

논문 2012-50-6-15

광각 카메라를 위한 저 복잡도 실시간 배럴 왜곡 보정 프로세서의 설계 및 구현

(Design and Implementation of a Low-Complexity Real-Time Barrel Distortion Corrector for Wide-Angle Cameras)

정 희 성*, 김 원 태*, 이 광 호*, 김 태 환**

(Hui-Seong Jeong[Ⓒ], Won-Tae Kim, Gwang-Ho Lee, and Tae-Hwan Kim)

요 약

광각 카메라는 단 초점 렌즈를 장착하여 넓은 시야의 이미지를 처리하는데, 렌즈의 광학 문제로 인해 이미지에 배럴 왜곡 (barrel distortion)이 발생한다. 본 논문에서는 배럴 왜곡을 실시간 디지털 신호처리를 통해 보정하기 위한 낮은 복잡도의 프로세서 구조를 제시하고 이를 실제 구현하여 유효성을 검증하였다. 제안하는 왜곡 보정 프로세서는 하드웨어 복잡도를 낮추기 위해서, 좌표 위치 보정에 필요한 계산을 점증적(incremental)으로 수행한다. 또한, 높은 보정 속도를 달성하기 위해 파이프 라인 구조로 설계하였다. 설계된 보정 프로세서는 0.11 μ m complementary metal-oxide semiconductor(CMOS) 공정을 사용하여 14.3K의 논리 게이트로 구현되었다. 2048 \times 2048 픽셀 영상에 대하여, 최대 314MHz의 동작 주파수로 초당 74.86번의 속도로 보정이 가능하다.

Abstract

The barrel distortion makes serious problems in a wide-angle camera employing a lens of a short focal length. This paper presents a low-complexity hardware architecture for a real-time barrel distortion corrector and its implementation. In the proposed barrel distortion corrector, the conventional algorithm is modified so that the correction is performed incrementally, which results in the reduction of the number of required hardware modules for the distortion correction. The proposed barrel distortion corrector has a pipelined architecture so as to achieve a high-throughput correction. The correction rate is 74.86 frames per sec at the operating frequency of 314MHz in a 0.11 μ m CMOS process, where the frame size is 2048 \times 2048. The proposed barrel distortion corrector is implemented with 14.3K logic gates.

Keywords : barrel distortion, distortion correction, wide-angle lens, hardware implementation, digital signal processing,

I. 서 론

광각 카메라는 단 초점 광각 렌즈를 장착하여 표준

렌즈를 장착한 카메라에 비해 넓은 화각을 제공한다. 이러한 점 때문에, 내시경, 자동차 후방 탐지 센서, 무인 감시 시스템 등 다양한 분야에 응용되고 있다. 하지만 [1-3]에서와 같이 단 초점 렌즈의 물리적 비 이상성 (non-ideality)으로 인하여 촬영된 이미지에 방사상의 형태적 왜곡이 발생한다. 이러한 왜곡 중에서 그림 1에서 예시하는 배럴 왜곡(barrel distortion)은 이미지 상의 물체의 상대적인 크기가 외곽으로 갈수록 작아지는 현상으로, 전체 시스템의 품질에 심각한 영향을 미칠 수 있다. 예를 들어, 그림 1의 내시경으로 촬영된 이미지에서 왜곡이 발생해 중앙의 크기를 오판할 수 있다.

* 학생회원, ** 정회원, 한국항공대학교 항공전자및정보통신공학부

(School of electronics, Telecommunication and computer engineering, Korea Aerospace University)

※ 본 연구는 경기도의 경기도지역협력연구센터사업의 일환으로 수행하였음 [GRRC항공-2012-A02, 실감영상시스템을 위한 광학기기의 왜곡 보정 기술 연구]

Ⓒ Corresponding Author (E-mail: taehwan.kim@kau.ac.kr)

접수일자 2013년3월12일, 수정완료일 2013년5월15일

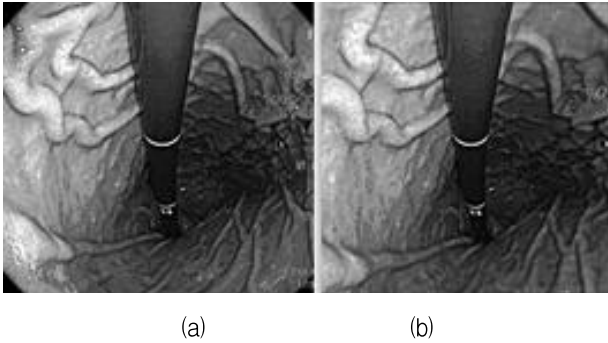


그림 1. 베럴 왜곡이 발생한 내시경 이미지의 예:
 (a) 왜곡된 이미지, (b) 원본 이미지
 Fig. 1. Examples of endoscopic image of the barrel distortion has occurred:
 (a): Distorted image, (b): Original image.

베럴 왜곡 현상을 해결하기 위해서는 물리적으로 이상적인 렌즈를 도입할 수 있겠지만, 이는 높은 제조비용을 요구하므로 현실적이지 못하다. 본 논문은 렌즈의 광학 왜곡에 의해 변형된 이미지를 신호 처리 과정을 통해 후 보정해주는 방식과 이에 대한 효율적인 하드웨어 구현에 대한 것이다.

베럴 왜곡을 위한 보정 처리 과정은 실시간 동작이 요구되며, 일반적으로 소형화된 시스템에 적용된다. 따라서 소프트웨어 형태로 구현하기 보다는 전용화 된 하드웨어로 구현하여 고속의 동작을 가능케 하는 것이 일반적이다^[3-9]. 또한, 소형화된 시스템에 적용하기 위해서는 낮은 복잡도를 갖는 저 면적의 구현이 필요하다. 기존 연구 중, [4]에서는 맵핑 다항식의 근사화를 통해 삼각 함수와 같은 복잡한 연산을 효과적으로 제거하여 낮은 복잡도의 보정 프로세서를 구현하였다. [5]에서는, 시간 다중화 방식의 하드웨어 모듈 공유 기법을 통해 고해상도가 요구되지 않는 응용 시스템을 위한 낮은 복잡도의 보정 프로세서를 제안하였다.

본 논문에서는 기존 보정 알고리즘을 하드웨어적으로 효율성을 고려하여 변형하고, 이에 기반한 고속의 보정 속도 달성을 위해 파이프라인 구조의 베럴 왜곡 보정 프로세서를 구현하였다. 제안하는 구조는 좌표 위치를 순차적으로 보정함에 있어, 기존 픽셀의 보정 값을 바탕으로, 현재의 보정 값을 점증적으로 계산한다. 이와 같은 변형된 알고리즘을 이용하여, 하드웨어 복잡도를 크게 낮추면서도, 고속의 왜곡 보정 품질을 달성할 수 있었다.

본 논문의 II장에서는 베럴 왜곡 보정 기술을 설명한다. III장에서는 제안하는 점증적 보정 기법과 보정 프로세서의 구조에 대하여 설명한다. IV장에서는 제안하

는 보정 프로세서의 구현 및 검증 결과를 고찰하고, 기존 연구 결과와 비교한다. 마지막으로, V장에서 결론으로 본 논문을 마치고자 한다.

II. 베럴 왜곡 보정 기술

본 장에서는 역 맵핑(backward mapping)에 기초한 왜곡 보정 기술을 소개한다^[10]. 왜곡 보정 과정에서 왜곡된 이미지 공간을 distorted image space(DIS), 보정된 이미지 공간을 corrected image space(CIS)라 하자. 역 맵핑은 CIS 상의 개별 픽셀의 좌표에, 대응되는 DIS 상의 좌표를 맵핑시키는 과정이다. 역 맵핑을 통해, CIS의 각 픽셀에 대응되는 DIS의 픽셀 위치를 구한 후, 해당 위치의 DIS의 픽셀 값을 바탕으로 CIS의 픽셀 값을 얻게 됨으로써, 주어진 DIS를 통해 CIS를 재구성(reconstruction)하여 왜곡 보정을 수행한다.

그림 2는 역 맵핑을 통해 CIS 상의 (u, v) 에 위치하는 P 에 대응되는 DIS 상의 P' 의 위치 (u', v') 을 구하는 과정을 설명하기 위한 것이다. 그림에서, (u_c, v_c) 는 각각 DIS와 CIS의 중심 위치를 나타낸다. 그림에서 r, r', θ, θ' 은 P 와 P' 의 위치를 극 좌표계에서 표현한 것으로 다음과 같이 기술 할 수 있다.

$$r = \sqrt{(u - u_c)^2 + (v - v_c)^2} \tag{1}$$

$$\theta = \tan^{-1}((v - v_c)/(u - u_c)) \tag{2}$$

$$r' = \sqrt{(u' - u'_c)^2 + (v' - v'_c)^2} \tag{3}$$

$$\theta' = \tan^{-1}((v' - v'_c)/(u' - u'_c)) \tag{4}$$

r 과 r' 의 관계는 일반적으로 다음과 같은 맵핑 다항

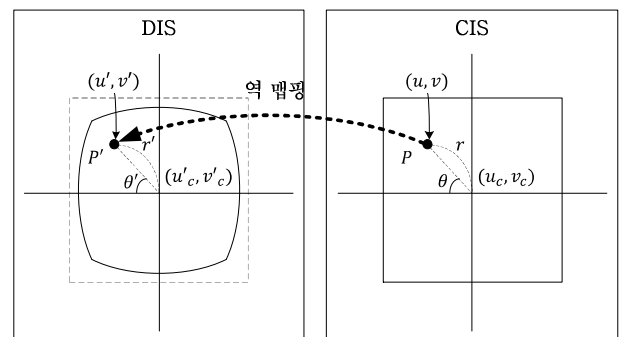


그림 2. 역 맵핑
 Fig. 2. Backward mapping.

$$\begin{aligned}
 & CIS(u, v) \\
 &= DIS(\lfloor u' \rfloor, \lfloor v' \rfloor)(1 - u' + \lfloor u' \rfloor)(1 - v' + \lfloor v' \rfloor) \\
 &+ DIS(1 + \lfloor u' \rfloor, \lfloor v' \rfloor)(u' - \lfloor u' \rfloor)(1 - v' + \lfloor v' \rfloor) \\
 &+ DIS(\lfloor u' \rfloor, 1 + \lfloor v' \rfloor)(1 - u' + \lfloor u' \rfloor)(v' - \lfloor v' \rfloor) \\
 &+ DIS(1 + \lfloor u' \rfloor, 1 + \lfloor v' \rfloor)(u' - \lfloor u' \rfloor)(v' - \lfloor v' \rfloor)
 \end{aligned} \tag{11}$$

식을 통해 모델링 된다.

$$r' = \sum_{n=1}^N c_n r^{2n-1} \tag{5}$$

여기에서, c_n 은 역 맵핑 계수이며, 최소 자승법을 통해 추정될 수 있다^[10]. 역 맵핑 과정에서 주어진 P 에 대응되는 P 의 위치는 다음과 같이 구한다.

$$u' = u_c' + r' \cos \theta' \tag{6}$$

$$v' = v_c' + r' \sin \theta' \tag{7}$$

본 논문에서 고려하고 있는 베럴 왜곡은 방사상으로 동일한 정도를 갖게 되므로, θ 와 θ' 은 같은 값이며, 이를 이용하여 위의 식은 다음과 같이 계산될 수 있다.

$$\begin{aligned}
 u' &= u_c' + r' \cos \theta = u_c' + r' \left(\frac{u - u_c}{r} \right) \\
 &= u_c' + s \cdot (u - u_c)
 \end{aligned} \tag{8}$$

$$\begin{aligned}
 v' &= v_c' + r' \sin \theta = v_c' + r' \left(\frac{v - v_c}{r} \right) \\
 &= v_c' + s \cdot (v - v_c)
 \end{aligned} \tag{9}$$

여기에서, s 는 (5)에 의해 다음과 같이 정의되는 scaling factor 라 볼 수 있다.

$$s \cong r'/r = \left(\sum_{n=1}^N c_n r^{2n-1} \right) / r = \sum_{n=1}^N c_n r^{2n-2} \tag{10}$$

일반적으로, scaling factor의 계산 과정에서 N 의 크기가 클수록 정밀한 맵핑이 가능해서 고 해상도의 이미지 보정이 가능하다. 본 논문에서는 $N = 3$ 의 맵핑 다항식을 기준으로 2048×2048 크기의 이미지를 보정하였다.

P 을 역 맵핑한 P 의 위치는 (8), (9)에 따라 구할 수 있고, DIS 상의 P 의 픽셀값을 CIS 상의 P 의 픽셀 값으로 사용하여 CIS를 재구성하여 왜곡 보정을 수행하게 된다. DIS 상의 (a, b) 에 위치한 픽셀 값을 $DIS(a, b)$ 으로 표현하자. 그러면, P 의 픽셀 값은 $DIS(u', v')$ 로 표현할 수 있는데, u' 및 v' 은 (8), (9)에 따라 계산되므로 유리수가 된다. 따라서 $DIS(u', v')$ 를 구하기 위해서

는 DIS에 대한 리샘플링(resampling)이 필요하고, 이를 위해 다음과 같이 이중 선형 보간(bilinear interpolation)이 사용될 수 있다^[6].

(11)에서, $\lfloor a \rfloor$ 은 a 보다 크지 않은 최대의 정수를 의미한다. 이와 같이 이중 선형 보간을 통해 구해진 $DIS(u', v')$ 을 CIS 상의 (u, v) 에 위치하는 픽셀 값인 $CIS(u, v)$ 로 사용하여 CIS를 재구성한다. 이와 같은 역 맵핑 과정과 이중 선형 보간을 통해 CIS 상의 모든 픽셀에 대한 DIS 상의 픽셀 값을 순차적으로 구하여, 주어진 DIS를 바탕으로 CIS를 복원한다.

III. 제안하는 왜곡 보정 프로세서

1. 점증적 왜곡 보정 기법

제안하는 베럴 왜곡 보정은 그림 3에 도시한 래스터 스캔 순서에 따라 이루어진다^[11]. 래스터 스캔 순서에 따르면, 현재 보정되는 픽셀의 CIS 상의 위치는 1) 바로 이전에 보정된 픽셀의 오른쪽 또는 2) 그 다음 줄의 첫 번째에 위치하게 된다. 제안하는 점증적 보정 기법은 이러한 래스터 스캔 순서에 따른 보정 픽셀의 상대적인 위치 관계를 이용한 것으로, 앞의 1), 2)의 경우 모두에 적용되지만, 여기에서는 설명의 편의성을 위해 $n+1$ 번째로 보정되는 픽셀의 위치가 n 번째로 보정되는 픽셀의 바로 오른쪽에 위치하는 경우로 한정 한다 ($0 \leq$

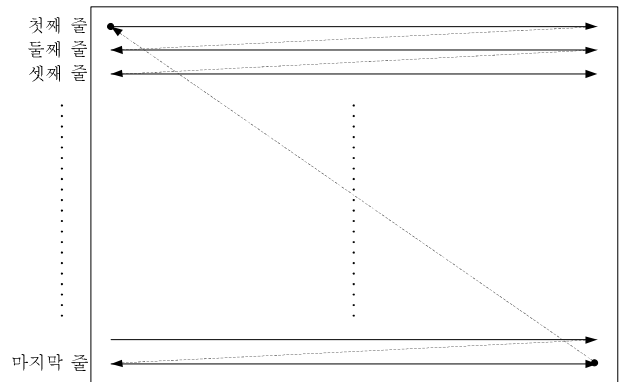


그림 3. 래스터 스캔 방식에 기반하는 왜곡 보정의 순서
Fig. 3. Order of the distortion correction according to the raster scan.

$n < 2048 \times 2048$). 제안하는 보정 프로세서에서 n 번째와 $n+1$ 번째로 보정되는 픽셀의 CIS 상의 위치를 각각 $(u_n, v_n), (u_{n+1}, v_{n+1})$ 이라 하면, (u_{n+1}, v_{n+1}) 은 (u_n+1, v_n) 이 된다. $n+1$ 번째 픽셀에 대하여 역 맵핑을 수행하기 위해서는, 먼저 해당 픽셀과 중심 간의 거리의 제곱인 r_{n+1}^2 을 구해야 하는데, 이는 바로 이전의 픽셀에 대한 계산 값인 r_n^2 을 사용하여 다음과 같이 점증적으로 계산할 수 있다.

$$\begin{aligned} r_{n+1}^2 &= (u_{n+1} - u_c)^2 + (v_{n+1} - v_c)^2 \\ &= (u_n - u_c + 1)^2 + (v_n - v_c)^2 \\ &= r_n^2 + \{u_n - u_c, 1\} \end{aligned} \tag{12}$$

위의 식에서, $\{a, b\}$ 는 a, b 를 비트 순서대로 연결하는 연산을 나타낸다. 기존의 방법에서는 r_{n+1}^2 을 매번 새로 계산해야 하기 때문에, 이를 위해 매 픽셀의 보정마다 두 번의 제곱 연산이 필요하다^[4]. 하지만 제안하는 기법에서는 r_0^2 이 외부에서 주어지도록 하고, 매 픽셀마다의 연산을 점증적으로 수행하도록 하여, 제곱 연산을 효과적으로 제거할 수 있다.

역 맵핑 과정을 위해서는 scaling factor의 계산이 필요하다. $n+1$ 번째 보정 픽셀에 대한 scaling factor를 s_{n+1} 이라 하면 이는 (10)에 따라 다음과 같이 계산된다.

$$s_{n+1} = c_1 + c_2 r_{n+1}^2 + c_3 r_{n+1}^4 \tag{13}$$

제안하는 방법에서는 s_{n+1} 역시 기존 픽셀의 scaling factor인 s_n 을 이용하여 다음과 같이 점증적으로 계산한다.

$$\begin{aligned} s_{n+1} &= s_n + \{u_n - u_c, 1\} \\ &\quad \cdot (c_2 + c_3 (r_n^2 \ll 1 + \{u_n - u_c, 1\})) \end{aligned} \tag{14}$$

위의 식에서 $a \ll b$ 는 a 를 b 비트만큼 좌측으로 이동하는 것을 의미한다. (13)과 같이 기존의 방법에 따라 scaling factor를 계산할 경우, 매 픽셀의 보정마다 네 제곱 및 제곱의 복잡한 계산이 필요한 반면에^[4], (14)와 같이 변형된 제안된 방식에서는, 초기 값 s_0 을 외부에서 입력 받아서 점증적으로 scaling factor를 계산하게 되어 두 번의 곱셈만이 필요하다.

2. 제안하는 왜곡 보정 프로세서의 구조

앞 절에서 제안한 점증적인 기법을 기반으로 고속의 왜곡 보정이 가능한 왜곡 보정 프로세서의 구조를 설계하였다. 제안하는 왜곡 보정 프로세서는 역 맵핑을 수행하는 부분과 이중 선형 보간을 수행하는 부분으로 구

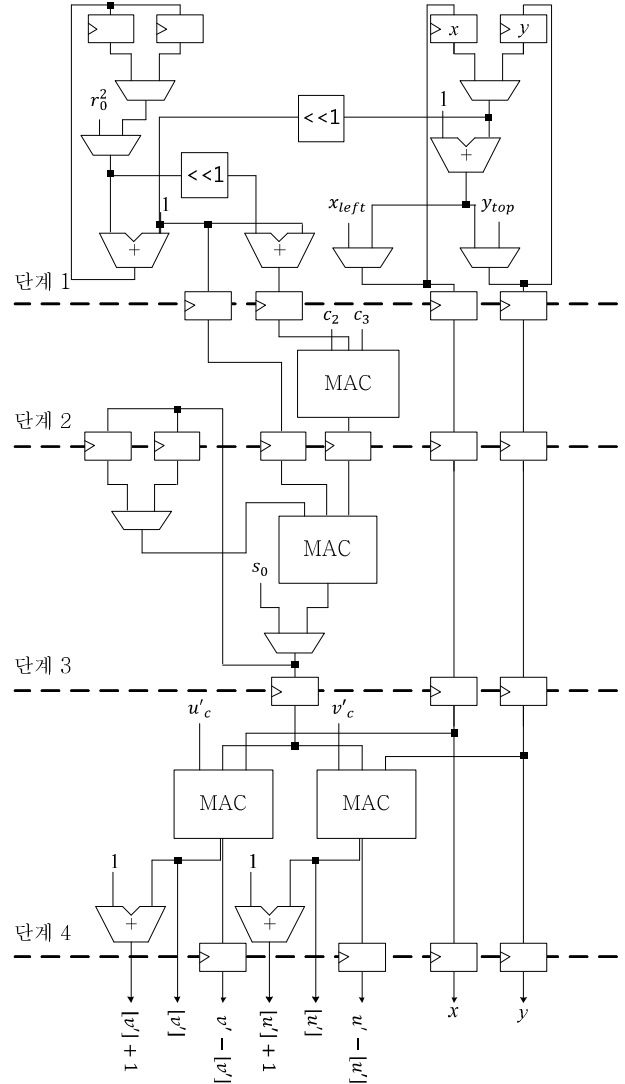


그림 4. 역 맵핑 파이프라인 구조
Fig. 4. Pipeline architecture for the backward mapping.

성되며, 고속의 보정 속도를 달성하기 위해서 해당 부분을 각각 4단계와 3단계로 구성된 파이프라인으로 설계하였다.

그림 4는 제안하는 보정 프로세서의 역 맵핑을 수행하는 부분의 데이터패스의 구조를 도시한 것이다. 구현의 편의를 위하여 $u_n - u_c, v_n - v_c$ 를 각각 x 와 y 로 설정하였다. 단계 1에서는 x 또는 y 의 좌표를 한 픽셀씩 이동시키는 동시에 r_n^2 을 계산하고, r_n^2 은 r_{n+1}^2 의 계산을 위해 재사용 된다. 점증적 계산을 위해서, x_{left} 와 r_0^2 , 그리고 s_0 는 프레임에서 한 줄이 바뀔 때마다 초기 값으로 입력되고, y_{top} 은 한 프레임이 바뀔 때 마다 초기 값으로 입력되도록 제어된다. 단계 2와 단계 3에서는 scaling factor의 계산을 위한 역 맵핑 계수 c_2, c_3 를 입력 받고, 그 값을 이용하여 scaling factor를 계산한다.

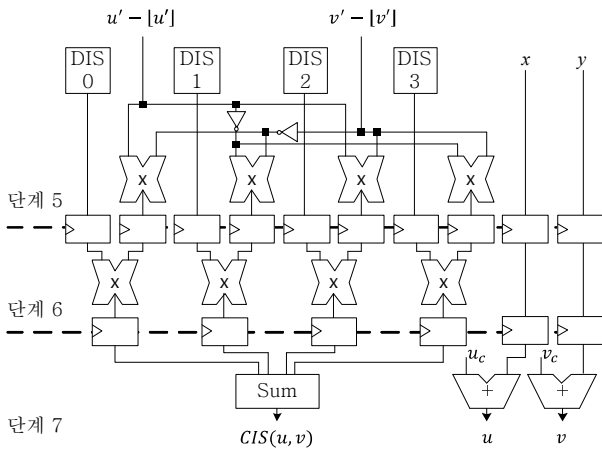


그림 5. 이중 선형 보간 파이프 라인 구조
Fig. 5. Pipeline architecture for the bilinear interpolation.

단계 4에서는 scaling factor를 이용하여 역 맵핑된 좌표의 위치를 계산하고, 이중 선형 보간을 수행하기 위한 변수들을 계산한다. 그림에서 multiply and accumulate(MAC)은 3개의 입력 값 중 오른쪽 두 개의 입력 값을 곱하고 나머지 하나를 더하는 동작을 수행하는 모듈이다. 내부적으로 곱셈의 부분 결과물을 나머지 하나의 입력 값에 더하는 다중 입력 가산기를 구현함에 있어 Wallace tree 기반의 구조를 사용함으로써 낮은 복잡도와 높은 동작속도를 달성할 수 있도록 했다^[12].

그림 5는 제안하는 보정 프로세서에서 이중 선형 보간을 수행하는 부분의 하드웨어 구조를 도시한 것이다. 그림 4에서의 출력인 $[u']$ 와 $[v']$ 및 $u - [u']$ 과 $v - [v']$ 을 이용하여 왜곡된 이미지 픽셀의 좌표에 인접해 있는 4개의 정수 좌표의 픽셀 값을 DIS 버퍼로부터 읽어서, 이를 사용하여 (11)에 따라 이중 선형 보간을 수행하여 픽셀 값을 구한 후, CIS 버퍼에 보정된 픽셀 값 $CIS(u, v)$ 을 쓰게 된다.

IV. 고 찰

본 장에서는 III장에서 제안한 점증적 왜곡 보정 방법과 이에 기반하는 보정 프로세서의 하드웨어 복잡도와 성능에 대해 고찰하고, 실제 구현 결과를 바탕으로 기존의 연구 결과와 비교하여 본 연구의 우수성을 보인다.

1. 하드웨어 복잡도

제안한 왜곡 보정 방법을 래스터 스캔 순서에 따른 픽셀간의 상대적인 위치 관계를 이용하여, 역 맵핑과

표 1. 기존 왜곡 보정 방법에 기반하는 보정 프로세서의 주요 하드웨어 모듈 수의 비교

Table 1. Comparison of the number of hardware modules.

주요 하드웨어 모듈	본 논문	[4]	[5]
24 비트 × 24 비트 곱셈기	0	7	4
24 비트 × 16 비트 곱셈기	0	2	2
19 비트 × 12 비트 곱셈기	1	0	0
16 비트 × 16 비트 곱셈기	0	2	2
12 비트 × 12 비트 곱셈기	4	0	0
12 비트 × 11 비트 곱셈기	2	0	0
11 비트 × 7 비트 곱셈기	1	0	0
8 비트 × 8 비트 곱셈기	0	3	8
8 비트 × 7 비트 곱셈기	4	0	0

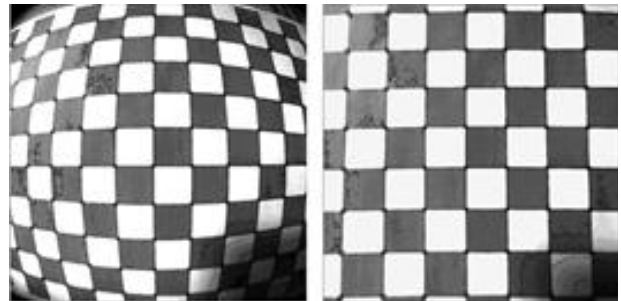


그림 6. 제안된 보정 프로세서를 통한 왜곡 보정 결과 예시:
(a) 왜곡된 이미지, (b) 보정한 이미지

Fig. 6. Result of distortion correction by the proposed correction processor:
(a) Distorted image, (b) Corrected image.

정을 점증적으로 수행함으로써 계산 량을 크게 줄일 수 있다. 이는 보정 프로세서를 하드웨어로 구현할 경우, 보정 과정을 수행하기 위해 필요한 하드웨어 모듈의 수의 감소로 귀결된다. 표 1은 기존의 보정 프로세서 [4][5]에서와 점증적 왜곡 보정 방법에 기반하는 제안된 보정 프로세서에서의 주요 하드웨어 모듈의 수를 비교한 것이다. 표에서 보는 바와 같이 제안된 보정 프로세서를 구현함에 있어 기존 연구결과와 비교하여 곱셈기의 수를 크게 줄일 수 있었다.

2. 구현 및 검증 결과

제안된 보정 프로세서는 Verilog HDL을 사용하여 합성 가능한 소프트웨어 형태로 설계하였다. 설계된 프로세서의 동작 검증을 위하여 소프트웨어를 통한 보정 결과와 하드웨어를 통한 보정 결과에 대하여 개별 픽셀 값을 비교하여 동일한 이미지임을 확인하였다. 그림 6은 본 논문에서 구현한 보정 프로세서를 이용하여 이미

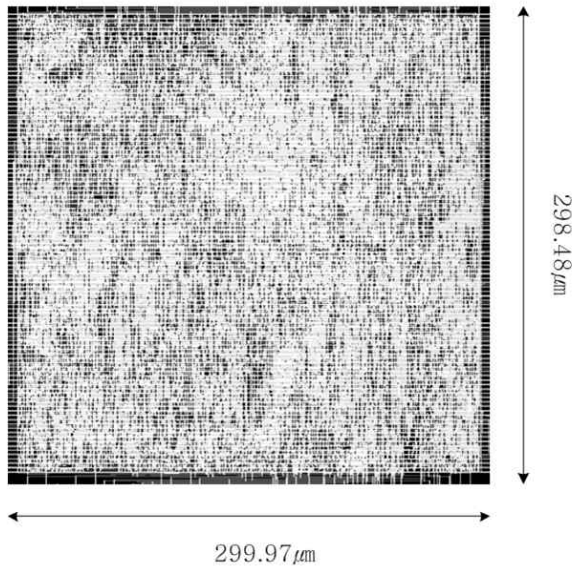


그림 7. 제안된 보정 프로세서의 레이아웃
Fig. 7. Layout of the proposed correction processor.

표 2. 왜곡 보정 프로세서의 구현 결과 비교
Table 2. Comparison of implementation results.

	본 논문	[4]	[5]
동작 주파수	314MHz	200MHz	200MHz
보정 이미지 크기	2048 × 2048	1024 × 1024	2560 × 2048
공정	0.11 μm CMOS	0.18 μm CMOS	0.18 μm CMOS
게이트 수	14,302	44,992	28,622
보정 속도	314M픽셀/초	140M픽셀/초	160M픽셀/초

지의 왜곡을 보정한 결과의 예시이다.

제안하는 보정 프로세서는 0.11 μm CMOS 공정의 셀 라이브러리를 사용하여 합성하였다. 그림 7은 합성된 결과물을 자동 배치한 레이아웃 결과이며, 전체 보정 프로세서 코어의 크기는 299.97 μm × 298.48 μm이다.

제안된 보정 프로세서의 구현 결과 및 기존 논문의 결과를 표 2에 정리하였다. 왜곡 보정 프로세서의 왜곡 보정 품질은 식 (5)의 역 맵핑 계수에 크게 영향을 받는다. 본 논문과 마찬가지로 표에 포함된 기존의 연구 결과물 [4], [5] 역시 [10]에서 제안하는 최소자승법을 통해 역 맵핑 계수를 추정하였으므로, 표에서 비교한 보정 프로세서들은 모두 동일한 왜곡 보정 품질을 갖는다. 제안하는 프로세서는 314MHz의 동작 주파수로 동작하며, 파이프 라인 구조에 의해 매 동작 주파수 마다 영상의 한 픽셀이 보정된다. 따라서 초당 314M 픽

셀의 배럴 왜곡을 보정할 수 있다. 총 게이트 수는 14,302이며 이는 가장 작은 2-입력 NAND 게이트를 1로 환산한 것이다. 제안된 보정 프로세서는 기존의 연구 결과 [4], [5]와 비교하여 각각 224%, 196% 수준의 높은 보정 속도를 갖는다. 이러한 상대적으로 높은 보정 속도를 달성하면서도, 총 게이트 수는 기존 연구 결과 [4], [5] 대비 31.79%, 49.97% 수준으로 매우 낮은 복잡도를 갖는다.

V. 결론

본 논문에서는 낮은 연산 복잡도를 갖는 배럴 왜곡 보정 방법을 제안하고, 이를 기반으로 고속의 왜곡 보정 프로세서를 설계하고 구현하였다. 제안된 방법은 래스터 스캔 순서에 따른 보정 픽셀간의 상대적 위치 관계를 이용하여 보정 계산을 점증적으로 수행하도록 하여 낮은 하드웨어 복잡도를 갖는 보정 프로세서의 구현이 가능하였으며, 고속의 보정 속도를 달성하기 위해 파이프 라인으로 설계하였다. 이에 따라 314MHz의 동작 주파수로 2048×2048의 크기의 영상을 초당 74.86번 보정 가능하다. 또한, 제안하는 보정 프로세서는 기존의 연구결과와 동일한 왜곡 보정 능력을 지니면서도 196% 수준의 보정 속도를 보이며, 49.97% 수준의 적은 논리 게이트로 구현되었다.

REFERENCES

- [1] 김선영, 윤인혜, 김동균, 백준기, “나선형 패턴을 사용한 어안렌즈 영상 교정 및 기하학적 왜곡 보정,” *전자공학회논문지*, 제49권 SP편, 제4호, 16-22쪽, 2012년 7월
- [2] 반 토안 카오, 조상복, “A new method using geometric invariability for lens distortion correction,” *전자공학회논문지*, 제48권 SP편, 제6호, 115-123쪽, 2011년 11월
- [3] L. Qiang, and N. Allison, “FPGA implementation of pipelined architecture for optical imaging correction,” *in Proc. IEEE Workshop on Signal Processing Systems (SIPS)*, pp. 182-187, Banff, Alta, Oct. 2006.
- [4] 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 Transactions on Circuits and Systems II Express Briefs*, vol. 56, no. 1, pp. 51-55, Jan. 2009.

- [5] S. Chen, H. Huang, and C. Luo, "Time multiplexed VLSI architecture for real-time barrel distortion correction in video-endoscopic images," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 11, pp. 1612-1621, November 2011.
- [6] H. T. Ngo and V. K. Asari, "A pipelined architecture for real-time correction of barrel distortion in wide-angle camera images," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 3, pp. 436-444, Mar. 2005.
- [7] H. Blasinski, W. Hai, and F. Lohier, "FPGA architecture for real-time barrel distortion correction of colour images," in *Proc. IEEE International Conference Multimedia and Expo (ICME)*, pp. 1-6, Barcelona, Spain, July. 2011.
- [8] H. Ngo, and V. Asari, "Developing a FPGA-based high performance, power-aware architecture for real-time radial lens distortion correction of video stream," *ICGST International J. on Programmable Devices, Circuits and Systems, PDCS*, vol. 7, pp. 33-41, May. 2007.
- [9] H. Zheng, and J. Li, "Real-time correction of distortion image based on FPGA," in *Proc. IEEE International Conference on Intelligent Computing and Integrated Systems (ICISS)*, pp. 167-170, Guilin, Oct. 2010.
- [10] V. K. Asari, S. Kumar, and D. Radhakrishnan, "A new approach for nonlinear distortion correction in endoscopic images based on least squares estimation," *IEEE Transactions on Medical Image*, vol. 18, no. 4, pp. 345-354, Apr. 1999.
- [11] Hughes, L. E. C., "Telecommunications XX-IV: The Raster," *Electrician 116*, pp. 351-352, Mar. 1936.
- [12] C. S. Wallace, "A suggestion for a fast multiplier," *IEEE Transactions on Electronic Computers*, vol. EC-13, no. 1, pp. 14-017, Feb. 1964.

— 저 자 소 개 —



정 희 성(학생회원)
2013년 한국항공대학교 항공전자 및 정보통신공학부 학사 졸업.
2013년 현재 한국항공대학교 항공전자 및 정보통신공학부 석사과정 재학중.

<주관심분야 : 회로 및 시스템, VLSI 설계, ASIC 설계>



김 원 태(학생회원)
2013년 현재 한국항공대학교 항공전자 및 정보통신공학부 학사과정 재학중.
<주관심분야 : 회로 및 시스템, VLSI 설계, ASIC 설계>



이 광 호(학생회원)
2013년 현재 한국항공대학교 항공전자 및 정보통신공학부 학사과정 재학 중.
<주관심분야 : 회로 및 시스템, VLSI 설계, ASIC 설계>



김 태 환(정회원)-교신저자
2005년 연세대학교 전기전자공학과 학사 졸업.
2007년 한국과학기술원 전기 및 전자공학과 석사 졸업.
2010년 한국과학기술원 전기 및 전자공학과 박사 졸업.

2013년 현재 한국항공대학교 항공전자 및 정보통신공학부 교수.
<주관심분야 : 회로 및 시스템, VLSI 설계, ASIC 설계>