

論文 2000-37SC-2-7

실시간 스테레오 정합을 위한 스테레오 영상 정합 프로세서 설계 (Design of Stereo Image Match Processor for Real Time Stereo Matching)

金演宰*, 沈德善*

(Youn-Jae Kim and Duk-Sun Shim)

요 약

스테레오 영상(stereo image)이란 같은 물체나 장면을 담고있는 서로 다른 시점의 두 영상이며 스테레오 영상에서 깊이 정보를 얻어내는 것을 스테레오 비전(stereo vision)이라 한다. 스테레오 비전에서 가장 중요한 과정은 두 영상에서 서로 일치하는 점을 찾아내는 스테레오 정합(stereo matching)이다. 그러나, 스테레오 정합은 매우 많은 계산을 필요로 하기 때문에 실시간으로 정합하기 어렵다. 본 논문에서는 실시간으로 스테레오 정합을 처리할 수 있는 스테레오 영상 정합 프로세서(stereo image match processor: SIMP)를 설계하고 구현하였다. 이를 위해 슬라이딩 메모리(sliding memory)와 최소 선택 트리(minimum selection tree)를 제안하였고 파이프라인 구조(pipeline architecture)와 병렬 처리 기법을 이용하였다. SIMP의 입력은 64 그레이 레벨인 두 개의 64×64 스테레오 영상이고 출력은 최대 7의 값을 가지는 변이(disparity)와 12비트의 주소로 이들을 이용하여 64×64 변이도(disparity map)를 구성할 수 있다. SIMP는 약 240 프레임/초의 속도로 스테레오 영상을 처리할 수 있다.

Abstract

Stereo vision is a technique extracting depth information from stereo images, which are two images that view an object or a scene from different locations. The most important procedure in stereo vision, which is called stereo matching, is to find the same points in stereo images. It is difficult to match stereo images in real time because stereo matching requires heavy calculation. In this paper we design a digital VLSI to process stereo matching in real time, which we call stereo image match processor (SIMP). For implementation of real time stereo matching, sliding memory and minimum selection tree are presented. SIMP is designed with pipeline architecture and parallel processing. SIMP takes 64 gray level 64×64 stereo images and yields 8 level 64×64 disparity map by 3 bit disparity and 12 bit address outputs. SIMP can process stereo images with process speed of 240 frames/sec.

* 正會員, 中央大學校 電子電氣工學部

(Chung-Ang University, School of Electrical and Electronic Engineering)

※ 본 연구는 1997년도 교육부 학술연구 조성비(반도체 97-E-2009)에 의하여 연구되었음.

接受日字: 1998年8月20日, 수정완료일: 2000年2月11日

I. 서론

서로 다른 시점에서 같은 물체나 장면을 보여주는 스테레오 영상에서 깊이 정보를 얻어내는 것을 스테레오 비전이라 한다. 두 영상에서 서로 일치하는 대응점(correspondent point)을 찾아내는 것을 스테레오

오 정합이라 하는데 찾아진 점은 두 영상에서 서로 다른 좌표를 가지게 되며 두 좌표의 거리는 그 점의 변이(disparity)라 한다. 변이를 이용하면 그 점의 3차원 공간상의 깊이(depth)를 계산할 수 있다. 그러므로 스테레오 비전에서 가장 중요한 과정은 스테레오 정합이다.

스테레오 정합에서 대응점을 찾으려면 먼저 한 스테레오 영상에서 기준이 되는 정합점(matching point)을 선택하고 다른 영상에서 그 점과 대응될 가능성이 있는 후보점(candidate)들을 찾아 모든 후보점들과 정합점사이의 정합도(matching strength)를 구하고 가장 좋은 정합도를 갖는 후보점이 대응점이 된다. 그러므로 후보점과 정합점 사이의 정합도계산은 스테레오 정합에서 가장 중요한 문제 중의 하나이다. 지금까지 정합도를 계산하기 위한 여러 가지 방법이 제시되어 왔다. Marr-Poggio^[1]는 깊이 일관성(depth constancy)을 가정하였고 Pollard-Mayhew-Frisby (PMF)^[2]는 변이 기울기(disparity gradient)의 한계를 이용하여 정합의 정확도를 높여 하였다. Prazdny^[3]는 변이 유사 함수(disparity similarity function)를 이용하였다. 이웃한 점들의 정합 정보에서 도움을 얻어 일부 후보정합들의 정합도를 높여줌으로써 정합도의 신뢰도를 높인다는 것이 이러한 연구들의 공통점이다. 그러나 스테레오 정합은 많은 데이터를 처리해야 하므로 연산이 많아져서 실시간으로 스테레오 정합을 처리하는 데 어려움이 있다.

Hakkarainen-Lee^[6]는 소프트웨어를 이용하여 스테레오 정합을 계산하는 것은 비효율적이며 많은 시간이 걸린다는 것을 지적하고 아날로그 VLSI를 이용하여 실시간으로 정합을 하려는 시도를 하였다. 그들이 설계한 아날로그 VLSI는 빠른 속도를 보여 주었으나 스테레오 정합의 일부 연산을 담당하는 AVD (absolute value of difference)연산기만 구현하였으므로 완전한 스테레오 정합을 실시간으로 구현하였고 할 수 없으며 칩의 연산 결과를 정확하게 예측할 수 없는 아날로그 VLSI로서의 한계를 가지고 있었다. 스테레오 정합과 유사한 알고리즘을 사용하는 운동추정(motion estimation)에서는 VLSI설계에 관한 연구가 많이 되고 있다^[7-9]. Hsieh-Lin^[7]은 탐색영역과 기준영역의 영상데이터를 연속적으로 입력하면서 병렬처리로 block matching을 할 수 있도록 해주는 쉬프트 레지스터 어레이(shift register array:SRA)를 설계

하였다. SRA는 정합용 프로세서의 데이터 이용효율을 높일 수 있게 하였다. 그러나 SRA는 기준영역을 변경시킬 때마다 탐색영역의 영상데이터를 새로 입력해야 하기 때문에 여전히 데이터 이용효율이 낮을 뿐 아니라 다른 프로세서의 도움을 받아야만 전체 영상을 정합할 수 있다.

본 논문에서는 완전한 스테레오 정합을 실시간으로 처리하는 스테레오 영상 정합 프로세서(Stereo Image Match Processor : SIMP)를 설계하고 FPGA를 이용하여 구현하였다. SIMP는 스테레오 영상 데이터를 순차적으로 입력받아 스테레오 정합을 수행하여 변이도를 출력할 수 있다. 이를 위해 슬라이딩 메모리(sliding memory : SM)와 최소 선택 트리(minimum selection tree : MST)를 설계하였고 파이프라인 구조(pipeline architecture)와 병렬 처리 기법을 이용하였다. 슬라이딩 메모리는 기준영역과 탐색영역을 바꿀 때마다 데이터를 새로 채워야 하는 SRA의 단점을 보완하여 데이터 이용 효율을 극대화하였으며 MST는 우수정합선택을 빠르게 할 수 있도록 하였다.

논문은 다음과 같이 구성되었다. 2절에서는 스테레오 정합의 기본개념과 스테레오 정합 알고리즘들을 다루고 실시간 정합과 칩 설계에 알맞도록 계산과정을 줄인 알고리즘을 설명하였다. 3절에는 SIMP의 회로구조를 설명하였다. SIMP의 구성을 설명하고 파이프라인을 적용한 SSD 연산장치와 슬라이딩 메모리와 최소 선택 트리(MST)의 구조를 다루었다. 4절에서는 SIMP를 이용한 스테레오 정합 결과를 보여주고 하드웨어 구현을 통하여 성능과 알고리즘을 검증하였다. 그리고 5절에서는 연구결과를 설명하고 앞으로 개선되어야 할 과제를 제시하였다.

II. 스테레오 정합 알고리즘

스테레오 정합은 기준영상과 탐색영상으로 이루어진 스테레오 영상을 대상으로 하며 기준영상에서 정합점을 선택하고 대응하는 점을 탐색영상에서 찾는다. 한 점의 정합을 위해 탐색되는 영역을 탐색영역이라 한다. 일반적인 스테레오 영상을 정합하는 경우에는 기준영상의 한 점을 정합하기 위해 탐색영상의 전체 영역을 탐색해야 하므로 정합과정이 복잡하다. 그러므로 대부분의 스테레오 정합 알고리즘들은 스테레오 영상에

에피폴라(epipolar) 제한조건을 두어 탐색영역을 줄임으로써 정합과정을 단순화시킨다. 에피폴라 제한조건이란 스테레오 영상의 두 영상은 높이방향(Y축)이 일치하여야 한다는 조건이다. 에피폴라 제한조건을 만족하는 스테레오 영상에서는 기준영상에서 임의의 Y좌표에 나타난 점은 탐색영상의 같은 Y좌표에서만 나타날 수 있다. 그러므로 에피폴라 제한조건을 적용하지 않은 경우의 스테레오 정합은 탐색영상 전체가 탐색영역이 되지만 에피폴라 제한조건을 만족하는 스테레오 영상의 경우에는 탐색영상에서 기준영상과 높이가 같은 일부영역이 탐색영역이 된다.

스테레오 정합 알고리즘들은 유사한 정합 계산 과정을 가지고 있는데 정합 계산 과정은 정합도(matching strength) 계산 과정과 우수 정합 선택 과정으로 나누어 질 수 있다.

정합도 계산에서 정합점과 후보정합들의 정합도를 얻는다. 정합도란 기준영상의 정합점과 탐색영역의 점들의 닮은 정도이며 정합도가 계산된 탐색영역의 점들을 후보정합(candidate match)이라 한다. 정합도는 일반적으로 SSD(sum of squared difference)와 CC(cross correlation)를 이용한다. SSD는 두 영역의 비교되는 점들간의 밝기 차이를 제공한 후에 모두 합한 값으로 오차의 크기를 측정하는 것이며 CC는 두 영역의 닮은 정도를 상호결합도(correlation)로 구한 값이다. 우수 정합 선택 과정은 정합도 계산 과정을 마친 후보정합들 중에서 우수 정합을 선택하는 과정이다. SSD방법으로 정합도를 계산한 경우에는 결과값이 작을수록 정합도가 높다.

정합의 정확성을 높이기 위해 우수 정합을 선택할 때 정합하려는 점의 후보정합만을 기준으로 하지 않고 이웃점들의 후보정합이나 정합결과를 참고로 삼을 수도 있다. 일반적인 물체의 스테레오 영상인 경우 두 점간의 좌표거리와 두 점의 변이차의 비인 변이기울기가 한계값을 가지게 되는 물리적인 성질이 있다. 그러므로 후보정합들과 이전의 정합결과와 변이기울기를 계산하여 한계값이내의 변이기울기를 갖는 후보정합들을 우선적으로 선택하도록 한다. 이를 로컬서포트(local support)라 하며 대부분의 스테레오 정합 알고리즘들이 로컬서포트 과정을 포함하고 있다.

대부분의 스테레오 정합 알고리즘들은 정합도 계산과 우수 정합 선택과정을 여러 번 반복 계산하여 최적의 해를 구하려하므로 이를 VLSI로 구현하기에 적합하지

않다. 운동추정 분야에서는 정합도 계산과 우수 정합 선택을 한 번만 계산하고 정합을 하는 방식으로 VLSI를 구현하고 있으며 그 성능도 검증되어 있다^[12]. 그러므로 본 논문에서도 VLSI구현을 위하여 SSD 정합도 계산과 우수 정합을 선택하는 과정을 되풀이하지 않고 한 번씩만 거치도록 하였으며 스테레오 정합의 특성을 고려해 정합도 계산 후 로컬 서포트 과정을 거치도록 하였다.

III. 스테레오 영상 정합 프로세서의 구조

스테레오 영상 정합 프로세서(Stereo Image Match Processor : SIMP)는 실시간으로 스테레오 영상을 정합하여 변이도를 출력하는 VLSI이다. 입력은 64 그레이 레벨(gray level)인 64 × 64 스테레오 영상이며 16 클럭사이클(clock cycle)마다 한번씩 기준영상과 탐색영상을 동시에 입력받도록 하였다. SIMP는 3 × 3 사각영역에 대한 SSD방식으로 정합도를 계산하며 탐색영역의 크기는 3×17 이고 최대변이는 7이다. 정합결과는 4비트 변이와 12비트의 주소로 출력된다. 12비트 주소는 함께 출력되는 변이의 기준영상 좌표로 변이도를 구성하기 편리하게 해준다. 12비트 주소 중에서 6비트 MSB(most significant bit)는 Y좌표이고 나머지 6비트 LSB(least significant bit)는 X좌표이다. 변이의 출력은 3비트이며 정합오류를 나타내는 1비트 출력이 있다.

SSD방식의 정합도는 다음과 같다.

$$SSD(k) = \sum_{i=1}^N \sum_{j=1}^N |S(i+k, j) - R(i, j)|^2 \quad (1)$$

$R(i, j)$ 는 $N \times N$ 정합윈도우(matching window)이며 $S(i+k, j)$ 는 탐색영역이다. M 은 최대 변이의 크기이고 $SSD(k)$ 는 k 번째 후보정합의 정합도이며 k 는 $1 \leq k \leq M$ 의 범위를 갖는다. 식 (1)에 알 수 있듯이 SSD를 얻기 위해서는 N^2 번의 뺄셈과 N^2 번의 제곱연산과 N^2-1 번의 덧셈연산을 해야하므로 매우 많은 계산을 해야하며 많은 데이터입력을 필요로 한다. 그러므로 빠른 연산을 위해 특별한 회로구조를 이용하여야 한다.

SSD는 3단 파이프라인 구조인 SSD 연산장치(SSD calculator)에서 계산된다.

그럼 1은 SIMP의 구성을 보여준다. SIMP는 두 개의 슬라이딩 메모리(SM)와 SSD 연산장치와 로컬서포

트 필터(local support filter)와 선택기(selector)로 구성되어 있다. 탐색영상 슬라이딩 메모리는 정합에 필요한 15개 후보정합들의 영상데이터를 공급하고 기준영상 슬라이딩 메모리는 기준영역의 데이터를 공급하며 슬라이딩 메모리의 동작은 슬라이딩 동기신호에 의해 제어된다. 15개의 후보정합들의 정합도는 SSD연산장치에서 순차적으로 계산되어 로컬서포트 필터의 래치에 기억된다. 정합도가 모두 계산된 후 로컬서포트 필터와 우수정합 선택기의 연산을 통해 변이도를 얻어낸다.

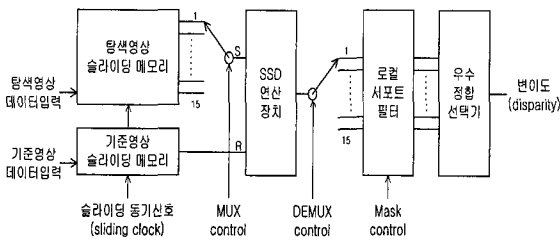


그림 1. SIMP의 블럭도
Fig. 1. Block diagram of SIMP.

그림 2 는 SIMP의 동작 타이밍을 보여주는 것으로 16 클럭사이클마다 한 번씩 정합결과를 출력하며 파이프라인을 이용함으로써 앞뒤 연산과 중첩되어 동작할 수 있다. SIMP는 6단계의 파이프라인 구조이며 그중 SSD단계는 다시 3단계의 파이프라인 구조로 되어있다. IN과 OUT은 각각 입출력단계이고 SM은 슬라이딩 메모리 동작 단계이다. LF는 로컬서포트 필터동작 단계이고 SEL 단계에서는 선택기도 동작한다.

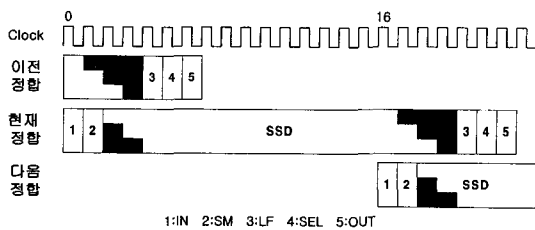


그림 2. SIMP의 파이프라인
Fig. 2. Pipeline of SIMP.

1. 슬라이딩 메모리(sliding memory)

한 번의 스테레오 정합을 위해 필요한 데이터양은 정합원도우의 크기의 제곱과 최대 변이에 비례하며 SIMP의 경우에는 기준영상에서 9개, 탐색영상에서 51개의 데이터가 필요하다. 그러나 영상취득장치에서 얻어지는 영상데이터는 순차적이어서 임의 위치의 데이

터를 얻기 어려우며 메모리와 같은 저장장치의 영상데이터를 이용할 때도 많은 수의 데이터를 동시에 얻는 것 역시 어려운 일이다. 그러므로 실시간 스테레오 정합을 구현하기 위해서는 순차적인 영상데이터를 입력 받으면서 필요한 데이터들을 한꺼번에 얻을 수 있는 저장장치가 필요하다.

이에 대한 해결책으로 SRA를 이용한 방법이 이용되고 있다. Hsieh-Lin^[12]은 탐색영역과 기준영역을 SRA에 순차적으로 입력하고 서로 다른 동기신호로 데이터를 이동시켜 계산에 필요한 데이터를 동시에 얻을 수 있는 회로구조를 설계하였다. 그러나 이 구조는 기준영역이 바뀔 때마다 SRA의 데이터를 모두 새로 바꿔야 하여 스테레오 정합을 하기에는 효율적이지 못하고 단독으로 스테레오 정합 전과정을 처리하기는 어렵다. 그러므로 본 논문에서는 기준영역과 탐색영역인 같은 동기신호에 데이터 이동하며 기준영역이 바뀔 때 다시 데이터를 입력할 필요가 없이 기준영역에 알맞은 탐색영역을 얻을 수 있도록 해주는 슬라이딩 메모리 회로를 제안하였다.

슬라이딩 메모리는 영상데이터를 순차적으로 입력받으면서 필요한 데이터들을 동시에 출력하며 SRA와 달리 기준영상이 바뀌어도 탐색영역의 데이터를 교체하지 않고 연속적으로 필요한 데이터를 사용할 수 있게 한다. 그림 3의 (a)는 슬라이딩 메모리의 기본 단위인 SM셀(SM cell)이다. SM셀은 두 단의 D래치로 구성되며 두 클럭, SMCLK1과 SMCLK2에 차례로 펄스를 발생시키는 슬라이딩 동기신호(sliding clock)로 동작한다. 슬라이딩 메모리는 SM셀들이 직렬로 연결된 구조를 가지고 있어 그림 4와 같다. 많은 SM셀을 직렬로 연결하면 입력 클럭에 클럭왜곡현상(clock skew)이 발생할 수 있으므로 non-overlapping two phase clock scheme을 적용하였다. SM셀들 중 일부는 슬라이딩 메모리의 출력을 담당하게 된다. 출력 윈도우의 크기가 N×M이고 입력영상의 크기를 R×R이면 슬라이딩 메모리는 다음 식으로 표현된다.

$$W_k(i, j) = SM(R \times (i - 1) + j), \quad 1 \leq i \leq N, \quad 1 \leq j \leq M \quad (2)$$

$W_k(i, j)$ 는 슬라이딩 메모리의 출력윈도우와 윈도우 내부의 위치를 나타내며 k 는 슬라이딩 클럭 수이다. $SM(x)$ 는 슬라이딩 메모리의 저장 데이터이고 괄호 안은 처음 셀을 기준으로 한 SM셀의 위치를 의미한

다. 그림 3의 (b)과 같이 입력영상에서 (u,v)위치의 N×M 영역의 데이터를 보려면 입력영상 데이터를 슬라이딩 메모리에 입력하여야 한다. 슬라이딩 메모리의 셀 개수가 S이고 N과 M이 홀수이면

$$\text{슬라이딩 클럭 수} = (u - \frac{N+1}{2}) \times R + (v - \frac{M-1}{2}) + S. \quad (3)$$

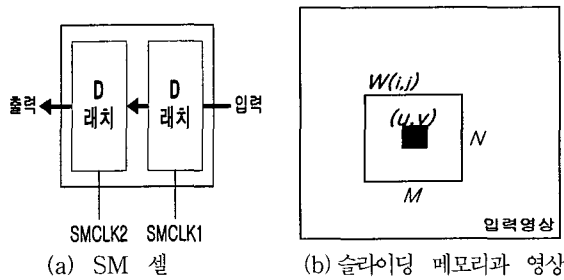


그림 3. 슬라이딩 메모리 셀의 구조와 데이터 이동
Fig. 3. Structure of sliding memory cell and data flow.

입력영상을 식 (3)에서 얻어진 슬라이딩 클럭 수만큼 입력하면 (u,v)위치의 윈도우를 슬라이딩 메모리의 출력 Wk(i,j)으로 얻을 수 있다. 출력윈도우의 위치가 영상의 가장자리에 있으면 슬라이딩 메모리의 출력윈도우의 일부가 입력영상의 벗어나게 된다. 그림 6은 3×3 출력윈도우인 경우의 예를 보여준다. 이런 경우에는 그림 6과 같이 출력과정에서 상하좌우의 영상데이터에서 올바른 영상데이터와 같은 값을 출력하도록 하였다.

그림 4과 그림 5에서는 8×8 영상을 예로 들어 슬라이딩 메모리를 설명하고 있다. 순차적으로 입력되는 영상데이터는 슬라이딩 메모리의 입력으로 들어가며 입력이 들어올 때마다 슬라이딩 명령을 보내어 데이터들이 슬라이딩 메모리에서 한 셀씩 앞으로 이동하도록 한다. 그림 4의 슬라이딩 메모리에서 굵은 테두리의 SM 셀은 출력을 슬라이딩 메모리의 외부로도 내보내는 출력 셀이며 셀안의 숫자는 ID를 나타낸다.

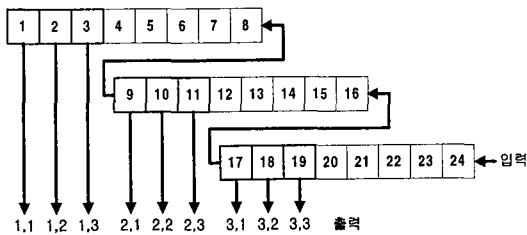


그림 4. 슬라이딩 메모리의 구조
Fig. 4. structure of sliding memory.

그림 5는 예를 들고 있는 8×8 영상을 슬라이딩 메모리에 순차적으로 입력한 상황을 표현한 것으로 화소의 데이터를 포함하고 있는 SM 셀의 ID를 화소의 위치에 표시하였다. 회색으로 칠해진 곳은 슬라이딩 메모리의 출력 셀들의 위치이며 3×3 윈도우를 형성하고 있다. 그림 5의 (a)에서 굵은 테두리 영역은 다음 정합에서 이용될 데이터들이다. 그림 5의 (b)는 (a)에 동기신호를 가한 후의 상태를 보여주는데 새로 필요한 정합데이터들을 얻을 수 있다. 슬라이딩 메모리의 출력을 관찰해보면 3×3 윈도우가 옆으로 슬라이딩한 것과 같은 것을 알 수 있다. 그러므로 한번 스테레오 정합을 계산한 후 슬라이딩 동기신호를 가하는 것만으로도 다음 정합 계산에 필요한 데이터를 얻을 수 있다.

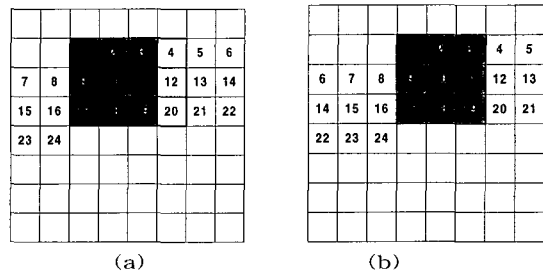


그림 5. 슬라이딩 메모리와 영상
Fig. 5. Sliding memory and image.

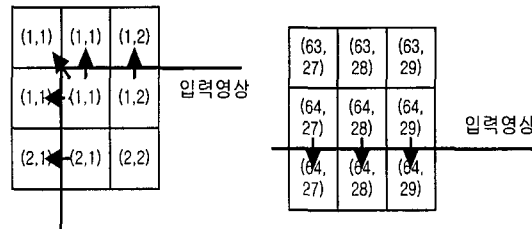


그림 6. 영상 가장자리에서 출력윈도우의 데이터
Fig. 6. Output window's data in edge of image.

슬라이딩 메모리의 크기는 필요한 데이터의 수와 입력되는 영상의 크기에 관련되어 있으며 필요한 데이터의 수는 최대 변이의 크기나 저장하는 영상이 기준영상인지 탐색영상인지에 따라 달라지게 된다. 스테레오 정합에 필요한 슬라이딩 메모리의 크기는 다음과 같이 계산할 수 있다. N×N의 영상을 정합하는 경우에 기준영역의 크기가 M×M이며 탐색영역의 크기는 M×K이라면 기준 영상 슬라이딩 메모리의 셀 수는 식 (4)와 같고 탐색영상 슬라이딩 메모리의 셀 수는 식 (5)과 같다.

$$\text{기준영상 } SM \text{ 셀의 개수} \geq (M-1) \times N + M \quad (4)$$

$$\text{탐색영상 } SM \text{ 셀의 개수} \geq (M-1) \times N + K \quad (5)$$

그러나 두 슬라이딩 메모리에 같은 영상좌표를 갖는 기준영상과 탐색영상의 데이터를 동시에 입력한다면 두 슬라이딩 메모리의 출력 윈도우 중심점은 영상좌표가 일치하여야 한다. 그러므로 기준영상 슬라이딩 메모리의 셀 수는 식 (6) 와 같게 된다.

$$\begin{aligned} \text{기준영상 } SM \text{ 셀의 개수} \\ \geq (M-1) \times N + \frac{K}{2} \quad \text{if } K \geq M \end{aligned} \quad (6)$$

64×64 영상을 스테레오 정합하는 SIMP의 경우에는 기준영상 슬라이딩 메모리의 셀 수는 최소 136개이고 탐색영상 슬라이딩 메모리의 셀 수는 최소 143개이나 구현이 용이하도록 기준영상은 145개를 탐색영상은 152개의 SM 셀을 사용하였다. 그리고 16 클럭마다 한 번씩 슬라이딩 동기신호를 발생시킨다.

2. SSD 연산장치(SSD calculator)

SIMP는 SSD 연산장치에서 3×3 윈도우의 SSD로 정합도를 계산하며 그림 7은 SSD 연산장치의 구조를 보여준다. SSD 연산장치는 AVD와 SQR과 SUM등의 3단계를 갖는 파이프라인을 가지고 있다. 9쌍의 데이터를 병렬 처리하여 3×3 윈도우의 정합도를 한 클럭사이클마다 한 번씩 계산할 수 있다.

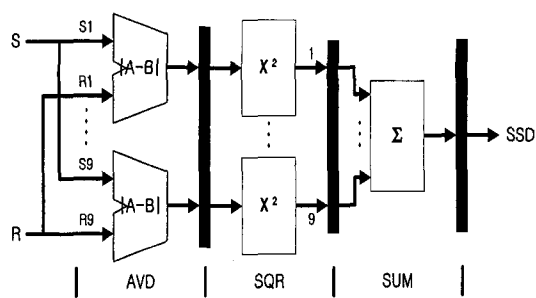


그림 7. SSD 연산장치의 구조
Fig. 7. Structure of SSD calculator.

AVD 단계에서는 비교할 후보정합과 기준영역의 9개 영상데이터를 각각 S1-S9와 R1-R9로 입력받아 AVD(absolute value of difference)를 계산한다. SQR 단계에서는 AVD의 결과를 제곱연산하여 9개의

SD(squared difference)를 얻으며 SUM 단계에서 9개의 SD를 모두 합하여 SSD를 얻는다.

AVD 단계는 두 개의 6비트 입력을 받아서 AVD를 4비트로 출력하는 연산을 처리한다. 6비트 입력 AVD연산의 출력은 6비트 데이터이지만 SIMP의 AVD 연산장치에서는 최대 AVD를 15로 제한하여 15이상의 결과는 모두 15를 출력하도록 하였다. 이는 큰 AVD를 갖는 입력 데이터는 잘못된 후보정합의 데이터일 가능성이 클 것이라는 가정에 근거한 것으로 실험을 통하여 결과에 나쁜 영향을 미치지 않음을 확인하였다. AVD 연산장치의 출력 비트 수를 줄이는 이유는 칩의 면적을 줄이기 위한 것으로 전체 칩 면적을 10%가량 줄일 수 있다.

SQR 단계는 4비트 입력의 제곱을 연산하는데 브라운 곱셈기(brown multiplier)를 이용한 제곱연산장치 9개를 가지고 있어 9개의 AVD값을 동시에 연산할 수 있다. 9개의 제곱된 AVD를 합하여 정합도를 얻어내는 SUM 단계에서는 CSA(carry save adder)를 2단으로 연결한 덧셈기 트리(adder tree)를 이용하였다. 앞에서 언급하였듯이 SIMP는 한 번의 정합을 하기 위해 15개 후보정합의 정합도를 계산하여야하고 SSD 연산장치는 한 클럭사이클에 하나의 정합도를 계산할 수 있으므로 15개 후보정합들의 데이터가 순차적으로 SSD 연산장치에 입력된다. SSD의 마지막 단계인 SUM 단계에서는 순차적으로 계산된 후보정합의 정합도들을 다시 모아서 다음의 LF 단계에 전달하는 역할을 수행한다.

3. 로컬서포트 필터(local support filter)

LSF(local support filter)는 이전 정합의 결과를 기준으로 하여 변이기울기 한계이내에 있는 후보정합들의 SSD값들을 조정하는 연산을 한다. 물리적으로 변이기울기 한계는 1로 하는 것이 타당하다^[2]. 본문에서는 바로 앞에서 정합된 결과만 참조하며 선택된 후보정합들의 SSD는 1/2을 곱하도록 하였다. 그러므로 이전 정합인 k-1번째 정합에서 s번째 후보정합이 우수정합으로 선택되었다면 LSF는 다음과 같은 연산을 하게 된다.

$$LF_k(i) = \begin{cases} LF_k(i)/2 & , s-1 \leq i \leq s+1 \\ LF_k(i) & \text{others} \end{cases}$$

그러나 k-1번째 정합이 실패하였다면 로컬 서포트 과정에서 아무런 연산도 일어나지 않는다.

정합점이 영상의 가장자리에 있어서 탐색영역의 일부가 영상을 벗어나 있거나 최대 변이가 7이하로 조정되어 있을 때는 하드웨어적으로 계산된 15개 후보정합들의 일부는 불필요하다. 이러한 후보정합들이 우수 정합 선택과정에서 배제되도록 SSD 데이터의 모든 비트를 '1'로 바꾸는 것을 마스킹이라 하며 로컬 서포트 과정 중에 포함된다.

4. 선택기(selector)

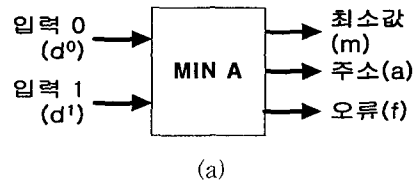
선택기는 15개의 후보정합 중에서 우수 정합을 선택하는데 SSD가 작을수록 높은 정합도를 가지므로 우수 정합은 가장 작은 값을 갖는 후보 정합이 된다. 선택기는 1 클럭사이클 동안에 15개의 후보정합 중 가장 작은 값을 가진 후보 정합을 알아낼 수 있도록 하였다.

본 논문에서는 15개의 후보 정합 중에서 우수 정합을 1 클럭사이클안에 알아내기 위해 최소 선택 트리(MST)를 제안하였다. MST는 두 가지 종류의 노드로 이루어진 트리 구조를 가지고 있으며 두 가지 노드는 그림 8 과 같다. 그림 8 의 (a)는 입력을 직접 받는 MIN A노드인데 두 입력을 받은 후 작은 값을 출력하며 입력0이 작은 경우에는 주소에 '0'을 내보내고 입력1이 작은 경우에는 '1'을 내보낸다. MIN A노드의 동작은 다음 식으로 표현된다.

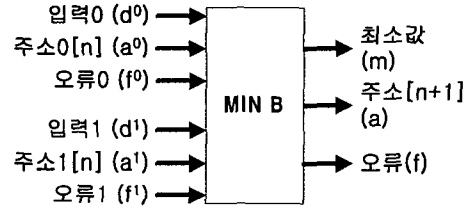
$$\begin{aligned} \text{최소값 } m &= \begin{cases} d^0 & \text{if } d^0 \leq d^1 \\ d^1 & \text{if } d^0 > d^1 \end{cases} \\ \text{주소 } a &= \begin{cases} 0 & \text{if } d^0 \leq d^1 \\ 1 & \text{if } d^0 > d^1 \end{cases} \\ \text{오류 } f &= \begin{cases} 1 & \text{if } d^0 = d^1 \\ 0 & \text{if } d^0 \neq d^1 \end{cases} \end{aligned}$$

그림 8 (b)의 MIN B노드는 MIN A의 출력이나 다른 MIN B의 출력을 입력으로 받을 수 있는 데이터와 주소에 관한 입력이 있다. MIN B는 주소출력은 주소입력보다 1비트를 확장하여 최상위 1비트에 작은 값을 갖는 입력의 번호를 출력하고 나머지 하위 비트 들에는 선택된 입력의 n 비트 주소를 출력한다. MIN B노드도 다음과 같은 식으로 표현된다.

$$\begin{aligned} \text{최소값 } m &= \begin{cases} d^0 & \text{if } d^0 \leq d^1 \\ d^1 & \text{if } d^0 > d^1 \end{cases} \\ \text{주소 } a_n a_{n-1} \dots a_1 a_0 &= \begin{cases} 0 a_{n-1}^0 a_{n-2}^0 \dots a_1^0 a_0^0 & \text{if } d^0 \leq d^1 \\ 1 a_{n-1}^1 a_{n-2}^1 \dots a_1^1 a_0^1 & \text{if } d^0 > d^1 \end{cases} \\ \text{오류 } f &= \begin{cases} 1 & \text{if } d^0 = d^1 \\ f^0 & \text{if } d^0 \leq d^1 \\ f^1 & \text{if } d^0 > d^1 \end{cases} \end{aligned}$$



(a)



(b)

그림 8. 두 종류의 MST 셀
Fig. 8. Two MST cells.

그림 9은 8개의 3비트 입력을 가지는 예를 들어 MST의 구조와 동작을 설명하고 있다. 모든 노드들의 왼쪽이 입력 0이고 오른쪽이 입력1이다. 각 MIN A의 두 입력들이 표시되어 있으며 3번 입력이 가장 작은 값을 갖는다. MIN A들은 두 입력을 비교하여 작은 값과 1비트의 주소를 출력한다. A 노드의 경우 "011"과 "110"이 입력되었으므로 "011"을 출력하면서 주소는 '0'을 출력한다. MIN A 노드들의 출력은 MIN B 노드인 E 노드와 F 노드로 전달된다. E의 입력은 "011"과 "010"이 되는데 작은 값은 "010"을 출력하고 입력1이 선택되었으므로 주소의 최상위비트에 '1'을 넣고 "010"의 address인 '1'을 나머지 비트에 넣어 "11"을 출력한다. 이러한 방법으로 마지막 G를 통과하면 주소에 최소입력의 위치인 "011"을 출력하며 최소입력인 "010"을 최소값으로 출력한다.

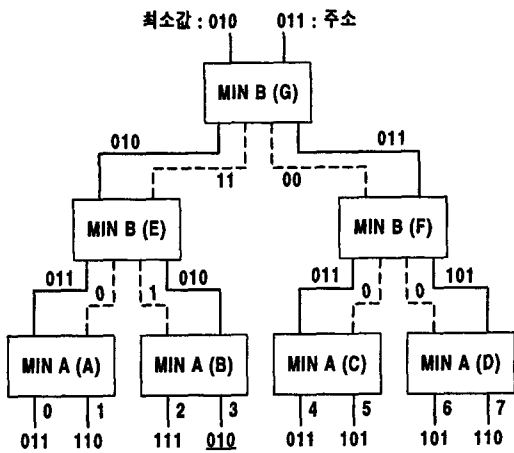


그림 9. 최소 선택 트리의 구조
Fig. 9. Structure of minimum selection tree.

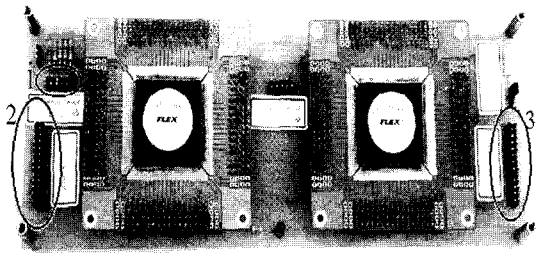


그림 10. SIMP 보드
Fig. 10. SIMP board.

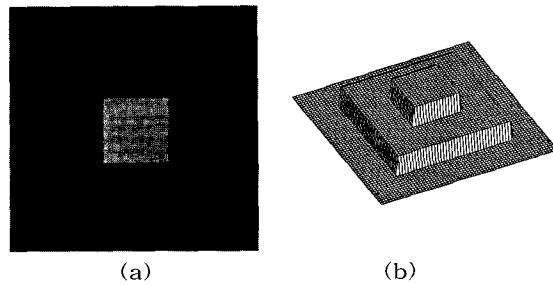
MST는 순차적인 최소값 탐색에 비해 빠르다는 이점이 있다. TC를 두 입력 데이터의 대소의 비교시간이라고 하면 N개의 데이터에서 최소값을 찾기 위해 순차적인 최소값 탐색을 하면 N 클럭사이클이 소요된다. 클럭사이클의 주기를 최소 TC까지 줄일 수 있다고 가정한다면 최소 $N \times TC$ 의 시간이 소요된다. 같은 조건에서 MST를 이용하며 계산시간이 최소 $\log_2(N) \times TC$ 로 줄어든다. 그러나 순차적인 최소값 탐색은 1개의 비교회로가 필요한데 반하여 MST는 N-1개의 비교회로가 필요하므로 면적이 늘어나게 된다. 최소 선택 트리는 비동기회로로 구성될 수 있을 뿐 아니라 파이프라인 구조를 적용할 수도 있다.

SIMP의 선택기는 MST를 이용하여 15개의 후보 정합 중에서 가장 작은 값을 찾아 우수 정합으로 선택하며 선택된 입력의 주소로 변이를 계산할 수 있다. 2개 이상의 값이 같은 경우에는 정합오류로 간주하고 15("1111")를 출력한다.

IV. 구현 및 실험결과

스테레오 영상 정합 프로세서(SIMP)는 VHDL을 이용하여 설계작업을 하였으며 FPGA 구현을 위해 Altera사의 FLEX10K 라이브러리를 이용하여 Synopsys에서 회로합성을 하였다. 합성된 회로는 Altera사의 Max+plusII를 이용하여 지연시간을 계산하였고 이를 백엔노테이션(back-annotation)하여 Synopsys에서 타이밍 시뮬레이션을 수행하였다. SIMP는 두 개의 FLEX10K70 FPGA칩으로 구현되었으며 타이밍 시뮬레이션에서 16MHz 입력 주파수에서 약 240 프레임/초의 처리속도로 동작하였다.

본 논문에서는 SIMP를 실험하기 위해서 균일불규칙 잡음(uniform random noise)를 이용하여 스테레오 영상을 만들었다. 이해의 편리를 위해서 주소출력에서 얻은 좌표를 이용하여 구성된 변이도를 흑백영상과 3차원영상으로 표현하였다. 흑백영상에서는 8단계의 변이크기와 정합오류를 나타내기 위해 9단계의 명암을 이용하여 변이가 클수록 밝게 하고 정합오류는 가장 밝은 흰색으로 표현하였다. 3차원 영상은 변이를 높이로 하여 구성하였으며 정합오류인 경우 가장 높게 표현하였다. 그림 11의 (c)와 (d)는 그림 11의 (a)와 같은 변이도를 가지는 스테레오 영상을 SIMP로 스테레오 정합한 결과들을 보여주고 있다. 그림 11의 (e)와 (f)는 SIMP의 변이도출력과 정확한 변이를 비교하여 얻은 오차인데 평면부분에는 오차가 없으나 불규칙한 변이도를 가지는 물체의 경계 부분에서 많은 오차가 발생하였다. 이는 SIMP에 적용된 스테레오 정합이 영역기반방법을 이용하기 때문에 물체의 경계영역에서 정확한 정합이 어려운 것과 변이기울기의 한계를 1로 하여 로컬서포트과정을 처리하였는데 경계부분의 변이 기울기가 이를 초과하였기 때문이다. 특히 오른쪽 경계부분은 왼쪽 경계부분에 비해 많은 오차가 발생하였는데 합성된 스테레오 영상의 시점의 위치로 인해 오른쪽 불규칙 영역에서 패색영역(occlusion region)이 발생하였기 때문이다.



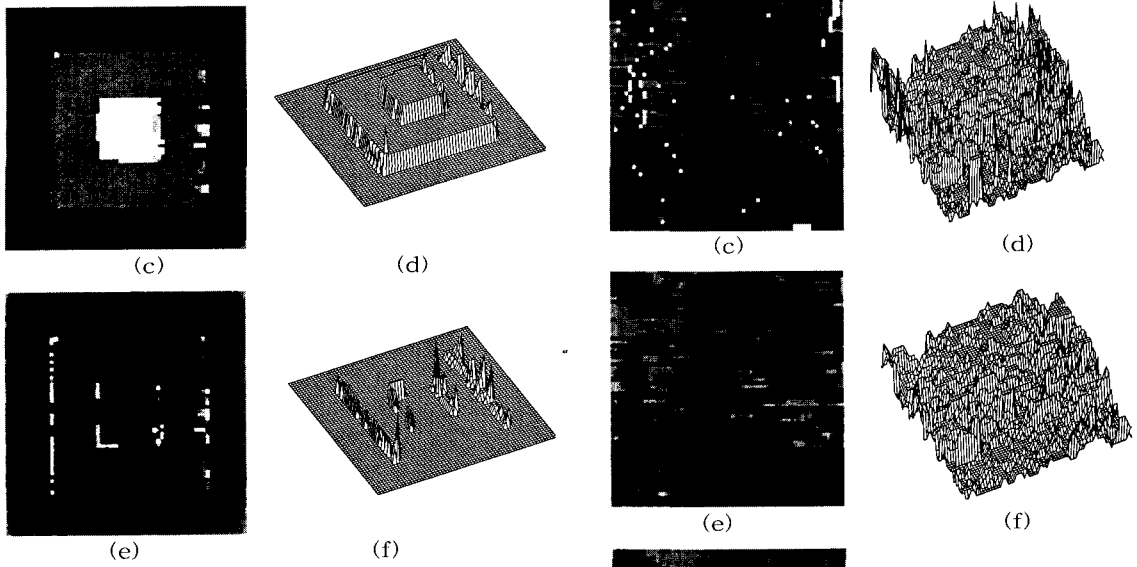


그림 11. 불규칙 잡음 스테레오 영상으로 실험한 결과 (a,b) 스테레오 영상의 변이도, (c,d) SIMP의 변이도 출력 결과, (e,f) 변이도의 오차

Fig. 11. Test result with random noise stereo image. (a,b) disparity map of stereo image, (c,d) output of SIMP, (e,f) error of disparity map.

그림 12 은 산간지역의 항공사진인 실제 스테레오 영상을 이용하여 정합한 결과를 보여주고 있으며 변이도를 내삽과정(interpolation)이나 미디안 필터(median filter)등의 방법으로 처리한 결과도 함께 보여준다. 그림 의 (a)와 (b)는 정합에 이용된 스테레오 영상으로 하위 부분은 도로가 있는 평지이며 상위 부분은 산간지역이다. (c)와 (d)를 보면 스테레오 영상에서 밝기와 변화가 거의 없는 영역에서 정합오류와 오차가 많이 발생한 것을 관찰할 수 있다. 이와 같은 경우에는 정합윈도우를 크게 하여 오차를 줄일 수 있다. 그림 12 의 (e)와 (f)는 오류(fault)가 발생한 점을 내삽과정으로 복구한 결과이며 (g)와 (h)는 이를 다시 미디안 필터를 이용하여 변이도를 부드럽게 만든 결과인데 (a)와 (b)의 예상되는 높낮이가 비교적 잘 나타난 것을 알 수 있다.

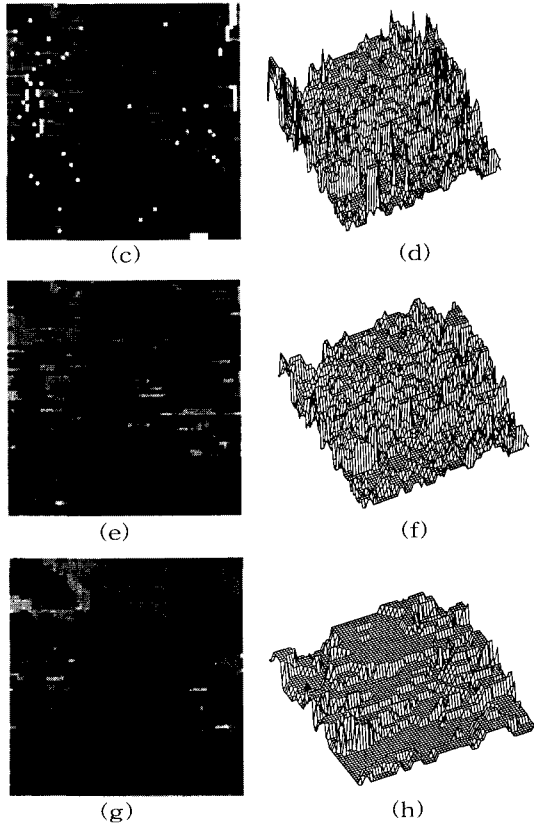
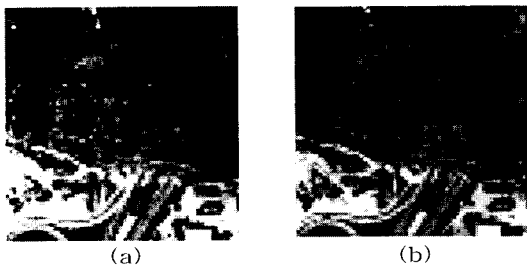


그림 12. 실제 스테레오 영상의 정합 결과 (a,b) 산간 지역의 스테레오 영상, (c,d) SIMP 스테레오 정합 결과, (e,f) 내삽과정을 이용하여 오류를 복구한 변이도, (g,h) 미디안 필터를 이용한 변이도

Fig. 12. Test result with real stereo image. (a,b) stereo image, (c,d) output of SIMP, (e,f) disparity map with interpolation, (g,h) disparity map with median filtering.

IV. 결 론

본 논문에서는 스테레오 정합을 실시간으로 할 수 있는 스테레오 영상 정합 프로세서(stereo image match processor:SIMP)를 제안하였다. SIMP는 64 그레이 레벨의 64×64 스테레오 영상을 입력받고 8 레벨 64×64 변이도를 출력하며 16MHz의 클럭입력에서 약 240 프레임/초(frames/sec)의 속도로 스테레오 정합을 처리할 수 있다. 이전의 스테레오 정합 처리용 VLSI들이 일부 계산을 빠르게 하는 것인데 반하여 SIMP는 스테레오 정합 전과정을 실시간으로 처리할 수 있다. 이를 위해 슬라이딩 메모리와 최소 선택 트리(MST)를 설계하였다.

SIMP는 정합윈도우나 탐색윈도우의 크기를 늘리고 입력영상의 크기를 늘림으로 더욱 성능을 향상시킬 수 있는데 이를 위해서는 풀커스텀(full-custom)이나 SOG 공정을 이용하여 구현하는 것이 필요하다. 또한 에피플라 제한조건 등 계산을 편리하게 하기 위한 스테레오 영상에 대한 조건들이 없이도 실시간 스테레오 정합을 할 수 있는 스테레오 정합 프로세서의 연구가 요구된다.

SIMP는 스테레오 비전 시스템 구성이 용이하도록 하여주며 자동항법시스템(auto guidance system)이나 3차원 물체인식이나 그 밖의 신속한 3차원 정보를 필요로 하는 어플리케이션에 사용될 수 있다. SIMP는 스테레오 정합뿐 아니라 물체의 속도측정이나 지문인식 등, 영상의 움직임이나 일치 여부를 판별하는 정합에도 응용될 수 있다.

참 고 문 헌

[1] D. Marr and T. Poggio, "Cooperative computation of stereo disparity," Science, vol. 194, pp. 283-287, Oct. 1976.
 [2] S. B. Pollard, J. E. W. Mayhew, and J. P. Frisby, "PMF: A stereo correspondence algorithm using a disparity gradient limit," Perception, vol. 14, pp. 449-470, 1985.
 [3] K. Prazdny, "Detection of binocular disparities," Biological Cybern., vol. 52, pp. 93-99, 1985.
 [4] M. Okutomi and T. Kanade, "A multiple-baseline stereo," IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 15, pp. 353-363, Apr. 1993.
 [5] T. Kanade and M. Okutomi, "A stereo matching algorithm with an adaptive window:

Theory and experiment," IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 16, pp. 920-932, Sept. 1994.
 [6] J. M. Hakkarainen and H.-S. Lee, "A 40x40 CCD/CMOS absolute-value-of-difference processor for use in a stereo vision system," IEEE J. Solid State Circuits, vol. 28, pp. 779-807, July 1993.
 [7] C. H. Hsieh and T. P. Lin, "VLSI architecture for block-matching motion estimation algorithm," IEEE Transaction on Circuits and Systems for Video Technology, Vol. 2, No. 2, June 1992.
 [8] V. L. Do and K. Y. Yun, "A low-power VLSI architecture for full-search block-matching motion estimation," IEEE Transaction on Circuits and Systems for Video Technology, Vol. 8, No. 4, August 1998.
 [9] T. Komarek and P. Pirsch, "Array architecture for block matching algorithms," IEEE Transaction on Circuits and Systems, Vol. 32, No. 10, October 1989.
 [10] P.J. Ashenden, ed., The Designer's Guide to VHDL, Morgan Kaufmann Publishers, Inc., 1996.
 [11] I. Koren, ed., Computer Arithmetic Algorithms, Englewood Cliffs, New Jersey: Prentice-Hall, Inc., 1993.
 [12] R. C. Gonzalez and R. E. Woods, eds., Digital Image Processing, Addison Wesley, 1993.
 [13] K. S. Fu, R. C. Gonzalez, and C. S. G. Lee, eds., Robotics: Control, Sensing, Vision and Intelligence, New York: McGraw-Hill, 1987.

저 자 소 개



沈 德 善(正會員)
 1961년 10월 18일생. 1984년 서울대학교 제어계측공학과 졸업(공학사), 1986년 동대학원 졸업(공학석사), 1993년 University of Michigan 항공우주공학과 졸업(공학박사), 1994년 1월~1995년 1월 University of

Michigan 전기 및 컴퓨터 공학과 Post-doc, 1995년 3월~현재 중앙대학교 전자전기공학부 부교수, 관심분야는 건실 제어, 전력 시스템 안정도 제어, 관성항법 시스템, GPS, VLSI 설계 등



金 演 宰(正會員)
 1974년 1월 16일생. 1996년 중앙대학교 제어계측공학과 졸업. 동대학원 석사(1999), 1999년~현재 동대학원 박사과정 재학 중. 관심분야는 H_∞ 제어 및 ASIC