

[https://wiki.itcollege.ee/index.php/Pilditöötlus\\_OpenCV\\_abil](https://wiki.itcollege.ee/index.php/Pilditöötlus_OpenCV_abil)

## Pilditöötlus OpenCV abil

### Andmete kodeerimise praktikum

Andmete kodeerimise loeng päevõppes toimub 20. aprillil, esitlus

siin: [https://docs.google.com/presentation/d/1QV\\_IMOadIEUTBW7Laqw59j8J7Y8HKwvSDmP-swEEx0s](https://docs.google.com/presentation/d/1QV_IMOadIEUTBW7Laqw59j8J7Y8HKwvSDmP-swEEx0s)

Prakikumis keskendume pilditöötlusele, et natukene aru saada mis moodi bitid/baidid liiguvad video ja pildifailides, selleks:

1. Laadi alla Robotex 2016 testpäeva videofailid aadressilt <http://upload.itcollege.ee/robotex-test-day/>
2. Kasuta programmi toorikuna allpool olevat koodijuppi, salvesta see faili pildituvastus.py
3. Käivita programm käsutealt: python pildituvastus.py failinimi.avi
4. Püüa leida vastused järgnevatele küsimustele
5. Saada oma vastused ja modifikatsioonidega programm lauri.vosandi@itcollege.ee aadressile ning märgi e-kirja teemaks "I027 Andmete kodeerimine praktikum"

Küsimused:

1. Mis on kaadri (frame) eraldusvõime (resolution)?
  2. Mis piksliformaadis on cap.read() väljastatud kaader?
  3. Mis on kaadrisagedus (framerate)? Mis on ühe kaadri kuvamise aeg?
  4. Mis on kaadri suurus baitides?
  5. Mitu kaadrit on failis? Kui pikk on videolõik ajaliselt?
  6. Kui salvestada kõik kaadrid pakkimata kujul kettale, palju need ruumi võtaksid?
  7. Kui palju ruumi säästab video pakkimine?
  8. Tee kaadri loendur türkiisiniseks [https://en.wikipedia.org/wiki/Turquoise\\_\(color\)#Turquoise\\_Blue](https://en.wikipedia.org/wiki/Turquoise_(color)#Turquoise_Blue)
  9. Kohenda koodi nii et tuvastataks pallid
  10. Täienda koodi et leitaks sinine värav
  11. Täienda koodi et leitaks kollane värav
  12. Mis on palli ja väravate tuvastamiseks sobilik värvikoodide vahemik HSV või YUV värviruumis? Selgita! Lisaülesanded nuputamiskes, kui seni meeldis:
1. Teades et kaamera on 22cm kõrgusel ning vaatenurk ülevalt horisondist alla on 72 kraadi arvuta välja palli kaugus ning kuva see palli kohal
  2. Leia palli nurk löögimehhanismist
  3. Leia palli vektor löögimehhanismi suhtes
  4. Millisele pallile kõige esimesena läheneda võiks praeguse kaadri järgi? Märgi ära see ekraanil.

Vihjed:

- Kasuta print lauset muutujate kuvamiseks terminalisse
- Kaader on sisuliselt mitmemõõteline massiiv
- Kes soovib oma Ubuntu masinas programmi käima ajada paigalda esmalt sõltuvused: apt install python-opencv python-numpy

```
# encoding: utf-8
import cv2
import os
```

```

import numpy as np
import sys
from time import sleep

try:
    _, filename = sys.argv
except ValueError:
    print("Anna argumendiks failinimi mida töödelda!")
    sys.exit(255)

print "Avan faili:", filename
cap = cv2.VideoCapture(filename)
fps = cap.get(5)
success, frame = cap.read()

# Kommenteeri need välja ja uuri välja mis need teevad?
#print frame
#print frame.shape
#print frame.dtype

frameno = 0
while True:

    # Loe ja paki lahti kaader failist
    success, frame = cap.read()
    frameno += 1
    if not success:
        break

    # Piksliformaadi teisendus
    hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
    yuv = cv2.cvtColor(frame, cv2.COLOR_RGB2YCR_CB)

    # Leia väärtuste vahemikku langevad pikslid
    mask = cv2.inRange(frame, (100,0,0), (255,100,100))
    #mask = cv2.inRange(hsv, (200,200,150), (255,255,255))

    # Eemalda üksikud pikslid
    mask = cv2.erode(mask, np.ones((2, 2)))

    # Maski pikslite visualiseerimiseks lõika need välja esialgsest kaadrist
    cutout = cv2.bitwise_and(frame, frame, mask=mask)

    # Joonista kaadriloendur
    cv2.putText(frame, "frame: %d" % frameno, (30, 50), cv2.FONT_HERSHEY_SIMPLEX, 1,
(0,0,255), 3)

    # Moodusta kaadrite pinu
    stacked = np.vstack([
        frame,
        np.stack([mask]*3, axis=2), # See on maski näitamiseks
        cutout,

```

```

1)
# Leia maskist kontuurid
contours, hierarchy = cv2.findContours(mask, cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)

# Visualiseeri kontuurid mida ümbristeva ristküliku suurus on suurem kui 5x5
pisklit
for cnt in contours:
    x,y,w,h = cv2.boundingRect(cnt)
    if w > 5 and h > 5:
        cv2.rectangle(stacked, (x,y), (x+w,y+h), (0,255,0), 2)
        cv2.putText(stacked, "pall", (x,y), cv2.FONT_HERSHEY_SIMPLEX, 1,
(255,255,255), 3)

# Tee pilt pisemaks et ekraanile mahuks
zoomed = cv2.resize(stacked, None, fx=0.5, fy=0.5)

# Kuva kaader ekraanil
cv2.imshow("fov", zoomed)

# Lõpeta programmi täitmine kui klaviatuuril vajutatakse suvalist nuppu
keycode = cv2.waitKey(int(1000/fps))
if keycode >= 0:
    break

cap.release()

```