

논문 2013-50-11-12

# 적응적 영역 가중치를 이용한 실시간 스테레오 비전 시스템 설계

(Design of a Realtime Stereo Vision System using Adaptive Support-weight)

류 동 훈\*, 박 태 근\*\*

(Donghoon Ryu and Taegeun Park<sup>©</sup>)

## 요 약

지역적 정합방법을 이용한 스테레오 시스템은 알고리즘의 특성상 하드웨어 설계가 용이하여 많이 사용되나 낮은 정합률로 인해 정확한 깊이 영상을 얻기 힘들기 때문에 많은 응용 분야에 사용하기에 제한이 있다. 본 논문에서 제안한 스테레오 시스템은 픽셀의 변화도(gradient)를 기반으로 한 적응적인 가중치 알고리즘을 이용하여 높은 정합 성능을 보이며 하드웨어로 설계하였을 때 실시간처리가 가능하다. 일반적으로 적응적인 가중치 윈도우를 적용할 경우 중간 결과를 재사용하기 불가능하지만 행, 열을 분리하여 처리함으로써 데이터를 재사용할 수 있고 따라서 처리성능이 개선되었다. 알고리즘에 필요한 지수 및 아크 탄젠트 함수를 구현하기 위해 선형(PWL, piecewise linear) 및 계단(step) 함수 등으로 근사화한 뒤 에러를 분석하여 최선의 파라미터를 선택하였다. 제안한 구조는 실시간처리를 위하여 9개의 프로세서를 사용하여 병렬처리를 하였으며, 동부하이텍 0.18um 라이브러리로 합성하였을 경우 최대 동작주파수 350MHz(33 fps)와 424K 게이트의 하드웨어 복잡도를 나타내었다.

## Abstract

The stereo system based on local matching is very popular due to its algorithmic simplicity, however it is limited to apply to various applications because it shows poor quality with low matching rates. In this paper, we propose and design a realtime stereo system based on an adaptive support-weight and the system shows low error rates and realtime performance. Generally, in the adaptive support-weight algorithm the intermediate computing results can not be reused to reduce the number of computations. In this research we modify the scheduling to reuse the intermediate results for the better performance by processing rows and columns separately. The nonlinear functions such as exponential or arc tangent have been designed with piecewise linear and step functions by empirical simulations and error analysis. The proposed architecture is composed of 9 processing elements for realtime performance. The proposed stereo system has been designed and synthesized using Donbu Hitek 0.18um standard cell library and can run up to 350Mhz operation frequency (33 frames per second) with 424K gates.

**Keywords** : stereo vision, realtime system, adaptive support-weight, VLSI architecture

\* 학생회원, \*\* 정회원, 가톨릭대학교 정보통신전자공학부

(Department of Information, Communication, and Electronic Engineering, The Catholic University of Korea)

© Corresponding Author (E-mail: parktg@catholic.ac.kr)

※ 이 논문은 2011년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임 (No.2011-0009513).

※ 본 연구는 2013년도 가톨릭대학교 교비연구비의 지원으로 이루어졌음(M-2013-B0002-00052).

접수일자 : 2013년7월30일, 수정완료일 : 2013년10월30일

## I. 서 론

스테레오 비전 시스템은 잡음이 많은 복잡한 환경에서도 외부의 영향을 최소화하여 깊이 정보를 계산할 수 있기 때문에 무인자동차<sup>[1]</sup>, 로봇 비전<sup>[2]</sup>, 3차원 재구성 등 다양한 분야 스테레오 비전에 대한 관심이 커지고 있으며 실제 애플리케이션들에 적용하기 위해 빠른 속도를 갖는 실시간 시스템 구현에 대한 요구가 커지고 있다. 스테레오 비전 시스템은 두 대 이상의 카메라를 사용하여 각기 다른 시점에서 촬영한 이미지를 바탕으로 거리정보를 추출하는 시스템으로 인간이 두 눈을 통해 깊이 정보를 얻는 것과 유사하게 각각의 이미지 사이에서 동일한 물체를 나타내는 대응점을 찾아서 그 대응점들이 떨어진 거리(disparity)를 계산하여 깊이 정보를 추출한다.

스테레오 시스템의 주된 과정은 각각 다른 시점에서 촬영된 이미지로부터 실제 세계의 같은 물체를 나타내는 점을 찾아내는 것으로 정합 방법에 따라 크게 지역 정합법과 전역 정합법으로 분류된다<sup>[3]</sup>. 지역 정합법의 경우 상대적으로 단순한 알고리즘을 반복적으로 사용하기 때문에 하드웨어로 설계하기 용이하지만 왜곡이나 불규칙한 특징로 인해 발생하는 애매모호한 지역에 대한 정확한 매칭이 어렵고, 전역 정합법의 경우 이런 문제들에 대해서는 덜 민감하여 상대적으로 높은 정합률을 갖지만 복잡한 연산을 사용하기 때문에 실시간 하드웨어 구현이 상대적으로 어렵다.

지역 정합법은 접근 방법에 따라 블록 매칭, 기울기 기반 매칭, 특징 기반 매칭 알고리즘 등이 있고, 전역 정합법의 경우 DP(dynamic programming), Graph Cuts, 신뢰 확산 알고리즘(BP, belief propagation) 등이 있는데 실시간 스테레오 시스템에 적합한 성능을 갖추기 위해 하드웨어로 구현할 때 주로 간단한 연산으로 구성되고 반복적인 특성을 갖는 블록 기반 알고리즘을 많이 사용한다<sup>[4-14]</sup>.

낮은 정확도를 갖는 지역 정합법의 성능을 개선하기 위해 다양한 연구가 진행되었는데, 특히 적응적 가중치 윈도우는 정합에 사용하는 윈도우에 적응적인 가중치를 줌으로서 왜곡과 복잡한 텍스처로부터 강인한 특성을 얻는 방법으로 적응적 가중치를 계산하는 다양한 방법들이 제안되었다<sup>[5-7]</sup>. 컬러정보의 유사성과 윈도우의 중심 픽셀로부터 떨어진 거리를 이용하여 윈도우에 가중

치를 적용하는 방법<sup>[5]</sup>, 주변 픽셀들과의 지오데식 거리(Geodesic distance)를 계산하여 윈도우 내의 픽셀간 연속성을 반영하면서 가중치를 적용하는 방법<sup>[6]</sup>, 가중치 윈도우에 픽셀값 대신 픽셀의 변화도를 사용하여 크기와 각도의 차이를 이용한 정합방법<sup>[7]</sup> 등이 있다.

실시간 스테레오 시스템을 구현한 연구들을 살펴보면, Nelson Chang 등은 4×5의 Jigsaw 블록을 사용한 SAD(Sum of Absolute Difference) 기반의 시스템을 DSP를 사용하여 구현하였고<sup>[8]</sup>, K. Ambrosch 등은 변화도 기반의 MCT(Modified Census Transform)와 SAD를 병합하여 정확도를 높인 시스템을 FPGA/ASIC을 사용하여 구현하였다<sup>[9]</sup>. S. Jin 등은 FPGA를 사용하여 스테레오 정합뿐만 아니라 직교화(rectification) 과정, 에러 검출, 잡음제거 등 스테레오 매칭 전, 후처리 전체를 포함하는 스테레오 시스템을 구현하였고<sup>[10]</sup>, Darabiha 등은 각도의 연관성과 차이점을 모두 이용하여 다해상도, 다양한 방향을 기반으로 하는 Local weighted phase-correlation 알고리즘을 FPGA를 사용하여 구현하였다<sup>[11]</sup>. S. Park 등은 격자 구조물 기반의 스테레오 매칭 기법을 사용하여 시스틀릭 어레이를 적용하여 single-line DP보다 우수한 정확도를 갖는 시스템을 구현하였다<sup>[12]</sup>.

본 논문에서 제안한 구조는 픽셀의 변화도를 기반<sup>[7]</sup>으로 컬러 유사성과 거리에 따라서 적응적 가중치를 적용하여 높은 정합률을 갖고 있으며, 9개의 PE를 사용하여 병렬 처리를 통해 실시간 스테레오 시스템에 적합한 성능을 만족한다. 적응적 가중치 알고리즘의 특성상 데이터 재사용이 불가능하여 많은 연산을 다시 해야 하는데 이를 효율적인 스케줄링과 행, 열을 분리하여 가중치를 갱신하는 계산 방법으로 해결하였으며, 윈도우의 열 단위로 데이터를 재사용하기 때문에 레지스터 뱅크를 사용하여 계산 효율을 크게 높였다. 알고리즘에 사용되는 지수, 아크탄젠트 연산은 하드웨어로 구현할 때 복잡도와 하드웨어 요구량이 커지기 때문에 선형 및 계단 함수 등으로 에러를 분석하여 경험적으로 근사화하여 허용할 수 있는 오차 범위를 갖도록 구현하였다.

본 논문의 구성은 다음과 같다. II장에서는 적응적 가중치를 이용한 정합 알고리즘들을 설명하고, III장에서 제안한 시스템의 스케줄링과 구조에 대해서 설명하고 IV장에서 합성 결과를 가지고 다른 스테레오 시스템과 성능을 비교 분석하고 V장에서 결론을 맺는다.

## II. Adaptive support-weight approaches

지역 정합법에서는 윈도우라 불리는 픽셀 영역에 대하여 연산을 수행한다. 이때 윈도우가 너무 클 경우 계산량도 많아지고 윈도우 내에 복잡한 모양의 텍스처나 여러 개의 텍스처가 포함될 확률이 커져서 정확한 정합이 어렵고, 작은 윈도우를 사용할 경우 텍스처의 다양성을 충분히 반영하지 못하여 정합률이 낮아지는데 적절한 윈도우의 크기와 모양을 정하는 것은 NP-hard 문제로 매우 어렵다. 그리하여 고정된 윈도우를 사용하면서 위의 문제를 해결하기 위해 윈도우에 속하는 모든 픽셀 값을 동등하게 사용하는 것이 아니라 윈도우의 픽셀 마다 정합도를 기반으로 다른 가중치를 적용하여 연산을 수행하며 각 가중치는 매번 적응적으로 갱신된다.

가중치를 정의하는 다양한 연구가 진행되었는데<sup>[5-7]</sup>, 윈도우의 중심 픽셀 값을 기준으로 다른 픽셀들의 컬러 유사성들을 비교하여 유사할수록 큰 가중치를 주는 방법과 윈도우의 중심에 위치한 중심 픽셀과의 유클리드 거리에 반비례하여 가중치를 반영하는 방법, 인접한 픽셀들의 지오데식 거리(Geodesic distance)를 계산하여 윈도우 내의 픽셀들 사이의 연속성을 유지하도록 가중치를 결정하는 방법 등이 있다.

### 1. 색 유사성과 거리를 이용한 알고리즘<sup>[10]</sup>

CIELab 컬러 형식은 인간이 색을 구별하는 방법에 기초하여 연구된 색 공간으로 반사율(인간의 시감과 같은 명도) L, 색도 다이어그램인 a, b 성분으로 표현된다. 노랑-파랑, 초록-빨강의 반대색에 기초하기 때문에 +a(빨강), -a(초록), +b(노랑), -b(파랑)을 나타내며, 이 요소들을 이용해 색을 3차원으로 표현한다. 각 컬러 성분의 유클리드 거리(Euclidean distance)는 인간이 색을 구별하는 동작과 매우 큰 상관관계를 보이기 때문에, CIELab 컬러 형식을 사용하고 픽셀  $p(c_p = [L_p, a_p, b_p])$ ,  $q(c_q = [L_q, a_q, b_q])$ 의 유사성  $\Delta c_{pq}$ 은 각 컬러 성분의 유클리드 거리를 사용하여 계산한다.

$$\Delta c_{pq} = \sqrt{(L_p - L_q)^2 + (a_p - a_q)^2 + (b_p - b_q)^2} \quad (1)$$

거리에 따른 가중치도 윈도우의 중심에 있는 값이 다른 부분에 비해 더 중요하게 사용되므로 중심 픽셀과 다른 픽셀과의 거리를 고려하여 가중치를 부여한다.

$$w(p, q) = \exp\left(-\left(\frac{\Delta c_{pq}}{\gamma_c} + \frac{\Delta g_{pq}}{\gamma_p}\right)\right) \quad (2)$$

컬러 성분의 유사성과 거리에 따른 가중치는 각각  $\gamma_c$ ,  $\gamma_p$ 로 나누어져 픽셀의 유사성과 거리 사이의 중요도를 조절하는 값으로 이미지에 따라 최적의 값이 달라지며, 지수연산을 통해 최종 가중치를 계산한다.

### 2. 지오데식 거리(Geodesic distance) 기반 알고리즘<sup>[6]</sup>

윈도우에 속하는 픽셀  $p$ 와 윈도우의 중심 픽셀  $c$ 가 있을 때 지오데식 거리  $D(p, c)$ 는 가장 짧은 경로로 정의된다.

$$D(p, c) = \min_{P \in P_{p,c}} d(P) \quad (3)$$

$P_{p,c}$ 는  $p, c$  사이의 가능한 모든 경로의 집합을 나타내고 경로  $P$ 는 연결 가능한 인접한 점들의 집합  $\{P_1, P_2, \dots, P_n\}$ 으로 정의한다. 경로에 대한 코스트  $d()$ 는 다음과 같이 계산하고  $d_c()$ 는 각 컬러 성분의 차이를 이용하는 함수이다.

$$d(P) = \sum_{i=2}^{i=n} d_c(P_i, P_{i-1}),$$

$$d_c(p, q) = \sqrt{(r_p - r_q)^2 + (g_p - g_q)^2 + (b_p - b_q)^2} \quad (4)$$

$$w(p, c) = \exp\left(-\frac{D(p, c)}{\gamma}\right) \quad (5)$$

$\gamma$ 는 세분화(segmentation) 정도를 조절해주는 파라미터로 값이 클수록 윈도우를 엄격하게 구분한다. 최종 가중치 윈도우는 지수연산을 통해 거리  $D(p, c)$ 가 짧을수록 큰 가중치를 갖도록 계산한다. 이에 따라 컬러 정보의 유사성과 거리에 따른 적응적 가중치를 적용했을 때는 윈도우의 중심에 위치한 픽셀 값을 기준으로 세분화가 잘 이루어지나 같은 물체가 아니지만 비슷하거나 같은 컬러 성분을 갖는 경우에도 큰 가중치 값을 갖는다. 그러나 지오데식 거리를 통한 적응적 가중치 윈도우의 경우 픽셀의 값뿐만 아니라 픽셀의 연속성을 이용하기 때문에 중심 픽셀에 속하는 물체만 큰 가중치를 갖게 되고 다른 물체가 비슷한 컬러 성분을 갖더라도 작은 가중치를 갖게 되어 윈도우 영역 내에서 배제될 수 있다.

### 3. 픽셀의 변화도 기반 알고리즘<sup>[7]</sup>

앞에서 설명한 컬러 성분의 유사성과 가까운 정도를 통한 적응적 영역 가중치를 적용하면서 픽셀 값 대신 픽셀의 변화도를 통해 크기와 각도를 사용하여 정합에 사용할 경우 픽셀 값을 사용할 때보다 조명과 노출에 따른 밝기에 대해 더 강인하고 정확한 연산이 가능해진다. 픽셀의 변화도는 행 방향과 열 방향으로 좌우, 상하 픽셀의 차이를 이용해서 계산 가능하고, 이를 이용하여 크기와 각도를 계산하여 2차원에 대한 정보를 표현할 수 있다.

$$G_x = [1 \ 0 \ -1]G_y = [1 \ 0 \ -1]^T,$$

$$m = \sqrt{G_x^2 + G_y^2}, \quad \phi = \arctan\left(\frac{G_y}{G_x}\right) \quad (6)$$

$G_x, G_y$ 는 각각  $x, y$  방향의 변화도를 나타내고  $m$ 은 변화도의 크기,  $\phi$ 는 변화도의 각도를 나타낸다. 스테레오 정합 알고리즘에서 정합을 판단하기 위해 크기와 각도의 변화도를 비교하여 변화도가 가장 작은 영역이 정합된 점으로 판단하는데, 크기와 각도의 변화량은 다음과 같은 식을 통해 계산한다.

$$e(q, \bar{q}_d) = \min\left\{\sum_{c \in r, g, b} \alpha |m_c(q) - m_c(\bar{q}_d)| + f(|\phi_c(q) - \phi_c(\bar{q}_d)|), T\right\} \quad (7)$$

$e(q, \bar{q}_d)$ 는 기준 윈도우  $q$ 와 탐색 영역  $d$ 에 속하는 영역  $\bar{q}_d$ 의 윈도우의 크기와 각도의 변화량을 나타내고,  $m$ 은 변화도의 크기,  $\phi$ 는 변화도의 각도,  $f(\cdot)$ 는 각도 성분이 0과  $2\pi$  사이의 값을 갖도록 조정해주는 함수이고, 크기와 각도 사이의 스케일링을 위해  $\alpha$ 를 크기의 변화량과 곱해준다.  $T$ 는 버림 값(Truncation value)으로  $T$ 보다 큰 결과는 정합에 사용하지 않음으로써 정합 결과를 안정적으로 만들기 위해 사용한다.

$$E(q, \bar{q}_d) = w(p, q) \times e(q, \bar{q}_d) \quad (8)$$

깊이 정보를 결정하는데 사용하는 정합 값  $E(q, \bar{q}_d)$ 은 계산한 크기와 각도의 변화량과 컬러 성분의 유사성과 거리에 따라 계산한 가중치 윈도우를 곱하여 계산한다.

### III. 제안한 구조

II장에서 언급한 지오테식 거리를 사용하여 가중치

를 계산한 경우, 윈도우 내의 픽셀의 연속성을 바탕으로 가중치를 계산하기 때문에 단순히 컬러 유사성과 거리를 통해서 가중치를 계산할 경우에 비해 좀 더 낮은 어려움을 보이지만 계산 복잡도가 8배 정도 증가하여 실시간 처리에 어려움이 있다. 따라서 본 논문에서는 실시간 처리 성능을 갖고 어려움도 상대적으로 우수한 픽셀 변화도 기반 적응적 영역 가중치 알고리즘을 바탕으로 시스템을 설계하였고, 논문 [14]의 방법과 유사하게 행 방향과 열 방향으로 나누어 계산함으로써 데이터를 재사용이 가능하여 계산량을 대폭 줄였다. 데이터 재사용에 관한 자세한 설명은 3.2에서 기술하였다.

$$E(q, \bar{q}_d) = w_{column}(p, q) \times w_{row}(p, q) \times e(q, \bar{q}_d) \quad (9)$$

$E(q, \bar{q}_d)$ 는 기준 이미지의 윈도우  $q$ 로부터 다른 이미지에서  $d$ 만큼 떨어진 거리에 있는 윈도우  $\bar{q}_d$  영역의 적응적 가중치와 픽셀의 변화도의 크기와 각도의 차이를 곱한 값으로 거리 정보를 결정하는데 기준이 되는 정합 값이다.  $w_{column}(p, q)$ 와  $w_{row}(p, q)$ 는 픽셀  $p$ 을 기준으로 각각 열 방향, 행 방향 거리를 계산하여 구한 가중치를 나타낸다.

#### 1. 제안한 스테레오 시스템의 구조

그림 1은 제안한 스테레오 시스템의 전체 구조로 실시간 시스템을 위하여 9개의 PE(processing element)를 병렬로 사용 하였으며 9개의 값을 동시에 더하기 위하여 CSA(Carry Save Adder) 기반의 가산기 트리(Adder tree)와 열에 해당하는 값을 만들기 위해 누산기, 데이

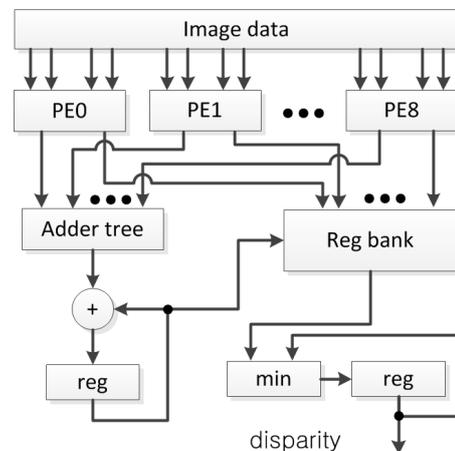


그림 1. 제안된 시스템 구조  
Fig. 1. Proposed system architecture.

터를 재사용하기 위하여 중간 데이터를 저장하고 행 방향 가중치를 곱함으로써 검색윈도우의 기준 값을 출력해주는 레지스터뱅크(Reg bank) 모듈, 깊이정보(disparity) 결정을 위한 최소값 비교기(Min) 등으로 구성된다.

각 PE는 적응적 영역 가중치 값과 픽셀의 변화도의 크기 및 각도 등을 계산하는 역할을 하는데 거리에 따른 가중치를 계산하는 과정이 열 방향 가중치를 계산하는 과정과 행 방향 가중치를 계산하는 과정으로 나뉜다. 표 1은 병렬처리를 적용했을 때 가중치 블록과 변화도 블록의 동작을 나타낸 것으로, 윈도우 크기 27 × 27 기준으로 9개의 모듈을 사용하여 병렬처리 했기 때문에 윈도우의 한 열, 또는 행을 계산하는데 각 3 clock을 필요로 한다. 처음 3 clock 동안은 가중치 모듈에서는 열 방향 가중치를 계산하고 변화도 모듈 또한 동작하여 출력을 내보내고 다음 3 clock 동안은 가중치 모듈은 행 방향 가중치를 계산하고 변화도 모듈은 동작하지 않는다. 또한 처음 1, 2 clock 동안에 열 방향 가중치를 계산하기 위해 그림 3의 Adder tree에서 값이 누적되고 3 clock에 열에 해당하는 모든 픽셀 값들이 누적되었을 때 Reg bank에 저장된다.

Reg bank 모듈에서는 적절한 타이밍에 맞춰 입력되는 데이터를 분류하여 재사용되는 데이터를 저장하거나 저장된 데이터와 입력을 사용하여 윈도우 전체에

표 1. 클록별 모듈의 동작

Table 1. Operations of each module by clock.

| clk      | 1  | 2  | 3  | 4  | 5  | 6  | 7  | ... |
|----------|----|----|----|----|----|----|----|-----|
| 가중치      | 열  | 열  | 열  | 행  | 행  | 행  | 열  | ... |
| 변화도      | 계산 | 계산 | 계산 | 휴식 | 휴식 | 휴식 | 계산 | ... |
| Reg bank | -  | -  | 입력 | 출력 | 출력 | 출력 | -  | ... |

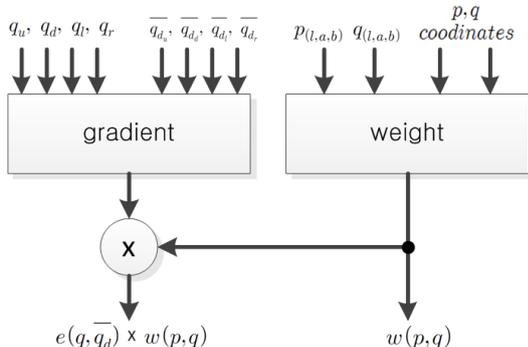


그림 2. PE의 구조  
Fig. 2. Structure of a PE.

대한 연산을 수행한다. 열 가중치와 변화도가 계산될 때 Reg bank 모듈은 레지스터 뱅크에 데이터를 저장하고 행 방향 가중치가 계산될 때 레지스터에 저장된 값을 사용하여 전체 윈도우에 대한 연산을 수행하여 값을 출력한다.

그림 2는 PE의 구조를 나타낸 것으로 크게 가중치 모듈과 변화도 모듈로 구성된다. 가중치 모듈은 해당 픽셀의 CIElab 컬러 형식의 각 컬러 정보(p(l, a, b), q(l, a, b))와 픽셀간 거리를 입력으로 받아 적응적 가중치를 계산하고 변화도 모듈은 양쪽 이미지 각각의 변화도의 크기와 각도의 차이를 계산하기 위해 한 이미지로부터 기준 픽셀로부터 상, 하, 좌, 위 네 개의 픽셀 값을 입력받아 계산한다. PE의 출력은 변화도 값과 가중치를 곱한 값과 그냥 가중치만을 따로 출력하여 타이밍에 따라 필요한 출력을 사용한다.

그림 3은 가중치 모듈의 구조로 이미지의 각 컬러 성분과 픽셀의 거리를 입력으로 받아 가중치를 계산하는 모듈로 식(2)를 계산한다. 왼쪽 세 개의 절대 값 블록은 컬러의 유사성을 계산하는데 사용되어 각 픽셀 성분의 차이를 계산한 뒤 누적하고 컬러 유사성의 중요도를 조절하는 파라미터 λ<sub>c</sub>와 곱해진다. 점선으로 표시된 블록은 두 픽셀의 거리의 차이를 계산하는 블록으로 입력이 규칙적으로 들어오기 때문에 카운터를 사용하여 설계할 수 있다. 계산된 거리의 차이 값은 파라미터 λ<sub>d</sub>와 곱해져 거리에 따른 유사성의 중요도를 조절하는 역할을 한다. 계산된 두 값은 더해져 지수 모듈로 들어가고 최종적인 가중치를 계산하게 된다.

지수 모듈과 같은 복잡한 연산은 일반적으로

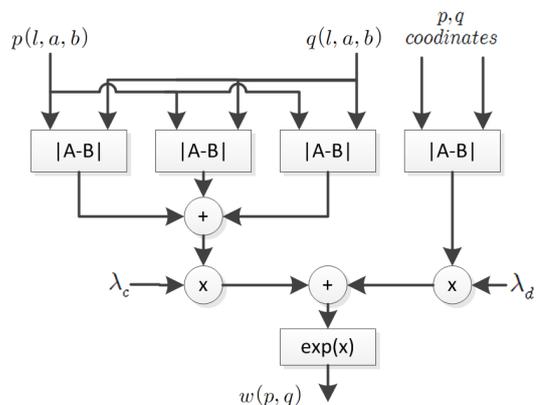


그림 3. 가중치 모듈의 구조  
Fig. 3. Structure of weight module.

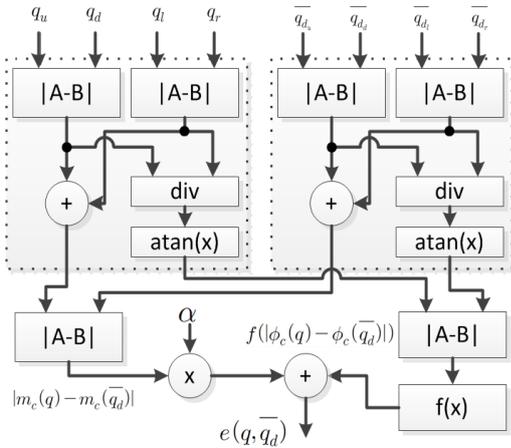


그림 4. 변화도 모듈의 구조  
Fig. 4. Structure of gradient module.

TLB(table look-up)를 사용하거나 계단 함수, 선형 함수 등을 사용하여 적절하게 설계할 수 있다. 본 연구에서는 선형 함수와 계단 함수를 다양한 구간으로 설계하여 에러를 분석한 결과, 계단 함수를 4개 구간으로 나누어서 구현하였을 때 가장 좋았다.

그림 4는 각각의 이미지에서 변화도를 계산하여 변화도의 크기와 각도의 차이를 계산하는 모듈로  $q_u, q_w, q_l, q_r, \bar{q}_{d_u}, \bar{q}_{d_w}, \bar{q}_{d_l}, \bar{q}_{d_r}$ 은 각각 이미지  $p, q$ 의 기준 픽셀로부터 상, 하, 좌, 우의 픽셀 값을 나타내고, 세 개의 컬러 성분을 이용하여 계산한다. 차이를 통해 가로, 세로방향 변화도의 절대 값을 구한 뒤 다시 절대 값 연산을 이용하여 크기를 구하고, 나누기 연산을 통해  $G_y/G_x$ 를 구한 뒤  $atan(x)$  모듈에서 각도를 구한다. 여기서 구현된 나눗셈기는 nonrestoring array divider를 사용하였다<sup>[15]</sup>.  $p, q$ 에 대해 구한 변화도의 크기와 각도는 다시 변화도를 구하고, 구한 각도의 차이는  $f(x)$  모듈에서 0에서  $2\pi$  사이값을 갖도록 조절되고 각도와 크기 사이의 스케일링을 위한 파라미터  $\alpha$ 와 곱해진 뒤 각도와 더해져 출력된다.

$atan(x)$  모듈 또한 다양한 실험을 통해 4개 구간으로 나눈 선형 함수로 설계하였다. 표 2는 지수 모듈과 아크탄젠트 함수를 하드웨어로 구현하기 위해 계단 함수와 선형 함수로 표현한 것으로 에러를 최소화하기 위해 다양한 구간으로 나누어 에러를 분석한 뒤 가장 좋은 결과를 보인 값들로 설계하였다.

데이터를 재사용하기 위해서는 재사용할 데이터가 만들어진 뒤 다시 사용할 때까지 저장하기 위한 공간이

표 2. 지수 및 아크탄젠트 함수 구현  
Table 2. Approximation of  $\exp(x)$  and  $\tan(x)$ .

| 구분        | 1영역                  | 2영역                            | 3영역                              | 4영역                               | 5영역    |
|-----------|----------------------|--------------------------------|----------------------------------|-----------------------------------|--------|
| $\exp(x)$ | $\sim 0.25$          | $\sim 0.6875$                  | $\sim 1.375$                     | $\sim 2.5$                        | $\sim$ |
| - step    | 1                    | 0.75                           | 0.5                              | 0.25                              | 0      |
| $\tan(x)$ | $\sim 0.375$         | $\sim 0.875$                   | $\sim 2.0625$                    | $\sim 14.0625$                    | $\sim$ |
| - PWL     | $0.9453125 \times x$ | $0.6953125 \times x + 0.09375$ | $0.3203125 \times x + 0.4453125$ | $0.0234375 \times x + 1.05468750$ | 1.5707 |

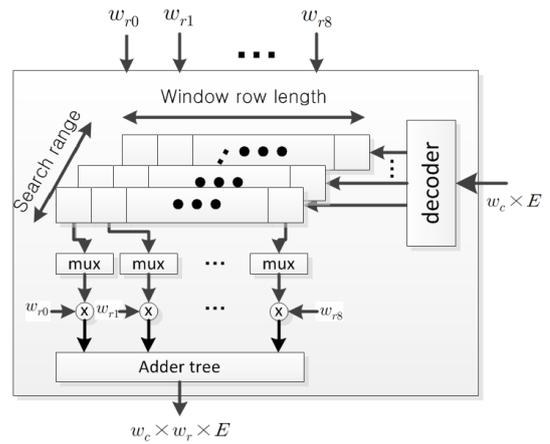


그림 5. 레지스터 뱅크 구조  
Fig. 5. Structure of register bank.

필요하다. 제안한 알고리즘에서는 행 단위로 재사용하기 때문에 데이터가 처음 만들어졌을 때를 기준으로 기준 윈도우가 1 픽셀 쉬프트 되어 계산했을 때 다시 사용되고, 기준 윈도우가 (윈도우 크기-1)만큼 움직였을 때까지 계속해서 재사용된다. 한 기준 윈도우에 대한 연산은 탐색 거리만큼의 윈도우 연산으로 이루어져 있고, 그러므로 필요한 저장 공간은 하나의 탐색 거리를 기준으로 윈도우의 행 방향 길이만큼의 값을 저장할 수 있어야하고, 하나의 저장 공간당 윈도우의 열에 해당하는 값을 갖고 있어야한다.

그림 5는 재사용되는 데이터를 저장하기 위한 레지스터 뱅크의 구조를 나타낸 것으로 윈도우(행 길이 × 탐색거리)만큼의 크기를 갖는 2차원 레지스터 뱅크를 사용하였다. 열 방향 가중치와 변화도의 차이가 곱해진 값이 입력으로 들어왔을 때는 디코더를 통해 해당하는 탐색 거리의 레지스터에 저장되고, 행 방향 가중치가 입력으로 들어올 때는 멀티플렉서를 통해 해당하는 탐색 거리의 출력 값과 곱해진다. 곱해진 열 단위의 값은 가산트리(Adder tree)에서 더해져 윈도우에 해당하는 기준 값을 출력하게 된다.

2. 스케줄링과 데이터 재사용

윈도우를 사용하는 지역정합법의 특징상 연산과정에서 열, 행 방향의 많은 중복연산이 존재하므로 일정량의 저장 공간을 이용하여 중간 연산결과를 재사용할 경우 한 픽셀에 해당하는 값만을 새로 계산하여 처리할 수 있으므로 고성능처리가 가능하다.

이 방법으로 가중치를 계산할 경우 기준 윈도우가 바뀌어도 열에 대한 기준 픽셀은 변하지 않으므로 다음 연산에서 값을 재사용할 수 있고 이때 기준 픽셀이 변하는 행 방향 거리에 대한 가중치만 새로 계산하여 사용할 수 있다.

그림 6은 윈도우 계산을 수행할 때 데이터를 재사용하기 위한 과정을 나타낸 그림으로, 하나의 윈도우는 27개의 열로 구성되어있고, 하나의 열은 27개의 픽셀로 구성된다. 회색 부분의 현재 윈도우의 계산을 수행할 때 레지스터에는 각 열에 해당하는 값 column0 ~ column26이 저장되어 있고 행 방향 가중치를 구하면서 저장된 출력 값을 사용하여 윈도우 계산을 마치게 된다. 이때 현재 윈도우의 가장 왼쪽 열에 위치한 column0의 경우 더 이상 다음 연산에서 사용되지 않으므로 레지스터에서 지워지고, 다음 윈도우 계산을 수행할 때 새롭게 필요한 column27을 계산하면서 새롭게

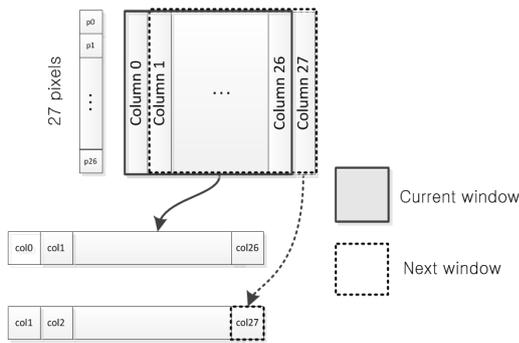


그림 6. 중간결과 값 재사용 방법

Fig. 6. Reusing scheme of intermediate values.

표 3. 데이터 재사용 방법에 따른 계산량 분석

Table 3. Performance analysis due to data reusing.

| 구분            | 윈도우크기   | 탐색거리 | 영상크기      | 사이클 수    |
|---------------|---------|------|-----------|----------|
| 데이터 재사용 안함    | 27 × 27 | 16   | 384 × 288 | 약 1,290M |
| 열 단위 재사용      | 27 × 27 | 16   | 384 × 288 | 약 48M    |
| 픽셀 단위 재사용     | 27 × 27 | 16   | 384 × 288 | 약 1.8M   |
| 제안한 방법(1 PE)  | 27 × 27 | 16   | 384 × 288 | 약 96M    |
| 제안한 방법(9 PEs) | 27 × 27 | 16   | 384 × 288 | 약 10M    |

레지스터에 저장하게 된다. 새롭게 계산한 column27과 이전에 계산한 column1 ~ column26을 통해 다음 윈도우 연산을 수행하게 되며 FIFO(first-in, first-out) 레지스터를 사용하여 설계하였다.

표 3은 데이터를 재사용하지 않을 때, 열 단위로 재사용 했을 때, 픽셀 단위로 재사용 했을 때, 전체 결과 값을 계산하는데 필요한 사이클 수를 비교한 것이다. 본 논문에서 제안한 구조는 기본적으로 열 단위 재사용을 사용하지만 가중치를 계산할 때, 열 방향 거리와 행 방향 거리로 나누어서 계산하기 때문에 열 단위 재사용보다 필요한 사이클 수가 두 배가 된다. 열 뿐만 아니라 행 방향으로도 데이터를 재사용할 경우 하나의 픽셀을 이용해서 윈도우 연산을 수행할 수 있는 데, 이 때 필요한 사이클 수는 약 1.8M로 가장 작지만 재사용할 데이터를 저장하기 위한 레지스터가 매우 커지는 문제로 구현하기 힘들다. 제안한 방법의 경우 계산량은 약 96M로 데이터를 재사용하지 않고 구현하였을 때의 약 1,290M보다 훨씬 작은 계산량을 필요로 하며 실시간 처리를 위하여 9개의 PE를 사용하였을 경우 계산량은 약 10M까지 감소한다.

IV. 설계 및 성능 분석

제안한 스테레오 비전 시스템은 VerilogHDL로 설계한 뒤 Active-HDL로 검증하였으며, Synopsys를 사용하여 합성하였다. 윈도우 크기 27×27, 탐색거리 16, 영상크기 384×288를 처리할 수 있는 제안한 구조를 동부 하이텍 0.18um 라이브러리를 사용하여 합성한 결과로 전체 시스템은 424K개의 게이트를 필요로 하며 데이터 재사용을 위하여 1KB의 저장 공간을 필요로 한다. 최대 클럭 속도는 350 MHz로 약 33 fps의 속도를 갖는다.

제안한 구조에서 사용한 알고리즘은  $\lambda_c, \lambda_d, \alpha$  등 다양한 파라미터를 사용하며 파라미터 값에 따라서 결과가 크게 바뀌게 된다. 본 논문에서는 최적의 파라미터를 찾기 위해 실험적으로 에러를 분석하여 결정하였으며  $\lambda_c$ 는 0.0774,  $\lambda_d$ 는 0.074,  $\alpha$ 는 0.125을 사용하였다.

그림 7은 지오데식 거리로 계산한 가중치<sup>[6]</sup> 방법, 컬러 유사성과 거리에 따른 적응적인 가중치<sup>[5]</sup> 방법과 제안한 시스템의 결과를 비교하여 나타낸 그림이다. 제안한 구조의 결과는 구현 시 고정소수점 비트 할당, 비선

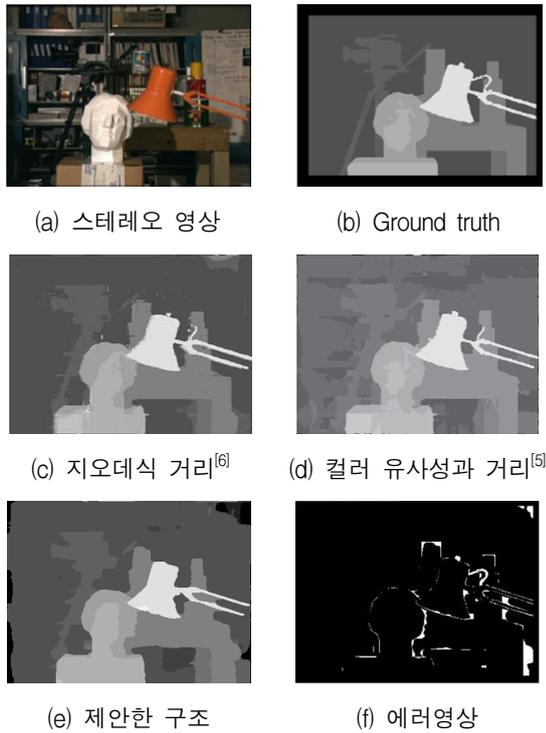


그림 7. 제안한 알고리즘의 깊이정보 분석  
Fig. 7. Disparity analysis of the proposed algorithm.

형 함수에 대한 근사화, 후처리에 사용된 미디언 필터링 등으로 다른 알고리즘에 비해 해상도는 조금 떨어지지만 에러를 비교 했을 때, (c)의 경우 1.45%, (d)의 경우 1.38%, 제안한 시스템의 경우 2.18%로 상당한 정확도를 유지하였다.

표 4는 제안한 스테레오 비전 시스템과 다른 실시간 스테레오 비전 시스템을 비교한 것이다. [8], [9], [10]의 경우 블록 매칭 알고리즘을 하드웨어로 구현했을 때의 특징으로 빠른 속도와 상대적으로 높은 에러율을 확인할 수 있으며, 전역적 정합 방법인 [12], [13]의 경우 반대로 상대적으로 느린 속도와 낮은 에러율을 확인할 수

표 4. 스테레오 비전 시스템의 성능분석  
Table 4. Performance analysis of stereo vision systems.

| 알고리즘             | 구현 방법               | 이미지크기   | 탐색거리 | 윈도우크기                            | 속도(fps) | 에러(%) |
|------------------|---------------------|---------|------|----------------------------------|---------|-------|
| SAD/jigsaw[8]    | DSP                 | 384×288 | 16   | jigsaw(4 × 5)                    | 50      | 21.5  |
| MCT/SAD[9]       | FPGA / ASIC         | 750×400 | 60   | 23 × 23                          | 60      | 11.85 |
| Census[10]       | FPGA                | 640×480 | 64   | census(11 × 11)<br>corr(15 × 15) | 230     | 9.79  |
| Local weight[11] | FPGA                | 256×360 | 20   | N/A                              | 30      | -     |
| BP[12]           | Nvidia Geforce 7900 | 320×240 | 16   | N/A                              | 16      | 3.40  |
| DP[13]           | AMD AthlonXP 2800   | 384×288 | 100  | N/A                              | 19      | 2.85  |
| 제안한 구조           | ASIC                | 384×288 | 16   | 27 × 27                          | 33      | 2.18  |

있다. 제안한 구조의 속도는 33 fps로 실시간 시스템이 요구하는 최소 성능인 30 fps를 만족하고 에러는 2.18%로 다른 실시간 스테레오 비전 시스템보다 높은 정합률을 나타내었다.

## V. 결 론

본 논문에서 제안한 스테레오 비전 시스템은 적응적 가중치 윈도우와 픽셀의 변화도 기반 알고리즘을 사용하여 다른 블록 매칭 알고리즘에 비해 우수한 정합률을 갖고 있으며 데이터 재사용과 병렬처리를 통해 실시간 시스템에 필요한 성능을 갖추고 있다. 향후 연구 과제로, 현재 알고리즘에서 가중치를 계산할 때 열 방향과 행 방향으로 나눠서 알고리즘을 수행하기 때문에 변화도 모듈의 경우 하드웨어 효율이 저하될 수 있는데, 가중치 모듈에 비해서 변화도 모듈의 지연 경로(critical path)가 훨씬 길다는 점을 고려하여 변화도 모듈이 동작하는 동안 행 방향, 열 방향 가중치를 모두 계산하여 하드웨어 효율을 개선시키는 구조를 연구하고 있다.

## 감사의 글

저자들은 본 연구를 위하여 설계 소프트웨어를 제공하여 준 IDEC(IC Design Education Center)에 감사드립니다.

## REFERENCES

[1] K. Y. Lee, J. W. Lee, and N. Houshangi, "A stereo matching algorithm based on top-view transformation and dynamic programming for

- road-vehicle detection,” *Int. J. of Control, Automation, and Systems*, vol. 7, no. 2, pp. 221-231, 2009.
- [2] D. S. Kim. “A real-time stereo depth extraction hardware for intelligent home assistant robot,” *IEEE Trans. on Consumer Electronics*, vol. 56, pp. 1782-1788, 2010.
- [3] D. Scharstein and R. Szeliski, “A taxonomy and evaluation of dense two-frame stereo correspondence algorithms,” *Int. J. of Computer Vision*, vol. 47, pp. 79-88, 2002.
- [4] M. Z. Brown, D. Burschka, and G. D. Hager, “Advances in computational stereo,” *Trans. Pattern Analysis and Machine Intelligence*, vol. 25, pp. 993-1008, 2003.
- [5] K. J. Yoon, and I. S. Kweon, “Adaptive support-weight approach for correspondence search,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 28, pp. 650-656, 2006.
- [6] A. Hosni, M. Bleyer, M. Gelautz, and C. Rhemann. “Local stereo matching using geodesic support weights,” *16th IEEE Int. Conf. on Image Processing on*, pp. 2093-2096, 2009.
- [7] L. De-Maeztu, A. Villanueva, and R. Cabeza, “Stereo matching using gradient similarity and locally adaptive support-weight,” *Pattern Recognition Letters*, vol.32, pp. 1643-1651, 2011.
- [8] N. Chang, T. M. Lin, T. H. Tsai, Y. C. Tseng, T. S. Chang, “Real-time DSP implementation on local stereo matching,” *IEEE Int. Conf. on Multimedia and Expo*, pp. 2090 - 2093, 2007.
- [9] K. Ambrosch, W. Kubinger, “Accurate hardware-based stereo vision,” *Computer Vision and Image Understanding*, vol. 114. pp. 1303-1316, 2010.
- [10] S. Jin, J. Cho, X. D. Pham, K. M. Lee, S. K. Park, M. Kim, and J. W. Jeon, “FPGA design and implementation of a real-time stereo vision system,” *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 20, no. 1, pp. 15-26, 2010.
- [11] A. Darabiha, J. Rose, and W. J. Maclean, “Video-rate stereo depth measurement on programmable hardware,” *2003 IEEE Conf. on Computer Vision and Pattern Recognition*, vol. 1, pp. 203-210, 2003.
- [12] S. Park, H. Jeong, “Real-time stereo vision FPGA chip with low error rate,” *2007 Int. Conf. on Multimedia and Ubiquitous Engineering*, pp. 751-765, 2007.
- [13] H. Hirschmuller, P. R. Innocent, and J. Garibaldi, “Real-time correlation based stereo vision with reduced border errors,” *Int. J. of Computer Vision*, vol. 47, pp. 229-246, 2004.
- [14] N. Chang, T. Tsai, B. Hsu, Y. Chen, and T. Chang, “Algorithm and architecture of disparity estimation with mini-census adaptive support weight,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 6, pp. 792-805, Jun. 2010.
- [15] K. Hwang, “Computer arithmetic: principles, architecture, and design,” John Wiley & Sons, 1979.

---

 저 자 소 개
 

---



류 동 훈(학생회원)  
 2012년 가톨릭대학교 정보통신  
 전자공학부 학사 졸업.  
 2012년~현재 가톨릭대학교 정보  
 통신전자공학부 석사과정.

<주관심분야 : VLSI 설계, 영상신호처리, SOC 설계>



박 태 근(정회원)-교신저자  
 1985년 연세대학교 전자공학과  
 학사 졸업.  
 1993년 Syracuse Univ.  
 컴퓨터 공학박사 졸업.  
 1994년~1998년 현대전자 System  
 IC 연구소, 책임연구원

1998년~현재 가톨릭대학교 정보통신전자공학부  
 교수

<주관심분야 : VLSI 설계, CAD, 컴퓨터 구조>