

논문 2014-51-2-14

부동 소수점 연산을 이용한 실시간 영상 편위교정 FPGA 하드웨어 구조 설계

(A Real-Time Hardware Architecture for Image Rectification Using Floating Point Processing)

한 동 일*, 최 재 훈**, 신 호 철***

(Dongil Han[Ⓢ], Jaehoon Choi, and Ho Chul Shin)

요 약

본 논문은 두 대의 카메라로 찍은 영상을 이용하여 사물의 3D 정보를 계산하는 스테레오 매칭(Stereo Matching) 기법의 전처리 과정에 관한 연구이다. 본 논문에서는 카메라 내부의 왜곡 및 두 카메라간의 정렬 문제로 인해 생긴 영상의 수직시차(vertical parallax)를 제거하기 위한 실시간 편위교정(Rectification) 하드웨어 설계 구조를 제안한다. 이를 위한 사전 단계로 J.Y Bouguet이 설계한 Matlab 툴박스를 이용해 영상의 보정 매개변수(calibration parameter)를 구한 후 Heikkilä 와 Silven의 알고리즘을 기반으로 하여 편위교정 하드웨어를 설계하였다. 이때 결과 이미지의 정밀도를 높이기 위하여 Xilinx의 Core 생성기를 이용해 부동소수점 연산기를 생성하여 사용하였으며, 이를 통하여 룩업 테이블(Look-Up Table) 등을 사용하여 설계된 타 편위교정 하드웨어에 비해 높은 정밀도를 가지면서도 실시간으로 작동하는 편위교정 하드웨어를 설계할 수 있음을 확인하였다.

Abstract

This paper suggests a novel hardware architecture of a real-time rectification which is to remove vertical parallax of an image occurred in the pre-processing stage of stereo matching. As an off-line step, Matlab Toolbox which was designed by J.Y Bouguet, was used to calculate calibration parameter of the image. Then, based on the Heikkilä and Silven's algorithm, rectification hardware was designed. At this point, to enhance the precision of the rectified image, floating-point unit was generated by using Xilinx Core Generator. And, we confirmed that proposed hardware design had higher precision compared to other designs while having the ability to do rectification in real-time.

Keywords : Rectification, Calibration, FPGA, Real-time implementation, Epipolar line

* 정회원, ** 학생회원, 세종대학교 컴퓨터공학과
(Dept. of Computer Engineering, Sejong University)

*** 정회원, 한국전자통신연구원 지능형인지기술연구부
(Robot/Cognitive Convergence Research Department, ETRI)

Ⓢ Corresponding Author(E-mail: dihan@sejong.ac.kr)

※ 본 연구는 미래창조과학부의 ETRI 연구개발지원사업(No. 11921-03001, Beyond 스마트TV 기술개발)과 한국연구재단의 기초연구사업(No. 2012-007498)과 산업통상자원부 산업융합원천기술개발사업(No. K10041900)에서 지원되었습니다.

접수일자: 2013년9월12일, 수정완료일: 2014년1월28일

I. 서 론

최근 방송, 통신, 컴퓨팅 기술의 발전으로 인하여 일상생활에서 스마트폰, 스마트TV 등의 기기를 사용한 다양한 콘텐츠 및 서비스에 대한 소비가 증가하고 있다. 기존의 TV 이용자들은 수동적으로 미디어를 수용하는 형태의 소비를 하였지만 스마트 TV의 발전으로 인하여 소비자들의 시청 형태가 점차 능동적으로 변화해 나가는 추세이다. 이에 따라 스마트 TV의 새로운 사용자 인터페이스 기술로써, TV 제어 및 콘텐츠 검

색 등을 편리하고 용이하게 처리하기 위한 제스처 인식 기반의 인터페이스에 대한 요구가 증가하고 있다.

현재까지 알고리즘 형태로 개발된 동작 인식 장치는 적외선 투사 방식 등을 이용하기 때문에 저 해상도의 결과 이미지를 얻는 단점이 존재한다. 이로 인하여 원하는 기능을 작동시키기 위해서는 큰 동작이 요구되며 해상도의 한계로 인해 잘못된 인식 결과를 얻게 될 확률이 높아지게 된다. 이러한 단점을 극복하기 위해서는 고해상도의 3D거리 정보를 인식하는 것이 필요하다. 인간의 눈과 같은 방식으로 두 대의 카메라를 이용해 동일한 화소를 찾아 거리 정보를 인식하는 스테레오 매칭 (stereo matching) 기법은 이러한 기능을 수행하는데 있어서 최근 좋은 대안으로 제시되고 있다.

스테레오 매칭 기법은 1900년대 중반부터 연구되기 시작한 기법으로 초기에는 주로 군사적인 목적으로 특정 사물의 거리를 파악하기 위해 연구되었다. 이를 위하여 화소단위로 두 이미지의 밝기 값 차이를 이용한 스테레오 매칭 기법이 제시되었다. 이러한 초기 기법의 경우 주변의 잡음에 민감하고 계산된 화소간의 거리가 부정확하여 최종 결과 이미지의 정밀도가 떨어지는 단점이 있었다. 이후 컴퓨팅 기술의 발달로 DP (Dynamic Programming)^[1], BP(Belief propagation)^[2], AD (Absolute intensity differences) census^[3] 등의 알고리즘을 사용하여 점차 성능이 향상되고 있다. 이러한 스테레오 매칭 기법은 최근 보편화 되고 있는 스마트TV, 스마트폰, 무인운전자동차 등에서 사물의 동작을 인식하기에 유용한 기술로서 재조명되고 있다.

스테레오 매칭 기법으로 영상을 얻을 경우 같은 제품의 카메라로 찍은 영상이라 하더라도 카메라 내부의 왜곡으로 인해서 오차가 발생할 수 있다. 또한 두 대의 카메라가 설치될 경우 두 카메라 사이의 기하학적인 오차로 인해서 대응점을 찾기가 어려워진다. 그렇지만 두 영상에서 서로 대응되는 점이 같은 라인에 위치해 있을 경우 해당 라인에서만 대응점을 찾아낼 수 있어 연산이 간편해지고 오차도 줄어들게 된다. 스테레오 영상에서 서로 대응되는 점이 위치한 라인을 에피폴라 라인(Epipolar line) 이라고 하는데 이러한 에피폴라 라인을 수평으로 그리고 같은 라인으로 맞추기 위해서 편위교정으로 불리는 기하학적 영상 변환 기법^[4~5]을 사용해야 한다. 이 편위교정을 통해 두 영상의 에피폴라 라인이 동일한 라인 상에 놓이게 되며 즉, 영상 간의 정합점들이 동일한 스캔 라인 위에 위치하게 된다. 따라서 이러한 편위교정 과정을 거친 후에 스테레오 매칭을

진행하면 동일한 알고리즘을 사용하였을 때 깊이 추정 (Depth estimation)이 쉬워지는 장점이 생기게 된다. 또한 하드웨어 구현이 용이해지는 장점이 생기게 된다.

이 기법을 적용하기 위해서는 카메라 내부 파라미터 및 외부 파라미터를 미리 확보해야한다. 이러한 카메라 파라미터를 확보하기 위해서 그림 1의 예와 같이 체스보드와 같은 보정패턴(Calibration pattern)을 이용한 스테레오 보정법이 사용되고 있다. 내부 파라미터는 카메라 초점거리(Focal length), 주점(Principal point)의 위치, 영상의 찌그러짐 등을 의미하며, 외부 파라미터는 카메라 간의 상대적인 회전 각도 및 위치를 의미한다. 위와 같은 스테레오 보정법을 통해 획득된 카메라 파라미터는 에피폴라 리샘플링(Epipolar resampling)을 위한 변환행렬 산출에 사용되며, 이 변환행렬을 통해 스테레오 원 영상은 수직시차가 제거된 형태로 변환될 수 있다.

본 연구에서는 스테레오 매칭 기법을 이용한 스마트 TV용 실시간 제스처 인식 장치를 개발하기 위한 사전 단계로서 실시간 편위 교정 하드웨어 구조를 제안하고자 한다. 실시간 스테레오 매칭의 경우 연산 속도의 문제로 인해 소프트웨어를 통한 방식 보다는 하드웨어를 이용한 방식이 선호되고 있다. 소프트웨어 상에서도 GPU를 통한 병렬처리 등을 사용하여 실시간으로 편위 교정을 하는 방법^[6,7]이 연구되고 있으나 현재까지 높은 정밀도를 가지는 실시간 편위교정 소프트웨어는 개발되지 않은 상황이다. 하드웨어 상에서 스테레오 매칭을 위한 편위교정 방식은 여러 가지 방법이 개발되어 있으며 룩업 테이블(Look-Up Table)을 사용하는 방식^[8~13]이 널리 사용되고 있다. 룩업 테이블을 사용하는 방식은 구현이 간단하다는 장점이 있으나 편위교정을 외부

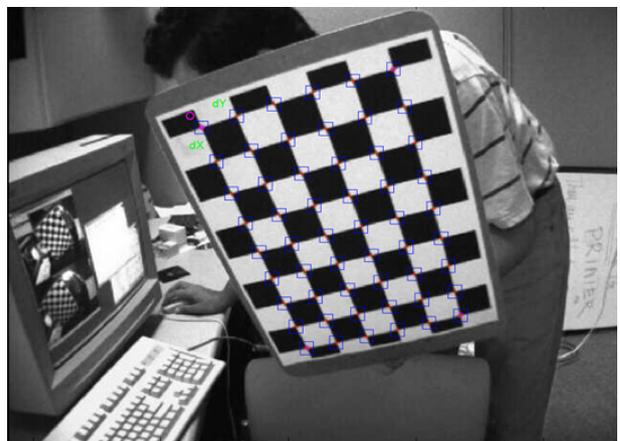


그림 1. 보정패턴을 이용한 스테레오 보정 과정
Fig. 1. Off-line stereo calibration step.

에서 계산하여 편위교정된 정수값의 좌표, 실수 형식의 가중치를 메모리에 저장하여 사용하기 때문에 별도의 메모리가 필요하게 되며 영상의 크기가 커질수록 더욱 큰 메모리가 요구되는 단점이 있다. 이러한 메모리 문제를 개선하고자 룩업 테이블을 압축하는 방법^[14]도 개발되었으나 정수부의 압축률은 높은 반면 실수부의 압축능력이 높지 않아 여전히 많은 메모리가 필요하다. 이러한 단점을 개선하고자 편위교정 과정을 하드웨어에서 실시간으로 수행하고자 하는 방식^[15~17]이 개발되었으나 하드웨어 구현의 한계로 인해 낮은 정밀도를 가지는 fixed-point를 사용해 연산을 하거나 연산량이 많은 왜곡(Distortion) 부분을 생략한 간략한 알고리즘^[18~19]을 사용하여 실제 사용 환경에서 많은 오차가 발생하게 된다. 또한 보정패턴 없이 입력되는 이미지를 통해 편위교정을 하는 알고리즘^[20~21]은 룩업 테이블을 위한 메모리를 사용하지 않고 별도의 캘리브레이션 작업이 필요 없는 장점이 있으나 하드웨어로 구현 시 실시간으로 계속 편위교정 결과가 변경되고 연산량이 많고 설계가 복잡하다는 단점이 있다.

본 논문에서는 Xilinx사의 Core 생성기^[22]를 사용해 생성한 Floating point IP를 활용하여 실시간으로 편위교정을 수행함으로써 높은 정밀도를 가지는 편위교정 하드웨어 구조 설계방법을 제안하고자 한다.

II. 본 론

1. Caltech의 Matlab Toolbox

현재 가장 널리 사용되고 있는 편위교정 표준 SW는 California Institute of Technology(Caltech)에서 개발되어 Matlab Toolbox로 제공되고 있는 캘리브레이션(Calibration), 편위교정(Rectification) 프로그램^[23]을 들 수 있으며 공개된 SW 중에서는 현재 최고의 성능을 제공하고 있다. 카메라를 이용하여 피사체의 3차원 거리 정보를 추출하는 경우 같은 제품의 카메라를 사용하더라도 카메라 내부의 기하학적 구조로 인해 카메라 별로 서로 다른 왜곡(distortion)이 발생하게 된다. 이러한 내부 왜곡과 두 대의 카메라가 위치한 3차원 상의 좌표 차이로 인해서 두 대의 카메라를 이용해서 피사체의 거리를 측정할 경우 스테레오 매칭 단계 이전에 카메라 캘리브레이션, 편위교정 단계를 필요로 한다.

캘리브레이션 과정은 오프라인으로 이루어지는데 설치된 두 대의 카메라의 초점거리(Focal length), 주점(Principal point), 비대칭 계수(Skew coefficient), 왜곡

(Distortion) 등을 파악하여 내부변수(Intrinsic parameter)를 생성하고, 두 카메라 간의 상대적인 회전(Rotation)과 이동거리(Translation)를 계산하여 외부변수(Extrinsic parameter)를 생성한다.

편위교정 과정은 Heikkilä와 Silven의 알고리즘^[24]을 기반으로 하여 실시간으로 이루어지는데 실제적인 스테레오 매칭 이전에 이루어지는 사전 단계로써 사용된다. 즉 캘리브레이션 과정에서 생성된 변수들을 이용해 편위교정을 수행하게 되며 월드 좌표(world coordinate)상의 같은 점들은 두 개의 카메라 영상에서 같은 스캔 라인에 에피폴라 라인이 생성되도록 영상의 수직수차를 실시간으로 제거한 후 스테레오 매칭부에 게 제공하게 된다.

가. 캘리브레이션(Calibration)

캘리브레이션 과정에서 생성되는 내부변수는 핀홀 카메라 모델^[25]을 통해 설명이 가능하다. 초점거리는 렌즈의 중심과 이미지 센서와의 거리를 의미하며 픽셀 단위를 사용한다, 표 1에서 초점거리를 하나의 값이 아닌 2개의 값으로 구분한 이유는 이미지 센서의 물리적인 셀 간격이 가로 방향과 세로 방향이 다를 수 있음을 모델링하기 위함이다. 주점은 카메라 렌즈의 광축이 이미지 센서와 만나는 점의 영상좌표로서 영상의 기하학적 중심을 의미하며 픽셀 단위를 사용한다. 비대칭계수는 이미지 센서의 셀 배열(cell array)이 기울어진 정도를 의미한다. 최근의 카메라는 이러한 비대칭 예러가 거의 없기 때문에 일반적인 카메라 모델에서는 비대칭계수를 거의 고려하지 않는다. 왜곡은 광선이 렌즈를 통해 굴절되면서 발생하는 반경수차(radial distortion)와 광선이 광축의 중심에서 이탈하면서 발생하는 접선수차(tangential distortion)로 나눌 수 있다.

캘리브레이션 과정에서 생성된 내부 변수는 표1 과 같으며 왼쪽과 오른쪽 카메라에서 각각 독립적으로 생성된다. 각각의 매개변수 Fc , Cc , $Alpha_c$, Kc 는 카메라

표 1. ^[26]내부 변수
Table 1. ^[26]Intrinsic parameter.

parameter	Description
$Fc=(Fc_1, Fc_2)$	2×1 vector, Focal length
$Cc=(Cc_1, Cc_2)$	2×1 vector, Principal point
$Alpha_c$	Scalar, Skew coefficient
$Kc=(Kc_1, \dots, Kc_5)$	5×1 vector, Distortion

라의 내부적인 요소인 초점거리, 주점, 비대칭계수와 반경수차와 접선수차를 포함한 왜곡을 의미한다.

표 1의 내부변수 중 초점거리, 주점, 비대칭 계수를 조합하여 식(1)의 변환행렬(KK)을 만들 수 있다.

$$KK = \begin{bmatrix} Fc_1 & \text{Alpah}_c * Fc_1 & Cc_1 \\ 0 & Fc_2 & Cc_2 \\ 0 & 0 & 1 \end{bmatrix} \quad (1)$$

카메라 외부 변수는 카메라 좌표계와 월드 좌표계 사이의 변환 관계를 설명하는 파라미터로, 카메라 고유의 파라미터가 아니기 때문에 카메라를 어떤 위치에 어떤 방향으로 설치했는지에 따라 달라진다. 외부 변수는 표 2 와 같으며 카메라 좌표계와 월드 좌표계 사이의 변환 관계를 설명한다. 각각의 매개변수 Omc , Tc 는 왼쪽과 오른쪽 카메라 각각에서 얻어진 변수를 이용해 계산되며 각각 두 좌표계의 회전과 평행이동변환이 표현된다.

Omc 는 식(2)와 같이 로드리게스 회전 공식(Rodrigues' rotation formula)을 통해 3×3 크기의 행렬로 표현 될 수 있다.

$$Rc = \text{rodrigues}(Omc) \quad (2)$$

X, Y, Z 가 이미지의 원본 좌표이고

X_c, Y_c, Z_c 가 회전과 평행이동 변환이 적용된 좌표라고 할 때 식(3)과 같이 표현이 가능하다.

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = Rc \cdot \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + Tc \quad (3)$$

$uu = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$ 이라고 했을 때 Tc 는 식(4)와 같이 표현이 가능하다.

$$\begin{aligned} T_0 &= Tc \times uu \\ T_1 &= \frac{T_0}{\|T_0\|} \\ T_2 &= \cos^{-1}\left(\frac{|Tc \cdot uu|}{\|Tc\| \cdot \|uu\|}\right) \cdot T_1 \\ T_3 &= \text{rodrigues}(T_2) \end{aligned} \quad (4)$$

표 2. ^[26]외부 변수

Table 2. ^[26]Extrinsic parameter.

parameter	Description
Omc	3×1 vector, Rotation parameter
Tc	3×1 vector, Translation parameter

식(3)과 식(4)를 통해 회전변환과 평행이동 변환을 모두 포함하는 3×3 크기의 회전 행렬 R 을 만들 수 있다.

$$R = T_3 \cdot Rc \quad (5)$$

소프트웨어에서는 편위교정을 할 이미지에 식(1)에서 계산한 변환 행렬(KK)을 적용한 후 식(5)와 같이 계산된 회전 행렬(R)을 계산한다.

하지만 본 연구에서는 하드웨어를 통해 편위교정을 진행하며, 연산량을 줄이기 위하여 변환 행렬(KK)과 회전 행렬(R)을 조합하여 내부변수와 외부변수가 모두 반영된 3×3 크기의 $Pose$ 행렬을 생성하여 하드웨어 연산에 사용한다.

$$Pose = R^T \cdot KK^{-1} \quad (6)$$

식(6)에서 사용된 T 는 전치행렬을 의미한다. 앞서 언급한 바와 같이 위의 표 1의 내부 변수와 식(1) ~ (6)를 통해서 계산되는 $Pose$ 행렬은 그림 1과 같은 보정 패턴을 통해서 오프라인 상태에서 계산되며 편위교정 단계에서 실시간 처리에 직접 사용된다.

나. 편위교정(Rectification)

편위교정 과정은 카메라에서 입력된 영상에서 왜곡을 제거하고 에피폴라 라인을 두 영상 메모리의 같은 스캔라인으로 일치시키는 과정이다. 이를 위하여 내부변수와 외부변수를 이용해 카메라 좌표계 상의 물체를 월드 좌표계로 변환한 후 왜곡을 제거하고 에피폴라 라인을 일치시킨다. 이후 월드 좌표계를 다시 카메라 좌표계로 역 변환한 후 2차원 영상 좌표계로 투영하면 원 영상 기준으로 실수 형태의 좌표가 나오게 되므로 해당 좌표에 인접한 4개의 화소 값을 이용해 양선형 보간을 적용하면 보정된 위치에서의 최종 화소값을 구할 수 있다. 편위교정 과정에 대한 자세한 사항은 아래 2절에서 하드웨어 설계와 같이 설명하고자 한다.

2. 제안된 하드웨어 설계

본 논문에서 제안하는 하드웨어 구조에서는 각각의 카메라로부터 영상과 동기신호를 입력 받고 파라미터 업로더(Parameter uploader)를 통해 오프라인 캘리브레이션 단계에서 생성된 표1의 Kc 파라미터(5×1 vector), Fc 파라미터(2×1 vector), Cc 파라미터(2×1 vector), Alpah_c 파라미터(Scalar)와 식(6)의 $Pose$ 행렬(3×3 matrix)을 입력 받아 편위교정을 수행한다.

실시간으로 입력되는 카메라 영상은 쓰기 동기 신

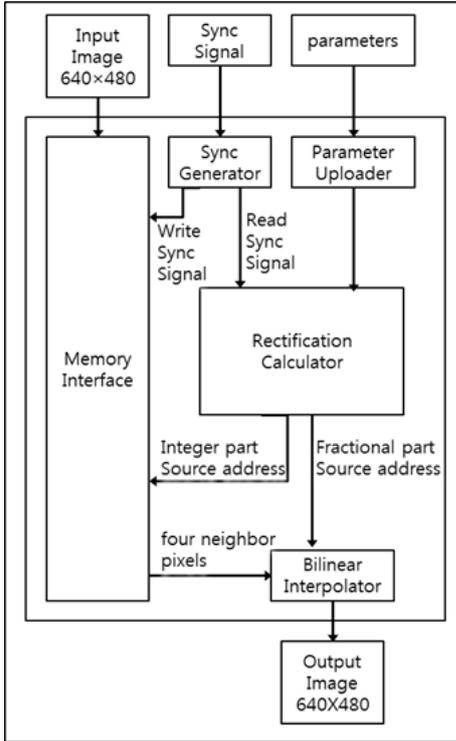


그림 2. 편위교정기 전체 블록도
Fig. 2. A block diagram of Rectifier.

호(Write sync signal)에 맞추어 프레임 메모리에 저장된다. 입력된 파라미터들은 편위교정 계산부(Rectification calculator)내에 위치하고 있는 목적 주소 생성기(Destination address generator)에서 생성된 목적 주소와 읽기 동기 신호(Read sync signal)를 따라 연산되어 실수 형태의 주소 값을 출력한다. 이 주소 값 중 정수 부분은 메모리에서 화소 값을 읽는데 사용되며 소수 부분은 양선형 보간기(Bilinear interpolator) 블록으로 전달되어 메모리에서 읽어드린 4개의 화소와 함께 양선형 보간에 사용된다. 각 블록에 대한 자세한 설명을 그림 2로 나타내었다.

가. 편위교정 계산부(Rectification Calculator)

편위 교정 계산부에서는 입력되어지는 파라미터를 계산하여 결과 이미지의 각 화소를 입력 영상의 어느 부분에서 가져 오는지를 매 화소 클럭마다 계산한다. 편위 교정 계산부의 구조를 그림 3에 나타내었다.

(1) 목적 주소 생성기

이 블록에서는 읽기 동기 신호에 맞추어 순차적으로 출력 영상의 좌표(px_n, py_n)를 출력한다.

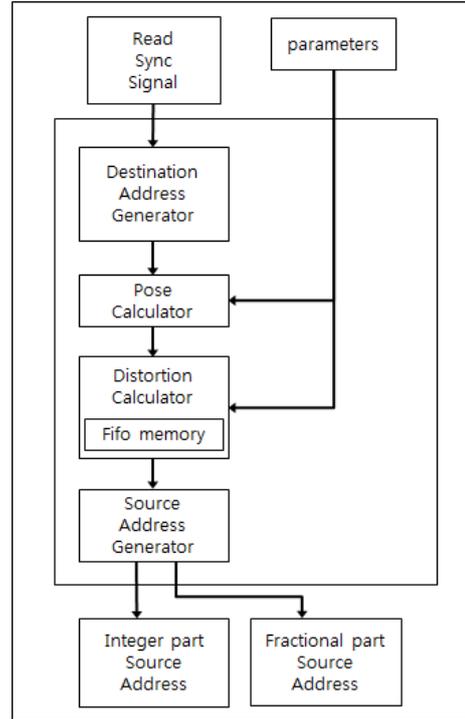


그림 3. 편위교정 계산부 블록도
Fig. 3. A block diagram of Rectification Calculator.

(2) Pose 계산부 (Pose Calculator)

이 블록에서는 식(2)의 **Pose** 행렬을 변수로 받아 회전 및 이동 변환을 수행한다.

$$\mathbf{ray} = \begin{bmatrix} ray_1 \\ ray_2 \\ ray_3 \end{bmatrix} = \mathbf{Pose} \cdot \begin{bmatrix} px_n \\ py_n \\ 1 \end{bmatrix} \quad (7)$$

Pose 행렬은 3x3 행렬이므로 아래와 같이 풀어쓰는 것이 가능하다

$$\begin{aligned} ray_1 &= px_n \times Pose(1,1) + py_n \times Pose(1,2) + Pose(1,3) \\ ray_2 &= px_n \times Pose(2,1) + py_n \times Pose(2,2) + Pose(2,3) \\ ray_3 &= px_n \times Pose(3,1) + py_n \times Pose(3,2) + Pose(3,3) \end{aligned} \quad (8)$$

$$\begin{bmatrix} x_n \\ y_n \end{bmatrix} = \begin{bmatrix} ray_1/ray_3 \\ ray_2/ray_3 \end{bmatrix} \quad (9)$$

위의 식(7)~식(9)을 하드웨어 블록도로 표현하면 그림 4와 같다.

(3) 왜곡 계산부(Distortion calculator)

이 블록에서는 Pose 계산부에서 회전 및 이동 변환을 적용하여 계산된 월드 좌표계에 표 1의 **Kc** 벡터 변수를 이용하여 카메라 내부의 왜곡을 수정한다.

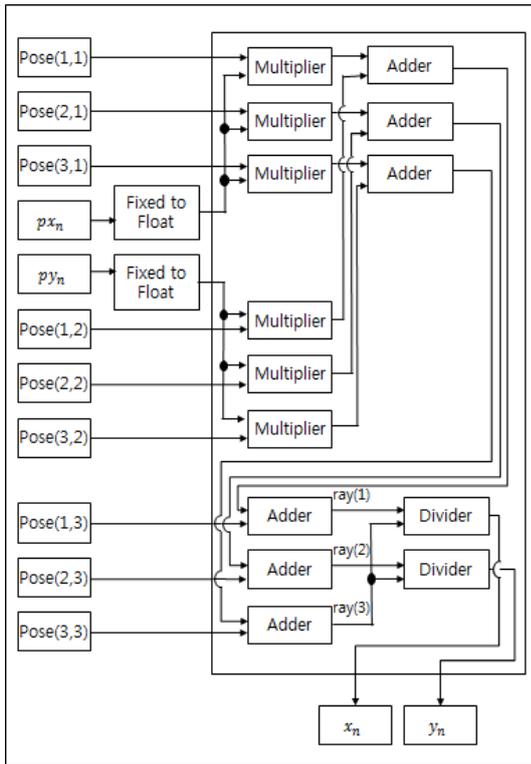


그림 4. Pose 계산부 블록도
Fig. 4. A block diagram of Pose calculator.

$$r^2 = x_n^2 + y_n^2 \quad (10)$$

$$\begin{bmatrix} xr_n \\ yr_n \end{bmatrix} = \begin{bmatrix} x_n \\ y_n \end{bmatrix} \times (1 + Kc_1 r^2 + Kc_2 r^4 + Kc_3 r^6) \quad (11)$$

$$\begin{bmatrix} xt_n \\ yt_n \end{bmatrix} = \begin{bmatrix} Kc_3 \times 2x_n y_n + Kc_4 \times (r^2 + 2x_n^2) \\ Kc_3 \times (r^2 + 2y_n^2) + Kc_4 \times 2x_n y_n \end{bmatrix} \quad (12)$$

$$\begin{bmatrix} xd_n \\ yd_n \end{bmatrix} = \begin{bmatrix} xr_n + xt_n \\ yr_n + yt_n \end{bmatrix} \quad (13)$$

위의 식(10)~식(13)을 하드웨어 블록도로 표현하면 그림 5와 같다.

(4) 소스 주소 생성기(Source address generator)

이 블록에서는 월드 좌표계에서 왜곡이 수정된 좌표를 표 1의 **Fc**, **Cc**, **Alpah_c** 벡터를 이용해 만든 **KK** 행렬과 내적하여 다시 카메라 좌표계로 사상해주는 역할을 한다.

$$\begin{bmatrix} xi_n \\ yi_n \\ xf_n \\ yf_n \end{bmatrix} = \begin{bmatrix} xs_n \\ ys_n \\ xs_n - \lfloor xs_n \rfloor \\ ys_n - \lfloor ys_n \rfloor \end{bmatrix} \quad (14)$$

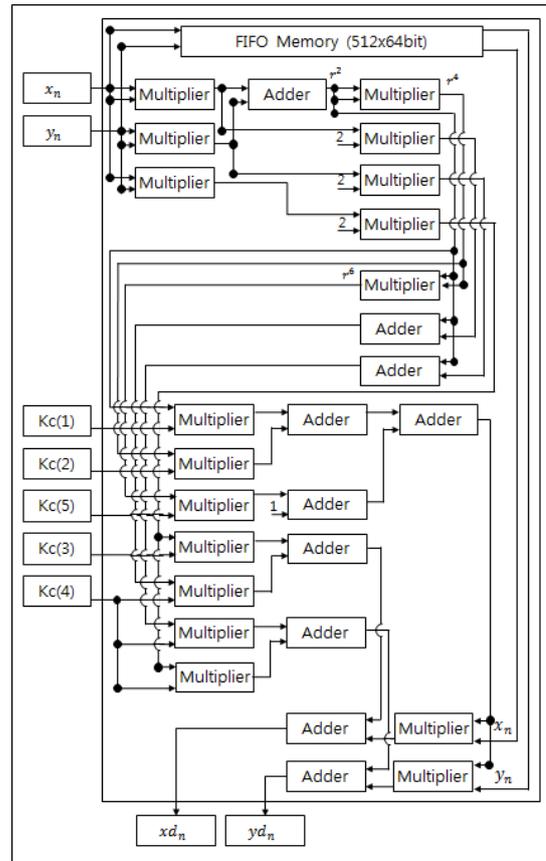


그림 5. 왜곡 계산부 블록도
Fig. 5. A block diagram of Distortion calculator.

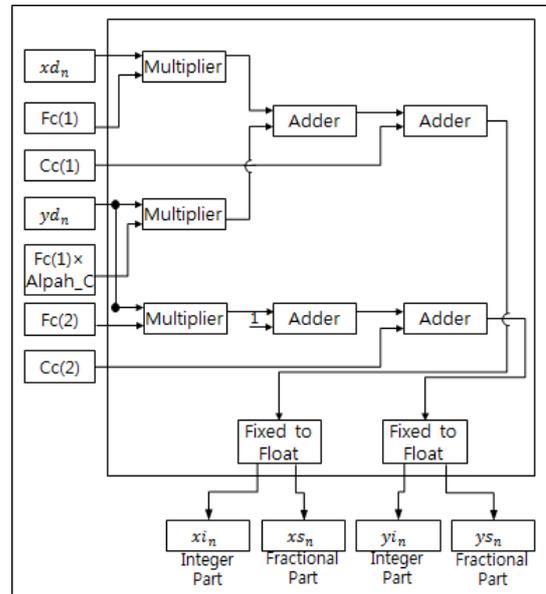


그림 6. 소스 주소 생성기 블록도
Fig. 6. A block diagram of source address generator.

$$\begin{bmatrix} xs_n \\ ys_n \end{bmatrix} = \mathbf{KK} \cdot \begin{bmatrix} xd_n \\ yd_n \end{bmatrix} \quad (15)$$

식(14)를 통해 구해진 xs_n, ys_n 은 카메라 좌표계에서의 주소를 나타내며 실수 형태의 값을 가진다. 하드웨어에서는 입력 카메라 영상의 메모리의 주소가 된다. 이를 정수부를 분리하여 정수부 좌표(xi_n, yi_n)는 메모리 인터페이스(Memory Interface)의 메모리 모드(Memory Mode)블록으로 보내어 인접한 4개 화소를 읽어오는데 사용하고 소수부 좌표(xf_n, yf_n)는 양선형 보간기로 전해져 양선형 보간을 수행하는데 사용된다. 소스 주소 생성기의 구조를 그림 6에 나타내었다.

나. 메모리 인터페이스

메모리 인터페이스는 입력되는 영상을 쓰기 동기 신호에 맞추어 메모리에 저장하고 읽기 동기 신호에 따라 편위교정 계산부에서 전달해 준 정수부 좌표 주위의 4개 화소를 읽어 양선형 보간기로 전달하는 역할을 수행한다. 메모리 인터페이스의 구조를 그림 7에 나타내었다.

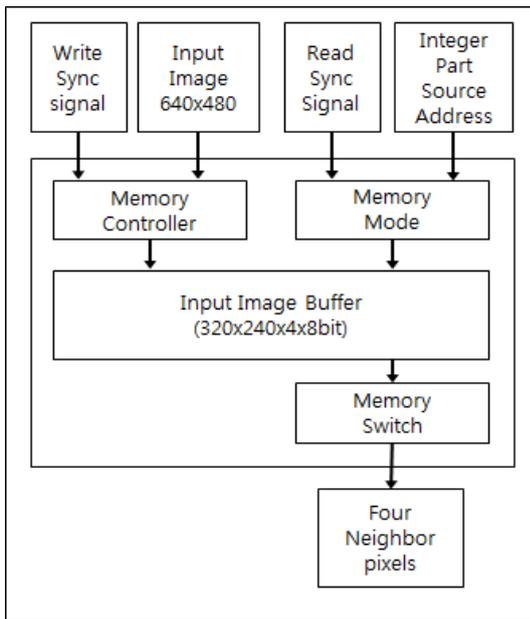


그림 7. 메모리 인터페이스 블록도
Fig. 7. A block diagram of Memory Interface.

(1) 메모리 제어기(Memory Controller)

이 블록에서는 입력받은 640×480 크기의 영상을 320×240×8bit 크기의 메모리 4개에 순차적으로 저장한다. 쓰기 동작 시에는 4개의 메모리 중 하나의 메모리를 선택하여 쓰게 되며 읽기 동작 시에는 읽어야 하는 화소를 둘러싸는 4개의 화소를 동시에 읽게 된다. 그림 8에서 같은 메모리는 동일한 색으로 표현하였다.

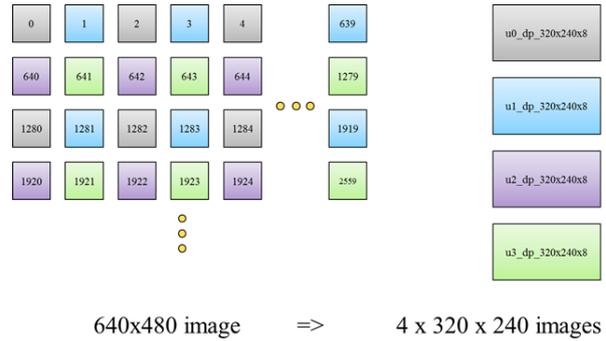


그림 8. 메모리 구조
Fig. 8. Memory Structure.

(2) 메모리 모드 (Memory Mode)

이 블록에서는 소스 주소 생성기 블록에서 식(15)를 통해 계산된 정수부 좌표(xi_n, yi_n)의 주소값을 이용해 해당 주소에 인접한 4개의 화소 값을 읽어온다. 이때 4개의 메모리에서 동시에 제공되는 화소와 함께 보간이 용이하도록 화소의 정렬 정보를 생성해 메모리 스위치(Memory Switch) 블록에 사용한다.

(3) 메모리 스위치(Memory Switch)

입력되는 4개의 화소 값을 제어 신호를 통해 각각 북서(NW), 북동(NE), 남서(SW), 남동(SE)로 정렬하여 출력해주는 기능을 수행한다. 제어 신호에 따른 메모리의 출력 제어 구조를 그림 9에 표현하였다.

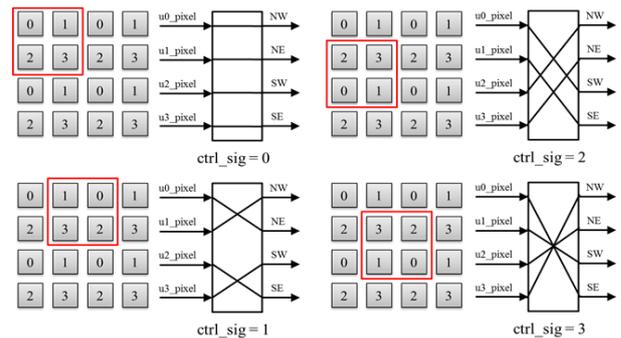


그림 9. 메모리 스위치의 역할
Fig. 9. Function of Memory Switch.

다. 양선형 보간기

이 블록에서는 편위교정 계산부를 통해 구해진 정수부 좌표(xi_n, yi_n) 주위 4개 화소(NW_n, NE_n, SW_n, SE_n)와 소수부 좌표(xf_n, yf_n)를 이용해 양선형 보간을 구현하며 여기서 계산되어지는 BI_n 이 본 논문에서 구현하는 하드웨어의 최종 출력 값이 된다.

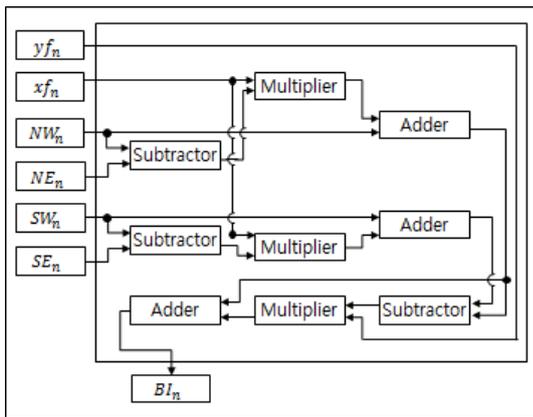


그림 10. 양선형 보간기 블록도
Fig. 10. A block diagram of Bilinear Interpolator.

$$BI_n = [1 - xf_n \quad xf_n] \begin{bmatrix} NW_n & NE_n \\ SW_n & SE_n \end{bmatrix} \begin{bmatrix} 1 - yf_n \\ yf_n \end{bmatrix} \quad (16)$$

위의 식(16)를 하드웨어로 구현한 블록도는 그림 10과 같다.

III. 실험

실험에 사용되는 영상은 640×480의 계조 (Gray Scale) 영상이며 실험 결과는 Matlab Toolbox의 편위 교정 기능을 수행 시 출력되는 결과와 VHDL 시뮬레이션에서 출력되는 결과의 차이 값을 비교하였다. VHDL 시뮬레이션 과정에서 얻을 수 있는 주요 인터페이스 신호 값을 그림 11에 나타내었다.

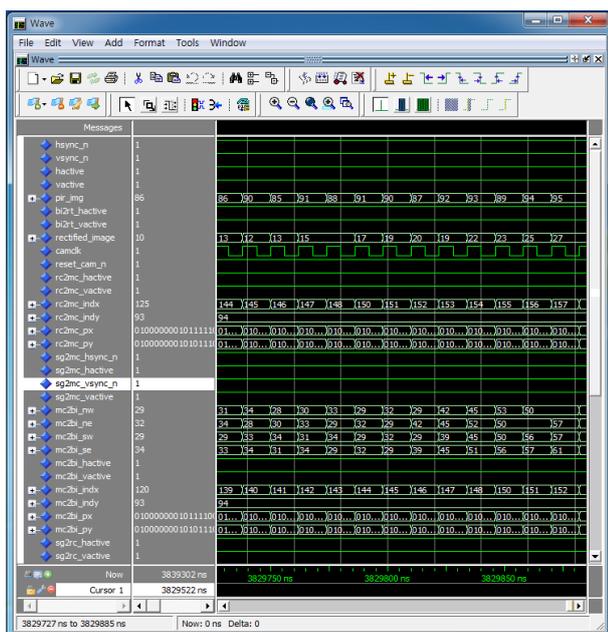


그림 11. ModelSim 시뮬레이션 결과 예
Fig. 11. ModelSim Simulation Example.

호 값을 그림 11에 나타내었다.

연산의 정밀도는 아래와 같이 3가지로 구분하여 설계와 실험을 실시하였다.

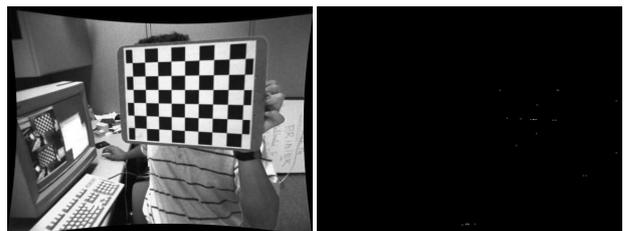
1. 단정밀도 부동소수점 연산 사용
2. 배정밀도 부동소수점 연산 사용
3. 배정밀도 부동소수점 연산 사용, 양선형 보간부에만 단정밀도 부동소수점 연산 사용.

양선형 보간부에 단정밀도 부동소수점 연산을 사용



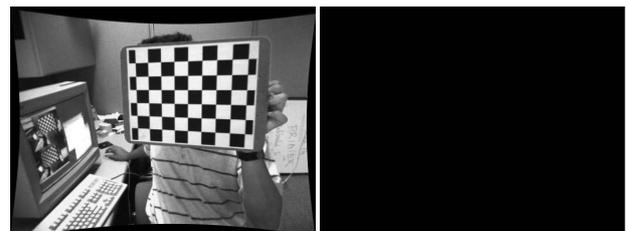
(a) 원본영상

(b) Matlab 수행결과



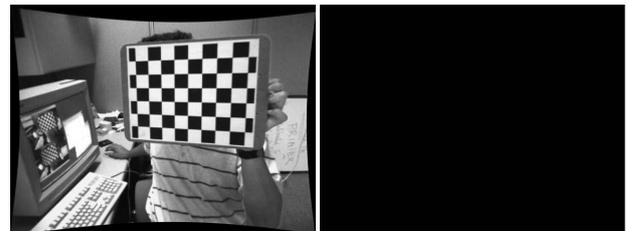
(c) 단정밀도 부동소수점연산
사용시 VHDL 결과

(d) (b)와(c)의 밝기값 차이
차이값 > 1 (55개 화소)



(g) 배정밀도 부동소수점
점 연산 사용시 VHDL 결과

(h) (b)와(g)의 밝기값 차이
차이값 > 0 (2개 화소)



(e) 배정밀도 부동소수점 연
산 사용시 VHDL

(f) (b)와(e)의 밝기값 차이
결과 정확히 일치

그림 12. Matlab 결과, VHDL시뮬레이션 결과 및 차이값 비교

Fig. 12. Matlab result, VHDL simulation result and pixel difference values.

하여 실험을 진행한 이유는 편위교정에 사용되는 연산에 비해 보간은 상대적으로 정밀도에 덜 민감한 특성을 지니기 때문이다.

PSNR, SSIM을 활용해 두 영상의 차이를 계산한 결과는 표 3과 같다.

그림 12와 표 3에서 볼 수 있듯이 본 논문에서 구현한 하드웨어 구조는 소프트웨어와 동일한 결과를 출력할 수 있으며, 정밀도를 낮춰 하드웨어 자원을 적게 소모하면서도 매우 높은 정밀도를 가짐을 알 수 있다. 배정밀도 부동소수점 연산을 사용 시 Matlab 수행 결과와 동일한 결과를 도출함을 알 수 있고 보간부 단정밀도 연산을 사용할 경우 103.03dB의 PSNR을 가져서 눈으로 보기에 차이를 발견할 수 없음을 알 수 있다.

개발된 VHDL코드를 Xilinx사의 Vertex-5 xc5vlx330-1ff1760모델에 매핑(Device mapping)을 시도하였으며 최고 동작속도 100.291MHz로 동작함을 확인

표 3. Matlab 결과와 VHDL 시뮬레이션 결과 비교
Table 3. Result comparison between Matlab and VHDL.

비교 알고리즘	단정밀도 부동소수점 연산	배정밀도 부동소수점 연산	배정밀도 부동소수점 연산 (보간부 단정밀도 부동소수점 연산)
PSNR	70.62 dB	∞	103.03 dB
SSIM	0.99999625	1	0.99999999

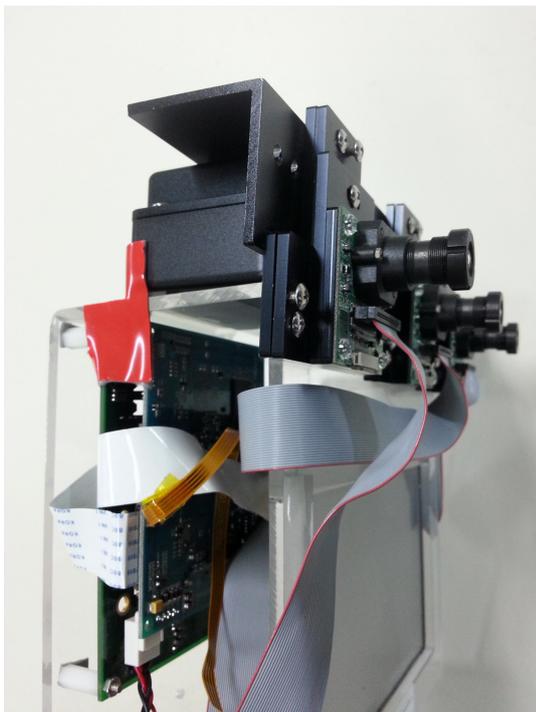


그림 13. 스테레오 매칭 FPGA 장치
Fig. 13. Stereo Matching FPGA device.

하였다. 그림 13에 개발된 스테레오 매칭 FPGA 장치의 사진을 나타내었다.

표 4는 여러 정밀도를 가지는 하드웨어 구조를 FPGA에 매핑 시켰을 때 사용되는 하드웨어 자원을 나타내었다. 단정밀도로 구현 시 FPGA의 17% 정도의 자

표 4. FPGA(Xilinx Virtex5 LX330) 논리합성 결과
Table 4. FPGA synthesis result summary.

Logic Utilization	Avilable	단정밀도 부동소수점 연산	배정밀도 부동소수점 연산	배정밀도 부동소수점 연산 (보간부 단정밀도 부동소수점 연산)
Slice Registers	207,360	34,872 (16%)	123,444 (59%)	107,443 (51%)
Slice LUTs	207,360	23,111 (11%)	94,060 (45%)	80,738 (38%)
Occupied Slices	51,840	10,390 (20%)	35,226 (67%)	30,762 (59%)
Bonded IOBs	1,200	24 (2%)	24 (2%)	24 (2%)
Memory Used	10,368	2,772 (26%)	2,808 (26%)	2,808 (26%)
DSP48E	192	50 (26%)	75 (39%)	73 (38%)



(a) 편위교정 미수행 영상으로 스테레오 매칭을 수행한 후 깊이 추정 결과



(b) 편위교정 수행 영상으로 스테레오 매칭을 수행한 후 깊이 추정 결과

그림 14. Simple Tree DP^[27] 수행 결과 영상
Fig. 14. Simple Tree DP^[27] Result Image.

원만을 사용하며 배경밀도로 구현 시 36% 정도의 자원을 사용함을 알 수 있다.

그림 12(a)의 영상을 이용하여 편위교정을 수행하기 전 원 영상들과 편위교정을 수행한 후의 영상들을, 그림 13의 스테레오 매칭 FPGA 장치를 통해 얻은 결과를 그림 14에 나타내었다. 편위교정 후의 영상을 이용한 깊이 추정 결과가 월등히 정확함을 확인할 수 있다. 스테레오 매칭 알고리즘은 DP 기법을 FPGA 동작 가능하도록 M. Jin, T. Maruyama가 제안한 Simple Tree DP 알고리즘^[27]을 사용하였다.

IV. 결 론

본 연구를 통해 스테레오 매칭을 위한 편위교정 기능을 하드웨어로 성공적으로 구현하였으며 실험 결과에서 알 수 있듯이 매우 높은 정밀도로 FPGA에서 실시간으로 동작함을 확인하였다. 특히 배경밀도 부동소수점 연산을 사용한 하드웨어 구조에서는 Matlab에서 제공하는 편위교정 Toolbox와 정확히 일치하는 결과를 구현하였으며 단정밀도 부동소수점 연산을 사용한 하드웨어 구조에서도 PSNR 70dB이상의 높은 정밀도의 결과를 확인할 수 있었다. 이를 통하여 스마트 TV의 제스처 인식을 위한 스테레오 매칭 장치 개발 시 실시간으로 정밀한 편위교정을 수행함으로써 보다 정확한 거리 정보를 추출하는 것이 가능함을 확인하였다. 또한 정확한 거리 정보를 제공함으로써 후 처리단계의 제스처 인식을 높여줄 것으로 기대된다.

REFERENCES

[1] M. Bleyer, M. Gelautz, "Simple but effective tree structure for dynamic programming based stereo matching", *VISAPP*, pp. 415 - 422. Madeira, Portugal, Jan 2008.

[2] Q. Yang, L. Wang, R. Yang, S. Wang, M. Liao, and D. Nister. "Real-time Global stereo matching using hierarchical belief propagation", *BMVC*, vol. 6, pp. 989-998, Sept 2006.

[3] X. Mei, X. Sun, M. Zhou, S. Jiao, H. Wang, and X. Zhang. "On building an accurate stereo matching system on graphics hardware", *ICCV Workshops*, pp.467-474, Barcelona, Spain, Nov 2011.

[4] D. Papadimitriou and T. Dennis, "Epipolar line estimation and rectification for stereo image

pairs", *IEEE Transactions on Image Processing*, Vol. 5, no. 4, pp. 672-676, Apr 1996.

[5] Zhang, Zhengyou. "A flexible new technique for camera calibration." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 22, no. 11, pp. 1330-1334. Nov 2000.

[6] Lee, Tung-Ying, et al. "Real-time correction of wide-angle lens distortion for images with GPU computing." *APCCAS*, pp. 456-459, Kaohsiung, Taiwan Dec 2012.

[7] Pantilie, Cosmin D., et al. "Real-Time Image Rectification and Stereo Reconstruction System on the GPU." *10th International Symposium on Parallel and Distributed Computing (ISPDC)*, 2011 pp. 79-85, Cluj Napoca, Romania, July 2011.

[8] Vancea, Cristian, and Sergiu Nedevschi. "LUT-based image rectification module implemented in FPGA." *IEEE International Conference on Intelligent Computer Communication and Processing*, pp. 147-154, Cluj-Napoca, Romania, Sept 2007.

[9] Jia, Yunde, et al. "A miniature stereo vision machine (MSVM-III) for dense disparity mapping." *Proceedings of the 17th International Conference on Pattern Recognition, ICPR*. Vol. 1, pp. 728-731, Cambridge, UK, Aug 2004.

[10] Yang, Ruigang, and Marc Pollefeys. "A versatile stereo implementation on commodity graphics hardware." *Real-Time Imaging*, Vol. 11 no. 1, pp. 7-18. Feb 2005.

[11] Maldeniya, Buddhika, et al. "Computationally efficient implementation of video rectification in an FPGA for stereo vision applications." *5th International Conference on Information and Automation for Sustainability (ICIAFs)*, 2010 pp. 129-224, Columbo, Srilanka, Dec 2010.

[12] Villalpando, Carlos Y., et al. "FPGA implementation of stereo disparity with high throughput for mobility applications." *Aerospace Conference, 2011 IEEE*, pp. 1-10, Big sky, USA, Mar 2011.

[13] Jawed, Khurram, et al. "Real time rectification for stereo correspondence." *International Conference on Computational Science and Engineering, CSE'09*. Vol. 2, pp. 277-284, Vancouver, Canada, Aug 2009.

[14] Park, Deuk Hyun, et al. "Real time rectification using differentially encoded lookup table." *ICUIMC '11 Proceedings of the 5th International Conference on Ubiquitous Information Management and Communication*, pp.47, Seoul, Korea, Feb 2011.

- [15] Pohl, Matthias, et al. "An efficient and scalable architecture for real-time distortion removal and rectification of live camera images." *International Conference on Reconfigurable Computing and FPGAs (ReConFig)*, pp. 1-7, Cancun, Mexico, Dec 2012.
- [16] Rodrigues, Joao GP, and Joao Canas Ferreira. "FPGA-based rectification of stereo images." *Conference on Design and Architectures for Signal and Image Processing (DASIP), 2010* pp. 199-206, Edinburgh, UK, Oct 2010.
- [17] Zicari, Paolo. "Efficient and high performance FPGA-based rectification architecture for stereo vision." *Microprocessors and Microsystems*, Vol. 37, No. 8, Nov 2013.
- [18] Islam, Jamin, et al. "Lowering power consumption using run-time reconfiguration for stereo rectification." *Canadian Conference on Electrical and Computer Engineering, 2008. CCECE 2008.* pp. 001693 - 001698, Niagara Falls, Canada, May 2008.
- [19] Hyo Su Jeong, "Real-time stereo image matching system", Thesis for master's degree of Kyungpook National University, 2011.
- [20] Huiyan, H. A. N., H. A. N. Xie, and Yang Fengbao. "Rectification of Uncalibrated Images for Stereo Vision." *TELKOMNIKA Indonesian Journal of Electrical Engineering* Vol.11, No. 1, pp. 322-329, Jan 2013.
- [21] Fusiello, Andrea, and Luca Irsara. "Quasi-Euclidean epipolar rectification of uncalibrated images." *Machine Vision and Applications*, Vol. 22, No. 4, pp. 663-670, July 2011.
- [22] online : <http://www.xilinx.com/tools/coregen.htm>
- [23] online : http://www.vision.caltech.edu/bouguetj/calib_doc
- [24] Heikkilä, J. and Silven, O., "A four-step camera calibration procedure with implicit image correction," *EEE Computer Society Conference on Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 Ipp.* 1106-1112, San Juan, Puerto Rico, Jun 1997.
- [25] Gremban, Keith D., Charles E. Thorpe, and Takeo Kanade. "Geometric camera calibration using systems of linear equations." *EEE International Conference on Robotics and Automation, 1988. Proceedings, 1988 I* pp. 562-567, Philadelphia, USA, Apr 1988.
- [26] online : http://www.vision.caltech.edu/bouguetj/calib_doc/htmls/parameters.html
- [27] Minxi Jin, Tsutomu Maruyama, "A Real-time Stereo Vision System Using a Tree-structured Dynamic Programming on FPGA", *FPGA '12 Proceedings of the ACM/SIGDA international symposium on Field Programmable Gate Arrays*, pp.21-24, Monterey, USA 2012.

저 자 소 개



한 동 일(정회원)
1988년 2월 고려대학교 전자전산
공학과 졸업(학사)
1990년 2월 한국과학기술원 전기
및 전자공학과 졸업(석사)
1995년 2월 한국과학기술원 전기
및 전자공학과 졸업(박사)

1995년 2월~2003년 2월 LG전자
디지털TV연구소 책임연구원.
2003년 3월~현재 세종대학교 컴퓨터공학과
정교수
<주관심 분야 : 영상 처리, 컴퓨터 비전, 디지털
TV, 디스플레이 화질 개선, 시스템 온 칩>



최 재 훈(학생회원)
2013년 2월 세종대학교 컴퓨터
공학과 졸업(학사)
2013년 3월~현재 세종대학교
컴퓨터공학과 석사과정
<주관심 분야 : 영상처리, 신호처
리, 컴퓨터 비전>



신 호 철(정회원)
1997년 2월 KAIST
기계공학과졸업 (학사)
1999년 2월 KAIST
기계공학과졸업 (석사)
2005년 2월 KAIST
기계공학과졸업 (박사)

2005년~현재 한국전자통신연구원 선임연구원
<주관심 분야 : 로봇 제어, 이미지/비디오 처리
및 분석>