

논문 2015-52-9-6

광각 카메라를 위한 저 복잡도 비네팅 및 배럴 왜곡 보정 프로세서

(A Low-Complexity Processor for Joint Vignetting and Barrel distortion Correction for Wide-Angle Cameras)

문 선 아*, 홍 진 우*, 김 원 태**, 김 태 환***

(Sun-A Moon, Jin-U Hong, Won-Tae Kim, and Tae-Hwan Kim[©])

요 약

본 논문에서는 광각 카메라에서 발생하는 비네팅 왜곡과 배럴 왜곡을 효율적으로 보정하기 위한 낮은 복잡도의 프로세서를 제안하고, 이를 구현한 결과를 보인다. 제안하는 프로세서에서는 비네팅 왜곡과 배럴 왜곡 보정 시 복잡한 연산을 수반하는 고차 다항식과 같은 피팅 함수를 구간 선형 근사하여 보정 품질을 유지하면서도 연산 복잡도를 크게 낮추었다. 이를 기반으로, 배럴 왜곡과 비네팅 왜곡을 중첩적으로 보정하도록 설계하여 전체적인 하드웨어 복잡도를 낮추었다. 제안하는 프로세서는 0.11 μ m CMOS 공정을 사용하여 18.6K의 논리 게이트로 구현되었으며, 2048 \times 2048 크기의 영상에 대하여 최대 200Mpixels/s의 속도로 보정이 가능하다.

Abstract

This paper proposes a low-complexity processor to correct vignetting and barrel distortion for wide-angle cameras. The proposed processor calculates the required correcting factors by employing the piecewise linear approximation so that the hardware complexity can be reduced significantly while maintaining correction quality. In addition, the processor is designed to correct the two distortions concurrently in a singular pipeline, which reduces the overall complexity. The proposed processor is implemented with 18.6K logic gates in a 0.11 μ m CMOS process and shows the maximum correction speed of 200Mpixels/s for correcting an image of which size is 2048 \times 2048.

Keywords : vignetting distortion correction, barrel distortion correction, wide-angle cameras,
piecewise linear approximation, digital circuits

* 학생회원, *** 정회원, 한국항공대학교 항공전자정보공학부

(School of Electronics and Information Engineering, Korea Aerospace University)

** 학생회원, 한국항공대학교 항공전자정보공학과 (Department of Electronics and Information Engineering, Korea Aerospace University)

© Corresponding Author(E-mail: taehwan.kim.kr@gmail.com)

※ 본 연구는 경기도지역협력연구센터(GRRC) 사업 [GRRC 항공-2015-B04]의 일환으로 수행하였음.

Received ; May 19, 2015 Revised ; July 31, 2015

Accepted ; September 3, 2015

I. 서 론

광각 카메라는 초점 거리가 짧고 넓은 화각을 갖는 광각 렌즈를 장착한 카메라로, 넓은 시야의 이미지를 촬영할 수 있다. 이와 같은 장점으로, 차량용 이미지 시스템, 이미지 감시 시스템, 내시경 등의 다양한 응용 분야에 탑재되고 있다. 하지만, 광각 카메라를 이용해 촬영한 이미지에서는 렌즈 시스템의 구조와 물리적 비이

상성으로 인해 비네팅 왜곡(vignetting distortion)과 배럴 왜곡(barrel distortion)이 관찰된다. 비네팅 왜곡이란 이미지의 중심에서 가장자리로 갈수록 도달하는 빛의 양이 감소하여 이미지의 밝기가 어두워지는 현상이며, 배럴 왜곡이란 렌즈의 중심과 가장 자리의 두께 차로 인해 렌즈의 배율이 비선형적으로 변화하여 이미지가 휘어져 보이는 현상이다. 그림 1은 광각 카메라를 이용해 획득한 이미지로, 비네팅 왜곡과 배럴 왜곡을 관찰할 수 있다. 이러한 비네팅 왜곡 및 배럴 왜곡은 이미지 상의 물체를 정확히 식별하는 데에 어려움을 유발하여 전체 시스템에 치명적인 결함이 발생할 수 있으므로, 왜곡을 효과적으로 보정할 필요가 있다.

비네팅 왜곡 및 배럴 왜곡 보정은 이미지 내의 모든 픽셀을 대상으로 이루어져야 한다. 또한, 앞에서 예시한 광각 카메라의 응용 분야를 고려할 때, 보정은 소형화된 시스템에서 실시간으로 수행되어야 함을 알 수 있다. 따라서, 범용의 프로세서에서 소프트웨어로 구현하기 보다는 전용의 하드웨어로 구현하는 것이 바람직하다. 이 경우, 높은 보정 속도를 달성하면서도 복잡도를 낮추기 위한 효율적인 보정 방법과 이를 위한 하드웨어 구조가 필요하다.

기존의 몇몇 연구에서 비네팅 왜곡 및 배럴 왜곡에 대한 보정 방법과 이를 위한 하드웨어 구조를 제시하였다. 비네팅 왜곡 보정(vignetting distortion correction, VDC)을 위한 방법으로, 이미지 내의 모든 픽셀의 상대적인 밝기 강도를 바탕으로 각 픽셀에 대응하는 shading factor를 측정하고 이를 look-up table(LUT)과 같은 메모리에 저장하는 방식이 [1]에서 제안되었다. VDC를 위한 다른 방법으로, shading factor를 영구적인 메모리에 저장하지 않고 측정된 shading factor를 바탕



그림 1. 비네팅 왜곡 및 배럴 왜곡 보정의 예: (a) 원본 이미지, (b) 비네팅 왜곡 및 배럴 왜곡이 보정된 이미지.

Fig. 1. An example of vignetting distortion correction and barrel distortion correction: (a) original image and (b) correction result.

으로 피팅 함수를 구하여 이를 계산하는 보정하는 방식이 [2~4]에서 제안되었다. 한편, 배럴 왜곡 보정(barrel distortion correction, BDC)을 위한 방법으로, 역 맵핑 과정에 기반을 두는 방법이 [5]에서 제안되었고, 역 맵핑 다항식을 근사하여 역 맵핑 과정에 수반되는 좌표 변환 과정을 제거함으로써 하드웨어 복잡도를 낮추는 방법이 [6]에서 제안되었다. 이러한 보정 방법들은 공통적으로 고차 다항식과 같은 복잡한 피팅 함수의 계산을 수반하기 때문에 높은 하드웨어 복잡도를 갖게 된다. 또한, 실제 카메라 시스템에서는 VDC와 BDC가 모두 수행되어야 하지만 기존의 연구들에서는 각각의 왜곡 보정을 개별적으로 다루었기 때문에^[2-10] 실제 시스템에 효율적으로 적용하기 어렵다.

본 논문에서는 VDC와 BDC를 중첩적으로 수행하는 낮은 복잡도의 프로세서 구조를 제안하고, 이를 구현한 결과를 보인다. 제안하는 구조에서는 VDC와 BDC에서 필요한 shading factor와 scaling factor를 구하기 위하여 피팅 함수를 직접 계산하지 않고, 피팅 함수를 구간 선형 근사하여 계산함으로써 보정 품질을 유지하면서도 연산 복잡도를 크게 낮추었다. 또한, VDC와 BDC를 단일 파이프라인 구조 상에서 중첩적으로 처리하도록 설계하여 높은 보정 속도를 달성하면서도 전체 시스템의 복잡도를 낮출 수 있도록 하였다. 제안하는 구조를 바탕으로 VDC와 BDC를 위한 프로세서를 18.6K의 게이트로 구현하였으며, 구현된 프로세서는 2048×2048 크기의 이미지에 대해서 200Mpixels/s의 보정 속도를 갖는다.

본 논문의 나머지 부분은 다음과 같이 구성된다. II장에서는 비네팅 왜곡과 배럴 왜곡의 보정 방법을 설명한다. III장에서는 비네팅 왜곡과 배럴 왜곡에 대한 낮은 복잡도의 보정 방법을 제안하고 이를 위한 효율적인 하드웨어 구조에 대해 설명한다. IV장에서는 제안하는 프로세서의 구현 결과를 제시하고, 기존 연구 결과와 비교하고 고찰한다. 마지막으로, V장에서는 본 논문의 결론을 제시한다.

II. VDC 및 BDC

1. 비네팅 왜곡 보정 (VDC)

VDC는 이미지의 상대적인 밝기 강도를 균일하게 하는 과정으로, 이미지 내의 각 픽셀에 대응하는 shading

factor를 구하고 이를 픽셀 값에 곱하여 다음과 같이 수행된다.

$$I'([u, v]) \leftarrow s_V([u, v]) \cdot I([u, v]) \quad (1)$$

(1)에서 $I(\cdot)$ 는 왜곡 보정 전의 픽셀 값을, $I'(\cdot)$ 는 왜곡 보정 후의 픽셀 값을 나타낸다. $s_V([u, v])$ 는 이미지 공간에서 $[u, v]$ 에 위치한 픽셀에 해당하는 shading factor를 의미한다. 모든 픽셀에 대한 shading factor를 측정하고, 측정된 shading factor의 피팅 함수를 구한다. 각 픽셀에 대하여 VDC 수행 시마다 피팅 함수를 계산하여 shading factor를 근사하여 구하는 방식이 있다^[2-4]. 비네팅 왜곡이 일반적으로 방사 대칭이라 가정하고 다음과 같이 $R \triangleq u^2 + v^2$ 에 대한 다항식 $f_V(R)$ 를 측정된 shading factor의 피팅 함수로 도입한다^[5].

$$f_V(R) \triangleq \sum_{n=1}^{D_V} \alpha_n R^{n-1} \quad (2)$$

위의 $f_V(R)$ 에서 α_n 은 비네팅 계수로, Nelder-Mead downhill simplex method^[11]에 의하여 추정할 수 있고, $f_V(R)$ 의 차수를 결정하는 D_V 가 클수록 정교한 보정이 가능하지만 피팅 함수를 계산하기 위한 연산 복잡도가 증가한다. VDC는 이미지의 모든 픽셀에 대하여 (2)를 이용하여 shading factor를 구해 (1)과 같이 픽셀 값을 보정한다.

2. 배럴 왜곡 보정 (BDC)

BDC는 보정된 이미지의 각 픽셀에 대응되는 왜곡된 이미지의 픽셀 값을 바탕으로 이미지를 재구성한다. 보정된 이미지의 픽셀 위치에 대응되는 왜곡된 이미지의 픽셀 위치를 구하는 과정을 역 맵핑이라 하며, 역 맵핑에 기반을 둔 BDC 과정은 다음과 같이 표현된다.

$$I'([u, v]) \leftarrow I([u', v']) \quad (3)$$

여기서 $[u', v']$ 은 보정 대상인 픽셀 위치를, $[u, v]$ 은 $[u', v']$ 가 역 맵핑된 픽셀 위치를 나타낸다. $I'(\cdot)$ 는 VDC에서와 마찬가지로 왜곡 보정 후의 픽셀 값을 가리키기 위한 표기로 사용되었으나, VDC와 BDC 이후의 이미지가 동일하다는 것은 아니다. 그림 2는 역 맵핑 과정을 예시한 것으로, 배럴 왜곡 역시 일반적으로 왜곡의 정도가 방사상으로 대칭이라는 가정을 할 수 있기

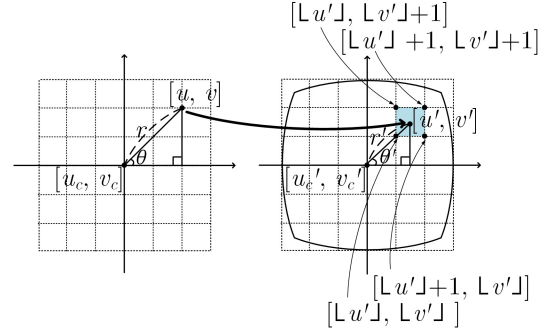


그림 2. BDC: 역 맵핑과 이중 선형 보간 (점 $[u', v']$ 와 $[u, v]$ 는 각각 왜곡된 이미지 공간과 보정된 이미지 공간에서의 중심)

Fig. 2. BDC: backward mapping and interpolation ($[u', v']$ and $[u, v]$ are distorted image space and corrected image space center, respectively).

때문에^[6-10] $\theta = \theta'$ 이고, 따라서 그림 2의 두 삼각형은 닮음 관계이다. 그림에서 두 삼각형의 닮음비인 scaling factor는 $s_B([u, v]) \triangleq u'/u = v'/v$ 로 정의되며, 이에 따라 역 맵핑은 다음과 같이 수행된다.

$$[u', v'] = s_B([u, v]) \cdot [u, v] \quad (4)$$

VDC에서와 유사한 방식으로, 측정된 scaling factor의 피팅 함수로 다음과 같이 $f_B(R)$ 을 도입한다^[6-10].

$$f_B(R) \triangleq \sum_{n=1}^{D_B} \beta_n R^{n-1} \quad (5)$$

$f_B(R)$ 에서 β_n 은 역 맵핑 계수로, 보통 정해진 격자무늬에 대한 왜곡 정도를 측정하여 최소 자승 추정에 의하여 구한다^[6-8]. 이 때, $f_B(R)$ 의 차수를 결정하는 D_B 가 클수록 정교한 보정이 가능하지만 피팅 함수를 계산하기 위한 연산 복잡도가 증가한다.

역 맵핑 과정을 통해 구한 픽셀 좌표는 정수가 아닐 수 있기 때문에, 해당 위치에서의 픽셀 값을 얻기 위해서는 다음과 같이 주변 픽셀 값을 이용하여 보간하는 과정이 필요하다.

$$I'([u, v]) = I([\lfloor u' \rfloor, \lfloor v' \rfloor]) \cdot (1 - u' + \lfloor u' \rfloor) \cdot (1 - v' + \lfloor v' \rfloor) + I([\lfloor u' \rfloor + 1, \lfloor v' \rfloor]) \cdot (u' - \lfloor u' \rfloor) \cdot (1 - v' + \lfloor v' \rfloor) + I([\lfloor u' \rfloor, \lfloor v' \rfloor + 1]) \cdot (1 - u' + \lfloor u' \rfloor) \cdot (v' - \lfloor v' \rfloor) + I([\lfloor u' \rfloor + 1, \lfloor v' \rfloor + 1]) \cdot (u' - \lfloor u' \rfloor) \cdot (v' - \lfloor v' \rfloor) \quad (6)$$

여기서 $[k]$ 은 k 보다 크지 않은 최대의 정수를 의미한다.

3. 기존 연구들의 한계점

실제 영상 시스템을 이용하여 촬영한 이미지에서는 비네팅 왜곡과 배럴 왜곡이 모두 발생하므로 두 왜곡에 대한 보정이 모두 필요함에도 불구하고, 기존의 연구들에서는 두 왜곡을 개별적으로 다루어 왔다^[2-10]. 앞 절에서 설명한 각 왜곡에 대한 보정 방법을 독립적으로 적용하여 두 왜곡을 보정할 수는 있으나, 이를 위해서는 VDC가 BDC보다 선행되어야 한다. 그 이유는, BDC에서 수반되는 역 맵핑 과정을 통해 이미지가 재구성되는데, 이 과정에서 기존의 왜곡된 이미지 공간에서의 각 픽셀 위치 정보가 소멸되어 재구성 이전 이미지에서의 각 픽셀 위치 정보를 이용한 shading factor의 계산이 불가능해지기 때문이다. 하지만 기존의 VDC 및 BDC를 적용하기 위해서는 (2), (5), (6)과 같은 계산을 이미지 상의 모든 픽셀에 대하여 수행해야 하므로, 높은 하드웨어 복잡도를 갖게 된다. 따라서 복잡도를 낮추기 위한 효율적인 구현에 대한 중요성이 대두된다.

III. 제안하는 프로세서

제안하는 프로세서에서는 VDC와 BDC에서 필요한 shading factor와 scaling factor를 구하기 위하여 복잡한 다항식의 피팅 함수를 직접 계산하지 않고, 피팅 함수를 구간 선형 근사하여 보정 품질을 유지하면서도 연산 복잡도를 크게 낮추었다. 또한, 비네팅 왜곡과 배럴 왜곡을 단일 파이프라인 구조 상에서 중첩적으로 처리하도록 설계하여 전체 시스템의 복잡도를 낮출 수 있도록 하였다.

1. 구간 선형 근사에 기반한 VDC 및 BDC

기존의 VDC 및 BDC 방법^[5-10]에서는, 각 픽셀에 대응하는 shading factor 와 scaling factor를 구하기 위하여 (2), (5)와 같은 $f_V(R)$, $f_B(R)$ 에 대한 복잡한 다항식의 계산이 필요하다. 더욱이, 이미지의 크기가 증가하여 보정해야 할 픽셀 수가 증가하면, 이에 따른 연산 복잡도 역시 크게 증가하므로 효율적인 구현을 가능케 하는 새로운 보정 방법이 필요하다.

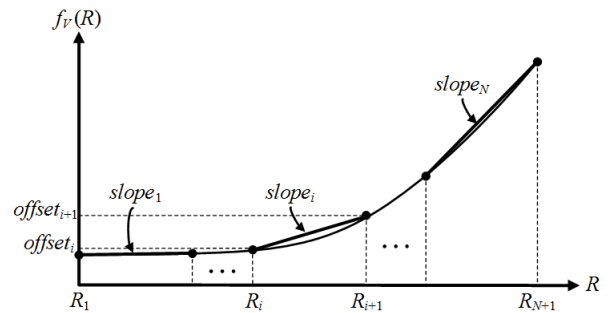


그림 3. VDC에서 피팅 함수의 구간 선형 근사
Fig. 3. Piecewise linear approximation of the fitting function in VDC.

제안하는 보정 방법에서는 shading factor와 scaling factor를 구할 때, 복잡한 피팅 함수를 직접 계산하지 않고, 피팅 함수를 구간 선형 근사를 통해 간단한 일차 함수로 대체하여 계산한다. 기존의 보정 방법^[5-10]에서는 주어진 측정 데이터를 바탕으로 커브 피팅을 통해 $f_V(R)$, $f_B(R)$ 의 계수를 추정한다. 그런데 이러한 측정 데이터는 그 자체로 어느 정도의 측정 오차가 포함된 것이므로, 이를 고차의 다항식을 도입하여 매우 정교하게 커브 피팅하는 것은 구현 관점에서 효율적이지 못하다.

제안하는 보정 방법에서는 고차 다항식으로 모델링된 피팅 함수를 구간 선형 근사하여 계산이 간단한 일차함수의 조합으로 대체하고, 이를 계산하여 shading factor와 scaling factor를 구한다. BDC에서의 scaling factor의 피팅 함수를 구간 선형 근사하는 과정은 VDC에서의 shading factor의 피팅 함수에서와 유사하므로 여기서는 shading factor의 피팅 함수를 구간 선형 근사하는 과정만을 설명한다. 그림 3은 곡선으로 표현된 shading factor의 피팅 함수 $f_V(R)$ 를 구간 선형 근사한 것으로, $f_V(R)$ 을 총 N 개의 구간으로 분할한다. 이 때, i 번째 구간, $1 \leq i \leq N$, 은 두 점 $[R_i, offset_i]$ 와 $[R_{i+1}, offset_{i+1}]$ 을 잇는 선분으로 정의되며, 이 구간에서의 $f_V(R)$ 는 다음과 같이 근사할 수 있다.

$$f_V(R) \approx slope_i \times (R - R_i) + offset_i \quad (7)$$

일차 함수의 기울기에 대한 정의에 의하여, $segment_i$ 에서 근사된 피팅 함수의 기울기 $slope_i$ 는 다음과 같이 $slope_i = (offset_{i+1} - offset_i) / (R_{i+1} - R_i)$ 로 구할 수 있다. $R_1 = 0$, $R_{N+1} = R_{MAX}$ 이고, $offset_i$ 과 R_i 사이에는 $offset_i = f_V(R_i)$ 인 관계를 만족한다. 여기

서 R_{MAX} 는 R_i 의 최대값이다. 이와 같이 구간 근사 과정을 통하여 선형 근사된 함수의 각 구간을 정의하는 R_i , $offset_i$, $slope_i$ 는 VDC나 BDC 이전에 미리 계산하여 외부 메모리에 별도로 저장해 놓은 후, 보정 중에는 미리 계산해 놓은 값을 반복적으로 재사용한다. 그 결과, 이미지의 크기가 증가하더라도 연산 복잡도가 크게 증가하지 않도록 함으로써 보다 효율적인 구현이 가능하도록 한다.

제안하는 방법에서는 구간 선형 근사에 기반을 두어 shading factor와 scaling factor를 구하는데, 주어진 피팅 함수를 구간 선형 근사하는 방식에 따라 해당 factor들에 대하여 피팅 함수를 직접 계산한 결과와 비교하여 오차가 발생할 수 있다. 이는 결과적으로 보정 품질에 영향을 미칠 수 있으므로 이러한 점을 고려하여, 구간 분할 방법을 적응적으로 택하거나^[14], 분할 구간 수를 증가시키는 방법을 통해 주어진 피팅 함수와 구간 선형 근사 사이의 오차를 줄일 수 있을 것이다. 하지만 본 논문에서는 주어진 피팅 함수에 대하여 구간 선형 근사로 유발되는 오차를 줄일 수 있는 새로운 구간 선형 방법을 제안하고자 하는 것은 아니며, VDC와 BDC에서 각

과정에 필요한 복잡한 피팅 함수 계산을 보정 품질의 열화 없이 구간 근사 과정으로 대체하는 방법 자체를 제안하고자 한다. IV장에서는 구간 근사에 기반을 둔 제안한 방법과 기존의 방법 사이의 보정 품질을 비교하여 제안하는 방법의 타당성을 평가할 것이다. IV장의 실험에서는 다항식 형태의 피팅 함수를 함수 값의 증감 정도에 따라 구간 분할 방법을 달리하였는데, 함수 값의 변화가 큰 곳에서는 구간의 폭을 좁게 하고, 함수 값의 변화가 작은 곳에서는 구간의 폭을 상대적으로 넓게 하였다. 하지만 이런 구간 분할 방법이 제안하는 방법에 종속적인 것은 아니며, 다른 새로운 구간 선형 근사 방식을 도입하여 보정 품질의 차이를 더 줄일 수도 있을 것이다^[15-16].

2. 제안하는 VDC 및 BDC 프로세서의 구조

앞 절에서 제안한 방법에 기반을 두어, 실시간으로 VDC와 BDC를 처리하는 왜곡 보정 프로세서를 설계하였다. 제안하는 프로세서는 단일 파이프라인 구조로 설계되었으며, VDC 유닛과 BDC 유닛으로 구성된다. 그림 4는 제안하는 프로세서의 전체 구조를 보인 것으로,

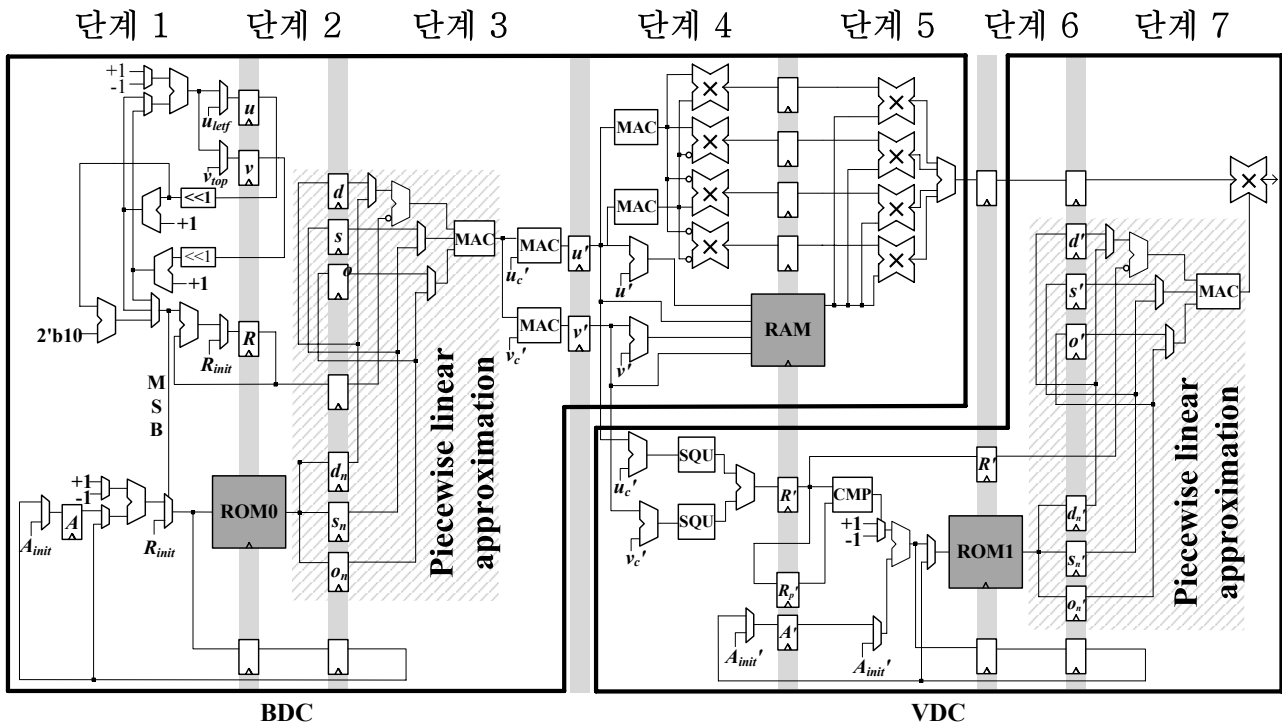


그림 4. 제안하는 VDC와 BDC 프로세서의 전체 구조
 Fig. 4. Overall architecture of the proposed processor for VDC and BDC..

왜곡 보정은 총 7단계에 걸쳐 수행된다. 단계 1-5에서는 BDC가, 단계 4-7에서는 VDC가 수행되며 특히, 단계 4-5에서는 VDC와 BDC가 중첩적으로 수행된다.

BDC를 수행하기 위하여, 단계 1에서는 u 의 초기값인 u_{left} 와 v 의 초기값인 v_{top} 을 입력 받은 후, u 또는 v 의 좌표를 한 픽셀씩 이동시키는 동시에 각 좌표에서의 R_n 을 계산한다. 계산된 R_n 은 R_{n+1} 의 계산을 위해 재사용되며, R_n 의 초기값은 R_{init} 으로 주어진다. 단계 2-3에서는 단계 1에서 계산된 R_n 을 바탕으로, 총 N 개의 구간 중 R_n 이 속한 구간의 $f_B(R)$ 을 구하기 위하여 구간 선형 근사를 수행한다. 이를 위해 각 구간에서의 R_i , $offset_i$, $slope_i$ 등을 외부의 read-only memory (ROM) 서 읽어온 후, 각 값을 이용하여 scaling factor를 구해 역 맵핑을 수행한다. 단계 4-5에서는, 역 맵핑 과정을 통해 구한 픽셀 좌표에서의 픽셀 값을 구하기 위하여, 왜곡된 이미지에서 보정하고자 하는 해당 픽셀 주변의 픽셀 값을 외부의 random access memory (RAM)에서 읽어와 이를 이용하여 보간한다.

VDC를 수행하기 위하여, 단계 4-5에서는 역 맵핑을 통해 구한 픽셀 좌표 u' 및 v' 을 바탕으로, 이를 이용하여 R' 을 구한다. 여기서, R' 은 비네팅 왜곡과 배럴 왜곡이 모두 발생한 이미지 공간상에서의 픽셀 반지름을 의미한다. 이처럼 VDC를 수행할 때, 배럴 왜곡이 포함된 이미지 공간에서의 반지름을 기준으로 하는 이유는, 광각 카메라를 통해 획득된 이미지에서는 두 왜곡이 함께 발생한다고 가정하므로 shading factor를 추정하는 과정에 배럴 왜곡이 영향을 주기 때문이다. 단계 6-7에서는, 단계 2-3에서와 마찬가지로, 단계 4-5에서 계산된 R_n' 을 바탕으로 외부의 ROM에 저장된 정보를 이용하여 shading factor를 계산한다. 최종적으로, 단계 7에서 BDC를 수행한 후 출력된 픽셀 값과 앞서 구한 shading factor를 곱하여 두 왜곡이 모두 발생한 이미지의 픽셀을 보정한다.

위에 언급한 것과 같이 단계 4-5를 수행할 때, 두 가지 보정이 중첩되어 이뤄질 수 있는 이유는, VDC가 왜곡된 이미지의 픽셀 값이 없이 픽셀 좌표만을 이용하여 수행될 수 있으므로 RAM을 액세스하지 않고도 보정이 이뤄질 수 있기 때문이다. 그 결과, 단계 1-3을 통하여 구한 왜곡된 이미지의 각 픽셀 좌표를 바탕으로 단계 4-5에서 VDC와 BDC가 중첩적으로 수행될 수 있다. 이러한 점을 기반으로 VDC와 BDC를 위한 하드웨어를

하나의 파이프라인 상에 적절히 결합하여 전체 시스템의 복잡도를 효과적으로 낮추었다.

제안하는 프로세서는 앞 절에서 설명한 것과 같이 구간 선형 근사를 이용한 방법에 기반을 두어 설계되었으며, shading factor 및 scaling factor에 대한 피팅 함수를 구간 선형 근사하기 위한 정보를 외부의 ROM에서 읽어온다. 구간 선형 근사를 기반으로 shading factor 및 scaling factor를 계산하는 부분을 그림 4에서 빗금으로 도시하였다. VDC와 BDC를 수행하기 위하여 각각 단계 6-7과 단계 2-3에서 각 피팅 함수를 구간 선형 근사하여 shading factor 및 scaling factor를 구한다. 구간 선형 근사에 기반을 둔 제안하는 프로세서에는 주어진 피팅 함수에 대하여 구간 선형 근사할 때, 어떠한 방식도 적용할 수 있다.

구간 선형 근사 시, 주어진 피팅 함수에 대하여 구간의 수를 증가시켜 보정 품질을 향상시킬 수 있는데, 이를 위해서는 외부 ROM의 크기가 증가되어야 한다. 하지만 본 프로세서는 외부 ROM의 크기와 무관하게 설계되었으므로, 이에 대한 추가적인 프로세서의 구조 변경이 불필요하다. 결론적으로, 제안하는 프로세서는 어떠한 새로운 구간 근사 방식을 적용하여 보정 품질을 향상시킴에 있어서도 확장 가능한 구조로 설계되었다.

IV. 구현 결과 및 고찰

본 장에서는 III장에서 제안한 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

제안하는 프로세서에서는 구간 선형 근사를 이용하여 shading factor 및 scaling factor를 직접 계산하지 않고, 구간 선형 근사를 통해 이를 간접적으로 구한다. 이를 통해, 복잡한 함수 대신 간단한 일차 함수를 계산함으로써 보정을 위해 요구되는 연산 복잡도를 효과적으로 낮출 수 있다. 또한, 고정 소수점(fixed-point arithmetic) 형식을 사용하여 효율적으로 하드웨어를 구현하였다. 제안하는 보정 프로세서를 구현함에 있어, 중요 변수에 대하여 각 변수별로 사용되는 비트 폭을 표 1에 정리하였다.

표 1. 제안하는 왜곡 보정 프로세서의 각 신호 별 비트 폭

Table 1. Bit width for each signal in the proposed processor.

신호	정수부	소수부
$d, d_n^{1)}$	23bits	0bit
$o, o_n^{2)}$	1bit	30bits
$s, s_n^{3)}$	0bit	13bits
u_{left}, v_{top}	11bits	0bit
R_{init}	23bits	0bit
$A_{init}, A_{init}'^{4)}$	4bits	0bit

¹⁾ $f_V(R)$ 또는 $f_B(R)$ 의 독립 변수

²⁾ i 번 째 구간에서 $offset_i$ 와 $offset_{i+1}$

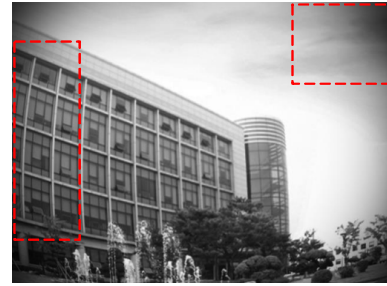
³⁾ i 번 째 구간에서 $slope_i$ 와 $slope_{i+1}$

⁴⁾ ROM0, ROM1의 초기 주소값

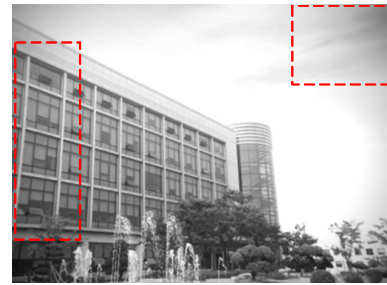
2. 구현 및 검증 결과

구간 선형 근사에 기반을 둔 제안하는 보정 방법과 기존의 방법^[5, 7]에 사이의 보정 품질을 비교하기 위한 실험을 수행하였다. 본 실험에서는, 기존의 방법^[5, 7]에 따라 shading factor 및 scaling factor의 피팅 함수를 각각 R' 과 R 에 관한 6차 다항식으로 모델링하였다 ($D_V=D_B=6$). 이와 같은 피팅 함수를 바탕으로 비네팅 왜곡과 배럴 왜곡을 보정한 영상을 획득하고, 각각의 피팅 함수를 제안하는 방법에 의해 구간 선형 근사한 함수를 바탕으로 보정한 영상을 비교하여, peak signal-to-noise ratio (PSNR)을 구하고 이를 표 2에 정리하였다. 표 2에 보인 결과를 보면, 제안하는 방법을 통해 피팅 함수 $f_V(R)$ 또는 $f_B(R)$ 를 구간 선형 근사하여 보정을 수행할 경우, 각각 근사함수의 구간의 수가 4 정도만 되더라도 평균 PSNR 수치가 30dB 이상임을 알 수 있다. 즉, 두 근사함수의 구간의 수가 4 이상인 경우, 피팅 함수를 직접 계산하는 기존 방식 대비 보정 품질의 열화가 크지 않음을 알 수 있다. 그림 5는 본 실험에서 기존의 방법과 제안하는 방법을 기반으로 보정한 이미지를 비교한 것이다.

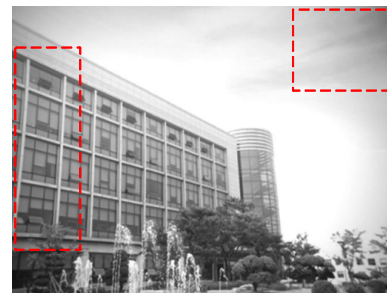
제안하는 프로세서는 0.11 μ m CMOS 공정의 셀 라이브러리를 사용하여 합성되었으며, 18.6K의 논리 게이트로 구현되었다. 제안하는 프로세서의 구현 결과 및 기존 논문의 결과를 비교하여 표 3에 정리하였다. 기존의



(a)



(b)



(c)

그림 5. 비네팅 왜곡과 배럴 왜곡 이미지의 보정 결과 비교 (a) 비네팅 왜곡과 배럴 왜곡 이미지, (b) [5]와 [12]의 보정 방법으로 보정된 결과, (c) 제안하는 보정 방법으로 보정된 결과

Fig. 5. Comparison of the correction results: (a) original image, (b) the result corrected by the conventional method [5], [7] and (c) the result corrected by the proposed method.

연구들 중에는, BDC를 위한 보정 방법과 이를 구현한 프로세서는 다수 존재하는 반면, VDC를 위한 보정 방법에 대한 연구 결과만 존재할 뿐 이를 하드웨어로 구현한 연구는 부족하다. 표 3에서 보인 것과 같이, 기존의 연구 [7], [8], [9]의 결과물은 BDC만 수행할 수 있음에 비해, 제안하는 프로세서는 BDC와 더불어 VDC도 가능하다. 또한, 게이트 수 관점에서 하드웨어 복잡도를 비교할 때, 제안하는 프로세서는 두 왜곡 보정이 모두 가능함에도 불구하고 기존의 연구 결과물 대비 효율적으로 복잡도를 낮춘 것을 알 수 있다. 제안하는 프로세

표 2. PSNR 관점에서의 보정 품질 비교 (단위 : dB)
Table 2. Comparison of the correction quality in terms of PSNR in dB.

Image ¹⁾	Number of segments		
	$N_V^{2)}=2,$ $N_B^{3)}=3$	$N_V=3,$ $N_B=3$	$N_V=4,$ $N_B=4$
01	26.112	28.187	31.154
02	25.642	27.270	35.646
03	28.521	29.904	35.081
04	31.680	33.650	34.545
05	28.680	29.399	37.485
Average	28.127	29.682	34.782

¹⁾ 640×480 크기의 이미지

²⁾ $f_V(R)$ 구간의 수

³⁾ $f_B(R)$ 구간의 수

표 3. 왜곡 보정 프로세서의 구현 결과 비교
Table 3. Comparison of implementation results.

	본 논문	[7]	[8]	[9]
기능	BDC, VDC	BDC	BDC	BDC
공정	0.11 μ m CMOS	0.18 μ m CMOS	0.18 μ m CMOS	0.11 μ m CMOS
게이트 수 ¹⁾	18,623	44,992	28,622	14,302
보정 속도	200 Mpixels/s	140 Mpixels/s	160 Mpixels/s	314 Mpixels/s
동작 주파수	200MHz	200MHz	200MHz	314MHz
보정 해상도	2048 × 2048	1024 × 1024	2560 × 2048	2048 × 2048

¹⁾ 가장 작은 2-입력 NAND 게이트를 1로 추산함.

서는 200MHz의 동작 주파수로 동작하며, 파이프 라인 구조에 의해 매 동작 주파수 마다 영상의 한 픽셀이 보정된다. 따라서 초당 200M 픽셀의 비네팅 및 배럴 왜곡을 보정할 수 있다. 동일한 동작 주파수를 갖는 기존의 결과물 중 [8]과 비교할 때, 제안하는 프로세서가 게이트 수는 0.65배 감소함과 동시에, 보정 품질을 개선하고 보정 속도 또한 1.25배 향상시켰다. 그림 6은 FPGA를 사용하여 제안하는 프로세서의 프로토타입 시스템을 구현한 것이다.

V. 결 론

본 논문에서는 비네팅 왜곡과 배럴 왜곡을 중첩적으로 보정하는 낮은 복잡도의 프로세서 구조를 제안하고 이를 구현한 결과를 제시하였다. 제안하는 프로세서에서는 구간 선형 근사를 통해 보정 과정에 필요한 고차 다항식의 계산과 같은 복잡한 연산을 효과적으로 제거하여 하드웨어 복잡도를 크게 낮추었다. 또한 비네팅 왜곡과 배럴 왜곡을 단일 파이프라인 상에서 중첩적으로 보정하도록 설계하여 전체 시스템의 복잡도를 낮출 수 있도록 하였다. 제안한 프로세서는 0.11 μ m 공정의 셀 라이브러리를 사용하여 18.6K의 게이트로 구현되었으며, 2048×2048 크기의 이미지에 대해 200Mpixels/s의 보정 속도를 갖는다.

REFERENCES

- [1] I. Dinstein, F. Merkle, T. D. Lam and K. Y. Wong, "Imaging system response linearization and shading correction," *IEEE Int'l. Conf. Robotics & Automation*, vol. 1, pp. 204 - 209, Mar. 1984.
- [2] S. J. Kim and M. Pollefeys, "Robust radiometric calibration and vignetting correction," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 30, no. 4, pp. 562 - 576, Ari. 2008.
- [3] S. B. Kang and R. Weiss, "Can We Calibrate a Camera Using an Image of a Flat, Textureless Lambertian Surface?" in *Proc. European Conf. Computer Vision*, pp. 640 - 653, Jul. 2000.
- [4] Y. Zheng, J. Yu, S. B. Kang, S. Lin and C. Kambhamettu, "Single-image vignetting correction using radial gradient symmetry," *IEEE Conf. Computer Vision and Pattern Recognition*, pp. 1 - 8, Jun. 2008.
- [5] D. B. Goldman and J. H. Chen, "Vignette and exposure calibration and compensation," in *Proc. IEEE Int'l Conf. on Computer Vision*, vol. 1, no. 1, pp. 899 - 906, Oct. 2005.
- [6] H. T. Ngo and V. K. Asari, "A pipelined architecture for real-time correction of barrel distortion in wide-angle camera images," *IEEE Trans. Circuits and Sys. for Vid. Tech.*, vol. 15, no. 3, pp. 436 - 444, Mar. 2005.
- [7] 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 and Sys. II: Express Briefs*, vol. 56, no. 1, pp. 51 - 55, Jan. 2009.
- [8] S. L. Chen, H. Y. Huang and C. H. Luo, "Time multiplexed VLSI architecture for real-time barrel distortion correction in video-endoscopic images," *IEEE Trans. Circuits and Sys. for Vid. Tech.*, vol. 21, no. 11, pp. 1612 - 1621, Nov. 2011.
- [9] H. S. Jung, W. T. Kim, G. H. Lee and T. H. Kim, "Design and Implementation of a Low-Complexity Real-Time Barrel Distortion Corrector for Wide-Angle Camera," *Journal of The Institute of Electronics Engineers of Korea*, vol. 50, no. 6, pp. 131 - 137, Jun. 2013.
- [10] H. S. Jung, Y. J. Park and T. H. Kim, "A Low-Complexity Real-Time Barrel Distortion Correction Processor Combined with Color Demosaicking," *Journal of The Institute of Electronics Engineers of Korea*, vol. 51, no. 9, pp. 57 - 66, Sep. 2014.
- [11] P. Muralikrishna, S. Prakash and B. H. Subbaraya, "Digital processing of spacelab imagery," *Advanced Space Research*, vol. 2, no. 7, pp. 107 - 110, 1982.
- [12] A. A. Sawchuk, "Real-time correction of intensity nonlinearities in imaging systems," *IEEE Trans. Computer*, vol. 26, no. 1, pp. 34 - 39, Jan. 1977.
- [13] W. P. Yu, Y. K. Chung and J. Soh, "Vignetting distortion correction method for high quality digital imaging," in *Proc. IEEE Int'l Conf. Pattern Recognition*, vol. 3, no. 3, pp. 666 - 669, Aug. 2004.
- [14] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing(2nd ed.)*. Cambridge University Press, 1992.
- [15] W. Li, S. Xu, G. Zhao and L. P. Goh, "Adaptive knot placement in B-spline curve approximation," *Computer-Aided Design*, vol. 37, no. 8, pp. 791 - 797, July, 2005.
- [16] J. G. Dunham, "Optimum uniform piecewise linear approximation of planar curves," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 1, pp. 67 - 75, Jan. 1986.

 저 자 소 개



문 선 아(학생회원)
2015년 현재 한국항공대학교 항공
전자정보공학부 재학 중.
<주관심분야 : 회로 및 시스템,
VLSI 설계, ASIC 설계>



홍 진 우(학생회원)
2015년 현재 한국항공대학교 항공
전자정보공학부 재학 중.
<주관심분야 : 회로 및 시스템,
VLSI 설계, ASIC 설계>



김 원 태(학생회원)
2013년 한국항공대학교 정보통신
공학과 학사 졸업.
2014년 현재 한국항공대학교 항공
전자정보공학과 석사과정
재학 중.

<주관심분야 : 회로 및 시스템, VLSI 설계, ASIC
설계>



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

2011년~현재 한국항공대학교 항공전자정보공학
부 교수.

<주관심분야 : 회로 및 시스템, VLSI 설계, ASIC
설계>