

## 목차

1. 프로젝트 개요

2. 시스템 구성

3. 알고리즘 및 하드웨어 구성

4. 정확도 개선 과정

5. 데모 영상

6. 향후 개선 방향

# 1. 프로젝트 개요

## 개요

4채널 마이크 배열을 이용하여 사람의 음성 방향을 추정

GCC-PHAT 기반 TDOA 계산

Subsample interpolation으로 정밀도 향상

GUI를 통해 실시간 위치 시각화

최종 목표: 발화 방향을 자동으로 추적하는 카메라 시스템 구축

## 프로젝트 필요성

사람 음성은 지속적으로 방향이 변함

강의실, 회의실, 원격 협업 환경에서는 발화자가 계속 움직임  
카메라가 자동으로 따라간다면 발표 촬영 품질 대폭 개선

기존 카메라 자동추적 방식의 한계

얼굴 인식 기반 추적은 가림·역광에 취약  
음성 기반 추적은 비가시·가림 상황에서도 안정적

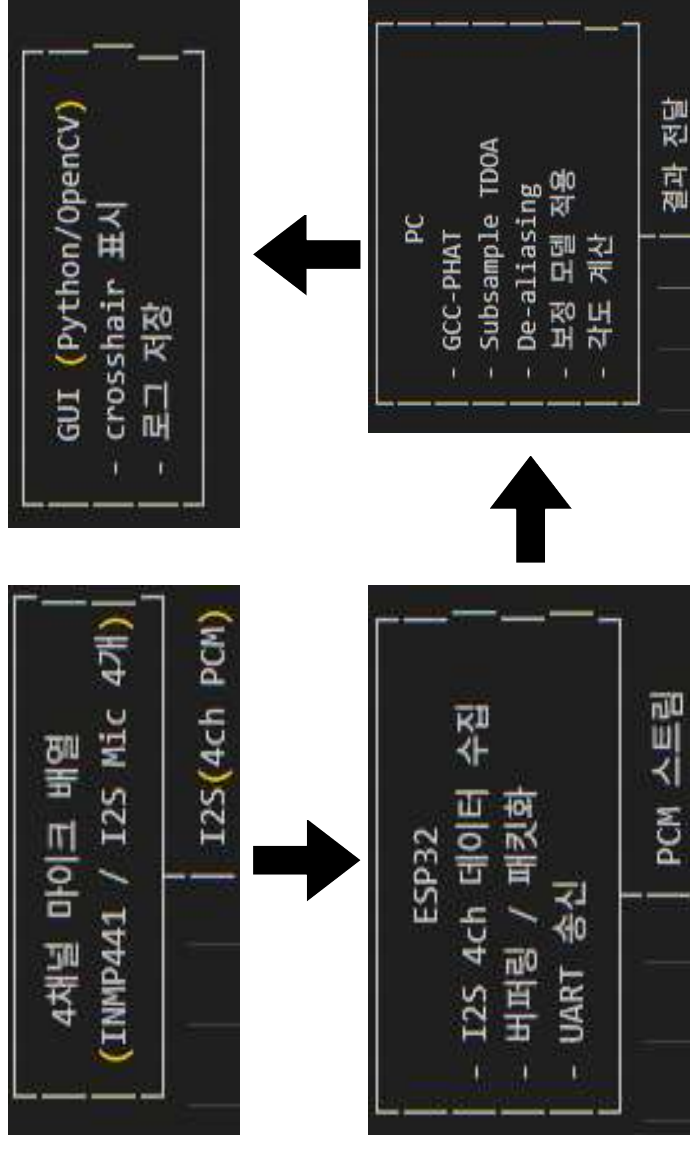
TDOA 방식은 비용 대비 효율이 높음

싱글 보드 컴퓨터 + 4개의 마이크만으로 방향 추정 가능

## 2. 시스템 구성

### 구성 및 다이어그램

- ESP32 → I2S 4채널 PCM 획득
- PC로 전송 - 고속시리얼 통신
  - 128KB/s의 데이터를 전송하기 위해 1.5Mbps 고속 시리얼 통신과 바이너리 패킷 프로토콜을 적용
- GCC-PHAT 기반 Cross-correlation 계산
- Subsample TDOA 추정
- OpenCV GUI에서 시각화



### 3. 알고리즘 및 하드웨어 구성

#### 사용한 알고리즘

##### (1) GCC-PHAT

두 마이크로 간 시간지연( $\tau$ )을 계산하는 기본 알고리즘  
PHAT 가중치로 잡음 및 잔향 환경에서 강건  
Cross-correlation peak 위치가 음원 도달 시간 차이(TDOA)

##### (2) Subsample TDOA

샘플 단위 해상도 한계를 극복하기 위해  
피크 근처 3개 값을 포물선 보간

결과: Fractional delay → 각도 오차 40~70% 감소

##### (3) De-aliasing & 대역 제한

300~3000Hz 대역이 정보가 가장 강함  
공간 에일리어싱제거

##### (4) 선형 보정 모델

$$\theta_{corrected} = a \cdot \theta_{raw} + b$$

실측 데이터 기반 회귀분석  
마이크 부착 오차로 인한 오류 감소

### 3. 알고리즘 및 하드웨어 구성

#### 하드웨어 구성

##### 마이크 배열

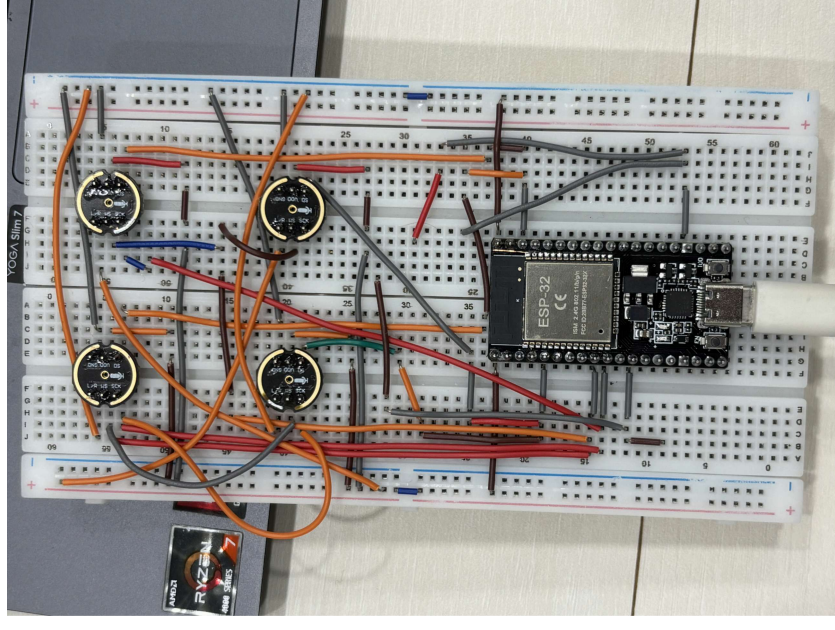
- 4채널 정사각형 배열
- half-side = 3.5cm
- I2S INMP441 4개 사용

##### ESP32

- I2S 멀티채널 프레임 수집 - I2S 동기화 문제
- ESP32의 독립적인 I2S 클럭으로 인한 샘플 시간 오프셋 문제를 해결
- 인터럽트 비활성화 후 두 I2S 포트를 동시에 시작, 동기화를 최대화
- Serial로 PC에 데이터 송신

##### PC

- Python 기반 신호처리(GCC-PHAT, TDOA)
- OpenCV GUI 실행



## 4. 정확도 개선 과정

과정별 가상 성능 수치(임의 샘플로 알고리즘 성능만 평가)

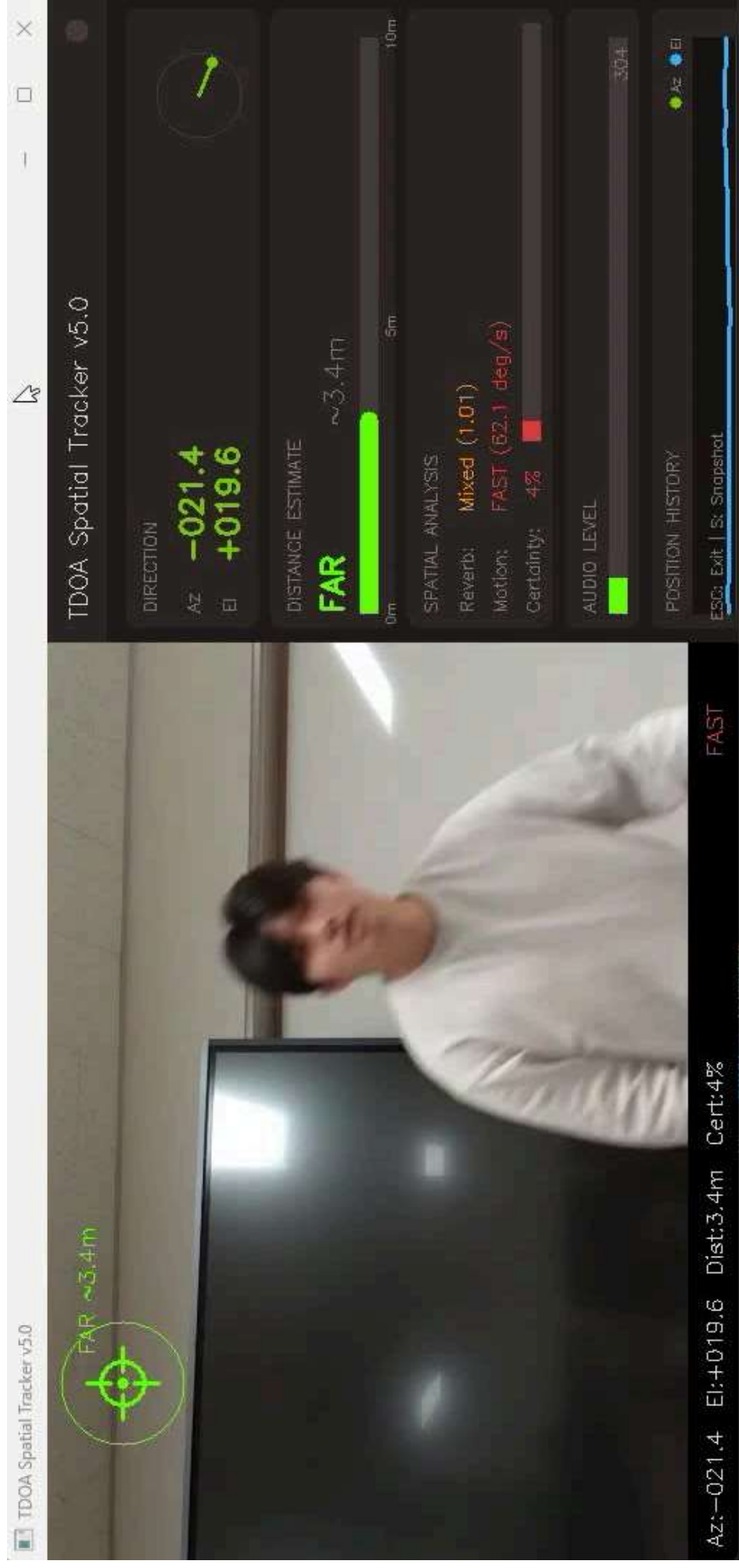
단계	알고리즘 / 설정	Azimuth 평균 오차	Elevation 평균 오차	2D 평균 오차	비고
① 초기	정수 TDOA, PHAT X	18°	25°	31°	가장 단순한 버전, 실사용 어려움
② GCC-PHAT	정수 TDOA + PHAT	14°	22°	26°	피크가 분명해져 약 15~20% 개선
③ Subsample	② + 포물선 보간	8°	15°	17°	fractional delay로 약 35% 추가 개선
④ Band-pass	③ + 2-6kHz 필터	6°	11°	12.5°	DOA 정보가 강한 대역만 사용, aliasing 감소
⑤ De-aliasing	④ + bin-wise 제거	5°	10°	11°	튀는 각도 제거, 안정성 향상
⑥ 보정 모델	⑤ + 선형 보정(a·θ+b)	3.5°	7°	7.8°	마이크 위치 오차 보정, 최종 성능

## 4. 정확도 개선 과정

### 실성능 수치

지표	값	지표	값	지표	값	지표	값
azimuth_error	16.47°	azimuth_error	2.20°	azimuth_error	10.8°	azimuth_error	1.9°
elevation_error	61.95°	elevation_error	2.92°	elevation_error	48.2°	elevation_error	2.5°
2d_angle_error	64.10°	2d_angle_error	3.65°	2d_angle_error	49.3°	2d_angle_error	3.1°
percentage_error	149.00%	percentage_error	8.50%	percentage_error	114.50%	percentage_error	7.20%
지표	값	지표	값	지표	값	지표	값
azimuth_error	1.67°	azimuth_error	4.18°	azimuth_error	1.3°	azimuth_error	3.2°
elevation_error	44.31°	elevation_error	63.06°	elevation_error	32.8°	elevation_error	47.9°
2d_angle_error	44.34°	2d_angle_error	63.19°	2d_angle_error	33.0°	2d_angle_error	48.1°
percentage_error	103.10%	percentage_error	146.90%	percentage_error	78.50%	percentage_error	112.30%

## 5. 데모 영상





## 5. 데모 영상 - UI



현재 추정된 방위각(Az)과 고도각(EI)을 숫자로 표시

에너지 기반으로 추정한 거리를 미터 단위로 표시한다. 하단의 프로그램스 바는 0~10m 범위를 나타내며, 거리에 따라 색상이 변한다

Reverb는 반향 지수로 Direct/Mixed/Reverb로 구분한다  
Motion은 음원 이동 속도를 분류한다  
Certainty는 위치 확실성을 퍼센트와 프로그램스 바로 표시한다

현재 추정현재 프레임의 RMS 에너지를 표시  
에너지가 임계값(800) 이상이면 헤더의 LED가 녹색으로 점등

최근 30프레임의 Az와 EI 변화를 시계열 그래프로 보여준다

## 6. 향후 개선 방향

### 달성결과

- 2D 음원 방향 추정 시스템 구현
- GCC-PHAT로 방향 환경 강건성 확보
- 에너지 기반 거리 추정 및 실시간 시각화

### 한계점

- 절대 거리 정확도 부족
- 2S 동기화 및 각도 분해능 제한
- 단일 음원만 추적 가능

### 향후 개선

- 하드웨어 동기화 및 마이크 수 증가
- 고급 DOA 알고리즘 적용
- 다중 음원 분리 및 실시간 대시보드 UI