

보정 이미지의 최 근접 좌표를 이용한 실시간 방사 왜곡 보정 하드웨어 설계

송 남 훈*, 이 준 환*

A Hardware Design for Realtime Correction of a Barrel Distortion Using the Nearest Pixels on a Corrected Image

Namhun Song*, Joonhwan Yi*

요 약

본 논문은 보정 이미지에서 최 근접 좌표를 이용한 방사 왜곡 보정 하드웨어 구조를 제안한다. 기존 보간법과는 달리 보정 이미지에서 최 근접한 좌표의 거리를 이용하기 때문에 이미지 전체 영역의 화질 향상과 함께 외각 영역에서 발생하는 계단 현상을 해결할 수 있다. 그러나 양 선형 보간법을 적용한 기존 구조에서 추가되는 연산으로 인해 하드웨어 크기가 증가한다. 이를 해결하기 위해 룩 업 테이블 구조를 제안하고, 코르딕 알고리즘을 적용한다. Design compiler를 이용하여 합성한 결과 보간법의 모든 과정을 하드웨어로 구현한 구조는 기존 구조에 비해 처리량이 높고, 차량용 후방 카메라의 경우 룩 업 테이블과 하드웨어를 함께 사용한 구조는 모든 과정을 하드웨어로 구현한 구조보다 하드웨어 크기를 10% 줄일 수 있다.

▶ Keywords : 방사형 왜곡 보정, 후방 카메라, 어안 렌즈, 보간법, 구면투영법 모델, 코르딕 알고리즘

Abstract

In this paper, we propose a hardware design for correction of barrel distortion using the nearest coordinates in the corrected image. Because it applies the nearest distance on corrected image rather than adjacent distance on distorted image, the picture quality is improved by the image whole area, solve the staircase phenomenon in the exterior area. But, because of additional arithmetic operation using design of bilinear interpolation, required arithmetic operation is

• 제1저자 및 교신저자 : 송남훈

• 투고일 : 2012. 11. 29. 심사일 : 2012. 12. 5. 게재확정일 : 2012. 12. 16.

* 광운대학교 컴퓨터공학과(Dept. of Computer Engineering, Kwangwoon University)

* 본 논문은 한국연구재단에서 지원하는 2012년도 일반연구자지원사업(No. 2012-0008329)의 연구수행으로 인한 결과물임을 밝힙니다.

increased. Look up table(LUT) structure is proposed in order to solve this, coordinate rotation digital computer(CORDIC) algorithm is applied. The results of the synthesis using Design compiler, the design of implementing all processes of the interpolation method with the hardware is higher than the previous design about the throughput, In case of the rear camera, the design of using LUT and hardware together can reduce the size than the design of implementing all processes with the hardware.

▶ Keywords : Barrel distortion correction, rear camera, fisheye lens, interpolation, spherical projection model, CORDIC algorithm

I. 서 론

어안렌즈는 화각이 180도 이상인 초 광각 렌즈로써 넓은 시야를 획득할 수 있어 다양한 산업 분야에 적용된다. 예를 들어, 감시 산업의 경우 시야의 사각 지역이 발생하지 않도록 전방위 360도 감시를 진행할 경우 일반 카메라 4대를 단일 어안 카메라로 해결이 가능하기 때문에 비용이 줄일 수 있고, 좁은 공간에서 많은 정보를 얻기 위하여 내시경[1]에 이용된다. 사물의 위치 정보를 얻기 위하여 자동차 전후방카메라[2], 로봇의 영상 센서[3] 등에도 사용된다. 그러나 이러한 장점에도 불구하고 일반 렌즈와는 다른 특수한 구성 때문에 영상의 중심으로부터 멀어질수록 방사 왜곡[4]이 생기며 영상의 정보가 압축되어 원하는 이미지를 얻기 힘들다.

왜곡을 제거한 후 올바른 영상을 얻기 위해 다양한 연구가 진행되었으며 좌표 변환과 보간법으로 분류할 수 있다. 좌표 변환의 경우, 측량적 방법과 비 측량적 방법으로 구분한다. 측량적 방법[5,6,7]은 제조 과정에서 발생하는 렌즈의 왜곡을 3차원 실제 이미지와 2차원 이미지의 관계를 추론하는 캘리브레이션(calibration) 과정을 통해 왜곡 보정 파라미터를 생성하는 것이다. 반면, 비 측량적 방법은 캘리브레이션 과정 없이 2차원 이미지만을 이용하여 다항식의 왜곡 계수를 반복적[8]으로 구하거나 기하학적 방법[9]을 이용하여 좌표 변환을 수행한다.

보간법의 경우, 어안 이미지에서 거리를 이용하는 최 근접 이웃 보간법(nearest neighbor interpolation), 양 선형 보간법(bilinear interpolation), 3차 회선 보간법(cubic convolution interpolation)[10,11]와 보정 이미지에서 거리를 이용하는 보정된 영상 위 양 선형 보간법(bilinear interpolation on corrected image)[12] 및 보정된 영상 위 최 근접 좌표를 이용한 보간법(nearest neighbor based

interpolation on corrected image)[13]이 있다.

실시간으로 왜곡 보정 처리가 가능한 하드웨어 연구는 다음과 같다. Asari[14]는 최소 자승법(least square estimation method)을 이용하여 좌표 변환을 수행하고, 양 선형 보간법을 적용한 하드웨어 구조를 제안했다. Ngo[15]는 실시간 처리를 위하여 파이프라인 구조를 제안했다. Pei[16]는 좌표 변환 식에 홀수 차수 계수만 사용함으로써 최대 신호 대 잡음비(peak signal to noise ratio)가 감소되거나 제곱근 연산을 생략할 수 있는 구조를 제안했다. Chen[1]은 호너스 알고리즘(honor's algorithm)을 이용하여 좌표 변환 식을 다항식에서 단항식으로 변경함으로써 하드웨어 크기를 줄일 수 있는 구조를 제안했다.

하드웨어 구현에 관련된 기존 연구는 대부분 좌표 변환에 초점이 맞춰져있고, 어안 이미지에서 거리를 이용한 보간법을 적용했다. 그러나 방사 왜곡이 발생하기 때문에 어안 이미지에서 좌표 간의 거리는 실제 거리를 반영하지 않는다. 한재영[12]은 어안 이미지에서 인접한 좌표 4개를 보정 이미지로 사상한 후 보정 이미지에서 거리를 이용한 보간법을 제안했다. 어안 이미지에서 거리를 이용한 보간법보다 보정 성능이 우수하나 보정 이미지 외곽 영역에서 발생하는 계단 현상을 해결하지 못했다.

이윤진[13]은 어안 이미지에서 인접한 좌표가 보정 이미지에서는 인접하지 않을 수 있기 때문에 보정 이미지에 최 근접한 4개 좌표의 거리를 이용한 보간법을 제안했다. 이미지의 외곽으로 갈수록 방사 왜곡이 커지기 때문에 보정된 영상 위 최 근접 좌표를 이용한 보간법은 기존 보간법에 비해 최대 신호 대 잡음비가 높으며 이미지의 중심 영역보다 외곽 영역에서 화질 향상이 우수하다. 이러한 이유로 기존보다 더 넓은 영역을 모니터로 출력할 수 있기 때문에 실제 정보를 더 많이 획득할 수 있다. 그러나 소프트웨어를 이용한 연산 수행시간이 양 선형 보간법에 비해 9.44배 증가하기 때문에 실시간으로 처리하기 어렵고, 요구되는 산술 연산 역시 증가한다.

본 논문에서는 보정된 영상 위 최 근접 좌표를 이용한 보 간법이 적용된 하드웨어를 구현한다. 먼저, 보정된 영상 위 최 근접 좌표를 이용한 보간법의 모든 연산 과정을 하드웨어 로 구현한 구조는 양 선형 보간법을 적용한 기존 구조보다 클 록 주파수(clock frequency)가 높기 때문에 처리량 (throughput)이 높다. 그러나 추가되는 연산으로 인해 여전 히 하드웨어 크기가 증가하기 때문에 최 근접한 4개 좌표를 찾는 과정에서의 정보를 록 업 테이블에 저장하는 구조를 제 안한다. 록 업 테이블과 하드웨어를 함께 사용하는 구조는 록 업 테이블의 크기가 해상도에 비례하기 때문에 자동차 후방 카메라와 같은 저 해상도 응용에 적용할 경우, 하드웨어 크기 가 감소한다. 본 논문은 다음과 같이 구성된다. 2장에서 관련 연구를 살펴본다. 3장에서 모든 연산 과정을 하드웨어로 구현 한 구조 및 록 업 테이블과 하드웨어를 함께 사용하는 구조를 설명한다. 4장은 구현 및 고찰, 5장에서는 결론을 맺는다.

II. 관련 연구

1. 구면 투영법 모델

구면 투영법 모델(spherical projection model)[9]은 렌즈 중심에서 렌즈 위의 평면으로 사영을 수행, 어안 렌즈를 완전한 반구라고 가정한다. Fig. 1은 구면 투영법 모델을 나타낸다. 는 카메라의 초점 거리, O 는 이미지 평면에서의 중심, O_c 는 카메라 렌즈의 중심, (x_d, y_d) 는 카메라 렌즈로부터 획득된 왜곡 영상의 지점, (x_c, y_c) 는 카메라 렌즈로부터 왜곡되지 않은 이상적인 지점, L 은 카메라 렌즈의 중심으로부터 왜곡되지 않은 이상적인 지점까지의 거리를 나타낸다. 피타고라스 정리를 이용하면 식(1), (2)가 생성된다.

$$(x_c, y_c) = \left(\frac{x_d F}{F^2 - (x_d^2 + y_d^2)}, \frac{y_d F}{\sqrt{F^2 - (x_d^2 + y_d^2)}} \right) \quad (1)$$

$$(x_d, y_d) = \left(\frac{x_c F}{\sqrt{F^2 + (x_c^2 + y_c^2)}}, \frac{y_c F}{\sqrt{F^2 + (x_c^2 + y_c^2)}} \right) \quad (2)$$

2. 보정된 영상 위 양 선형 보간법

어안 이미지에서 인접한 좌표의 거리는 1이다. 그러나 방사 왜곡이 발생하기 때문에 이미지의 중심에서 멀어질수록 보정 이미지로 사상된 인접한 좌표의 거리는 점점 증가하며 어안 이미지의 거리를 보간 계수로 선택하는 것은 보정 성능이

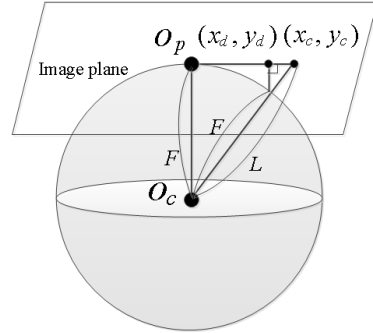


그림 1. 구면투영법 모델
Fig. 1. Spherical projection model

떨어진다. [12]에서는 보정 이미지에서 거리를 이용한 보정된 영상 위 양 선형 보간법을 제안했다. 어안 렌즈 왜곡 보정 처리 과정은 다음과 같다. <1>보정 이미지의 좌표 $C(x_c, y_c)$ 에서 어안 이미지의 좌표 $D(x_d, y_d)$ 로 사상. <2> $D(x_d, y_d)$ 에 인접한 4개의 정수 좌표 $D_k(x_d^k, y_d^k) (k=0,1,2,3)$ 를 보정 이미지로 사상. D_k 를 보정 이미지로 사상한 좌표를 $C_k(x_c^k, y_c^k) (k=0,1,2,3)$ 라 한다. <3>어안 이미지의 화소 값 $P(x_d^k, y_d^k) (k=0,1,2,3)$ 획득. <4> $C(x_c, y_c)$ 와 $C_k(x_c^k, y_c^k)$ 의 거리 계산. <5>보간법 수행.

Fig. 2는 보정된 영상 위 양 선형 보간법의 연산 과정을 나타낸다. 삼각 함수, 내분, 외분을 이용하여 좌표 T_2 을 계산하는 수식은 식(3)과 같다.

$$T_1 = (x_c, y_c^0 \pm (|x_c - x_c^0| \times \tan \alpha)),$$

$$T_2 = (x_c, y_c^2 \pm (|x_c - x_c^2| \times \tan \beta)). \quad (3)$$

이때 $C_0(x_c^0, y_c^0)$ 과 T_1 사이의 거리 d_1 은 식(4)와 같다.

$$d_1 = \sqrt{(x_c^0 - x_c)^2 + (|x_c - x_c^0| \times \tan \alpha)^2}. \quad (4)$$

$d_2 \sim d_6$ 은 식(4)와 같이 계산할 수 있다. d_k 와 $P(x_d^k, y_d^k)$ 를 이용하여 T_1, T_2 의 화소 값 $P(T_1), P(T_2)$ 와 (x_c, y_c) 의 화소 값 $P(x_c, y_c)$ 를 구할 수 있다. 해당 과정은 식(5)와 같다.

$$P(T_1) = \frac{d_2}{d_1 + d_2} P(x_d^0, y_d^0) + \frac{d_1}{d_1 + d_2} P(x_d^1, y_d^1)$$

$$P(T_2) = \frac{d_4}{d_3 + d_4} P(x_d^2, y_d^2) + \frac{d_3}{d_3 + d_4} P(x_d^3, y_d^3)$$

$$P(x_c, y_c) = P(T_1) \left(\frac{d_6}{d_5 + d_6} \right) + P(T_2) \left(\frac{d_5}{d_5 + d_6} \right) \quad (5)$$

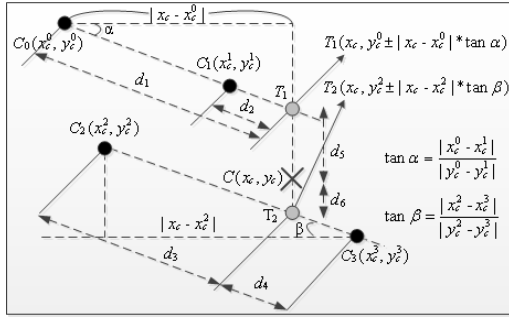


그림 2. 보정된 영상 위 양 선형 보간법
Fig. 2. Bilinear interpolation on corrected image

어안 이미지에서 거리를 이용한 보간법보다 보정 성능이 우수하다는 장점이 있으나 추가되는 연산과 복잡한 보간 과정으로 인해 하드웨어가 커진다는 단점이 있다. 이를 해결하기 위해 구면 투영법 모델을 적용하여 룩 업 테이블의 크기를 1/4로 줄이는 방법을 제안했다.

3. 보정된 영상 위 최 근접 좌표를 이용한 보간법

보정된 영상 위 양 선형 보간법의 경우 중심 좌표에서 멀어질수록 어안 이미지에서 인접한 좌표가 보정 이미지에서는 인접하지 않은 경우가 발생한다. 이를 해결하기 위해 어안 이미지에서 인접한 16개 정수 좌표를 보정 이미지로 사상한 후 (x_c, y_c) 에 최 근접한 4개 좌표 거리를 이용하는 보간법을 제안했다. 어안 렌즈 왜곡 보정 처리 과정은 다음과 같다.

〈1〉 $C(x_c, y_c)$ 에서 $D(x_d, y_d)$ 로 사상. 〈2〉 $D(x_d, y_d)$ 에 인접한 16개 정수 좌표 $D_m(x_d^m, y_d^m)$ ($m=0 \sim 15$)를 보정 이미지로 사상. 〈3〉 $C(x_c, y_c)$ 와 사상된 16개 좌표 $C_m(x_c^m, y_c^m)$ ($m=0 \sim 15$)의 거리 d_m 계산. 〈4〉 $C(x_c, y_c)$ 에서 최 근접 4개 좌표 $C_k(x_c^k, y_c^k)$ 선택. 〈5〉보간법 수행. Fig. 3은 어안 이미지와 보정 이미지의 사상 관계를 나타낸다. Fig. 3(a)는 어안 이미지에서 좌측 상단을 확대한 것을 나타낸다. Fig. 3(b)는 Fig. 3(a)에서 표현된 16개 좌표가 보정 이미지로 사상된 것을 나타낸다. $D(x_d, y_d)$ 에 인접한 좌표는 6, 7, 10, 11이지만 $C(x_c, y_c)$ 에 최 근접 좌표는 3, 6, 7, 9인 것을 알 수 있다. 식(6)은 $C(x_c, y_c)$ 와 $C_m(x_c^m, y_c^m)$ 의 거리 계산 수식이다.

$$d_m = (x_c - x_c^m)^2 + (y_c - y_c^m)^2 \tag{6}$$

식(7)은 $C_k(x_c^k, y_c^k)$ 에 해당하는 d_k 를 이용하여 보간 계수 ic_k ($k=0,1,2,3$)와 ic_{sum} 을 생성하는 것을 나타낸다.

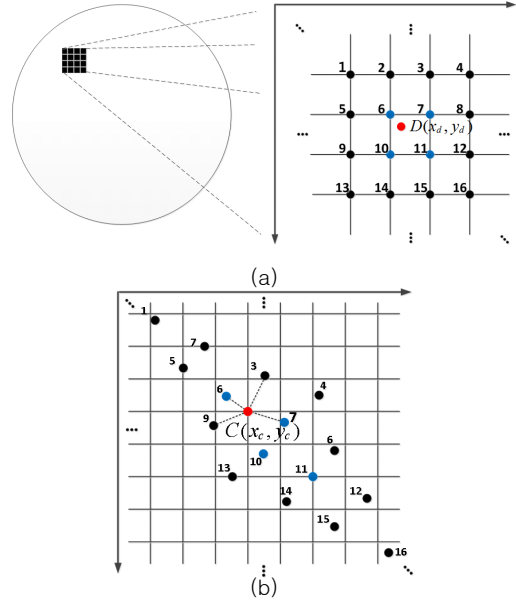


그림 3. 어안 이미지와 보정 이미지 상에서의 좌표관계
Fig. 3. Coordinate relationship between distorted image and corrected image

$$ic_0 = d_1 d_2 d_3, ic_1 = d_0 d_2 d_3, ic_2 = d_0 d_1 d_3, ic_3 = d_0 d_1 d_2$$

$$ic_{sum} = \sum_{k=0}^3 ic_k \tag{7}$$

식(8)은 ic_k 와 ic_{sum} , $P(x_d^k, y_d^k)$ 를 이용하여 $P(x_c, y_c)$ 을 계산하기 위한 보간법 수식을 나타낸다.

$$P(x_c, y_c) = \sum_{k=0}^3 ic_k \times P(x_d^k, y_d^k) / ic_{sum} \tag{8}$$

III. 제안 구조

보정된 영상 위 최 근접 좌표를 이용한 보간법이 적용되어 실시간 어안 렌즈 왜곡 보정을 하기 위한 2가지 구조를 제안한다. 먼저, 보간법의 모든 연산 과정을 하드웨어로 구현한 구조(HW only design)를 설명한 후 자동차 후방카메라와 같이 저 해상도 응용에서 하드웨어만 사용한 구조보다 하드웨어 크기를 줄일 수 있는 하드웨어와 룩 업 테이블을 함께 사용한 구조(HW+LUT design)를 설명한다.

1. 하드웨어만 사용한 구조

Fig. 4는 하드웨어만 사용한 구조를 나타내며 처리 과정은 크게 8 단계로 진행된다. 〈1〉Stage 1~7에서는 $C(x_c, y_c)$

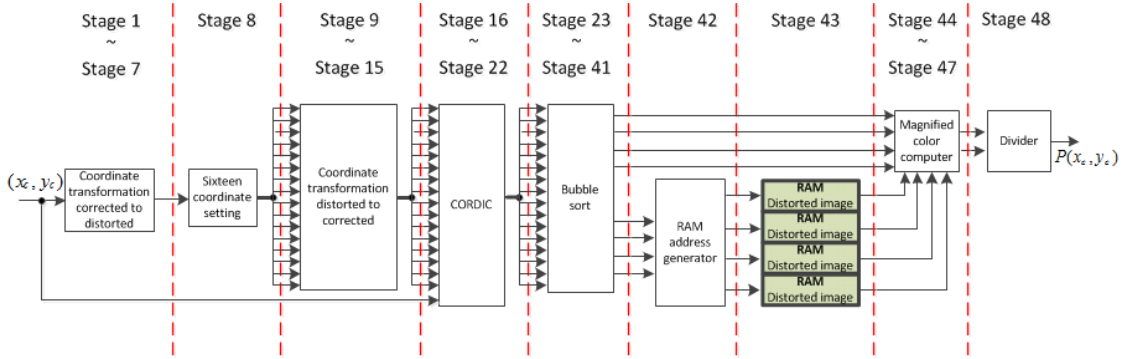


그림 4. 하드웨어만 사용한 구조
Fig. 4. HW only design

를 어안 이미지 좌표 x_d, y_d 로 사상한다. <2>Stage 8에서는 $D(x_d, y_d)$ 에 인접한 16개 좌표를 $D_m(x_d^m, y_d^m)$ 로 설정한다. <3>Stage 9~15에서는 $D_m(x_d^m, y_d^m)$ 을 보정 이미지의 16개 좌표 $C_m(x_c^m, y_c^m)$ 로 사상한다. <4>Stage 16~22에서는 코르딕 알고리즘(CORDIC algorithm)을 이용하여 거리 d_m 을 계산한다. <5>Stage 23~41에서는 버블 소트 알고리즘을 적용하여 d_k 를 찾는다. <6>Stage 42~43에서는 d_k 에 해당하는 어안 이미지의 좌표 $D_k(x_d^k, y_d^k)$ 를 이용하여 화소 값 $P(x_d^k, y_d^k)$ 이 저장된 RAM에 접근한다. <7>Stage 44~47에서는 ic_{sum} 을 계산하고, ic_c 와 $P(x_d^k, y_d^k)$ 을 이용하여 식(8)의 분자를 계산한다. <8>끝으로 나눗셈기를 이용하여 ic_{sum} 을 나눔으로써, 보간된 화소 값 $P(x_c, y_c)$ 를 식(8)에 따라 계산한다. 해당 구조에서 사용된 각 block의 동작을 설명한다.

Coordinate Transformation

식(1), (2)를 적용하여 어안 이미지와 보정 이미지 간에 좌표 변환을 수행한다. 요구되는 제곱근, 나눗셈 연산 유닛은 DesignWare IP[17]를 이용한다.

먼저, stage 1~7에서는 $C(x_c, y_c)$ 와 제곱근 유닛을 이용하여 식(2)의 분모 $F^2 + (x_c^2 + y_c^2)$ 를 계산한 후 나눗셈 유닛을 이용하여 $\frac{F}{\sqrt{F^2 + (x_c^2 + y_c^2)}}$ 를 계산한다. 그 후 x_c, y_c 와 곱셈 연산을 통해 $D(x_d, y_d)$ 를 계산한다.

Sixteen coordinate setting block에서는 Fig.3(a)와 같이 $D(x_d, y_d)$ 에 인접한 16개 좌표를 $D_m(x_d^m, y_d^m)$ 로 설정

한다. Stage 9~15에서는 $D_m(x_d^m, y_d^m)$ 을 이용하여 $C_m(x_c^m, y_c^m)$ 을 계산하며 처리 과정은 stage 1~7와 같다.

CORDIC

$C(x_c, y_c)$ 와 $C_m(x_c^m, y_c^m)$ 의 거리 d_m 을 코르딕 알고리즘 [18]을 이용하여 계산한다. 해당 알고리즘은 매 반복마다 덧셈기와 쉬프트 연산만을 사용하기 때문에 하드웨어 자원이 적게 요구된다. 그러나 최종 반복을 수행한 후 반복 횟수에 따라 결정되는 고정된 상수 값을 곱해야 한다는 단점이 있다. 그러나 보정된 영상 위 최 근접 좌표를 이용한 보간법에 코르딕 알고리즘을 적용하면 최종 반복에서 곱셈 연산을 수행하지 않아도 되기 때문에 곱셈기를 4개 줄일 수 있다. Fig. 5는 cordic top block을 나타내며 $C(x_c, y_c)$ 와 $C_m(x_c^m, y_c^m)$ 의 뺄셈 연산을 통해 초기 좌표 (x_0^m, y_0^m) 을 생성하는 cordic init block와 d_m 을 계산하는 cordic cal block으로 구성된다. 식(9)는 cordic cal block에 적용되는 수식을 나타낸다. 변수마다 i 는 반복 횟수를 나타내며 m 은 16개 좌표의 순서를 나타낸다. s_i^m 은 y_i^m 의 부호에 따라 1 or -1이 된다.

$$\begin{aligned} x_{i+1}^m &= x_i^m - s_i^m y_i^m 2^{-i} \\ y_{i+1}^m &= y_i^m + s_i^m x_i^m 2^{-i} \\ s_i^m &= \begin{cases} +1, & y_i^m < 0 \\ -1, & \text{else} \end{cases} \end{aligned} \quad (9)$$

n 번 반복한 후의 거리 d_m 은 식(10)와 같다. 여기서 α_n 은 n 번 반복했을 때의 상수 값을 나타낸다.

$$d_m = x_n^m \times \alpha_n \quad (10)$$

식(7)에 식(10)를 적용하면 분모와 분자에 동일한 $(\alpha_n)^3$ 이 곱해진다. 따라서 최종 반복 후에 α_n 를 곱하지 않아도 된다.

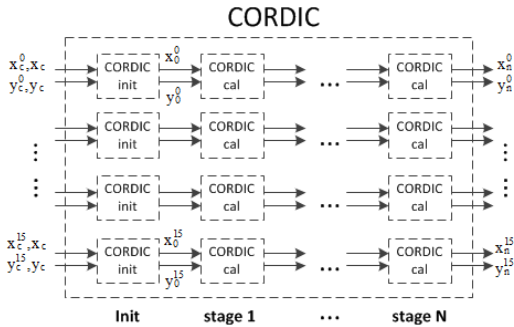


그림 5. 코르딕 블록
Fig. 5. CORDIC block

Bubble Sort

x, y_c)에 최 근접한 4개 좌표 $C_k(x_c^k, y_c^k)$ 에 해당하는 어안 이미지의 좌표 $D_k(x_d^k, y_d^k)$ 와 d_k 를 찾기 위해 버블 소트를 이용한다. Fig. 6은 4개의 데이터 중에서 가장 작은 데이터를 찾기 위한 block을 나타내고, comparison and result(CAR) block와 update block으로 구성된다. CAR block은 4개의 데이터와 이에 해당하는 인덱스를 입력 받아 이 중에서 가장 작은 데이터와 인덱스를 update block으로 넘긴다. Update block은 2 단계 파이프라인으로 구성된다. 1 단계에서는 2개의 데이터와 인덱스를 입력 받아 가장 작은 데이터와 이에 해당하는 인덱스를 출력한다. 만약, 선택된 데이터 다음으로 작은 데이터를 찾기 원하는 경우, 2 단계의 dec block에 인덱스를 전달한다. Dec block은 1 단계에서 선택된 데이터가 다음 번 정렬 과정에서 선택되지 않도록 해당 데이터를 가장 큰 값으로 변경하기 위한 제어 신호를 출력한다. 제어 신호의 비트 길이와 개수는 데이터와 동일하며 인덱스에 해당하는 제어 신호는 가장 큰 값을 표현하기 위해 msb는 0, 나머지 bit는 1로 설정하고, 그 외의 제어 신호는 0을 출력한다. 마지막으로 각 데이터와 제어 신호의 or 연산을 통해 가장 작은 값을 가장 큰 값으로 변경되고, 나머지 신호는 값을 그대로 유지한다. Fig. 7은 16개 거리 d_m 중에서 가장 작은 4개 거리 d_k 와 이에 해당하는 인덱스를 찾기 위한 block이다. 총 19 단계 파이프라인, 4 단계로 나뉘지며 각 단계에서 순차적으로 d_0, d_1, d_2, d_3 와 이에 해당하는 인덱스가 출력, 마지막 단계에서는 다음 번의 정렬을 고려하지 않기 때문에 update block은 1 단계만 적용한다.

RAM Address Generator

Bubble sort block으로부터 d_k 에 해당하는 인덱스와

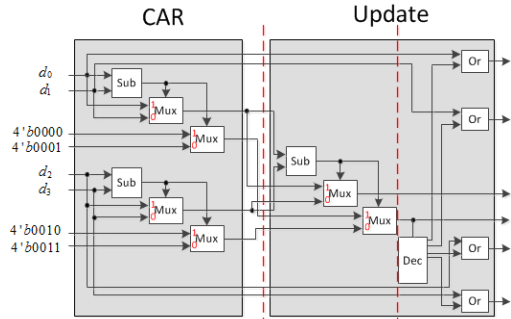


그림 6. 4 입력 버블 정렬 블록
Fig. 6. 4 input bubble sort block

$D(x_d, y_d)$ 를 입력 받아 화소 $P(x_d^k, y_d^k)$ 가 저장된 distorted image RAM의 입력 주소를 생성한다.

Magnified Color Computer

화소 값 $P(x_c, y_c)$ 를 생성하기 위해 식(8)의 분모 ic_{sum} 와 분자 $\sum_{k=0}^3 ic_k \times P(x_d^k, y_d^k)$ 를 계산한다.

Divider

앞서 구한 식 (8)의 분자 값을 나눗셈 유닛을 이용하여 식 (8)의 연산을 완료한다. 즉, 보정된 color $P(x_c, y_c)$ 를 계산한다.

2. 하드웨어와 룩 업 테이블을 함께 사용 구조

구면 투영법 모델은 어안 렌즈를 완전한 반구라고 가정하기 때문에 보정 이미지의 중심 좌표 $Cc(x_c, y_c)$ 로부터 원점 대칭 방향으로 방사 왜곡이 발생하며 한 사분면의 룩 업 테이블 정보만으로 나머지 사분면의 정보를 얻을 수 있다.

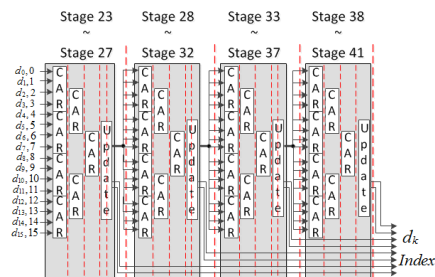


그림 7. 16 입력 버블 정렬 블록
Fig. 7. 16 input bubble sort block

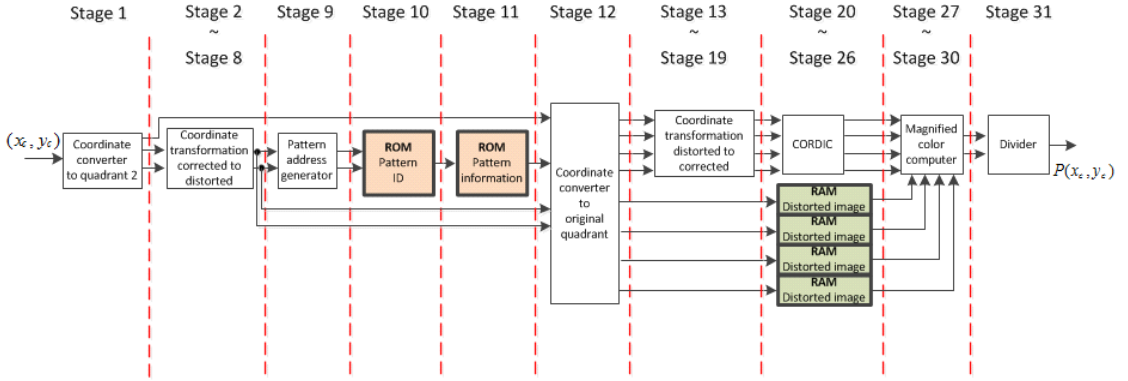


그림 8. 하드웨어와 룩 업 테이블을 함께 사용 구조
Fig. 8. HW+LUT design

Fig. 8은 하드웨어와 룩 업 테이블을 함께 사용한 구조를 나타내며 처리 과정은 7 단계로 진행된다. <1>Stage 1에서는 룩 업 테이블의 크기를 1/4로 줄이기 위해 (x_c, y_c) 를 보정 이미지 2 사분면 좌표 $C2(x_{d2}, y_{d2})$ 로 사상한다. <2>Stage 2~8에서는 $C2(x_{d2}, y_{d2})$ 를 어안 이미지 2 사분면 좌표 $D2(x_{d2}, y_{d2})$ 로 이동한다. <3>Stage 9~12에서는 $D2(x_{d2}, y_{d2})$ 를 이용하여 보정 이미지 상의 최 근접 좌표 $C2_k(x_{d2}^k, y_{d2}^k)$ 에 해당하는 어안 이미지의 좌표 $D2_k(x_{d2}^k, y_{d2}^k)$ 를 계산하고, $C(x_c, y_c)$ 가 속한 사분면 정보를 이용하여 화소 값 $P(x_d^k, y_d^k)$ 이 저장된 RAM의 입력 주소를 계산한다. <4>Stage 13~19에서는 $D2_k(x_{d2}^k, y_{d2}^k)$ 를 $C2_k(x_{d2}^k, y_{d2}^k)$ 로 이동한다. <5>Stage 20~26에서는 식 (6)에 보인 거리 d_k 를 코르티 알고리즘을 이용하여 구한다. <6>Stage 27~30에서는 식 (7)에 보인 ic_k 와 ic_{sum} 을 계산하고, 이중 ic_k 와 $P(x_d^k, y_d^k)$ 을 이용하여 식(8)의 분자를 계산한다. <7>나눗셈 유닛을 이용하여 ic_{sum} 을 나눔으로써, $P(x_c, y_c)$ 를 식(8)에 따라 계산한다. 해당 구조의 block 동작을 설명한다.

Coordinate Converter to Quadrant 2

$C(x_c, y_c)$ 에 최 근접한 4개 좌표에 해당하는 어안 이미지의 좌표 정보를 저장하는 룩 업 테이블의 크기를 1/4로 줄이기 위해 $C(x_c, y_c)$ 를 한 사분면 좌표로 이동하고, 해당 사분면의 위치 정보를 출력한다. Fig. 9는 $Cc(x_c, y_c)$ 를 기준으로 원점 대칭이 발생하여 $C(x_c, y_c)$ 가 각 사분면

$Ck(x_{dk}, y_{ck})$ ($k=0,1,2,3$)로 이동한 것을 나타낸다. $C(x_c, y_c)$ 를 $C2(x_{d2}, y_{d2})$ 로 이동하면 (0,0)부터 시작하기 때문에 ROM의 입력 주소를 생성하는 pattern address generator block에서 (0,0)로 좌표를 이동하기 위한 추가 연산이 필요하지 않다. 식(11)은 $C(x_c, y_c)$ 가 속한 사분면 정보를 표현하기 위한 연산 과정을 나타낸다.

$$\begin{aligned} x &= x_c - xc_c \\ y &= y_c - yc_c \end{aligned} \tag{11}$$

$C(x_c, y_c)$ 와 $Cc(x_c, y_c)$ 의 좌표 차는 (x, y) 가 되며 표 1와 같이 (x, y) 의 msb bit (x_{msb}, y_{msb})를 이용하여 현재 $C(x_c, y_c)$ 의 사분면을 판단한다.

표 1. $C(x_c, y_c)$ 의 사분면 정보
Table 1. Quadrant information of $C(x_c, y_c)$

sb	y_{msb}	사분면
1	0	1
1	1	2
0	1	3
0	0	4

Nearest Neighbor Patterns

보정 이미지 상에서 최 근접 좌표 $C2_k(x_{d2}^k, y_{d2}^k)$ 에 해당하는 어안 이미지의 좌표 $D2_k(x_{d2}^k, y_{d2}^k)$ 를 저장하는 룩 업 테이블의 크기를 줄이기 위해 상대 좌표를 저장하는 방법과 최 근접 좌표들의 패턴을 이용하는 방법을 적용한 2 단계 룩 업 테이블 구조와 룩 업 테이블의 입력 주소를 생성하는 방법에 대해 설명한다.

상대 좌표를 저장하는 방법은 어안 이미지 상에서 선택되

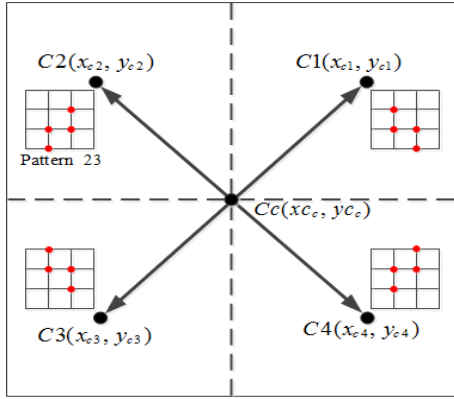


그림 9. 방사 왜곡의 대칭성
Fig. 9. Symmetry property of barrel distortion

는 4개 좌표를 저장할 때 절대 좌표가 아닌 상대 좌표를 룩업 테이블에 저장하는 방법이다. 최 근접 좌표들의 패턴을 이용하는 방법은 보정 이미지 상에서 최 근접 좌표에 해당하는 어안 이미지의 좌표가 일정한 패턴으로 발생하는 것을 이용한 방법이다. 보정된 영상 위 최 근접 좌표를 이용한 보간법은 어안 이미지 상에서 인접한 16개 좌표 중 4개 좌표를 선택한다. 이때 발생하는 경우의 수는 최대 $16 = 1820$ 개가 되며 어안 이미지의 해상도가 1024x1024인 경우에 발생하는 패턴은 132개이다. 위 방법을 적용하면 $D_k(x_{d2}^k, y_{d2}^k)$ 의 모든 좌표를 룩업 테이블에 저장하는 구조보다 룩업 테이블의 크기를 줄일 수 있다. Fig. 3에서 $D(x_d, y_d)$ 에 인접한 16개 좌표 중 보정 이미지에서 선택되는 4개 좌표 3, 6, 7, 9는 하나의 패턴이 된다. 정수 부분을 표현한 좌표 6을 기준으로 인접한 15개 좌표의 상대적인 거리 범위는 -1 ~ 2가 되며 각 좌표마다 범위를 표현하기 위해 6 bit, 4개 좌표를 표현하기 위해 24 bit가 필요하다. 좌표 6을 (0,0)으로 설정하면 (-1,1), (0,0), (0,1), (1,1)이 되고, bit로 표현 시 (111,001), (000,000), (000,001), (001,001)가 된다. 2 단계 룩업 테이블 구조는 위 2가지 방법을 적용하기 위한 룩업 테이블 구조이다.

먼저, $C2(x_{c2}, y_{c2})$ 를 이용하여 패턴 번호를 저장하는 pattern ID ROM에 접근한다. 이때 보정 이미지의 width와 height를 CI_{width} , CI_{height} , N_p 를 패턴 번호로 정의하면 2 단계 룩업 테이블 구조의 크기는 식 (12)와 같다. 아래 식에서 LUT_{PID} 는 pattern ID ROM의 크기, LUT_{PIF} 는 pattern information ROM의 크기, LUT_{TWO} 는 최 근접 좌표를 저장하는 전체 ROM의 크기를 나타낸다.

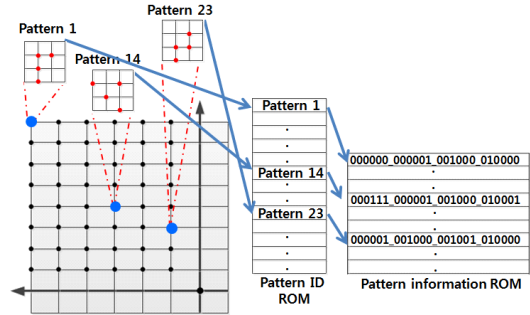


그림 10. 2 단계 룩업 테이블 구조 및 동작
Fig. 10. Two level LUT structure and operation

$$LUT_{PID} = \frac{CI_{width}}{2} \times \frac{CI_{height}}{2} \times \lceil \log_2 N_p \rceil$$

$$LUT_{PIF} = 24 \times N_p$$

$$LUT_{TWO} = LUT_{PID} + LUT_{PIF} \tag{12}$$

Pattern ID ROM에 접근하기 위한 주소 생성은 pattern address generator block에서 이뤄지며 식(13)와 같다.

$$ROM_{address} = \left(\frac{CI_{height}}{2} \times x_{c2} \right) + y_{c2} \tag{13}$$

각 패턴 번호는 해당 pattern의 information을 저장하는 pattern information ROM의 입력 주소가 된다. Fig. 10은 보정 이미지 2 사분면에서 각각의 $C2(x_{c2}, y_{c2})$ 가 pattern ID ROM을 거쳐 패턴 번호 1, 14, 23을 얻은 후 pattern information ROM에 접근하여 24 bit로 구성된 패턴 정보를 출력하는 것을 나타낸다.

Coordinate Converter to Original Quadrant

$D2(x_{d2}, y_{d2})$ 와 24 bit 정보를 입력 받아 $D2_k(x_{d2}^k, y_{d2}^k)$ 를 생성한 후 $C(x_c, y_c)$ 의 사분면 정보 (x_{msb}, y_{msb})를 이용하여 어안 이미지의 최소 값 $P(x_d^k, y_d^k)$ 이 저장된 distorted image RAM의 입력 주소 식(14)를 생성한다.

$$RAM_{address} = (DI_{height} \times x_d^k) + y_d^k \tag{14}$$



그림 11. 왼쪽 상단 1/16 영역 보정 결과
Fig. 11. Correction result of upper left (sixteenth part)

IV. 구현 및 고찰

1. 소프트웨어 실험 환경

a. 최대 신호 대 잡음비

Kodak lossless true color image[19]를 이용하여 각 보간법의 최대 신호 대 잡음비를 알아본다. Focal length 20mm를 적용하여 원본 이미지를 어안 이미지로 변환한 후 각 보간법을 적용하여 생성된 보정 이미지와 원본 이미지의 최대 신호 대 잡음비를 비교하였다. 표 2는 테스트에 사용된 24장 이미지에 각 보간법을 적용한 후 이미지 전체 영역과 왼쪽 상단 1/16 영역의 최대 신호 대 잡음비의 평균을 측정한 것이다. 전체 영역의 경우 보정된 영상 위 최 근접 좌표를 이용한 보간법은 양 선형 보간법에 비해 0.426 dB, 외각 영역의 경우 0.917 dB 높다. 보정된 영상 위 최 근접 좌표를 이용한 보간법은 보간법 중에서 가장 최대 신호 대 잡음비가 높은 것을 확인 할 수 있고, 중심 영역보다 외각 영역에서의 성능 향상이 우수하다. Fig. 11은 [19]에서 13번 이미지의 왼쪽 상단 1/16 영역에 대해 양 선형 보간법(왼쪽)과 보정된 영상 위 최 근접 좌표를 이용한 보간법(오른쪽)을 적용한 것이다. 오른쪽의 그림이 더 선명하며 계단 현상이 발생하지 않는 것을 확인할 수 있다.

표 2. 보간법 종류에 따른 최대 신호 대 잡음비 비교
Table 2. Peak signal to noise ratio comparison according to the kind of the interpolation method(unit: dB)

	최 근접 이웃	양 선형	3차 회선	(12)	(13)
전체	24.955	26.926	26.727	27.218	27.352
왼쪽 상단	27.478	29.656	29.359	29.894	30.573

b. 수행 시간

표 3은 보정 이미지(800x480)를 1, 30 프레임 처리하는

데 소요되는 시간을 Intel parallel Studio를 이용하여 측정한 결과를 나타낸다. 보정된 영상 위 최 근접 좌표를 이용한 보간법은 양 선형 보간법에 비해 수행 시간이 9.44배 증가하고, 30 프레임을 처리하는데 3.11초가 걸리므로 실시간 처리가 어려운 것을 알 수 있다.

표 3. 보간법 종류에 따른 실행 시간 비교
Table 3. The run time comparison according to the kind of the interpolation method (unit: second)

	양 선형 보간법	(13)
1 프레임	0.01(1.0)	0.1037(9.44)
30 프레임	0.33(1.0)	3.1113(9.44)

c. 요구되는 산술 연산 횟수

표 4는 각 보간법을 적용하여 1 프레임 처리하는데 요구되는 산술 연산 횟수를 나타낸다. 좌표 변환(보정-어안)은 (x_2, y_{e2}) 에서 $D2(x_{d2}, y_{d2})$ 로 사상하는 과정, 좌표 변환(어안-보정)은 $D2(x_{d2}, y_{d2})$ 에 인접한 16개 좌표를 보정 이미지로 사상하는 과정, 거리 계산은 $C2(x_{e2}, y_{e2})$ 와 16개 좌표 거리를 계산하는 과정, 정렬은 (x_d^k, y_d^k) 를 찾는 과정, 식(7), (8)은 $ic_k, ic_{sum}, P(x_d^k, y_d^k)$ 를 이용하여 $P(x_e, y_e)$ 을 계산하는 과정이다. 보정된 영상 위 최 근접 좌표를 이용한 보간법은 양 선형 보간법에 비해 좌표 변환(어안-보정), 거리 계산, 정렬이 추가된다. 정렬은 16개 데이터에 대해 버블 소트 알고리즘을 적용한다. 이때 인접한 데이터를 비교하는 과정에서 요구되는 덧셈 횟수만 계산한다. 양 선형 보간법에서 요구되는 산술 연산의 전체 횟수를 1로 정규화 시 보정된 영상 위 최 근접 좌표를 이용한 보간법은 덧셈 17.3배, 곱셈 15.7배, 나눗셈 18배, 루트 33배 증가한다.

표 4. 보간법 종류에 따라 요구되는 산술 연산 횟수
Table 4. The arithmetic operation times required according to the kind of interpolation method

	양 선형 보간법	(12)	(13)
좌표 변환 (보정-어안)	덧셈 2번 곱셈 4번 나눗셈 1번 루트 1번	덧셈 2번 곱셈 4번 나눗셈 1번 루트 1번	덧셈 2번 곱셈 4번 나눗셈 1번 루트 1번
좌표 변환 (어안-보정)	N/A	덧셈 8번 곱셈 16번 나눗셈 4번 루트 4번	덧셈 32번 곱셈 64번 나눗셈 16번 루트 16번
거리 계산		덧셈 22번 곱셈 14번	덧셈 48번 곱셈 32번

		나눗셈 2번 루트 4번	루트 16번
정렬		N/A	덧셈 120번
식(7) 식(8)	덧셈 10번 곱셈 3번	덧셈 9번 곱셈 6번 나눗셈 6번	덧셈 6번 곱셈 10번 나눗셈 1번
산술 연산 횟수	덧셈 12번 곱셈 7번 나눗셈 1번 루트 1번	덧셈 41번 곱셈 40번 나눗셈 13번 루트 9번	덧셈 208번 곱셈 110번 나눗셈 18번 루트 33번

2. 하드웨어 실험 환경

a. 코르딕 반복 횟수와 코르딕 정밀도의 관계

Fig. 12는 1024x1024 해상도의 어안 및 보정 이미지에 대해 코르딕 알고리즘을 이용하여 d^k 를 계산하는 과정에서 반복 횟수(색상 선)와 매 반복마다 적용되는 정밀도(x 축)가 변화함에 따라 발생하는 오차(y 축)를 나타낸다. 여기서 정밀도는 d^k 의 소수 부분의 비트 길이를 나타낸다. 정밀도가 증가함에 따라 오차가 적어지며 반복수가 6 이상부터는 오차가 거의 발생하지 않는 것을 알 수 있다.

b. 코르딕 정밀도와 보간 계수 정밀도의 관계

Fig. 13은 1024x1024 해상도의 어안 및 보정 이미지에 대해 코르딕 반복 횟수가 6인 경우, d^k 를 계산하는 과정에서 매 단계마다 적용되는 정밀도(색상 선)와 보간 계수 생성 과정에서 적용되는 CORDIC 정밀도(x 축)에 따라 발생하는 오차(y축)를 나타낸다. 정밀도가 증가함에 따라 오차(y축)가 줄어든다.

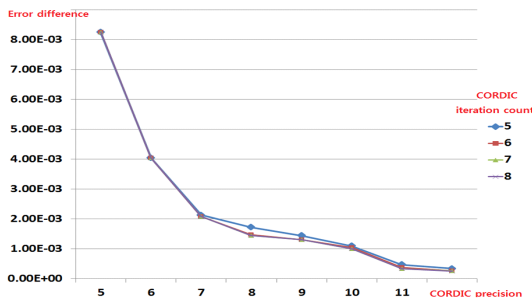


그림 12. 코르딕 반복 횟수와 매 반복마다 적용되는 정밀도에 따른 오차

Fig. 12. The error difference according to the precision which sets with CORDIC number of occurrence and precision

c. 하드웨어 합성

표 5는 Synopsys Design Compiler와 TSMC 0.18 um을 이용하여 [1]의 구조에서 최대 클럭 주파수인 200 MHz를 적용한 후 합성한 것을 나타낸다. [1]은 최소 자승법을 적용하여 좌표 변환 수행, 양 선형 보간법을 적용한 기존 구조[1],[14],[15],[16] 중에서 가장 하드웨어 자원이 적게 요구되는 구조이며 31 단계 파이프라인을 적용했다. 31 사이클이 지난 후에 첫 픽셀이 생성되며 200 MHz에서 동작하기 때문에 보정 이미지 2560x2560 해상도를 30 프레임/초 처리할 수 있다.

제안 구조는 하드웨어와 룩 업 테이블을 함께 사용한 구조와 하드웨어만 사용한 구조로 구분된다. 코르딕 반복 횟수는 6번, 반복 과정에서 요구되는 정밀도는 4 bit, 보간 계수 생성 과정에서 요구되는 정밀도는 4 bit로 설정한다. 표 6은 200MHz에서 하드웨어만 사용한 구조를 합성한 결과이다. 먼저, 하드웨어만 사용한 구조는 [1]의 구조에서 최대 클럭 주파수인 200 MHz보다 더 높은 220 MHz까지 동작하기 때문에 처리량이 증가한다. 그러나 기존 구조에서 추가되는 연산으로 인하여 [1]에 비해 전체 게이트 카운트가 7.15배 증가한다. 이를 해결하기 위한 하드웨어와 룩 업 테이블을 함께 사용한 구조는 해상도에 따라 룩 업 테이블 내부 데이터를 변경해줘야 하는 단점이 있으나 하드웨어 크기는 해상도에 비례하여 증가하는 룩 업 테이블의 크기에 의존하기 때문에 해상도가 낮아질수록 하드웨어만 사용한 구조와의 전체 게이트 카운트가 감소한다. 룩 업 테이블의 크기가 97.7 Kbyte 이하부터는 하드웨어만 사용한 구조보다 전체 게이트 카운트가 감소하며 모니터 38만(800x480) 화소를 이용하여 차량용 후방 카메라 32만(656x492) 화소를 보정할 경우에는 하드웨어만 사용한 구조보다 전체 게이트 카운트가 10% 작다.

표 5. 하드웨어 합성 결과
Table 5. The results of hardware synthesis

	(1)	하드웨어 + 룩 업 테이블	하드웨어
공정	0.18 um		
동작 주파수	200 MHz		
게이트 카운트	28,622	68,342	205,700
LUT _{Total} 크기	N/A	84 Kbyte	N/A
전체 게이트 카운트 (LUT _{Total} 포함)	28,622	185,508	205,700

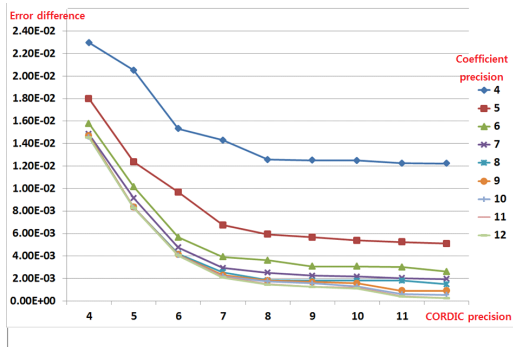


그림 13. CORDIC 정밀도와 보간 계수 정밀도의 관계
 Fig. 13. The relationship between CORDIC precision and the interpolation coefficient precision

표 6. 하드웨어만 사용한 구조의 합성 결과
 Table 6. The synthesis result of HW only design

	하드웨어만 사용한 구조
좌표 변환 (보정->어안)	9,011
좌표 변환 (어안->보정)	144,176
거리 계산	27,732
버블 소트 정렬	8,427
식(7), 식(8)	16,354
전체 게이트 카운트	205,700

V. 결론

본 논문은 보정 영상의 최 근접 좌표를 이용한 방사 왜곡 하드웨어 구조를 제안했다. 기존 대부분의 구조가 좌표 변환에 집중한 반면 제안 구조는 보간법에 초점을 맞추었다. 제안 구조에서 적용한 보정된 영상 위 최 근접 좌표를 이용한 보간법은 양 선형 보간법에 비해 외각 영역의 최대 신호 대 잡음 비가 0.917dB이 높기 때문에 기존보다 보정 이미지의 더 넓은 영역이 모니터로 출력된다. 또한, 하드웨어만 사용한 구조는 기존 구조보다 클록 주파수가 높기 때문에 처리량이 증가한다. 그러나 기존 구조에서 추가되는 연산으로 인하여 하드웨어 크기가 증가한다. 이를 해결하기 위한 하드웨어와 룩업 테이블을 함께 사용한 구조는 룩업 테이블의 크기가 97.7 Kbyte 이하부터는 하드웨어만 사용한 구조보다 하드웨어 크기가 감소하고, 자동차 후방 카메라의 경우 하드웨어 크기를 10% 줄일 수 있다. 차후 연구 내용은 다음과 같다. 좌표 변환과 보간법을 수행하는 block은 독립적으로 구현할 수 있기 때문에 좌표 변환 과정에 측량적 방법을 적용하여 보정 화질을 높이는 것과 룩업 테이블의 크기를 줄이기 위한 방법을 연구하는 것이다.

참고문헌

- [1] S. Chen, H. Huang, and C. Luo, "Time multiplexed VLSI architecture for realtime barrel distortion correction in video endoscopic images," *Circuits and Systems for Video Technology, IEEE Trans. on*, Vol 21, No 11, pp. 1612-1621, November 2011.
- [2] W. Feng, Y. Liu, "Omnidirectional vision tracking and positioning for vehicles," *Natural Computation, 2008. ICNC '08. Fourth International Vol 6*, pp. 183 -187, October 2008.
- [3] X. ming Dong, "Landmark design and realtime recognition based on fisheye image for robot navigation," *Computer Engineering and Technology (ICCET), 2010 2nd International Conference on*, Vol 1, pp. 437-440, April 2010.
- [4] J. Brauers, "Geometric calibration of lens and filter distortions for multispectral filter wheel cameras," *IEEE Trans on Image Processing*, Vol 20, No 2, pp. 496-505, February 2011.
- [5] T. Rahman, "An efficient camera calibration technique offering robustness and accuracy over a wide range of lens distortion," *IEEE Trans on Image Processing*, Vol 21, No 2, pp. 626-637, February 2012.
- [6] Zijian Zhao, "Camera calibration with three noncollinear points under special motions," *IEEE Transactions on Image Processing*, Vol 17, No 12, pp. 2393-2402, December 2008.
- [7] K. Y. Shin, "Multi camera calibration method for optical motion capture system," *Journal of the Korea society of computer and information*, Vol 14, No 6, pp. 41-49, June 2009.
- [8] M. Ahmed, "Nonmetric calibration of camera lens distortion: differential methods and robust estimation," *IEEE Transactions on Image Processing*, Vol 14, No 8, pp. 1215-1230, August 2005.
- [9] K. Johnson, P. W. Smith, and M. A. Abidi, "A quadric surface projection model for wide

- angle lenses," Proc. of SPIE Conf. on Intelligent Robots, and Computer Vision XVII: Algorithms, Techniques, and Active Vision, Vol. 3522, Boston, MA, November, 1999, pp. 424-434.
- [10] M. Sone, N. Sudo, H. Yokotsuka, "Comparison of image interpolation methods applied to least squares matching," Computational Intelligence for Modelling Control & Automation, 2008 International Conference on, pp. 1017-1022, December 2008.
- [11] S. M. Jung, "Quality improvement scheme of interpolated image using the characteristics of the adjacent pixels," Journal of the Korea society of computer and information, Vol 16, No 11, pp. 95-102, October 2011.
- [12] J. Y. Han, "Fisheye image correction and low power scaler design," master's thesis, Kwangwoon University, 2010.
- [13] Y. J. Lee, "Near pixels fisheye lens image correction based on near pixels of a corrected image," master's thesis, Kwangwoon University, 2011.
- [14] K. Vijayan Asari, "Design of an efficient VLSI architecture for non linear spatial warping of wide angle camera images," Journal of Systems Architecture, Vol. 50, No 12, pp. 743-755, August 2004.
- [15] H. T. Ngo and V. K. Asari, "A pipelined architecture for realtime correction of barrel distortion in wide angle camera images," IEEE Trans. Circuits Syst. Video Techno., Vol. 15, No. 3, pp. 436-444, March 2005.
- [16] P. Y. Chen, C. C. Huang, Y. H. Shiau, and Y.T. Chen, "A VLSI implementation of barrel distortion correction for wide angle camera images," IEEE Trans. Circuits Syst. 2 Express Briefs, Vol. 56, No. 1, pp. 51-55, January 2009.
- [17] Synopsys Inc, "DesignWare building block IP." [Online]. Available: <http://www.synopsys.com>
- [18] J. Volder, "The CORDIC trigonometric computing technique," Electronic Computers, IRE Transactions on, Vol EC-8, No 3, pp. 330-334, September 1959.
- [19] Kodak lossless true color image, <http://r0k.us/graphics/kodak>

저자 소개



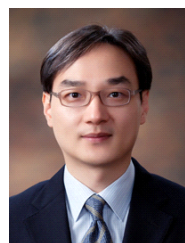
송 남 훈

2011 : 광운대학교 컴퓨터공학과
공학사

2011년 ~ 현재 :

광운대학교 컴퓨터공학과 석사과정
관심분야 : Computer Vision,
SoC 설계

Email : windpick@kw.ac.kr



이 준 환

1991 : 연세대학교 전자공학과
학사 졸업

1998 : Univ. of Michigan,
EECS 석사 졸업

2002 : Univ. of Michigan,
EECS 박사 졸업

1991 ~ 1995년 :

삼성전자 시스템LSI 연구원

2003 ~ 2008년 :

삼성전자 통신연구소 수석연구원

2008 ~ 현재 :

광운대학교 컴퓨터공학과 부교수

관심분야 : SoC 구조설계, 저전력
설계, Computer Vision,
반도체설계

Email : joonhwan.yi@kw.ac.kr