

임베디드 시스템 환경에서 최적화된 Homography 알고리즘

천승환* · 유영호* · 장시웅*

*동의대학교

An Optimized Homography Algorithm for Embedded System Environment

Seung-hwan Cheon* · Young-ho Yu* · Si-woong Jang*

*Dong-Eui University

E-mail : perari0@nate.com, yhyu@pusan.ac.kr, swjang@deu.ac.kr

요 약

자동차를 위한 전방향(omnidirectional) 감시 시스템, 교통 정보 수집 시스템 등 다양한 비전 시스템에 카메라가 장착되어 사용되고 있다. 최근에는 운전자의 편의를 위하여 광각 카메라의 비선형적인 방사 왜곡을 해결하는 왜곡 보정 작업 등의 영상처리 시스템이 많이 발전하여 운전자의 사각지대를 효율적으로 최소화하고 있다. 그러나 기존의 연구에서는 카메라로부터 입력되는 왜곡 영상을 보정하는데 별도의 H/W인 DSP(Digital Signal Processes) 또는 SOC(System On Chp) 형태의 전용 H/W를 추가하여 임베디드 시스템의 성능을 보완하고 있다. 하지만 위와 같은 별도의 H/W를 추가하여 임베디드 시스템의 성능을 보완할 경우 시스템이 복잡해지고 가격이 비싸진다는 단점이 있다. 본 논문에서는 이러한 문제점을 보완하기 위하여 왜곡 보정 알고리즘과 호모그래피(Homography) 알고리즘의 연산량을 감소시켜 임베디드 시스템 환경에서 추가의 H/W 비용없이 왜곡 보정을 수행하는 알고리즘을 제안하고, 제안한 알고리즘을 구현하여 실제 시스템에 적용한 결과를 제시하여 구현 타당성을 검증한다.

키워드

homography, 임베디드 시스템

1. 서 론

오늘날 비전 산업은 로봇의 시각 역할, 얼굴 인식 시스템 그리고 교통 정보 수집 시스템 등 카메라로부터 들어오는 영상의 정보를 이용하는 시스템들이 주를 이루고 있고 그 중에서도 영상의 3차원 구조 정보를 획득하는 방법에 대한 요구가 커지고 있다. 하지만 광각 렌즈가 부착된 카메라로 얻어진 영상들은 방사왜곡과 원근 왜곡이 존재하게 된다. 이러한 광각 카메라의 중심에서 벗어난 가장자리 부근에 심한 방사 왜곡(radial distortion)이 나타나고, 매우 큰 화각에서 촬영될 수 있는 얼굴 또는 특정 물체 등에 대해서 심한 원근 왜곡(perspective distortion)이 발생할 수 있게 된다. 이러한 문제점들을 해결하기 위해 정확한 왜곡 보정과 호모그래피 작업을 수행하여 정확한 영상을 획득하여야 한다.

기존의 연구에서는 카메라로부터 입력되는 방사 왜곡 영상을 실시간으로 보정하는데 소프트웨어 적인 방법으로 80%이상의 효과를 보았다 [1]. 하지만 호모그래피 알고리즘을 적용하기 위해서는 많은 연산량이 추가로 필요하며, 현재에는 호모그래피 연산을 하기 위해 별도의 시스템을 추가하거나 DSP(Digital Signal Processes) 또는 SOC (System On Chip) 형태의 전용 H/W를 추가하여 임베디드 시스템의 성능을 보완하고 있다. 이러한 별도의 시스템들은 이미 충분한 성능과 검증을 거쳤지만, 비용이 많이들고 시스템이 복잡해지는 단점이 있다. 이러한 문제점들을 개선한다면 품질 향상과 비용감소를 기대할 수 있다.

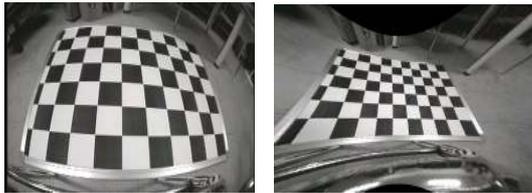
본 논문의 2장에서는 호모그래피에 관한 연구에 대해 설명하고 3장에서는 소프트웨어적인 개선방안을 살펴보고, 4장에서는 개발환경과 효율적인 호모그래피 적용 방법의 구현 및 성능평가를 하였으며, 5장에서 결론을 맺는다.

II. 관련 연구

2.1 방사 왜곡 보정

방사 왜곡 현상이란 카메라를 이용해 보여지는 영상을 화면에 출력하게 되면 직선이 곡선처럼 보여지는 현상을 왜곡 현상이라 한다. 이러한 현상을 카메라 렌즈의 왜곡 현상이라고 한다. 카메라 렌즈에서 발생하는 왜곡 현상에는 방사 왜곡 현상이 있다[1].

그림 1-(a)는 광각 카메라의 원본 영상이고, 그림 1-(b)는 방사왜곡 알고리즘을 적용한 후의 영상이다.



(a) 광각 카메라 영상 (b)방사왜곡보정 영상
그림 1. 방사왜곡 보정 결과

2.2 원근 왜곡 보정

호모그래피란 카메라를 이용해 보여지는 영상을 화면에 출력했을 때 정면이 아닌 측면이나 다른 방향을 바라보는 영상을 마치 정면에서 보는 듯한 영상으로 바꾸주는 일종의 변환 행렬이다. 만일 특정 물체의 영상이 그림 2-(a)과 같은 상황에서 촬영되었다면 호모그래피를 사용하여 그림 2-(b)와 같은 상황에서 촬영된 영상으로 변형시킬 수 있게 된다. 이러한 호모그래피(H)는 카메라 내부변수의 매트릭스(K)와 회전각을 통해 얻을 수 있는 회전 매트릭스(R)의 조합으로서 다음의 식(1)과 같이 표현될 수 있다[2].

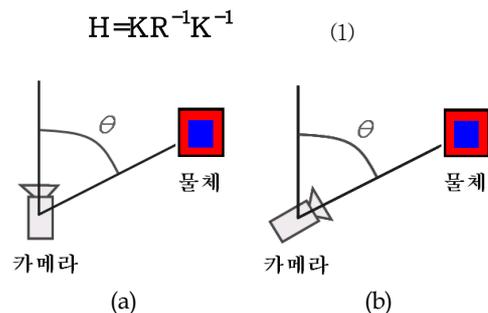


그림 2. 사물의 위치에 따른 원근왜곡보정

III. 개선 방안

2장에서 설명한 원근 왜곡을 보정하는 방법을 임베디드 시스템에 구현하여 적용할 경우 많은 연산량으로 인해 프레임의 실시간으로 받아

오지 못하는 문제점이 있었다. 이를 개선하기 위해 중복되거나 불필요한 소스를 수정 및 보완하는 다음 두 가지 방안을 제시한다.

- 각 픽셀마다 수행되는 for문에서 행렬 접근 순서의 변경을 통한 개선
- 호모그래피 평균값을 계산하는 과정을 단순화하여 개선

첫 번째 방법은 함수 내부의 for문을 사용할 때 대부분 0에서 1씩 증가하는 업카운팅 방식을 이용하지만 본 논문에서는 다운카운팅 방식을 이용하여 for문을 사용하게 되면 20% 정도의 성능 향상을 기대해 볼 수 있다[3].

두 번째 방법은 미리 호모그래피의 평균값 범위인 h[0]~h[7]까지의 8개의 평균값을 계산하는 과정을 단순화하기 위해 미리 정해진 평균값을 이용하여 연산을 통해 보정하는 방법 대신 미리 계산된 값을 파일에 저장하여 불러오는 방식을 사용함으로써 프로그램 내부에서 호모그래피 연산을 최소화시켜 25%정도의 성능 향상을 기대해 볼 수 있다.

IV. 구현 및 성능 평가

4장에서는 3장에서 제시한 두 가지 방법을 직접 구현하여 연산량 감소 효과를 실험을 통하여 확인한다. 그리고 제안하는 알고리즘이 임베디드 시스템에서 사용하기에 적합한 성능을 나타내는 지 실험을 통하여 검증한다.

4.1 구현 환경

본 논문에서 구현하는 개선된 호모그래피 알고리즘의 구현 환경은 다음의 표 1과 같다.

표 1. 시스템 사양

구성	사양
제품개발키트	IMX35 ARM11 Applications Processor 512MB of NAND Flash Memory 128MB of 32bit DDR2 SDRAM memory 운영체제: WinCE 6.0
카메라	Model: FO-3000CN Pixels: 320K (656H×492V) Lens: 180° (D)×140° (H)×100° (V)
개발언어	컴파일러: Visual C++ 2005 플랫폼 : iMX35-3DS-PDK1.7-Mobillity

4.2 성능 평가

4장에서는 3장에서 제안한 개선 방안을 직접 구현하여 호모그래피 알고리즘의 성능을 평가한다. 먼저 동일한 성능을 얻기 위하여 (640 X 480)크기의 영상으로 모두 통일시켰다. 원 영상은 초당 31프레임의 영상을 받아오는 결과를 얻

었다. 하지만 방사왜곡보정 영상을 개선 방안을 적용하지 않고 구현한 결과 2~3초당 1프레임을 받아오는 결과를 얻을 수 있었다. 이를 해결하기 위해 3장에서 제안한 각각의 개선 방안에 따라 얼마만큼의 성능 향상이 있는지를 테스트 한다.

먼저, 함수 내부의 for문을 다운카운팅 방식을 적용한 다운카운팅 알고리즘을 구현하여 성능을 평가한다. 다운카운팅을 했을 때, for문의 비교부분을 제거하여 테스트 단계를 최소화하도록 구현하였다.

다음으로 호모그래피 평균값을 파일에 미리 지정했다가 불러오는 방식을 사용한 알고리즘을 구현하여 성능을 평가한다. 호모그래피 평균값을 알고리즘에 바로 적용했을때보다 평균값을 파일에 저장되도록 한 뒤에 시스템에 불러와서 적용시키는 방법을 사용하여 알고리즘을 최소화하도록 구현하였다.

그림 3은 위의 두 가지 개선 방안을 적용한 알고리즘의 성능을 계산하여 초당 평균 프레임수를 나타낸 것이다.

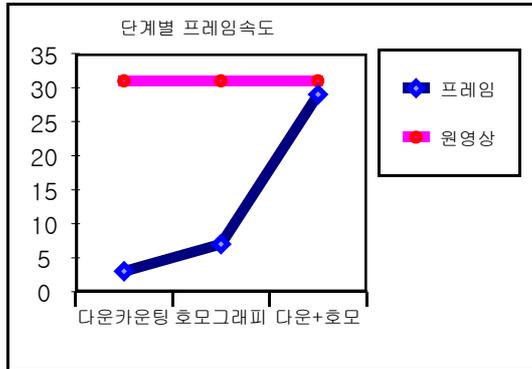


그림 3 개선 방안을 적용한 알고리즘

그래프에 나타나듯이 다운카운팅 알고리즘은 초당 3프레임을 받아들 수 있었고, 호모그래피 평균값을 파일에 미리 지정했다가 불러왔을 경우에는 7프레임의 성능향상이 있었다. 다운카운팅과 호모그래피 평균값을 동시에 적용하였을때는 원본 영상 31프레임에서 평균적으로 2프레임의 로스(loss)가 발생하여 29프레임을 받아와 처리하는 결과를 얻을 수 있었다.

그래프의 결과를 보면 원영상과 제안한 개선 방안을 모두 적용한 왜곡 보정 알고리즘은 초당 2프레임의 차이를 보이고 있다. 이 차이는 사용자가 디스플레이상에서 변화를 거의 느낄 수 없을 만큼의 수치이며, 원영상의 왜곡을 보정한 영상이므로 패턴인식 및 얼굴 검색 그리고 자동차의 전방위 감시 카메라 등의 영상처리 시스템의 기본 영상으로 사용될 수 있다.

그림 4는 원영상과 방사왜곡보정 결과, 그리고 원근왜곡보정 결과를 단계별로 적용한 결

과영상을 보여준다.

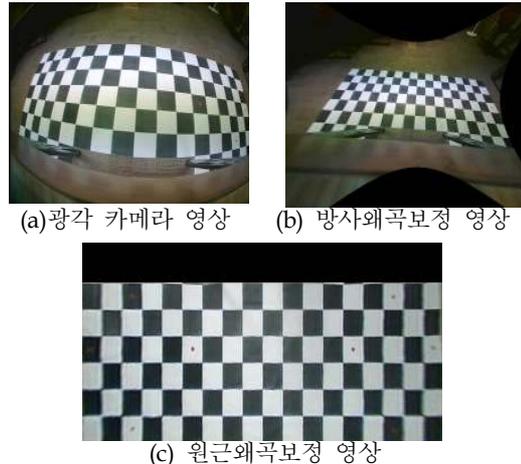


그림 4. 물체의 위치에 따른 원근왜곡 보정

V. 결 론

본 논문에서는 임베디드 시스템 환경에서 적용할 수 있는 광각 카메라로부터 입력되는 방사왜곡과 원근왜곡이 있는 영상의 왜곡 보정 알고리즘을 제안하였다. 알고리즘의 성능 향상을 위해 왜곡 보정 과정에 필요한 연산들을 분석하고, 연산량을 줄일 수 있는 개선 방안을 제안하였다. 또한, 제안한 개선 방안을 적용한 알고리즘을 직접 설계 및 구현하여 성능을 평가하였다.

본 논문에서는 임베디드 시스템 환경에서 영상을 제공할 때, 실시간으로 사용자에게 방사왜곡과 원근왜곡이 보정된 결과영상을 제공하는 것을 목표로 하였다. 현재의 결과는 초당 2프레임의 손실이 있지만, 사용자 실제로 사용하기엔 무리가 없는 성능을 보이고 있다.

향후 연구에서는 개선을 통하여 원영상과 동일한 성능을 낼 수 있도록 성능 향상 방안을 연구할 것이다. 또한 비전 산업의 로봇의 시각 역할, 얼굴인식 시스템 등의 여러 비전 산업에서 방사왜곡과 원근왜곡 현상뿐만 아니라 영상 정합, 마스크 적용 등의 고차원의 영상 변환 방법이 있다. 이러한 영상 변환 방법의 S/W적인 성능 향상 방법을 연구하고 위와 같은 영상 변환 방법들을 임베디드 시스템 환경에서 프레임의 손실 없이 사용할 수 있는 알고리즘들을 구현할 수 있을 것이다.

본 논문의 결과는 임베디드 시스템 환경에서 저비용의 영상처리 방법으로 활용될 수 있을 것으로 기대된다.

참고문헌

- [1] 천승환, 유영호, 장시웅 “임베디드 시스템 환경에서 효율적인 카메라 왜곡 보정 방법”, 한국해양정보통신학회 논문집, 제15권, 제1호, pp.623~626, 2011년 5월
- [2] R., Swaminathan, S., Nayar, "Nonmetric calibration of wide-angle lenses and polycameras", IEEE TPAMI 22(10), 2000
- [3] 김유진, “임베디드 프로그래밍 C코드 최적화”, (주)한빛미디어, pp.215-223, 2010년 3월.