

## LAMPIRAN

```
import cv2
import argparse
import sys
import math
import numpy as np
import filterpy

#####
#####

keep_processing = True
selection_in_progress = False
fullscreen = False

parser = argparse.ArgumentParser(description='Perform '
+ sys.argv[0] + ' example operation on incoming
camera/video image')
parser.add_argument("-c", "--camera_to_use", type=int,
help="specify camera to use", default=0)
parser.add_argument('video_file', metavar='video_file',
type=str, nargs='?', help='specify optional video
file')
args = parser.parse_args()

#####
#####

boxes = []
current_mouse_position = np.ones(2, dtype=np.int32)

def on_mouse(event, x, y, flags, params):

    global boxes
    global selection_in_progress

    current_mouse_position[0] = x
    current_mouse_position[1] = y

    if event == cv2.EVENT_LBUTTONDOWN:
        boxes = []
        sbox = [x, y]
        selection_in_progress = True
        boxes.append(sbox)
```

```

elif event == cv2.EVENT_LBUTTONUP:
    ebox = [x, y]
    selection_in_progress = False
    boxes.append(ebox)
#####
#####

def center(points):
    x = np.float32((points[0][0] + points[1][0] +
points[2][0] + points[3][0]) / 4.0)
    y = np.float32((points[0][1] + points[1][1] +
points[2][1] + points[3][1]) / 4.0)
    return np.array([np.float32(x), np.float32(y)],
np.float32)

#####
#####

def nothing(x):
    pass

#####
#####

cap = cv2.VideoCapture(1)

windowName = "Kalman Object Tracking"
windowName2 = "Hue histogram back projection"
windowNameSelection = "initial selected region"

kalman = cv2.KalmanFilter(4,2)
kalman.measurementMatrix = np.array([[1,0,0,0],
[0,1,0,0]],np.float32)

kalman.transitionMatrix = np.array([[1,0,1,0],
[0,1,0,1],
[0,0,1,0],
[0,0,0,1]],np.float32)

kalman.processNoiseCov = np.array([[1,0,0,0],
[0,1,0,0],
[0,0,1,0],

```

```

[0,0,0,1]],np.float32) * 0.03

measurement = np.array((2,1), np.float32)
prediction = np.zeros((2,1), np.float32)

print("\nObservation in image: BLUE")
print("Prediction from Kalman: GREEN\n")

if ((args.video_file and
(cap.open(str(args.video_file))))
or (cap.open(args.camera_to_use))):

    cv2.namedWindow(windowName, cv2.WINDOW_NORMAL)
    cv2.namedWindow(windowName2, cv2.WINDOW_NORMAL)
    cv2.namedWindow(windowNameSelection,
cv2.WINDOW_NORMAL)

    s_lower = 60
    cv2.createTrackbar("s lower", windowName2, s_lower,
255, nothing)
    s_upper = 255
    cv2.createTrackbar("s upper", windowName2, s_upper,
255, nothing)
    v_lower = 32
    cv2.createTrackbar("v lower", windowName2, v_lower,
255, nothing)
    v_upper = 255
    cv2.createTrackbar("v upper", windowName2, v_upper,
255, nothing)

    cv2.setMouseCallback(windowName, on_mouse, 0)
    cropped = False

    term_crit = ( cv2.TERM_CRITERIA_EPS |
cv2.TERM_CRITERIA_COUNT, 10, 1 )

    while keep_processing:

        if cap.isOpened:
            ret, frame = cap.read()
            frame = cv2.flip(frame, 1)

            start_t = cv2.getTickCount()

```

```

        s_lower = cv2.getTrackbarPos("s lower",
windowName2)
        s_upper = cv2.getTrackbarPos("s upper",
windowName2)
        v_lower = cv2.getTrackbarPos("v lower",
windowName2)
        v_upper = cv2.getTrackbarPos("v upper",
windowName2)

        if (len(boxes) > 1) and (boxes[0][1] <
boxes[1][1]) and (boxes[0][0] < boxes[1][0]):
            crop =
frame[boxes[0][1]:boxes[1][1],boxes[0][0]:boxes[1][0]].
copy()

            h, w, c = crop.shape
            if (h > 0) and (w > 0):
                cropped = True

                hsv_crop = cv2.cvtColor(crop,
cv2.COLOR_BGR2HSV)

                mask = cv2.inRange(hsv_crop,
np.array((0., float(s_lower),float(v_lower))),
np.array((180.,float(s_upper),float(v_upper))))

                crop_hist = cv2.calcHist([hsv_crop],[0,
1],mask,[180, 255],[0,180, 0, 255])

cv2.normalize(crop_hist,crop_hist,0,255,cv2.NORM_MINMAX
)

                track_window =
(boxes[0][0],boxes[0][1],boxes[1][0] -
boxes[0][0],boxes[1][1] - boxes[0][1])

                cv2.imshow(windowNameSelection,crop)

        boxes = []

        if selection_in_progress:
            top_left = (boxes[0][0], boxes[0][1])
            bottom_right = (current_mouse_position[0],
current_mouse_position[1])
            cv2.rectangle(frame,top_left, bottom_right,
(0,255,0), 2)

```

```

        if cropped:

            img_hsv = cv2.cvtColor(frame,
cv2.COLOR_BGR2HSV)

            img_bproject =
cv2.calcBackProject([img_hsv],[0,1],crop_hist,[0,180,0,
255],1)

            cv2.imshow(windowName2,img_bproject)

            ret, track_window =
cv2.CamShift(img_bproject, track_window, term_crit)

            x,y,w,h = track_window
            frame = cv2.rectangle(frame, (x,y),
(x+w,y+h), (255,0,0),2)

            pts = cv2.boxPoints(ret)
            pts = np.int0(pts)

            kalman.correct(center(pts))

            prediction = kalman.predict()

            frame = cv2.rectangle(frame,
(prediction[0]-(0.5*w),prediction[1]-(0.5*h)),
(prediction[0]+(0.5*w),prediction[1]+(0.5*h)),
(0,255,0),2)

            l=(prediction[0]-(x+int(w/2)))
            t=(prediction[1]-(y+int(h/2)))
            s=math.sqrt((l**2)+(t**2))

            a=(l/s)
            teta=np.arcsin(a)

            px = prediction[0]+l*10
            py = prediction[1]+t*10

            print("prediksi x = ", px)
            print("prediksi y = ", py)
            print("sudut = ", teta)

cv2.arrowedLine(frame, (x+int(w/2),y+int(h/2)), (px,py),

```

```

(255,0,255),thickness=2, line_type=8, shift=0,
tipLength=0.1)

    else:

        img_hsv = cv2.cvtColor(frame,
cv2.COLOR_BGR2HSV)

        mask = cv2.inRange(img_hsv, np.array((0.,
float(s_lower),float(v_lower))),
np.array((180.,float(s_upper),float(v_upper))))

        cv2.imshow(windowName2,mask)

        cv2.imshow(windowName,frame)
        cv2.setWindowProperty(windowName,
cv2.WND_PROP_FULLSCREEN, cv2.WINDOW_FULLSCREEN &
fullscreen)

        stop_t = ((cv2.getTickCount() -
start_t)/cv2.getTickFrequency()) * 1000

        key = cv2.waitKey(max(2, 40 -
int(math.ceil(stop_t)))) & 0xFF

        if key == ord('x'):
            keep_processing = False
        elif key == ord('f'):
            fullscreen = not(fullscreen)

        cv2.destroyAllWindows()
        cap.release()
else:
    print("No video file specified or camera
connected.")

#####
#####

```



## LEMBAR REVISI dan TUGAS UJIAN SARJANA

Berdasarkan Rapat Tim Penguji Ujian Sarjana

Hari : Senin  
Tanggal : 30 September 2019  
Tempat : R. Sidang

Memutuskan bahwa mahasiswa :

Nama : Daniyah  
NIM : 30601501703  
Judul TA : Prediksi Arah Datang Bola pada Robot Kiper Sepak Bola Beroda dengan Extended Kalman Filter (EKF)

wajib melakukan perbaikan dan membuat tugas seperti tercantum dibawah ini:

NO	REVISI	BATAS REVISI
1.	Dijahani alur & Rumus yg (diteliti) H, S, V, Kalman.	Acc. 10/10/19
2.	Buat Flowchart & matriks & Algoritma: HSV & Kalman	
3.	Kemampuan harus ada alasan kenapa dapat dilengkap prediksi	
4.	Harus yg (extended), kon hanya Kalman Filter	

NO	TUGAS
5.	Belajar term algoritmanya & jagan hanya coba-coba & bikin alur

Mengetahui,  
Ketua Tim Penguji

  
Dr. Hj. Sri Arttini Dwi P., M.Si  
NIDN. 0620026501

Semarang, 30 September 2019  
Penguji, I

  
Dr. Hj. Sri Arttini Dwi P., M.Si  
NIDN. 0620026501



### LEMBAR REVISI dan TUGAS UJIAN SARJANA

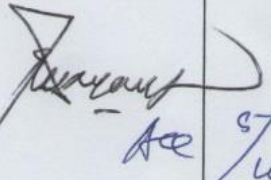
Berdasarkan Rapat Tim Penguji Ujian Sarjana

Hari : Senin  
Tanggal : 30 September 2019  
Tempat : R. Sidang

Memutuskan bahwa mahasiswa :

Nama : Daniyah  
NIM : 30601501703  
Judul TA : Prediksi Arah Datang Bola pada Robot Kiper Sepak Bola Beroda dengan Extended Kalman Filter (EKF)

wajib melakukan perbaikan dan membuat tugas seperti tercantum dibawah ini:

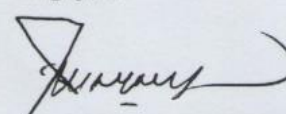
NO	REVISI	BATAS REVISI
	Numbering romawi Pembahasan disesuaikan dengan hal 2 dan komponen yg dibahas saja, termasuk di teori bab 2 Tambahkan teori tentang CITRA → RGB, grayscale, thresholding.	 30/9/19

NO	TUGAS
	Jelaskan Citra → 24 bit, 8 bit, proses pengolahan Citra (RGB to grayscale, grayscale to biner, thresholding)

Mengetahui,  
Ketua Tim Penguji

  
Dr. Hj. Sri Artini Dwi P., M.Si  
NIDN. 0620026501

Semarang, 30 September 2019  
Penguji, II

  
Eka Nuryanto Budisusila, ST, MT  
NIDN. 0619107301



YAYASAN BADAN WAKAF SULTAN AGUNG  
UNIVERSITAS ISLAM SULTAN AGUNG (UNISSULA)

Jl. Raya Kaligawe Km.4 Semarang 50112 Telp.(024) 6583584 (8 Sal) Fax.(024) 6582455  
email: [informasi@unissula.ac.id](mailto:informasi@unissula.ac.id) web : [www.unissula.ac.id](http://www.unissula.ac.id)

Fakultas Teknologi Industri

Bismillah Membangun Generasi Khaira Ummah

**LEMBAR REVISI dan TUGAS UJIAN SARJANA**

Berdasarkan Rapat Tim Penguji Ujian Sarjana

Hari : Senin  
Tanggal : 30 September 2019  
Tempat : R. Sidang

Memutuskan bahwa mahasiswa :

Nama : Daniyah  
NIM : 30601501703  
Judul TA : Prediksi Arah Datang Bola pada Robot Kiper Sepak Bola Beroda dengan Extended Kalman Filter (EKF)

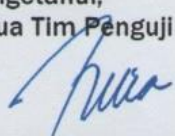
wajib melakukan perbaikan dan membuat tugas seperti tercantum dibawah ini:

NO	REVISI	BATAS REVISI
	<ul style="list-style-type: none"><li>- Abstrak → hasil.</li><li>- Daftar masalah ?</li><li>- penulisan gambar, tabel.</li><li>- Daftar pustaka → publisher.</li></ul>	

15/10  
10  
A. S. H.

NO	TUGAS
	Demo Alet sebau' judul - ✓

Mengetahui,  
Ketua Tim Penguji

  
Dr. Hj. Sri Artini Dwi P., M.Si  
NIDN. 0620026501

Semarang, 30 September 2019  
Penguji, III

  
Muhammad Khosyul In, ST, MT  
NIDN. 0625077901

# Prediksi Arah Datang Bola

by Daniyah Daniyah

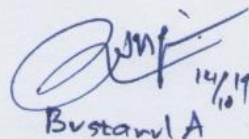
**Submission date:** 29-Sep-2019 07:23AM (UTC+0800)

**Submission ID:** 1181963575

**File name:** Prediksi\_Arah\_Datang\_Bola\_Daniyah.docx (3.68M)

**Word count:** 13739

**Character count:** 77713

  
Bustanul A

# Prediksi Arah Datang Bola

*Bustanul A*  
14/19

## ORIGINALITY REPORT

**16%**

SIMILARITY INDEX

**12%**

INTERNET SOURCES

**2%**

PUBLICATIONS

**11%**

STUDENT PAPERS

## PRIMARY SOURCES

1	<b>repository.its.ac.id</b> Internet Source	2%
2	<b>Submitted to Universitas Brawijaya</b> Student Paper	1%
3	<b>Submitted to Universitas Semarang</b> Student Paper	1%
4	<b>Submitted to Sultan Agung Islamic University</b> Student Paper	1%
5	<b>media.neliti.com</b> Internet Source	1%
6	<b>id.123dok.com</b> Internet Source	1%
7	<b>fit.labs.telkomuniversity.ac.id</b> Internet Source	1%
8	<b>iitmvcvg.github.io</b> Internet Source	1%
9	<b>www.scribd.com</b> Internet Source	1%