

Car PC 기반 주변 감시 시스템에서 스레드를 이용한 왜곡 보정 알고리즘의 설계 및 구현

윤희돈* · 유영호* · 장시웅*

*동의대학교

Design and Implementation of Calibration Algorithm using Thread in a AVM System based on Car PC

Hee-Don Yoon* · Young-Ho Yu* · Si-Woong Jang*

*Dong-Eui University

E-mail : kiogee@lycos.co.kr, yhyu@pusan.ac.kr, swjang@deu.ac.kr

요 약

최근 네비게이션, GPS 등과 같은 운전 보조를 위한 장치 및 DVD 플레이어 등과 같은 운전자의 인포테인먼트 요구를 위한 장치 등 다양한 차량용 ECU들이 등장하고 있다. 차량용 네트워크를 통해 다양한 ECU들을 연결하여 상호 협력을 통해 ECU 고유 기능뿐만 아니라 추가의 기능을 수행할 수 있는 다양한 기술도 활발히 연구되고 있다. 차량용 네트워크를 통해 다양한 ECU들이 연결되어 상호 동작하기 위해서는 다양한 ECU 및 네트워크를 관리하는 서버 역할을 할 수 있는 고성능의 Car PC의 장착이 필수적이다. 기존 AVM 시스템은 차량 주변 상황을 실시간으로 제공하기 위해 임베디드 시스템 또는 SOC의 형태로 개발되었다. 그러나, Car PC가 차량에 장착되면 AVM 시스템을 추가의 비용이 없는 소프트웨어로 구현할 수 있다. 이를 위해서는 차량 주변 영상을 실시간으로 제공할 수 있도록 카메라 보정 같은 영상 처리 모듈의 성능이 실시간 영상을 제공할 수 있는 성능을 갖추어야 한다. 본 논문에서는 차량의 전·후방 및 좌·우측에 장착된 4대의 카메라로부터 입력된 차량 주변 상황을 한눈에 보여주는 AVM 시스템을 위해 카메라 보정 및 통합 처리를 위한 모듈을 스레드(thread)를 이용하여 설계하고 구현한다. 또한, 제안하는 영상 처리 모듈의 성능과 스레드를 사용하지 않는 방식의 영상 처리 모듈의 성능을 비교 분석함으로써 제안하는 영상 처리 모듈을 이용하여 추가의 비용이 없는 소프트웨어 AVM 시스템의 구현 가능성을 검증한다.

키워드

왜곡 보정, 주변 감시 시스템, 광각 카메라

1. 서 론

최근 자동차의 안전 운행 및 운전자의 요구를 수용하기 위해 다양한 차량용 ECU들이 등장하고 있다. 또한, 자동차에 장착되는 다양한 ECU들을 통합적으로 관리하고 활용하기 위해 차량용 네트워크로 ECU들을 연결하여 상호 동작하게 함으로써 ECU의 기능을 확장하는 기술도 활발히 연구되고 있다. 이러한 환경에서 차량용 네트워크 및 연결된 ECU 같은 장치들과 이들 장치들의 상호 협력을 통해 제공되는 다양한 서비스들에 대한 통합적인 관리를 위해 서버 역할을 수행하는 Car PC의 장착이 일반화되고 있다.

Car PC는 문서 작업, 음악 감상, 영화 감상 등 일반적인 개인용 PC와 성능 및 구조가 같으며 자

동차 내부에서는 GPS기능과 네비게이션과 같은 기능으로도 활용이 가능하여 Car PC의 활용 범위는 폭 넓고 다양하다. 운전자의 안전한 운행에 도움이 되는 AVM(Around View Monitoring) 시스템도 영상 처리 모듈의 성능이 실시간 영상을 제공할 수 있는 성능만 된다면 Car PC에서 소프트웨어로 개발하여 운영하는 것이 가능하다.

본 논문에서는 운전자에게 전·후방 및 좌·우측과 사각 지대까지 하나의 화면에 동시에 보여주는 주변 감시 시스템을 위한 영상 처리 모듈을 스레드(thread)를 이용한 알고리즘을 설계하고 구현한다. 제안하는 알고리즘의 성능을 스레드를 사용하지 않은 알고리즘과 비교함으로써 제안하는 알고리즘의 성능 향상을 분석하고, 제안하는 알고리즘을 이용하여 추가의 비용이 없는 소프트웨어

AVM 시스템의 구현 가능성을 검증한다.

2장에서는 AVM 시스템의 일반적인 처리 과정 및 카메라의 왜곡 현상에 대해 설명하고 3장에서는 스테드를 이용한 알고리즘과 스테드를 사용하지 않은 알고리즘에 대한 구조 및 처리과정에 대해 설명한다. 4장에서는 두 가지의 알고리즘의 성능을 비교 분석 하고, 5장에서는 결론을 서술한다.

II. AVM 시스템

운전자의 운행에 도움이 되는 AVM(Around View Monitor) 시스템은 주차, 골목길 운행 등의 경우에 차량의 앞뒤와 양옆의 360° 상황을 마치 차량의 위에서 내려다보듯 내부 모니터를 통해 실시간으로 보여줌으로써 운전자가 안전하게 운전할 수 있도록 돕는 시스템이다[1].

2.1 AVM 시스템의 구성 및 처리 절차

AVM 시스템은 차량 주변의 상황을 사각지대 없이 제공하기 위해 차량의 앞 그릴 아래, 뒤 번호판 위, 그리고 좌우 사이드 미러 밑 등의 위치에 광각(180°) 카메라를 장착하여 구성한다. 그림 1은 닛산의 AVM 시스템을 위해 장착한 카메라 위치를 보여준다[2].

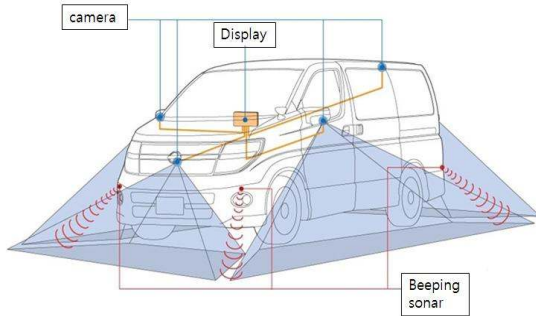


그림 1 AVM 시스템의 카메라 위치

카메라로부터 입력된 영상은 카메라 특성에 따라 왜곡된 영상이다. AVM 시스템은 영상의 왜곡 현상을 보정하기 위해 왜곡 보정작업, 호모그래피, 영상의 정합 과정을 통해 실시간으로 차량의 사각지대의 상황을 영상으로 제공하는 역할을 한다. 그림 2는 AVM 시스템의 처리 절차 및 구성을 나타내는 그림이다[5].

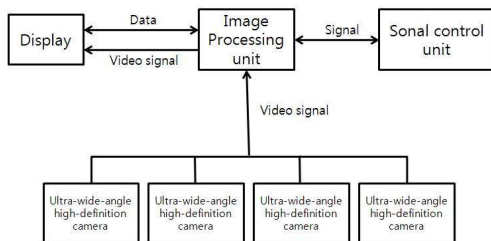
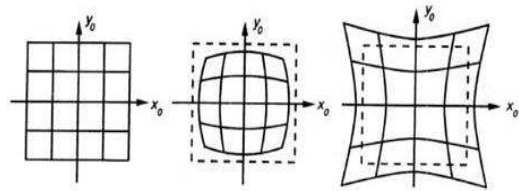


그림 2. AVM 시스템의 처리 절차 및 구성

2.2 방사 왜곡(radial distortion) 현상

카메라 왜곡 현상은 렌즈의 모양에 의해 발생하며, 카메라 렌즈의 영상 센서의 가장자리 부근에서 픽셀의 위치가 왜곡되는 현상을 발생 시킨다[4]. 다음 그림 3은 광각 카메라의 왜곡 현상을 보여준다[3].



(a)원본영상 (b)방사왜곡 (c)핀 쿠션

그림 3. 카메라 왜곡 현상

방사 왜곡 현상으로 인해 광각 카메라에서 받아들이는 영상이 인해 둥글게 보이게 된다. 그림 4는 AVM 시스템에서 받아들이는 왜곡 보정하기 전의 차량 외부 영상과 왜곡 보정을 한 후의 영상을 그림으로 나타낸 것이다[1].



(a) 왜곡 보정 전 출력 화면



(b) 왜곡 보정 후 출력화면

그림 4. AVM 시스템의 화면 출력 영상

III. 스테드를 이용한 왜곡 보정 알고리즘의 설계 및 구현

Car PC 기반 AVM 시스템은 차량의 전·후방 및 좌·우측에 장착된 4대의 광각 카메라를 장착하여 연결하고, 카메라로부터 입력된 차량 주변의 영상을 처리하여 하나의 통합된 영상을 실시간으로 운전자에게 제공하는 소프트웨어 AVM 시스템이다.

3.1 Car PC 기반 AVM 시스템의 처리 과정

Car PC 기반 주변 감시 시스템은 운전자에게 사각지대가 없는 차량 주변 상황을 보여주기 위

해 왜곡을 가진 4대의 광각 카메라의 입력 영상에 대해 왜곡 보정 및 호모그래피 연산을 수행한 후, 4개의 왜곡 보정된 영상을 하나의 영상으로 통합한다. 다음 그림 5는 Car PC 기반 AVM 시스템의 처리 과정을 보여준다.



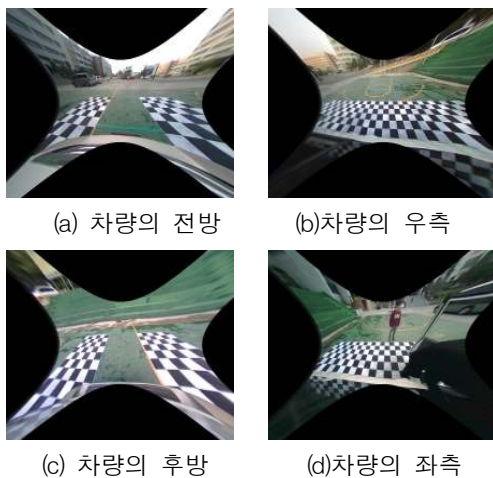
그림 5 Car PC 기반 AVM 시스템의 처리 절차

본 논문에서는 왜곡 보정 및 호모그래피 과정을 스테드로 구성하여 영상 처리 과정의 성능을 개선한다.

3.2 Car PC 기반 AVM 시스템의 왜곡 보정

Car PC 기반 AVM 시스템에서는 차량에 장착된 4대의 광각 카메라에서 발생하는 방사 왜곡 현상을 제거하기 위해 카메라 모델을 미리 파악한 후 전 \square 후방 및 좌 \square 우측 카메라 영상을 따로 받아들여 한 픽셀마다 왜곡 계수 들을 계산한다. 전 \square 후방 및 좌 \square 우측 4대의 카메라 각각에 들어오는 영상의 왜곡 현상을 제거 한 뒤 왜곡 현상이 제거된 이미지를 다른 평면 이미지로 옮기기 위해 호모그래피(Homography) 과정을 수행한다. Car PC 기반 AVM 시스템에서 사용되는 호모그래피는 3차원 공간상의 한 평면을 바라보았을 때 실제 보여지는 2차원 평면 형태로 바꾸는 투시 변환을 사용한다.

그림 6은 차량에 장착된 전 \square 후방 및 좌 \square 우측 카메라에서 받아들이는 영상을 왜곡 보정하여 하나의 이미지로 출력한 화면이다.

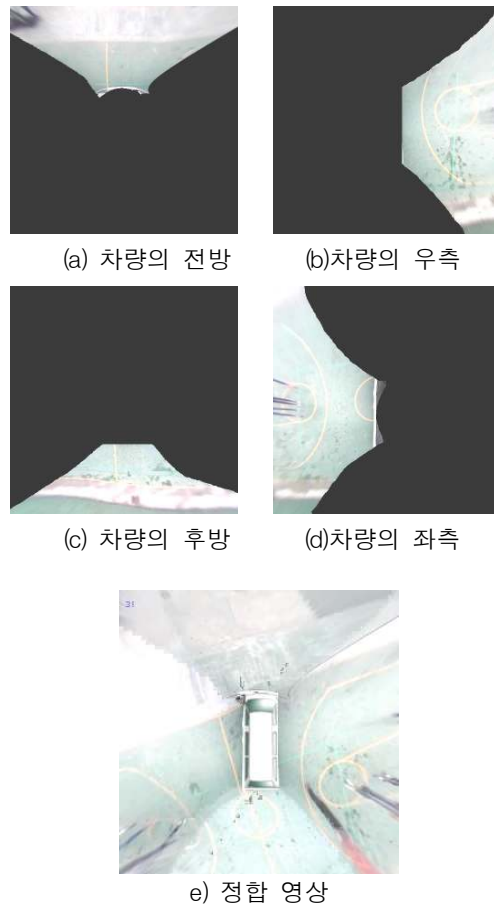


(a) 차량의 전방 (b)차량의 우측
(c) 차량의 후방 (d)차량의 좌측
그림 6 주변 감시 시스템의 왜곡 보정

3.3 Car PC 기반 AVM 시스템의 정합

차량을 위에서 아래로 바라보는 영상을 구현하기 위해서는 그림 6과 같이 왜곡 보정된 결과 영상과 카메라 전 \square 후방 및 좌 \square 우측에 대한 호모그래피 결과 값을 가지고 각각 전 \square 후방 및 좌 \square 우측 영상을 하나로 합쳐 주는 정합 과정을 거쳐야 된다. 정합 과정에서 프로그램의 연산량을 줄이기 위해 마스크 이미지를 사용하여 차량의 전 \square 후방 및 좌 \square 우측 영상을 하나로 정합 한다.

그림 7은 왜곡 보정 및 호모그래피 계산을 한 차량의 전 \square 후방 및 좌 \square 우측 결과 영상과 하나의 화면으로 정합한 영상이다.



(a) 차량의 전방 (b)차량의 우측
(c) 차량의 후방 (d)차량의 좌측
e) 정합 영상
그림 7 Car PC 기반 AVM 시스템의 정합 영상

IV. 성능 평가

Car PC 기반 AVM 시스템을 위해 구현한 스테드를 이용한 카메라 왜곡 보정 알고리즘의 성능을 평가하기 위해 스테드를 이용한 방식의 알고리즘과 스테드를 이용하지 않은 두 가지 방식의 알고리즘을 Car PC에서 실행하여 성능을 비교 평가한다.

4.1 알고리즘 구현 및 실험 환경

본 논문에서 성능 평가를 위해 사용한 Car PC

기반 AVM 시스템의 실험 환경은 다음 표 1과 같다.

구성	사양
Car PC	CPU: Pentium Dual-Core E5500 (2.8GHz) RAM: 2GB OS: MS Windows XP
프레임 그래버	Model: PICOLO Tetra (4 channel) 제조사: EureSys
카메라	Model: FO-3000CN Pixels: 320K (656H×492V) Lens: 180° (D)×140° (H)×100° (V)
개발 언어	컴파일러: Visual C++ 6.0 비전 Library: OpenCV Library 1.1

표 1. 시스템 사양

4.2 성능 평가

Car PC가 구현된 알고리즘이외의 다른 시스템 동작은 하지 않고 구현한 알고리즘만 동작할 경우의 성능을 분석하는 방법과 구현된 알고리즘이외에 다른 작업을 하는 과정에서 구현한 알고리즘의 성능을 분석하는 두 가지 방식으로 주변 감시 시스템의 성능을 평가하였다.

그림 8은 Car PC가 구현한 알고리즘만 동작할 경우의 성능을 평가한 결과를 나타낸다. Car PC가 다른 동작을 하지 않을 경우 스레드를 이용한 알고리즘의 경우에 초당 평균 31 프레임 정도의 속도로 영상을 제공하고 스레드를 이용하지 않은 알고리즘의 경우 초당 평균 12 프레임 정도의 속도로 운전자에게 영상을 제공하고 있다. 스레드를 통해 입력 영상을 기다리는 시간과 듀얼 CPU의 성능을 효율적으로 이용한 효과이다.

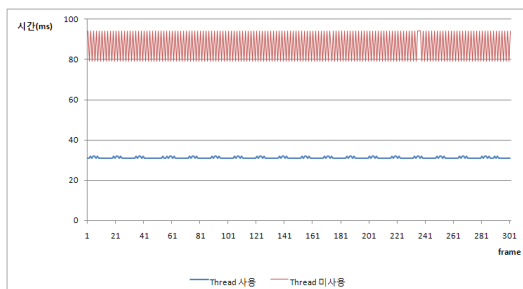


그림 8 왜곡 보정 작업만 동작할 경우 성능

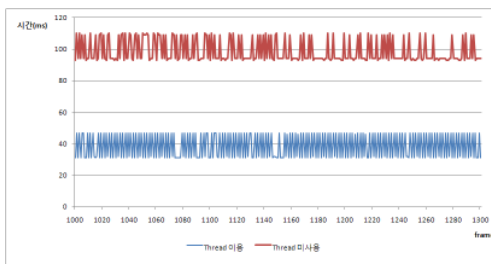


그림 9 다른 작업과 함께 동작할 경우 성능

그림 9는 Car PC가 KBS, MBC, SBS 홈페이지에서 각각 영상을 스트리밍 하는 상황에서 실행한 결과이다. 스레드를 이용한 알고리즘의 경우 초당 평균 25 프레임 정도의 속도로 영상을 제공하며 스레드를 이용하지 않은 알고리즘의 경우 초당 평균 10 프레임 정도의 속도로 영상을 제공한다. Car PC가 차량용 네트워크를 통해 다른 작업을 수행하는 중에도 AVM 기능을 수행할 수 있음을 알 수 있다. 또한, 스레드를 이용하지 않은 알고리즘의 경우 CPU 사용율이 50%이하인 경우가 많았으나 스레드를 이용하여 처리하는 시스템의 경우에는 CPU 사용율이 50%이상인 것으로 나타났다.

V. 결 론

본 논문에서는 Car PC기반 AVM 시스템에서 사용할 수 있는 스레드를 이용한 왜곡 보정 알고리즘을 설계하고 구현하였다. 또한, 스레드를 사용하지 않은 알고리즘과 비교 분석하고, 제안하는 알고리즘이 Car PC가 카서버 역할을 수행하면서도 AVM 기능을 수행할 수 있음을 보였다. 제안하는 알고리즘을 사용하면 Car PC가 장착되는 환경에서 추가의 비용없이 AVM 기능을 구현할 수 있다.

향후 연구에서는 제안하는 알고리즘을 초당 평균 30 프레임 이상의 성능을 낼 수 있도록 개선할 것이다. 또한, 제안하는 알고리즘을 사용하여 Car PC 기반의 AVM 시스템을 구현할 것이다. 이를 통해 소프트웨어 기반의 AVM 시스템을 구현함으로써 차량용 AVM 시스템의 구현 비용을 줄일 수 있는 방안을 제시한다.

Acknowledgement

본 논문은 중소기업청에서 지원하는 2010년도 산학연공동기술개발사업(No.00042243)의 연구수행으로 인한 결과물임을 밝힙니다.

참고문헌

- [1] INFINITY와 AVM <http://www.camenbert.co.kr/119>
- [2] Nissan's Elgrand Sensor System Simulates Birds-Eye View http://www.newlaunches.com/archives/nissans_elgrand_sensor_system_simulates_birdseye_view.php
- [3] Elements of Geometric Computer Vision, "http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/FUSIELLO4/tutorial.html#x1-67001r131"
- [4] Gary Rost Bradski, Adrian Kaehler, "Learning OPENCV 제대로 배우기" 2010. 3
- [5] NISSAN Around View Monitor <http://www.nissan-global.com/EN/TECHNOLOGY/OVERVIEW/avm.html>