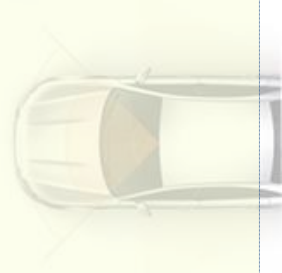


영상인식 기술기반 FCW 임베디드 시스템 기술



FCW(Forward Collision Warning)

박상현 책임연구원

소프트웨어디바이스연구센터

전자부품연구원

KETI

Korea Electronics Technology Institute

Contents



자동차 안전기술 동향

영상 인식 기술 동향 및 시장규모

보유기술 소개(LDWS, FCW)

Embedded Board 포팅 (FCW)

활용방안

지능형 자동차 안전기술 동향



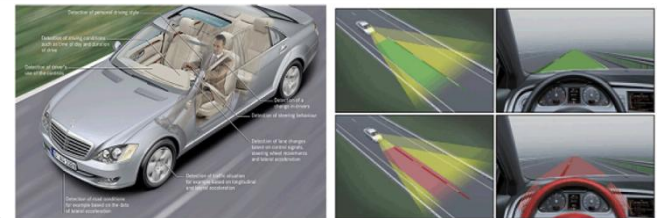
지능형 자동차 안전 시스템

- 정보제공 및 경고, 사고회피, 추돌 피해 경감, 충돌 후 피해 경감, 피해확대 방지



[국외 개발 동향]

- BMW, 벤츠, GM, 포드, 도요타, 폭스바겐 등 글로벌 자동차 업체
→ 2020년까지 충돌을 100% 방지하는 지능형 안전자동차 개발 목표로 함
- 기술 상용화를 위해 IT 업체들과 제휴중
- 초음파, 적외선, 위치센서 등 다양한 IT 기술을 신차에 적용하여 교통사고의 위험을 줄이기 위해 노력 중



[국내 개발 동향]

- 차세대 자동차 기술개발에서 기존 기계공학 위주의 기능 개발 및 신기술 개발이 주를 이루고 있어, 소프트웨어를 중심으로 한 IT 기술 적용은 아직 부족한 편
- 졸음경보시스템, 전방장애물 충돌방지 시스템, 전방감지 카메라, 후방감지 카메라, 전방충돌 방지 지원장치, 차량간격 자동제어 시스템 등 여러 기술이 개발되어 상용화 진행

지능형 자동차 안전기술 동향



GM

- 2010년 상하이 엑스포에서 차량간 교신과 거리 측정 센서를 결합한 보행자 보호 시스템 공개
- 보행자의 휴대폰 신호를 감지하여 일정 거리 내로 사람이 접근하면 주행속도를 줄이고, 카메라를 통한 보행자 정보를 파악하여 충돌 회피

포드

- 위성항법장치와 와이파이 신호를 감지하면서 다른 차량의 이동 속도를 파악하여 사고를 방지하는 차량 간 충돌사고 예방 시스템 연구

벤츠

- 2011년 액티브 브레이크 어시스트 2(ABS2)를 개발하여 적용
- 차량에 설치된 3개의 레이더 빔을 통해 전방 100~200m 지점까지 감지하고, 기후화 조명 조건에 영향을 받지 않으며, 시속 15km 부터 89km 까지의 속도 전 구간에서 작동

아우디

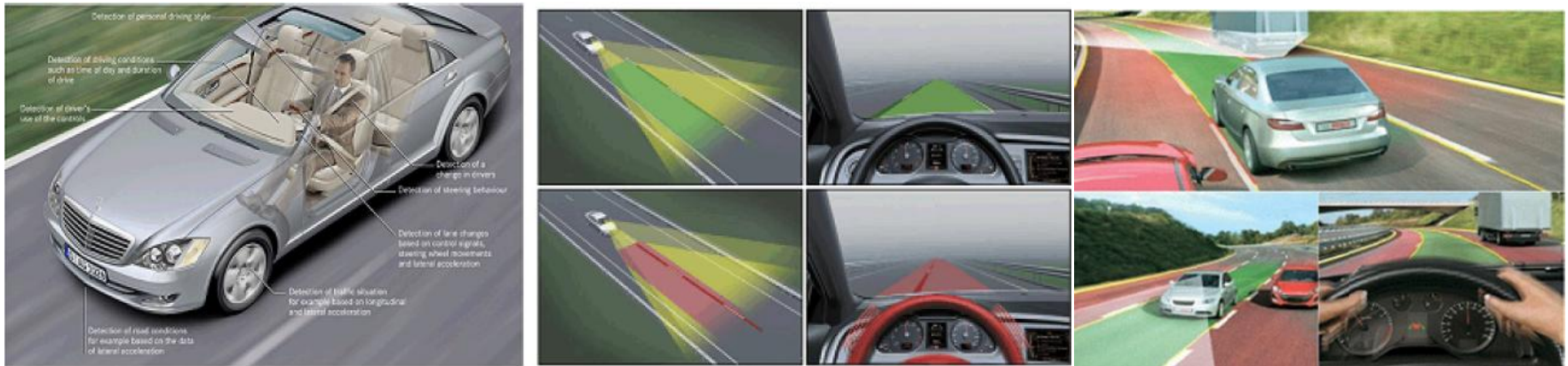
- 차선 이탈방지 경고 시스템이 'A8'에 탑재되어 시속 65km이상 주행 시 경고 메시지를 보내며, 차선이탈 시 반응하는 핸들 진동의 정도를 직접 조절 가능

지능형 자동차 안전기술 동향



BMW

→ 차선유지 지원시스템(Lane Keeping Support System:LKSS)은 자동차가 차선을 벗어나면 스스로 핸들을 돌려주는 시스템으로 5 시리즈에 장착



볼보

→ 보행자 추돌방지 시스템이 세계 최초로 개발되어 'S60'에 적용됨

→ 전방 차량과의 거리가 급격히 줄어 충돌이 위험하다고 감지되면, 브레이크 작동

지능형 자동차 안전기술 동향



Nissan

- 차선이탈 방지시스템은 인피니티 M 모델에 장착
- 운전자에게 경고를 주는 것은 물론, 벗어나는 방향의 반대쪽 바퀴의 브레이크 압력을 조절하여 자연스럽게 원래의 차선으로 복귀 시켜줌
- 시속 60Km이상에서 레이더 센서를 통해 앞 차와의 거리를 측정하고 알람 발생 및 차량 제어



토요타

- 사고안전 확보시스템은 레이더와 카메라로 장애물을 인식하며, 렉서스에 적용중
- 도로 위 물체를 인식하며, 보행자를 인식하여 충돌 직전 차를 정지 시키는 안전 기능

지능형 자동차 안전기술 동향



현대 & 기아

- 차간거리 제어시스템, 자율주행, 차선이탈 경보-방지 시스템, 지능형속도제한 시스템 등 개발 중
- 차간 거리 시스템 : 레이더로 앞차와의 거리를 측정하고, 일정한 거리를 유지시켜주는 지능형 크루즈 컨트롤 시스템과 서행제어 시스템 기술이 포함
- 스마트 순항 제어(ASCC)는 신형 그랜저에 적용, 운전자가 설정한 속도를 자동으로 유지 및 앞차와의 거리 유지
- 차선이탈경보시스템(LDWS)은 차량 스스로 차선을 인식하여 운전자가 차선을 넘으면 경고를 하는 시스템



ASCC



차선이탈경보시스템

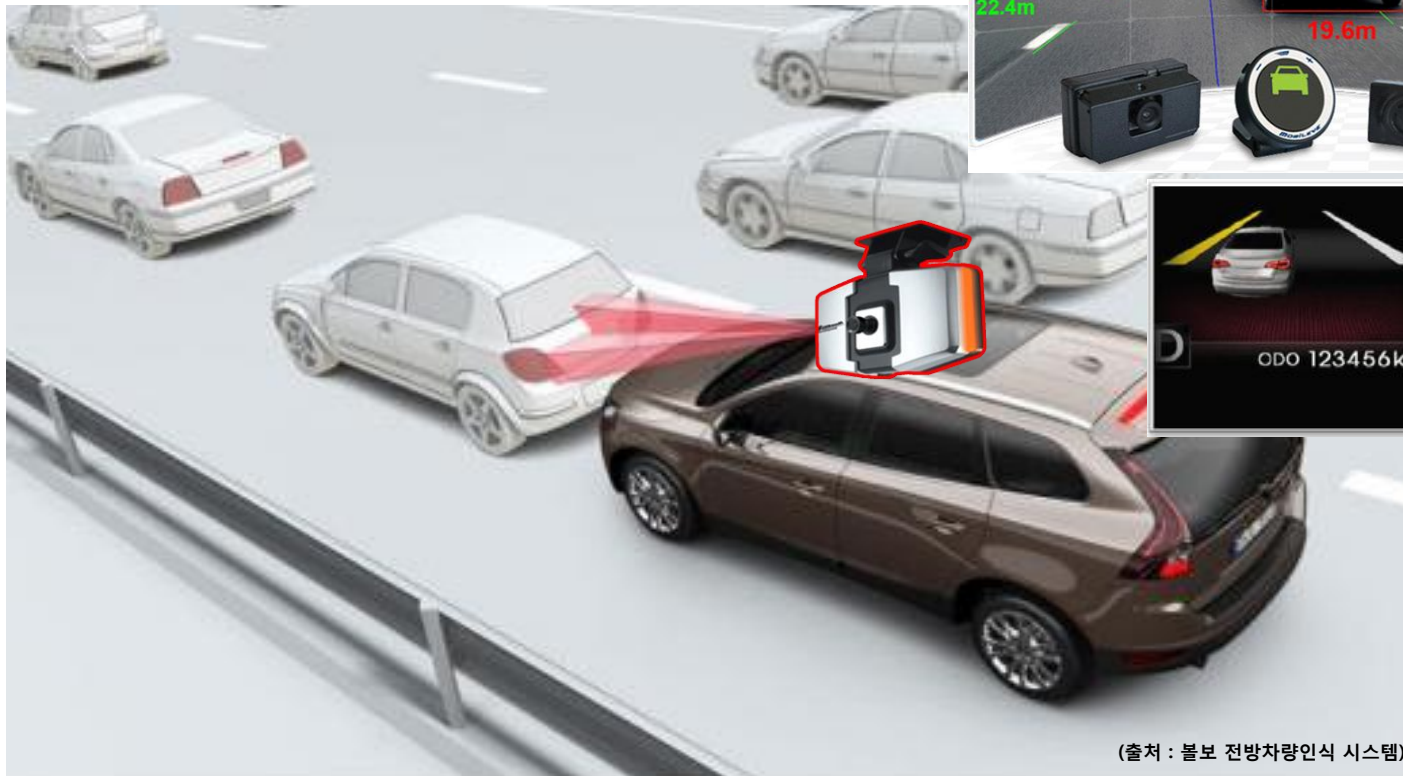
현대모비스 및 만도

- 차간거리 제어장치(SCC), 차선유지 도움장치(LKAS)
- 차선이탈경보, 레이더, 전방차량인식 등 안전장치 개발중

영상인식 기술 기반 LDWS, FCW



- 차량 내 설치된 카메라를 이용하는 지능형 영상인식 기술
- LDWS(Lane Departure Warning System)
- FCW(Forward Collision Warning) → ISO-15623규격문서



(출처 : 볼보 전방차량인식 시스템)

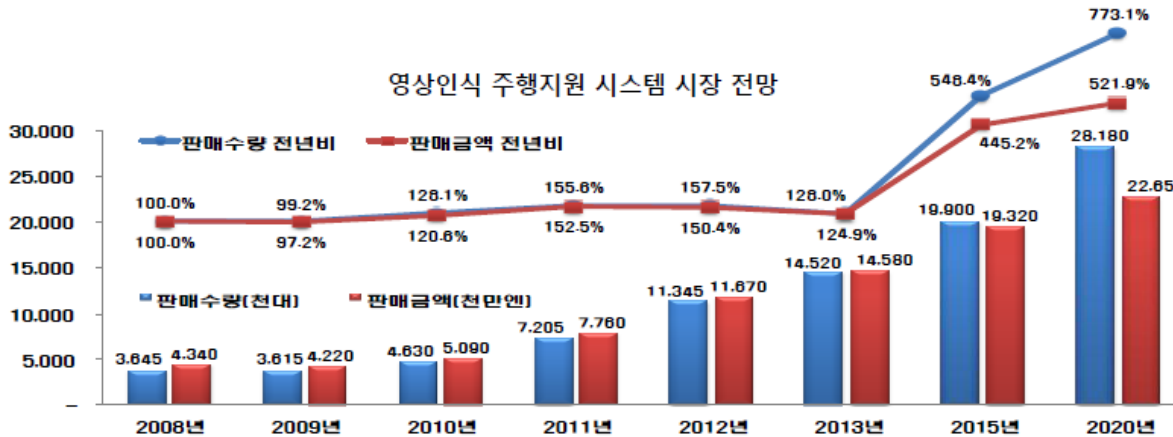
Embedded Vision

- Machine Vision – 영상에서의 특징을 추출 후 학습모델을 이용한 차량 검출 방법
- 차량 내 설치를 위한 임베디드 시스템 개발 기술

자동차 영상 인식 시장 규모



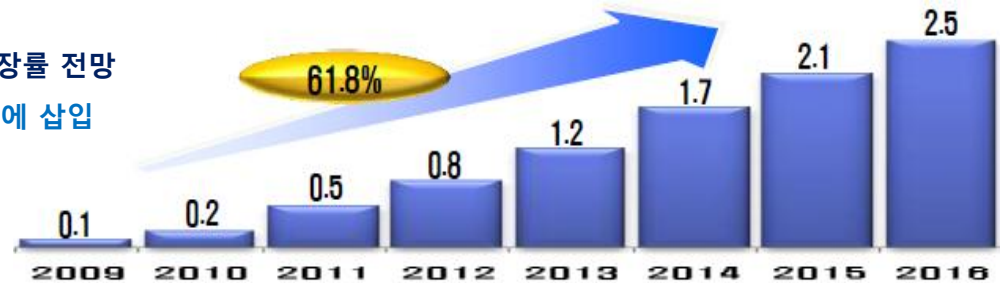
- LDWS, FCWS, PCWS, AVM 등 자동차용 카메라 시스템의 '14년 세계 시장 규모는 약 2조원에 달할 것으로 전망



참고 : Fuji Chimera Research Institute, Inc. 차재전장 디바이스 & 컴포넌트 Selected 2010

북미 LDWS 시장 규모

- '09년 118억원, '16년 2,961억원, 평균 61.8% 성장률 전망
- US NCAP : 2011년 LDWS, FCWS 안전평가 기준에 삽입



유럽 LDWS 시장 규모

- '09년 219억원, '16년 1,289억원, 평균 28.6% 성장률 전망
- 2013년 신규 상용차에 LDWS장착 의무화 추진

• 참고 : Frost&Sullivan 2009
 • 판매금액 : \$103, 1\$= ₩ 1,150

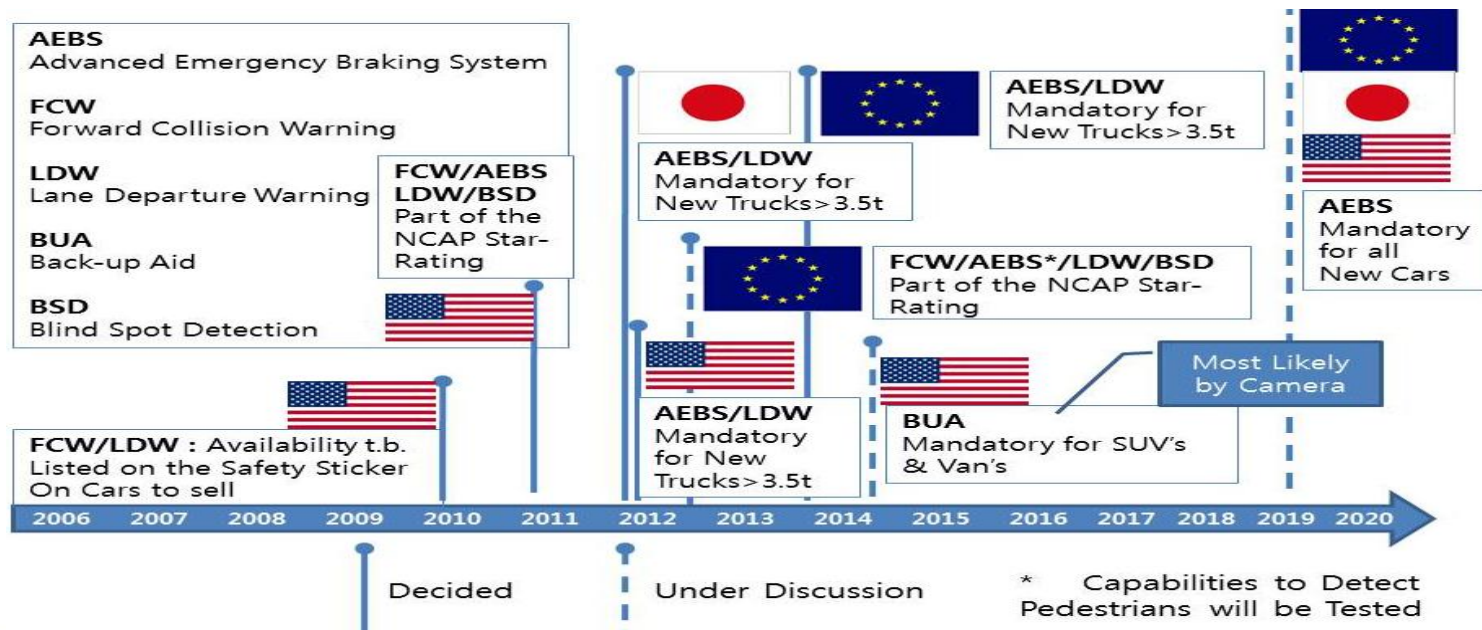
국내외 기술동향



기술 동향

- 유럽에서는 2013.11.부터 출시되는 상용차 신모델부터 단계적으로 LDSW와 AEBS가 의무 장착
- 영상기반 전방 장애물 인식 시스템, 사고예방 시스템 등의 법제화가 추진중
- 미주, 유럽, 일본지역은 2019년까지 모든 차량에 LDW, FCW, 후방카메라, 운행기록장치 등을 의무 장착하도록 규정
- 국내의 경우 2015년까지 상용차량의 LDWS 의무장착이 법제화로 추진되고 있음

▶ 국가별 법제화 현황



LDWS 시장동향



- 2001, Nissan Cima
- 2002, Toyota Crown Majesta, Lexus LS460
- 2003, Honda Inspire
- 2005, Nissan Infinity FX, Citron C4, C5, C6, Audi Q7



제조사	차량 모델	출시년도	제품	사진
GM	Cadillac CTS, DTS Buick Lucerne	2007	차선이탈 경고 시스템	
BMW	5, 6 Series	2007	차선이탈 경고 시스템	
Volvo	S80, V70, XC70	2007	차선이탈 경고 시스템	
Benz	E-Class	2009	차선이탈 경고 시스템	
Hyundai	EQUUS	2009	차선이탈 경고 시스템	
Hyundai	GENESIS	2009	차선이탈 경고 시스템	
Kia	K7	2009	차선이탈 경고 시스템	

현재, 중형급 이상 차량에 장착

FCW 시장동향



- 전방차량과의 거리를 인식하여 전방차량과 충돌 예상 시간을 표기
추돌 경고 및 제어 단계 : 경고 → 탑승객 보호(안전벨트 강화) → 브레이크 보조 → 부분 or 완전 제동
- Volvo에서만 카메라 기술을 적용(그 외, Radar-Camera 퓨전 타입)

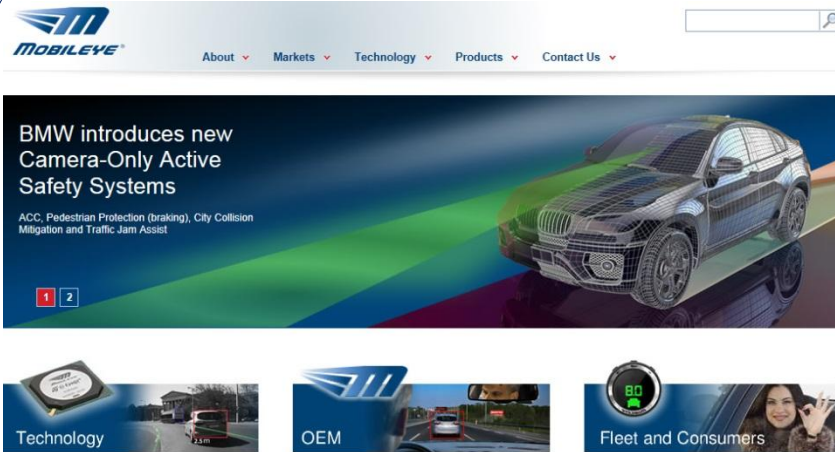
제조사	차량 모델	출시년도	제품	적용센서	주요기능	사진
Audi	A8	2010	사전 감지 및 제동	Rader	경고→안전벨트 강화 →제동	
Ford	Lincoln MKS, Taurus	2009	제동 보조를 겸한 추돌 경고	Rader	경고→제동보조	
Honda	Acura	2003	추돌 지연 제동 시스템	Rader	경고→안전벨트 강화 →제동	
Nissan	Infiniti	2006		Laser		
Benz	S-Class	2005	사전 안전 제동	Rader	경고→부분 제동	
Toyota	Lexus	2003	추돌 예방 시스템	Rader, Infrared	경고→안전보조→제동	
Volvo	S80	2007	자동 제동을 겸한 추돌 경보	Rader, Camera	경고, 제동 보조	

“상용차량 및 고급 차량에 장착”

차량 영상 인식 관련 기술 동향



국 외 - 대표기업



- 모빌아이(이스라엘)
ISO 9001:2008 certified
- Mobileye 5-Series, C2-Ceries
- 차선인식, 전방차량인식, 보행자인식, 표지판인식,
지능형 하이빔, 이륜차 인식
- BMW, VOLVO, Cadillac, Toyota 등 BM 및 AM
마켓을 통한 시장 확대
- 차량 영상 인식 시장의 선두 기업



국 내 - 대표기업



- PLK technologies
- 주요제품 : ADAS series
- 차선인식, 하이빔 어시스트
- 전방차량인식, 보행자인식, 표지판인식 사업화 진행
- 국내 현대 & 기아 자동차 협력사

1 st Generation A	1 st Generation B	1 st Generation D	2 nd Generation
LDW	LDW	LDW + HBA	LDW + HBA + FCW
2006 ~ NOW	2009 ~ NOW	2010 ~ NOW	2013

차량 영상 인식 관련 기술 동향



● ADAS(Advanced Driver Assistant System) 영상 인식 분야

단일 카메라 이용 Multi-Application 탑재

단순인식에서 제어 결합용으로 발전(LDW→LKA 등)

■ BOSCH Multi Purpose Camera

단일 H/W 기반 기능통합 영상 인식 S/W 및 주행 시스템 개발 중
(HBA, LAK, FCW, RSR)

■ HELLA Scalable Multi Function Camera

단일 H/W 기반 기능통합 영상 인식 S/W 및 주행 시스템 개발 중
(LDWS, HBA, FCS, NV, TSR)

■ Valeo

- LaneVue™ : 차선이탈 경보 기능 양산
- BeamAtic™ : HBA, 안개등 제어 기능 양산
- XtraVue™ : 나이트비전 기능 개발 중

BOSCH

Vision Based Driver Assistance

Night Vision

- ✓ NV: Far Beam Sight with Low Beam Light
- ✓ NVplus: NV with Highlighting of Pedestrians

Lane Vision

- ✓ Lane Departure Warning
- ✓ Lane Keeping Support
- ✓ Support ACC and Predictive Safety Functions
- ✓ Driver Alertness Monitoring

Intelligent Headlamp Control

Object Vision

- ✓ Support of Radar Based ACC and Predictive Safety
- ✓ Predictive Emergency Braking
- ✓ ACC Full Speed Range
- ✓ Collision Warning / Mitigation
- ✓ Support Passive Safety

Traffic Sign Vision

- ✓ Speed Limit Assistance
- ✓ Traffic Sign Memory

• Series development
• Development of prototypes

HELLA

Front Camera

High Beam Assistant (HBA)

Dynamic Cut Off Line

Verkehrszeichen-Erkennung (VZE)

Lane Departure Warning (LDW) / Heading Control (HC)

Objekterkennung Über Fusionsansatz

Night Vision I

Night Vision II

Valeo

Software Roadmap

Market Introduction*	2004	2005	2006	2007	2008	2009	2010	2011	2012
Infrastructure Detection		LaneVue™		LaneGuide™		Speed Limit Support (Statoic)			
Lighting Wiping		Beam Atic™							
Night Vision									
Obstacle Detection									

production development concept evaluation

* Market Introduction may vary depending on the targeted level of performance

차량 영상 인식 관련 기술 동향



개발 업체	납품 업체	상세 기능
Mobileye	BMW Volvo Cadillac GM	<ul style="list-style-type: none"> • LDW(Lane Departure Warning), • FCW(Forward Collision Warning) • PCW(Pedestrian Collision Warning) • TSR(Traffic Sign Recognition), BSD(Blind Spot Detection)
Continental	BMW (7 series)	<ul style="list-style-type: none"> • LDW(Lane Departure Warning), FCW(Forward Collision Warning) • IHC(Intelligent Headlamp Control), SLM(Speed Limit Monitoring)
Hella	GM Opel	<ul style="list-style-type: none"> • LDW(Lane Departure Warning), TSR(Traffic Sign Recognition)
Visteon	Nissan, Volkswagen	<ul style="list-style-type: none"> • LDW(Lane Departure Warning), FCW(Forward Collision Warning) • BSD(Blind Spot Detection), Map-based Curve Speed Warning • Driver Monitoring System
Bosch	Ford, Audi Peugeot, Renault Mercedes	<ul style="list-style-type: none"> • ACC(Adaptive Cruise Control) with FCW • Parking Aid System, Night Vision
Autoliv	Audi, Fiat, Jaguar, Seat, Toyota	<ul style="list-style-type: none"> • Night Vision System with PCW
HCL		<ul style="list-style-type: none"> • Rear View Aid System with 4 cameras • TSR(Traffic Sign Recognition), AHC(Auto Highbeam Controller) • License Plate Recognition

보유기술 소개

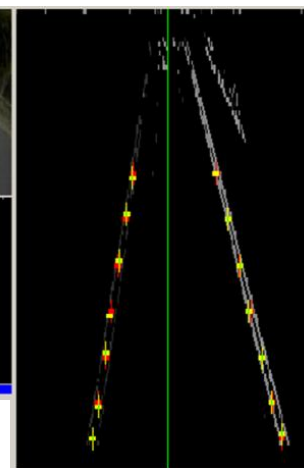
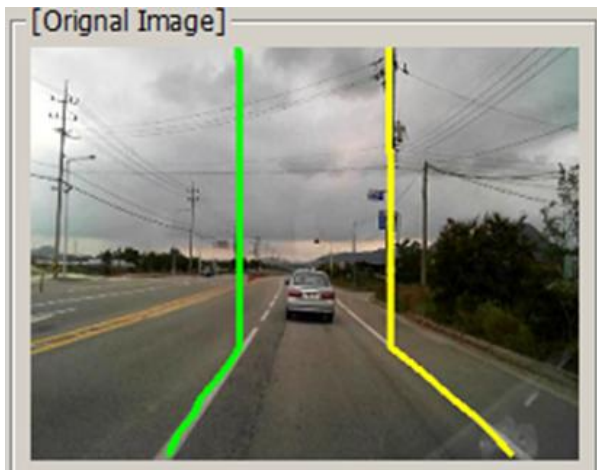
- LDWS(Lane Departure Warning System)
- FCW(Forward Collision Warning)
- Embedded System
- 활용기술 분야





- LDWS(Lane Departure Warning System)

- 운전자가 좌/우측 방향 등을 켜지 않고 차선을 변경할 경우 운전자에게 경고 제공
- 차량 내 설치된 카메라에서 실시간 영상을 입력 받아 도로 위의 차선을 인식
- 일정 속도 이상에서 차선 이탈 알람 발생

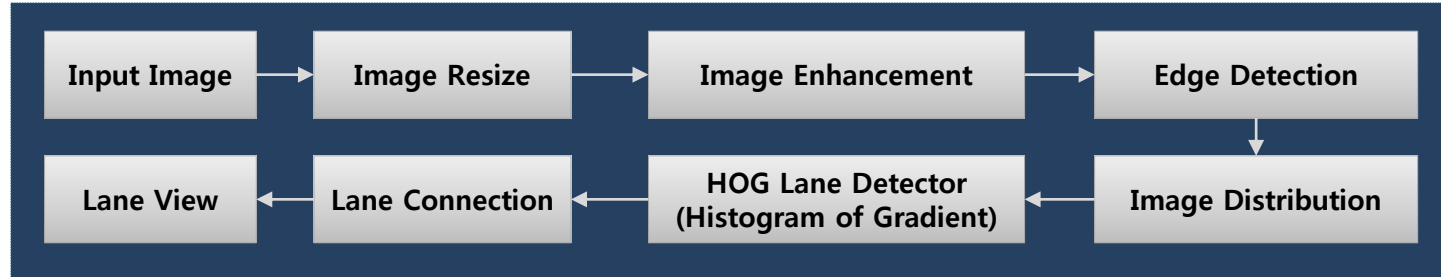


KETI - LDWS



- **LDWS(Lane Departure Warning System)**

- 입력영상을 분할하여 차선 정보를 추출
- 분할된 영역에서의 차선검출 후 차선 영역 확장
- FCW를 위한 ROI 제공



KETI 차선인식 방법 흐름도

- 블랙박스에서 녹화된 동영상을 기반으로 한 차선 인식 알고리즘
- 현재 주로 사용되는 블랙박스 경우 HD급 영상이며, 실시간 영상인식 지원을 위해 입력 영상의 크기 변경



KETI - LDWS



- **Image Resize**

- Bilinear Interpolation 방법을 통한 영상 크기 변경

- **Image Enhancement**

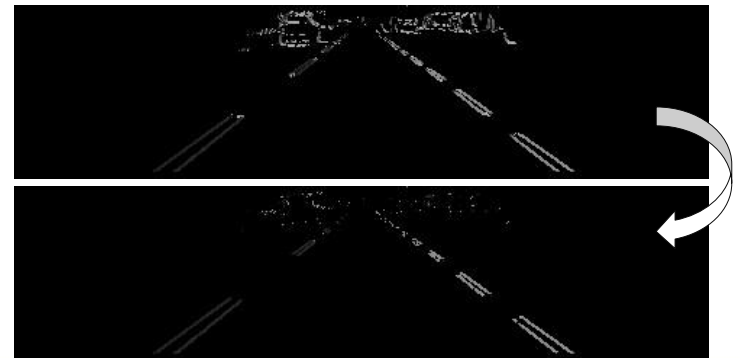
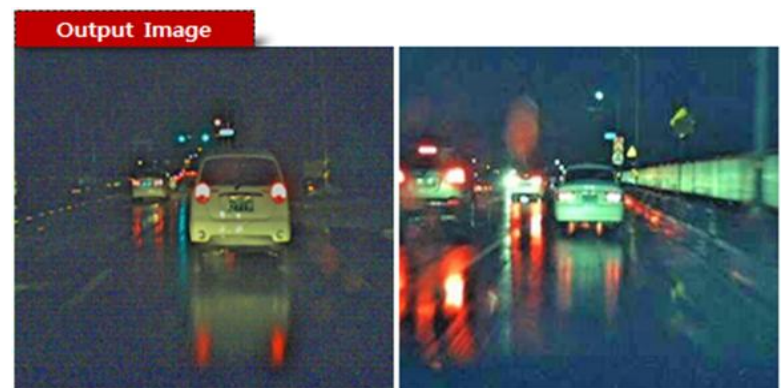
- 실행 속도를 감안한 히스토그램 기반의 영상 개선 방법 적용

- **Edge Detector**

- Sobel Edge Operator를 이용한 영상의 Edge 추출

- 왼쪽 및 오른쪽 차선은 서로 다른 기울기 정보를 가짐

- 차선을 제외한 Edge 제거 필터

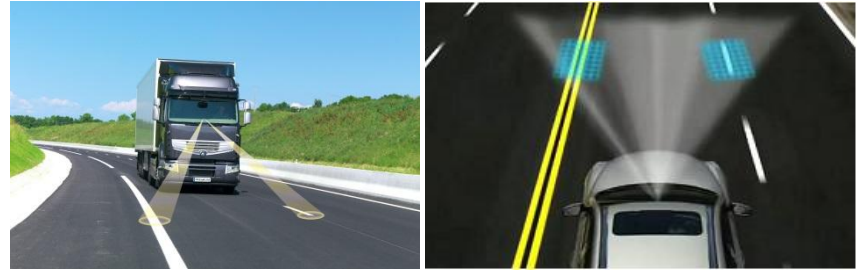


KETI - LDWS



- Image Distribution

 - HOG Lane 추출을 위한 영상 분할



- HOG Lane Detector

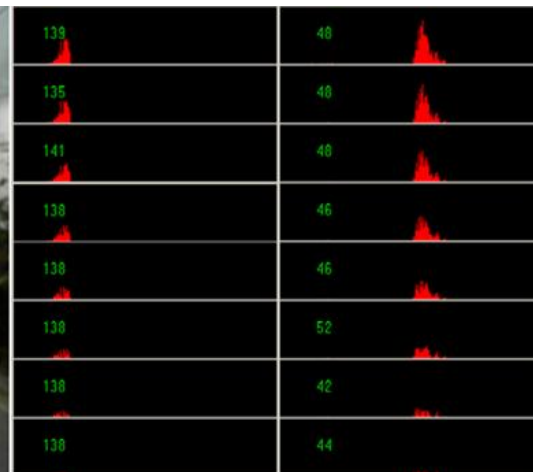
 - 추출된 Edge에서 차선 후보 Edge 성분을 추출

 - 차선 Edge의 경우 일정 거리에 pair edge가 존재

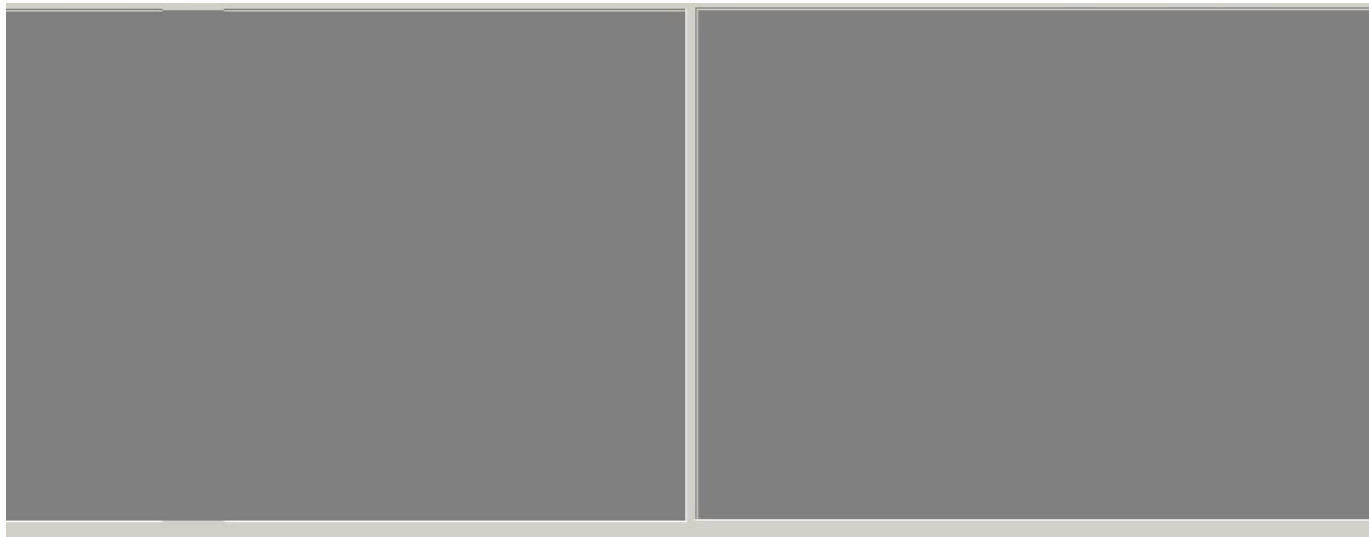
 - 분할된 영상에 따라 Distance Map을 활용하는 차선 추출 방법

- Lane Connection & View

 - 분할된 영역에서의 차선 연결 및 예측(점선 차선의 경우)



DEMO - LDW



RoadLaneDetector

Get_FrontTraffic Info (Start)

[Original Image] [Front Vehicle Detection] [Front Road Lane Detection]

Connect CAM

:: 전자부품연구원 전북ES센터 ::

B_OK B_CANCEL

The interface is a software window titled 'RoadLaneDetector'. It features a top status bar with the text 'Get_FrontTraffic Info (Start)'. Below this, there are three main display panels: '[Original Image]' on the left, '[Front Vehicle Detection]' in the middle, and '[Front Road Lane Detection]' on the right. The left panel includes a volume slider and two buttons with play and pause symbols. At the bottom left is a 'Connect CAM' button. The bottom center contains the text ':: 전자부품연구원 전북ES센터 ::'. The bottom right has two buttons labeled 'B_OK' and 'B_CANCEL'.



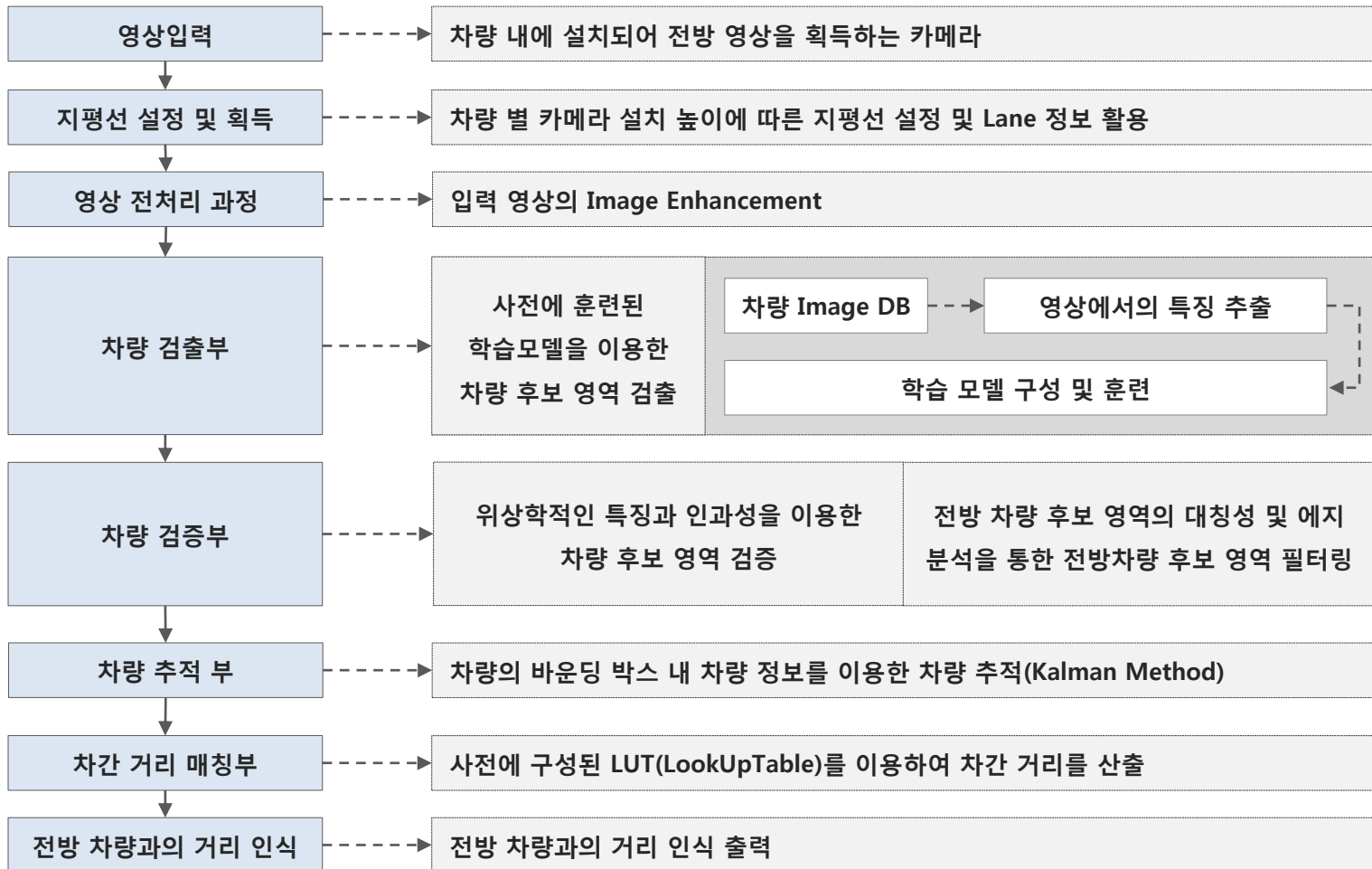
- **FCW(Forward Collision Warning)**

- 전방 차량 추돌 경보 시스템으로 차량이 일정속도 이상으로 주행하는 경우 동작
- 차량 내 설치된 카메라에서 실시간 영상을 입력 받아 전방 차량을 인식하고 전방 차량과의 거리를 산출
- Windows PC 기반 FCW 알고리즘을 임베디드 시스템에 포팅





● FCW(Forward Collision Warning) 알고리즘 전체 흐름도



- 지평선 설정 및 획득

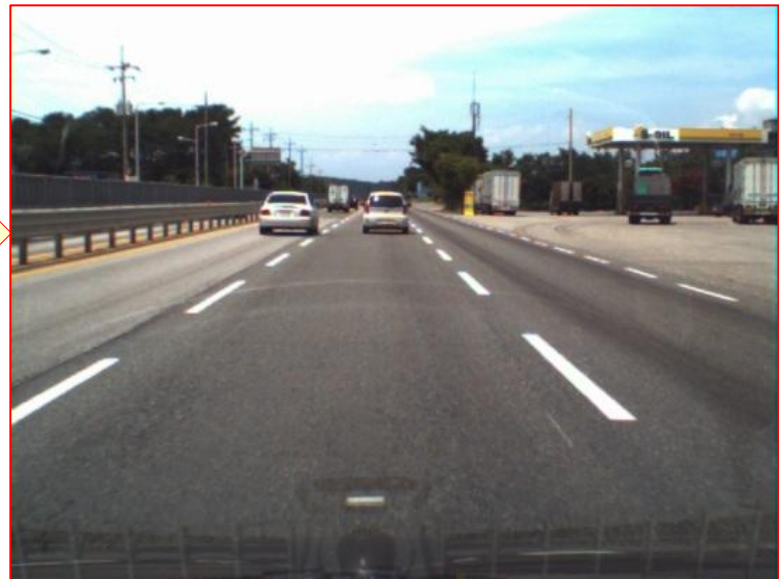
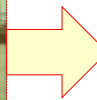
- LDW의 차선 정보 및 Vanishing Point를 FCW 알고리즘에 활용

- LDW의 정보가 없을 경우 사전에 입력된 값을 이용

- 영상 전처리 과정

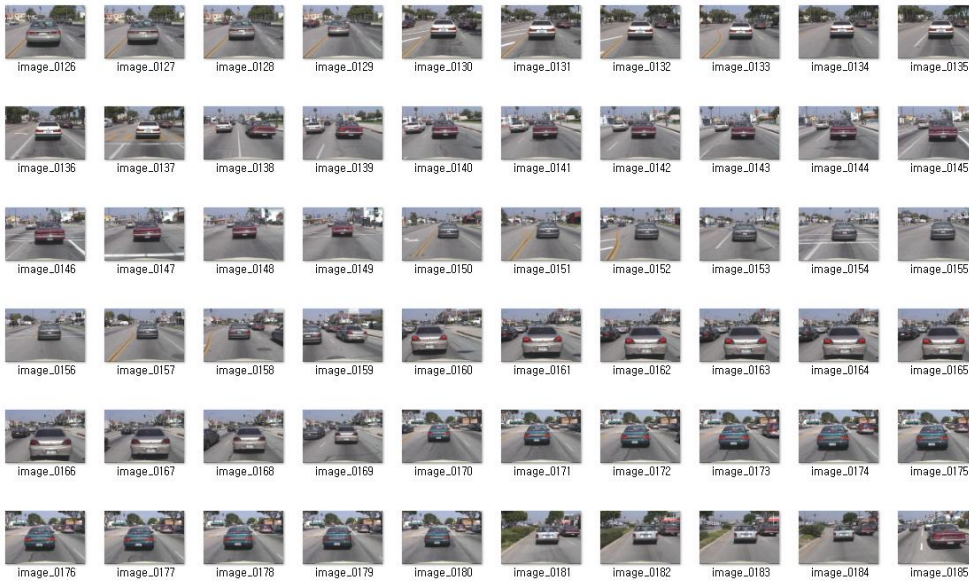
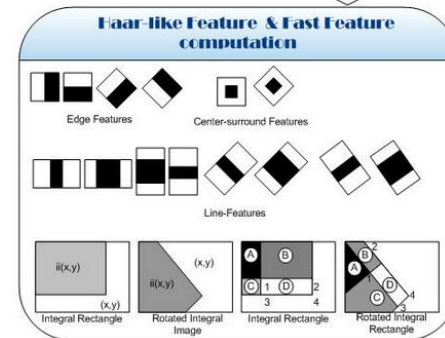
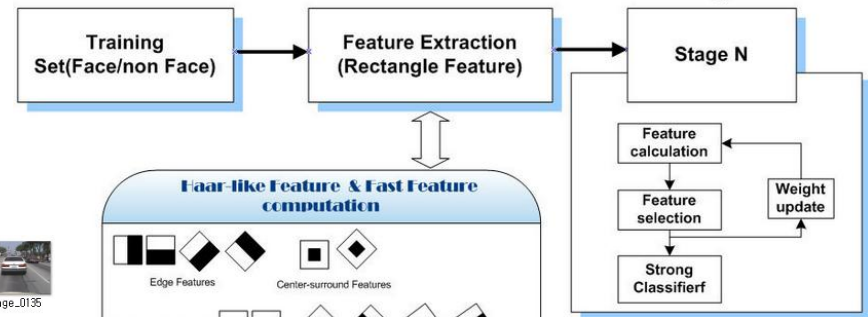
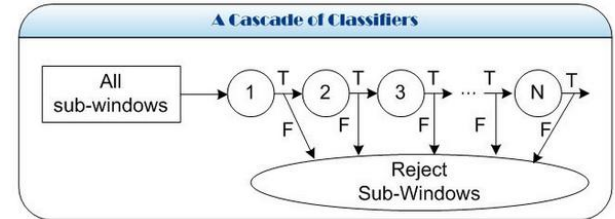
- 입력 영상을 개선하여 영상의 특징점 추출이 용이 하도록 영상을 개선

- 입력 영상의 크기를 고려한 영상 개선 방법 적용



● 차량 검출부

- 차량 DB에서 Harr-like Feature 추출
- Cascade Classifier(Adaboost) 학습 모델을 이용한 차량 검출부 구성
- OpenCV를 이용한 학습모델 훈련
- 훈련된 XML 파일 생성
- 실시간 영상에서의 차량 후보 영역 추출에 사용



[훈련에 사용된 차량 DB]

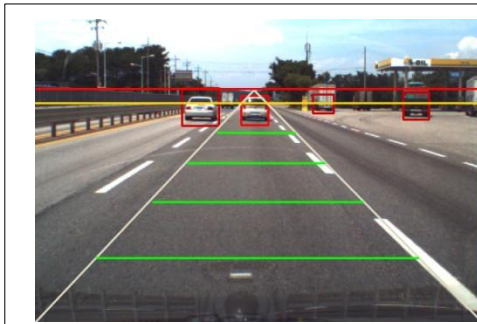
```
<?xml version="1.0"?>
<opencv_storage>
  <!-- Automatically converted from E:\vehicle\train\car\classifier, window size = 70x55 -->
  <- car70x55Cascade15 type_id="opencv-haar-classifier">
    <size> 70 55 </size>
    <stages>
      <- stage 0 -->
      <- trees>
        <- tree 0 -->
        <- root node -->
        <- features>
          <- rects>
            <rect> 40 32 2 2 1.</rect>
            <rect> 40 34 2 2 1.</rect>
          </rects>
          <- features>
            <- tilted> 0</tilted>
            <- feature>
              <threshold> 0.0235 </threshold>
              <left_val> 1.0 </left_val>
              <right_val> 1.1 </right_val>
            </feature>
          </features>
        </tree>
      </trees>
      <stage_threshold> 1.05 </stage_threshold>
      <parent> -1 </parent>
      <next> -1 </next>
    </stages>
  </car70x55Cascade15>
</opencv_storage>
```

0.3028	0.6546	0.6505	0.2804						
0.6872	0.6892	-0.6867	0.2333	0.7363	-0.9004				
0.7108	0.5519	-0.3057	0.1925	0.4008	0.6455				
0.5278	0.4562	-0.3056	-0.9281	0.2965	-0.3134				
-0.5022	-0.7336	0.5133	-0.7919	-0.6666	0.1690	0.2146			
0.5007	-0.7343	-0.3328	0.4000						
0.3154	-0.4614	0.2304	0.2267	0.9124	0.1541	0.2699			
0.4711	-0.2163	-0.8306	-0.4607	-0.9001	0.1959	0.1977	0.1784		
0.5130	0.6908	0.2033	-0.6228	-0.8570	-0.3324	0.6822	-0.1998	0.7019	

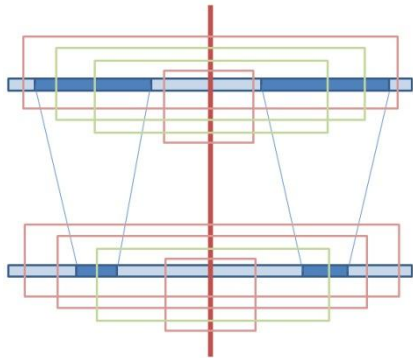


● 차량 검증부

→ 위상학을 이용한 차량 후보 제한



- 붉은선 - 원근법에 의한 소실점과의 접선
- 노란선 - 배경 구분선
- 초록선 - 원근법에 의한 차폭 후보

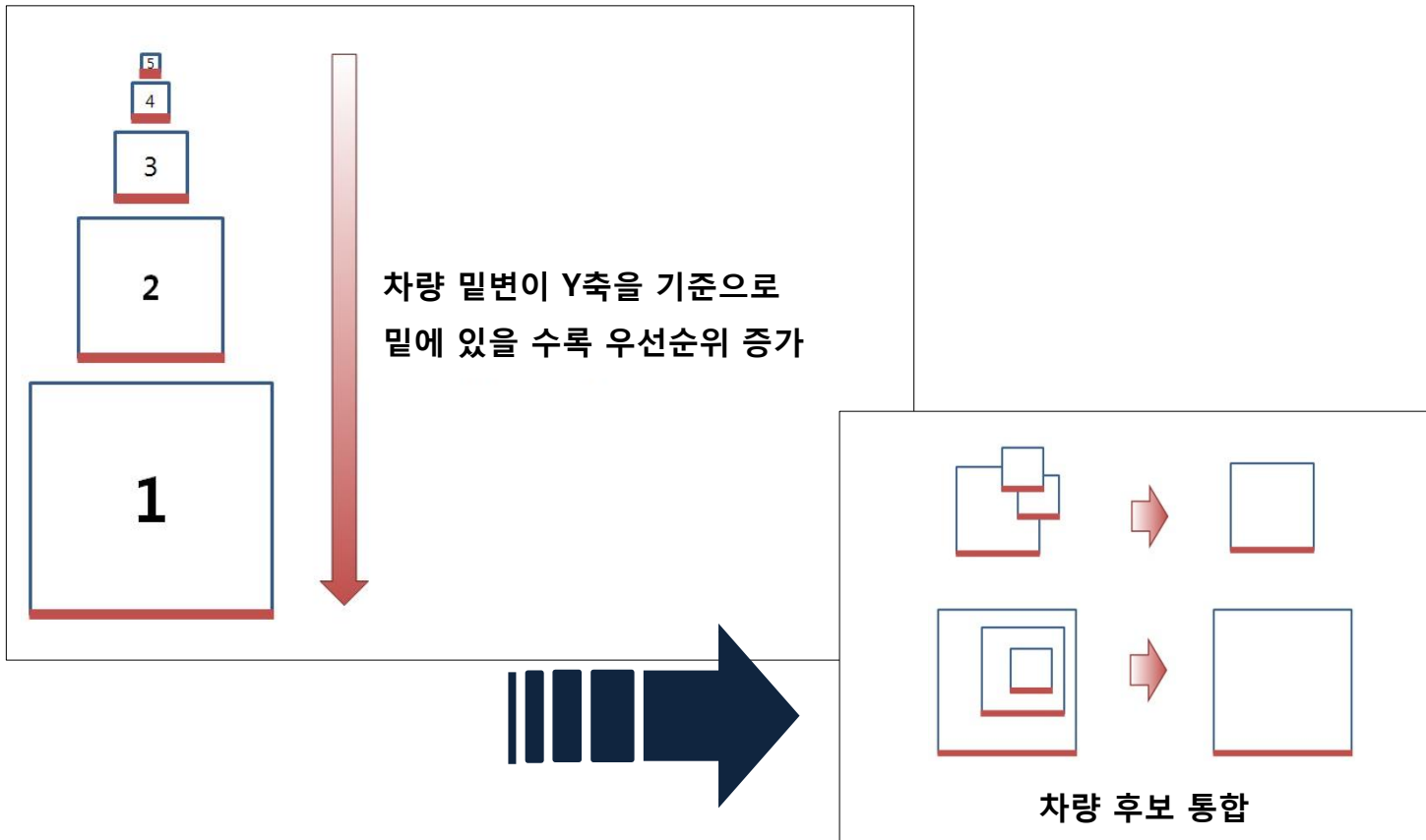


- 입력 허용 차량 너비 : 검출된 차량의 위상학적 허용너비 계산
허용 너비를 만족하지 못한 차량의 경우
차량 후보에서 제외
- 출력 허용 차량 너비 : 입력 너비를 통과한 차량 후보로부터
윤곽 정보를 이용하여 차량 후보 제한

차량 너비는 [-1, 0, 1] filter를 이용한 HOG의 기울기 정보와 Canny filter의 윤곽 정보, Hough를 이용한 누적 Cell단위의 energy정보를 이용하여 추출

- 차량 검증부

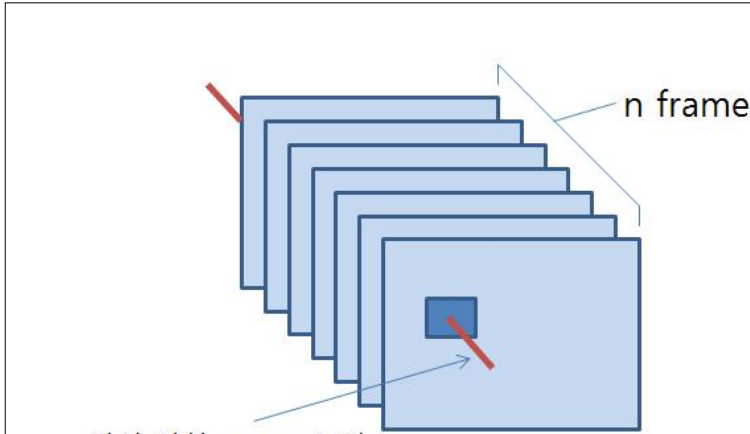
→ 위상학을 이용한 차량 후보 제한





● 차량 검증부

→ 위상학을 이용한 차량 후보 제한



- 현재 frame을 기준으로 후보 영역 내의 각 pixel당 n frame 만큼 누적 Energy 산출
- 후보 영역 내에서 최대 Energy가 임계값을 넘을 경우 인과성 차량
- 후보 영역 내에서 최대 Energy가 임계값을 넘지 못할 경우 비인과성 차량

관심 위치 energy 누적

비인과성 차량 후보 제거

- 이전 frame과의 차량 크기를 비교하여 인과성 조건을 만족시킬 경우 검색 영역 내의 최적의 차량 크기 후보 추출

- 차량 검증부

→ 위상학과 시간정보를 활용하는 차량 검증

- 위상학과 시간정보를 이용하여 차량 후보를 제한한 이후에 다시 형태 기반을 이용하여 인증
- HOG + Canny + Hough를 이용한 형태 기반



(a)



(b)



(c)

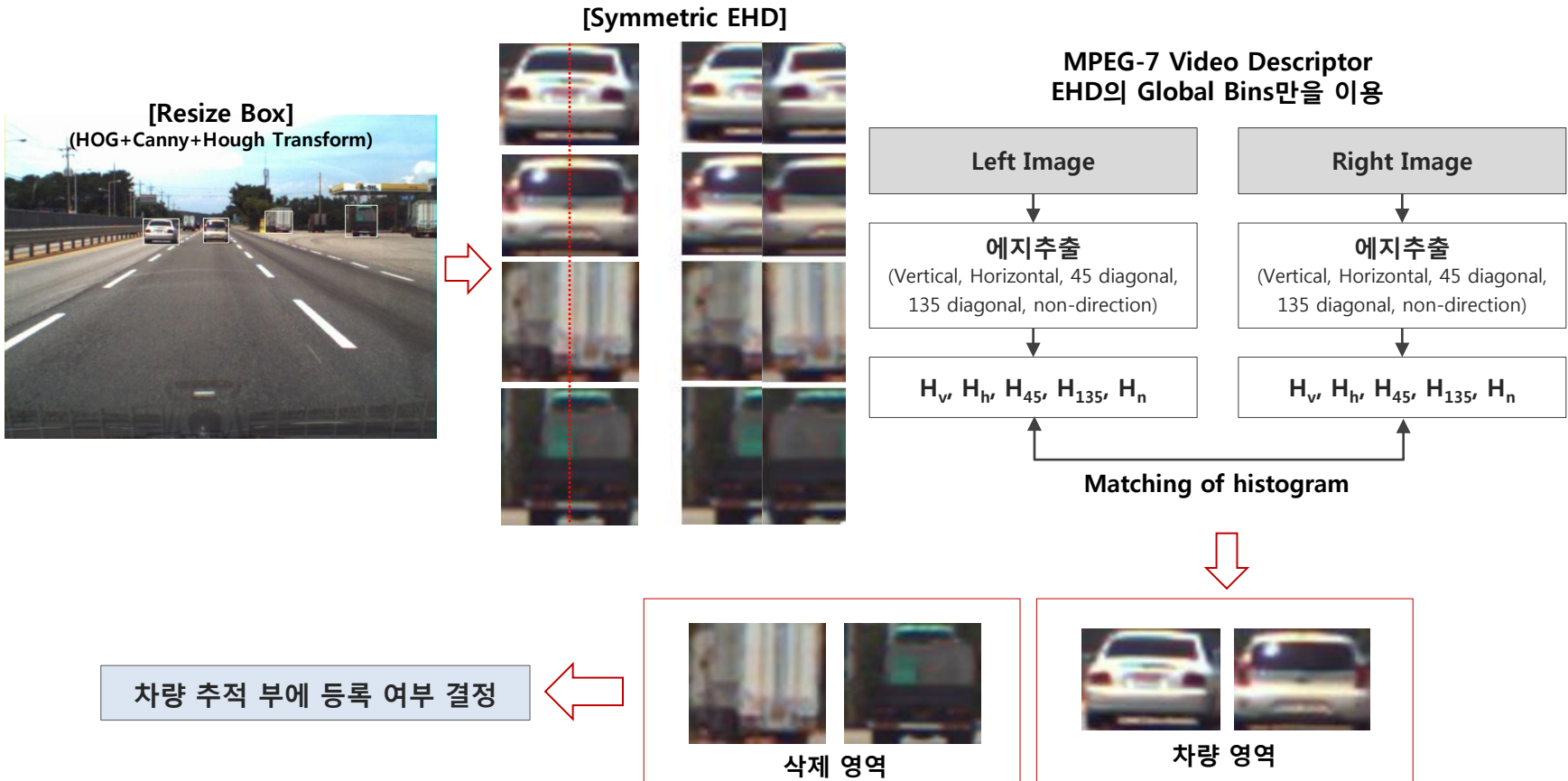
- (a) Sample Image, (b) Sample Image의 수평 성분 추출, (c) Sample Image의 Hough map
- Hough map에 차량 데이터 및 비차량 데이터를 학습하여 차량 데이터는 +(plus), 비차량 데이터는 -(minus) 에너지를 할당하여 Energy map 생성
- 입력된 차량 데이터 Hough map의 총 Energy값이 양수이면 차량, 음수이면 비차량으로 구분



● 차량 검증부

→ 검증된 영역의 차량 대칭성 여부 검증

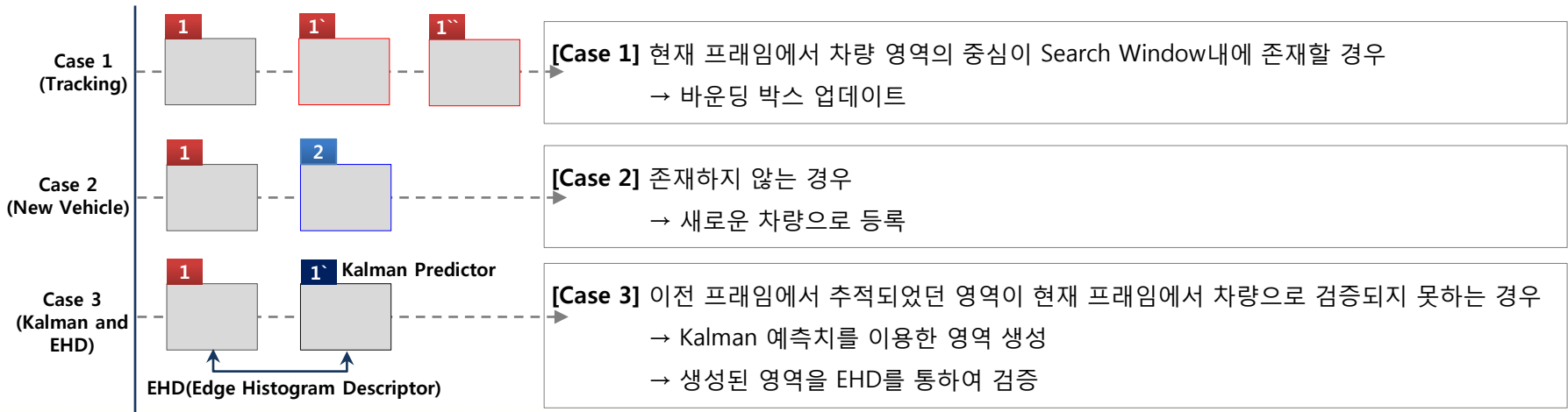
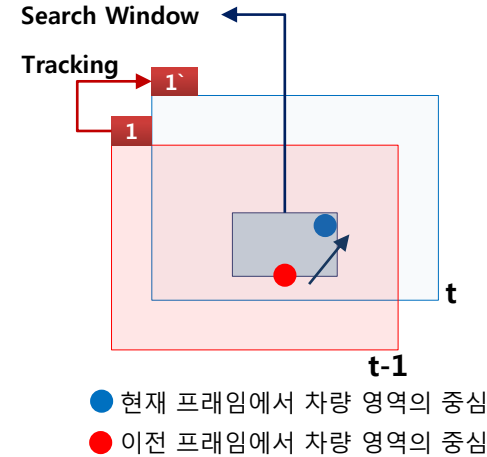
→ EHD(Edge Histogram Descriptor)는 MPEG-7의 Video Descriptor 중의 하나로 사용되고 있는 서술자





● 차량 추적부

→ 추출된 차량 영역의 추적





● 차량 추적부

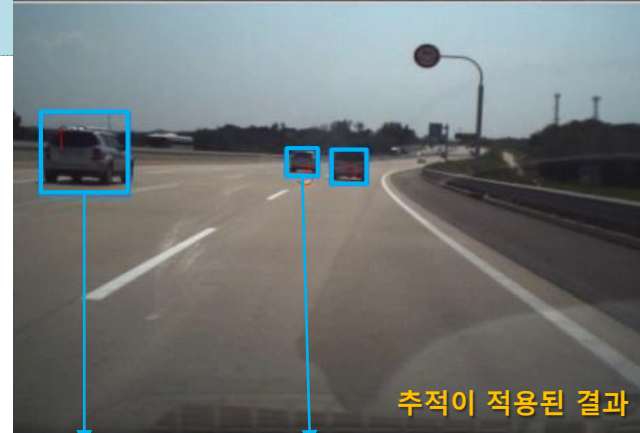
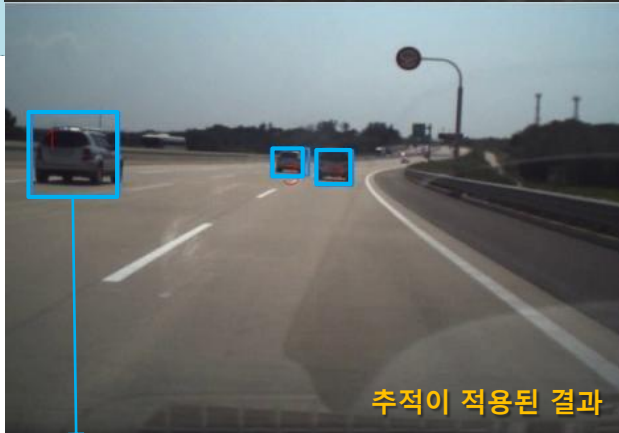
→ 추출된 차량 영역의 추적

실시간 영상에서 차량으로 검색된 영역

1023 frame



1024 frame



- 검증된 차량 영역을 기반으로 차량 추적
- 이전 프레임에서 차량으로 검증된 영역을 추적하여 현재 프레임에서 차량으로 검색되지 못하는 경우에 차량 정보 보정

KETI - FCW



FCW 기술개발에 필요한 영상인식 알고리즘의 권리 확보

Vehicle Detection

- 차량 내 설치된 카메라를 이용한 전방 차량과의 거리 감지
- 전방차량과의 거리가 가까울 경우 → 위험 경보음 발생



→ KETI FCW : 특허출원 4건

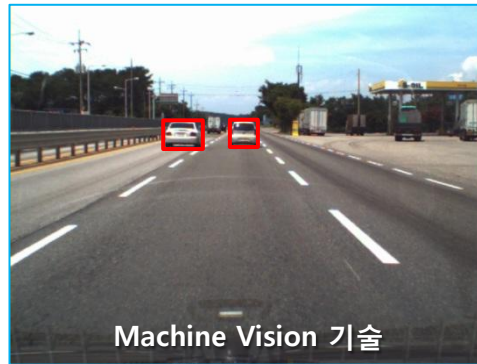
[차량 후보 영역 검출]



[Box 중첩 영역 제거]



[Resize Box]



DEMO - FCW



- **FCW DEMO**

Start Processing	Save Result
* Red Box : Ground True(Hand Make) * Green Box : Result of FCW Algorithm	

Embedded System

TI Davinci DM8148(DM8127) Processor



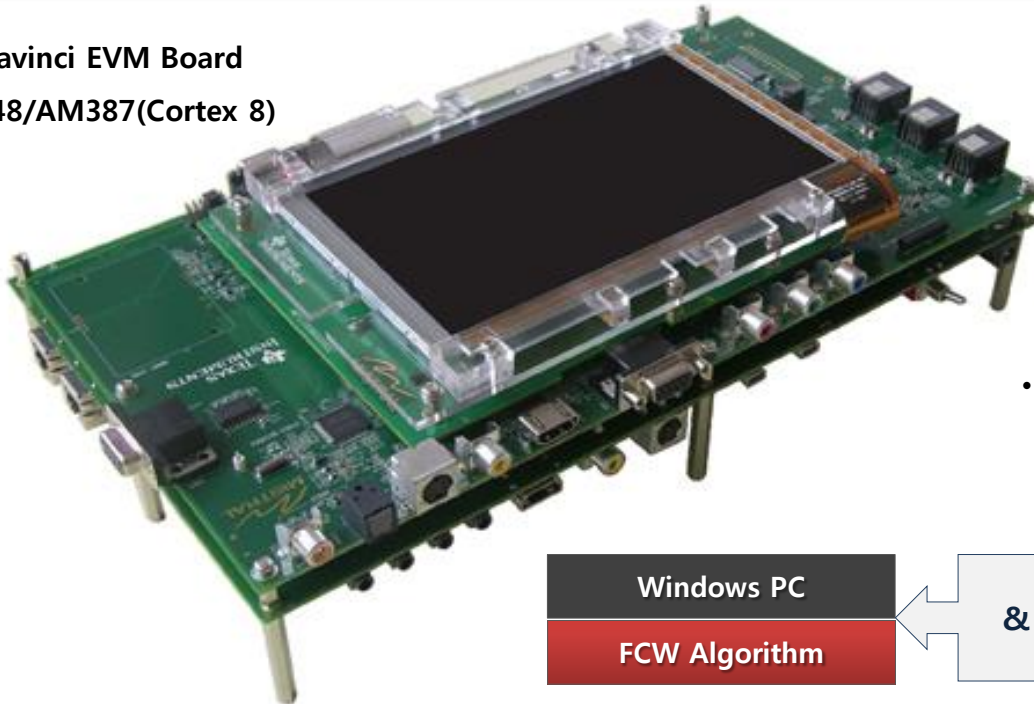
Embedded System



FCW 알고리즘 포팅

- TI Davinci DM8148(8127) Embedded System에 포팅
- Windows 기반으로 개발된 알고리즘을 DSP에 포팅
- FCW 알고리즘 : Harr-Lkie, Adaboost 알고리즘 OpenCV를 활용
→ DSP에서는 OpenCV 라이브러리가 없으므로 OpenCV 종속성을 제거 후 포팅
- ARM-DSP의 데이터 전달 문제

TI Davinci EVM Board
DM8148/AM387(Cortex 8)



• DSP : 1GHz

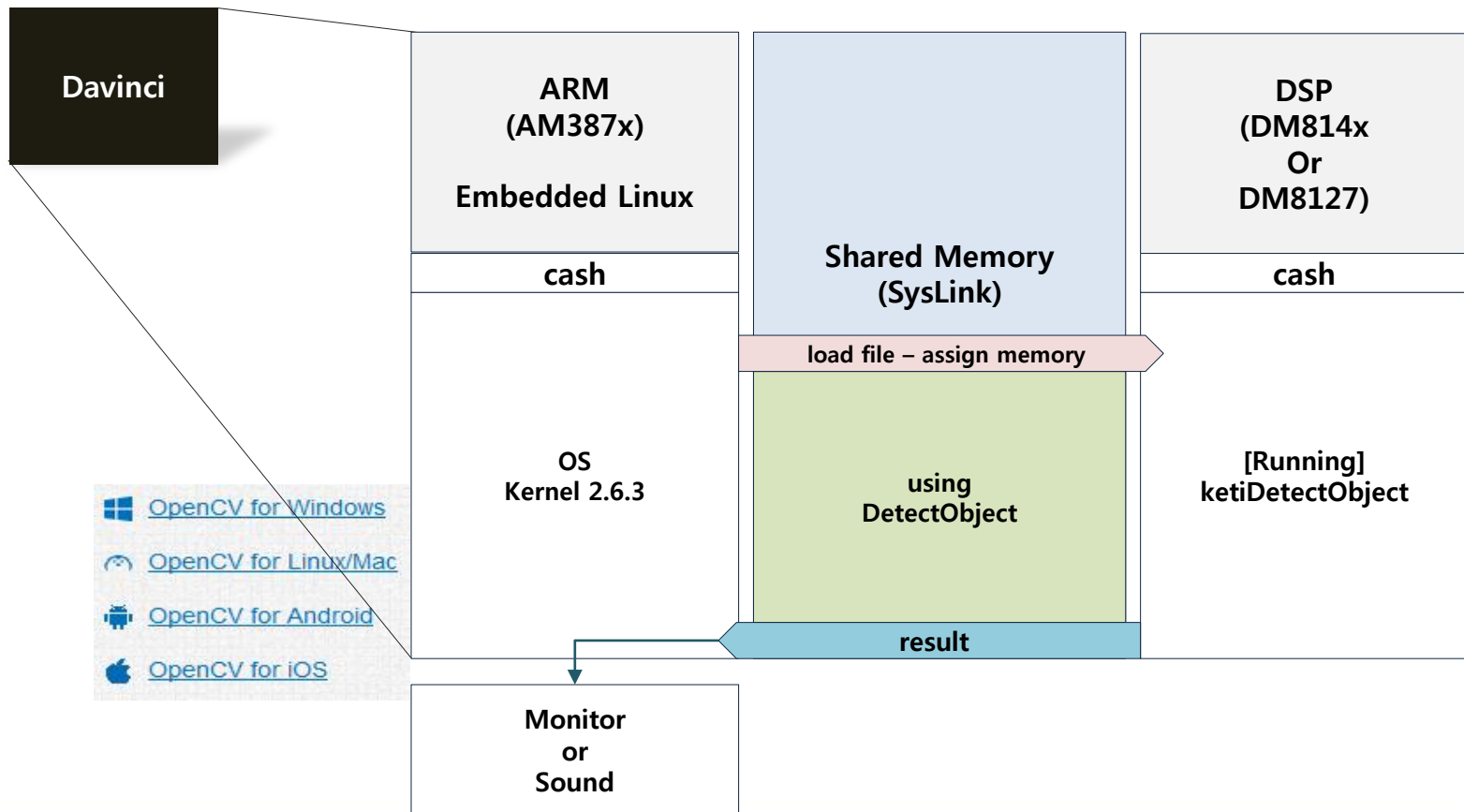


Embedded System



동작 구조 설계

- ARM 프로세서 : OS 및 주변장치 입출력 담당
- DSP 프로세서 : 입력 영상에서 FCW 알고리즘 동작
- Shared Memory를 통한 영상 입력 및 FCW 작동 결과 전송



Embedded System

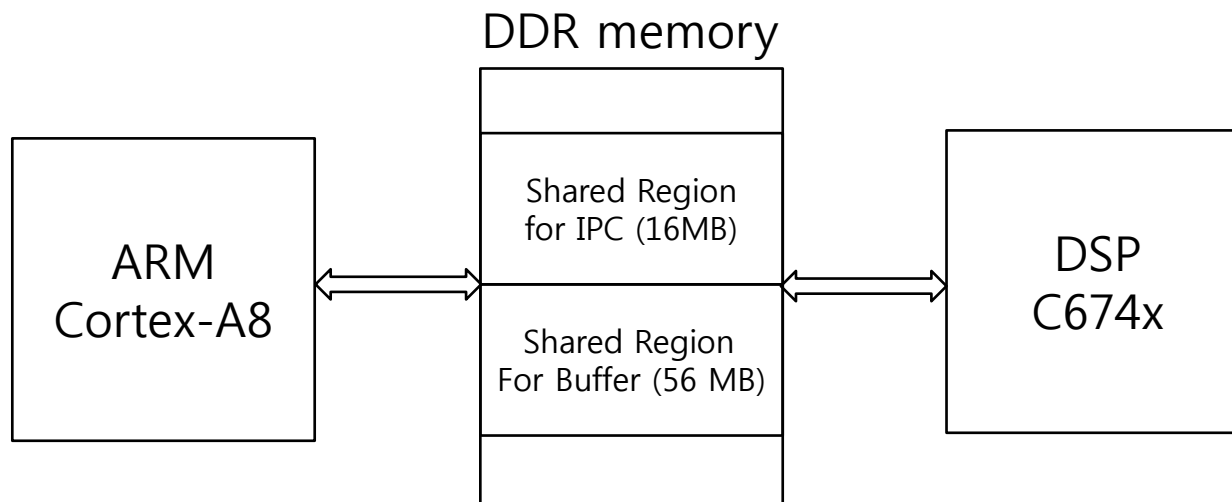


ARM과 DSP 연동 방법

- Codec Engine방식(xDM, xDAIS) : C++/malloc를 원활하게 지원하지 못함
→ http://processors.wiki.ti.com/index.php/Porting_GPP_code_to_DSP_and_Codec_Engine#How_do_I_make_C.2B.2B_code_XDAIS_compatible.3F [참고]
- Syslink 방식 : shared memory를 이용한 IPC(Inter-Process Communication) 및 데이터를 공유하는 방식

• SysLink란?

- ARM과 DSP간의 통신방법
- Shared Memory를 이용한 방법
- IPC용 메모리와 Buffer용 공유 메모리 사용



OpenCV 종속성 제거

- OpenCV는 다양한 영상인식 알고리즘을 지원하도록 되어 있는 영상처리 라이브러리
- 일반적 리눅스에서는 OpenCV 라이브러리를 제공하나, DSP에서 작동하기 위한 라이브러리는 없음
→ OpenCV를 이용한 Harr-Like, Adaboost 알고리즘의 사용을 위한 OpenCV 종속성 제거

• 사용하는 OpenCV

→ cvHaarDetectObject() 함수에서 사용하고 있는 OpenCV 소스 뜯어내기

- . CvHaarClassifierCascade
- . CvMemStorage
- . CvMemBlock
- . CvSeq
- . CvMat
- . CvMatND
- . Class Mat
- . CvSize
- . CvSeqReader
- . Cv
- . #define, Inline, Static



```
class Mat;  
cv::Mat, cv::parallel_for(), groupRectangles(), cv::Ptr<CvMat>, cv::AutoBuffer, std::vector  
func( src.data, src.step, sum.data, sum.step, sqsum.data, sqsum.step, tilted.data, tilted.step, src.size(), cn );  
  
//ba.cpp  
func(int /*i*/, int /*j*/, CvMat *point_params, CvMat* cam_params, CvMat* estim, void* /*data*/)
```

Embedded System



OpenCV 클래스 및 함수

```
void CvArr;
```

```
template<typename _Tp>
class Matx
class Vec : public Matx
class Point_
class Point3_
class Rect_
class Ptr
class Size_
class Scalar_ : public Vec
class Range
class MatAllocator
class AutoBuffer
class MatOp
class MatExpr
class _InputArray
class _OutputArray : public _InputArray
class Mat
class Mat_
class NArrayMatIterator
class SVD
```

```
class MatOp_Identity : public MatOp
class MatOp_AddEx : public MatOp
class MatOp_Bin : public MatOp
class MatOp_Cmp : public MatOp
class MatOp_GEMM : public MatOp
class MatOp_Invert : public MatOp
class MatOp_T : public MatOp
class MatOp_Solve : public MatOp
class MatOp_Initializer : public MatOp
```

typedef struct

```
_IplROI
_IplImage
CvMatND
CvSize
CvString
CvMemBlock
CvMemStorage
CvMemStotagePos
CvSeqBlock
CvSeq
CvSetElem
CvSet
CvRect
CvMat
CvSparseMat
CvSparseNode
CvSparseMatIterator
CvSaclar
CvSeqReader
CvPoint
CvPoint2D32f
CvSize2D32f
CvBox2D
CvPoint3D32f
CvPoint2D64f
CvPoint3D64f
```

Class Function

- 각 클래스별 생성자 및 생성자 오버라이드
- 각 클래스별 함수

void, int, Mat_, Point_ etc... Functions

```
cvSVD()
checkRange()
cvCheckArr()
cvSetSeqReaderPos()
cvStartReadSeq()
cvSliceLength()
cvCvtSeqToArray()
cvarrToMat()
cvConvertScale()
cvCvtSeqToArray()
cvSliceLength()
cvSetSeqReaderPos()
```

```
enum = { }
```

```
#definc CV_INLINE, CV_HAAR_DO_CANNY_PRUNING etc....
```

OpenCV 종속성 제거 - 예시

● Kalman 필터 - C++ 코드로 변경

```
class CMatrix
{
public:
    double Minor(int row, int column) const;
    double Cofactor(int row, int column) const;
    CMatrix Adjoint() const;
public:
    CMatrix operator^(const double oper) const;
    BOOL operator!=(const CMatrix &rhs) const;
    BOOL operator==(const CMatrix &rhs) const;
    CMatrix operator=(const CMatrix &rhs);
    CMatrix operator=(const double &oper);
    CMatrix operator=(const double *element);
    CMatrix operator+(const CMatrix &oper) const;
    CMatrix operator+(const CMatrix &oper) const;
    CMatrix operator+(const double oper) const;
    CMatrix operator-(const CMatrix &oper) const;
    CMatrix operator-(const double oper) const;
    double operator()(int row, int column) const;
    double operator()(int row, int column, double element);
public:
    CMatrix GauseJordanElimination() const;
    CMatrix GauseElimination() const;
    double Determinent() const;
    CMatrix PseudoInverse() const;
    CMatrix Inverse() const;
    CMatrix InverseGauseJordan() const;
    CMatrix Transpose() const;
    void Create(int row, int column);
    void Create(int row, int column, double element);
    BOOL CreateOnes(int row, int column);
    BOOL CreateZeros(int row, int column);
    BOOL CreateIdentity(int square);
};

//파라미터 초기화
void CKalmanFilter::InitParameters()
{
    m_A.Create(2,2);
    m_X.Create(2,1);
    m_X0.Create(2,1);
    m_Q.Create(2,2);
    m_R.Create(1,1);
    m_H.Create(1,2);
    m_P.Create(2,2);
    m_PO.Create(2,2);
    m_Z.Create(2,1);
    m_PredictPk.Create(2,2);
    m_PredictKk.Create(2,1);
    m_TransA.Create(2,2);
    m_TransH.Create(2,1);
    m_l.Create(2,2);
    m_tempS.Create(1,1);
    m_tempZ.Create(1,1);
    m_tempK.Create(2,1);

    m_X0(1,1,0.0);
    m_X0(2,1,0.0);
    m_X = m_X0;

    m_l(1,1,1.0);    m_l(1,2,0.0);
    m_l(2,1,0.0);    m_l(2,2,1.0);

    m_A(1,1,1.0);    m_A(1,2,1.0);
    m_A(2,1,0.0);    m_A(2,2,1.0);

    m_H(1,1,1.0);    m_H(1,2,0.0);
};

CMatrix::CMatrix(const CMatrix &rhs)
{
    m_nRow=rhs.m_nRow;
    m_nColumn=rhs.m_nColumn;
    m_pdbiElement = new double *[m_nRow];
    int i,j;
    for(i=0; i<m_nRow; i++)
        m_pdbiElement[i] = new double [m_nColumn];

    for(i=0; i<m_nRow; i++)for(j=0; j<m_nColumn; j++)
        m_pdbiElement[i][j]=rhs.m_pdbiElement[i][j];
}

CMatrix::CMatrix(const double &element)
{
    m_nRow=1;
    m_nColumn=1;
    m_pdbiElement = new double *[m_nRow];
    int i,j;
    for(i=0; i<m_nRow; i++)
        m_pdbiElement[i] = new double [m_nColumn];
    for(i=0; i<m_nRow; i++)for(j=0; j<m_nColumn; j++)
        m_pdbiElement[i][j]=element;
}

CMatrix::~CMatrix()
{
    for(int i=0; i<m_nRow; i++)
        delete [] m_pdbiElement[i];
    delete [] m_pdbiElement;
}
```

Embedded System



DSP 포팅 방법

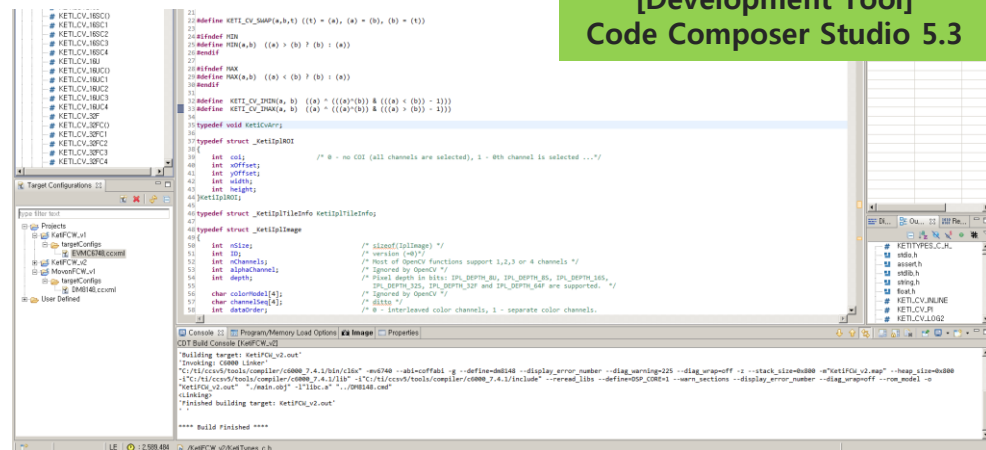
- Code Composer Studio를 이용한 DSP 코딩 및 소스 최적화
- USB Emulator를 이용



JTAG Emulator(XDS100v2 USB Emulator)



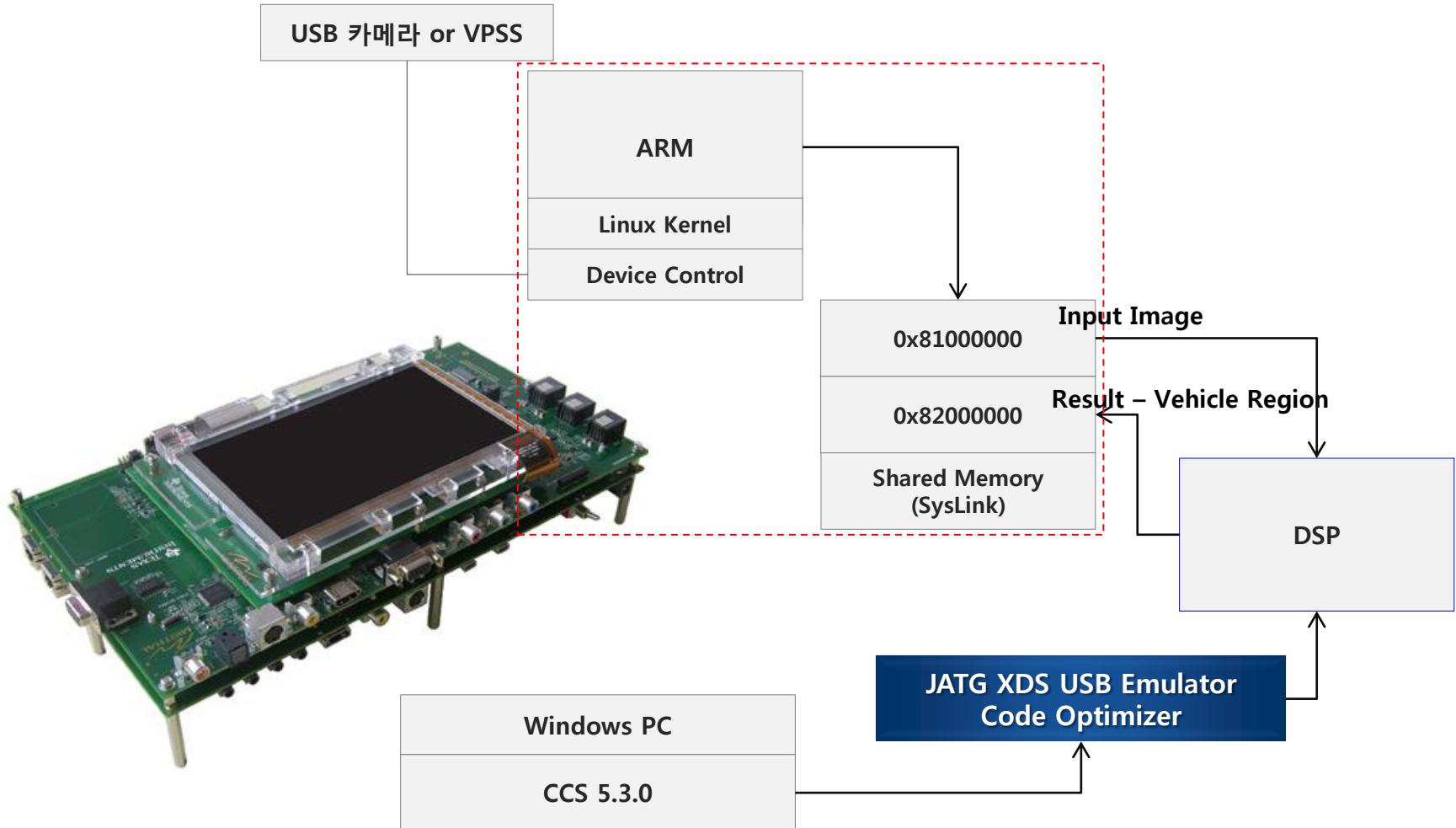
[Development Tool]
Code Composer Studio 5.3



Embedded System



CCS를 이용한 DSP 포팅



Embedded System



- Code Composer Studio 5.3.0
- Compile Option(Optimization Level) : 1~3

The screenshot displays the Code Composer Studio (CCS) 5.3.0 interface. The main window shows the source code for `KetiMatrix.cpp` with the following content:

```
104 {
105     for(int i=0; i< m_nRow; i++)
106         delete [] m_pdblElement[i];
107     delete [] m_pdblElement;
108     m_nRow=0;
109     m_nColumn=0;
110 }
111
112 double CMatrix::GetElement(int row, int column) const { return m_pdblElement[row-1][column-1]; }
113 double CMatrix::GE(int row, int column) const { return m_pdblElement[row-1][column-1]; }
114 double CMatrix::operator()(int row, int column) const { return m_pdblElement[row-1][column-1]; }
115
116 double CMatrix::SetElement(int row, int column, double element) { m_pdblElement[row-1][column-1]=element; return m_pdblElement[row-1][column-1]; }
117 double CMatrix::SE(int row, int column, double element) { m_pdblElement[row-1][column-1]=element; return m_pdblElement[row-1][column-1]; }
118 double CMatrix::operator()(int row, int column, double element) { m_pdblElement[row-1][column-1]=element; return m_pdblElement[row-1][column-1]; }
119
120 int CMatrix::GetRow() const { return m_nRow; }
121 int CMatrix::GetColumn() const { return m_nColumn; }
122
123 CMatrix CMatrix::Transpose() const
124 {
125     CMatrix temp;
126     temp.Create(m_nColumn, m_nRow);
127 }
```

The **Properties for KetiFCW_v2** dialog box is open, showing the **Optimization** settings:

- Configuration: Debug [Active]
- Optimization level (--opt_level, -O): 3
- Optimize for code size (--opt_for_space, -ms): [Unchecked]

The Project Explorer on the left shows the file structure for `KetiMatrix.cpp`, including `Math.h`, `EPISILON`, `NINF`, `PINF`, and various `CMatrix` methods.

Embedded System



Embedded Board 제작

- 모본과 협력하여 TI DM8127 프로세서를 이용한 전용 보드 제작
- FCW 알고리즘 포팅



DEMO – FCW Embedded



- FCW Embedded DEMO



활용방안

▷ 블랙박스 적용 기술

차선이탈감지시스템



전방추돌경고시스템



블랙박스



Camera



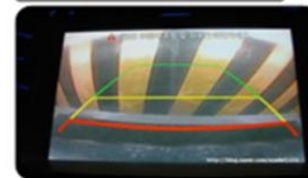
Display



Laser



Rear view



Application



- 블랙박스 Viewer
 - OBD2 연동 + Telematics
- 차량 자가진단 및 스캔
고장코드진단
사고발생 확인 후 119 연락



활용방안



'블랙박스 영상정보 처리기기 보안 기술 보유'

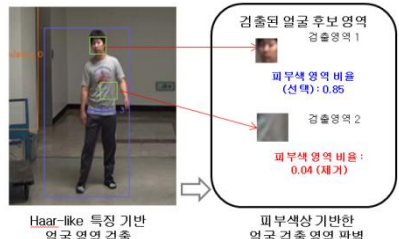
영상 개선 기술

- 얼굴 및 번호판 영역을 추출하기 위한 영상 개선 기술
- 영상 개선방법의 경우 히스토그램, 비선형 기반 등의 다양한 개선방법, 블랙박스에서 작동되어야 하므로, **실시간성이 보장되는 개선 방법을 적용**해야 함



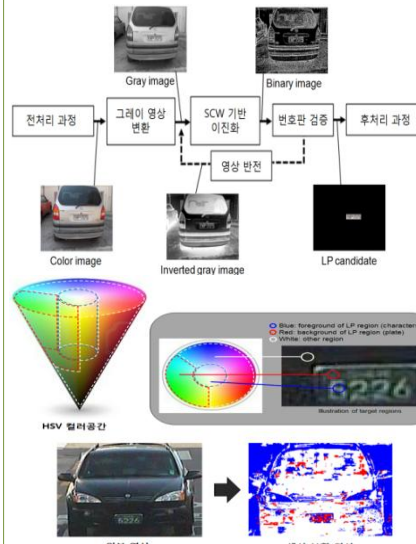
얼굴 검출 기술

- 공개 DB를 이용하여 얼굴 영역 특징을 추출하고 Cascade Classifier 모델을 훈련하여 분류기를 생성
- **분류기를 통해 얼굴 후보 영역을 검출**하고 피부 색상 및 얼굴 특징을 이용한 **얼굴 영역 검증 기술**



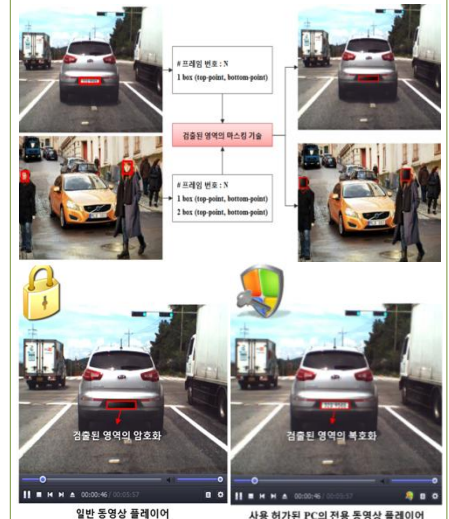
차량 번호판 검출 기술

- 엣지(edge) 기반 차량 검출
- 전처리 과정, 그레이 영상 변환, SCW(Sliding Concentric Window) 기반 이진화, 번호판 검증, 후처리 과정을 통한 **차량의 번호판 영역 검출**



마스킹/언마스킹 기술

- **검출된 영역의 마스킹 암호화 기술**
- 일반 동영상 플레이어에서는 검출된 영역이 보이지 않으며 전용 플레이어와 사용 허가된 PC에서만 검출된 영역의 **언마스킹 기술**





감사합니다.

영상인식 기술기반 FCW 임베디드 시스템 기술