의 카메라와 물체와의 거리 차이에 따른 변위 지도의 흑백 영상의 정보는 TV 화면을 통하여 확인하였다. 본 논문에서 제안한 방법을 이용하면 물체의 거리정보를 파악할 수 있기에 물체 인식, 움직임 추출, 제스처 인식 기술 등에 사용이 가능하다고 할 수 있다.

## 참고 문헌

- [1] R. C. Gonzalez and R. E. Woods, Digital Image Processing, Addison-Wesley, 1993.
- [2] 김형근, 안상철, 김남규, "정합 화소 수를 이용한 유사도 측정 방법 및 이를 구현하기 위한 장치", 대한민국 출원 특허, 출원번호:10-1998-013970, 1998.4.
- [3] 오승호, "순차병렬처리를 이용한 스테레오 변위 결정장치 및 방법", 대한민국 출원 특허, 출원번호:10-2001-0022680, 2001.4.
- [4] 노윤향, 고병철, 변혜란, 유지상, "웨이블릿 변환과 기준점을 이용한 변위 추정", 한국통신학회 논문지, Vol. 27, No. 2A, pp. 137-145, 2002.2.
- [5] 홍두일, 이장희, 오승호, 강봉순, 홍창희 "스테레오 영상을 이용한 실시간 Disparity Estimator의 설계", 대한전자공학회 CAD 및 VLSI 설계연구회 학술발표대회, pp. 232-237, 2002.5.
- [6] U. R. Dhond and J. K. Aggarwal, "Structure from Stereo-A Review", IEEE Trans. on Systems Man and Cybernetics, Vol. 19, No. 6, pp. 1489-1510, 1989.
- [7] K. Woo, D. Hong, S. Kwak, J. Kim and B. Kang, "Design and Implementation of Stereo Disparity Decision System", IDEC Conference 2002-Summer, pp. 181-182, 2002.8.
- [8] Samsung Electronics, KS0119 (Video Encoder) Datasheet, 2000.
- [9] Keith Jack, Video Demystified 3rd Edition, LLH Technology Publishing, 2001.
- [10] B. Grob and C. E. Herndon, Basic Television and Video Systems, Glencoe McGraw-Hill, 1999.
- [11] Samsung Electronics, KM718V895 (Synchronous SRAM) Datasheet, 1999.
- [12] Omnivision Corporation, OV7620A (CMOS Sensor Camera) Datasheet, 2000.
- [13] Altera Corporation, APEX20K1000EBC652-3 (FPGA Device) Datasheet, 1999.
- [14] Hynix(LG) Semiconductor, CB35OS152 (ASIC Library) Datasheet, 1998.3.

저 자 소 개



강 봉 순(정회원)  
 1985년 연세대학교  
 전자공학과 (공학사).  
 1987년 미국 University of Pennsylvania  
 전기공학과 (공학석사).  
 1990년 미국 Drexel University  
 전기 및 컴퓨터공학과  
 (공학박사).

1989년 12월 ~ 1999년 2월 삼성전자 반도체  
 수석연구원.

1999년 3월 ~ 현재 동아대학교  
 전기전자컴퓨터공학부 조교수.

<주관심분야: VLSI Design, ASIC Design>



양 훈 기(정회원)  
 1985년 연세대학교  
 전자공학과(공학사).

1987년 SUNY at Buffalo  
 전기 및 컴퓨터공학과  
 (공학석사).

1992년 SUNY at Buffalo 전기 및  
 컴퓨터공학과 (공학박사).

1993년 3월 ~ 현재: 광운대학교 전파공학과 교수.  
 <주관심분야: 무선통신시스템, UWB>