

어라운드 뷰 모니터링 시스템과 자동 주차 시스템의 통합 구현

홍재근*, 정해°

Integrated Implementation of Around View Monitoring and Automatic Parking System

Jae-geun Hong*, Hae Chung°

요약

최근 차량 개발의 추세는 운전자에게 운송 수단뿐만 아니라 보다 더 많은 안정성과 편리함을 제공하고자 한다. 어라운드 뷰 모니터링 (Around View Monitoring, AVM) 시스템과 자동 주차 시스템 (Automatic Parking System, APS)은 운전자가 저속 주행 할 때 혹은 주차를 할 경우 도움을 주는 대표적인 주차 보조 시스템이다. AVM 시스템은 광각 카메라 모듈을 활용하여 운전자에게 운전자의 차량 주변의 영상을 제공한다. 본 논문에서는 AVM 시스템에서 운전자에게 차량 주변의 영상 이외에 주차선 검출기능, 고정된 장애물 검출기능, 그리고 이미지의 캘리브레이션 기능에 대한 구현 방안을 제시한다. 또한 주차 인접 지역에 도달한 차량이 자동으로 주차하는 장소를 찾고, 주차를 위한 경로를 생성하고, 초음파 센서를 통한 장애물 검출 및 경로 추종의 방법을 활용하여 자동 주차 기능을 제공한다. 이러한 기능들에 대해서 실제 차량을 기반으로 AVM과 APS를 통합하는 하나의 시스템을 구현한다.

Key Words : AVM, Calibration, APS, Path planing, Path tracking, Path following

ABSTRACT

Recently, the trend of vehicle development provides the driver with more stability and convenience as well as transportation. The Around View Monitoring system and the automatic parking system are typical parking assistance systems that are helpful when the driver is driving at low speed or when parking. The AVM system utilizes a wide-angle camera module to provide the driver with a video image of the driver's vehicle surroundings. In this paper, we propose the implementation method of PLD (Parking Line Detection), SOD (Static Object Detection) function and camera image calibration function in addition to AVM view which is the images around the vehicle in AVM system. Also, we provide a method for automatic parking function by searching for the parking place when the car arrives in the parking area, tracking the route, and using the ultrasonic sensor to detect obstacles. For these functions, we implement the system by integrating the AVM and APS system based on real vehicle.

* 이 연구는 금오공과대학교 학술연구비에 의하여 지원된 논문임.

• First Author : (ORCID:0000-0001-7782-0347)Kumoh National Institute of Technology, Department of Electronic Engineering, jaeguen.hong@lge.com, 정회원

° Corresponding Author : (ORCID:0000-0002-6920-2714)Kumoh National Institute of Technology, Department of Electronic Engineering, hchung@kumoh.ac.kr, 종신회원

논문번호 : KICS2017-10-322, Received October 26, 2017; Revised January 8, 2018; Accepted January 18, 2018

1. 서 론

차량 개발의 추세는 운전자에게 운송 수단뿐만 아니라 보다 더 많은 안정성과 편리함을 제공하고자 한다. 특히, 센서(Sensor)를 기반으로 운전자에게 정보를 전달하여 운전자의 안전과 편의성 및 즐거움을 제공하는 다양한 기술들이 발전하고 있고 늘어나는 추세이다¹⁻³⁾. 그림 1은 AVM(Around View Monitoring)의 조감도를 나타낸다.

AVM은 광각 카메라를 차량의 전방과 후방, 좌측과 우측에 장착하고, 영상처리 하는 ECU (Electronic Control Unit)를 차량 내부에 장착하여 MMI (Multi-Media Interface)로 운전자의 차량 주변 영상을 보여주는 장치로 ADAS (Advanced Driver-Assistance Systems)에서는 보편적으로 사용되는 용어이다. 운전 보조 시스템은 차량의 속도를 기준으로 저속의 경우엔 주차 보조 (parking assistance) 기능과 고속의 경우 주행 보조 (driving assistance)로 구분할 수 있다²⁾. AVM은 저속에서 주요 기능을 수행하여 주차 보조 기능이 높다고 할 수 있다. 주요한 기능은 차량 주위의 영역을 마치 하늘에서 내려다보는 이미지 (image)를 운전자에게 제공하여, 주차 및 좁은 골목 등에서 종종 발생하는 교통사고를 사전에 방지할 수 있도록 운전자의 사각 지대 (blind spot zone)를 포함한 차량 주위의 모든 방향에 대한 시야에 해당하는 이미지를 운전자에게 제공한다. AVM 시스템 본연의 차량 주변을 보여주는 뷰 (view) 기능 이외에 움직이는 물체 검출 등 다양한 검출 기능으로 확대되어가고 있는 추세이며, 초음파 혹은 레이더 (radar)와 함께 차량 주변의 이미지를 기반으로 자동주차 기능의 센서의 역할로도 확대되어 사용되고 있다³⁾.

본 논문에서는 AVM 시스템에서 카메라 설치 위치에 따른 광량의 차이로 인해 발생하는 이미지의 밝기 차이를 줄이기 위한 PA (Photometric

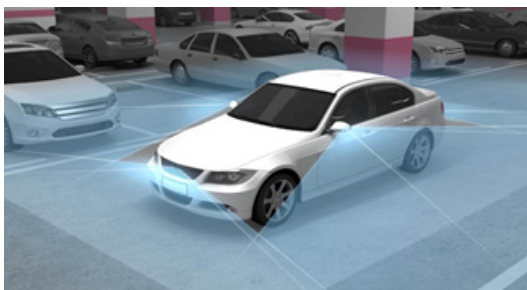


그림 1. AVM 조감도
Fig. 1. AVM Configuration

Alignment)와 셰이딩 (shading) 기법을 적용하였다. 또한 AVM 뷰와 자동주차 (Automatic Parking System, APS) 기능에서 사용하는 주차선 검출 기능 (Parking Line Detection, PLD) 및 고정된 장애물 검출 (Static Object Detection, SOD) 기능과 카메라의 자동 보정 기능을 구현하였다.

자율 주행의 일부분으로 혹은 주차에 대한 어려움에 대한 편의기능 제공 측면에서 자동주차 시스템 및 반자동 주차에 대한 방법이 상용화 되었고 또한 연구되고 있다. APS 방식은 사용하는 센서에 따라 분류 가능하며 널리 사용되는 두 가지 방식은 초음파 센서 기반과 카메라 영상 기반으로 나눌 수 있다. 초음파 방식은 조명, 환경, 날씨 등에 강인하게 빈 공간을 인식할 수 있으나, 주변에 주차된 차량이 존재하지 않는 상황에서는 사용할 수 없고 최종 주차 품질이 주변 차량의 자세에 의존하는 특성을 가진다. 영상 기반 방식은 주변 차량의 자세 및 존재 여부와 관계없이 주차 위치를 정확하게 인식하나 조명 환경, 날씨 등의 주변 환경에 민감하다는 특성을 갖고 있어 두 가지 주차 방식이 상호 보완적이라고 할 수 있다⁴⁾.

본 논문에서는 센서 즉, 초음파와 AVM에서 사용하는 광각 카메라를 기반으로 자동 주차 기능을 구현하였다. APS 기능이 수행되면 초음파 및 AVM 영상을 기반으로 주차 가능 공간을 찾고 고정된 장애물 검출 기능을 활용해서 주차 공간의 장애물을 확인하여 최종 주차 가능한 구역을 확정하는 과정을 수행한다. 주차 가능한 장소가 정해지면 주차를 수행하기 위해 차량의 이동 경로를 계획하기 위해 최소 비용 (minimum cost) 알고리즘으로 알려져 있는 Bi-directional Hybrid A* 알고리즘을 활용해 주차 경로 생성 (Path planning)을 수행한다⁷⁾.

주차 경로 추종 (Path following)에서는 위치 예측과 계획된 주차 경로를 바탕으로 어떻게 이동할 것인가를 결정한다. 센서와 차량 환경에 따라 계획한 경로를 따라 가속과 감속을 제어하는 ESC (Electronic Stability Control)와 회전을 담당하는 EPS (Electronic Power Steering) 및 기어 쉬프트를 사용하여 TCU (Transmission Control Unit)를 설정하고 차량을 움직이는 과정을 수행한다. 경로를 추종하는 과정 중에 장애물이 나타나면, 즉시 브레이킹 기능을 수행하여 주차 기능을 종료한다.

본 논문에서는 AVM과 APS에 대한 구현 내용을 바탕으로, 기존에 AVM과 APS가 분리되어 있는 시스템을 HW 및 SW의 통합 구조 설계를 통해 하나의 ECU에 통합 구현함으로써 가격 경쟁력을 높이고 차

량의 무게를 줄이고자 한다.

II. 광각 카메라 및 초음파 센서 모듈

2.1 광각 카메라 모듈

AVM에서 사용하는 카메라 모듈은 입력 받을 수 있는 해상도에 따라 SD (Standard Definition) 혹은 HD (High Definition) 급으로 나눌 수 있고, ECU로 전송하는 방식에 따라 LVDS (Low Voltage Differential Signaling) 혹은 Ethernet 혹은 아날로그 방식으로 나눌 수 있다. 차량에 장착하는 디스플레이의 추세가 HD급으로 사용되고 있는 추세에 있어, 입력 센서의 일종으로 사용되는 카메라 모듈도 해상도에 따른 영상의 열화를 줄이고자 대부분 HD급 카메라를 LVDS 혹은 Ethernet으로 전송하는 추세이다^{8,9)}.

그림 2는 AVM에 사용하는 전장용 카메라 모듈의 구조를 나타내는 것으로 이미지 센서 블록과 ISP (Image Signal Processor) 및 송신부로 구성되어 있다.

이미지 센서 블록은 렌즈와 CMOS 이미지 센서를 구동하기 위한 주변 소자로 구성되어 있다. 센서 블록의 주요한 역할은 렌즈를 통해 들어온 광 이미지를 전기적인 신호로 변환하는 것이다. ISP와 Tx 블록은 전기적인 신호로 변환된 이미지의 후처리 과정을 통해 이미지 향상 과정을 수행하고 전송 선로를 통해 이미지를 ECU로 보내는 역할을 한다.

AVM에서 사용하는 렌즈는 넓은 영역을 보기 위해 광각 렌즈를 사용한다. 광각 렌즈는 렌즈의 중앙부에서 가장자리로 갈수록 렌즈 곡률에 의해 발생하는 이미지 왜곡이 심해지는데, 이러한 왜곡을 렌즈 왜곡이라고 한다. 또한 카메라 모듈을 제작하면서 발생하는 유의차에 따른 왜곡과 차량에 장착하는 각도 위치에 따른 왜곡도 있다. 이러한 왜곡을 처리하기 위해서 카메라 모듈 자체의 왜곡을 변수로 나타내며, 이를 내부 매개변수 (intrinsic parameter)라고 하고, 차량 장착으

로 발생하는 왜곡변수를 외부 매개변수 (extrinsic parameter)라고 한다¹⁰⁾.

광각 렌즈의 왜곡보정을 수행하기 위해서는 광각 렌즈의 왜곡 함수 또는 왜곡 방식을 알고 있어야 한다. 왜곡 함수를 렌즈 설계자로부터 공급을 받으면, 이의 역함수 또는 근사 함수를 계산하여 왜곡된 이미지에 적용하여 왜곡을 보정할 수 있다. 렌즈 왜곡의 경우, 일반적으로 넓은 FOV (Field Of View)를 가지기 위해서 볼록한 형태의 배럴 왜곡(barrel distortion)을 가지고 있다. 그림 3은 광각 렌즈를 반구형 영역을 이미지 평면에 투영하는 형태를 도식화한 그림이다.

r_u 는 왜곡되지 않은 투영 반경 (주점으로부터 투영된 상까지 거리), r_d 는 왜곡된 투영 반경, 그리고 f 는 투영면과 주점의 거리를 나타낸다. θ 는 라디안 (radian) 단위를 가지는 입사각이다. Equidistance 함수로 광각 렌즈를 설계한 경우, 렌즈 왜곡에 의한 투영 반경은

$$r_d = f \tan^{-1} \frac{r_u}{f} \tag{1}$$

이다. Equidistance 함수에 의해 왜곡된 이미지 보정은 역함수를 활용하여 아래와 같이 구할 수 있다.

$$r_u = f \tan \frac{r_d}{f} \tag{2}$$

왜곡보정은 이러한 계산된 왜곡 관계식과 순방향 맵핑 혹은 역방향 맵핑 방법을 통해서 수행 할 수 있다. 순방향 맵핑을 적용하였을 경우, 왜곡보정의 결과 이미지에 빈 공간이 발생하는 단점이 발생하여 일반적으로 역방향 맵핑을 사용한다.

그림 4는 본 논문에서 사용한 체커 보드 패턴에서 광각 카메라 모듈의 출력 이미지를 왜곡 보정한 이미지를 나타낸다.

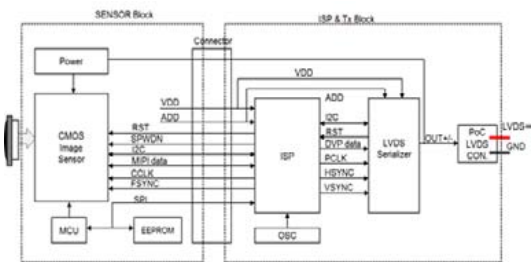


그림 2. 카메라 모듈의 구조
Fig. 2. Camera module block diagram.

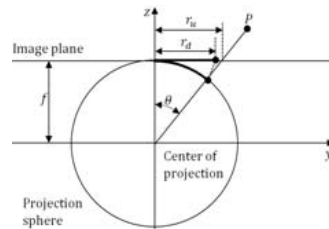


그림 3. 광각 렌즈 투영 반경.
Fig. 3. fish-eye lens projection diameter.



(a) 왜곡 이미지 (b) 역방향 맵핑 이미지 결과

그림 4. 왜곡이미지 역방향 맵핑 결과
Fig. 4. mapping image. (a) distorted image (b) reversed mapping image result

렌즈의 왜곡으로 인해 중심부를 기준으로 볼록한 영상임을 확인 할 수 있고, 이를 펼 경우, 가운데 부분이 움푹 들어간 형태임을 확인할 수 있다.

2.2 초음파 센서

초음파 센서는 차량에서 분산된 전자기기를 연결하여 사용하는 직렬 통신 시스템으로, 장애물을 검출하는 목적을 가지는 저렴한 솔루션 중에 하나이다. 초음파 센서는 차량 앞과 뒤쪽 범퍼 위치에 장착하여 장애물이 있을 경우, 장애물에 대한 거리 정보를 획득하여 운전자에게 경고 혹은 MMI를 통하여 정보를 제공하기 위해 사용한다. 그림 5는 차량에 장착한 초음파 센서의 위치와 거리 측정원리를 나타낸다.

본 논문에서는 자동 주차 기능을 수행하기 위해 2m 거리에 있는 장애물을 검출 할 수 있는 단거리 초음파 8개를 차량의 앞과 뒤에 각각 4개의 모듈을 장착하였다. 5m까지 검출할 수 있는 장거리 초음파의 경우 앞쪽 펜더 (fender) 양쪽에 장착하였다. 초음파 센서의 거리측정은 초음파 센서에서 물체까지의 거리를 d , 음파의 속도를 v_c , 초음파 센서의 송신부와 수신부의 거리가 동일함을 가정하면($\cos\theta = 1$), 송출한 초음파가 물체에 반사되어 되돌아오는 시간 T (TOF: Time of Flight)를 다음과 같이 계산 할 수 있다.

$$T = \frac{2d}{v_c} \quad (3)$$

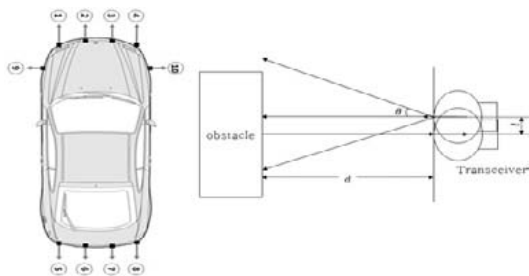


그림 5. 초음파 위치 및 거리 측정
Fig. 5. ultrasonic positions and distance measurement

III. AVM

그림 6은 AVM 2D 뷰 기능을 구현하기 위한 일반적인 과정을 도식화 하였다.

첫째 광각 카메라의 이미지의 왜곡을 펴는 과정이다. 둘째 시점변환을 통해 차량의 위에서 아래를 내려 보는 영상을 만드는 과정이다. 셋째, 전후좌우의 시점 변환한 이미지를 잘라서 붙이는 과정으로 AVM 뷰를 구현할 수 있다.

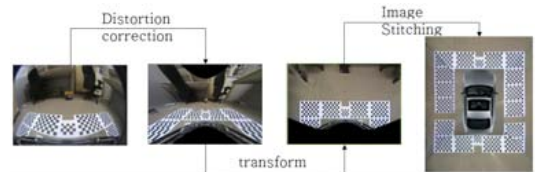


그림 6. AVM 뷰 이미지 생성 과정
Fig. 6. AVM View generation process

3.1 AVM 3D View

초기 AVM 뷰 (view)의 경우, 차량을 위에서 아래로 내려다보는 bird eye 뷰 혹은 TOP 뷰 형태로 차량 주변을 보여주는 형태의 UI를 나타내었다. 최근 SoC의 발전으로 처리속도 및 가격 경쟁력이 발생하면서 다양한 뷰를 운전자에게 제공하는 추세이며, 이런 추세 중 대표적인 것이 3D 뷰이다. 그림 7은 F사가 구현한 3D 뷰의 예로써, Warp around view를 나타낸다.

AVM용 광각 카메라 이미지는 이미지의 원근을 차량 주변 일정 거리 공간에 특정한 Bowl 형태의 구조를 가정하여 카메라 각각의 이미지를 특정한 bowl에 스티칭 (stitching) 혹은 맵핑하는 방법으로 표현하여 운전자가 요구하는 다양한 형태의 뷰를 제공할 수 있다.

그림 8의 (a)는 중심에서 차량의 전방과 후방, 좌측과 우측의 카메라로부터 받은 이미지를 bowl 형태로 맵핑하는 것을 형상화 하였다. 그림 8의 (b)는 카메라 입력 관점에서 g만큼은 바닥을 기준으로 맵핑하는 영역을 나타내고, s는 bowl의 곡면에 해당하는 이미지를



그림 7. Wrap Around View
Fig. 7. Warp Around View

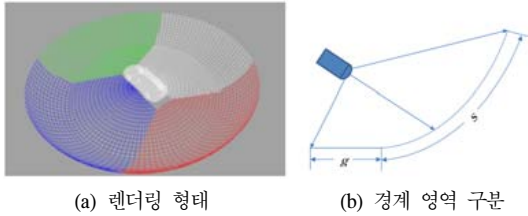


그림 8. 렌더링 형태 및 경계영역 구분
 Fig. 8. rendering type and board boundary area classification (a) rendering type (b) boundary area classification

맵핑하는 것을 나타낸다.

II절에서 언급한 광각 렌즈 투영 반경 모델을 기반으로 한 왜곡보정에서 무왜곡을 r_u , 왜곡을 r_d 라고 가정한 경우 관계식은 아래와 같이 표현할 수 있다.

1. 무왜곡 관계식: $r_u = \tan g(r_d)$,
2. 왜곡 관계식: $r_d = g^{-1}(\tan^{-1}(r_u))$

여기서, $f=1$, $\theta = g(r_d)$ 와 r_d 쌍의 값은 $(r_d(k), \theta(k))$, $\theta(k) = (0^\circ, \dots, 90^\circ, \dots)$, $k = 1, 2, \dots, N$ 이다. 보정에 대한 일련의 과정은 $g(r_d)$ 곡선을 정하는 것으로 구할 수 있다. 역함수 곡선 $g^{-1}(r_d)$ 는 $g(r_d)$ (입사각)을 획득한 이후에, $g(r_d)$ 곡선으로부터 획득된 많은 수의 $(r_d(k), \theta(k))$ 데이터 쌍으로 예측된다. 특정한 i 위치에서, 카메라 이미지 포인트와 실제 지면에서의 포인트는

$$x_i = HX_i \quad (4)$$

로 표현할 수 있다. 여기서 X_i 는 실제 이미지 평면의 이미지 포인트들이고, x_i 는 카메라 이미지 포인트이며, 투영 변환의 호모그래피 H 는

$$H = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \quad (5)$$

로 표현이 가능하다^[11].

본 논문에서는 임의의 bowl 형태에 맵핑하는 행태로 텍스처 맵핑을 SoC 내부의 GPU 블록과 OpenGL2.0을 활용하여 구현하였다^[12,13]. 메쉬 테이블(Mesh table) 생성은 Vertex attributes의 조합으로, 반구 형태의 부분과 4개의 카메라 각각에 대한 Vertex

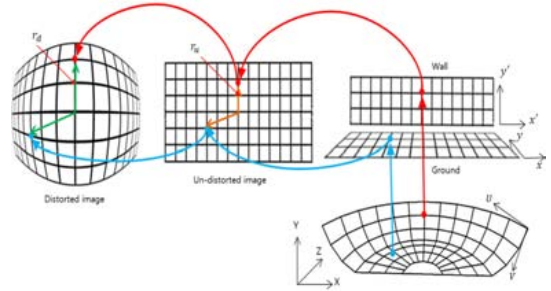


그림 9. 맵핑 과정에 대한 도식도
 Fig. 9. mapping diagram

좌표(X, Y, Z)와 텍스처(Texture) 좌표(U, V)의 정점의 특성 집합이다. 그림 9는 최종 이미지가 맵핑되는 좌표를 가져오는 과정을 나타낸다.

왼쪽 그림은 카메라의 입력 이미지로 왜곡 이미지이고, 가운데 그림은 왜곡을 편 이미지이다. 오른쪽 그림은 왜곡을 편 이미지를 바다와 벽으로 맵핑한 것이고, 오른쪽 아래 그림은 이를 기반으로 반구 형태로 맵핑하는 것이다. 반구에 맵핑되는 텍스처 좌표(U, V)를 획득하는 과정은 다음과 같다.

1. UV 공간에서 이미지 공간으로 스케일링을 한다. : $[-1, 1] \times [-1, 1] \rightarrow [0, w] \times [0, h]$
2. 호모그래피(H' or H)에 따라 왜곡을 편 이미지 공간에 맵핑한다.
3. 변환하고자 하는 지점에 대해서 g 왜곡 모델을 활용하는 기준 좌표계에 맵핑한다. 다른 측면에서 보면, Vertex 좌표 (X, Y, Z)을 계산할 수 있고, Vertex 좌표 (X, Y, Z)는 텍스처의 UV 좌표에 해당한다. 여기서 $r_d = g^{-1}(\tan^{-1}(r_u))$, $r_u = \sqrt{x'_{im}{}^2 + y'_{im}{}^2}$, $\begin{pmatrix} x'_{im} \\ y'_{im} \end{pmatrix} = H' \begin{pmatrix} x' \\ y' \end{pmatrix}$ 의 관계이다.

좌표의 맵핑 이후의 카메라 입력 이미지 각각에 대한 경계 영역 처리는 Global alpha blending을 활용하였다^[14]. Fig. 7의 오른쪽 그림에서 AVM 뷰의 상/하 좌/우 이미지의 부자연스러움은 카메라의 장착 위치에 따른 입력 광량의 차이 때문에 발생한다. 본 논문에서는 이러한 부자연스러움을 해결하기 위해서 셰이딩(shading)을 활용한 음영 처리 방법과 PA(Photometric Alignment)를 적용하여 광량의 차이를 보정하였다^[6]. 셰이딩은 스무드 셰이딩 방법을 사용하였다. PA는 카메라 모듈의 설치 위치에 따른 광량의 입력 차이로 인해 발생하는 밝기의 차이 부분을 보상



그림 10. 3D AVM 뷰 생성 결과
Fig. 10. 3D AVM View result

하는 기술이다¹⁶⁾. 본 논문에서는 인접한 카메라 이미지가 겹치는 이미지의 영역을 80×60 크기의 픽셀 영역으로 설정하고, 겹치는 이미지 영역의 각각의 평균 휘도를 계산하여 가장 낮은 휘도를 모든 이미지에 적용하였다. 그림 10은 텍스처 맵핑, 렌더링 및 PA를 적용하여 구현한 3D 뷰의 결과물이다.

3.2 Calibration

Calibration(캘리브레이션) 과정은 카메라 모듈의 설치 이후, 차량의 생산 라인의 마지막 과정에서 실시하는 것으로써 작업자의 공차를 줄이고자 실시하는 과정이다. 그림 11은 캘리브레이션 과정의 블록도이며, 다음과 같은 처리를 수행한다.

1. 광각 카메라 렌즈의 왜곡 모델을 기반으로 내부 매개 변수와 외부 매개 변수를 활용하여 입력된 광각 카메라 이미지에 대한 왜곡을 펴서 왜곡이 없는 이미지로 변환한다.
2. 케니 에지 검출(canny edge detection)을 활용하여 차량의 생산 라인에 설치된 캘리브레이션 패턴(그림 10의 체커보드 패턴)과 각 영역의 카메라에서 왜곡을 편 이미지를 비교해서 패턴을 검출한다¹⁵⁾.
3. 예측한 외부 매개 변수 및 광각 카메라의 위치에 관한 정보를 활용하여 Projection 캘리브레이션 절차를

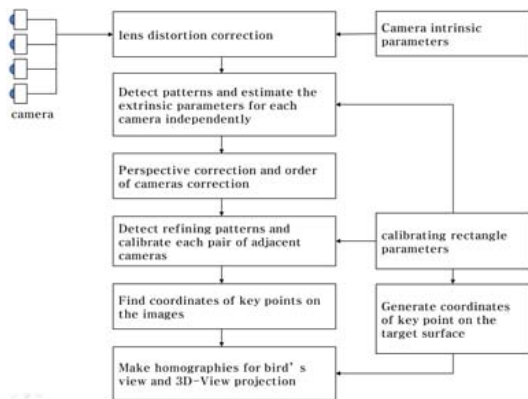


그림 11. 캘리브레이션 과정 블록도
Fig. 11. calibration process block diagram

를 수행한다.

4. 캘리브레이션된 패턴(왜곡 보정된 패턴)에 대해서 재정의한 패턴을 검출하고 구역화 한다. 캘리브레이션한 카메라 이미지의 영역에 겹쳐진 영역을 활용하여 외부 매개 변수를 갱신한다.

5. 이미지 좌표계를 활용해서 96개(8×12개)의 좌표를 계산한다.

6. 바닥면의 규칙적인 매트릭스와 수직을 계산하기 위한 파라미터를 사용해서 96개의 좌표를 다시 생성한다.

7. 이미지들의 쌍과 실제 좌표를 활용하여, TOP 뷰와 TOP 뷰를 90° 회전하여 대응되는 사이드 뷰 호모그래피를 만들고, 시스템을 갱신한다.

IV. APS

자동차의 보편화 및 운전의 편의성 및 주차의 편의성에 대한 요구가 증가하고 있다. 궁극적으로 운전하지 않으면서 이동하는 방법으로 자율 주행에 대한 활발한 연구가 진행되고 있다. 이러한 자율 주행의 일부 부분으로 또한 주차에 대한 어려움에 대한 편의 기능 제공 측면에서 자동 주차 및 반자동 주차에 대한 방법이 다양한 방식이 상용화 되었고 또한 연구되고 있다¹⁷⁾.

시장에 출시한 제품 관점에서 보면, TOYOTA의 2003년 Prius 모델은 주차 기능에 대하여 "Park assist"라 불리는 제품으로 수평주차 기능이 소개되었다. 이 기술은 센서로부터 주차 공간에 대한 정보를 취득하여 자동으로 움직이는 스티어링 휠을 차량이 조절하는 가운데, 운전자는 가속 페달과 제동 페달을 조작하는 일종의 반자동 주차 기술이라고 할 수 있다. 반자동 주차 기술은 주차를 진행하는 과정에서 운전자가 일정 부분 기여하는 방식으로 즉 가속과 감속 혹은 정지를 수행하는 방식이다. 주차 기능에서 운전자의 기여 부분까지도 고려한 자동 주차의 기술은 TOYOTA의 Lexus 2007년의 LS460에 장착되었다. 이 시스템은 운전자가 주차에 대한 명령을 내리면 차량 자체적으로 스티어링 휠 조작 및 가속, 감속 및 정지를 자동으로 실행하여 수평 주차하는 시스템이다^[16,17].

그림 12은 본 논문에서 구현한 자동 주차 블록도이다.

주차 기능은 실제 차량을 움직이게 하는 EPS, ESC, TCU을 제어하기 위한 블록으로 구성되며, 주차하려는 차량 위치를 기반으로 주차 가능 공간 탐색과 주차 경로 예측 및 추종을 통해서 차량을 제어한다.

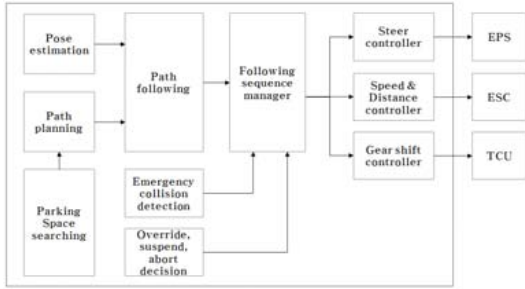


그림 12. 자동 주차 블록도
Fig. 12. auto parking block diagram

사용한 차량의 모델은 차량이 주차를 수행할 경우, 저속(15km 이하)으로 움직이기 때문에 타이어 슬립(tire slip)을 고려하지 않은 기하학적 모델(kinematic model)을 활용하였다¹⁸⁾.

그림 13은 차량의 움직임을 단순화하여 차량의 이동 거리를 나타낸다.

여기서, C 는 차량의 회전축 중심이고, 차량의 진행 방향의 좌표축은 (x_b, y_b) 이고, 움직이는 거리는 Δs 이다. 차량의 움직임이 작다고 가정하면 차량의 뒷바퀴의 위치 (x, y) 에서 차량의 회전 반경을 따라 이동하는 것을 직선화하여 그림 13의 아래 그림과 같이 표현할 수 있다. 전역 좌표계에서 x, y 방향으로 이동한 거리를 $\Delta x, \Delta y$ 로 나타낼 때 아래와 같이 표현할 수 있다.

$$\begin{aligned} \Delta s &\approx \Delta s_1 \\ \Delta x &= \Delta s \cos\left(\psi + \frac{\Delta\psi}{2}\right) \\ \Delta y &= \Delta s \sin\left(\psi + \frac{\Delta\psi}{2}\right) \end{aligned} \quad (6)$$

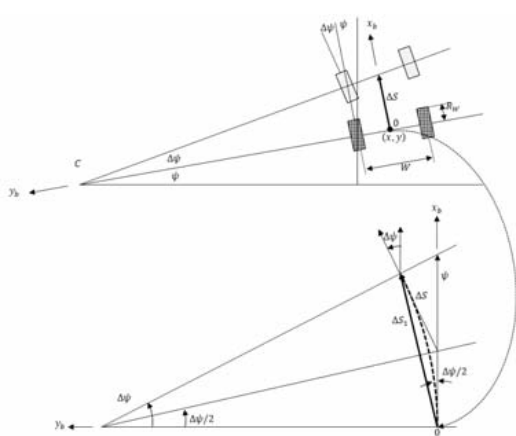


그림 13. 차량 움직임에 따른 거리 계산
Fig. 13. distance measurement according to vehicle moving

주차 공간은 주차장의 주차선 혹은 길 가장자리 혹은 노면의 주차선 등의 다양한 형태를 가지고 있다. 한국의 경우 주차장법 시행규칙으로 수직 주차선 및 사선 주차선과 수평 주차선을 제시하고 있으며, 호주에서 제안한 ISO 16787에서는 주차장의 크기, 주차 형태, 주차 최대 속도, 일부 평가 방법이 기술되어 있다^{19,20)}.

그림 14는 주차 가능 공간을 찾기를 수행하는 블록도이다.

영상을 활용한 주차 공간 인식은 주차선 인식과 주차선 내에 포함된 물체가 있을 경우 물체를 인식하는 기능으로 구분된다. 주차선 인식의 방법은 AVM 카메라로 영상을 획득한 이후, 전처리 단계로 R/G/B 영상을 그레이 레벨로 변경하고 (3×3) 크기의 필터를 활용하여 가우시안 블러 (Gaussian blur)를 수행한다. 주차선의 간격을 추가로 활용하여 주차선을 추출하는 과정을 수행하고 수직 주차선, 수평 주차선 및 사선 주차선을 분류한다. 분류한 구획선은 차량의 속도 정보를 활용하여 순서대로 번호를 매긴다. 번호가 매겨진 추종하는 정보는 SOD 블록에 전달한다. 전달 받은 정보를 바탕으로 SOD 블록은 ROI (300×720 픽셀)영역에 대해 이전 프레임($t-1$ frame)과 현재 프레임(t frame)의 이미지 데이터를 비교하고, 차량의 heading 각도와 차량의 속도를 고려하여 차이 나는 영역을 추출한다. PLD 영역의 주차선 좌표와 동기화해서 고정된 물체를 추출한다. 그림 15는 SOD를 포함한 PLD의 수행 결과를 나타낸다.

(a)는 AVM 뷰 원본 이미지를 나타내고, (b)는 고정된 물체를 그리드 맵으로 보여준다. (c)는 ROI 영역 (녹색 부분)에 대한 주차 가능 공간에 대한 좌표와 고정된 물체의 좌표를 나타낸다. 주차 가능 공간(d)은 SOD를 수행하면서, 주차 가능 영역을 나타내는 것

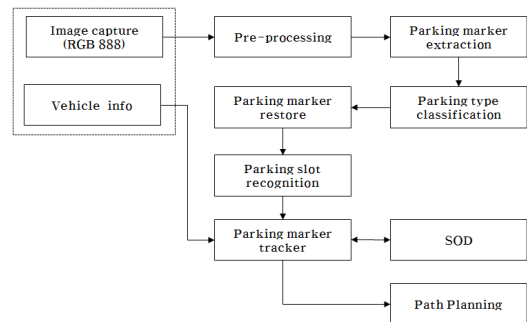


그림 14. 주차 가능 공간 찾기 블록도
Fig. 14. parking space searching block diagram

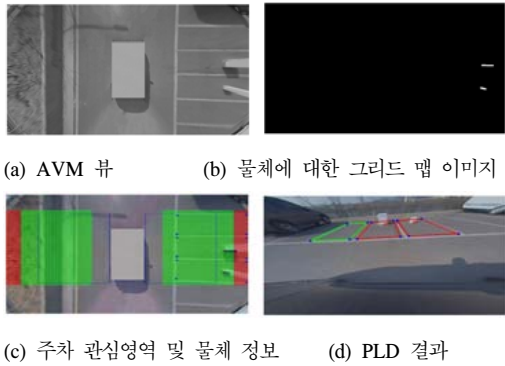


그림 15. 고정된 물체를 검출하는 주차 공간 찾기 결과
 Fig. 15. PLD result with SOD (a) AVM view (b) grid map about object (c) parking slot and coordinate value (d) PLD result with SOD

로 주차 가능 영역의 고정된 물체가 있는 것은 붉은 색으로 나타내었다. 주차 탐색 과정은 SOD의 최종 결과 받아서 주차 가능 구역을 확정함으로써 최종 주차 탐색을 완료한다.

경로 계획은 전역 경로 계획(global path planning)과 지역 경로 계획(local path planning)으로 구분 가능하다. 전역 경로 계획은 맵상의 출발점과 도착점의 위치 정보를 바탕으로 차량이 이동하기 전에 최적 경로를 찾기 위한 목적이고, 지역 경로 계획은 주차를 진행하고 있는 차량의 센서 기반으로 획득한 데이터를 기반으로 장애물을 회피하는 목적이다. 이러한 지역 경로는 경로 추종을 위해 사용된다.

본 논문에서는 Bi-directional Hybrid A* 알고리즘을 활용해 주차 경로 계획을 수행하였다^[5]. A* 알고리즘은 그림 16에서 볼 수 있듯이 8가지의 경로 생성이 가능하지만 차량의 이동이 불가능한 수직 방향을 포함한다.

Hybrid A* 알고리즘은 차량이 목표 위치까지 역방향 포함하여 6가지 방향으로 이동 가능한 경로를 찾는다. 지역 경로계획은 두빈스 커브 (Dubins curve)와 레드-셰프 커브 (Reeds-Shepp curve)를 사용하였다^[21].

주차 경로 추종에서는 위치 예측과 주차 계획을 바탕으로 센서와 차량 환경에 따라 계획한 경로를 따라 차량의 가감속을 제어하는 ESC와 회전을 담당하는 EPS 및 기어 쉬프트를 사용하여 TCU를 설정하고 차량을 움직이는 과정을 수행한다^[22].

그림 17은 차량제어를 위한 인터페이스 블록을 나타낸다.

자동 주차 기능은 차량의 위치와 경로 예측을 통해

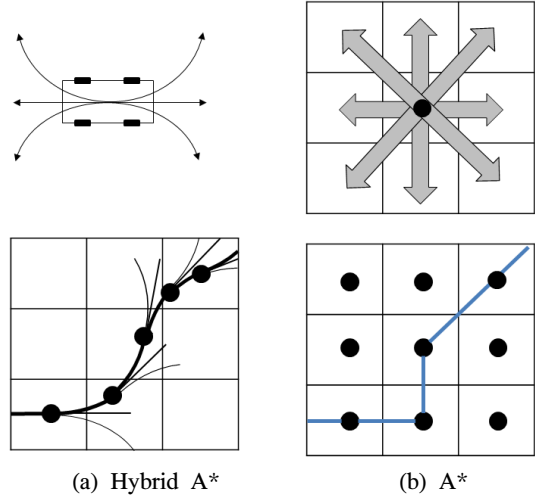


그림 16. Hybrid A*와 A*의 경로 예측 비교
 Fig. 16. Path planing Comparison of Hybrid A* and A* (a) Hybrid A* (b) A*

차량이 어떻게 움직여야 하는지 결정하고, 실제 액추에이터를 제어하기 위한 신호를 CAN (Controller Area Network) 메시지를 통해 해당 디바이스로 정보를 전달한다.

Steer controller는 결정한 차량의 조향각을 SAS (Steer Angle Sensor)와 비교해서 EPS를 제어한다. Steer controller의 look up table은 차량 조향부의 non-linear steer ratio map을 통해 타이어 각도를 조향각으로 변환하는 역할을 수행한다.

Rate limiter 블록은 EPS 모터 사양에 맞게 회전 속도의 최대값을 제한하는 역할을 수행한다. LPF 블록은 노이즈 제거 및 차량의 동적 응답을 고려한 입력을 만들어주기 위한 필터 블록이다. 이렇게 기준 정보를 활용하여 최종 EPS 제어 신호를 발생시킨다. Speed & distance controller는 차량의 속도와 이동 거리를 제어하는 블록으로 차량의 ESC를 제어하기 위한 인

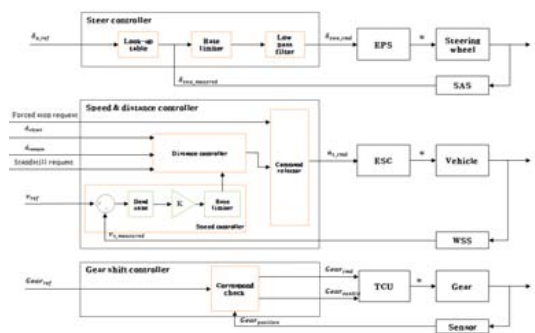


그림 17. 자동주차 제어의 인터페이스 블록도
 Fig. 17. interface block diagram of the APS control

터페이스 블록으로 최종 출력은 ESC에 차량의 가속도 정보 (a_{x_cmd})를 CAN 인터페이스를 통해 전달한다. Speed controller 블록은 속도 비교기, dead zone 판단기 및 rate limiter로 구성되어 있다.

Dead zone 블록은 차량의 WSS (Wheel Speed Sensor)의 값과 생성한 기준 속도의 차이를 활용해서 차량 속도의 흔들림을 제어하기 위한 블록이다. Rate limiter 블록은 가속도의 미분 값을 jerk limit로 제한하기 위한 블록이다. Distance controller는 현재 경로의 목표 거리와 장애물 거리를 실시간으로 판단하여 목표 위치에 멈추기 위한 제어기이다.

Command selector 블록은 FSM (Following Sequence Manager)로부터 출력된 플래그(flag)를 통해 우선순위가 가장 높은 상황에 대한 가속과 감속도를 출력하는 블록이다. Gear shift controller 블록은 차량의 센서를 통해 현재의 기어 정보와 기준 기어 정보를 비교하여 TCU를 활성화 시키고, 기어 명령을 출력하는 블록이다.

그림 18에서는 수평주차, 수직주차, 사선주차에 대해 주차 경로 계획에 따른 실제 차량의 주차 경로 및 추종 결과를 MATLAB을 활용하여 그래프로 나타내었다.

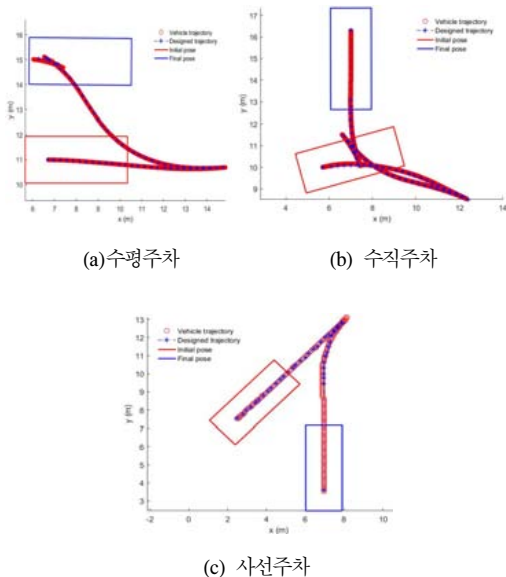
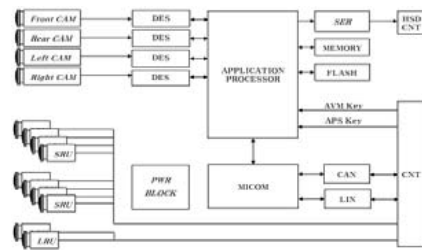


그림 18. 주차 방법에 따른 주차 경로 및 추종 결과
Fig. 18. path planning and tracking results according to parking methods. (a) parallel parking (b) perpendicular parking (c) angled parking

V. 통합 시스템

AVM 시스템과 초음파 모듈을 활용하여 장애물의 거리를 측정하는 초음파와 장애물 검출 시스템과 자동 주차를 담당하는 시스템의 ECU는 일반적으로 각각 분리된 별도의 시스템으로 구성되어 있다. II절과 III절에서 설명한 3가지 시스템을 통합의 장점은 전체 시스템 가격 경쟁력 및 ECU의 무게를 줄일 수 있는 장점을 가지고 있다. 이러한 통합은 애플리케이션 프로세서의 성능이 과거 대비 급속하게 좋아지고 있는 추세이고, 자동차 제조업자들의 가격경쟁력 확보 측면에서 일치하여 차량용 시스템의 통합은 점점 더 활발해지고 있다.

그림 19는 AVM와 APS 기능을 통합 구현한 시스템 블록도 및 시스템 연결 구조와 시스템을 장착한 차량 및 ECU HW를 나타낸다.



(a) 시스템 블록도



(b) 시스템 연결 구조 및 장착 차량



(c) ECU HW

그림 19. 시스템 블록도 및 연결 구조 및 장착 차량 형상 및 ECU HW
Fig. 19. system diagram, connection method and vehicle feature mounted system
(a) system diagram (b) connection method and vehicle feature mounted system (c) ECU HW

AVM 시스템은 HD (1280×720) 해상도의 LVDS 인터페이스를 가지는 광각 카메라를 장착하였다. 초음파 센서 모듈은 SRU (Short Range Ultrasonic sensor) 8개 (검출 거리: 2m)와 LRU (Long Range Ultrasonic sensor) 2개 (검출 거리: 5m)를 장착하여 자동 주차 기능에서 장애물 검출용으로 활용하였다. 카메라 영상을 입력 받은 ECU는 카메라 모듈에서 직렬화한 이미지 정보를 DES (Deserializer)를 통해 병렬화하여 영상 데이터를 원 영상으로 복원하고 애플리케이션 프로세서로 보낸다. 애플리케이션 프로세서에서는 영상을 프레임 단위로 캡처하여, 메모리로 전달하고 관련 영상 알고리즘 연산을 수행한다. 알고리즘 연산을 위해 영상 처리 데이터를 버퍼링하기 위해서 DDR 메모리를 사용한다. 연산을 수행한 이후 SER (Serializer) 블록에서 직렬화하여 ECU에서 MMI 인터페이스로 AVM 영상을 전달한다. 전달한 AVM 영상은 MMI의 디스플레이를 통해서 표현된다. MICOM은 차량 네트워크와 통신하는 CAN과 초음파 센서 모듈과의 통신을 위해 LIN (Local Interconnect Network) 인터페이스를 가진다.

그림 20은 통합 시스템의 소프트웨어 구조를 나타낸다.

통합 ECU의 SW 구조는 영상 처리 기능을 담당하는 애플리케이션 프로세서 부분과 차량 네트워크 통신과 전원관리 및 주차기능 및 시스템의 진단 및 안전 기능을 담당하는 MICOM 부분으로 나뉘어 구현하였다. 애플리케이션 프로세서는 시스템에 전원이 ON 되면, AVM 시스템은 차량의 배터리 전원이 공급되는 시점에 초기화 과정을 수행한다.

운전자가 기어를 “R” 상태로 조작 할 경우, AVM의 후방 카메라 영상을 MMI로 보여줘야 하기 때문에 AVM 출력은 2초 이내에 영상을 MMI로 보내주는 것을 NHTSA (National Highway Traffic Safety

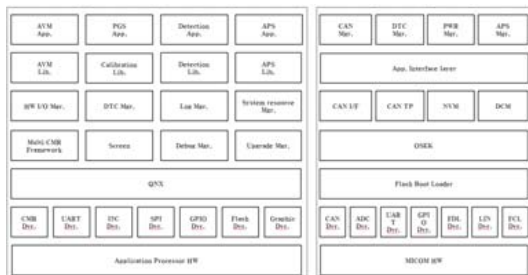


그림 20. AVM과 APS 통합 SW 구조
Fig. 20. Integrated S/W architecture of the AVM and APS

Association)에서는 권고하고 있다. 애플리케이션 프로세서에 실시간 운영체제인 QNX 기반으로 CDA (Custom Display Application)을 활용하여 카메라 영상을 2초 이내 영상을 표현할 수 있도록 하였다. MICOM의 경우, OSEK OS를 사용한다. 초기화 이후 차량과 CAN 통신이 가능하며, 통합 시스템의 상태 정보 및 동작을 MMI의 명령어에 따라 수행한다. APS의 차량 연동 블록은 기능 안전에 관련되어 있어 MICOM 블록에 별도 구현하였다.

그림 21은 AVM 시스템의 동작시나리오이다.

시스템 전원이 On 및 ACC가 On (Accessory on) 되면, AVM은 초기화 수행 이후, 차량의 정보를 받아 해당 동작을 수행한다. 시스템은 ACC의 On/Off 상태와 차량의 속도와 기어 상태와 timeout 시간에 따라 standby1/2 모드, 동작모드로 천이된다. 시스템은 동작한 상태에서 차량의 속도가 특정 속도가 넘을 경우, 애플리케이션 프로세서의 영상 출력을 끊음으로써 소모 전력을 줄인다. 마스터 역할을 하는 MMI와 슬레이브 역할을 하는 AVM 시스템은 CAN 메시지를 통해 상태 정보를 교환하거나 MMI의 요청으로 상태를 천이하고, 천이한 이후 AVM의 상태 정보를 재전송한다. 또한 차량의 턴 시그널(turn signal)에 따라 UI를 표현한다.

그림 22는 APS 동작 시나리오이다. APS 기능은 MMI 혹은 외부 버튼을 통해 진입한다. 진입한 이후 운전자가 주차할 것인지 혹은 출차 할 것인지를 선택한다. 주차를 수행할 경우에는 주차 방식을 수직/수평/대각선 주차 타입을 선택한 이후 4.2절에 설명한 PLD 및 SOD 기능을 활용하여 주차 공간 찾기를 수행한다. 통합 시스템의 APS 제어기에 의한 액츄에이터 동작으로 인해 속도 센서 값이 30 Km/h 이상일 경우, MMI에 팝업 (pop-up) 메시지를 보내고, 주차 기능을



그림 21. AVM 동작 시나리오
Fig. 21. AVM operation scenario

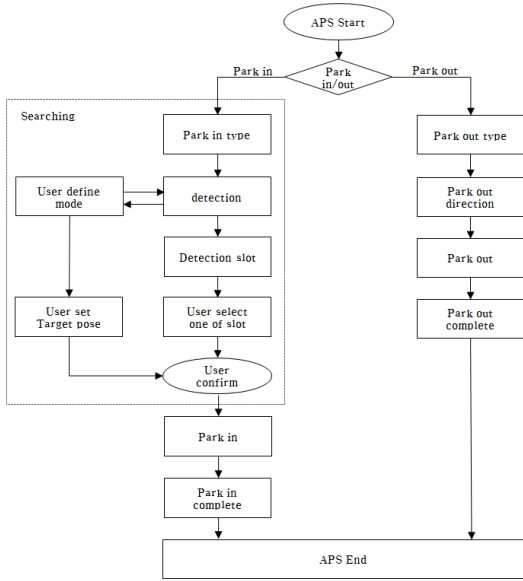


그림 22. APS 동작 시나리오
Fig. 22. APS operation scenario

종료한다. 또한 검출 기능과 주차 진입과정에서 초음파 센서 모듈에서 장애물을 검출 할 경우, 차량을 멈추고 기어를 “P”로 변경한 후 EPB (Emergency Parking Braking)기능을 수행하여 주차 과정을 종료한다. 주차 공간을 찾는 과정이 필요 없는 경우에는 운전자가 AVM 뷰에 나타나 있는 주차 가능 공간을 선택한다. 운전자가 주차 공간을 확정 한 이후, 주차 기능을 실시한 이후 종료한다.

그림 23은 통합한 시스템의 주차 정확도를 평가 방법을 나타낸다.

$(\Delta X_{fl}, \Delta Y_{fl})$ 는 실제 차량이 주차를 완료하였을 때 앞쪽 왼 바퀴에 대한 목표로 하는 주차위치에 대한 차이를 나타내고, $(\Delta X_{rl}, \Delta Y_{rl})$ 는 뒤쪽 왼 바퀴의 차이를 나타낸다. L 은 Wheelbase를 나타낸다. 주차 정확도에 대한 평가 방법은 차량의 주차 목표 지점 대비 실제 차량의 마지막 위치의 차이로 판단하였다. 계산

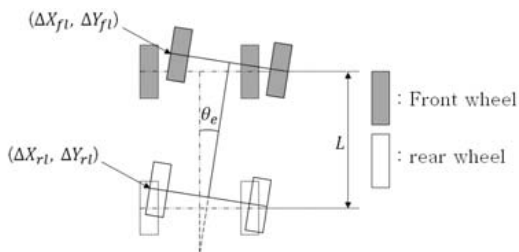


그림 23. 주차 정확도를 평가 방법
Fig. 23. a method for evaluating the parking accuracy

방법은 아래와 같은 수식을 사용하였다.

$$\Delta X = \frac{\Delta X_{fl} + \Delta X_{rl}}{2} \quad (7)$$

$$\Delta Y = \frac{\Delta Y_{fl} + \Delta Y_{rl}}{2} \quad (8)$$

$$\theta = \tan^{-1} \left(\frac{\Delta Y_{fl} - \Delta Y_{rl}}{L + \Delta X_{fl} - \Delta X_{rl}} \right) \quad (9)$$

표 1은 실제 차량을 활용하여 구현한 통합시스템의 수직/수평/사선 주차에 대한 측정 결과이다. 여기서 x 방향의 최대 오차는 0.14m, y 방향의 최대 오차는 0.13m, 최대 각도 오차는 1.49°로 측정되었다.

표 2는 기존 초음파 처리 ECU와 AVM ECU와 APS ECU가 각각 분리된 시스템의 무게와 주요 소자의 사용 개수를 나타낸다. 통합 시스템의 무게는 기존 각각 분리된 시스템의 무게 대비 400g 줄었으며, 주요 사용부품 관점에서는 애플리케이션 프로세서와

표 1. 자동 주차 종류에 따른 실제 차량 시험 결과
Table 1. Actual vehicle test results according to the type of automatic parking

Parking Type	ΔX	Front [m]	Rear [m]	Error [m]	θ [degree]
	ΔY				
Perpen-dicular	ΔX	0.12	0.17	0.14	1.49
	ΔY	-0.16	-0.23	-0.2	
Parallel	ΔX	-0.3	-0.29	-0.3	0
	ΔY	0.13	0.13	0.13	
Angled	ΔX	-0.31	-0.28	-0.3	0.5
	ΔY	0.13	0.12	0.12	

표 2. 분리된 시스템과 통합시스템의 무게 및 주요소자 비교
Table 2. Comparison of weight and major components of isolated and integrated system

		Separated system	Integrated system
System weight	Ultrasonic processing ECU	180g	410g
	AVM ECU	380g	
	APS ECU	250g	
Total weight of system		810g	410g
Number of major component	Application processor	2 개	1 개
	MICOM	3 개	1 개
	Memory	4 개	4 개

MICOM 사용 개수를 줄일 수 있어 시스템 가격을 줄일 수 있다.

VI. 결 론

본 논문에서는 광각 카메라를 사용하는 AVM과 APS의 통합 시스템 구현에 대해 검토하였다. 통합 시스템에서 사용한 센서인 광각 카메라 모듈의 구조와 렌즈 왜곡에 대한 왜곡보정과 초음파 센서의 거리측정 원리를 확인하였다.

AVM 시스템은 광각 카메라 이미지를 활용하여 보다 자연스러운 이미지를 운전자에게 제공하는 3D 뷰 생성 방법 및 블렌딩, 셰이딩, PA를 적용한 이미지를 구현하였으며, 차량 장착과정에서 발생하는 캘리브레이션 과정을 살펴보았다.

APS 시스템은 AVM 영상 및 초음파 센서를 활용하여 운전자에게 자동 주차 기능을 제공한다. 본 논문에서는 주차하고자 하는 위치에 대한 고정된 물체를 찾는 기능을 바탕으로 주차 공간 탐색, 주차 계획, 주차 추종 기능을 구현하였다. 이를 바탕으로 본 논문에서는 AVM 시스템과 APS 시스템 각각의 시스템을 하나의 시스템으로 통합 구현하였다. 여기서 제공하는 다양한 방안들은 운전자 보조시스템은 물론이고 무인 자동차에도 널리 적용될 수 있을 것으로 본다.

References

[1] Wikimedia, *Autonomous car*, Retrieved Oct. 22, 2017, from <http://en.wikipedia.org>

[2] HYUNDAI MOTOR COMPANY, *Advanced Driver Assistance System*, Oct. 22, 2017, Retrieved from <http://brand.hyundai.com/en/brand/technology/adas.do>

[3] J. D. Choi, "Development of spatial awareness and autonomous driving technology for automatic valet parking," *ETRI J.*, 2013.

[4] Wikimedia, *Automated parking system*, Retrieved Oct. 22, 2017, from <http://en.wikipedia.org>

[5] P. Hart, N. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths in graphs," *IEEE Trans. Systems Sci. and Cybernetics*, vol. SSC-4, no. 2, pp. 100-107, 1968.

[6] B. Zhang, et al., "A surround view camera solution for embedded systems," *2014 IEEE*

Conf. CVPRW, pp. 676-681, Jun. 2014.

[7] Wikimedia, *A*_search_algorithm*, Retrieved Oct. 22, 2017, from <http://en.wikipedia.org>

[8] omniVision, *Automotive Companion Chip*, Retrieved Oct. 22, from <http://www.ovt.com>

[9] Wikimedia, *Image processor*, Retrieved Oct. 22, 2017, from <http://en.wikipedia.org>

[10] J. Weng, P. Cohen, and M. Herniou, "Camera calibration with distortion models and accuracy evaluation," *IEEE Trans. Pattern Anal. and Machine Intell.*, vol. 14, no. 10, pp. 965-980, 1992.

[11] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2003.

[12] J. R. Villar, *Advanced shaders*, Retrieved Oct. 22, 2017, from <http://opencv.org>

[13] omniVision, *OmniVision_OV493*, Retrieved Oct. 22, 2017, from <http://en.wikipedia.org>

[14] Wikimedia, *Alpha_compositing*, Retrieved Oct. 22, 2017, from <http://en.wikipedia.org>

[15] D. Scaramuzza, A. Martinelli, and R. Siegwar, "A flexible technique for accurate omnidirectional camera calibration and structure from motion," in *Proc. IEEE Int. Conf. Vision Syst.*, p. 45, Jan. 2006.

[16] Wikimedia, *Automatic parking*, Retrieved Oct. 22, 2017, from <http://en.wikipedia.org>

[17] Wikimedia, *Lexus*, Retrieved Oct. 22, 2017, from <http://en.wikipedia.org>

[18] S. J. Han and J. D. Choi, "Parking space recognition for autonomous valet parking using height-grid map and salient-line probability map," *ETRI J.*, vol. 37, no. 6, Dec. 2015.

[19] Ministry of Land, Infrastructure and Transport regulations No. 382, *Enforcement Rule of the Parking Lot*, 30. Dec. 2016.

[20] ISO, *ISO 16787*, Retrieved Oct. 22, 2017, from <http://www.iso.org>

[21] J. A. Reeds and L. A. Shepp, "Optimal paths for a car that goes both forwards and backwards," *Pacific J. Math.*, vol. 145, no. 2, pp. 367-393, 1990.

[22] G. G Kim, J. W. Yoon, and B. G. Kim, "Estimation of the steering angle based on

extended kalman filter,” *Int. J. Multimedia and Ubiquitous Eng.*, vol. 11. no. 12, pp. 295-306, 2016.

홍 재 근 (Jae-Geun Hong)



1999년 2월 : 금오공대 전자통신공학과 (학사)
2001년 2월 : 금오공대 전자통신공학과 (석사)
2001년~현재 : LG전자 책임연구원
2011년 9월~현재 : 금오공대 전자공학과 박사과정

<관심분야> FPD, Access Network, 3D, ADAS

정 해 (Hae Chung)



1987년 2월 : 한양대학교 전자통신공학과 (학사)
1991년 2월 : 한국과학기술원 전기 및 전자공학과 (석사)
1996년 2월 : 한국과학기술원 전기 및 전자공학과 (박사)
1995년~1998년 : LG정보통신 (현 LG전자) 선임연구원

1998년 8월~현재 : 금오공대 전자공학부 정교수
2004년 1월~2005년 1월 : University of Texas at Dallas 방문교수

<관심분야> 이동통신, 가입자액세스망, 스마트 디바이스, ADAS