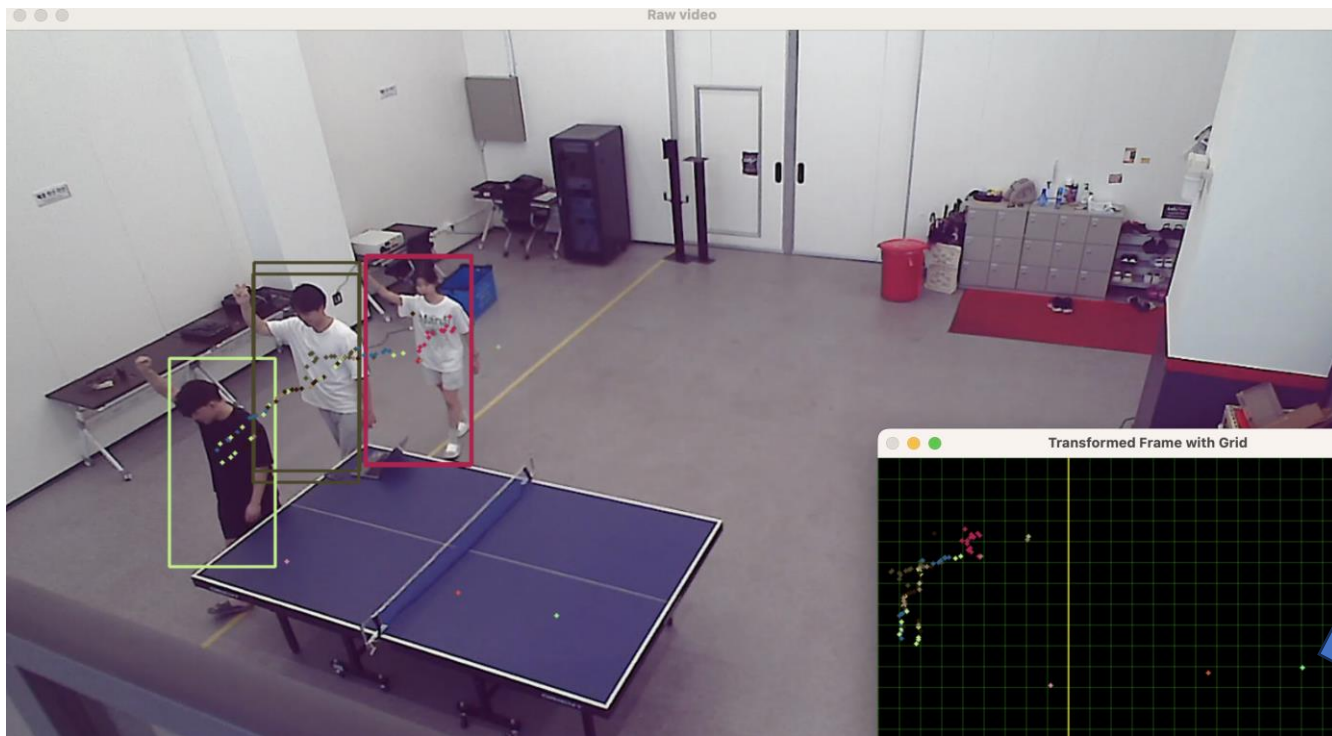


1팀 경과보고(7/25)

객체 인식 및 추적



- 다중객체일 경우 각 박스의 색과 경로 점의 색을 다르게 표시 및 그리드에 붉은 점으로 실시간 위치 표시
- 점은 10프레임뒤에 사라지게 두어 화면이 지저분해지는 경우 방지
- 잔여 점들이 남아있는 현상 수정중

코드 수정사항

```
four_cord = [[445, 154], [938, 315], [2, 555], [495, 716]]

#원본 좌표에서 투시변환된 좌표 출력함수
def original_to_conv(x, y, original_cord):
    w, h = 1280, 720 # 원본 영상의 해상도
    #원본 영상의 ROI 4개의 좌표(좌상단, 우상단, 좌하단, 우하단)
    srcQuad = np.array(original_cord, np.float32)
    #투시 변환 지정
    dstQuad = np.array([[0, 0], [w-1, 0], [0, h-1], [w-1, h-1]], np.float32)
    srcPoint = np.array([[x, y]], dtype=np.float32)
    # 투시변환 진행
    dstPoint = cv2.perspectiveTransform(srcPoint.reshape(-1, 1, 2), cv2.getPerspectiveTransform(srcQuad, dstQuad))
    #메인함수에서 투시변환 창을 0.4배로 resize하였기 때문에 해당 좌표에서 0.4배 적용하여 변환(resize 배율 변경시 반드시 바꿔줄 것!)
    return int(dstPoint[0, 0, 0])*0.4, int(dstPoint[0, 0, 1])*0.4

print(original_to_conv(445, 154, four_cord))

✓ 0.0s

(0.0, 0.0)
```

- 기존 좌표 변환 코드에 좌표값 하드코딩된 부분 수정.

- 이제 인자로 x, y, 원본영상의 네 모퉁이 좌표를 받음

```
four_cord = [[445, 154], [938, 315], [2, 555], [495, 716]]

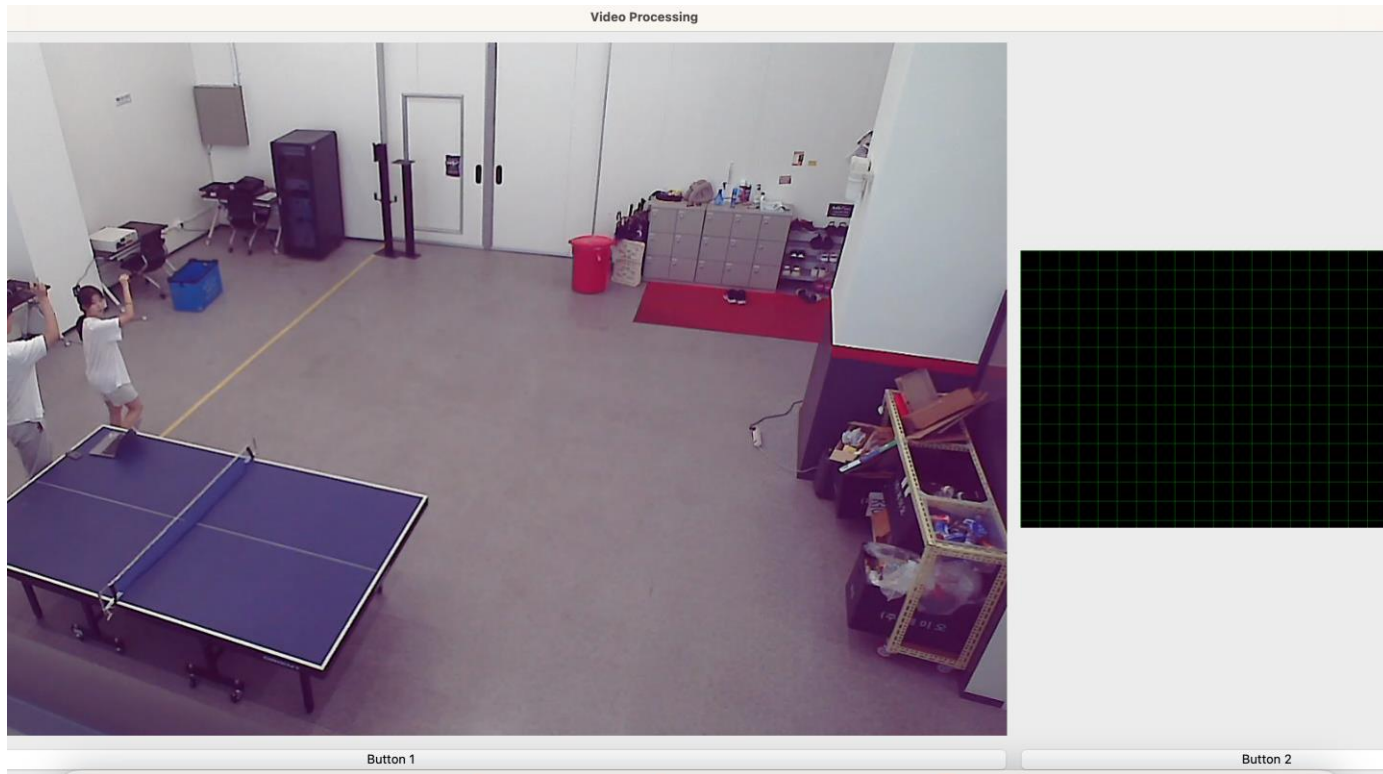
#투시변환된 좌표에서 원본좌표 출력함수
def conv_to_original(x, y, original_cord):
    w, h = 1280, 720
    #원본 영상의 ROI 4개의 좌표(좌상단, 우상단, 좌하단, 우하단)
    srcQuad = np.array(original_cord, np.float32)
    dstQuad = np.array([[0, 0], [w-1, 0], [0, h-1], [w-1, h-1]], np.float32)
    #메인함수에서 투시변환 창을 0.4배로 resize하였기 때문에 변환된 좌표에서 0.4를 나누어준다(resize 배율 변경시 반드시 바꿔줄 것!)
    dstPoint = np.array([[x/0.4, y/0.4]], dtype=np.float32)
    # 투시변환 역으로 진행
    srcPoint = cv2.perspectiveTransform(dstPoint.reshape(-1, 1, 2), cv2.getPerspectiveTransform(dstQuad, srcQuad))
    return int(srcPoint[0, 0, 0]), int(srcPoint[0, 0, 1])

print(conv_to_original(0,0,four_cord))

✓ 0.0s

(445, 154)
```

GUI



- 간단한 원본 영상 창, 그리드 영상 창, 실행 버튼 정도로 구성.
- 기존의 그리드에서 좌표 확인 및 이동 경로 표시 기능 유지
- 화면 밖으로 창이 튀어 나가는 현상 수정 중.