

論文2000-37SD-10-8

MOEPE : 스테레오 정합 하드웨어를 위한 *Merged Odd-Even PE* 구조 (MOEPE : *Merged Odd-Even PE* Architecture for Stereo Matching Hardware)

韓 必 宇*, 梁 榮 日**
(Phil-Woo Han and Yeong-Yil Yang)

요 약

본 논문에서는 동적 프로그래밍에 기반한 스테레오 정합 알고리즘을 구현하는 새로운 하드웨어 구조를 제안하였다. 제안된 MOEPE(*Merged Odd-Even PE*) 구조는 시스톨릭 방법으로 동작하고, 극상선상의 화소의 밝기 값으로부터 변이를 찾는다. MOEPE 구조에서 사용된 PE 수는 변이 제약조건의 수와 일치하는데, 이는 극상선상의 화소 수만큼의 PE를 사용하는 기존의 방법에 비하여 훨씬 적은 수의 PE를 사용한다. MOEPE 구조에서 사용된 PE 수는 일반적 크기의 영상에 대하여, 기존의 방법에 비하여 약 25배 적은 수의 PE를 사용한다. 제안된 구조는 VHDL로 기술하였고, Synopsys 설계 환경에서 시뮬레이션을 수행하였다.

Abstract

In this paper, we propose the new hardware architecture which implements the stereo matching algorithm using the dynamprogrammethod. The proposed MOEPE(*Merged Odd-Even PE*) architecture operates in the systolic manner and finds the disparities from the intensities of the pixels on the epipolar line. The number of PEs used in the MOEPE architecture is the same number of the range constraint, which reduced the nuMber of the necessary PEs dramatically compared to the traditional method which uses the PEs with the same number of pixels on the epipolar line. For the normal sized images, the numof the PEs in the MOEPE architecture is less than that of the PEs in the traditional method by 25×The proposed architecture is modeled with the VHDL code and simulated by the SYNOPSIS tool.

I. 서 론

스테레오 시각은 서로 다른 위치에서 두 대의 카메라

* 正會員, (주) 한국엑시스 技術研究所
(Korea Axis Co., LTD R&D center)

** 正會員, 慶尙大學校 電氣電子工學部
(GyeongSang National University, Division of
Electric and Electronics Engineering)

※ 본 연구는 경상대학교 생산기술연구소 지원에 의하여 이루어졌음

接受日字: 1999年5月31日, 수정완료일: 2000年8月25日

라로 획득한 2차원 영상들로부터 3차원 거리 정보를 추출하며, 기본적으로 카메라 모델링, 영상 획득, 대응 점 정합, 삼각 측량법에 의한 3차원 거리 정보 추출 단계 등으로 이루어진다. 특정한 목적을 위한 산업 분야에 실용적인 스테레오 시각 시스템이 등장하고 있으며, 가까운 장래에는 로봇에 시각 기능을 부여하는 것도 가능할 것으로 예측되고 있다.^[1] 3차원 거리정보를 실시간으로 처리하기 위하여 전용 하드웨어가 필요하다. 스테레오 정합 방법은 좌우 영상의 차이, 즉 변이(disparity)를 이용하여 3차원 정보를 획득하는 것으로 거리를 측정하는 방법으로 널리 사용하고 있다.^[2]

스테레오 정합에서의 중요한 과제는 좌우 영상간의

일치점을 찾는 것이다.^[3] 일치점을 찾는 방법으로는 필터링 된 영상의 경계점(edge), 연결성을 가진 경계선(boundary), 분할된 경계선(line segment), 모서리(corner) 등의 특징(feature)을 찾아서 좌우 영상의 일치점을 찾는 방법(feature-based method)과^[4-6] 영상의 밝기 정보를 이용하여 좌우 영상의 일치점을 찾는 방법(intensity-based method)^[7,9] 등이 있다. 이 방법들 중 반복적이고 정교화 된 계산 특성을 갖는 동적 프로그래밍 방법을 사용한 밝기 기반 스테레오 정합 방법이 하드웨어 구현에 가장 적합하다.^[8] 스테레오 정합 알고리즘의 하드웨어 구조는 Guerra & Kanade^[8] 등에 의해 제안되었으나, 이 구조에서 사용된 알고리즘^[7]은 스테레오 영상에서 중요한 정보인 폐색 정보가 비용함수에 포함되지 않고, 두 화소의 대응점 정합 비용은 인접한 대응점뿐만 아니라 3 또는 4단계 떨어진 곳의 대응점들에 대한 정합 비용도 고려되므로 하드웨어 구현에 적합하지 않다. 그리고, Guerra & Kanade^[8] 등이 제안한 스테레오 정합 구조에서는 극상선상의 화소 수만큼의 PE가 사용된다. 본 논문에서는 Cox et al^[9]이 제안한 스테레오 정합 알고리즘을 구현하는 하드웨어 구조를 개발하였다. Cox 등이 제안한 알고리즘에서는 폐색 정보에 대한 비용함수를 확률적으로 계산하고, 두 화소의 대응점 정합 비용은 두 화소의 밝기 값의 차와 인접한 대응점들의 비용과의 합에 의해 결정된다. 본 논문에서는 변이제약조건 R을 만족하면서 최소의 PE를 사용하는 MOEPE(Merged Odd-Even PE) 구조를 제안하였다. 예를 들어 변이 제약조건 R이 9인 경우, 512×512 영상에서 변이를 구하기 위해서는 기존의 구조^[8]에서는 512개의 PE가 사용되나 본 논문에서 제안한 구조에서는 9개의 PE가 사용되므로 VLSI로 구현이 가능하다.

II장에서는 동적 프로그래밍 방법에 기반 한 스테레오 정합 알고리즘을 기술하고, III장에서는 본 논문에서 제안한 스테레오 정합 하드웨어인 Odd-Even PE 구조와 이를 개선한 MOEPE(Merged Odd-Even PE) 구조의 동작원리를 설명한다. 그리고 IV장에서 결론을 보인다.

II. 동적 프로그래밍에 의한 스테레오 정합 알고리즘

이 장에서는 동적 프로그래밍을 사용하여 스테레오

영상으로부터 변이를 추출하는 스테레오 정합 알고리즘을 설명한다. 1절에서는 정합에 사용되는 비용함수(maximum likelihood cost function)를 정의하고, 2절에서는 동적 프로그래밍을 이용하여 최소비용 값과 최소비용 경로를 계산하는 방법을 설명한다.

1. 비용함수(Cost function)

용어를 먼저 정의하면, 2대의 카메라를 $s = \{1, 2\}$ 로 표기하고, Z_s 는 각 카메라의 극상선을 따라 측정된 값의 집합을 의미한다. 집합 Z_s 는 $Z_s = \{z_{s,i}\}_{i=0}^{M_s}$ 이다. 여기서 M_s 는 카메라 s로부터 측정된 값의 총 수이며, $z_{s,i}$ 는 밝기 값으로 공간상의 한 점 x_k 로부터 첫 번째 카메라에서 측정된 밝기 값을 z_{1,i_1} 로 나타내고, z_{2,i_2} 는 두 번째 카메라에서 측정된 밝기 값을 의미한다. $z_{s,0}$ 는 dummy measurement로 대응되는 점이 없다는 것을 의미하고, M_{i_1,i_2} 는 z_{1,i_1} 과 z_{2,i_2} 이 상응한다는 것을 의미한다. $M_{i_1,0}$ 는 첫 번째 카메라에서는 공간상의 한 점이 측정되었지만, 두 번째 카메라에서는 폐색되었다는 것을 의미한다.

공간의 한 점 x_k 로부터 발생된 대응 쌍 M_{i_1,i_2} 이 존재할 가능성(likelihood)은 다음과 같이 정의된다.

$$\Lambda(M_{i_1,i_2} | x_k) = \left(\frac{1 - P_D}{\phi} \right)^{\delta_{i_1,i_2}} [P_D p(z_{1,i_1} | x_k) \times P_D p(z_{2,i_2} | x_k)]^{1 - \delta_{i_1,i_2}} \quad (1)$$

단, 폐색 영역이면 $\delta_{i_1,i_2} = 1$ 이 되고, 정합 영역이면 0이 된다. $p(z | x)$ 는 각 영상의 밝기 값 z_i 가 공간상의 점 x 로부터 발생했다는 가정하의 확률밀도 함수를 나타내며, ϕ 는 카메라의 시각범위, P_D 는 3차원 정보가 영상 평면에 투영될 확률을 나타낸다. 측정치 $z_{s,i}$ 가 공간의 참값 z 에 대해 정규 분포되었다면,

$$p(z_{s,i} | x_k) = |(2\pi)^d S_s|^{-\frac{1}{2}} \times \exp \left\{ -\frac{1}{2} (z_{s,i} - z)' S_s^{-1} (z_{s,i} - z) \right\} \quad (2)$$

여기서, d 는 측정값 $z_{s,i}$ 의 차원, S_s 는 $z_{s,i}$ 의 공분산 행렬이다. z 의 실제 값을 모르기 때문에 대응 쌍 M_{i_1,i_2} 의 측정치로부터 얻어진 최우 추정량 \hat{z} 로 z

를 근사화 시킨다.

$$z \approx \hat{z} = \frac{S_j}{S_{i_1} + S_{i_2}} \times z_{1,i_1} + \frac{S_j}{S_{i_1} + S_{i_2}} \times z_{2,i_2} \quad (3)$$

모든 쌍들에 대한 likelihood는 다음과 같다.

$$L(\gamma) = p(Z_1, Z_2 | \gamma) = \prod_{z_{i_1, i_2} \in \gamma} \Lambda(M_{i_1, i_2} | x) \quad (4)$$

여기서 γ 는 가능한 대응 쌍의 집합을 의미하고, $L(\gamma)$ 을 최대화시키는 것은 다음 식과 동가이다.

$$\begin{aligned} \min J(\gamma) &= \min [- \ln(L(\gamma))] \\ &= \min \sum_{z_{i_1, i_2} \in \gamma} \left\{ \delta_{i_1, i_2} \ln \left(\frac{P_D^2 \phi}{(1 - P_D)} \frac{1}{|(2\pi)^d S|^{1/2}} \right) \right. \\ &\quad \left. + (1 - \delta_{i_1, i_2}) \left[\frac{1}{4} (z_{1, i_1} - z_{2, i_2})^T S^{-1} (z_{1, i_1} - z_{2, i_2}) \right] \right\} \end{aligned} \quad (5)$$

여기서 첫 번째 항은 좌측 혹은 우측 시각에서의 패색 비용이고 두 번째 항은 두 화소가 대응점이 되었을 때의 비용(정합 비용)을 나타낸다.

2. 동적 프로그래밍에 의한 스테레오 정합 알고리즘

좌우 영상의 극상선(epipolar line)의 화소 값들에 대하여 식 (5)를 만족하는 대응 쌍들을 구하기 위하여 동적 계획법이 사용된다. 동적 계획법에 의한 스테레오 정합은 극상선의 화소 값에 대하여 그림 1과 같이 탐색 영역을 만들고, 탐색 영역의 시작점에서부터 목표점에 이르는 최소비용을 구함으로써 좌우 영상을 정합 시키는 것이다. 정합비용이 작다는 것은 좌우 영상의 화소 밝기의 차이가 작다는 것을 의미한다. 동적 프로그래밍을 이용하여 시작점에서 목표점까지 최소비용을 구한 후, 최소비용을 역 추적하여 최소비용 경로를 구하고, 이로부터 변이(disparity)를 구한다. 알고리즘 1은 Cox et al^[9]이 제안한 방법으로, 좌우 영상의 극상선의 화소 값으로부터 최소비용과 최소비용 경로 정보를 저장하는 알고리즘이다.

$$\text{Occlusion} = \left\lfloor \ln \left(\frac{P_{D_s}^2 \phi}{(1 - P_{D_s})} \frac{1}{|(2\pi)^d S_s^{-1}|^{1/2}} \right) \right\rfloor$$

알고리즘 1. 최소비용 경로를 구하는 알고리즘의 의사 코드

for (i = 1; i ≤ N; i++) { C(i, 0) = i * Occlusion }

```

for (j = 1; j ≤ M; j++) { C(0, j) = j * Occlusion }
for (i = 1; i ≤ N; i++) {
for (j = 1; j ≤ M; j++) {
cost1 = C(i-1, j-1) + c(z1, i1, z2, i2);
cost2 = C(i, j-1) + Occlusion;
cost3 = C(i-1, j) + Occlusion;
C(i, j) = cmin = Min(cost1, cost2, cost3);
if(cost1 == cmin) P(i,j) = 0
if(cost2 == cmin) P(i,j) = 1
if(cost3 == cmin) P(i,j) = 2
}
}
    
```

Algorithm 1. Pseudo-code of the algorithm to find the minimum mcost

그림 1에서 컬럼(column)은 왼쪽에서부터 순서대로 우 영상의 화소 값을 나타내고 로우(row)는 위에서 아래로 순서대로 좌 영상의 화소 값을 나타낸다. 각 요소는 좌 영상에서 극상선의 한 화소(z_{1, i_1})와, 우 영상에서 극상선상의 한 화소(z_{2, i_2})의 쌍을 나타낸다. 검정으로 표시된 요소는 좌 영상에서 4번째 화소와 우 영상에서 6번째 화소 쌍에 해당된다. 계산 순서는 첫째 열에 대하여 좌측에서부터 우측으로 각 화소 쌍에 대하여 최소비용을 계산하고 완료되면 다음 열에 대하여 순서대로 계산하며 마지막 열까지 반복 수행한다.

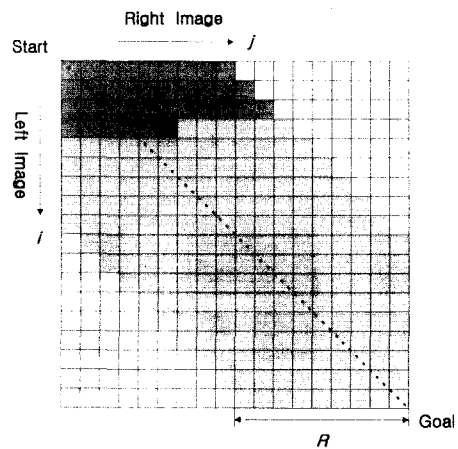


그림 1. 변이제약조건이 R인 경우에 최소비용을 구하기 위한 탐색범위

Fig. 1. The search ranges to find the minimum matching cost under the disparity constraint R.

그림 1에서 짙은 회색으로 표시된 화소 쌍들은 최소비용이 구하여진 화소 쌍들이고 검정으로 표시된 화소 쌍은 계산이 이루어질 화소 쌍이다. 그리고 옅은 회색 사각형들은 아직 최소비용이 계산되지 않은 화소 쌍들을 나타낸다. 그림 1과 같이 모든 화소 쌍에 대하여 최소비용을 구하지 않고 대각선에서 일정한 범위 내의 화소 쌍에 대해서만 탐색 영역으로 정의되는데, 이와 같이 변이가 일정한 값으로 한정 되어있다고 가정하는 것을 변이제약조건(disparity constraint)이라 한다. 좌 영상의 극상선 상의 i 번째 화소는 우 영상에서 $i-R$ 번째에서 $i+R$ 번째의 화소들과 정합이 이루어진다. 변이가 일정한 범위 내에 존재한다는 조건에 의해서 극상선 길이 N 과 M 전체에서 최소비용 경로를 구하지 않으므로 시간을 $O(NM)$ 에서 $O(NR)$ 로 줄일 수 있다.

그림 2(a)에서 회색으로 된 화소 쌍들은 최소비용이 구하여진 화소 쌍들이고 좌 영상의 i 번째 화소와 우 영상의 j 번째 화소(■로 표시된 화소 쌍)는 최소비용 $C(i, j)$ 이 계산될 화소 쌍이다. 그림 2(b)에서 보는 것과 같이 화소 쌍 (i, j) 에 도달하는 경로는 대각방향인 화소 쌍 $(i-1, j-1)$ 에서 화소 쌍 (i, j) 로의 경로와

화소 쌍 $(i, j-1)$ 에서 화소 쌍 (i, j) 로의 수평방향 경로, 그리고 화소 쌍 $(i-1, j)$ 에서 화소 쌍 (i, j) 로의 수직방향 경로가 존재한다. 각각의 경로에 대한 비용은 알고리즘 1에서 $cost1, cost2, cost3$ 으로 계산되고, 이는 정합 점에 대한 비용, 왼쪽 폐색 비용 및 오른쪽 폐색 비용을 나타낸다. $cost1$ 에서 $c(z_{1,i}, z_{2,i})$ 는 좌 영상의 i 번째 화소와 우 영상의 i 번째 화소의 정합비용으로 식 (5)의 두 번째 항으로 계산된다. 3개의 가능한 경로에서 가장 적은 비용을 갖는 경로가 선택되고, 최소비용 경로를 구하기 위하여 알고리즘 1에서 $P(i, j)$ 에 최소비용이 발생한 경로, 즉 최소비용 경로 정보를 기록한다. 마지막 화소에 도달할 때까지 최소비용이 구해지면, 최소비용경로정보 $P(i, j)$ 를 역 추적하여 최소비용 경로와 변이를 구한다.

III. 스테레오 정합을 위한 하드웨어의 구조

이 장에서는 본 논문에서 제안한 동적 프로그래밍을 사용하는 스테레오 정합 하드웨어의 구조와 동작에 대해 기술한다. 1절에서는 스테레오 정합 알고리즘을 시스틀릭 어레이 구조의 동작으로 변환하는 방법에 대해 설명하고, 2절에서는 제안한 Odd-Even PE 구조 및 동작을 설명한다. 그리고 3절에서는 이를 개선한 MOEPE(merged Odd-Even PE) 구조에 대해 기술한다.

1. 시스틀릭 어레이(Systolic Array) 구조

동적 프로그램을 사용하는 스테레오 정합 알고리즘(알고리즘 1)에서 화소 쌍들이 계산되는 순서는 그림 3(a)와 같이 좌측 화소 쌍에서부터 우측 화소 쌍까지 각 열에 대해 순차적으로 반복 수행한다. 주어진 화소 쌍 (i, j) 에 대하여 최소비용을 계산하기 위해서는 인접한 3개 화소 쌍의 최소비용, $(i-1, j)$, $(i, j-1)$, $(i-1, j-1)$ 이 결정되어 있어야 한다. 그림 3(b)와 같이 계산 순서를 바꾸어도 이 조건을 만족시킨다. 그림 3(b)를 살펴보면 2번째와 3번째 계산되는 화소 쌍들은 독립적으로 최소비용을 계산할 수 있다. 또한 4번째, 5번째, 6번째 계산되는 화소 쌍들도 서로 독립적으로 계산이 이루어질 수 있다. 동시에 여러 개의 최소비용을 계산할 수 있으면 그림 3(b)에서 그림 3(c)와 같이 계산순서를 변환할 수 있다. 비용 계산은 모든 화소 쌍에서 같은 방법으로 이루어지므로 이를

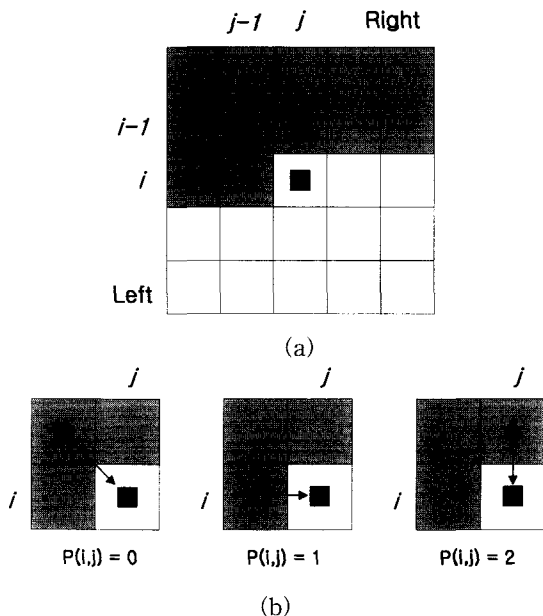


그림 2. (a) 최소비용 경로 탐색 (b) 화소 쌍 (i, j) 로 가능한 세 가지 경로

Fig. 2. (a) Searching the minimum matching cost (b) Three possible paths reaching to the pixel pair (i, j) .

동일한 PE(Processing Element)로 구현할 수 있다.

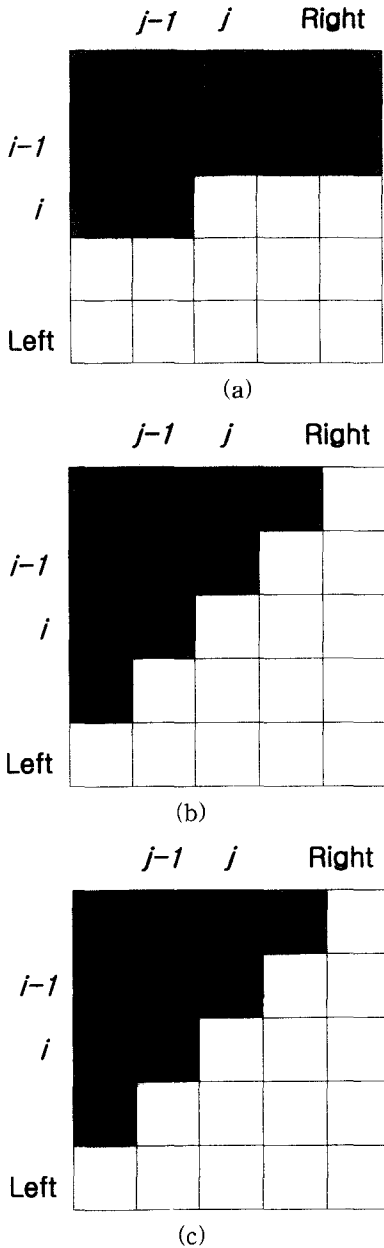


그림 3. 화소 쌍의 계산 순서 (숫자는 계산 순서를 나타낸다.) (a) 동적 프로그래밍에서의 계산 순서 (b) 계산 순서의 재배열 (c) 여러 개의 PE를 사용할 경우의 계산 순서

Fig. 3. The calculation order of the pixel pairs (Numbers denote the order of the calculation.) (a) The calculation order in the conventional dynamic programming method (b) The reordering of the calculation (c) The calculation order when the multiple PEs are available.

그림 3(c)와 같은 방법은 그림 4와 같은 데이터 흐름을 갖는 시스톨릭 어레이 구조로 구현될 수 있다. 화살표는 데이터의 흐름을 나타낸다. 그림 4에서 동일한 점선 상의 PE들은 계산이 동시에 이루어지고, 계산 결과는 다음 점선 상의 PE들의 입력으로 사용된다. 첫 번째 클럭 CLK1에서는 PE1이 계산이 이루어지고 두 번째 클럭 CLK2에서는 PE2와 PE5가 계산되어진다. 그리고 마지막 클럭 CLK7이 인가되면 PE16의 계산이 수행되어 최소비용이 구해진다. 각 PE는 해당되는 클럭에서만 동작하므로 그림 4와 같이 4×4 화소 쌍에서의 계산은 CLK4에서 4개의 PE가 동시에 활성화된다. 그러므로 이 경우에는 4개의 PE로 계산이 가능하다. 짝수 회색으로 표시된 PE는 홀수 클럭에 활성화되는 PE이고, 옅은 회색으로 표시된 PE는 짝수 클럭에 활성화되는 PE이다. 짝수 클럭에 동작하는 PE들을 Even PE, 그리고 홀수 클럭에 동작하는 PE들을 Odd PE라 정의한다.

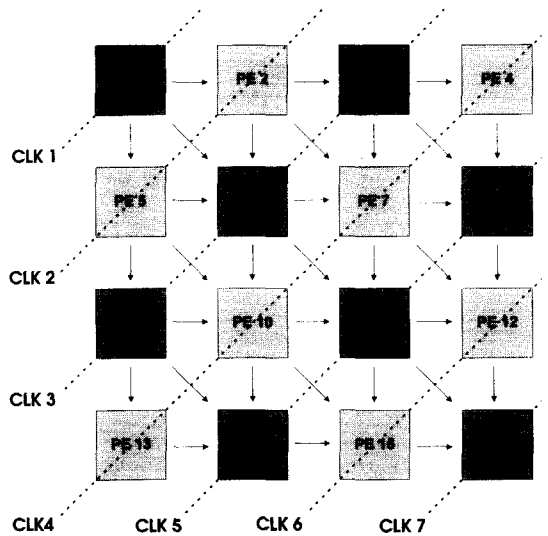


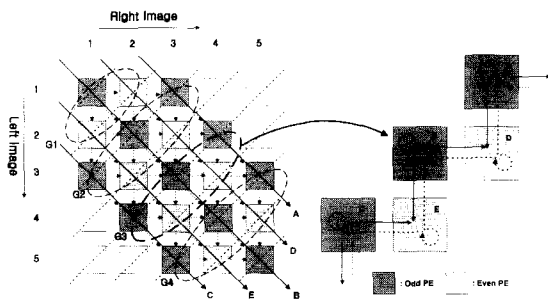
그림 4. 시스톨릭 어레이 구조

Fig. 4. The structure of the systolic array.

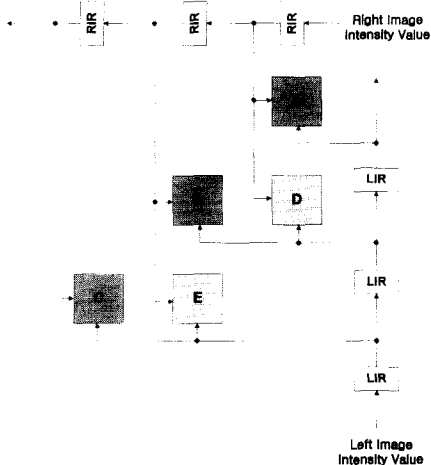
2. Odd-Even PE 구조

그림 5(a)의 왼쪽 그림은 변이제약조건 R이 3인 경우(변이가 +2에서 -2까지 변할 때)의 시스톨릭 구조를 보여준다. 동시에 계산이 이루어지는 PE의 수는 홀수 클럭에는 3개이고, 짝수 클럭에는 2개이다. 이 절에서는 데이터의 흐름을 간단히 하기 위하여 그림 5(a)오른쪽에는 홀수 클럭에 해당하는 요소 쌍을 계산하는 Odd PE와 그리고 짝수 클럭에 해당하는 요소

쌍을 계산하는 *Even PE*로 이루어진 *Odd-Even PE* 구조를 보여준다. 그림 5(a)의 왼쪽 그림에서 $G1, G2, \dots$, 등은 연속된 홀수 클럭과 짝수 클럭에서 동작하는 *Odd PE*와 *Even PE*로 이루어진 그룹을 나타낸다. 그림 5(a)의 왼쪽 그림에서 화살표로 표시된 같은 대각선 A, B, C, D, E 상의 *PE*들은 *Odd-Even PE* 구조에서 각각 *PE A, PE B, PE C, PE D, PE E*에 의하여 계산이 이루어진다. *Odd-Even PE* 구조에서는 $G1$ 의 계산을 수행하고, $G2, G3, \dots$ 에서 이루어지는 계산을 순차적으로 수행한다. *Odd-Even PE* 구조는 *Odd PE*가 먼저 계산을 수행한 후 *Even PE*가 계산을 수행하는 계산과정을 반복한다. *PE*들은 계산을 수행한 후 자신의 결과를 *PE*에 저장하여 인접한 *PE*에 보내준다.



(a)



(b)

그림 5. (a) *Odd-Even PE* 구조와 데이터 흐름 (b) 화소값을 고려한 *Odd-Even PE* 구조

Fig. 5. (a) The architecture of the *Odd-Even PE* and the data flows (b) The architecture of the *Odd-Even PE* with the pixel

values.

Odd-Even PE 구조에서 실선은 *Odd PE*에서 연산된 결과가 사용되는 *PE*를 가리키고, 점선은 *Even PE*에서 연산된 결과가 사용되는 *PE*를 가리킨다. $G3$ 에서의 *Odd-Even PE*의 계산 과정을 살펴보자. *Odd PE*가 계산되기 위해서는 $G2$ 의 *Even PE*의 연산 결과와 $G2$ 의 *Odd PE*의 연산 결과가 요구된다. 이 데이터는 *Odd-Even PE* 구조에서 각각 *Even PE*에서의 데이터와 *Odd PE* 자신이 저장하고 있는 데이터를 사용하면 된다. *Even PE*가 계산되기 위해서는 $G3$ 의 *Odd PE*의 연산 결과와 $G2$ 의 *Even PE*의 데이터가 요구된다. 이 데이터는 *Odd-Even PE* 구조에서 각각 *Odd PE*의 계산 결과와 *Even PE* 자신이 저장하고 있는 연산 결과를 사용하면 된다.

각 *PE*에서 최소비용을 계산하기 위해서는 *PE*간의 연산결과 외에도 식 (5)의 두 번째 항의 정합비용을 계산하기 위한 화소 쌍의 밝기 값이 주어져야 한다. 그림 5(b)는 극상선 상의 화소의 밝기 값이 고려된 *Odd-Even PE* 구조를 보여준다. 예를 들어 그림 5(a)에서 *Odd-Even PE*가 $G3$ 의 계산을 수행할 때, *PE A*는 우 영상의 4번째 화소 값과 좌 영상의 2번째 화소 값이 필요하고, *PE B*는 우 영상의 3번째 화소 값과 좌 영상의 3번째 화소 값이 필요하다. 또한 *PE C*는 우 영상의 2번째 화소 값과 좌 영상의 4번째 화소 값이 필요하다. *PE A, PE B*와 *PE C*에서 필요한 밝기 값은 우 영상의 4번째, 3번째, 2번째 그리고 좌 영상의 2번째, 3번째, 4번째 화소 값이 필요하므로 그림 5(b)와 같이 *LIR*(Left image Intensity Register)와 *RIR*(Right image Intensity Register)를 사용하여 화소 값을 저장하고, 레지스터의 값은 홀수 클럭 마다 인접한 *PE*로 이동한다. *PE D*와 *PE E*에서 필요한 화소 값은 각각 우 영상의 4번째와 3번째 그리고 좌 영상의 3번째와 4번째이다. 짝수 클럭에는 *LIR*과 *RIR*의 밝기 값이 유지되므로 그림 5(b)와 같이 연결하면 된다.

3. MOEPE(merged *Odd-Even PE*) 구조

그림 5에 설명한 *Odd-Even PE* 구조는 홀수 클럭에는 *Even PE*가 계산을 하지 않고 짝수 클럭에는 *Odd PE*가 계산을 하지 않으므로 이를 병합하여 최소의 *PE* 수로 구성되어지는 *MOEPE* 구조를 제안

하였다. 그림 6(a)의 왼쪽 그림은 *Odd-Even PE* 구조를 나타내고 오른쪽은 이를 개선한 *MOEPE (Merged Odd-Even PE)* 구조를 나타낸다. *MOEPE* 구조에는 계산에 필요한 최소한의 *PE*, 즉 변이제약조건 R 개의 *PE*들로 구성된다. *Odd-Even PE* 구조에서 2개의 *PE*(*Even PE*와 *Odd PE*)가 *MOEPE* 구조에서는 그림 6과 같이 하나의 *PE*로 병합된다. 가운데 프로세싱 엘리먼트인 *PE B*는 병합이 이루어지지 않고, 가운데 *PE B*를 중심으로 오른쪽의 *PE*는 수직방향으로 *Odd PE*와 *Even PE*가 병합이 이루어지고, 아래의 *PE*는 수평방향으로 *Odd PE*와 *Even PE*가 병합이 이루어진다. 그림 6(a)의 *Odd-Even PE* 구조의 *PE A*와 *PE D*의 연산은 *MOEPE* 구조의 *PE U1*에서, *Odd-Even PE* 구조의 *PE C*와 *PE E*의 연산은 *MOEPE* 구조의 *PE L1*에서 이루어진다. 즉, *PE U1*은 홀수 클럭에는 *PE A*에서 이루어지는 연산을, 짝수 클럭에는 *PE D*에서 이루어지는 연산을 수행하고, *PE L1*은 홀수 클럭에는 *PE C*에서 이루어지는 연산을, 짝수 클럭에는 *PE E*에서 이루어지는 연산을 수행한다. 그리고 *PE M*은 *PE B*의 연산을 홀수 클럭에만 수행하고 짝수 클럭에는 연산을 수행하지 않는다.

그림 6(a)의 *MOEPE* 구조에서 데이터의 흐름을 살펴보자. *PE U1*은 각각 *PE A*와 *PE D*의 연산을 수행하는데, 홀수 클럭에 계산되어진 연산 결과(그림에서 실선의 데이터)는 다음 짝수 클럭에 *PE U1*에서 (*PE U1*이 *PE D*의 연산을 수행할 때) 사용되어질 뿐만 아니라 다음 홀수 클럭에 *PE U1*에서도(*PE U1*이 *PE A*의 연산을 수행할 때) 사용이 된다. *PE L1*에서도 *PE U1*에서와 동일한 동작 특성을 갖는다. *PE M*은 홀수 클럭에만 연산이 이루어지고, 계산 결과를 인접한 *PE*로 보내고 자신의 *PE M*에 저장한 후 다음 홀수 클럭에 저장된 계산 결과를 사용한다.

각 *PE*에서 최소비용을 계산하기 위해서는 정합비용을 계산하기 위한 화소 쌍의 밝기 값이 주어져야 한다. 그림 6(b)는 화소 값이 고려된 *MOEPE*의 데이터 흐름을 보여준다. 화소 값은 *LIR*와 *RIR*를 사용하여 저장하고 레지스터의 값은 홀수 클럭마다 시프트 된다. 그림 6(b)에서 *PE A*와 *PE D*는 우 영상의 화소 값 $R3$ 을 공유하지만 *PE A*는 *PE D*보다 한 클럭 지연된 좌 영상의 화소 값 $L1$ 과 $L2$ 를 사용한다. *PE A*와 *PE D*의 연산을 수행하는 *PE U1*은 우 영상의 화소

값 $R3$ 를 짝수 클럭과 홀수 클럭 모두에서 사용하지만, 좌 영상의 화소 값은 홀수 클럭에는 $L1$ 값을 사용하고 짝수 클럭에는 $L2$ 값을 사용한다. 그러므로 그림 6(b)와 같이 좌 영상의 화소 값은 *MUX*를 사용하여 선택한다. *PE C*와 *PE E*는 좌 영상의 화소 값 $L3$ 를 공유 하지만 *PE C*는 *PE E*보다 한 클럭 지연이 된 우 영상 화소 값 $R1$ 과 $R2$ 를 사용한다. *PE C*와 *PE E*의 연산을 수행하는 *PE L1*은 좌 영상의 화소 값 $L3$ 를 짝수 클럭과 홀수 클럭 모두에서 사용하지만, 우 영상의 화소 값은 *MUX*를 사용하여 선택한다. *PE M*은 홀수 클럭에만 동작하므로 인가된 화소 값 $L2$ 와 $R2$ 를 사용하면 된다.

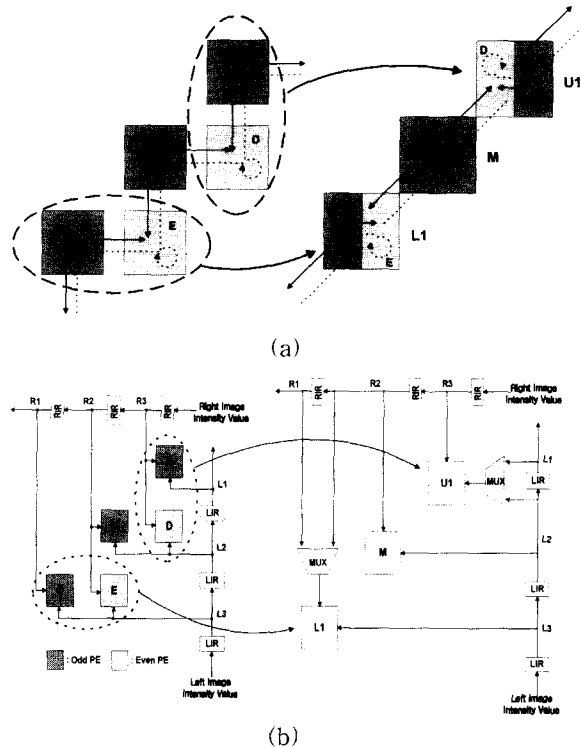


그림 6. (a) *Merged Odd-Even PE* 구조와 데이터 흐름 (b) 화소 값을 고려한 *MOEPE* 구조
 Fig. 6. (a) The architecture of the *MOEPE (Merged Odd-Even PE)* and the data flows (b) The architecture of the *MOEPE* with the pixel values.

좌영상과 우영상을 이용하는 스테레오 정합에서의 변이는 0와 양의 값을 갖거나 0와 음의 값을 갖는다. 그러므로, 그림 6(a)에서 *PE M*과 *PE U1*만 사용하거나 *PE M*과 *PE L1*만 사용하여 변이를 구한다. 변이제약조건이 R 인 경우 사용되는 *PE* 수는 $(R-1)$

$/2+1$ 개의 PE로 구현이 가능하다. 제안된 구조는 VHDL로 기술하였고, Synopsys 설계 환경에서 시뮬레이션을 수행하였다.

IV. 결 론

본 논문에서는 동적 프로그래밍에 기반 한 스테레오 정합 알고리즘을 구현하는 새로운 하드웨어구조를 제안하고, 설계한 후 시뮬레이션을 수행하였다. 제안된 MOEPE(Merged Odd-Even PE) 구조는 많은 연산량을 요구하는 스테레오 정합에서 깊이 정보를 실시간으로 처리하는 것을 가능하게 하였다.

제안된 MOEPE 구조의 스테레오 정합 하드웨어는 변이제약조건에 해당하는 PE를 사용하므로, 512×512 영상에서 변이제약조건이 9인 경우 기존의 제안된 구조에 비해 필요한 PE의 수를 약 25배정도 줄일 수 있다.

참 고 문 헌

- [1] R. C. Jain and A. K. Jain, "Review of the NSF Range Sensing Workshop," *Computer Vision and Pattern Recognition Conf.*, Ann Arbor, MI, May 1988.
- [2] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Addison Wesley, pp. 68-71, 1993.
- [3] D. Marr and T. Poggio, "Cooperative computation of stereo disparity," *Science*, Vol. 194, pp. 283-287, 1976.
- [4] Y. S. Sim and S. H. Park "A Study on the Feature Points Detection for Stereo Matching," *Journal of Research Institute of Industrial Sciences*, Vol. 30, pp. 27-33, 1990.
- [5] N. M. Nasrabadi and C. Y. Choo, "Hopfield network for stereo vision correspondence," *IEEE Trans. on Neural Networks*, Vol. 3, No. 1, Jan. 1992.
- [6] T. Kanade and M. Okutomi, "A stereo matching algorithm with an adaptive window: theory and experiment," *IEEE Trans on Pattern Anal. Machine Intell.*, Vol. PAMI-16, No. 9, pp. 920-932, Sep. 1994.
- [7] Y. Ohta and T. Kanade, "Stereo by intra- and inter-scanline search using dynamic programming," *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 7, No. 2, pp. 139-154, 1985.
- [8] C. Guerra and T. Kanade, "A Systolic Algorithm for Stereo Matching," *Proc. of International Workshop on Parallel Computing and VLSI*, Amalfi, Italy, May 1984.
- [9] I. J. Cox, S. L. Hingorani and S. B. Rao, "A Maximum likelihood stereo algorithm," *Computer Vision and Image Understanding*, Vol. 63, No. 3, pp. 542-567, May 1996.
- [10] N. Weste and D. J. Burr, "Dynamic Time Warp Pattern Matching Using an Integrated Multiprocessing Array," *IEEE Trans. on Computers*, Vol. c-32, No. 8, 1983.

저 자 소 개



韓 必 宇(正會員)

1997년 2월 경상대학교 전자재료공학과 졸업(학사). 1999년 2월 경상대학교 대학원 전자재료공학과 졸업(석사). 1999년 2월~8월 (주)아이피에스 주임연구원. 1999년 9월~2000년 2월 한국기계연구원 연구원. 2000년

2월~현재 (주)한국엑시스 연구원. 주 관심분야는 영상 처리 및 음성인식용 ASIC 설계, 가상현실 등



梁 榮 日(正會員)

1983년 2월 경북대학교 전자공학과 졸업(학사). 1985년 한국과학기술원 전기 및 전자공학과 졸업(석사), 1989년 한국과학기술원 전기 및 전자공학과 졸업(박사). 1990년~현재 경상대학교 전기전자공학부 부교수.

1994년 1월~1995년 1월 UC, Irvine 교환교수. 주 관심분야 VLSI&CAD, 영상신호처리 등