

mediapipe_pose.py

```

import cv2
import mediapipe as mp
import time
import math
mpDraw = mp.solutions.drawing_utils
mpPose = mp.solutions.pose
pose = mpPose.Pose()
cap = cv2.VideoCapture(0)
pTime = 0
a = b = c = d = 0
cx = cy = cz = 0
def findpose(img, results, lmlist):
    global cx, cy, cz

    if results.pose_landmarks:
        mpDraw.draw_landmarks(img, results.pose_landmarks, mpPose.POSE_CONNECTIONS)
    for id, lm in enumerate(results.pose_landmarks.landmark):
        h, w, c = img.shape
        #print(id, lm)
        cx, cy, cz = int(lm.x * w), int(lm.y * h), int(lm.z * c)
        cv2.circle(img, (cx, cy), 5, (0, 255, 0), cv2.FILLED)
        lmlist.append([id, cx, cy, cz])
        #print(lmlist)
    len(lmlist) != 0の場合:
        印刷(lmlist[11], lmlist[12])
        印刷(lmlist[20], lmlist[21])
        印刷(lmlist[15][1])
        cv2.putText(img, f'left_wrist'+ str(lmlist[11][3]) + '    right_wrist ' +
str(lmlist[12][3]), org=(20, 300), fontFace=cv2.FONT_HERSHEY_SIMPLEX, fontScale=0.5, color=
(255, 0, 0), thickness=2, lineType=cv2.LINE_4)
        cv2.putText(img, f'right_shoulder'+ str(lmlist[12][2]) + '    right_index ' +
str(lmlist[20][2]) + '    right_hip'+ str(lmlist[24][2]) + '    right_index ' +
str(lmlist[20][2]), org=(20, 150), fontFace=cv2.FONT_HERSHEY_SIMPLEX, fontScale=0.5, color=
(255, 0, 0), thickness=2, lineType=cv2.LINE_4)
        return lmlist
    else:
        return []

    # #cmd = 'rc {a} {b} {c} {d}' rc{roll}{pitch}{throttle}{yaw}
    # ,x軸、z軸throttle前後に対する処理
def forward_back():
    global b
    b = 0 #bを初期化
    if lmlist[11][2] > lmlist[19][2]:
        b = 20 #forward
        cv2.putText(img, f'forward ', org=(100, 100), fontFace=cv2.FONT_HERSHEY_SIMPLEX,
fontScale=0.9, color=(255, 0, 0), thickness=2, lineType=cv2.LINE_4)
    elif lmlist[23][2] < lmlist[19][2]:
        b = -20 #back
        cv2.putText(img, f'back ', org=(100, 100), fontFace=cv2.FONT_HERSHEY_SIMPLEX,
fontScale=0.9, color=(255, 0, 0), thickness=2, lineType=cv2.LINE_4)
    else:
        b = 0
    return
def up_down(): #上下運動
    global c

```

```

    c = 0
#     print(cy)
    if lmlist[12][2] > lmlist[20][2]:
        c = 20 #forward
        cv2.putText(img, f'up ', org=(100, 100), fontFace=cv2.FONT_HERSHEY_SIMPLEX,
fontScale=0.9, color=(255, 0, 0), thickness=2, lineType=cv2.LINE_4)

        elif lmlist[24][2] < lmlist[20][2]:
            c = -20 #back
            cv2.putText(img, f'down ', org=(100, 100), fontFace=cv2.FONT_HERSHEY_SIMPLEX,
fontScale=0.9, color=(255, 0, 0), thickness=2, lineType=cv2.LINE_4)
        else:
            c = 0

def yaw(): #上下運動
    global d

    d = 0
#     print(cy)
    if lmlist[11][3] != 0:
        d = 30 #up
        cv2.putText(img, f'yaw right ', org=(100, 100), fontFace=cv2.FONT_HERSHEY_SIMPLEX,
fontScale=0.9, color=(255, 0, 0), thickness=2, lineType=cv2.LINE_4)
        elif lmlist[12][3] != 0:
            d = -30 #back
            cv2.putText(img, f'yaw left ', org=(100, 100), fontFace=cv2.FONT_HERSHEY_SIMPLEX,
fontScale=0.9, color=(255, 0, 0), thickness=2, lineType=cv2.LINE_4)
        else:
            d = 0

#     if lmlist[16][2] > 100 and lmlist[16][2] < 188:
#         c = 0
#     elif 400 > lmlist[16][2] > 188 : #frame h:360の1/2=180 顔の下降 (座標が大きくなる)
#         c = -20
#     elif 50 < lmlist[16][2] < 100 :
#         c = 20
#     return
#     print('c' + str(c))

#         print('throttle: ' + str(c))

#def yaw(): #クロックワイズ旋回運動
#     global d

#     d = 0
#         print(cx)

#     if lmlist[16][1] > 100 and lmlist[16][1] < 200:
#         d = 0
#     elif 500 > lmlist[16][1] > 200 : #frame h:360の1/2=180 顔の下降 (座標が大きくなる)
#         d = -20

#     elif 30 < lmlist[16][1] < 100 :
#         d = 20
#     return

```

```

#         print('d'+ str(d))

#def land():    #landコマンド

#         if lmlist[15][2] < 100 :
#             cv2.putText(img, f'LAND', org=(100, 50), fontFace=cv2.FONT_HERSHEY_SIMPLEX,
fontScale=0.9, color=(255, 0, 255), thickness=2, lineType=cv2.LINE_4)
#             sock.sendto(b'land', tello_address)

#         return
#         print('left'+ str(lmlist[15][2]))

while True:
    for i in range(5):
        ret, img = cap.read()
        if img is None or img.size == 0:
            continue
#     success, img = cap.read()
imgRGB = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
results = pose.process(imgRGB)

# print(results.pose_landmarks)
lmlist = []
lmlist = findpose(img, results, lmlist)
forward_back()
up_down()
yaw()
#     yaw()
#     land()
#     lmlist = pitch(lmlist)
#     lmlist = throttle(lmlist)
#     lmlist = yaw(lmlist)
#     cTime = time.time()
#     fps = 1 / (cTime - pTime)
#     pTime = cTime
#     cv2.putText(img, str(int(fps)), (70, 50), fontFace=cv2.FONT_HERSHEY_SIMPLEX,
fontScale=0.9, color=(255, 0, 0), thickness=2, lineType=cv2.LINE_4)
#     rc_command = "rc {0} {1} {2} {3}".format(a, b, c, d)    # #cmd = 'rc {a} {b} {c} {d}'
rc{roll}{pitch}{throttle}{yaw}
#         sock.sendto(rc_command.encode('utf-8'),tello_address)
#
#         cv2.putText(img, f' roll: '+ str(a) + f' pitch:' + str(b) + f' throttle: ' + str(c) + f'
yaw: ' + str(d), org=(100, 400), fontFace=cv2.FONT_HERSHEY_SIMPLEX, fontScale=0.6, color=(255,
0, 255), thickness=2, lineType=cv2.LINE_4)
#         print("Sending RC command with values:", a, b, c, d)

cv2.imshow('frame', img)

if cv2.waitKey(1) & 0xFF == 27 :
#
#         sock.sendto(b'land', tello_address)
cap.release()
cv2.destroyAllWindows()
#         sock.close()
break

```