

사고 예방을 위한 사각지대 이동 물체 추적 및 경로 예측 기술 개발

김건주, 백장현, 허준영

1. 프로젝트 소개
2. 프로젝트 목표
3. 프로젝트 내용
4. 프로젝트 결과
5. 결론

1. 프로젝트 소개

1. 프로젝트 개발 배경



<https://mn.kbs.co.kr/news/view.do?ncd=3121094>

1. 프로젝트 소개

1. 프로젝트 개발 배경



그림 1-1 AEB 구성 (현대모비스)



그림 1-2 아파트단지 사고사례 (KBS NEWS)

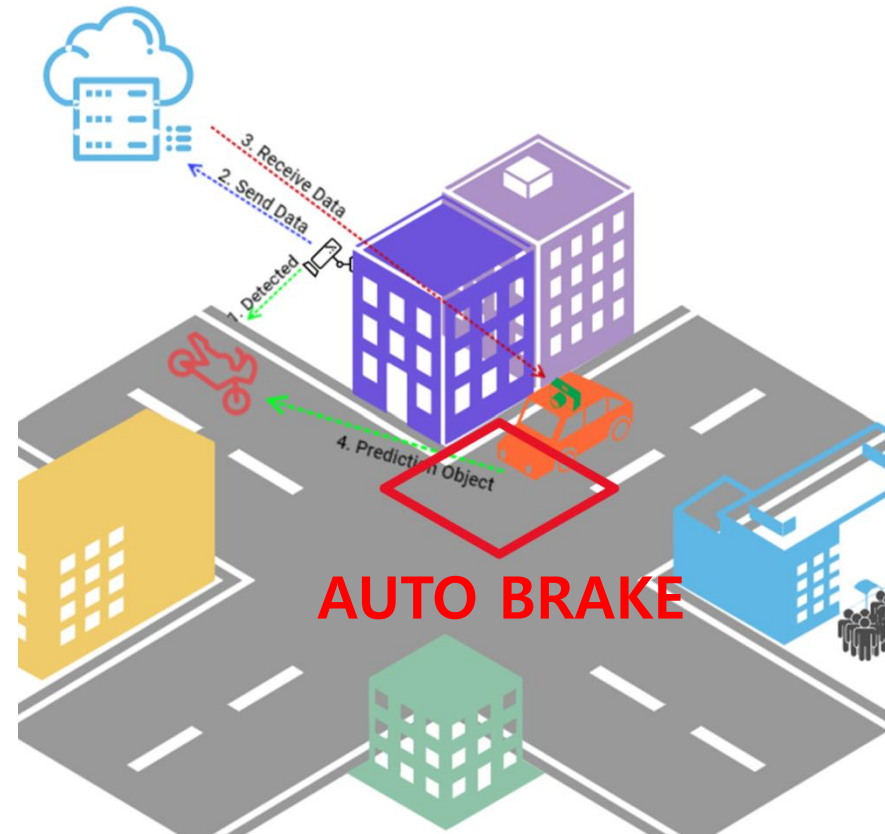
- 현재 AEB는 직진 차량, 자전거, 횡단 보행자에 대해서만 동작
- 카메라를 통해 사각지대에서 접근하는 오토바이, 전동 킥보드도 검출 가능

1. 프로젝트 소개

2. 확장된 AEB (Extended Autonomous Emergency Braking)



기존의 AEB



Extended AEB

2. 프로젝트 목표

1. 제동거리, 충돌 확률 기준 설정

● 차량 속도에 따른 제동거리 변화

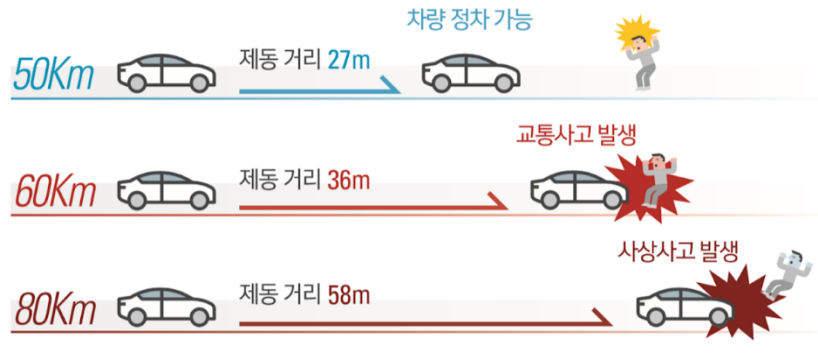


그림 2-1 제동거리 (경찰청)

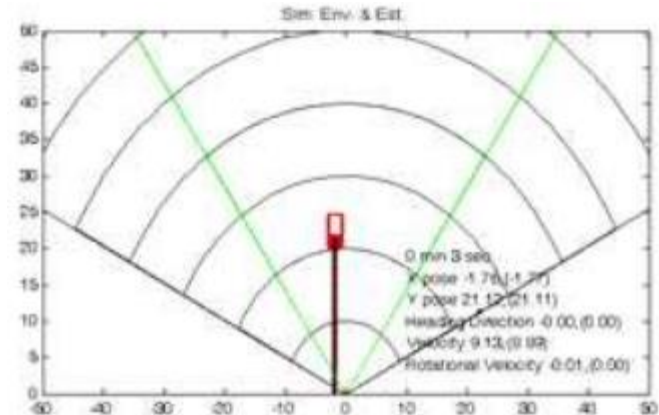


그림 2-2 전방 센서 범위 (현대모비스)

[참조] 현대모비스, 전방 차량 정보를 이용한 충돌 위험도 판단 알고리즘 2013

1. 현대 모비스에서 사용하는 카메라 센서 범위 전방 40m(보행자 인식)
2. 해당 센서 범위를 토대로 작성된 충돌 위험도 판단 알고리즘의 경우

충돌 위험도 확률 70% : 충돌 위험 알림, Active Seatbelt 동작
충돌 위험도 확률 90% : 충돌이라고 판단, AEB 동작

2. 프로젝트 목표

2. 팀원 소개



팀장 : 김건주



팀원 : 백장현



팀원 : 허준영

역할

1. Kalman Filter 구현
2. Collision Risk model 구현
3. Thread 관리
4. 통합 TEST

역할

1. Github 관리
2. DarkNet 네트워크 성능 비교
3. 좌표 값 Calibration 구현
4. 단안카메라를 이용한 object의 거리 검출

역할

1. ROS 통신 구현
2. PWM을 이용한 AEB 구현
3. 통합 TEST
4. 보고서 작성

3. 프로젝트 내용

1. 전체 시스템 구성도

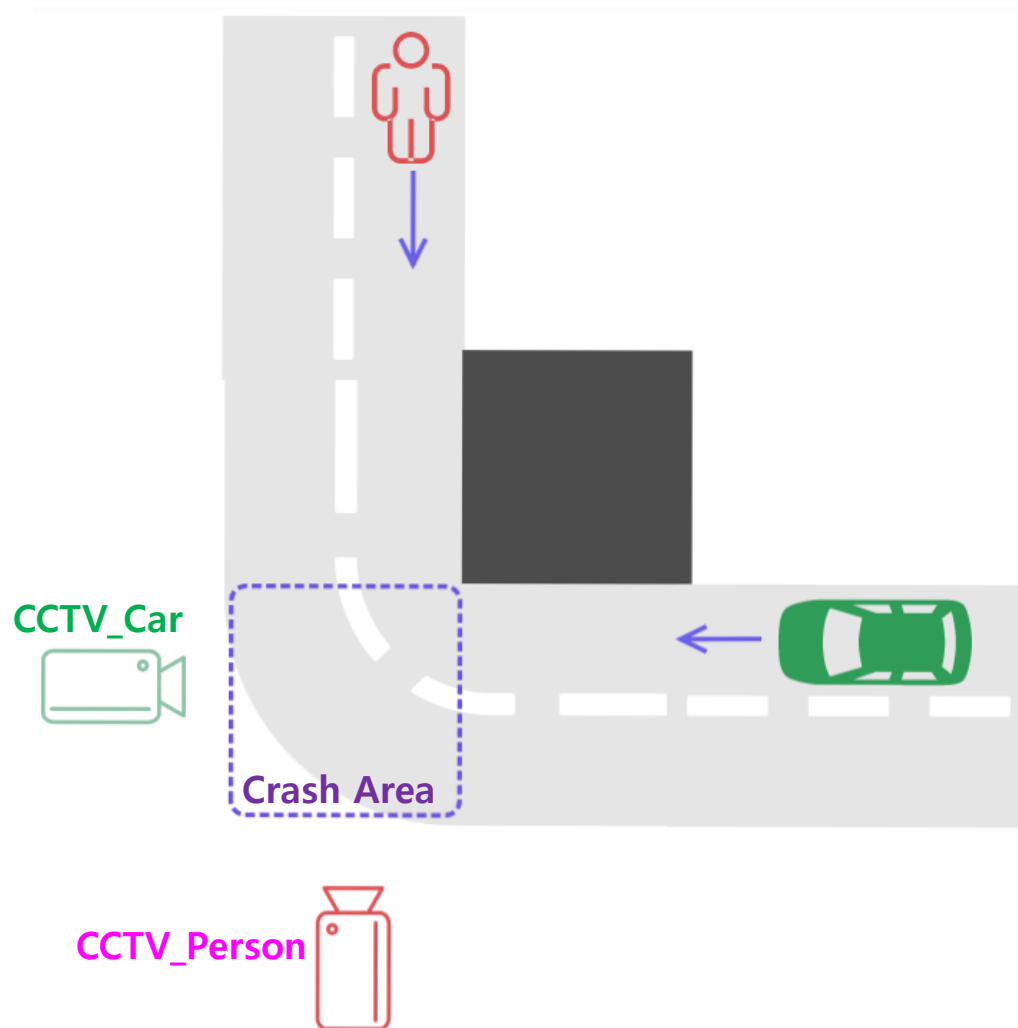


그림 3-1 Testbed 구성도

3. 프로젝트 내용

2. H/W 구성도

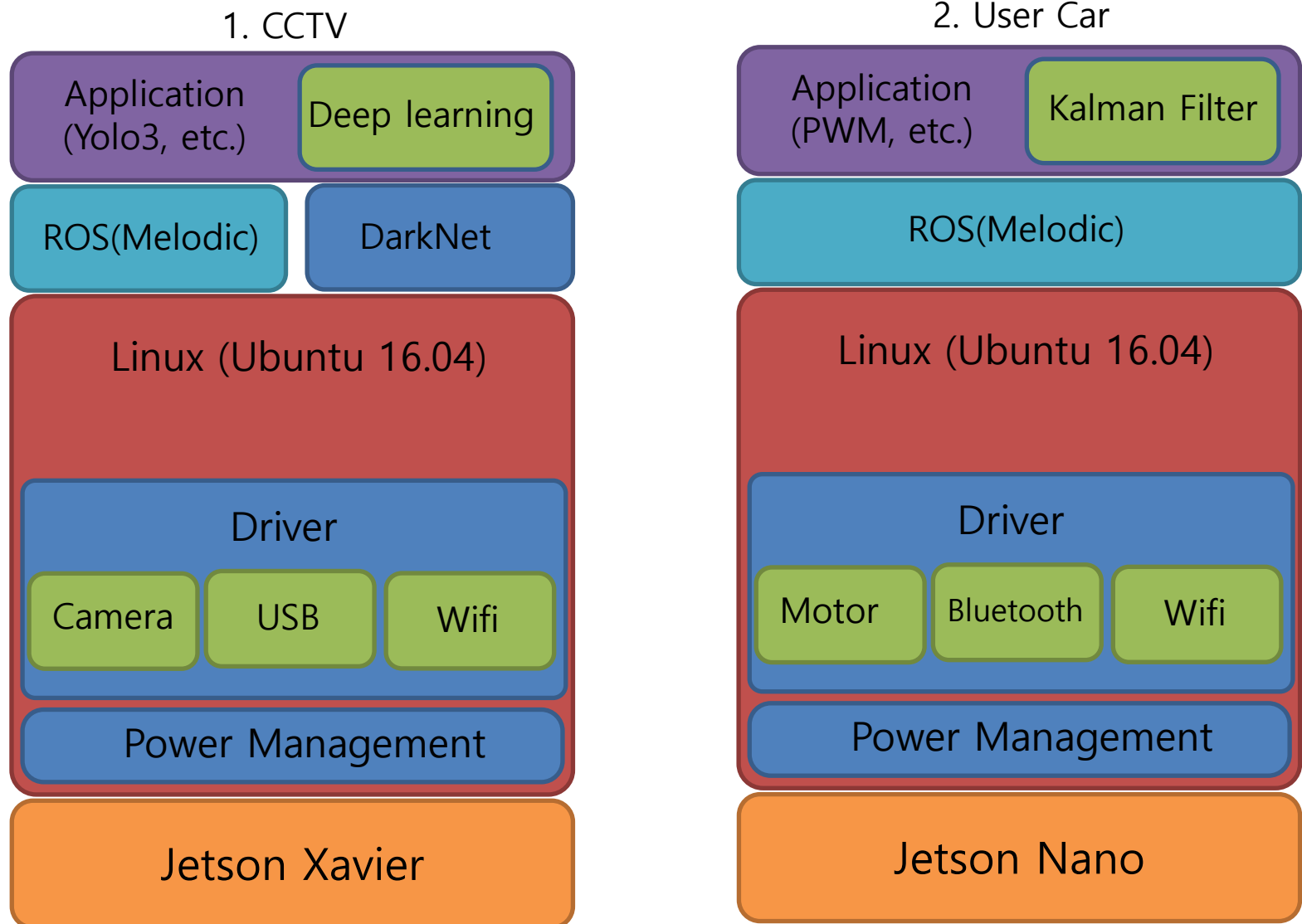


그림 3-2 아키텍처 구성도

3. 프로젝트 내용

2. H/W 구성도

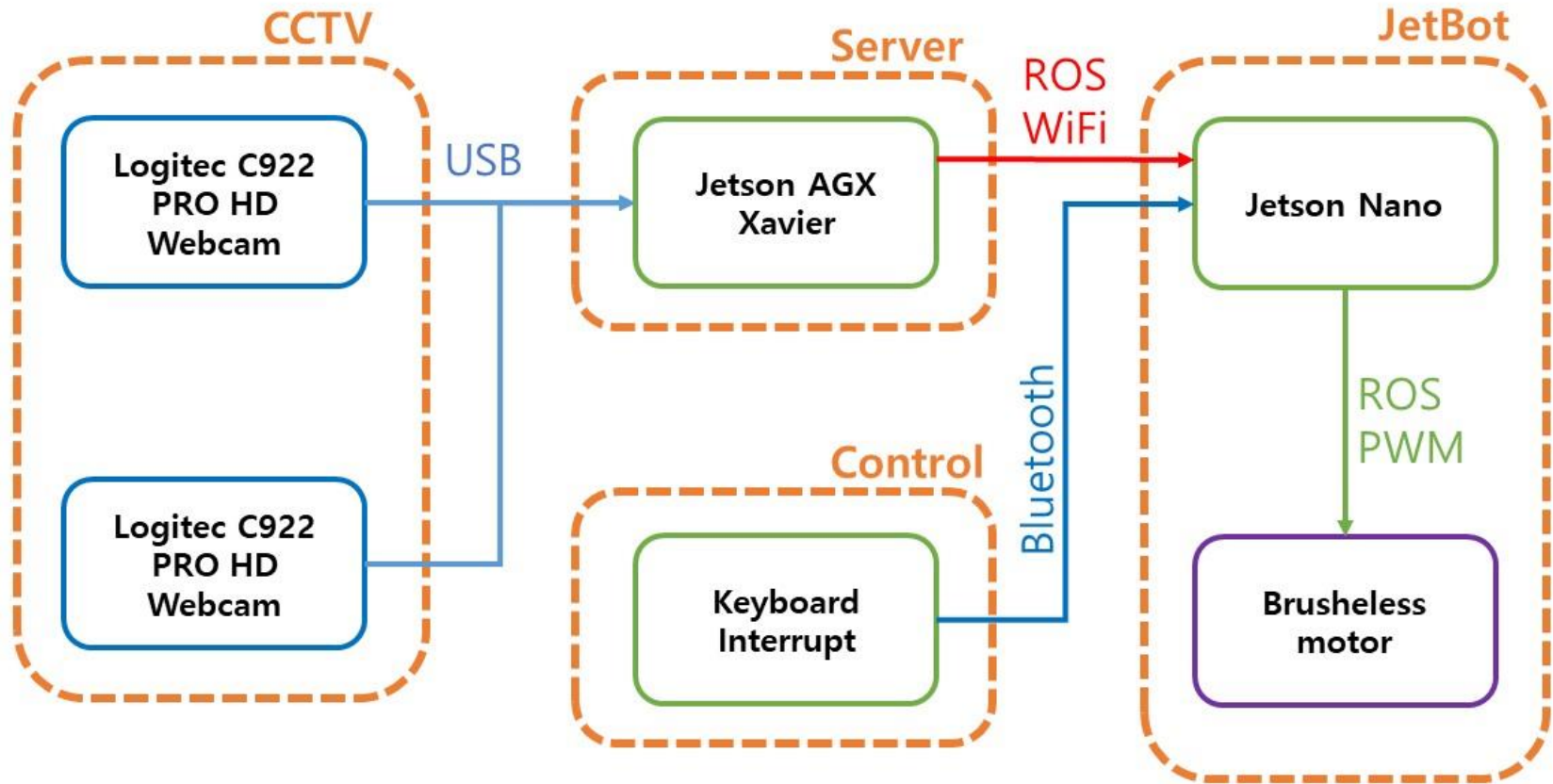


그림 3-3 H/W 구조

3. 프로젝트 내용

3. S/W 구성도

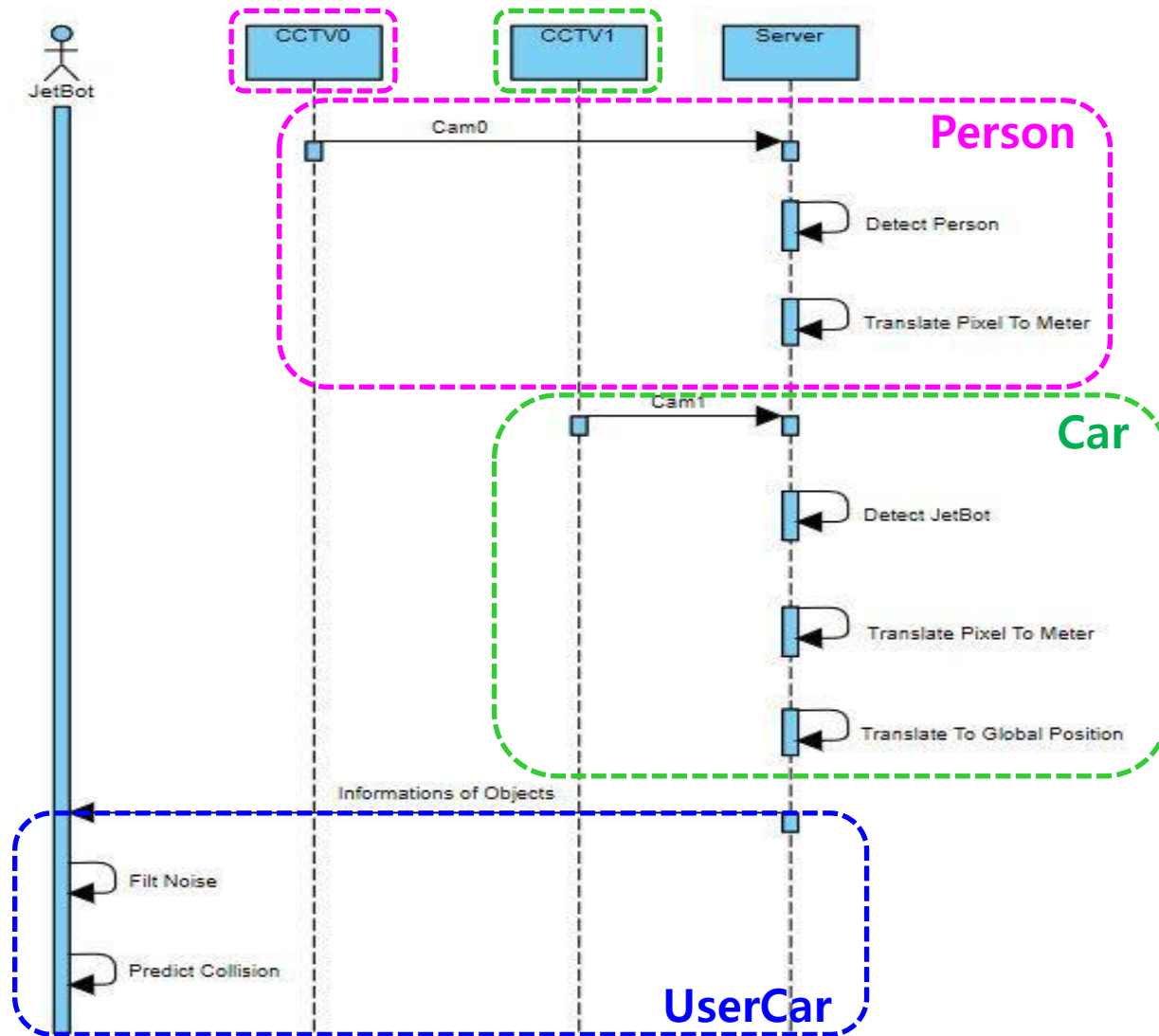


그림 3-4 프로세스

3. 프로젝트 내용

3. S/W 구성도

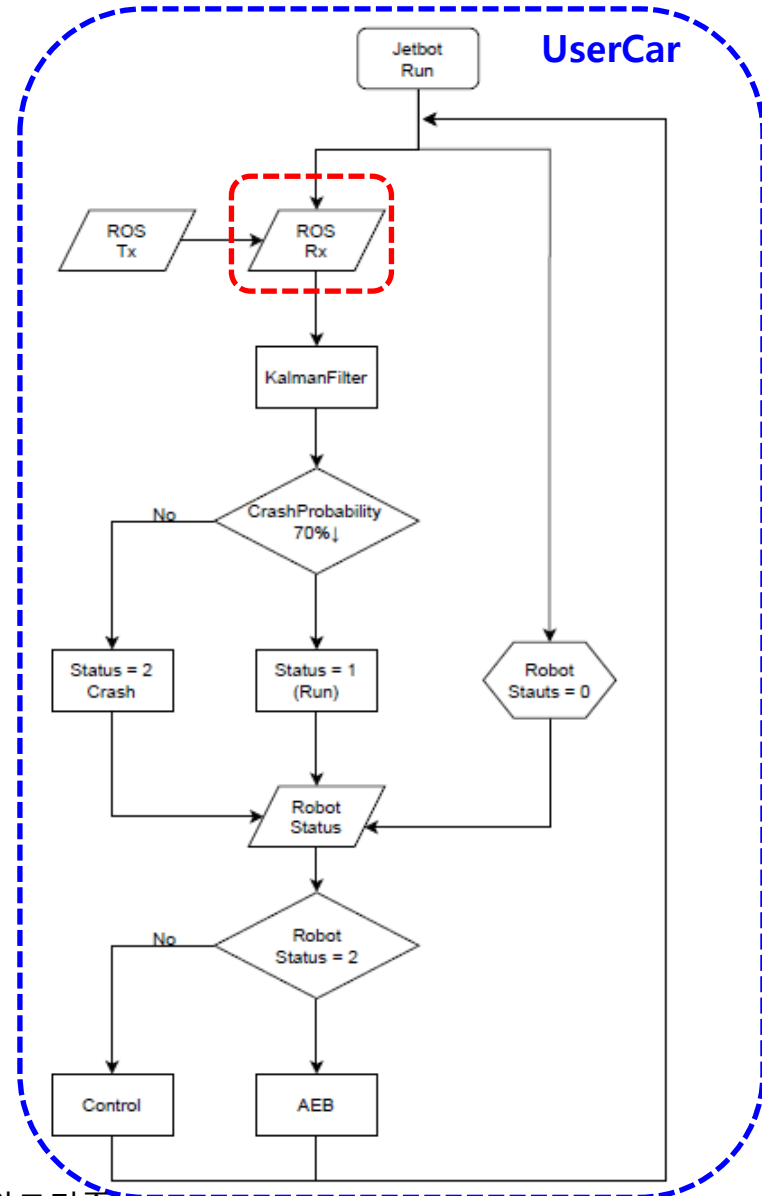
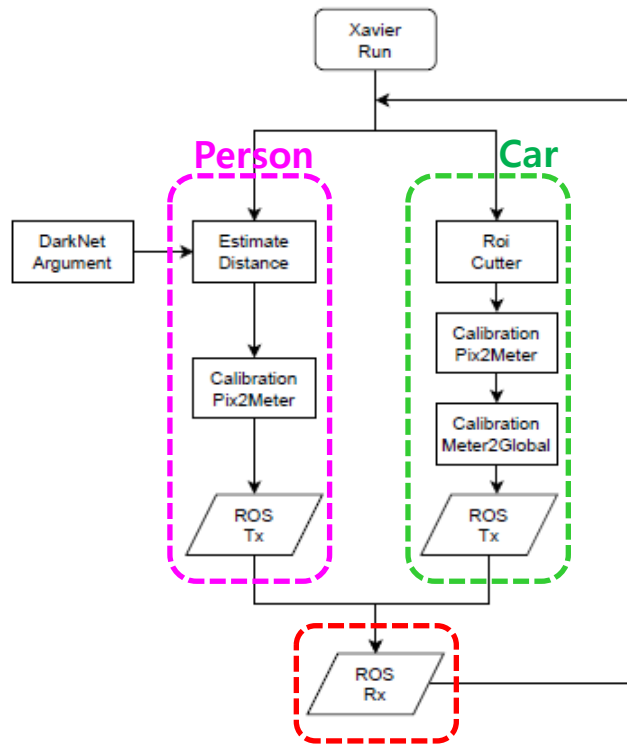
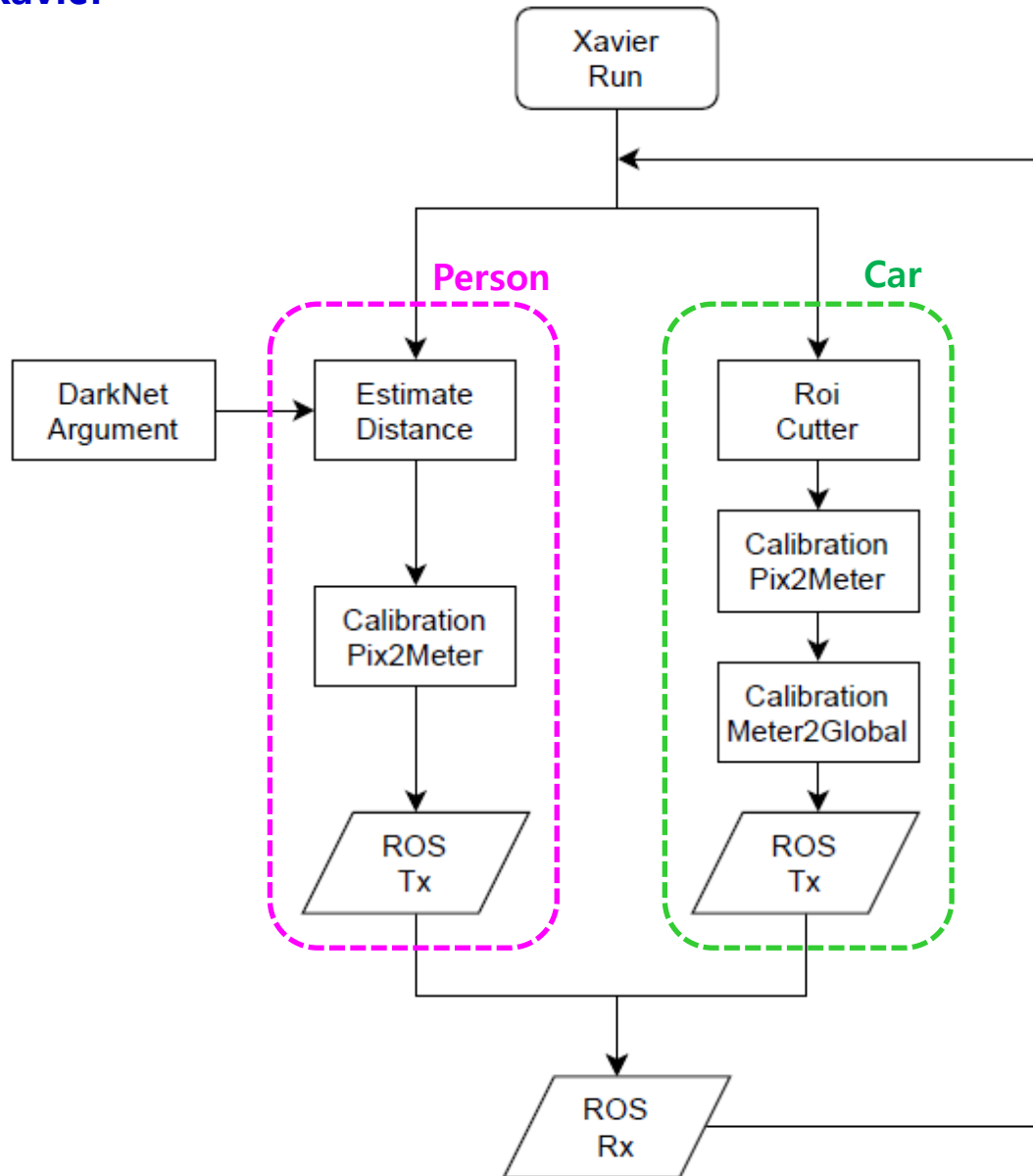


그림 3-5 알고리즘

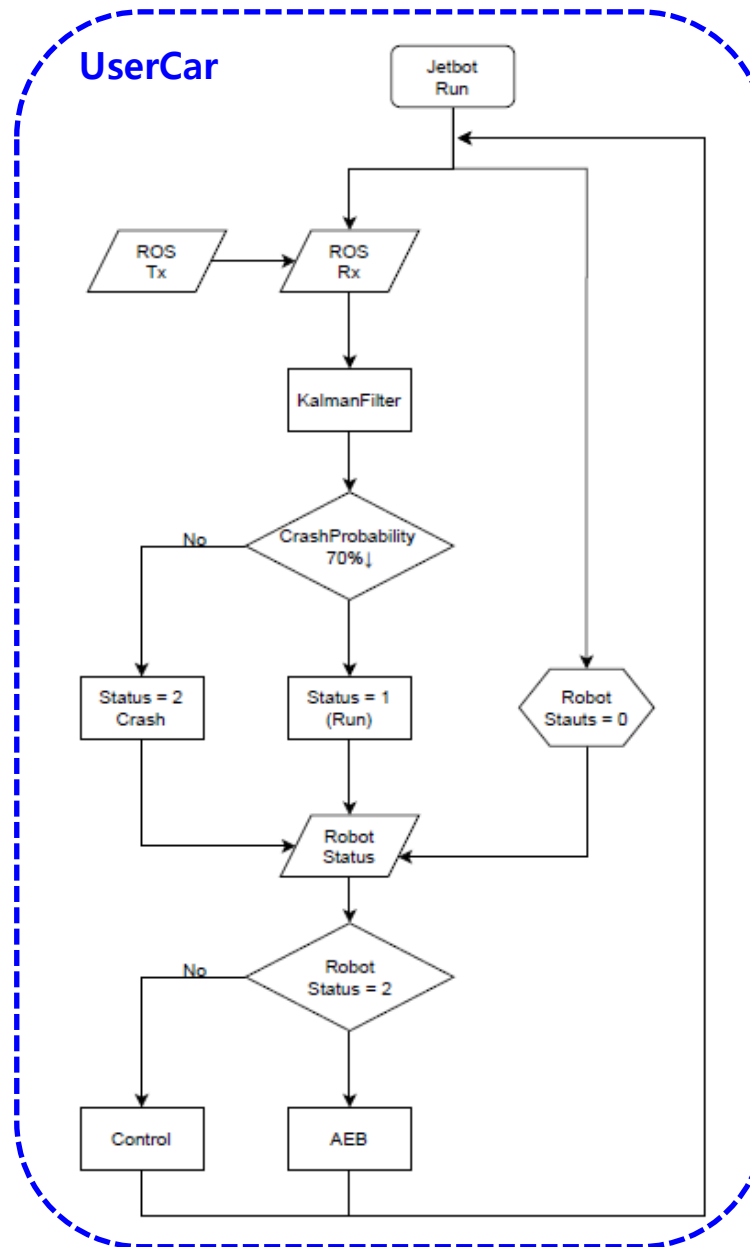
3. 프로젝트 내용

3. S/W 알고리즘_Xavier



3. 프로젝트 내용

3. S/W 알고리즘_Jetbot



4. 프로젝트 결과

1. Testbed 구축

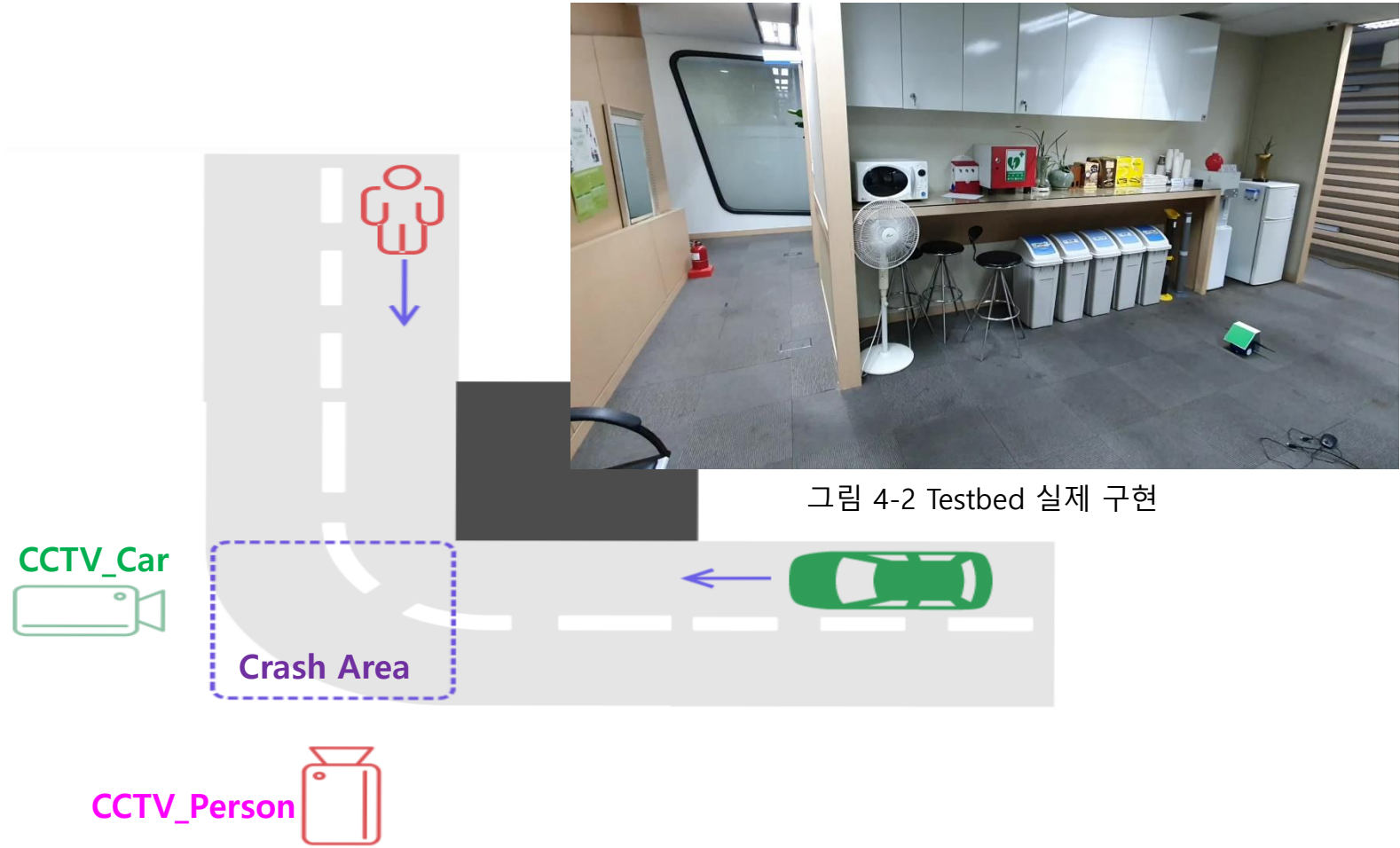


그림 4-2 Testbed 실제 구현

그림 4-1 Testbed 구성도

4. 프로젝트 결과

2. 좌표계 변환 (Pixel to Meter)

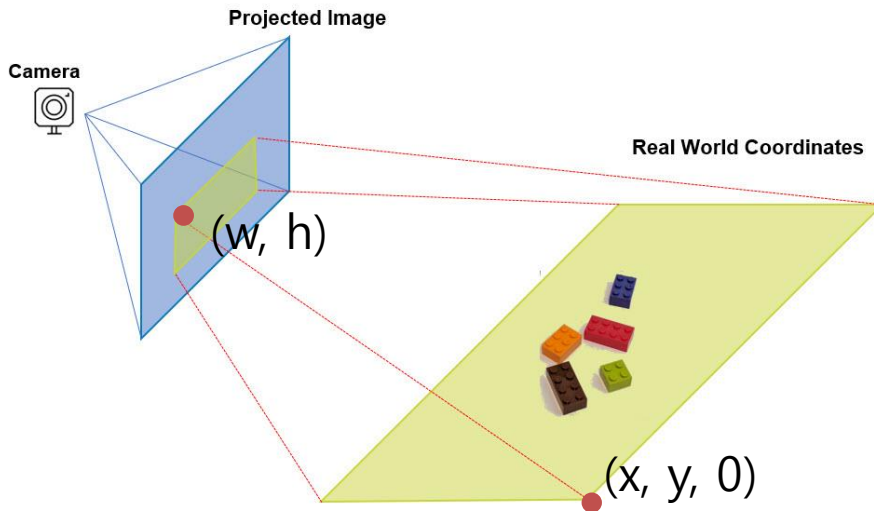


그림 4-3 카메라 좌표계

$$\begin{pmatrix} a & b & 0 & c \\ d & e & 0 & f \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} w \\ h \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} x \\ y \\ 0 \\ 1 \end{pmatrix}$$

$A \qquad P \qquad R$

- 픽셀좌표 P와 대응하는 실제좌표 R의 순서쌍을 세 개 이상 구하면 Z=0인 픽셀좌표 P를 실제좌표 R로 변환시키는 변환행렬 A를 구할 수 있음
- 실제 좌표 R의 **영점은 카메라의 수직 아래방향 바닥 지점**으로 기준
- Numpy lib의 linalg.inv함수를 이용하여 A^{-1} 을 구하면 각 카메라마다 픽셀좌표 P를 실제좌표 R로 변환시킬 수 있음

4. 프로젝트 결과

2. 좌표계 변환 (Pixel to Meter)

CCTV_Person



그림 4-4 CCTV_Person Calibration

CCTV_Car

거리에 따라 X, Y 변화량이 다름
빨강 : X축 파랑 : Y축

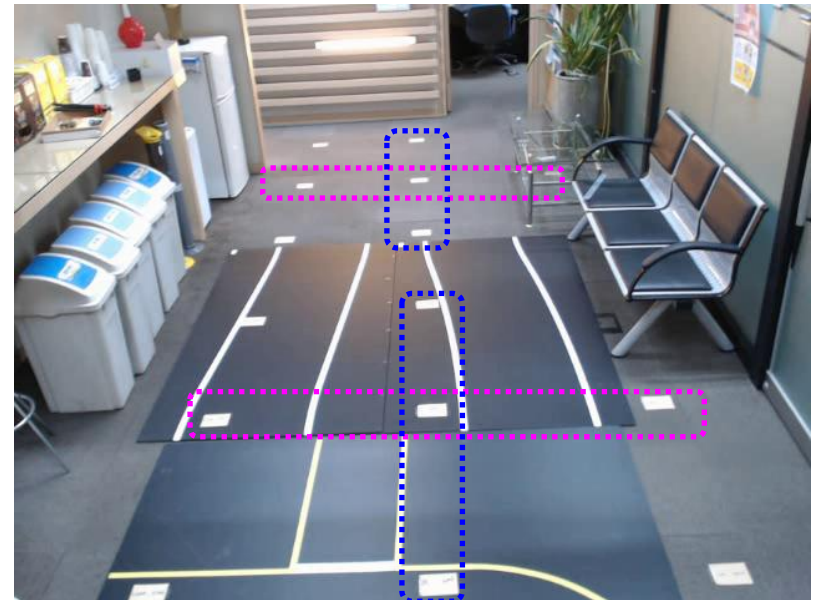


그림 4-5 CCTV_Car Calibration

4. 프로젝트 결과

2. 좌표계 변환 (Pixel to Meter)

Num	Pixel X	Pixel Y	Real X	Real Y
1	104	480	-100	200
2	341	474	0	200
3	580	467	100	200
4	160	339	-100	300
5	336	330	0	300
6	520	326	100	300
7	190	258	-100	400
8	332	244	0	400
9	217	191	-100	500
10	327	187	0	500
11	234	149	-100	600
12	326	145	0	600
13	246	114	-100	700
14	325	112	0	700

표 4-1 CCTV_Car 좌표 값

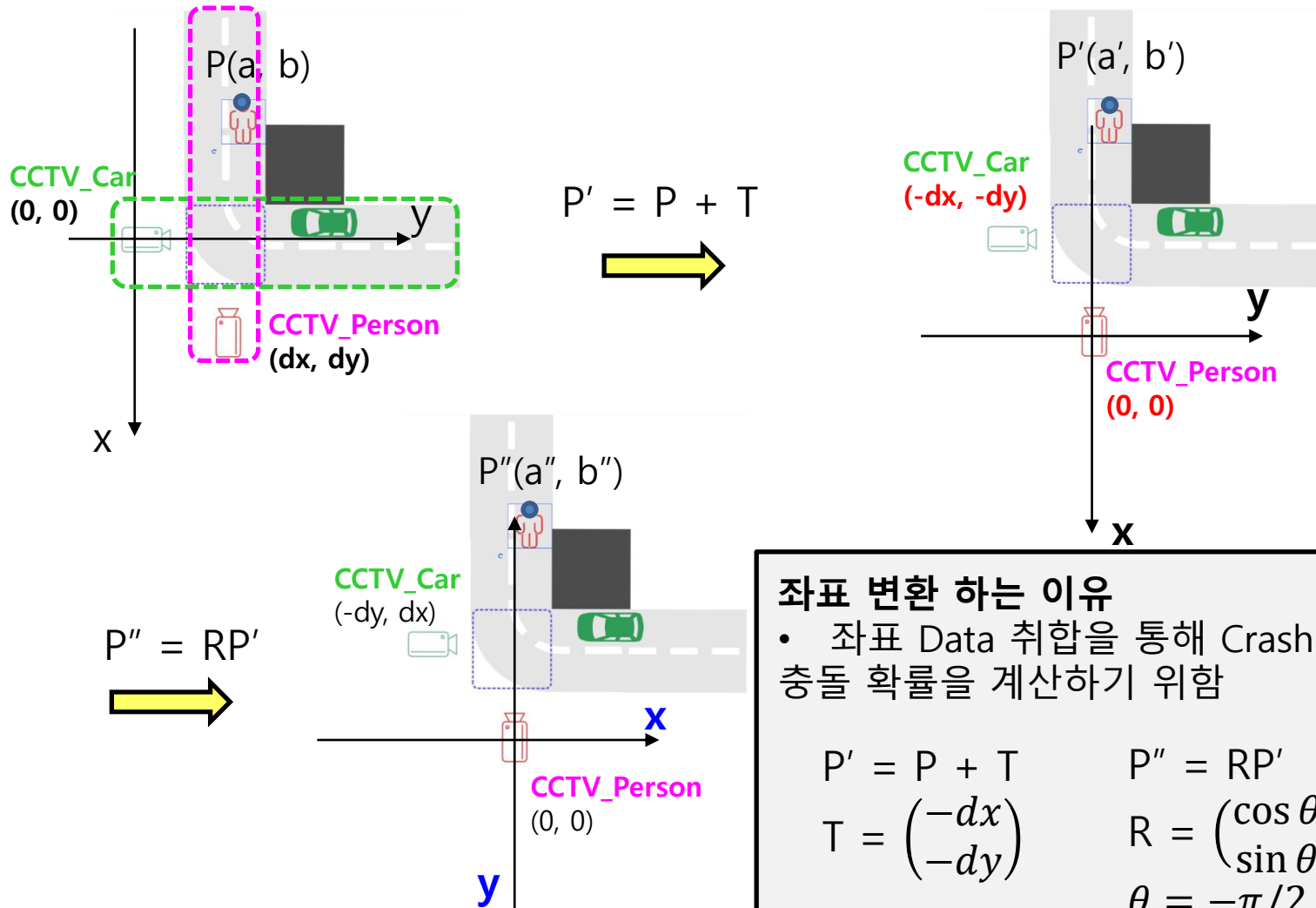
CCTV_Car



그림 4-6 CCTV_Car 좌표 지점

4. 프로젝트 결과

3. 좌표계 변환 (Meter to Global)



좌표 변환 하는 이유

- 좌표 Data 취합을 통해 Crash Area에 대해 충돌 확률을 계산하기 위함

$$P' = P + T$$

$$T = \begin{pmatrix} -dx \\ -dy \end{pmatrix}$$

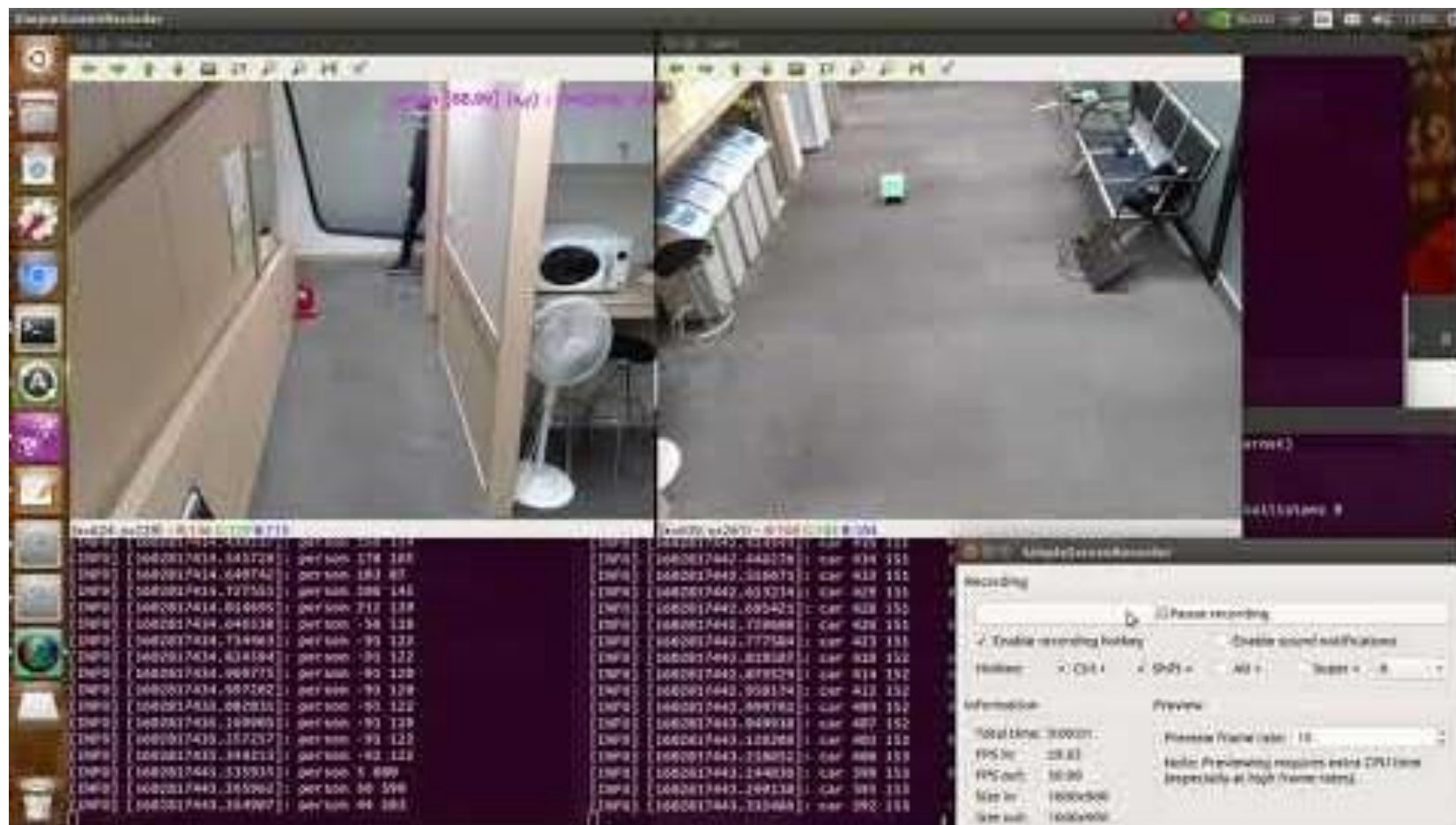
$$P'' = RP'$$

$$R = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix}$$

$$\theta = -\pi/2$$

4. 프로젝트 결과

4. TX_Xavier 영상



4. 프로젝트 결과

4. TX_Xavier

Person

Car

SimpleScreenRecorder

bbox

cam1

person [99.90] (x,y) : (18,548), dist : 54

person -91 120

person -91 122

person -91 120

person -91 122

person -92 122

person 5 880

person 50 590

person 44 583

person 43 575

person 43 578

person 39 582

person 36 582

person 30 571

person 26 562

person 20 557

person 19 540

person 18 548

Class,
X(cm),Y(cm)
Pixel to Meter

[INFO] [1602817443.299138]: car 395 153

[INFO] [1602817443.335466]: car 392 153

[INFO] [1602817443.396018]: car 388 153

[INFO] [1602817443.478076]: car 386 153

[INFO] [1602817443.500000]: car 383 153

[INFO] [1602817443.500000]: car 381 153

[INFO] [1602817443.500000]: car 377 153

[INFO] [1602817443.500000]: car 374 153

[INFO] [1602817443.500000]: car 372 153

[INFO] [1602817443.500000]: car 368 154

[INFO] [1602817443.877064]: car 365 154

[INFO] [1602817443.915817]: car 363 155

[INFO] [1602817443.994751]: car 359 155

[INFO] [1602817444.025988]: car 353 155

[INFO] [1602817444.075810]: car 351 155

[INFO] [1602817444.108246]: car 347 156

[INFO] [1602817444.166889]: car 344 156

Class,
X(cm),Y(cm)
Pixel to Global

그림 4-7 CCTV 좌표Data 출력

4. 프로젝트 결과

5. ROS

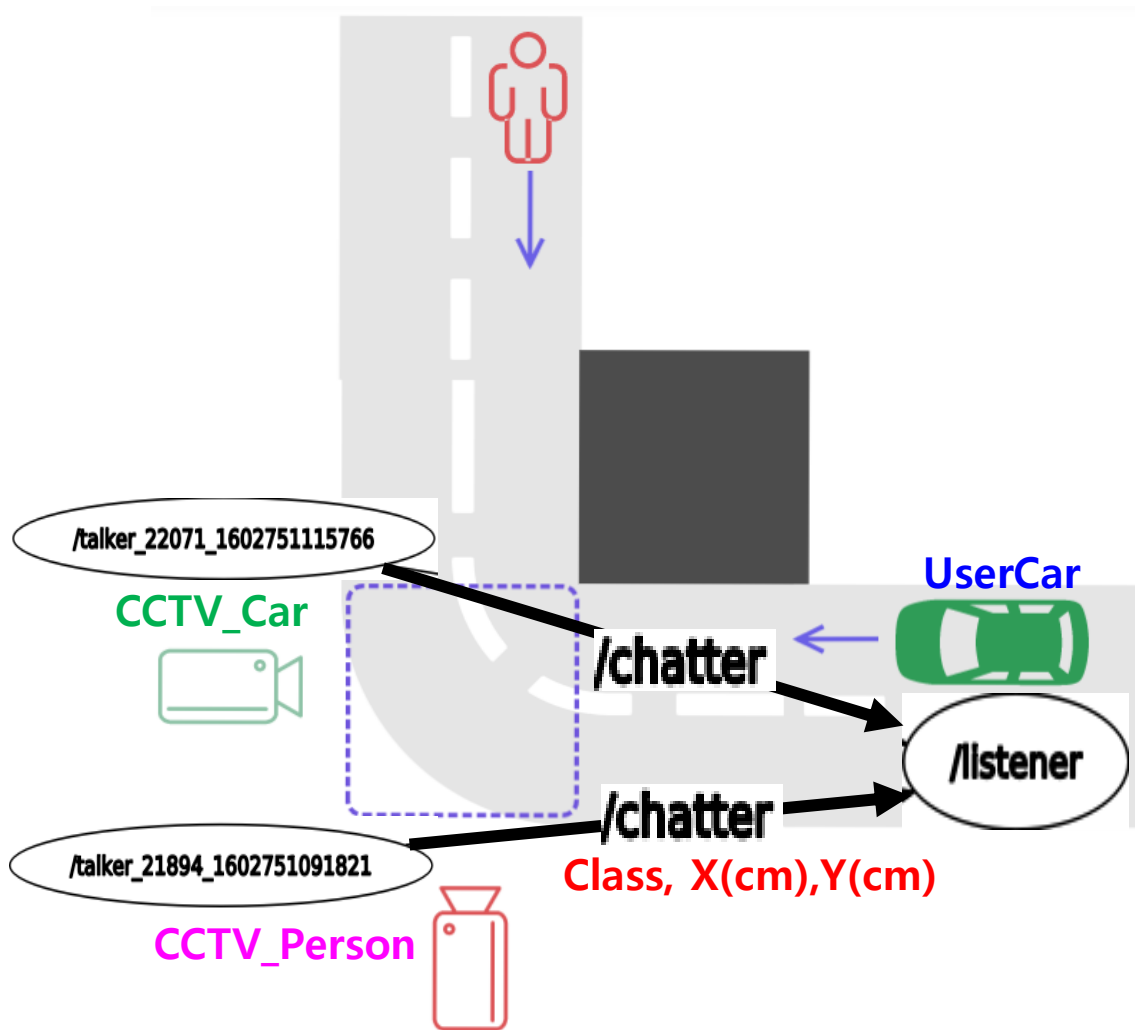


그림 4-8 rqt_graph

4. 프로젝트 결과

6. Kalman Filter

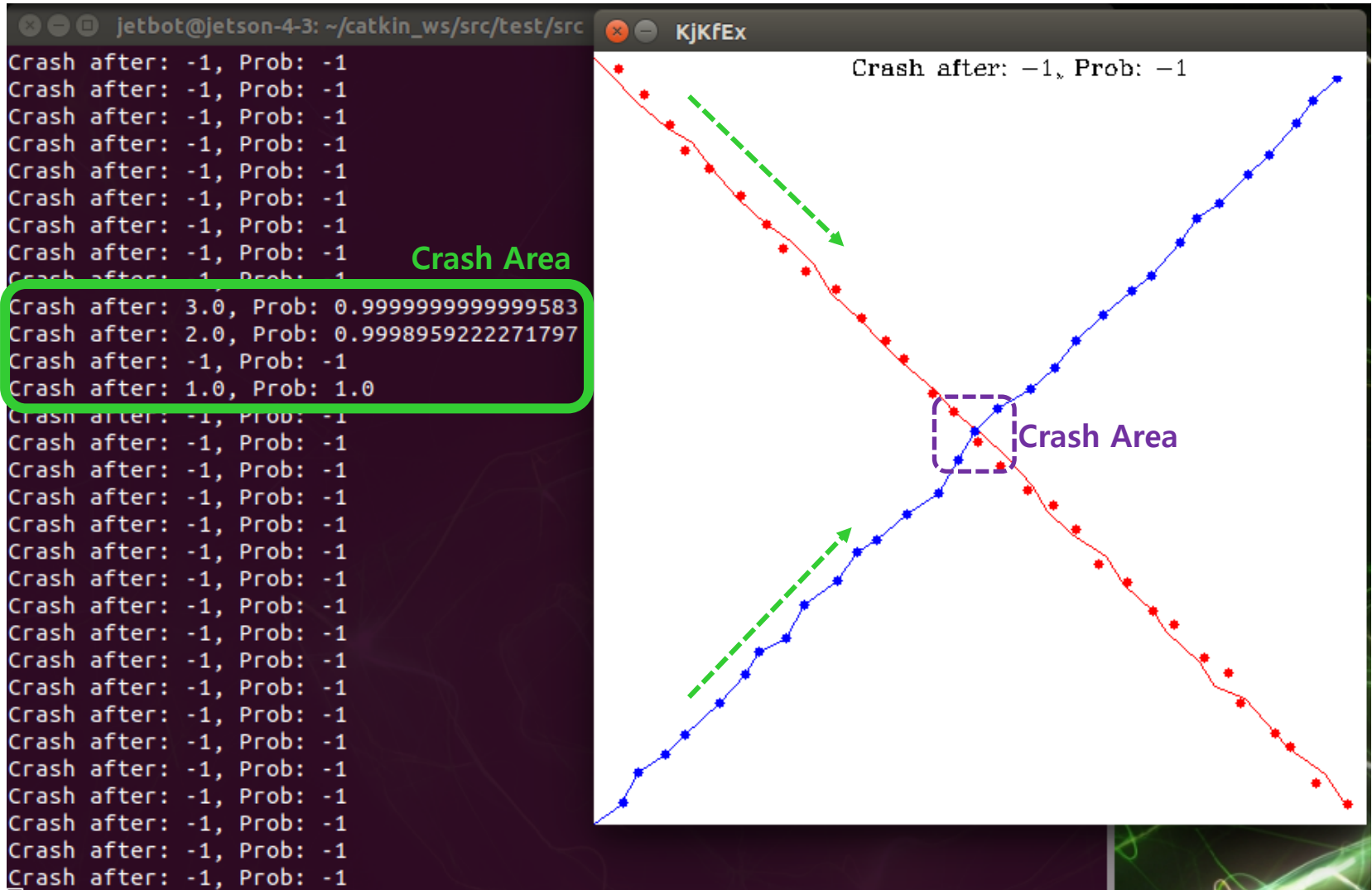


그림 4-9 모의 충돌예상시간, 충돌확률 출력

4. 프로젝트 결과

7. RX_Jetbot

```
jetbot@jetson-4-3: ~/catkin_ws/src/test/src
crashTime: -1 crashProbability: -1 robot run 1
object: car 177 183
crashTime: -1 crashProbability: -1 robot run 1
object: car 175 184
crashTime: -1 crashProbability: -1 robot run 1
object: car 173 183
crashTime: -1 crashProbability: -1 robot run 1
object: car 171 183
crashTime: -1 crashProbability: -1 robot run 1
object: car 169 184
crashTime: -1 crashProbability: -1 robot run 1
object: person -14 382
object: car 167 184
crashTime: -1 crashProbability: -1 robot run 1
object: person -14 366
crashTime: -1 crashProbability: -1 robot run 1
object: person -11 369
crashTime: -1 crashProbability: -1 robot run 1
object: car 165 184
crashTime: -1 crashProbability: -1 robot run 1
crashTime: 5.0 crashProbability: 1.0 robot crash 2
object: person -9 366
crashTime: 5.0 crashProbability: 0.9570486327382626 robot crash 2
object: person -7 362
crashTime: 5.0 crashProbability: 0.9638238796733909 robot crash 2
object: person -7 336
crashTime: 3.0 crashProbability: 0.6981532523402839 robot run 1
object: person -6 346
object: car 161 184
crashTime: 3.0 crashProbability: 0.6605295871360923 robot run 1
crashTime: 2.0 crashProbability: 0.9999627959217945 robot crash 2
object: person -10 293
object: car 159 184
crashTime: -1 crashProbability: -1 robot run 1
object: car 156 184
crashTime: -1 crashProbability: -1 robot run 1
object: person -10 286
crashTime: -1 crashProbability: -1 robot run 1
crashTime: -1 crashProbability: -1 robot run 1
```

그림 4-10 ROS 수신 후 충돌예상시간, 충돌확률 출력

4. 프로젝트 결과

7. RX_Jetbot

```
object: person -11 369
crashTime: -1 crashProbability: -1 robot run 1
object: car 165 184
crashTime: -1 crashProbability: -1 robot run 1
crashTime: 5.0 crashProbability: 1.0 robot crash 2
object: person -9 366
crashTime: 5.0 crashProbability: 0.9570486327382626 robot crash 2
object: person -7 362
crashTime: 5.0 crashProbability: 0.9638238796733909 robot crash 2
object: person -7 338
crashTime: 3.0 crashProbability: 0.6981532523402839 robot run 1
object: person -6 346
object: car 161 184
crashTime: 3.0 crashProbability: 0.6605295871360923 robot run 1
crashTime: 2.0 crashProbability: 0.9999627959217945 robot crash 2
object: person -10 293
object: car 159 184
crashTime: -1 crashProbability: -1 robot run 1
object: car 156 184
```

그림 4-11 ROS 수신 후 충돌예상시간, 충돌확률 출력

4. 프로젝트 결과

8. 시연 영상



5. 결과

1. 결과물의 활용 방안



그림 5-1 어린이 보호구역 사고 예방



그림 5-2 물류창고 작업자 사고 예방

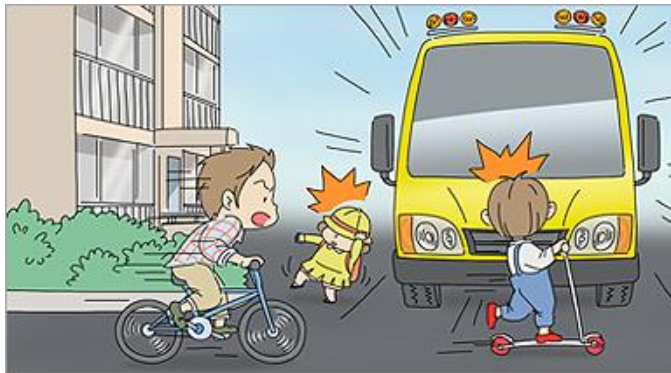


그림 5-4 아파트 단지 내 추돌사고 방지



그림 5-4 긴급 차량 추돌사고 방지

5. 결과

2. 향후 연구 방안



그림 5-6 빛 반사로 명도가 높아짐

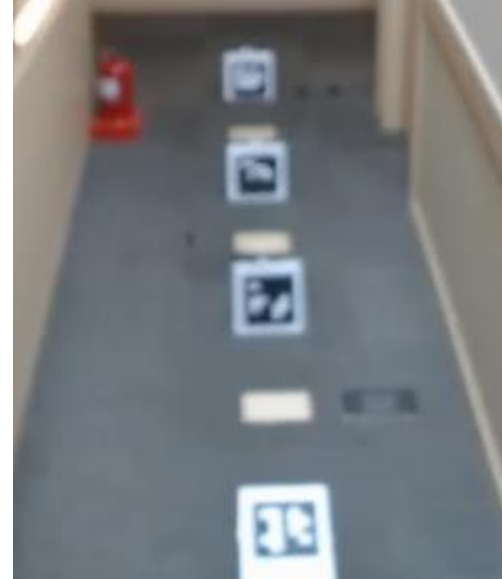


그림 5-7 카메라 Auto Focus 동작

※ 카메라 Image Detect, AR marker Detect 피사체의 상태에 영향을 받음

1. 빛 반사
2. 피사체의 각도에 따른 왜곡
3. 초점
4. 피사체의 해상도
5. 화면상의 크기

1. 개발 프로세스

Process	내용	상세내용	1주차	2주차	3주차	4주차	5주차	6주차	7주차	8주차	9주차	10주차	비고
초기	개발환경설정	개발 환경 Research											
Object Detection	Yolo3 / Yolo3-tiny 성능비교 (서버)	환경 정하기 / 구축 및 장비 셋팅(Xavier 2대, Jetbot 1대)											
		환경(data set / FPS/검출률/카메라 해상도, CCTV 성능이 1080p)											
		성능 비교후 문서화 해서 정리(for 최종보고서)											
거리측정	방법론 조사	모노카메라 거리 측정 알고리즘 조사											
		픽셀좌표 <-> BridEyeView 좌표 변환											
		자랑 위치추정											
좌표변환	변환할수 조사 (서버)	픽셀좌표 <-> BridEyeView 좌표 변환											
		구현											
		픽셀좌표 <-> BridEyeView 좌표간 변환행렬 최적화											
Tracking & Prediction	변환할수 조사 (서버)	CCTV기온 검출된 object 좌표 -> 자랑 기준 좌표로 변환											
		최적화 필요											
		구현											
Tracking & Prediction	Filtering 구현	Kalman Filter로 Noise Filtering 기능 구현											
		Prediction 구현 (자랑, 사람)											
		충족율들 가정 하에 n초 뒤 위치 예측											
충신	예제지 구현	Prediction 후 충돌 확률 계산 (자랑, 사람 위치 예측에 따름)											
		collision 모듈 오류 수정											
		구현											
충신	예제지 구현	그래프 출력 확인											
		Xavier <-> Jetbot 충신 환경 구축											
		Ros Wifi충신 환경 조사											
하드웨어	UserDisplay시각화	Object detection 이후 검출된 내용 실시간 전송											
		(class,(X1,Y1),(X2,Y2)... 좌표는 B-Box 의 좌표											
		구현											
최종	테스트	자랑에서 충신으로 받아오는 API 작성 / (parameter까지 다정하기)											
		기본 주행은 키보드로 제어											
		구현											
최종	문서화/발표준비	구현											
		구현											
		구현											

2. 개발 이슈 리스트

- 1. 멀티 쓰레드 작성시 ROS는 Main 함수에서 동작
- 2. 무선랜 사용시 ROS는 localhost가 아닌 IP주소 할당
- 3. ROS는 동일 네트워크에서 동작
- 4. 카메라 빛 반사, 각도, 초점, 해상도가 Detect에 영향
- 5. Queue사이즈 초과 시 Jetbot 강제종료
- 6. 카메라 한대로 사람(Yolo3), Color 동시 Detect 시 메모리 부족으로 프로그램 정지
 - > SW의 문제가 아닌 Jetson Xavier 프로세스 메모리 문제로 판단
- 7. 한 코드안에서 1번 카메라로 사람(Yolo3), 2번 카메라로 Color Detect시 먼저 실행중인 프로세스가 정지
 - > Ros로 프로세스 나눠서 실행
- 8. USB Hub 사용 시 Cam Number가 고정이 안되는 것
 - > USB Hub 사용하지 않고 물리적으로 직접연결
- 9. 충돌확률 계산 시 테일러 급수로 근사하면 오차가 발산
 - > ERF 방식으로 근사

3. BOM(Bills of Material) List

BOM LIST FOR Tracking and Route Prediction of Moving Objects in Blind Spot for Autonomous Driving

팀장 : 김건주


팀원 : 박장현, 이현영

Revised Date : 2020-10-21

필요 부품 리스트

NO	Part Name	Part Number	Parts Description	Manufacturer	UNIT	Q'ty	Prev
1	USB Hub	NEXT-707U3	USB3.0 허브 7포트 유선원 개별스위치	NEXT	EA	3	
2	USB Connector	NEXT-1512TC	USB B to C 젠더	NEXT	EA	3	
3	Camera	Logitec C922 PRO HD Webcam	-	Logitec	EA	2	
4	ROBOAT	JetBot AI Kit	Non fixed camera	WAVESHARE	SET	1	
5	Board	Jetson AGX Xavier	-	NVIDIA	EA	1	
6	WIFI RF	ip TIME A1000 Mini	IEEE 802.11ac/a/b/g/n	ipTIME	EA	2	
7	Keyboard	BloothKeyboard		Logitec	EA	1	

4. 회의록

	Object Detection	거리측정	좌표변환 	Tracking & Prediction	통신	H/W
Task	<ul style="list-style-type: none"> Yolo V3 Yolo V3-tiny 비교 	어떻게? DisNet FPS	함수찾아보기	Kalman Filter 시각화 Global 좌표	ROS data msg	PWM 제어 코딩
Issue	<div> <div> 카메라 (카메라 Pix) Deep learning 오차 허용 범위 EAEB 개발 한다 치고 <u>1M 앞 에 멈출 거 겠 다.</u> </div> <div> 통신 Data 형태 </div> <div> 여러 object 검출 ----- 동시 Tracking </div> <div> ROS? TCP/IP? </div> <div> Jetbot? Turtlebot? 최종 시나리오 어떻게 보여줄지? 무션 실시간 </div> </div>					

5. 사용 환경 및 GitHub

사용 환경

- Ubuntu 18.04 LTS
- OpenCV 3.4.0
- Cuda 10.0
- DarkNet Yolo3
- NVIDIA Xavier AGX
- NVIDIA JetsonNano
- ROS Melodic
- Logitech HD USB Camera
(1080p)

GitHub

<https://github.com/baek0307/IncidentPredictionSystem>

A low-angle, blue-tinted photograph of two men in business suits shaking hands. In the background, a world map is visible, with the Americas on the left and Europe/Africa on the right. The scene is brightly lit, creating a professional and positive atmosphere.

감 사 합 니 다.