

Holo Safety Sensor

Hologram과 AI의 만남, 안전한 사회를 만들어가다



목차

1. 문제 인식

2. 아이디어 제안

3. 아이디어 설명

4. 아이디어 사용 예시

5. 기대 효과

1. 문제 인식

2020 민식이법 적용 이후에도
사고건수/부상자수 증가

스쿨존 내 어린이(12세 이하) 교통사고

[출처: 도로교통공단 교통사고 분석 시스템]



경기 평택 청아초
① 굴착기 등 건설기계 통행
② 과속방지턱·안전펜스 등 설치
③ 어린이보호구역 눈에 띄지 않음(적색포장 미비)
경남 창원 영산초
① 신호등 없는 사거리
② 과속방지턱·안전펜스 일부 설치
③ 정문 앞 보도·펜스 미설치, 주정차 단속 미비
서울 강남 언북초
① 가파른 골목길, 보도 없는 구간
② 적색포장, 차량·보행자 진입 센서, 보도·안전펜스 설치
③ 없음

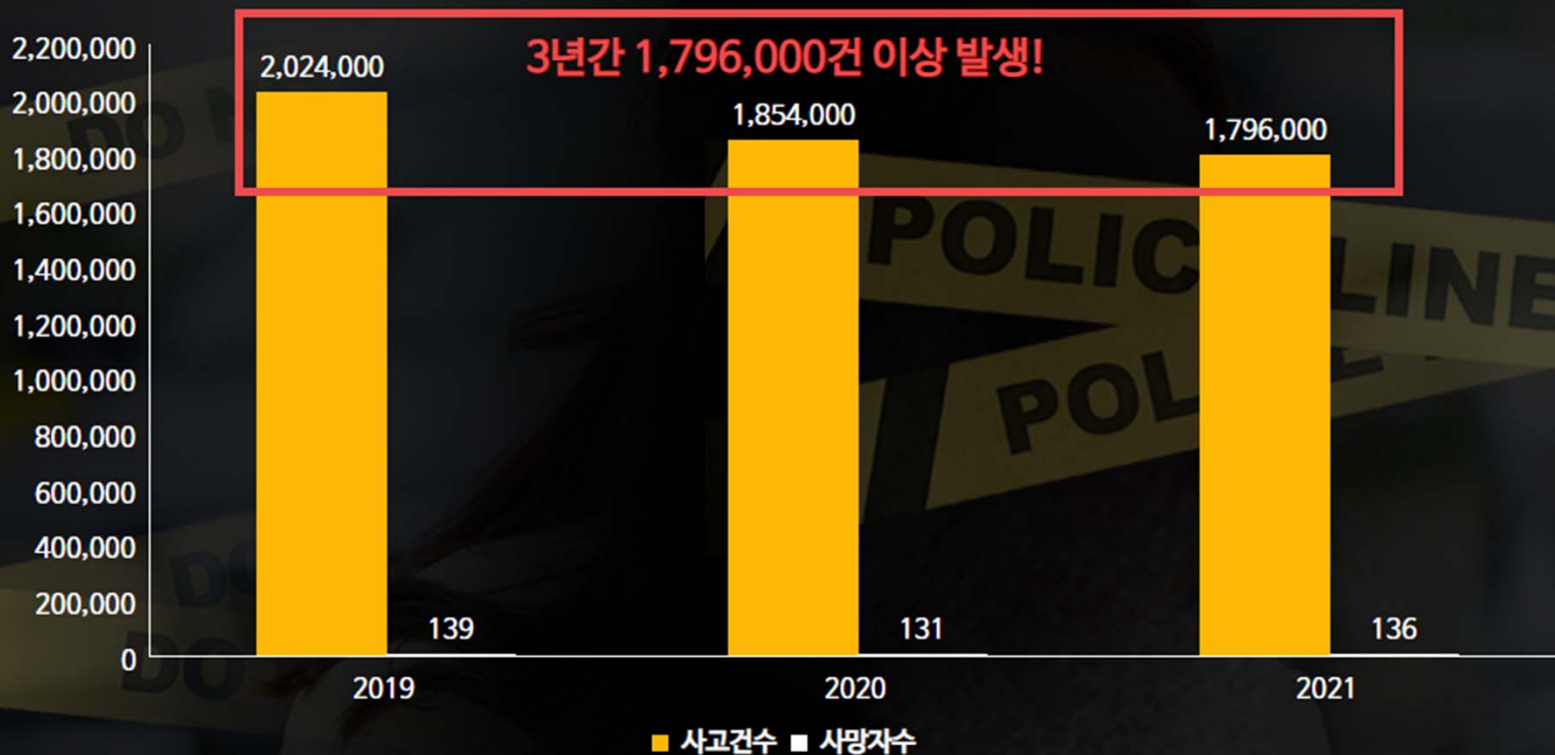
스쿨존 어린이 사망사고 대처 현황
※2021~2022년, 사고발생 시간순
① 사고 특징 ② 고쳐진 것 ③ 고쳐지지 않은 것
인천 중구 신광초
① 대형 화물차 통행, 복잡한 사거리
② 오후 1~4시 일부 구간 화물차 통행 제한
③ 통교시간 화물차 정상 운행, 신호등 미설치, 횡단보도 관리 부재, 차로 앞 학교 정문 방향 변동 없음
서울 영등포 도림초
① 골목길, 무단횡단 잦은 곳
② 불법 주정차 단속, 횡단보도·단속카메라 설치
③ 안전펜스 미설치, 차량 통제 없음

도로교통공단 교통사고 분석 시스템에 따르면, 2021년 스쿨존 내 어린이 교통사고 건수 및 부상자 수가 2013년에 비해 28.5% 증가

한겨레 스쿨존 안전 시설 분석에 따르면, 2021~2022년에 발생한 5개의 스쿨존 사망 사고 현장 모두 안전시설이 미흡

우회전 차량에 의한 교통사고

[출처 : 도로교통공단 / 단위: 명]



도로교통공단 교통사고 통계에 따르면, 우리나라 인구 10만 명당 보행 중 사망자 수 2.5명(2019년 기준)으로 **OECD 회원국 평균에 비해 2.3배 많은 수준**

경찰청과 도로교통공단 교통사고 분석에 따르면, 횡단보도 위 보행자 교통사고 중 우회전 차량에 의한 사고가 **59.4%를 차지**

1. 문제 인식

3p

도로교통공단 경기도지부가 최근 5년간 경기지역에서 발생한 빗길 교통사고를 분석한 결과, 빗길 야간 교통사고 비율은 54.5%로 맑은 날 야간(39.3%)에 비해 약 15% 높음

출처 : 중부일보 - 경기·인천의 든든한 친구(<http://www.joongboo.com>)

도로교통공단이 2019년부터 2021년까지 3년간 3~5월 교통사고를 분석한 결과, 졸음운전으로 인한 사고는 총 1,833건으로 47명 사망, 3,423명 부상

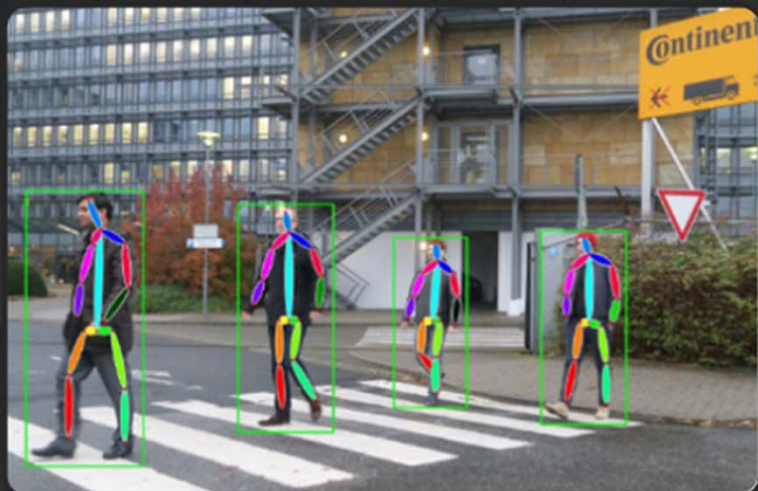
출처 : 강원일보 - <https://m.kwnews.co.kr/page/view/2023032715172797443>

운전자가 보행자를 사전 인지한다면?

운전자가 보행자의 행동을
★ 예측하는 시스템 도입 ★



LSTM 모션 인식 AI 모델



AI 모델이란?

→ 컴퓨터가 스스로 외부 데이터를 조합,
분석하여 학습하여
주어진 문제를 해결하는 모델

[LSTM 모델]

→ 시간에 따라 연속적인 모션 데이터를
학습하여 대상의 행동을
파악할 수 있는 RNN 모델 선택



홀로그래픽 디스플레이



홀로그램이란?

→ 홀로그래피에서, 입체상을 재현하는
간섭 줄무늬를 기록한 매체

1 회절된 빛의 간섭을 컴퓨터로 계산

2 간섭무늬를 통해 3D 이미지 형성

Holo Safety Sensor

“홀로그램 구현을 통해 교통사고를 예방할 수 있는 센서”

AI 모델이 횡단보도의 일정 반경 내 보행자의 Run, Walk, Stop Motion을 인식



Run(0)/Walk(1) Motion 정보가 입력될 경우 운전자가 볼 수 있도록 홀로그램 영상 구현



운전자가 실제와 같은 홀로그램을 보고, 보행자가 있음을 미리 인지해 교통사고 예방

→ 시간에 따라 연속적인 보행 데이터를 학습하여 대상의 행동을 파악할 수 있는 RNN 모델 선택

2. 간섭무늬를 통해 3D 이미지 형성

LSTM 모델

1

미디어 데이터(이미지, 비디오)를 처리하고 분석하는 Tool인 미디어파이프(MediaPipe) 활용

2

보행자들의 움직임을 Run(0), Walk(1), Stop(2)으로 분류해 데이터 확보

4

학습한 모델을 이용해 사고 다발 횡단보도의 보행자 motion을 인식해 Run(0), Walk(1), Stop(2) 정보를 디스플레이에 전달

3

확보한 데이터를 기반으로 LSTM 모델이 보행자의 달리기, 걷기, 정지 학습

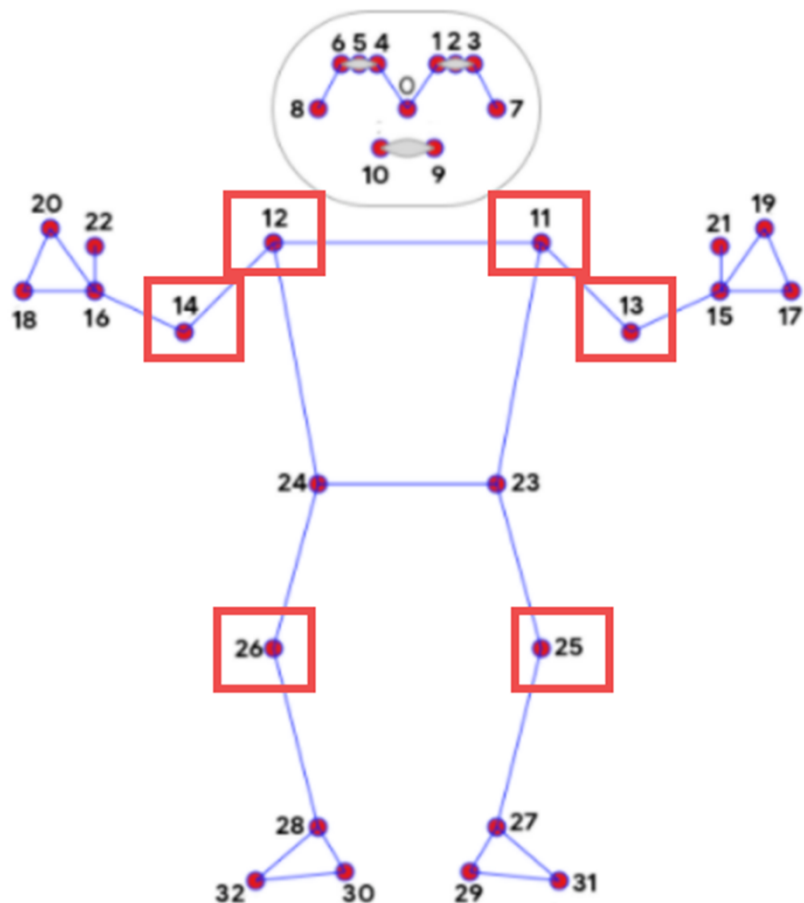


디스플레이에 전달

관절을 기준으로 각도를 계산하기 위한 함수

12번, 14번, 16번 landmark

오른쪽 어깨, 오른쪽 팔꿈치, 오른쪽 손목 landmark angle 값 계산



- | | |
|--------------------|----------------------|
| 0. nose | 17. left_pinky |
| 1. left_eye_inner | 18. right_pinky |
| 2. left_eye | 19. left_index |
| 3. left_eye_outer | 20. right_index |
| 4. right_eye_inner | 21. left_thumb |
| 5. right_eye | 22. right_thumb |
| 6. right_eye_outer | 23. left_hip |
| 7. left_ear | 24. right_hip |
| 8. right_ear | 25. left_knee |
| 9. mouth_left | 26. right_knee |
| 10. mouth_right | 27. left_ankle |
| 11. left_shoulder | 28. right_ankle |
| 12. right_shoulder | 29. left_heel |
| 13. left_elbow | 30. right_heel |
| 14. right_elbow | 31. left_foot_index |
| 15. left_wrist | 32. right_foot_index |
| 16. right_wrist | |

33개의 포즈 랜드마크들

```
landmarks[mp_pose.PoseLandmark.LEFT_ELBOW.value],  
landmarks[mp_pose.PoseLandmark.LEFT_WRIST.value])
```

신체의 관절 기준으로 구부러진 각도를
계산하는 코드 작성

Base Code

관절을 기준으로 각도를
계산하기 위한 함수

```
# 앵글 계산 함수
def calculateAngle(landmark1, landmark2, landmark3):

    # 랜드마크 좌표
    x1, y1, _ = landmark1
    x2, y2, _ = landmark2
    x3, y3, _ = landmark3

    # 세 점 사이 각도 계산
    angle = math.degrees(math.atan2(y3 - y2, x3 - x2)
        - math.atan2(y1 - y2, x1 - x2))

    if angle < 0:

        angle += 360

    return angle
```

각도 계산식

[] # 분류 함수

```
def classifyPose(landmarks, output_image, display=False):
```

```
# Initialize the label of the pose. It is not known at this stage.
label = 'Stop'
```

```
# Specify the color (Red)
color = (0, 0, 255)
```

```
# 11번, 13번, 15번 landmark
# 왼쪽 어깨, 왼쪽 팔꿈치, 왼쪽 손목 landmark angle 값 계산
left_elbow_angle = calculateAngle(
    landmarks[mp_pose.PoseLandmark.LEFT_SHOULDER.value],
    landmarks[mp_pose.PoseLandmark.LEFT_ELBOW.value],
    landmarks[mp_pose.PoseLandmark.LEFT_WRIST.value])
```

Stop(2) 상태가 default 값

```
# 12번, 14번, 16번 landmark
# 오른쪽 어깨, 오른쪽 팔꿈치, 오른쪽 손목 landmark angle 값 계산
right_elbow_angle = calculateAngle(
    landmarks[mp_pose.PoseLandmark.RIGHT_SHOULDER.value],
    landmarks[mp_pose.PoseLandmark.RIGHT_ELBOW.value],
    landmarks[mp_pose.PoseLandmark.RIGHT_WRIST.value])
```

```
# 13번, 15번, 23번 landmark
# 왼쪽 어깨, 왼쪽 팔꿈치, 왼쪽 엉덩이, landmark angle 값 계산
left_shoulder_angle = calculateAngle(
    landmarks[mp_pose.PoseLandmark.LEFT_ELBOW.value],
    landmarks[mp_pose.PoseLandmark.LEFT_SHOULDER.value],
    landmarks[mp_pose.PoseLandmark.LEFT_HIP.value])
```

```
# 12번, 14번, 24번 landmark
# 오른쪽 어깨, 오른쪽 팔꿈치, 오른쪽 엉덩이 landmark angle 값 계산
right_shoulder_angle = calculateAngle(
    landmarks[mp_pose.PoseLandmark.RIGHT_HIP.value],
    landmarks[mp_pose.PoseLandmark.RIGHT_SHOULDER.value],
    landmarks[mp_pose.PoseLandmark.RIGHT_ELBOW.value])
```

```
# 23번, 25번, 27번 landmark
# 왼쪽 엉덩이, 왼쪽 무릎, 왼쪽 발목 landmark angle 값 계산
left_knee_angle = calculateAngle(
    landmarks[mp_pose.PoseLandmark.LEFT_HIP.value],
    landmarks[mp_pose.PoseLandmark.LEFT_KNEE.value],
    landmarks[mp_pose.PoseLandmark.LEFT_ANKLE.value])
```

```
# 24번, 26번, 28번 landmark
# 오른쪽 엉덩이, 오른쪽 무릎, 오른쪽 발목 landmark angle 값 계산
right_knee_angle = calculateAngle(
    landmarks[mp_pose.PoseLandmark.RIGHT_HIP.value],
    landmarks[mp_pose.PoseLandmark.RIGHT_KNEE.value],
    landmarks[mp_pose.PoseLandmark.RIGHT_ANKLE.value])
```

신체의 관절 기준으로 구부러진 각도를
계산하는 코드 작성

Base Code

[] # 물체가 달리고 있을경우 / Run일 경우

왼쪽, 오른쪽 팔, 어깨, 팔꿈치의 각도가 30 ~ 90도 사이에 위치
if left_elbow_angle > 30 and left_elbow_angle < 90 and
right_elbow_angle > 30 and right_elbow_angle < 90:

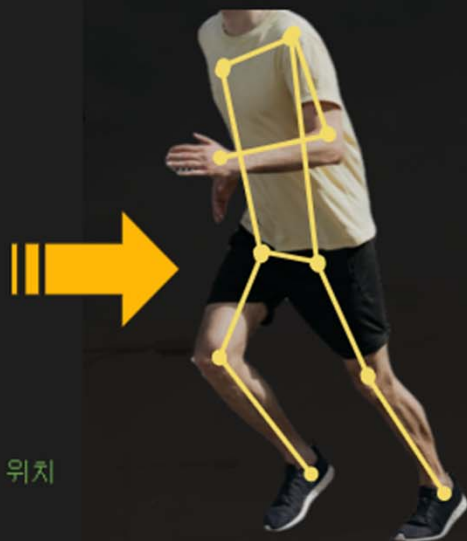
양 팔의 팔꿈치, 어깨, 엉덩이까지 각도가 20 ~ 70도 사이에 위치
if left_shoulder_angle > 20 and left_shoulder_angle < 70 and
right_shoulder_angle > 20 and right_shoulder_angle < 70:

#왼쪽 엉덩이, 왼쪽 무릎, 왼쪽 발목까지 각도가 10 ~ 70도 사이에 위치
if left_knee_angle > 10 and left_knee_angle < 70:

#오른쪽 엉덩이, 오른쪽 무릎, 오른쪽 발목까지 각도가 140~180도 사이에 위치
if right_knee_angle > 140 and right_knee_angle < 180:

label = 'Run'

달리는 모션 인식 코드



[] # 물체가 걷고 있을경우 / Walk일 경우

왼쪽, 오른쪽 팔, 어깨, 팔꿈치의 각도가 120 ~ 180도 사이에 위치
if left_elbow_angle > 120 and left_elbow_angle < 180
and right_elbow_angle > 120 and right_elbow_angle < 180:

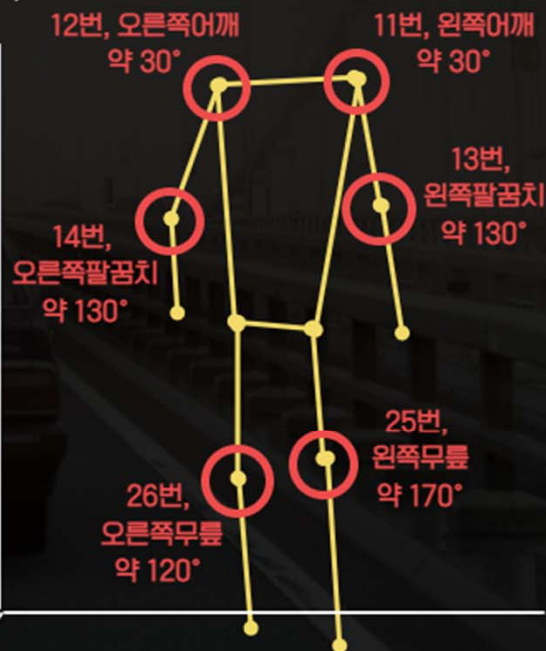
양 팔의 팔꿈치, 어깨, 엉덩이까지 각도가 20 ~ 70도 사이에 위치
if left_shoulder_angle > 20 and left_shoulder_angle < 70
and right_shoulder_angle > 20 and right_shoulder_angle < 70:

#왼쪽 엉덩이, 왼쪽 무릎, 왼쪽 발목까지 각도가 140 ~ 180도 사이에 위치
if left_knee_angle > 140 and left_knee_angle < 180:

#오른쪽 엉덩이, 오른쪽 무릎, 오른쪽 발목까지 각도가 140~180도 사이에 위치
if right_knee_angle > 140 and right_knee_angle < 180:

label = 'Walk'

걷는 모션 인식 코드



Base Code

Run, Walk, Stop을
0/1배열로 표현

```
[ ] # 레이블 원핫인코딩
from tensorflow.keras.utils import to_categorical

y_data = to_categorical(labels, num_classes=len(actions))

[ ] from sklearn.model_selection import train_test_split

x_data = x_data.astype(np.float32)
y_data = y_data.astype(np.float32)

x_train, x_val, y_train, y_val = train_test_split(x_data, y_data,
                                                test_size=0.1,
                                                random_state=2023)
```

저장된 데이터를
학습/검증으로 분리

```
[ ] # 모델 학습

from tensorflow.keras.callbacks import ModelCheckpoint, ReduceLR0nPlateau

history = model.fit(
    x_train,
    y_train,
    validation_data=(x_val, y_val),
    epochs=200,
    callbacks=[
        ModelCheckpoint('models', monitor='val_acc', verbose=1,
                        save_best_only=True, mode='auto'),
        ReduceLR0nPlateau(monitor='val_acc', factor=0.5, patience=50,
                          verbose=1, mode='auto')
    ]
)
```

200번 반복해 학습하도록 설정
정확도가 가장 높을때 학습 종료

```
[ ] from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense

model = Sequential([
    LSTM(64, activation='relu', input_shape=x_train.shape[1:3]),
    Dense(32, activation='relu'),
    Dense(len(actions), activation='softmax')
])

#x_train.shape[1:3] = (30,99) (window크기, 랜드마크+visibility+각도)

model.compile(optimizer='adam', loss='categorical_crossentropy',
              metrics=['acc'])
model.summary()
```

LSTM 모델의 출력층에
Run, Walk, Stop 3개의 레이블에 대해
출력 가능하도록 layer 연결

손실함수와 옵티마이저를 설정하고 compile

Holographic Display 구조

J. An et al., "Slim-Panel Holographic Video Display" Nat. Commun. 11, 5568

[Spatial Light Modulator]

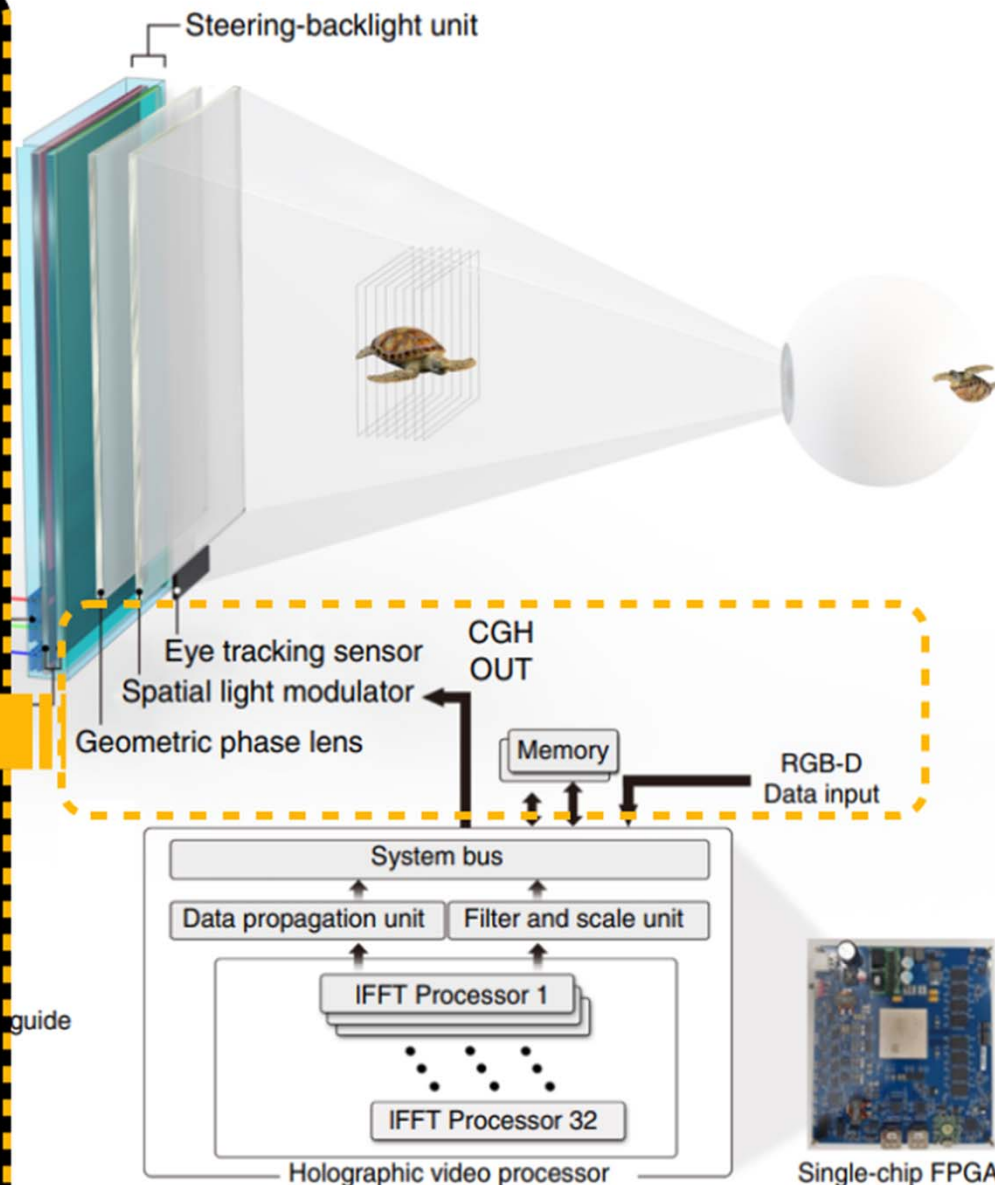
픽셀별로 뒤에서 투과하는 레이저 빛의 위상을 픽셀 별로 변조해 원하는 상을 구현

[Geometric Phase Lens]

홀로그램을 볼 수 있는 거리를 기존 대비 25배 줄일 수 있는 얇은 박막 렌즈

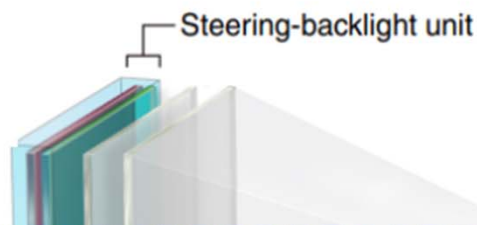
[Computer Generated Hologram]

홀로그램 간섭 무늬를 계산해 생성하는 기술인 CGH를 단일 칩셋으로 구성해 연산 효율 향상



Holographic Display 구조

J. An et al., "Slim-Panel Holographic Video Display" Nat. Commun. 11, 5568

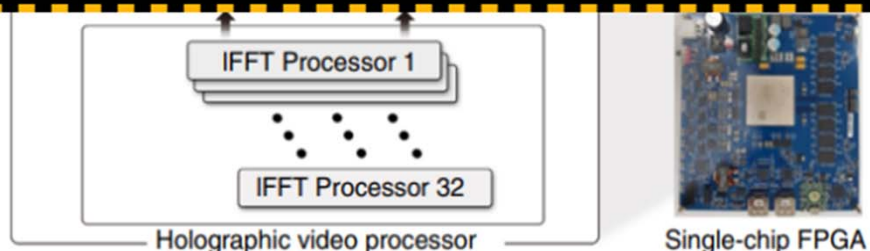
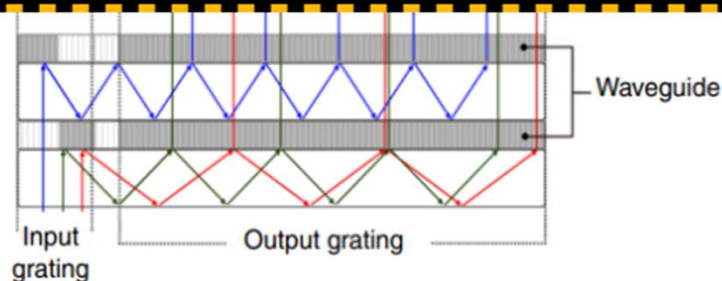


[홀로그래픽 디스플레이의 한계점]

$$\text{시야각} \uparrow \times \text{상크기} \downarrow = \text{픽셀수} \times \text{파장}$$

Trade-Off

Constant

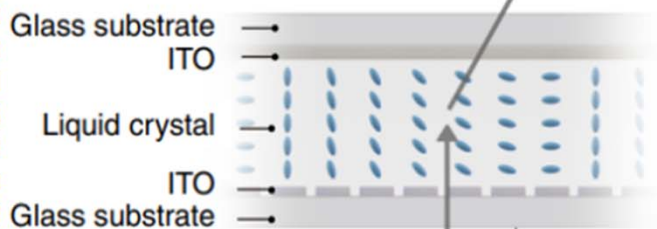


Holographic Display 구조

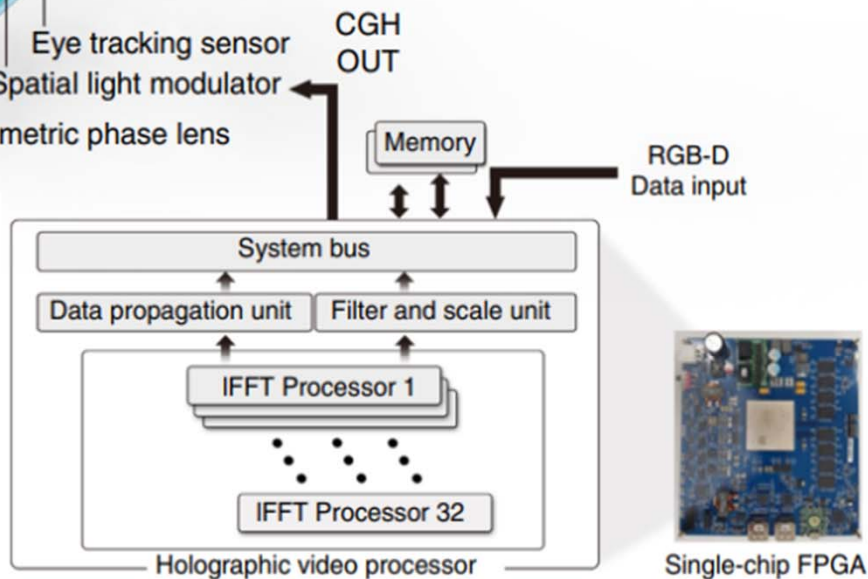
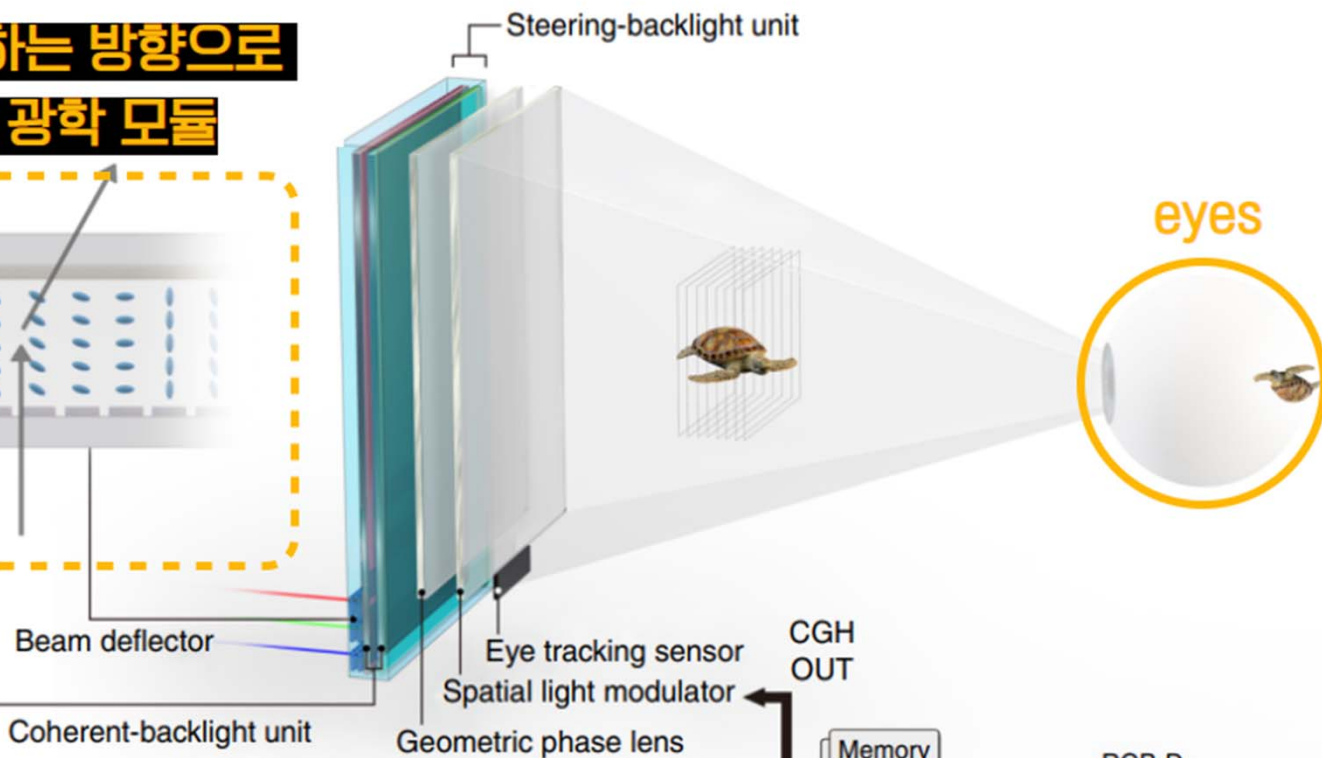
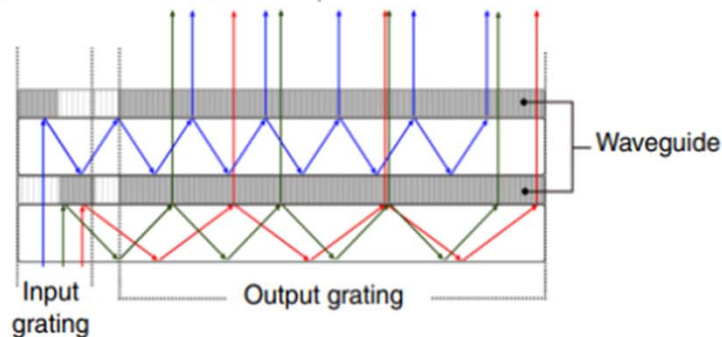
J. An et al., "Slim-Panel Holographic Video Display" Nat. Commun. 11, 5568

액정 기반 빛을 원하는 방향으로

조절할 수 있는 광학 모듈

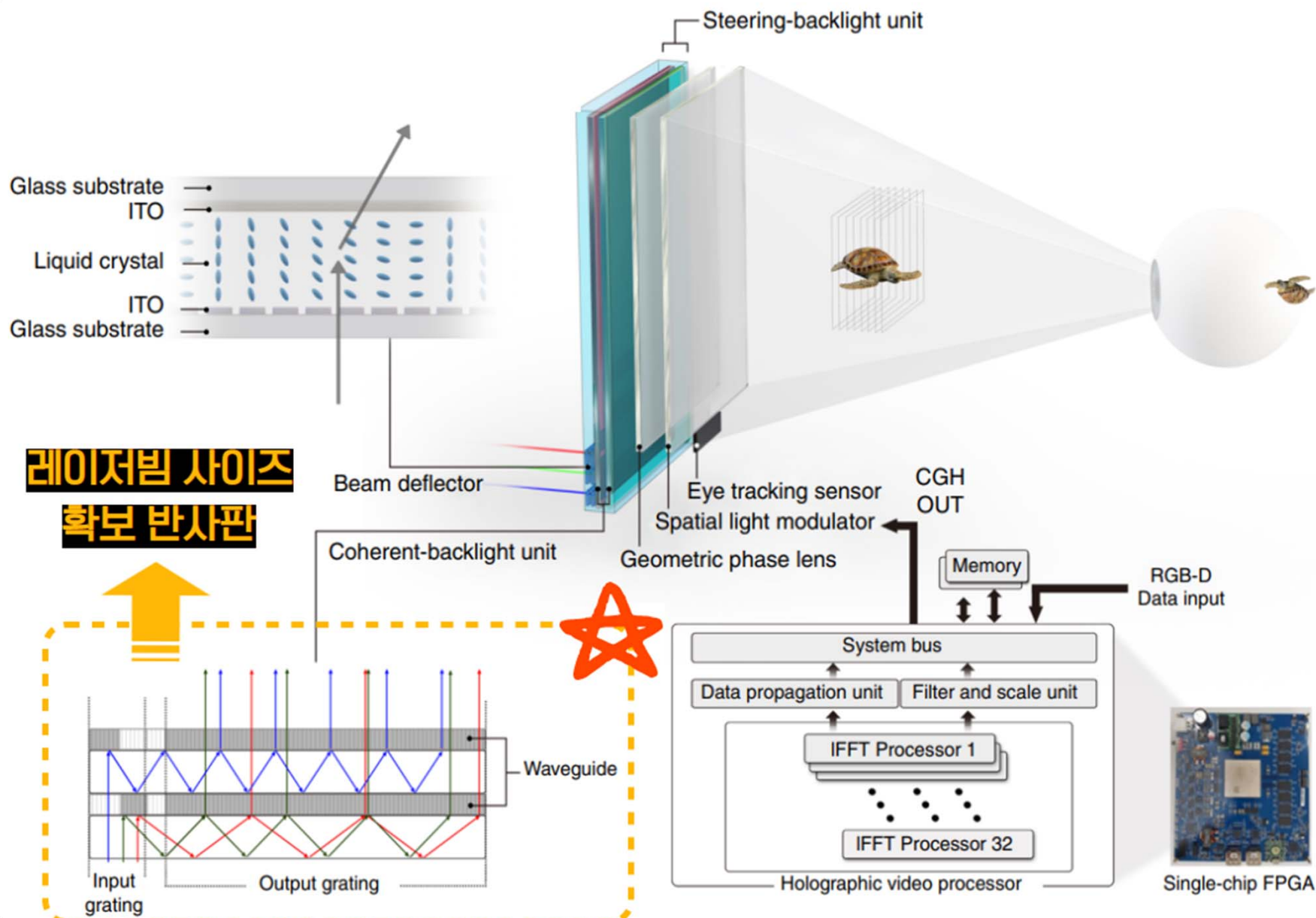


**픽셀수 증가로
시야각 & 상 크기
한계 극복**



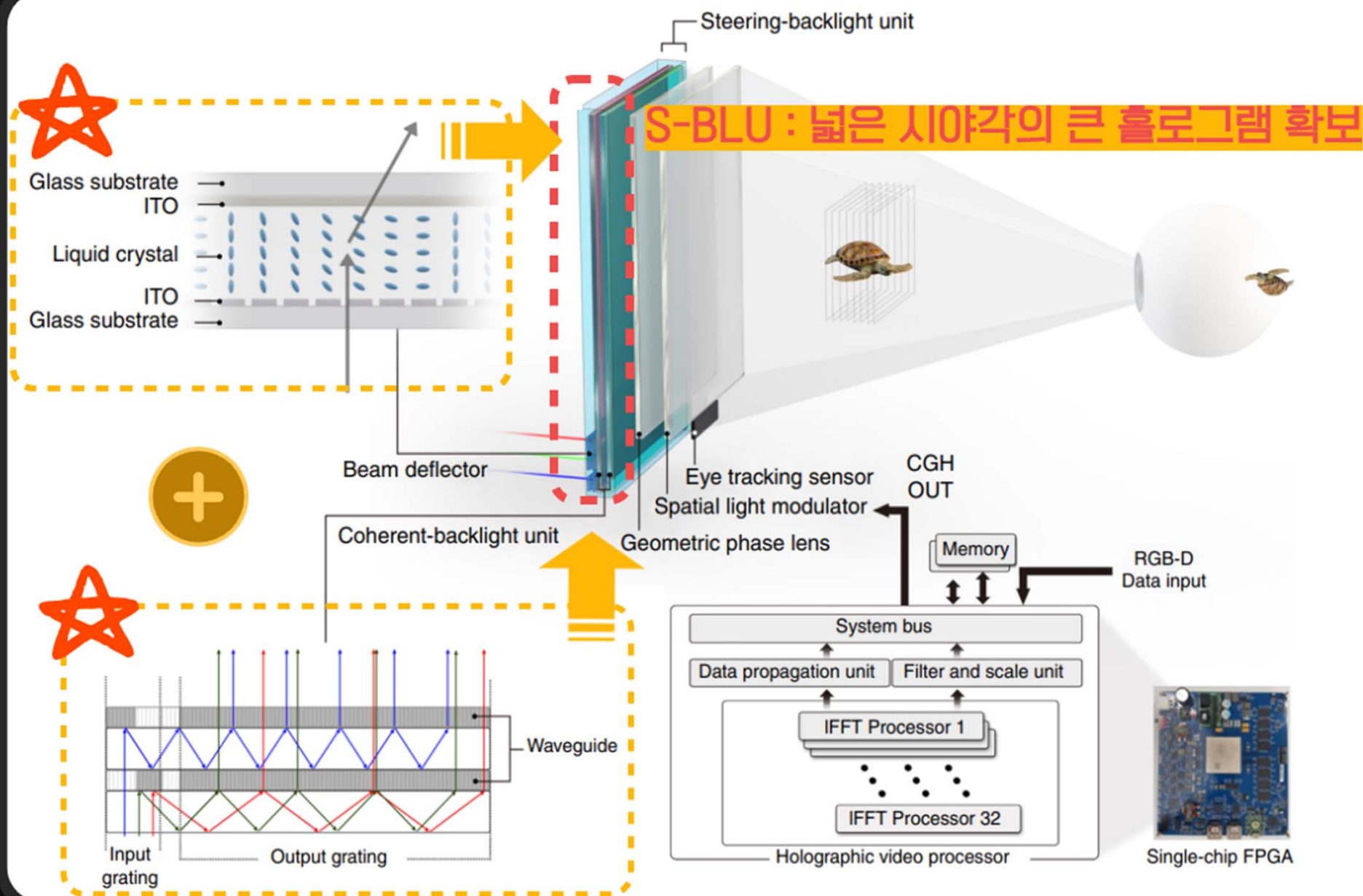
Holographic Display 구조

J. An et al., "Slim-Panel Holographic Video Display" Nat. Commun. 11, 5568

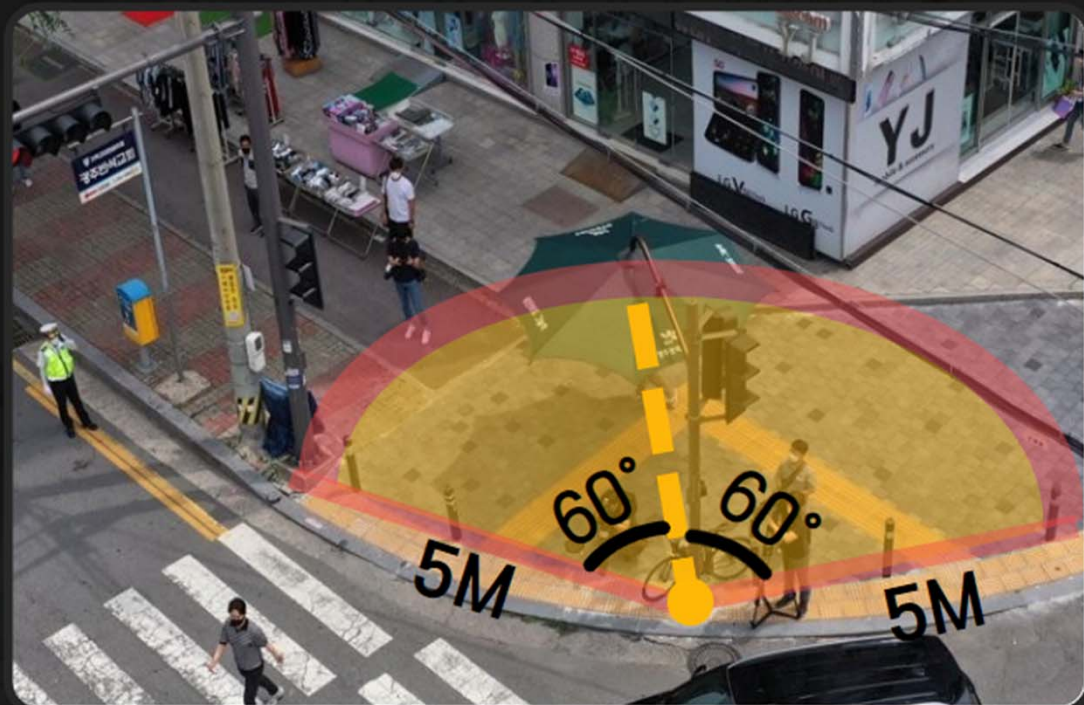


Holographic Display 구조

J. An et al., "Slim-Panel Holographic Video Display" Nat. Commun. 11, 5568



HSS 설치 방법



60° 시야각 모션 인식
카메라 2대



홀로그래픽 디스플레이 1대



- ① 초등학교 근처 - 학생들의 키를 고려해 1~1.5m 높이로 설치
- ② 교통섬 및 일반 횡단보도 - 1.5~2m 높이로 설치

HSS 알림 시스템



"전방 20m 앞 홀로그램 세이프티 센서가 존재합니다. 보행자 홀로그램을 주의해주세요"

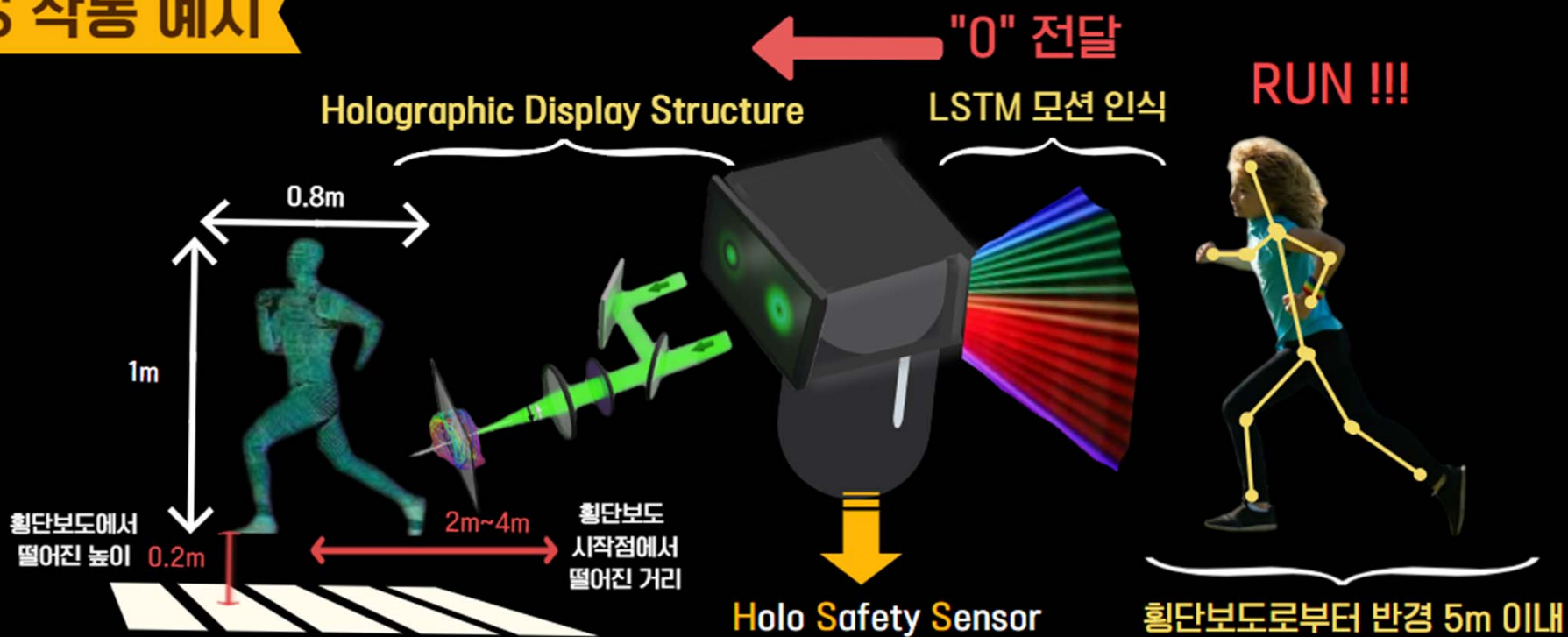
UWB 기술

고주파수에서 전파를 통해 작동하는 단거리 무선 통신기술인 UWB를 사용해 약 10m 근방의 HSS와 통신

① 운전자에게 홀로그램 세이프티 센서가 존재함을 미리 알려 홀로그램으로 인해 운전자가 급정거하거나 차량 간 사고나는 것을 방지

② 홀로그램 세이프티 센서 홍보 효과를 주고, 운전자들에게 경각심을 주는 역할을 함

HSS 작동 예시



HSS 응용



휠체어 인식

휠체어의 형태 및 휠체어 탑승자의 움직이는 모션을 인식해 홀로그램 구현



전동 킥보드 인식

전동 킥보드 형태와 헬멧과 같은 안전 장치 및 탑승자의 고정된 포즈를 인식해 홀로그램 구현

YOLO V8 객체 인식 모델을 추가로 사용해 휠체어 인식

STEP 분석

Social

비오는 밤길, 사각지대 위치에서 보행자를 운전자가 미리 인지



보행자와 운전자 간 교통사고 발생률 감소 및 교통사고 관련 불안감 해소

S

Technical

모션 인식을 통해 각 상황에 맞게 빠른 속도로 모션에 대한 정보를 전달



홀로그램으로 움직이는 3D 입체영상을 구현해 효과적으로 상황을 전달

T

Economical

① HSS 도입을 통해 무인단속카메라 등 고가의 안전시설 설치를 대체하여 교통사고 억제 비용 절감

② HSS 도입을 통해 교통 사고를 억제해 경찰 초동 조사 비용(약 533억), 구호 비용(약 238억) 등 인건비 절감

E

Political

민식이법 및 강화된 우회전 차량의 도로교통법 시행규칙 등 교통법규에 대한 서포터로서 역할



운전자와 보행자 간 교통사고 발생을 감소시키는 시너지 효과 기대

P

SAFETY

DISPLAY

SMART



A close-up photograph of a hand gripping a steering wheel. The image is dark and has a semi-transparent watermark that reads 'THANK YOU' in large, bold, capital letters across the center. The Korean text '감사합니다' is overlaid on the watermark.

THANK YOU

감사합니다