```
import numpy as np
import cv2
import math

detect_cascade=cv2.CascadeClassifier('haarcascade_a.xml')
detectc_cascade=cv2.CascadeClassifier('haarcascade_star.xml')
camR=cv2.VideoCapture(2)
camR.set(cv2.CAP_PROP_FRAME_WIDTH,720)
camR.set(cv2.CAP_PROP_FRAME_HEIGHT,405)
camL=cv2.VideoCapture(1)
camL.set(cv2.CAP_PROP_FRAME_WIDTH,720)
camL.set(cv2.CAP_PROP_FRAME_HEIGHT,405)

X=0
Y=0
Z=0

Xc=0
Yc=0
Zc=0

XX=0
YY=0
ZZ=0

XXc=0
YYc=0
ZZc=0


while(True):
    a1=2000 #right
    a2=2000 #left
    tfR,frameR=camR.read()
    gray = cv2.cvtColor(frameR, cv2.COLOR_BGR2GRAY)
    detect = detect_cascade.detectMultiScale(gray, 1.3, 5)
    for (xrr,yrr,wr,hr) in detect:
        cv2.rectangle(frameR,(xrr,yrr),(xrr+wr,yrr+hr),(255,0,0),2)
        cv2.putText(frameR, "A",(xrr,yrr-50), cv2.FONT_HERSHEY_SIMPLEX,
0.75,(255,0,0),2);
        cv2.putText(frameR, "X=" + `X`,(xrr,yrr-30), cv2.FONT_HERSHEY_SIMPLEX,
0.5,(255,0,0),2);
        cv2.putText(frameR, "Y=" + `Y`,(xrr,yrr-15), cv2.FONT_HERSHEY_SIMPLEX,
0.5,(255,0,0),2);
        cv2.putText(frameR, "Z=" + `Z`,(xrr,yrr), cv2.FONT_HERSHEY_SIMPLEX,
0.5,(255,0,0),2);

        cv2.putText(frameR, "XX=" + `XX`,(xrr,(yrr+hr)+15),
```

```
cv2.FONT_HERSHEY_SIMPLEX, 0.5,(255,0,0),2);
        cv2.putText(frameR, "YY=" + `YY`,(xrr,(yrr+hr)+30),
cv2.FONT_HERSHEY_SIMPLEX, 0.5,(255,0,0),2);
        cv2.putText(frameR, "ZZ=" + `ZZ`,(xrr,(yrr+hr)+45),
cv2.FONT_HERSHEY_SIMPLEX, 0.5,(255,0,0),2);

        xr=xrr+(wr/2)
        yr=yrr+(hr/2)
        a1=(50*math.tan(32.5*3.14/180)*(xr-360)/360)-3.4
        hr=(50*math.tan(17*3.14/180)*(yr-202.5)/202.5)


    cv2.imshow('frame_Right',frameR)

    tfL,frameL=camL.read()
    gray = cv2.cvtColor(frameL, cv2.COLOR_BGR2GRAY)
    detect = detect_cascade.detectMultiScale(gray, 1.3, 5)
    for (xll,yll,wl,hl) in detect:
        cv2.rectangle(frameL,(xll,yll),(xll+wl,yll+hl),(255,0,0),2)
        cv2.putText(frameL, "A",(xll,yll-50), cv2.FONT_HERSHEY_SIMPLEX,
0.75,(255,0,0),2);
        cv2.putText(frameL, "X=" + `X`,(xll,yll-30), cv2.FONT_HERSHEY_SIMPLEX,
0.5,(255,0,0),2);
        cv2.putText(frameL, "Y=" + `Y`,(xll,yll-15), cv2.FONT_HERSHEY_SIMPLEX,
0.5,(255,0,0),2);
        cv2.putText(frameL, "Z=" + `Z`,(xll,yll), cv2.FONT_HERSHEY_SIMPLEX,
0.5,(255,0,0),2);

        cv2.putText(frameL, "XX=" + `XX`,(xll,(yll+hl)+15),
cv2.FONT_HERSHEY_SIMPLEX, 0.5,(255,0,0),2);
        cv2.putText(frameL, "YY=" + `YY`,(xll,(yll+hl)+30),
cv2.FONT_HERSHEY_SIMPLEX, 0.5,(255,0,0),2);
        cv2.putText(frameL, "ZZ=" + `ZZ`,(xll,(yll+hl)+45),
cv2.FONT_HERSHEY_SIMPLEX, 0.5,(255,0,0),2);

        xl=xll+(wl/2)
        yl=yll+(hl/2)
        a2=(50*math.tan(32.5*3.14/180)*(xl-360)/360)+3.4
        hl=(50*math.tan(17*3.14/180)*(yl-202.5)/202.5)


    cv2.imshow('frame_Left',frameL)

    if (a1<1000):
        if (a2<1000):
            x=((3.4*((-a2)-a1))/((-6.8)+a2-a1))
            y=(50-(340/(6.8-a2+a1)))
            X=-x
```

```
            Y=-(50-y)
            z=-(Y*hr)/50
            Z=z
            TT=1

            ZZ=Z+13
            XX=38-(Y+1)
            YY=X+28

            print 'Xa='
            print X
            print 'Ya='
            print Y
            print 'Za='
            print Z
```

```
    ac1=2000 #right
    ac2=2000 #left
    gray = cv2.cvtColor(frameR, cv2.COLOR_BGR2GRAY)
    detectc = detectc_cascade.detectMultiScale(gray, 1.3, 5)
    for (xrrc,yrrc,wrc,hrc) in detectc:
        cv2.rectangle(frameR,(xrrc,yrrc),(xrrc+wrc,yrrc+hrc),(0,0,255),2)
        cv2.putText(frameR, "Star",(xrrc,yrrc-50), cv2.FONT_HERSHEY_SIMPLEX,
0.75,(0,0,255),2);
        cv2.putText(frameR, "X=" + `Xc`,(xrrc,yrrc-30), cv2.FONT_HERSHEY_SIMPLEX,
0.5,(0,0,255),2);
        cv2.putText(frameR, "Y=" + `Yc`,(xrrc,yrrc-15), cv2.FONT_HERSHEY_SIMPLEX,
0.5,(0,0,255),2);
        cv2.putText(frameR, "Z=" + `Zc`,(xrrc,yrrc), cv2.FONT_HERSHEY_SIMPLEX,
0.5,(0,0,255),2);

        cv2.putText(frameR, "XX=" + `XXc`,(xrrc,(yrrc+hrc)+15),
cv2.FONT_HERSHEY_SIMPLEX, 0.5,(0,0,255),2);
        cv2.putText(frameR, "YY=" + `YYc`,(xrrc,(yrrc+hrc)+30),
cv2.FONT_HERSHEY_SIMPLEX, 0.5,(0,0,255),2);
        cv2.putText(frameR, "ZZ=" + `ZZc`,(xrrc,(yrrc+hrc)+45),
cv2.FONT_HERSHEY_SIMPLEX, 0.5,(0,0,255),2);

        xrc=xrrc+(wrc/2)
        yrc=yrrc+(hrc/2)
        ac1=(50*math.tan(32.5*3.14/180)*(xrc-360)/360)-3.4
        hrc=(50*math.tan(17*3.14/180)*(yrc-202.5)/202.5)
```

```
    cv2.imshow('frame_Right',frameR)


    gray = cv2.cvtColor(frameL, cv2.COLOR_BGR2GRAY)
    detectc = detectc_cascade.detectMultiScale(gray, 1.3, 5)
    for (xllc,yllc,wlc,hlc) in detectc:
        cv2.rectangle(frameL,(xllc,yllc),(xllc+wlc,yllc+hlc),(0,0,255),2)
        cv2.putText(frameL, "Star",(xllc,yllc-50), cv2.FONT_HERSHEY_SIMPLEX,
0.75,(0,0,255),2);
        cv2.putText(frameL, "X=" + `Xc`,(xllc,yllc-30), cv2.FONT_HERSHEY_SIMPLEX,
0.5,(0,0,255),2);
        cv2.putText(frameL, "Y=" + `Yc`,(xllc,yllc-15), cv2.FONT_HERSHEY_SIMPLEX,
0.5,(0,0,255),2);
        cv2.putText(frameL, "Z=" + `Zc`,(xllc,yllc), cv2.FONT_HERSHEY_SIMPLEX,
0.5,(0,0,255),2);

        cv2.putText(frameL, "XX=" + `XXc`,(xllc,(yllc+hlc)+15),
cv2.FONT_HERSHEY_SIMPLEX, 0.5,(0,0,255),2);
        cv2.putText(frameL, "YY=" + `YYc`,(xllc,(yllc+hlc)+30),
cv2.FONT_HERSHEY_SIMPLEX, 0.5,(0,0,255),2);
        cv2.putText(frameL, "ZZ=" + `ZZc`,(xllc,(yllc+hlc)+45),
cv2.FONT_HERSHEY_SIMPLEX, 0.5,(0,0,255),2);

        xlc=xllc+(wlc/2)
        ylc=yllc+(hlc/2)
        ac2=(50*math.tan(32.5*3.14/180)*(xlc-360)/360)+3.4
        hlc=(50*math.tan(17*3.14/180)*(ylc-202.5)/202.5)


    cv2.imshow('frame_Left',frameL)

    if (ac1<1000):
        if (ac2<1000):
            xc=((3.4*((-ac2)-ac1))/((-6.8)+ac2-ac1))
            yc=(50-(340/(6.8-ac2+ac1)))
            Xc=-xc
            Yc=-(50-yc)
            zc=-(Yc*hrc)/50
            Zc=zc
            TTc=2

            ZZc=Zc+13
            XXc=38-(Yc+1)
            YYc=Xc+28

            print 'Xc='
            print Xc
```

```
            print 'Yc='
            print Yc
            print 'Zc='
            print Zc









    key=cv2.waitKey(1)

    if key==ord('a'):
        ser1.write(struct.pack('>BBBB',XX,YY,ZZ,TT))

    if key==ord('s'):
        ser1.write(struct.pack('>BBBB',XXc,YYc,ZZc,TTc))

    if key==ord('k'):
            break

camR.release()
camL.release()
cv2.destroyAllWindows()
```