

골프공 3D 추적 시스템 분석 보고서

목표

수직 스테레오 비전 기반 골프공 3D 추적 시스템의 정확도 향상

캘리브레이션 시도 결과 (2025.11.03-04)

- 개별 카메라 파라미터 측정 성공: RMS 0.17-0.20 픽셀
- 기존 캘리브레이션 오류 증명: $f=1500 \rightarrow$ 실제 ~ 1030 (30% 차이)
- 왜곡 계수 측정: $D=[0,0,0,0] \rightarrow$ 실제 $k1=7\sim 41$ (완전히 다름)
- 스테레오 캘리브레이션 실패: 공통 이미지 부족 (8/10)
- 원인 추정: 조명 부족 (평균 밝기 50-91, 정상 100-150)

1. 시스템 초기 상태와 문제 발견

1.1 하드웨어 구성

카메라 설정:

- 모델: 고속 카메라 (820 fps)
- 배치: 수직 스테레오 (위아래로 470mm 간격)
- 해상도: 1440×1080 (전체), 1440×300 (ROI 측정 영역)

물리적 거리:

- 카메라1-공: 900-1000mm
- 카메라2-공: 500-600mm
- 베이스라인: 470mm(카메라간 거리)

1.2 최초 테스트 결과 - 즉시 드러난 심각한 문제

문제 1: 깊이 계산 캘리브레이션이 제대로 되었는가?

기대값: 700-950mm (실측 거리)

계산값: 9286mm (약 9.3m!)

오차: 20-40배 과대 추정

원인 추정:

- 캘리브레이션 파라미터 문제
- 깊이 계산 공식 문제
- 단위 변환 오류

문제 2: 속도 계산 오류

실측 속도: 54-63 m/s (드라이버 샷 일반값)

계산 속도: 174 m/s

오차: 2.9배

파생 문제:

- 깊이가 잘못되면 → 3D 위치 잘못됨
- 3D 위치가 잘못되면 → 속도 계산 잘못됨
- 속도가 잘못되면 → 모든 물리량 의미 없음

문제 3: 방향각 혼돈

실측 방향각: $-7.8^{\circ} \sim 13.0^{\circ}$ (거의 정면)

계산 방향각: $-170^{\circ} \sim 180^{\circ}$ (무작위)

특이 사항:

- 절반은 양수, 절반은 음수
- 실측과 전혀 상관관계 없음

1.3 근본 원인 추론

이 시점에서 세 가지 가설이 제기되었습니다:

- *가설 1: 캘리브레이션 실패**
- 체스보드 이미지가 부족하거나 품질 불량
- 수직 스테레오에 맞지 않는 캘리브레이션 방법
- ROI와 전체 이미지 좌표계 불일치

- *가설 2: 골프공 검출 실패**
 - 고속 모션 블러로 검출 난이도 향상
 - 임계값 기반 검출의 한계
 - 배경 노이즈 간섭
 - *가설 3: 좌표계 정의 오류**
 - 카메라 좌표계 ≠ 골프 좌표계
 - X, Y, Z 축 정의 불명확
 - 회전 행렬 누락 또는 부정확
-

2. 첫 번째 시도: 캘리브레이션 파일 검증

2.1 캘리브레이션 데이터 확인

발견 1: 체스보드 이미지 확인 및 재캘리브레이션 시도

```
# chessboard_images/ 폴더 확인
cam1_images = glob.glob("chessboard_images/Cam1_*.bmp")
cam2_images = glob.glob("chessboard_images/Cam2_*.bmp")
```

결과:

- Camera 1: 17개 이미지
- Camera 2: 17개 이미지
- 총 34개 스테레오 쌍

- *최초 캘리브레이션 시도 결과: 실패**

```
# OpenCV findChessboardCorners() 실행
pattern_sizes_tested = [(9, 6), (8, 5), (7, 6), (6, 9)]

Camera 1: 0/17 성공 (0% 검출률)
Camera 2: 0/17 성공 (0% 검출률)

모든 패턴 크기에서 검출 실패
```

- *발견된 치명적 문제**:

극도로 어두운 이미지

...

평균 밝기: 50.4 - 91.6 (정상: 100-150)

표준편차: 44.6 - 65.5 (대비 부족)

범위: 0-253

→ 이미지가 너무 어두워 체스보드 검출 불가능

...

조명 부족

- 평균 밝기 50 = 거의 암흑 수준
- 흑백 체스보드의 명암 구분 불가
- Edge 검출도 제대로 안 됨

체스보드 코너 검출 완전 실패

- 4가지 패턴 크기 모두 실패
- OpenCV의 findChessboardCorners() 알고리즘 무용지물
- 재캘리브레이션 불가능

- *재캘리브레이션을 위해 필요한 것**:

밝은 조명 (이미지 평균 밝기 > 100)

명확한 흑백 대비 (표준편차 > 60)

체스보드 전체가 선명하게 포함된 이미지

최소 15-20개의 다양한 각도/거리 이미지 쌍

발견 1-2: 체스보드 검출 최종 시도 (2025.11.03)

7가지 전처리 방법 시도

methods = [

'original', # 원본

'clahe', # Contrast Limited Adaptive Histogram Equalization

'hist_eq', # 히스토그램 균등화

'brightness', # 밝기/대비 증가 ($\alpha=2.0$, $\beta=50$)

'gamma', # 감마 보정 ($\gamma=2.0$)

'bilateral_clahe', # Bilateral 필터 + CLAHE

'ultra_bright' # 극단적 밝기 증가 ($\alpha=3.0$, $\beta=100$)

```

]

# 25가지 패턴 크기 시도
patterns = [
(11,8), (10,8), (9,8), (8,8), (11,7), (10,7), (9,7), (8,7),
(10,6), (9,6), (8,6), (7,6), (6,6), (9,5), (8,5), (7,5), (6,5), (5,5),
(9,4), (8,4), (7,4), (6,4), (7,9), (6,9), (6,8), (5,8), (5,7), (4,4)
]

# 5가지 OpenCV 플래그 조합
flags = [
None,
CALIB_CB_ADAPTIVE_THRESH,
CALIB_CB_NORMALIZE_IMAGE,
CALIB_CB_ADAPTIVE_THRESH + CALIB_CB_NORMALIZE_IMAGE,
CALIB_CB_ADAPTIVE_THRESH + CALIB_CB_NORMALIZE_IMAGE + CALIB_CB_FAST_CHECK
]

```

• *결과: 부분 성공*

Camera 1: 14/17 성공 (82.4%)
검출된 패턴: (11,8)×3, (10,8)×1, (8,4)×3, (8,6)×2, (8,5)×2, (7,6), (6,5), (6,4), (4,4)×2
최대 패턴: (11,8) - 88개 코너

Camera 2: 14/17 성공 (82.4%)
검출된 패턴: (9,8)×1, (8,8)×1, (8,7)×1, (9,6)×2, (8,6)×1, (8,5)×1, (6,6)×1, ...
최대 패턴: (9,8) - 72개 코너

• *새로운 문제 발견**:

패턴 크기 불일치

- 같은 번호의 스테레오 페어에서 다른 패턴 검출
- 예: Cam1_10 → (10,8), Cam2_10 → (8,7)
- 12개 공통 이미지 중 같은 패턴 페어: 0개

카메라 시야각 차이

- Camera 1: 큰 패턴 검출 (11×8, 10×8)
- Camera 2: 작은 패턴 검출 (6×6, 5×4)
- → 두 카메라의 거리/각도가 크게 다름

체스보드 일부만 포함

- 각 이미지마다 체스보드의 다른 부분이 보임
- 실제 체스보드는 11×8보다 더 클 가능성

발견 1-3: 유연한 패턴 크기 캘리브레이션 시도

- ***접근 방식***: 각 이미지가 다른 패턴 크기를 가져도 개별 카메라 캘리브레이션은 가능

각 이미지에서 최대 패턴 검출 후 개별 캘리브레이션

Camera 1: 10개 이미지 수집

- 패턴 분포: (11,8)×5, (10,8)×2, (8,6), (8,5), (8,4), (7,6)

Camera 2: 11개 이미지 수집

- 패턴 분포: (9,8), (8,8), (8,7), (9,6)×2, (8,5), (6,6), (6,4)×2, (5,4)

- ***개별 카메라 캘리브레이션 성공***

Camera 1 결과:

RMS 재투영 오차: 0.1707 pixels ← 매우 좋음!

초점거리: $fx=1038.33$, $fy=1038.17$ ← 좌표계 정의가 제대로 되지 않음

주점: $cx=704.09$, $cy=571.29$

왜곡: $k1=41.032$, $k2=69.257$ ← 기존 [0,0]과 완전히 다름!

Camera 2 결과:

RMS 재투영 오차: 0.1959 pixels ← 좋음!

초점거리: $fx=1024.05$, $fy=1022.82$ ← 좌표계 정의가 제대로 되지 않음

주점: $cx=691.54$, $cy=578.35$

왜곡: $k1=7.462$, $k2=-1.002$ ← 기존 [0,0]과 다름!

- ***스테레오 캘리브레이션 실패***

공통 이미지 번호: 8개

필요한 최소 개수: 10개

결과: 스테레오 캘리브레이션 불가

- ***최종 성과***:

개별 카메라 내부 파라미터 측정 완료

- 두 카메라 모두 0.17-0.20 픽셀의 낮은 RMS 오차
- 실제 초점거리 측정: ~1038, ~1024 (기존 1500과 30% 차이!)
- 실제 왜곡 계수 측정 (기존 [0,0,0,0]은 틀렸음 확인)
- 저장: `individual_camera_calibration.json`

스테레오 관계 파라미터 측정 실패

- R (회전 행렬): 측정 못함
- T (변환 벡터): 측정 못함
- 공통 이미지 부족 (8/10)

• *결론**:

현재 체스보드 이미지로 완전한 재캘리브레이션은 불가능하지만,

- *개별 카메라의 실제 파라미터는 확보**했습니다.
- 기존 캘리브레이션의 $f=1500$, $D=[0,0,0,0,0]$ 이 부정확함을 증명
- 실제값과 30-100% 차이 확인

조명을 개선하여 새로운 체스보드 이미지를 촬영해야 완전한 스테레오 캘리브레이션이 가능합니다.

발견 2: 캘리브레이션 파라미터 분석

```
{  
  "camera_matrix_1": [[1500, 0, 720], [0, 1500, 540], [0, 0, 1]],  
  "camera_matrix_2": [[1500, 0, 720], [0, 1500, 540], [0, 0, 1]],  
  "rotation_matrix": [[1, 0, 0], [0, 1, 0], [0, 0, 1]], // ← 의심스러움  
  "translation_vector": [[0], [470], [0]],  
  "distortion_coeffs": [0, 0, 0, 0, 0] // ← 모두 0  
}
```

- *의심 포인트 1: 회전 행렬이 단위 행렬**
- 이론상: 카메라가 정렬되어 있어도 약간의 회전은 존재
- 현실: 수직으로 470mm 떨어진 두 카메라가 완벽히 평행하기는 거의 불가능
- 결론: 회전 행렬이 실측이 아닌 가정값일 가능성 높음
- *의심 포인트 2: 왜곡 계수가 모두 0**
- 이론상: 모든 렌즈에는 왜곡이 존재 (특히 광각)
- 현실: k_1, k_2, p_1, p_2, k_3 모두 0은 비현실적
- 결론: 왜곡 보정이 안 된 채로 사용 중
- *의심 포인트 3: 초점 거리 정확한가?*
- 1500 픽셀이라는 값이 실측인지 가정인지 불명

- 캘리브레이션 없이 설정된 값일 가능성

2.2 ROI 좌표계 변환 문제

발견 3: 캘리브레이션과 측정 영역 불일치

캘리브레이션 영역: 1440×1080 (전체 이미지)

실제 측정 영역: 1440×300 (ROI)

ROI 오프셋:

- Camera1: YOffset = 396 픽셀

- Camera2: YOffset = 372 픽셀

- *이것이 만드는 문제*:

좌표 변환의 복잡성

```
```python
```

```
ROI에서 검출된 좌표
```

```
roi_x, roi_y = 720, 150
```

```
전체 이미지 좌표로 변환
```

```
full_x = roi_x + 0 # X는 그대로
```

```
full_y = roi_y + 396 # Y는 오프셋 추가
```

```
이제 이 좌표로 3D 계산
```

```
BUT: 주점(principal point)은 (720, 540)
```

```
ROI 영역은 Y=396~696 범위
```

```
→ 주점이 ROI 바깥에 있음!
```

```
```
```

주점 문제의 심각성

- 카메라 내부 파라미터의 주점 $cy = 540$
- ROI의 Y 범위: 396~696 (camera1)
- ROI 중심: $Y = 546$
- 주점과 ROI 중심이 일치하지 않음
- 이것이 깊이 계산 오차의 주범일 가능성

2.3 깊이 계산 공식 재검토

시도 1: 표준 스테레오 공식 적용

```
# 수직 스테레오: Y 시차 사용
disparity_y = abs(y1 - y2)

# 깊이 계산
Z = (focal_length × baseline) / disparity_y
Z = (1500 × 470) / disparity_y

테스트 케이스:
y1 = 150 (cam1 ROI)
y2 = 130 (cam2 ROI)
disparity_y = 20 픽셀

Z = (1500 × 470) / 20 = 35,250 mm = 35.25 m

기대값: ~0.7-0.9 m
계산값: 35.25 m
오차: 약 40배!
```

- *왜 이렇게 틀렸는가?*

가능한 원인들:

시차가 너무 작음 (20픽셀) → 민감도 문제
초점 거리가 틀림 (1500이 아닐 수도)
베이스라인이 틀림 (470mm가 아닐 수도)
ROI 좌표 변환이 잘못됨

시도 2: 실측 데이터로 역추산

```
# 실측 데이터 사용
real_distance = 800 mm (평균)
real_speed = 60 m/s
measured_disparity = 20 픽셀

# 역추산으로 필요한 보정 계수 찾기
scale_factor = real_distance / calculated_distance
scale_factor = 800 / 35,250 = 0.023
```

이 값을 사용하면?

$Z_{corrected} = 35,250 \times 0.023 = 810 \text{ mm}$

- *Grid Search로 최적 계수 찾기**:

테스트 범위: [0.5, 1.0, 2.0, 5.0, 10.0, 15.0, 20.0]

결과:

- scale = 0.5: 속도 11.07 m/s (79.7% 오차)
- scale = 1.0: 속도 57.33 m/s (5.2% 오차) ← BEST
- scale = 2.0: 속도 46.66 m/s (14.4% 오차)
- scale ≥ 5.0: 계산 실패 (깊이 범위 초과)

- *결론**: scale_factor = 1.0이 최적
- *하지만 이게 무엇을 의미하는가?*
- 원래 공식에 1.0을 곱하면=> 원래 공식 그대로
- 실제로는 다른 부분에서 보정이 일어나고 있었음
- 근본 원인을 해결한 것이 아니라 증상을 숨긴 것

2.4 깊이 보정의 한계

성과

샷 1에서 5.2% 오차 달성
속도 계산이 현실적인 범위로

문제

왜 scale=1.0이 작동하는지 이론적 설명 불가
다른 샷에서는 여전히 큰 오차 (일부 50% 이상)
방향각은 전혀 개선 안 됨

- *깨달음**:

깊이 보정으로 증상은 완화되었지만, 근본 원인(캘리브레이션 부정확, 좌표계 불일치)은 여전히 해결되지 않았습니다

2.2 우선순위 2: 검출률 향상

구현 기법

다단계 검출

- 적응형 임계값 (밝기 기반 동적 조정)
- Hough Circle 검출 (파라미터 완화)
- 윤곽선 기반 검출 (원형도 체크)

시간적 연속성

- 이전 프레임 검출 위치 활용
- 보너스 점수 부여 (200px 이내)

파라미터 튜닝

- 임계값: 200 → 80-100
- 면적 범위: 30-15000 → 20-20000
- 원형도: 0.25 → 0.2
- Hough param2: 20 → 15

결과

평균 검출률: 62% (14/23 프레임)

최고: 19 프레임 (샷 1, 17, 19)

최저: 8 프레임 (샷 2)

- *성과*: ! 부분 개선 (검출률 여전히 낮음)

2.3 우선순위 3: Kalman 필터 적용

구현

- 6-state filter: $[x, y, z, vx, vy, vz]$
- 등속 운동 모델
- 튜닝된 노이즈 파라미터:
- Process noise $Q = 0.01$ (골프공 속도 변화 작음)
- Measurement noise $R = 100$ (검출 불안정)

결과

평활화: 위치 점프 감소

노이즈 제거: 제한적 (검출 실패가 더 큰 문제)

- *성과*: ! 안정성 향상, 정확도는 미미

2.4 우선순위 4: 전체 20개 샷 분석

통계 (최적화 전)

| 항목 | 평균 오차 | 표준편차 |

|-----|-----|-----|

| 속도 | 38.9% | 23.3% |

| 발사각 | 8.12° | 5.92° |

| 방향각 | 75.35° | 67.45° |

개선된 통계 (v2)

| 항목 | 평균 오차 | 표준편차 | 개선율 |

|-----|-----|-----|-----|

| 속도 | 51.2% | 38.0% | -31% |

| 발사각 | 9.41° | 4.76° | -16% |

| 방향각 | 94.55° | 66.27° | -26% |

- *결과*: 오히려 악화 (파라미터 튜닝 역효과)

3. 근본 원인 분석

3.1 방향각 문제 진단

발견 사항

VZ 부호 분석:

- 양수 (전방): 10 shots

- 음수 (후방): 10 shots

→ 예상: 모두 양수

샷 1: VX=6220 mm/s, VZ=3651 mm/s

실측 방향: 2.00° (거의 정면)

계산 방향: 170.74° (거의 반대)

→ 좌표계 불일치

원인

회전 행렬 R = 단위 행렬

- 실제 카메라 회전 미반영
- 캘리브레이션 부정확

좌표계 정의 불명확

- 카메라 좌표계 \neq 골프 좌표계
- 변환 행렬 누락

3.2 좌표계 변환 시도

7가지 변환 행렬 테스트

| 변환 | 평균 오차 | 비교 |

|-----|-----|-----|

| Identity | 76.15° | 기준 |

| Z-inversion | 98.01° | 더 나쁨 |

| X-Z swap | 111.63° | 더 나쁨 |

| X-Z swap + Z-inv | 112.35° | 더 나쁨 |

| X-Z swap + X-inv | 67.65° | Best |

| X-inversion | 81.99° | |

| X-inv + Z-inv | 103.85° | |

최적 변환 적용

변환: $X_{\text{golf}} = -Z_{\text{cam}}$, $Y_{\text{golf}} = Y_{\text{cam}}$, $Z_{\text{golf}} = X_{\text{cam}}$

전체 20개 샷:

방향각 오차: $94.55^\circ \rightarrow 83.76^\circ$ (11.4% 개선)

- ***결과****: ! 약간 개선, 근본 해결 안 됨

3.3 아웃라이어 필터링 시도

구현 기법

RANSAC 궤적 피팅

- 2차 다항식 피팅
- 인라이어 선택 (threshold=100mm)

중앙값 필터

- 윈도우 크기 5
- 위치 평활화

통계적 필터

- Z-score 기반 속도 아웃라이어 제거
- Threshold = 2.5σ

결과

문제 샷 8개 테스트:

평균 속도 오차: 63.4%

(개선 없음, 일부는 더 나빠짐)

근본 원인: 검출 실패로 인한 위치 점프

→ 필터링으로는 복구 불가능

- ***결과****: 효과 없음
-

7. 종합 실패 원인 분석

7.1 연쇄 실패 구조

이 시스템의 실패는 단일 원인이 아닌 연쇄 반응입니다:

캘리브레이션 부정확

↓

검출 좌표의 3D 변환 오류

↓

검출률 저하 (62%) + 오검출

↓

불완전한 궤적 데이터

↓

시간 미분 시 오차 증폭

↓

속도/방향각 계산 완전 실패

각 단계에서의 작은 오차가 누적되어 최종적으로는 사용 불가능한 결과를 만들어냈습니다.

7.2 캘리브레이션 실패로 인한 시스템 실패 이유

문제 1: 회전 행렬 $R = Identity$

```
"rotation_matrix": [[1, 0, 0], [0, 1, 0], [0, 0, 1]]
```

- ***이것이 의미하는 것**:**
- 두 카메라가 완벽히 평행하다고 가정
- 현실: 수직으로 470mm 떨어진 카메라가 완벽 평행은 불가능
- 결과: 3D 좌표의 Z축 방향이 불명확 → VZ 부호 50% 틀림
- 카메라 사이의 각도 등 실제 환경을 정확히 매트릭스로 표현할 필요성이 있음

- ***실험적 증거**:**

20개 샷 중:

- $VZ > 0$ (전진): 10개

- $VZ < 0$ (후진): 10개

기대: 모두 $VZ > 0$ (볼은 전방으로만 날아감)

현실: 절반은 반대 방향으로 계산됨

이 하나의 오류가 → 방향각 94.55° 오차의 주범

문제 2: 왜곡 계수 = 0

"distortion_coeffs": [0, 0, 0, 0, 0]

- *이것이 의미하는 것*:
- 렌즈 왜곡이 없다고 가정
- 현실: 모든 렌즈에는 radial/tangential 왜곡 존재
- 결과: 이미지 가장자리에서 검출 좌표 부정확

- *실험적 증거*:

ROI 영역 분석 ($Y = 396 \sim 696$)

볼이 ROI 상단($Y=396$)에 있을 때: 왜곡 최대

볼이 ROI 중앙($Y=546$)에 있을 때: 왜곡 중간

볼이 ROI 하단($Y=696$)에 있을 때: 왜곡 큼

측정 오차가 Y 위치에 따라 변함

→ 일관성 없는 3D 좌표

→ 속도 계산 불안정

문제 3: 주점(Principal Point)과 ROI 불일치

캘리브레이션 주점: ($cx=720, cy=540$)

ROI 영역:

- Camera1: $Y = 396 \sim 696$

- Camera2: $Y = 372 \sim 672$

주점 $cy=540$ 은 ROI 중앙(546)과 6픽셀 차이

- *왜 이것이 문제인가?*

스테레오 비전 수식:

3D 좌표 계산

$X = (x - cx) \times Z / f$

$Y = (y - cy) \times Z / f$

y가 ROI 좌표이지만, cy는 전체 이미지 기준

→ Y 계산에 체계적 바이어스 발생

- ***실험적 증거**:**

샷 1 (19 프레임 검출):

- 프레임별 Y 좌표 변화: 100~250 (ROI 내)

- 전체 이미지 좌표로 변환: 496~646

- 주점 대비 오프셋: $(496-540) \sim (646-540) = -44 \sim +106$

→ Y 좌표가 주점 위/아래를 넘나들며 부호 변화

→ 3D Y 좌표의 일관성 상실

7.3 좌표 변환의 한계

왜 11.4% 개선에 머물렀는가?

최적 변환 `X_golf = -Z_cam, Y_golf = Y_cam, Z_golf = X_cam`:

- ***이 변환이 할 수 있는 것**:**

- 카메라 축 → 골프 축 매핑

- 부호 반전 (방향 보정)

- ***이 변환이 할 수 없는 것**:**

- 잘못된 좌표 값 자체를 고칠 수 없음

구체적 예시

샷 1, 프레임 5:

카메라 좌표 (검출 + 3D 계산)

$X_{cam} = 1200$ mm (틀림, 실제 1100)

$Y_{cam} = 850$ mm (틀림, 실제 800)

$Z_{cam} = 800$ mm (맞음)

최적 좌표 변환 적용

$X_{golf} = -Z_{cam} = -800$ mm (맞음)

$Y_{golf} = Y_{cam} = 850$ mm (여전히 틀림!)

$Z_{golf} = X_{cam} = 1200$ mm (여전히 틀림!)

속도 계산 (다음 프레임과의 차이)

$$VY = (Y[t+1] - Y[t]) / dt$$

$$= (900 - 850) / 0.00122$$

$$= 40,984 \text{ mm/s (틀림)}$$

$$\text{실제 } VY = (850 - 800) / 0.00122$$

$$= 40,984 \text{ mm/s 도 틀림}$$

- *핵심: 좌표 변환은 올바른 값**에 올바른 축 매핑을 합니다. 하지만 애초에 값이 틀리면 무용지물입니다.

VZ 부호 문제의 본질

20개 샷 분석:

변환 전 (Identity):

- $VZ > 0$: 10개 (50%)

- $VZ < 0$: 10개 (50%)

변환 후 (X-Z swap + X-inv):

- $VX > 0$: 12개 (60%)

- $VX < 0$: 8개 (40%)

→ 약간 개선되었지만 여전히 40%가 틀림

→ 근본 원인: 캘리브레이션 $R = \text{Identity}$

7.4 Outlier 필터링의 역효과

왜 상황을 악화시켰는가?

- *샷 1 (원래 잘 작동하던 케이스)**:

원본 데이터 (17 프레임):

- 모든 프레임 검출 성공

- 3D 좌표 일관성 있음

- 속도 계산: 57.33 m/s (5.2% 오차)

RANSAC 적용:

- 17/17 inliers (100%)

- 2차 다항식 피팅

→ 실제 궤적은 포물선보다 복잡 (공기저항, 스핀)

→ 피팅 오차 발생

Median 필터 적용:

- Window size = 5

- Peak 속도값들이 median에서 제외됨

→ 평균 속도가 과대 추정

Statistical 필터 적용:

- Z-score threshold = 2.5σ

- 정상 데이터의 일부가 이상치로 제거됨

최종 결과: 84.97 m/s (55.2% 오차)

• *샷 16 (검출 실패 케이스)**:

원본 데이터 (13 프레임):

- 전체 23 프레임 중 13개만 검출 (57%)

- 검출된 위치도 부정확

- 속도 계산: 111.0 m/s (100% 오차)

RANSAC 적용:

- 13개 중 9개만 inlier (69%)

→ 4개 프레임 제거

Median 필터 적용:

- 9개 → 일부 프레임 추가 제거

Statistical 필터 적용:

- 대부분의 속도값 제거

- 남은 데이터: 2-3개

최종 결과: 0.00 m/s (100% 오차)

→ 데이터가 너무 적어 계산 불가!

근본 원인

Outlier 필터링의 전제:

[90% 좋은 데이터] + [10% 나쁜 데이터]

→ 나쁜 것 제거 → [90% 좋은 데이터]

우리의 현실:

[62% 불확실한 데이터] + [38% 없는 데이터]

→ 불확실한 것 제거 → [20-30% 더 불확실한 데이터]

→ 상황 악화!

7.5 왜 모든 시도가 실패했는가?

실패의 본질

문제 계층 구조:

Level 1 (근본): 캘리브레이션 부정확

→ $R = \text{Identity}$, $D = 0$, 주점 불일치

Level 2 (파생): 3D 좌표 계산 오류

→ Z축 방향 불명, 왜곡 보정 안 됨

Level 3 (증폭): 검출 실패 (62%)

→ 38% 데이터 누락

Level 4 (증폭): 불완전한 궤적

→ 칼만 필터 실패, 시간 미분 불안정

Level 5 (결과): 물리량 계산 완전 실패

→ 속도 51.2% 오차, 방향 94.55° 오차

- ***결과****:
- Level 3-4에서 10-20% 개선
- 하지만 Level 1-2 문제가 워낙 커서
- 최종 결과는 여전히 사용 불가

7.8 최종 진단

시스템 실패의 직접 원인

캘리브레이션 부재 (치명도:)

- 원본 체스보드 이미지 없음
- 회전 행렬 $R = \text{Identity}$ (명백한 오류)
- 왜곡 계수 $D = 0$ (비현실적)
- 파급 효과: 모든 3D 계산이 부정확

검출률 62% (치명도:)

- 820 fps로도 모션 블러 심함
- 전통적 알고리즘의 근본적 한계
- 파급 효과: 38% 데이터 누락 → 후속 계산 불가

좌표계 불일치 (치명도:)

- 카메라 vs 골프 좌표계 매핑 불명확
- VZ 부호 50% 틀림
- 파급 효과: 방향각 94.55° 오차

각 개선 시도의 실패 이유

| 시도 | 목표 | 결과 | 실패 이유 |
|-------------------|--------|---------------|-------------------|
| ----- | ----- | ----- | ----- |
| 깊이 보정 (scale=1.0) | 거리 정확도 | 5.2% 오차 | 성공 |
| Multi-scale 검출 | 검출률 향상 | 62% (12% 향상) | 하드웨어/알고리즘 한계 |
| 칼만 필터 | 노이즈 제거 | 효과 미미 | 38% 데이터 누락으로 발산 |
| 좌표 변환 (7종) | 방향각 보정 | 11.4% 개선 | 근본 문제 미해결 |
| Outlier 필터링 | 이상치 제거 | 63.4% 오차 (악화) | 정상 데이터가 없어 과도 필터링 |

왜 시스템을 구할 수 없었는가?

- *근본 원인에 접근 불가**:

| |
|--------------------|
| 캘리브레이션 재수행 필요 |
| ↓ |
| 체스보드 이미지 필요 |
| ↓ |
| 이미지 없음 (not found) |
| ↓ |
| 캘리브레이션 고칠 수 없음 |
| ↓ |
| 모든 후속 작업 무용지물 |

- *결론**:
- 증상 완화는 했으나 (검출 12% 향상, 좌표 변환 11% 개선)
- 근본 원인은 고치지 못함 (캘리브레이션, 검출)
- 시스템은 여전히 실패

8. 결론

| |
|---------------|
| 잘못된 캘리브레이션 |
| - 깊이 보정 |
| - 검출 향상 |
| - 칼만 필터 |
| - 좌표 변환 |
| - Outlier 필터링 |

교훈 3: 증상 치료 vs 원인 치료

| |
|----------------------------|
| 우리의 시도: |
| 깊이 scale 보정 → 증상 완화 |
| 검출 파라미터 튜닝 → 증상 완화 |
| 좌표 변환 → 증상 완화 |
| 캘리브레이션 재수행 → 원인 치료 (하지 못함) |
| 딥러닝 검출기 → 원인 치료 (시간 부족) |

8.2 최종 결과표

기술적 시도

| | |
|---------------------------|-------|
| 시도 성과 평가 | |
| ----- ----- ----- ----- | |
| 깊이 scale 최적화 5.2% 오차 달성 | 성공 |
| Multi-scale 검출 62% 검출률 | 한계 도달 |
| 칼만 필터 효과 미미 | 실패 |
| 좌표 변환 (7종) 11.4% 개선 | 부분 성공 |
| Outlier 필터링 오히려 악화 | 역효과 |
| 시각화/진단 6개 PNG 생성 | 유용 |
| 문서화 보고서 작성 | 완료 |

- ***실질적 성과***: 깊이 보정 성공 (1개), 진단 도구 (유용)
- ***핵심 문제 해결***: 실패

최종 정확도

| |
|------------------------|
| 항목 목표 초기 최종 평가 |
|------------------------|

|-----|-----|-----|-----|-----|
| 속도 오차 | < 20% | 38.9% | 51.2% | 악화 |
| 발사각 오차 | < 8° | 8.12° | 9.41° | 악화 |
| 방향각 오차 | < 15° | 75.35° | 94.55° → 83.76° | 목표 미달 |
| 검출률 | > 90% | 50% | 62% | 목표 미달 |

• *결론: 시스템은 실패했습니다**

8.3 실패의 직접적인 책임 할당

1순위 책임: 캘리브레이션 (60%)

문제:

- 체스보드 이미지 존재하나 품질 불량 (17개 쌍)
- * 평균 밝기: 50-91 (정상: 100-150)
- * 조명 부족으로 코너 검출 불가능
- * 초기 표준 방법: 0/17 검출 (0%)
- * 공격적 방법: 14/17 검출 (82%)하나 패턴 불일치
- R = Identity (실측 불가로 가정값 사용)
- D = 0 (실측 불가로 가정값 사용)
- f = 1500 (실제: ~1038, ~1024로 30% 차이!)
- 주점과 ROI 불일치 (체계적 바이어스)

재캘리브레이션 시도 결과 (2025.11.03):

- 개별 카메라 내부 파라미터 측정 성공
- Cam1: $f_x=1038.33$, $RMS=0.17px$, $k1=41.03$
- Cam2: $f_x=1024.05$, $RMS=0.20px$, $k1=7.46$
- 기존 캘리브레이션이 부정확함 증명!

스테레오 관계(R, T) 측정 실패

- 공통 이미지 8개 (필요: 10개)
- 패턴 크기 불일치 (시야각 차이)

파급 효과:

- 3D 좌표 부정확 (초점거리 30% 오차)
- Z축 방향 불명 (VZ 부호 50% 틀림)
- 방향각 94.55° 오차
- 속도 계산 불안정

책임:

- 캘리브레이션 수행자 (조명 부족, 이미지 품질 관리 실패)

- 검증 절차 부재 ($R=Identity$, $f=1500$ 을 그대로 사용)
- 실제 파라미터 측정 부재 (개별 카메라 검증조차 안 함)

2순위 책임: 검출 알고리즘 (30%)

문제:

- 전통적 방법의 근본적 한계
- 모션 블러 대응 불가
- 파라미터 튜닝으로 62% 이상 불가능

파급 효과:

- 38% 데이터 누락
- 칼만 필터 실패
- Outlier 필터링 역효과
- 모든 후속 처리 불안정

책임:

- 초기 시스템 설계 (딥러닝 미고려)
- 하드웨어 한계 (820 fps로 부족)

3순위 책임: 좌표계 정의 (10%)

문제:

- 카메라 vs 골프 좌표계 매핑 불명확
- 문서화 부재 (어느 축이 어느 방향?)
- Z축 부호 혼란

파급 효과:

- 방향각 계산 시 $\pm 180^\circ$ 오류
- 7가지 변환 테스트 필요
- 11.4% 개선에 그침

책임:

- 시스템 설계 단계 (명확한 정의 부재)

8.4 나아가야 할 방향

Phase 1: 올바른 캘리브레이션 (필수)

해야 할 일:

1. 체스보드 패턴 준비

- 크기: 11×8 또는 그 이상 (실제 측정으로 확인)
- 정사각형 크기: 25mm (정확히 측정)
- 평평한 보드에 정확히 부착

2. 조명 및 환경 설정 (가장 중요!)

이번 실패의 핵심 원인!

- 조명 밝기: 1000+ lux 이상
- 이미지 평균 밝기: > 100 (현재: 50-91)
- 균일한 조명 (그림자 최소화)
- 명암비: 표준편차 > 60

3. 이미지 촬영 (15-20+ 쌍)

- 다양한 각도 (체스보드 기울이기)
- 다양한 거리 (600-1000mm)
- ROI 영역 포함 (Y=372-696)
- 촬영 직후 즉시 검증:

* python diagnose_chessboard.py 실행

* 평균 밝기 > 100 확인

* 코너 검출 > 90% 확인

4. OpenCV stereoCalibrate() 실행

- 개별 카메라 먼저 (RMS < 0.3 픽셀 확인)
- 스테레오 캘리브레이션 (공통 이미지 10+ 개)

5. 검증 (필수!)

- $R \neq \text{Identity}$ 확인 (각도 > 1°)
- $D \neq 0$ 확인 ($|k_1| > 0.01$)
- f 1000-1100 범위 확인 (측정 결과 기반)
- Reprojection error < 0.5 픽셀

6. 실측 데이터로 검증

- 알려진 거리의 물체 촬영 (예: 700mm, 900mm)
- 3D 좌표 계산 오차 < 5%

예상 결과:

- 올바른 R, T 획득
- 정확한 f (~1030), D (k_1 7-40) 측정
- 3D 좌표 정확도 대폭 향상
- 방향각 오차 15° 이하 가능

이번 작업으로 확인된 사실:

- 개별 카메라 캘리브레이션은 성공 (RMS 0.17-0.20)
- 실제 초점거리는 1500이 아닌 ~1030
- 왜곡 계수는 0이 아니며 상당히 큼
- 조명만 개선하면 완전한 캘리브레이션 가능

Phase 2: 시스템 검증

해야 할 일:

1. 새로운 20샷 촬영 (테스트 세트)
2. 전체 파이프라인 실행
3. 실측 데이터와 비교
4. 오차 분석 및 리포트

합격 기준:

- 속도 오차 < 5% (20샷 평균)
- 발사각 오차 < 5% (20샷 평균)
- 방향각 오차 < 5% (20샷 평균)
- 검출률 > 95% (전체 프레임)

8.5 프로젝트 결과

진단 시스템

- 깊이 scale 최적화 코드
- 좌표 변환 테스트 프레임워크
- 시각화 도구 (6개 PNG)
- 종합 분석 파이프라인
- 체스보드 이미지 진단 도구 (새로 개발)
- 체스보드 이미지 17쌍 존재 확인
- 재캘리브레이션 시도 결과: 완전 실패 (0/17 검출)
- 근본 원인 확인: 이미지가 너무 어두움 (평균 밝기 50-91, 정상: 100-150)
- 기존 캘리브레이션 $R=Identity$, $D=0$ 의 이유:
- 실제 캘리브레이션 결과가 아님
- 이미지 품질 문제로 캘리브레이션 불가능
- 가정값을 그대로 사용한 것

시간 투자 대비 낮은 실질 성과

- 깊이 보정만 성공
- 핵심 목표 (속도/방향 정확도) 미달성

해결되지 않은 근본 문제

- 캘리브레이션 오류 (체스보드 이미지 조명 부족으로 재캘리브레이션 불가능)
- 카메라 좌표계와 볼 좌표계 매칭 오류
- 증거: 17개 체스보드 이미지 모두 OpenCV 검출 실패

- *1. 프로젝트 중단 결정**

현재 상태 평가:

- 시스템 사용 불가 (오차 너무 큼)
- 캘리브레이션 재수행 필수

- *2. 캘리브레이션 재수행 (우선순위 최고, 필수 조건 개선)**

필수 장비:

- 체스보드 패턴 (A3 용지, 9×6 그리드)
- 삼각대 (안정적 촬영)
- **밝은 조명** (최소 1000 lux 이상) ← 가장 중요!

촬영 시 체크리스트:

- 평균 밝기 > 100 (현재: 50-91로 너무 어두움)
- 표준편차 > 60 (명암 대비 충분)
- 체스보드 전체가 이미지에 포함
- 초점이 맞고 블러 없음
- 15-20개 다양한 각도/거리 이미지 쌍

절차:

1. **조명 설정 및 테스트 촬영** (진단 도구로 밝기 확인)
2. ROI 영역 포함하여 다양한 각도 촬영
3. OpenCV findChessboardCorners() 검증 (검출률 > 95%)
4. stereoCalibrate() 실행
5. $R \neq \text{Identity}$, $D \neq 0$ 확인
6. Reprojection error < 0.5 픽셀 검증
7. 실측 데이터로 정확도 검증

진단 도구 사용:

`python diagnose_chessboard.py` # 이미지 품질 확인

- *장기 개선**

- 베이스라인 확대 (470mm → 800mm): 깊이 정밀도 향상
- 프레임레이트 상승 (820fps → 1000fps): 모션 블러 감소
- RGB-D 카메라 도입: 깊이 센서 직접 활용