고밀도 3D 형상 계측 시스템에서의 고속 시차 추정을 위한 NCC 알고리즘 기반 하드웨어 구조

배경렬 · 권 순* · 이용환** · 이종훈* · 문병인†

A hardware architecture based on the NCC algorithm for fast disparity estimation in 3D shape measurement systems

Kyeong-ryeol Bae, Soon Kwon*, Yong-Hwan Lee**, Jong-Hun Lee*, and Byungin Moon†

Abstract

This paper proposes an efficient hardware architecture to estimate disparities between 2D images for generating 3D depth images in a stereo vision system. Stereo matching methods are classified into global and local methods. The local matching method uses the cost functions based on pixel windows such as SAD(sum of absolute difference), SSD(sum of squared difference) and NCC(normalized cross correlation). The NCC-based cost function is less susceptible to differences in noise and lighting condition between left and right images than the subtraction-based functions such as SAD and SSD, and for this reason, the NCC is preferred to the other functions. However, software-based implementations are not adequate for the NCC-based real-time stereo matching, due to its numerous complex operations. Therefore, we propose a fast pipelined hardware architecture suitable for real-time operations of the NCC function. By adopting a block-based box-filtering scheme to perform NCC operations in parallel, the proposed architecture improves processing speed compared with the previous researches. In this architecture, it takes almost the same number of cycles to process all the pixels, irrespective of the window size. Also, the simulation results show that its disparity estimation has low error rate.

Key Words : NCC, disparity, stereo matching, block-based, box filtering

1. 서 론

고밀도 3D 형상 정보를 획득하는 센서 시스템은 크게 3D 레이저 스캐너, TOF(time-of-flight) 등과 같이 능동소자를 사용한 시스템과 스테레오 비전과 같은 수동소자를 사용한 시스템으로 나누어진다(Fig. 1). 능동소자를 사용한 센서 시스템은 물체나 환경에 직접 빛 또는 초음파 등을 발산하여 반사되는 형상을 인식하는 구조^[1]이며, 측정결과와 정확도가 높으나 센서 자체의 가격이 고가이며 능동발산에 따른 대상의 제약이 따르고

측정 가능 거리 또한 한계가 있다. 반면 수동소자 기반의 센서시스템은 일반적으로 스테레오 비전과 같이 이미지 센서를 사용하여 대상으로부터 반사되는 시각정보를 획득한 후 컴퓨터비전 기반의 정합 알고리즘을 통해 3D 환경을 추정함으로써 능동센서에 비해 정확도는 떨어지나 시스템 구현비용이 저렴하고 적용 대상, 거리 등의 제약이 상대적으로 자유로운 장점이 있다.

스테레오 비전을 기반으로 하는 3D 형상계측 시스템은 Fig. 2와 같이 두 개의 이미지 센서를 사용하여

경북대학교 전자전기컴퓨터학부(School of Electrical Engineering & Computer Science, Kyungpook National University)
*대구경북과학기술원(Daegu Gyeongbuk Institute of Science & Technology)
**금오공과대학교 전자공학부(School of Electronic Engineering, Kumoh National Institute of Technology)

†Corresponding author : bihmoon@knu.ac.kr
(Received : October 27, 2009, Revised : January 25, 2010
Accepted : March 8, 2010)

Fig. 1. Sensor systems for 3D distance measurement.

Fig. 2. Distance estimation model of the stereo vision.

공간상의 동일 지점에 대해 양쪽 이미지 센서의 영상 평면(image plane)에 맺힌 화소(pixel)간의 시차(disparity)를 계산하고, 이를 이용하여 이미지 센서와 대상 간의 거리 정보를 구한다^[2].

시차 추정을 위한 스테레오 영상 정합 법은 컴퓨터 비전 영역에서 오랫동안 많은 연구가 이루어져 왔으며, 크게 전역 정합(global matching)과 지역 정합(local matching)으로 구분할 수 있다^[3,4]. 그 중 실시간 스테레오 비전 계측 시스템에 적용된 사례를 살펴보면 과거에는 하드웨어 성능의 한계로 인해 SAD(sum of absolute difference) 또는 SSD(sum of squared difference)와 같이 간단한 차(difference) 기반 비용함수를 사용한 지역정합 기반 스테레오 비전 시스템에 대한 연구들이 이루어졌다^[2,5-7]. 그러나 이러한 시스템들은 비용 함수의 연산이 간단한 대신에 좌·우 카메라의 영상오차에 따른 시차추정의 오차율이 높아 신뢰성이 낮은 단점이 있다. 이를 극복하기 위해 지역탐색 기반 스테레오 정합방법에 있어 영상의 밝기정보를 단순 비교하는 비용함수를 획득된 이미지에 강인한 특성을 지닌 방법으로 개선한 기법들이 제안되어 왔다. Census transform, NCC(normalized cross correlation)와 같은 영역의 상관 기반 정합알고리즘의 경우 일반적으로 발생 가능한 좌·우 이미지 센서에서 획득하는 센서의 밝기 차와 같은 오류에 강인성을 보인다.

NCC 기반 지역 정합은 좌·우 영상의 조명 환경(lighting condition) 및 잡음(noise) 차에 강인하지만, 연산 복잡도가 높은 특성을 가지고 있으므로, 기존의 NCC 알고리

즘을 기반으로 한 연구들은 알고리즘의 불필요한 연산의 최소화를 통해 동작 속도의 향상을 시도해 왔다. L. Di Stefano 등은 BPC(bounded partial correlation) 기법을 사용하여 불필요한 NCC 값 계산을 최소화 하였고^[8-10], Shou-Der Wei 등은 winner update 기법을 다단계(multi-level)로 적용함으로써, 불필요한 NCC 값 계산을 최소화 하였다^[11]. 또 다른 방법으로, D. M. Tsai 등은 NCC 값 계산 중 가장 많은 연산을 필요로 하는 부분에 합 표(sum table)를 사용함으로써 동작속도를 빠르게 하였다^[12]. 그러나 이러한 기존 연구들은 하드웨어 구현 측면이 아닌 알고리즘적인 연산 방법을 제한한 것으로서, 하드웨어를 통한 실시간 구현에 대한 연구로서는 미흡하다는 한계를 가지고 있다.

따라서 본 논문에서는 실시간 스테레오 비전 시스템의 구현을 위해, NCC 기법을 활용하여 시차 결과의 오차율이 비교적 낮고 좌·우 영상 간 잡음 및 조명 차에 강인하며, 실시간 정합 비용 계산이 가능한 파이프라인 형태의 FNCC(fast NCC) 하드웨어 구조를 제안한다. 본 논문의 2장에서는 NCC 알고리즘에 대해 간단히 살펴보고, 3장에서는 하드웨어 구조의 기본이 되는 블록 기반(block-based) 박스 필터링(box filtering)과 이를 사용한 파이프라인 하드웨어 구조를 제안한다. 4장에서는 제안된 FNCC구조를 사용한 스테레오 정합기의 시차추정 오차 성능을 분석하고, 마지막으로 5장에서 본 논문의 결론을 맺는다.

2. NCC 기반 스테레오 정합 알고리즘

스테레오 정합 과정은 Fig. 3과 같이 이미지 교정(rectification and calibration^[13]) 단계를 거친 좌·우 영상에 대해 동일한 에피폴라 라인(epipolar line) 상에서 탐색 과정을 거친 후 정합 유사도가 가장 높은 지점에서의 시차를 얻게 된다. 따라서 정합에 사용되는 비용 연산함수에 따라 추정 정확도가 다른데 일반적으로 차 기반 함수와 상관(correlation) 기반 함수로 나눌 수 있다. 차 기반 함수로는 SAD와 SSD 등이 있고, 상관 기반 함수로는 NC(normalized correlation), NCC, Census transform 등이 있다. 차 기반 함수는 화소 단위 정합이 가능하나, 실영상에서 영상의 잡음이 많은 경우 또는 물체의 경계영역에서 정합성능을 높이기 위해서, 윈도우 기반의 비교 정합을 주로 사용하게 된다. 조명 환경이 열악하거나 밝기 오프셋이 존재하는 영상의 경우 상관 기반의 정합 방법이 차 기반 정합 방법에 비해 상대적으로 우수한 결과를 보인다.

본 논문에서 사용하려는 NCC 기반의 비용 연산함

Fig. 3. Stereo matching of left and right images.

수는 스테레오 정합에 사용될 경우 두 영상간의 선형적인 밝기 변화나 영상왜곡 등에 강인한 특성을 보인다. NCC 알고리즘의 일반적인 식은 다음과 같다.

$$NCC(d) = \frac{Cov[I_L(x, y), I_R(x + d, y)]}{\sqrt{Var_{I_L(x, y)} Var_{I_R(x + d, y)}}} \quad (1)$$

NCC 알고리즘은 식 (1)과 같이 좌 · 우 영상의 화소값의 공분산(covariance)을 분자로 하고 각 화소의 표준편차(standard deviation)를 분모로 한다. 이때 표준편차는 분산(variance)의 제곱근 값을 취한다. 이를 통해 영상의 밝기 상호 관련도(brightness correlation)를 비교하여 유사도가 가장 큰 값을 찾는 것이다.

좌 영상 기반의 스테레오 정합 과정의 경우 수식 (1)에서 동일한 에피폴라 라인 상에 놓인 두 윈도우 영역에 대해 우 영상의 시차범위 $[0, d]$ 에 대해 각각의 NCC 결과를 비교하게 된다. NCC 결과의 범위는 $[-1, 1]$ 을 가지게 되나 스테레오 정합과정에서는 역 상관관계는 제외하여 $[0, 1]$ 의 범위를 그 대상으로 하게 된다.

$$ZNCC(d) = \frac{\sum_{j=1}^{N-1} \sum_{i=1}^{M-1} [R(x+d+i, y+j) - \mu(R(x+d, y))] \times [L(x+i, y+j) - \mu(L(x, y))]}{\sqrt{\sum_{j=1}^{N-1} \sum_{i=1}^{M-1} [R(x+d+i, y+j) - \mu(R(x+d, y))]^2} \times \sqrt{\sum_{j=1}^{N-1} \sum_{i=1}^{M-1} [L(x+i, y+j) - \mu(L(x, y))]^2}} \quad (2)$$

$$ZNCC(d) = \frac{\sum_{j=1}^{N-1} \sum_{i=1}^{M-1} R(x+d+i, y+j) \times L(x+i, y+j)}{\sqrt{\sum_{j=1}^{N-1} \sum_{i=1}^{M-1} R^2(x+d+i, y+j) - M \times N \times \mu(R(x+d, y))^2} \times \sqrt{\sum_{j=1}^{N-1} \sum_{i=1}^{M-1} L^2(x+i, y+j) - M \times N \times \mu(L(x, y))^2}} \quad (3)$$

본 논문에서는 이와 같은 NCC 알고리즘을 사용하되 화소 값을 그대로 사용하는 것이 아닌 화소 값에서 그 윈도우 전체의 평균값을 뺀 값을 사용하는 식 (2)의 ZNCC(zero-mean NCC) 알고리즘을 사용한다^[14]. 여기서 x, y 는 이미지에서의 좌표를 뜻하고 d 는 시차, L 과 R 은 각각 왼쪽 이미지, 오른쪽 이미지의 화소를 의미한다. 그리고 M 과 N 은 윈도우의 가로, 세로 길이를 뜻하며 μ 는 윈도우 전체의 평균값을 나타낸다. 그리고 식 (2)를 계산이 용이하게 풀어쓰면 식 (3)과 같이 된다^[15]. 최종적으로 식 (3)의 간략화 된 ZNCC 함수를 사용하여 왼쪽 이미지 I_L 에 대한 정합 점 I_R 을 찾으려 한다.

3. 블록 기반의 박스 필터링을 적용한 ZNCC 파이프라인 하드웨어 구조

본 장에서는 ZNCC 시스템 동작 구조에 대해 설명하고 연산 속도 증가를 위한 블록 기반의 박스 필터링 기법에 대해 설명하며, 이를 적용한 파이프라인 형태의 ZNCC 하드웨어 구조에 대해 설명한다.

3.1. ZNCC 시스템 동작 구조

식 (3)의 ZNCC 연산구조를 분석해 보면 양 이미지의 상관정도를 연산하는 분자 부(numerator) 및 비교하고자 하는 영상 L, R 에 대한 제곱의 합(squared sum), 그리고 그것의 곱에 대한 제곱근을 적용한 분모 부(denominator)로 이루어져 있다. 그리고 각 분자 및 분모부에는 정규화를 위해 L 및 R 에 대한 평균의 차가 적용되어 있다. Fig. 4는 양 이미지에서 화소 값들을 불러와 ZNCC 결과 값을 출력하는 과정을 나타내었다. 이와 같은 구조에서 연산 량이 많이 요구되는 부분은 L 및 R 영역의 평균을 구하는 연산 부, 영상 L, R 의 곱의 합(이하 $R \times L$), 영상 L, R 에 대한 제곱의 합, 분자 및 분모의 제곱과 곱을 위한 곱셈, 분모의 제곱근 연산 그리고 분자와 분모를 나누는 나눗셈을 들 수 있다. 이들을 하드웨어로 구현하려면 많은 하드웨어 비용 및 시간을 필요로 한다. 이러한 연산은 한번만 이루어지고 끝나는 것이 아니라 L, R 영상 내에서 동일한 크기의 윈도우 영역을 순차적으로 이동해 가며 계속 비교가 이루어지는 구조가 되는데 이때의 비교 횟수는 참조(reference) 윈도우가 형성될 수 있는 전체 경우의 수, 즉 이미지 크기만큼의 경우의 수가 생기며 각각의 참조 윈도우마다 비교하는 대상이 되는 비교(searching) 윈도우가 최대 시차 수만큼 생기므로(이미지의 가로길이) $\times$ (이미지의 세로길이) $\times$ (최대 시차) 만큼의 비교 횟수가 생긴다. 더구나 일반적으로 카메라로부터 입

Fig. 4. A basic system structure for ZNCC calculation.

력되는 영상은 초당 수십 프레임 이므로 실시간으로 들어오는 영상을 빠짐없이 ZNCC 연산과정을 통해 처리하기 위해서는 상기 수식에 초당 프레임수를 곱한 처리 속도를 가져야 한다.

본 논문에서 제안된 구조는 식 (3)의 복잡한 연산을 블록 기반의 박스 필터링을 사용하고 파이프라인을 적용하여 빠른 연산이 가능하도록 하였다.

3.2. 블록 기반 박스 필터링

식 (3)을 보면 분자식에는 윈도우 전체에서 왼쪽 이미지의 화소와 오른쪽 이미지의 화소 값을 곱하여 모두 더하는 부분이 있고, 분모 식에는 윈도우 전체의 값을 제곱하여 모두 더하는 부분이 있다. 그리고 추가적으로 평균값을 구할 때 역시 윈도우 전체의 값을 모두 더하여 윈도우 전체 크기로 나눠 주어야 한다.

이때, 만약 윈도우의 크기가 작아서 윈도우 전체의 값을 더해도 시간과 비용에 별 무리가 없는 경우에는 상관없이 윈도우의 크기가 큰 상황에서 윈도우가 이동할 때 마다 윈도우 전체의 값을 더하려면 많은 하드웨어 비용과 시간이 소요된다. 이 같은 문제점을 해결하기 위해 본 논문에서는 2D 데이터 배열에서 일정한 크기의 윈도우의 합을 윈도우 크기에 상관없이 고속으로 구할 수 있는 방법인 박스 필터링 기법^[16]을 사용하되, 이를 하나의 화소가 아닌 여러 화소에 대해 동시에 적용함으로써 더욱 빠른 연산속도와 중복되는 연산량을 최소화 하였다. Fig. 5는 인접한 $M \times N$ 개의 화소들을 중심으로 한 $M \times N$ 개의 윈도우를 하나의 블록으로 묶는 방법을 나타내고 있고 Fig. 6은 블록을 2D 데이터 배열에 적용하여 나타내었다. 본 논문에서는 이러한 윈도우($K \times L$) 여러 개($M \times N$)를 하나의 블록으로 묶어 동시에 계산함으로써 연산속도를 $M \times N$ 배

Fig. 5. Basic example of block-based box filtering.

Fig. 6. Block-based box filtering applied to a 2D data array.

향상시키는 방법을 제안한다.

기본적인 방법은 박스 필터링 기법을 그대로 적용하

Fig. 7. Block-based box-filtering: (a) calculation of the first block, (b) horizontal movement of the first block, (c) vertical block movement, (d) horizontal block movement in the other area.

되 윈도우 기반이 아닌 블록 기반이므로, 윈도우를 이동하는 것이 아니라 블록을 이동해 가면서 기존의 합에서 추가로 들어오는 화소 값들은 더 하고, 나가는 화소 값들은 빼 주어 전체 합을 구하는 방식이다. 이때, 블록의 크기는 $(K+M-1) \times (L+N-1)$ 이 된다. 각각의 윈도우의 가로 합을 블록 좌측에 저장하고 가로 합을 다 더한 전체 합은 블록의 좌측 상단에 저장 한다.

3.2.1. 첫 번째 블록의 계산 (Fig. 7-(a))

Fig. 7은 블록 기반의 박스 필터링 기법을 블록의 이동에 따라 설명한 그림이다. 첫 번째(맨 첫줄, 맨 좌측) 블록은 이전에 계산해 둔 가로줄의 합(H), 전체 합(S)이 없으므로 블록 전체를 더해 주어야 한다. 먼저 각 윈도우 별로 가로줄의 합을 먼저 구한 후 다시 구해진 가로줄의 합을 모두 더하여 전체 합을 구한다. 물론 세로줄의 합을 먼저 구해도 상관없다. 가로 합인 경우 식 (4)와 같이 윈도우 안의 모든 합을 더하여 저장한다.

$$H_{ij} = \sum_{k=i}^{i+M-1} P_{k,j} \quad (4)$$

i 는 열 번호, j 는 행 번호를 나타낸다. 이때 2M번째 부터 (M+K)번째까지의 화소는 공통적으로 적용되는 값들의 합이므로 한 번의 연산을 통해 구한 후 동일 열에 속한 화소들의 가로 합을 구하는데 재사용할 수 있다. 이와 같이 구한 $M \times (L+N-1)$ 개의 가로 합을 식 (5)와 같이 세로로 더하여 전체 합을 구한다.

$$S_{ij} = \sum_{l=j}^{j+N-1} H_{i,l} \quad (5)$$

역시 마찬가지로 2N번째부터 (N+L)번째까지의 화소는 한 번의 연산을 통해 구한 후 재사용 한다.

3.2.2. 첫 번째 블록의 가로 이동 (Fig. 7-(b))

블록을 가로로 이동시킬 때에는 $M \times N$ 개의 화소에 대한 윈도우 모두 계산이 끝났으므로 가로로 블록을

Fig. 8. Proposed hardware architecture based on the block-based box-filtering for ZNCC and disparity calculation.

M만큼 이동 시킨다. 이렇게 이동시킨 블록은 이동하기 전 블록과 가로로 $(K-1)$ 만큼 중복 구간이 생기며, 추가로 M행만큼 새로운 구간이 늘어나고 M행만큼 빠져나가는 구간이 생긴다. 이 중복되는 구간을 다시 계산하지 않도록 하기 위해 식 (6)과 같이 앞의 블록에서 계산해 놓은 가로줄의 합에서 추가로 들어온 만큼의 화소 값들을 더해주고 빠져나간 만큼의 화소 값들을 빼주면 적은 연산량으로도 가로 합을 구할 수 있다. 이렇게 구한 가로 합을 식 (5)를 통해 모두 더하면 전체 합이 구해진다.

$$H_{ij} = H_{i-Mj} - \sum_{k=i}^{i+M-1} P_{kj} + \sum_{k=i+K}^{i+K+M-1} P_{kj} \quad (6)$$

3.2.3. 블록의 세로 이동 (Fig. 7-(c))

블록의 가로 이동이 끝나고 세로로 이동시킬 때에도 마찬가지로 방향만 다를 뿐, 가로로 적용한 방법을 똑같이 적용할 수 있다. 세로로 블록을 N만큼 이동 시키면 이동하기 전 블록과 세로로 $(L-1)$ 만큼 중복되는 구간이

Fig. 9. Hardware architecture of the average module.

생기므로 추가된 화소들의 가로 합을 식 (4)를 이용하여 계산하고 식 (7)을 이용하여 이전에 계산해 놓은 중간 값과 전체 합을 이용하여 전체 합을 구할 수 있다.

$$S_{ij} = S_{ij-N} - \sum_{l=j}^{j+N-1} H_{il} + \sum_{l=j+L}^{j+L+N-1} H_{il} \quad (7)$$

3.2.4. 나머지 영역에서의 블록의 가로 이동 (Fig. 7-(d))

첫 번째 줄이 아닐 경우 블록의 가로 이동은 식 (6)과 (7)을 이용하여 가로, 세로 모두 이전의 중간 값과 전체 합을 이용할 수 있다. 블록이 가로로 이동하면서 앞의 블록과도 중복되는 구간이 생기지만 위의 블록과도 중복되는 구간이 생겨 이전의 중간 값과 전체 합을 이용할 수 있다.

Fig. 10. Hardware architecture of the square sum module.

3.3. 블록 기반의 박스 필터링을 적용한 ZNCC 파이프라인 하드웨어 구조

Fig. 8은 블록 기반의 박스 필터링을 적용한 ZNCC 파이프라인 하드웨어 구조를 블록 다이어그램으로 나타낸 것이다. 8단계의 파이프라인으로 구성되며 한 클록마다 N개의 결과를 출력하는 구조이다. 제안된 ZNCC 하드웨어 구조에서는 평균을 구하는 부분, 제곱의 합을 구하는 부분, $R \times L$ 을 구하는 부분, 이들을 이용하여 NCC 결과 값을 구하고 최대값을 찾아 이때의 시차를 저장하는 부분, 이렇게 4부분으로 나뉜다.

Fig. 9는 평균값 계산 모듈의 구조를 나타낸 그림이며, Fig. 10은 제곱의 합 계산 모듈, Fig. 11은 $R \times L$ 의 합 계산 모듈을 나타낸 그림이다. 그리고 Fig. 12는 ZNCC 결과 값 계산 및 최대 정합 점 시차 추정 모듈을 나타낸 그림이다. 그림에서 회색 블록들은 중간 결과 값을 저장하는 레지스터를 나타내며 $R \times L$ 의 합 계산, 제곱의 합 계산, 평균값 계산 모듈은 병렬적으로

Fig. 11. Hardware architecture of the $R \times L$ sum module.

Fig. 12. Hardware architecture of the module to calculate ZNCC results and find the disparities of matching points.

한꺼번에 동작한다.

참조 윈도우가 되는 왼쪽 이미지의 윈도우로 구성된 블록(이하: 왼쪽 블록)의 분모 값과 평균값 계산을 먼저 하고 비교 윈도우가 되는 전체 시차 수만큼의 오른쪽 이미지의 윈도우로 구성된 블록(이하: 오른쪽 블록)의 분모 값과 평균값, 그리고 분자 값을 동시에 계산한다. 결론적으로 하나의 참조 윈도우에 대해 모든 시차 수만큼의 비교 윈도우와의 비교를 위해 총(시차 수+1)회의 연산 과정을 거치게 된다.

평균을 구하기 위해 박스 필터링 방법을 사용하여 세로 합을 먼저 구하여 저장하고 이들을 다시 더하여 전체 합을 저장한다. 그리고 제공의 세로 합과 전체 합, $R \times L$ 의 세로 합과 전체 합 역시 마찬가지로 박스 필터링 방법을 사용하여 저장한다.

제안된 구조에 블록 기반의 박스 필터링을 적용 하면서 가로와 세로 모두 적용 시 많은 메모리를 필요로 한다는 문제로 인해 세로로는 적용하지 않고 가로로만 적용하여 구현 하였다. 블록 기반의 박스 필터링을 적용하기 위해서는 앞에서 연산 해놓은 중간 값들과 결과 값들을 저장해 놓을 추가적인 메모리를 필요로 한다. 평균값과 제공의 합을 위해 블록 기반의 박스 필터링을 적용하는데 있어서 메모리 사용량이 그리 크지 않지만 $R \times L$ 은 같은 시차를 가지는 블록사이에서만 중복 되는 구간이 발생하므로 이를 위해 블록 기반의 박스 필터링을 적용하기 위해서는 시차 수만큼 배가 된 메모리를 필요로 하기 때문이다.

가로로만 블록 기반의 박스 필터링을 사용 한다고 했을 때, 각 행의 맨 좌측의 왼쪽 블록과 그에 따른 오른쪽 블록들은 블록내의 모든 열을 다 계산해야 하므로 블록 당 블록의 가로 길이($K+M-1$)만큼의 클록 사이클이 필요하고 나머지 블록들에 대해서는 추가로 들어오는 화소들만 계산하면 되므로 M 클록 사이클이 필요하다. 각 행의 맨 좌측의 블록들의 전체 사이클 타임을 생각해 보면 평균값 계산을 위해 매 클록마다 N 개의 세로의 평균을 구하여($K+M-1$)클록 후에 총($K+M-1$) $\times N$ 개의 세로의 평균들을 구하고 이들을 이용하여 바로 다음 클록에 $M \times N$ 개의 평균값을 구한다. 그리고 다음 클록에 $M \times N$ 개의 평균의 제공과 양 블록의 평균값들의 곱을 계산한다. 마찬가지로 제공의 합과 $R \times L$ 역시 이미지 데이터를 불러와 제공하고 매 클록마다 $R \times L$ 을 계산한 뒤 N 개의 세로 합을 구하여($K+M$) 클록 후에 총($K+M-1$) $\times N$ 개의 세로 합이 구할 수 있고 그 다음 클록에 $M \times N$ 개의 제공의 합을 구한다.

그리고 이렇게 구해진 평균과 제공의 합, $R \times L$ 의 합

Fig. 13. Input images(a) and ground-truth disparity maps of input images(b).

Table 1. Detailed information of input images

이미지	크기	최대 시차	시차 스케일
1	384 × 288	16	16
2	434 × 383	32	8
3	284 × 216	32	8
4	434 × 380	32	8

을 이용하여 NCC 결과 값을 구하고 최대값을 찾아 이때의 시차를 저장하며 결과적으로 매 클록마다 N 개의 NCC 결과 값을 구할 수 있다.

4. 시스템 설계 및 성능 분석

4장에서는 시스템 설계 및 성능 분석을 위해 세부적인 정밀도 결정을 위한 오차 분석과 제안된 구조를 C로 모델링 하여 다른 여러 가지 구조와 성능을 비교 분석한다.

4.1. 성능 분석 환경

본 논문에서 제안한 구조에서 세부적인 정밀도를 결정할 때 실험을 통하여 하드웨어 비용 대비 가장 성능이 좋고 오차율이 낮은 것을 찾는 방법을 사용하였다. 본 절에서는 실험에 사용된 영상과 실험을 위한 배경에 대해 언급한다. Fig. 13은 Middlebury 대학 연구소^[17]에서 제공하는 스테레오 정합을 위한 테스트 이미지들과 기준(ground-truth) 시차 맵을 나타내고 있다. 각 시차 이미지는 화소마다의 시차정보를 스케일링하여 0-255 이내의 밝기정보로 변환하여 표현된 것이다. Table 1은 각 이미지들의 상세 정보를 나타내고 있으며 실험 결과에서 나타내는 시차 오차율은 모두 이 시차 맵을 기준으로 얼마만큼 오차가 있는가를 나타낸다. Table 1에서 크기는 각 이미지의 크기를 나타내고, 시차 스케일은 출력되는 시차 값 당 화소의 그레이(gray)값 차이를 나타낸다.

Fig. 14. Disparity error rates depending on the window size.

4.2. 제안된 ZNCC 하드웨어 모듈의 시차추정 오차율 평가

본 절에서는 제안된 구조의 윈도우 크기 결정, 세부적인 정밀도 결정을 위한 오차율을 분석한다.

4.2.1. 윈도우 크기에 따른 오차율 비교

윈도우 크기는 대표하는 화소의 특징을 나타내기 위해 얼마만큼의 주변 화소들까지 계산에 포함시킬 것인가를 나타내는 중요한 요소이다. 크기가 작으면 연산량은 줄어들지만 잡음에 쉽게 영향을 받고 크기가 크면 연산량도 늘어날 뿐더러 너무 넓은 범위를 다 고려

하게 되어 특정 화소들의 특징이 잘 드러나지 않게 된다. Fig. 14에서는 결과를 8비트로 하였을 경우 윈도우 크기에 따른 시차 오차율을 보여준다. 시차 오차율이란 기준 시차맵과의 비교 결과 시차가 1 이상 나는 화소들의 비율을 의미한다.

각 이미지마다 최적의 윈도우 크기가 다 다른 듯 보이지만 Table 2에서 보는 것처럼 윈도우 크기가 작으면 작을수록 잡음이 많고 윈도우가 크면 클수록 그림이 일그러지는 것을 볼 수 있다. 본 논문에서 제안한 구조에서는 잡음의 영향과 일그러짐 현상모두를 고려하여 21×21 크기를 사용하였다.

4.2.2. 최종 ZNCC 결과 값의 비트 수 결정

본 절에서 언급할 내용은 최종 ZNCC 결과 값의 비트 수 결정을 위한 각 비트 수 별 시차 오차율, 제공된 계산을 위한 참조 표의 입출력 비트 수 결정에 대한 내용을 언급한다. 앞에서 언급한 바와 같이 본 논문에서 제안한 구조의 ZNCC 값은 항상 0에서 1사이의 값을 가진다. 이 값의 범위의 설정은 성능 및 비용의 상반관계(trade off)에 큰 영향을 미친다. 제안한 구조에서 최종적으로 출력되는 ZNCC 결과 값의 비트 수에 따른 시차 오차율을 Fig. 15로 나타내었다.

분석 결과를 보면 비트 수가 증가하면서 오차율의 감소정도도 점점 줄어드는 것을 알 수 있다. 8비트 이

Table 2. Output images for different window sizes

이미지	3×3	5×5	7×7	15×15	21×21	35×35
1						
2						
3						
4

Fig. 15. Disparity error rates depending on the bit width of ZNCC results.

상의 경우에는 오차율의 변화가 거의 없었다. 또한 최종 결과를 8비트로 정하면 오차율이 낮은 면도 있지만 바이트 단위로 데이터를 표현할 수 있어 데이터 전송 및 처리가 용이하다.

4.2.3. 참조 표의 크기 결정

분모 값 계산을 위한 과정 중 제곱근 연산을 위하여 참조 표를(look-up table)사용하였다. 제곱근을 위한 참조 표의 크기를 정하는데 있어서 고려해야 할 사항은 입력되는 값과 출력되는 값의 비트 수이다. 입력되는 값의 범위가 작으면 오차가 커지게 되고 입력되는 값의 범위가 크면 그만큼 경우의 수가 많아지게 되므로 참조 표의 크기가 커지게 된다.

이에 본 논문에서는 10비트 입력, 8비트 출력을 가지는 참조 표를 사용하였다. 입력은 10비트 정수형으로 받고 출력은 소수 3째 자리까지 표현하도록 하였다. 이 같은 방법을 사용하면 입력되는 값과 출력되는 값의 영점이 달라 추가로 보정해 주어야 하지만 소수 3째 자리까지 표현하므로 오차율도 적고 참조 표의 크기도 구현 가능한 크기로 제한 할 수 있다.

Table 3은 참조 표 사용 유무에 따른 결과 값의 오차율을 비교한 표이다.

Table 3에서 참조 표 사용 란은 참조 표를 사용하여 계산했을 때 출력되는 ZNCC 값들의 평균값을 나타낸 것이고 미사용은 참조 표를 사용하지 않고 계산했을 때 출력되는 ZNCC 값들의 평균값을 나타낸 것이다. 그리고 ZNCC 오차율은 식 (8)을 이용하여 모든 점에 대한 결과 값의 오차율을 평균 낸 값이다.

$$\frac{(\text{참조표사용값}) - (\text{참조표미사용값})}{(\text{참조표미사용값})} \times 100 \quad (8)$$

Table 4에서는 제안된 참조 표를 사용했을 때의 시차 오차율과 사용하지 않았을 때의 시차 오차율, 그리

Table 3. ZNCC average values and average ZNCC error rates when using the look-up table with a 10-bit input and an 8-bit output

이미지	참조 표 사용 시 평균 ZNCC 결과 값	참조 표 미사용 시 평균 ZNCC 결과 값	평균 ZNCC 오차율
1	121.7	121.1	0.68%
2	181.9	180.4	1.09%
3	212.2	211.7	0.32%
4	177.9	177.1	0.82%

Table 4. Disparity error rates: (a) disparity error rates with the look-up table, (b) disparity error rates without the look-up table, (c) disparity error rates between the two disparity maps

이미지	(a)	(b)	(c)
1	19%	19%	1%
2	14%	13%	0%
3	20%	20%	0%
4	15%	15%	0%

Table 5. Disparity maps with and without the look-up table

이미지	참조 표 사용	참조 표 미사용
1		
2		
3		
4

Fig. 16. Required clock frequency for 30-fps processing according to the image size and the maximum disparity.

고 두 시차 맵의 오차율을 비교해 보았다. 그리고 이때의 출력 이미지들을 Table 5에 나타내었다. 참조 표의 사용 유무에 따른 결과 값의 차이가 아주 작고 둘 사이의 오차율 또한 낮았다. 이런 결과로부터, 참조 표를 사용했을 때 기준 시차맵에 대한 높은 오차율은 참조 표 사용 때문이 아니며, ZNCC 알고리즘 자체의 한계임을 알 수 있다. 결론적으로 참조 표 사용 결과 여부는 시차맵 결과에 거의 영향을 주지 않음을 보여 준다.

4.3. 속도 측면의 성능 분석

본 논문에서 제안하는 구조의 성능 분석을 위해 몇 가지 구조와 클럭 사이클과 이때의 하드웨어 비용을 비교해 보았다. 제안된 구조와의 비교를 위하여 기존의 윈도우 기반의 박스 필터링을 사용한 구조와 박스 필터링을 사용하지 않는 블록 기반의 ZNCC 연산 구조를 추가로 모델링하여 비교해 보았다.

첫 번째 구조는 박스 필터링을 사용하지 않고 매번 윈도우 전체를 더하지만 여러 화소의 ZNCC 값을 동시에 병렬적으로 구하는 구조(구조 1)^[2]이며 두 번째 구조는 블록 기반의 박스 필터링이 아닌 기존의 박스 필터링을 사용한 구조(구조 2)^[16]이다. 이러한 구조는 제안된 블록 기반의 박스 필터링을 사용한 구조(구조 3)와 전체 구조에서 크게 다르지 않으며 다만 블록 기반이 아니므로 한 번에 하나의 윈도우에 대해서만 실행한다.

동일한 환경에서 성능 비교를 위해 윈도우 크기는 21×21, 윈도우 1×16개를 하나의 블록으로 지정하고 최종 결과 값은 8비트로 지정하였다. 그리고 참조 표 역시 제안된 구조와 동일하게 적용하였고 모든 단계는 한 클럭에 실행이 된다고 가정하였다. Fig. 16는 영상의 크기와 최대 시차에 따른 각 구조별 동작 주파수를

비교한 그림이다. 초당 30 frame의 출력을 위한 동작 주파수를 나타낸다. 이 결과에서 알 수 있듯이 제안된 구조는 구조 1보다는 약 20배, 구조 2보다는 약 17배 빠른 성능을 보였다. 제안된 구조는 다른 구조와 달리 VGA(640×480) 크기의 이미지에 최대 시차 64 적용 시 30 MHz 정도의 낮은 주파수에서도 초당 30 프레임의 출력을 확인 할 수 있다.

5. 결 론

본 논문에서는 스테레오 비전을 기반으로 하는 고밀도 3D 형상 계측 시스템의 응용에 있어 적용 가능한 ZNCC 기반 정합도 분석 알고리즘의 빠른 연산을 위한 동작 구조에 대하여 기술하고 이에 대한 하드웨어 파이프라인 구조를 기술 하였다. 또한 이를 C언어 기반으로 모델링 하여 다양한 시뮬레이션을 거쳐 구조의 세부적인 정밀도를 결정하였고, 속도 측면에서의 성능을 측정하여 다른 구조와 비교 평가 하였다.

비트 수에 따른 시차 오차율 비교를 위한 시뮬레이션을 통해 최종 결과 값의 비트 수가 점점 늘어날수록 그에 따른 시차 오차율이 점점 감소하지만 그에 따라 감소율도 점점 줄어드는 것을 확인하고 최종 비트 수를 결정 하였고, 윈도우 크기에 따른 시차 오차율 비교를 위한 시뮬레이션을 통해 윈도우 크기가 작으면 작을수록 잡음에 영향을 많이 받고 윈도우 크기가 클수록 특정 화소들의 특성이 잘 드러나지 않고 일그러짐 현상이 일어나는 것을 확인하여, 윈도우 크기를 결정 하였다. 또한 여러 파라미터를 사용한 시뮬레이션을 통해 제공된 연산을 위한 참조 표의 크기와 구조를 결정 하였다.

속도 측면에서 성능 측정 및 평가 결과 제안된 하드웨어 구조는 실제 하드웨어로 구현이 가능한 다른 구조들보다 동작 속도가 훨씬 빠른 특징을 나타내었다. ZNCC 기법의 많은 연산 량을 박스 필터링 기법을 적용하여 윈도우 크기에 상관없이 빠르게 연산 할 수 있도록 하였고 또한 이를 블록 기반으로 여러 화소에 대해 병렬적으로 적용함으로써 속도를 더욱 향상시켰다.

제안된 하드웨어 구조는 스테레오 비전뿐만 아니라 실시간 환경 인식기술이 사용되는 다양한 시스템에 적용될 수 있을 것으로 기대된다.

감사의 글

본 연구는 교육과학기술부 기관고유사업 연구비 지원에 의해 수행되었습니다.

참고 문헌

- [1] 김종만, 도용태 “이동 물체의 3차원 계측을 위한 PSD 센서 배열 설계”, 센서학회지, 제17권, 제2호, pp. 106-113, 2008.
- [2] S. Perri, D. Colonna, P. Zicari, and P. Corsonello, “SAD-based stereo matching circuit for FPGAs”, *Proc. of the 13th IEEE International Conference on Electronics, Circuits, and Systems*, pp. 846-849, Nice, France, 2006.
- [3] L. Di Stefano, M. Marchionni, and S. Mattoccia “A fast area-based stereo matching algorithm”, *Image and Vision Computing*, vol. 22, no. 12, pp. 983-1005, 2004.
- [4] D. Scharstein and R. Szeliski, “A taxonomy and evaluation of dense two-frame stereo correspondence algorithms”, *International Journal of Computer Vision*, vol. 47 (April-June), pp. 7-42, 2002.
- [5] C. Geogloulas, L. Kotoulas, G. Ch. Sirakoulis, I. Andreadis, and A. Gasteratos, “Real-time disparity map computation module”, *Microprocessors and Microsystems*, vol. 32, no. 3, pp. 159-170, 2008.
- [6] M. Hariyama, N. Yokoyama, and M. Kameyama, “FPGA implementation of a stereo matching processor based on window-parallel-and-pixel-parallel architecture”, *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E88-A, no. 12 pp. 3516-3522, 2005.
- [7] M. Hariyama, T. Takeuchi, and M. Kameyama, “VLSI processor for reliable stereo matching based on adaptive window-size selection”, *Proc. of the 2001 IEEE International Conference on Robotics & Automation*, pp. 1168-1173, Seoul, Korea, 2001.
- [8] L. Di Stefano and S. Mattoccia, “Fast template matching using bounded partial correlation”, *Machine Vision and Applications*, vol. 13, no. 4, pp. 213-221, 2003.
- [9] L. Di Stefano, S. Mattoccia, and M. Mola, “An efficient algorithm for exhaustive template matching based on normalized cross correlation”, *Proc. of the 12th International Conference on Image Analysis and Processing*, pp. 322-327, Mantova, Italy, 2003.
- [10] S. Mattoccia, F. Tombari, and L. Di Stefano, “Fast full-search equivalent template matching by enhanced bounded correlation”, *IEEE Transactions on Image Processing*, vol. 17, no. 4, pp. 528-538, 2008.
- [11] Shou-Der Wei and Shang-Hong Lai, “Fast template matching based on normalized cross correlation with adaptive multilevel winner update”, *IEEE Transactions on Image Processing*, vol. 17, no. 11, pp. 2227-2235, 2008.
- [12] D. M. Tsai and C. T. Lin, “Fast normalized cross correlation for defect detection”, *Pattern Recognition Letters*, vol. 24, no. 15, pp. 2625-2631, 2003.
- [13] 도용태, 이대식, 류석환, “3차원 위치측정을 위한 스테레오 카메라 시스템의 인공 신경망을 이용한 보정”, 센서학회지, 제7권, 제6호, pp. 418-425, 1998.
- [14] L. Di Stefano and S. Mattoccia, “A sufficient condition based on the Cauchy-Schwarz inequality for efficient template matching”, *Proc. of the 2003 International Conference on Image Processing*, pp. I - 269-272, Barcelona, Spain, 2003.
- [15] L. Di Stefano, Ss Mattoccia, and F. Tombari, “ZNCC-based template matching using bounded partial correlation”, *Pattern Recognition Letters*, vol. 26, no. 14, 2129-2134, 2005.
- [16] M. J. McDonnell, “Box-filtering techniques”, *Computer Graphics and Image Processing*, vol. 17, pp. 65-70, 1981.
- [17] <http://www.middlebury.edu/stereo>

배 경 렬

- 2009년 2월 경북대학교 전자전기컴퓨터 학부(공학사)
- 2009년 3월~현재 경북대학교 전자전기 컴퓨터학부 석사과정
- 주관심분야: SoC 설계, Stereo Vision, 컴퓨터 구조

권 순

- 2003년 8월 고려대학교 전자공학과 학사
- 2006년 2월 서울대학교 전자공학과 석사
- 2006년 2월~현재 DGIST 연구원
- 주관심분야: 스테레오 비전, DSP architecture

이 용 환

- 1993년 2월 연세대학교 전자공학과 (공학사)
- 1999년 2월 연세대학교 전자공학과 (Ph.D.)
- 1999년~2002년 하이닉스반도체
- 2003년~2004년 삼성전자
- 2004년~현재 금오공과대학교 전자공학부
- 주관심분야: SoC, 임베디드 시스템 및 소프트웨어, 마이크로프로세서 구조

이 종 훈

- 1996년 2월 성균관대학교 전자공학과 공학사
- 1998년 2월 성균관대학교 전기전자및 컴퓨터공학과 공학석사
- 2002년 2월 성균관대학교 전기전자및 컴퓨터공학과 공학박사
- 2002년 1월~2005년 7월 삼성전자 통신 연구소 책임
- 2005년 7월~현재 대구경북과학기술연구원(DGIST)과제책임자/선임
- 2007년 3월~현재 영남대학교 정보전자공학부 겸임교수
- 주관심분야: 레이더/비전 융합신호처리, 차량용 레이더, lidar, V2V, V2I통신

문 병 인

- 1995년 2월 연세대학교 전자공학과 (공학사)
- 1997년 2월 연세대학교 전자공학과 (공학석사)
- 2002년 2월 연세대학교 전기전자공학과 (공학박사)
- 2002년~2004년 하이닉스반도체 선임연구원
- 2004년~2005년 연세대학교 연구교수
- 2005년~현재 경북대학교 전자전기컴퓨터학부 조교수
- 주관심분야: SoC, 디지털 VLSI, 컴퓨터 구조