



## Pracoviště automatizované kontroly výstupu vibračních kruhových zásobníků

### Příloha 2 – Zdrojový kód programu

```
import cv2
import numpy as np
import matplotlib.pyplot as plt
import imutils
import copy
import math
from Tkinter import *
import tkFileDialog as FileDialog
import os

global zaznam
zaznam = 1

cap = cv2.VideoCapture()
cap.set(3, 1280)
cap.set(4, 720)

class Application(Frame):
    def __init__(self, master):
        zaznam = BooleanVar()
        Frame.__init__(self, master)
        self.grid()
        self.create_widgets()

    def create_widgets(self):
        _, frame_cam = cap.read()
        height, width, channels = frame_cam.shape

        self.instruction = Label(self, text = "Vítejte, nastavte příslušné parametry a spusťte zaznam:")
        self.instruction.grid()
        self.submit_button = Button(self, text = "1) Kalibrace", command = self.kalibrace).grid()

        self.submit_button = Button(self, text = "Uložit", command = self.files).grid()
        self.submit_button = Button(self, text = "Nacist", command = self.fileo).grid()

        Label(self, text = "SOUCAST - DETEKCNI OBLAST:").grid(sticky = W)
        Label(self, text = "Odsazení vlevo:").grid(sticky = W)
        self.var1 = IntVar()
        Slider1 = Scale(self, orient=HORIZONTAL,length=width,width=20,sliderlength=10,
            from_=0, to=width, variable=self.var1, tickinterval=50).grid(sticky = W)
        self.var1.set(140)

        Label(self, text = "Odsazení shora:").grid(sticky = W)
        self.var2 = IntVar()
        Slider2 = Scale(self, orient=HORIZONTAL,length=height,width=20,sliderlength=10,
            from_=0, to=height, variable=self.var2, tickinterval=50).grid(sticky = W)
        self.var2.set(150)

        Label(self, text = "Sirka:").grid(sticky = W)
        self.var3 = IntVar()
        Slider3 = Scale(self, orient=HORIZONTAL,length=width,width=20,sliderlength=10,
            from_=0, to=width, variable=self.var3, tickinterval=50).grid(sticky = W)
        self.var3.set(170)

        Label(self, text = "Vyska:").grid(sticky = W)
        self.var4 = IntVar()
        Slider4 = Scale(self, orient=HORIZONTAL,length=height,width=20,sliderlength=10,
            from_=0, to=height, variable=self.var4, tickinterval=50).grid(sticky = W)
        self.var4.set(130)

        Label(self, text = "OBRAZ - PRACOVNI OBLAST:").grid(sticky = W)
        Label(self, text = "Horní hrana:").grid(sticky = W)
        self.var5 = IntVar()
```



```
Slider3 = Scale(self, orient=HORIZONTAL,length=height,width=20,sliderlength=10,
    from_=0, to=height, variable=self.var5, tickinterval=50).grid(sticky = W)
self.var5.set(150)

Label(self, text = "Spodni hrana:").grid(sticky = W)
self.var6 = IntVar()
Slider4 = Scale(self, orient=HORIZONTAL,length=height,width=20,sliderlength=10,
    from_=0, to=height, variable=self.var6, tickinterval=50).grid(sticky = W)
self.var6.set(350)

self.submit_button = Button(self, text = "2) Vzorke (podrz ESC pro konec)", command =
    self.snimky).grid()

self.left = BooleanVar()
Checkbutton(self, text="Soucasti se pohybují zleva doprava", variable =
    self.left).grid()

Label(self, text = "Nazev videa:").grid()
self.video_name=StringVar(root, value='Default')
self.name = Entry(self, textvariable=self.video_name)
self.name.grid()

self.rec = BooleanVar()
Checkbutton(self, text="Zaznamenat obraz do videa", variable = self.rec).grid()

self.submit_button = Button(self, text = "3) Spustit sledovani (podrz ESC pro konec)",
    command = self.main)
self.submit_button.grid()

def kalibrace (self):
    _, frame_cam = cap.read()

    cv2.rectangle(frame_cam, (self.var1.get(),self.var2.get()), (self.var3.get()+self.var1.
        get(),self.var4.get()+self.var2.get()), (0,0,255),5)
    cv2.rectangle(frame_cam, (0,self.var5.get()), (640,self.var6.get()), (255,0,0),2)
    plt.imshow(frame_cam)
    plt.show()

def snimky (self):
    font = cv2.FONT_HERSHEY_SIMPLEX
    k=20
    while (1):
        _, frame_cam = cap.read()
        if frame_cam is None:
            print 'Konec'
            break
        crop_kalibr = frame_cam[self.var2.get():self.var4.get()+self.var2.get(),
            self.var1.get():self.var3.get()+self.var1.get()]
        cv2.imshow('foto_C',crop_kalibr)
        if cv2.waitKey(1) & 0xFF == ord('a'):
            cv2.imwrite('kalibr0.png',crop_kalibr)
            k=0
        if cv2.waitKey(1) & 0xFF == ord('s'):
            cv2.imwrite('kalibr1.png',crop_kalibr)
            k=0
        if cv2.waitKey(1) & 0xFF == ord('d'):
            cv2.imwrite('kalibr2.png',crop_kalibr)
            k=0
        if cv2.waitKey(1) & 0xFF == ord('f'):
            cv2.imwrite('kalibr3.png',crop_kalibr)
            k=0
        if k<10:
            cv2.putText(frame_cam, '_____OK!', (0,25), font, 1, (200,255,155), 2,
                cv2.LINE_AA)
            k=k+1

        cv2.rectangle(frame_cam, (self.var1.get(),self.var2.get()), (self.var3.get()+self.var1.
            get(),self.var4.get()+self.var2.get()), (0,0,255),5)
        cv2.imshow('foto', frame_cam)

        q = cv2.waitKey(1) & 0xFF
```

```
        if q == 27:
            break
cv2.destroyAllWindows()

def files (self):
    name=FileDialog.asksaveasfile(mode='w',defaulttextextension=".txt")

    text=[self.var1.get(),self.var2.get(),self.var3.get(),self.var4.get(),self.var5.get()
    ,self.var6.get()]
    print text
    name.write(str(text))
    name.write('\n\n'+[Vlevo/shora/sirka/vyska/horni hrana/spodni hrana'])
    name.close

def fileo (self):
    filename = FileDialog.askopenfilename( filetypes = (("config files", "*.txt"),("All
    files", "*..*")))
    text_file = open(filename,"r")
    config=text_file.readline()
    config=config.replace("\n", "")
    config=config.replace(" ", "")
    config=config.replace("[", "")
    config=config.replace("]", "")
    config=config.split(',')
    text_file.close()
    num=[0,0,0,0,0,0]
    for i in range (6):
        num[i]=int(config[i])
    self.var1.set(num[0])
    self.var2.set(num[1])
    self.var3.set(num[2])
    self.var4.set(num[3])
    self.var5.set(num[4])
    self.var6.set(num[5])

def main (self):
    _, frame_cam = cap.read()
    frame = frame_cam[self.var5.get():self.var6.get(), 0:640]
    cv2.imwrite('test0.png',frame)
    global screen
    screen = 0
    casti = 1
    bad = []
    souradnice_prev = [0,0,0,1,-1]
    shape_prev=0
    zaznam = self.rec.get()
    left = self.left.get()

    if zaznam:
        vider=[0,0]
        fourcc = cv2.VideoWriter_fourcc('M','J','P','G')
        timing = 0
        video = 0

    while (True):
        screen = screen+1
        ret, frame_cam = cap.read(1)
        if frame_cam is None:
            print 'Konec'
            break
        frame = frame_cam[self.var5.get():self.var6.get(), 0:640]

        frame_orig = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
        crop = frame_cam[200:300, 150:300]

        template = cv2.imread('test0.PNG',1)
        diff = cv2.absdiff(frame,template,50)
        thresh0 = cv2.threshold(diff, 40, 255, cv2.THRESH_BINARY)[1]
        thresh = cv2.cvtColor(thresh0, cv2.COLOR_BGR2GRAY)

        templatel = cv2.imread('kalibr0.PNG',0)
```

```
template2 = cv2.imread('kalibr1.PNG',0)
template3 = cv2.imread('kalibr2.PNG',0)

w1, h1 = template1.shape[::-1]
w2, h2 = template2.shape[::-1]
w3, h3 = template3.shape[::-1]

res = cv2.bitwise_and(frame_orig, frame_orig, mask= thresh)
blurred = cv2.GaussianBlur(res, (5, 5), 0)
thresh_shape = cv2.threshold(blurred, 60, 255, cv2.THRESH_BINARY)[1]

cnts = cv2.findContours(thresh_shape.copy(), cv2.RETR_EXTERNAL,
                        cv2.CHAIN_APPROX_SIMPLE)
cnts = cnts[0] if imutils.is_cv2() else cnts[1]

shape = 0
safe = w1/3
plocha_ref = w1*h1
det_hranice = plocha_ref-plocha_ref/6
det_hranice_h = plocha_ref+plocha_ref/6

souradnice = []

for c in cnts:
    M = cv2.moments(c)
    if M["m00"] >= det_hranice:
        cX = int(M["m10"] / M["m00"])
        cY = int(M["m01"] / M["m00"])
        print M["m00"]
        k=int((M["m00"]/plocha_ref)+0.5)
        mod=k%2
        for p in range(1,k+1):
            dist=int(cX+((-k/2)-mod+p)*w1-(0.5*w1*abs(mod-1)))
            if (dist < 0.6*w1 and left==0) or (dist > 640-(0.6*w1) and left==1):
                continue
            temp=[]
            for j in range(6):
                temp.append(0)
            souradnice.append(temp)

            souradnice[shape][5]=k
            souradnice[shape][1] = dist
            souradnice[shape][2] = cY
            souradnice[shape][0] = screen

            if shape_prev>shape:
                if abs(souradnice[shape][1] - souradnice_prev[shape][1])
<=safe:
                    souradnice[shape][3] = souradnice_prev[shape][3]
                    souradnice[shape][4] = souradnice_prev[shape][4]
            else:
                souradnice[shape][3] =0
                souradnice[shape][4] =-1
                shape = shape+1

if shape <= shape_prev:
    for c in range(shape):
        if abs(souradnice[c][1] - souradnice_prev[c][1]) >safe:
            for i in range(shape_prev):
                if abs(souradnice[c][1] - souradnice_prev[i][1]) <=safe:
                    souradnice[c][3]=souradnice_prev[i][3]
                    souradnice[c][4]=souradnice_prev[i][4]

if shape > shape_prev:
    for c in range(shape_prev):
        if abs(souradnice[c][1] - souradnice_prev[c][1]) >safe:
            found=0
            for i in range(shape_prev):
                if abs(souradnice[c][1] - souradnice_prev[i][1]) <=safe:
                    souradnice[c][3]=souradnice_prev[i][3]
                    souradnice[c][4]=souradnice_prev[i][4]
```

```
        found=1
    if found==0:
        souradnice[c][3]=casti
        casti=casti+1
        souradnice[c][4] ==-1
    rozdil = shape - shape_prev
    for c in range(rozdil):
        for i in range(shape_prev):
            if abs(souradnice[shape-1-c][1] - souradnice_prev[i][1]) <=safe and
abs(souradnice[shape-1-c][2] - souradnice_prev[i][2]) <=safe:
                souradnice[shape-1-c][3]=souradnice_prev[i][3]
                souradnice[shape-1-c][4]=souradnice_prev[i][4]
    for c in range(shape):
        if souradnice[c][3]==0:
            souradnice[c][3]=casti
            casti=casti+1
            souradnice[c][4] ==-1
    for c in cnts:
        M = cv2.moments(c)
        if M["m00"] >= det_hranice:
            cv2.drawContours(frame, [c], 0, (255, 0, 0), 1)

    for c in range(shape):
        if souradnice[c][4] ==-1:
            col=(255, 0, 0)
        if souradnice[c][4] ==1:
            col=(0, 255, 0)
        if souradnice[c][4] ==0:
            col=(0, 0, 255)
        cv2.circle(frame, (souradnice[c][1], souradnice[c][2]), 7, col, -1)
        cv2.putText(frame, str(souradnice[c][3]), (souradnice[c][1] - 20,
souradnice[c][2] - 20),
                    cv2.FONT_HERSHEY_SIMPLEX, 0.75, (0, 0, 255), 2)
        cv2.putText(frame, str(souradnice[c][5]), (souradnice[c][1] - 20,
souradnice[c][2] + 20),
                    cv2.FONT_HERSHEY_SIMPLEX, 0.75, (255, 0, 255), 2)
    font = cv2.FONT_HERSHEY_SIMPLEX
    det1 = cv2.matchTemplate(frame_orig, template1, cv2.TM_CCORR_NORMED)
    threshold = 0.95
    loc = np.where( det1 >= threshold)
    i=0
    for d in zip(*loc[::-1]):
        i=i+1
    part=np.zeros((i, 2), dtype=np.int)
    i=0
    for pt in zip(*loc[::-1]):
        part[i][0]=pt[0] + w2/2
        part[i][1]=pt[1] + h2/2
        i=i+1
        cv2.rectangle(frame, pt, (pt[0] + w1, pt[1] + h1), (0,255,255), 1)
        cv2.putText(frame, 'OK!', (0,25), font, 1, (200,255,155), 2, cv2.LINE_AA)
    for p in range(i):
        for k in range(i):
            if k!=p:
                if abs(part[p][0]-part[k][0])<w1/4 and abs(part[p][1]-
part[k][1])<h1/4:
                    part[p]=np.zeros(2, dtype=np.int)
    for p in range(i):
        if (part[i-p-1]==np.zeros(2, dtype=np.int)).all():
            part=np.delete(part, (i-p-1), axis=0)

    distance_part=math.sqrt(w1*w1+h1*h1)/4

    for a in range(shape):
        for b in range(part.shape[0]):
            x=abs(souradnice[a][1]-part[b][0])
            y=abs(souradnice[a][2]-part[b][1])
            distance=math.sqrt(y*y+x*x)
            if distance<distance_part:
                souradnice[a][4] = 1
```

```
det2 = cv2.matchTemplate(frame_orig,template2,cv2.TM_CCORR_NORMED)
threshold = 0.95
loc = np.where( det2 >= threshold)
for pt in zip(*loc[::-1]):
    cv2.rectangle(frame, pt, (pt[0] + w2, pt[1] + h2), (0,0,255), 1)
    cv2.putText(frame,'NOK!',(0,25), font, 1, (200,255,155), 2, cv2.LINE_AA)

det3 = cv2.matchTemplate(frame_orig,template3,cv2.TM_CCORR_NORMED)
threshold = 0.95
loc = np.where( det3 >= threshold)
for pt in zip(*loc[::-1]):
    cv2.rectangle(frame, pt, (pt[0] + w3, pt[1] + h3), (255,0,255), 1)
    cv2.putText(frame,'NOK',(0,25), font, 1, (200,255,155), 2, cv2.LINE_AA)

for a in range(shape):
    if souradnice[a][4] == -1 and ((souradnice[a][1]<self.var1.get() and left ==0)
or (souradnice[a][1]>(self.var1.get()+self.var3.get()) and left==1)):
        souradnice[a][4]=0
        if not zaznam:
            video = -1
            temp=[screen,video,souradnice[a][3]]
            bad.append(temp)
        if zaznam:
            vider[video%2]=1

    cv2.putText(frame_cam,"Frame:"+str(screen),(0,40), font, 0.9, (0,0,255), 1,
cv2.LINE_AA)
    cv2.putText(frame_cam,"Total:"+str(casti-1),(0,80), font, 0.9, (0,0,255), 1,
cv2.LINE_AA)
    cv2.putText(frame_cam,"NOK:"+str(len(bad)),(0,120), font, 0.9, (0,0,255), 1,
cv2.LINE_AA)

    cv2.putText(frame_cam,"In:"+str(shape),(300,40), font, 0.9, (0,0,255), 1,
cv2.LINE_AA)

    if shape!=0:
        print souradnice, safe

    souradnice_prev = copy.deepcopy(souradnice)
    shape_prev = copy.deepcopy(shape)

cv2.rectangle(frame_cam,(self.var1.get(),self.var2.get()),(self.var3.get()+self.var1.
get(),self.var4.get()+self.var2.get()),(255,0,0),5)
cv2.rectangle(frame_cam,(0,self.var5.get()),(640,self.var6.get()),(0,0,255),2)

cv2.imshow('thresh_shape',thresh_shape)
cv2.imshow('mainwind',frame_cam)

if zaznam:
    fps = 18.0
    name=self.video_name.get()
    if 'out' not in locals():
        out = cv2.VideoWriter(name+'_'+str(video)+'.avi',fourcc, fps, (640,480))
    else:
        if timing>(30.0*fps):
            if video!=0 and vider==[0,0]:
                os.remove(name+'_'+str(video-1)+'.avi')
            if (vider[1-video%2]==-1):
                vider[video%2]=0
            if (vider[video%2]==1):
                vider[1-video%2]=-1
            if (vider[video%2]==0 and vider[1-video%2]==-1):
                vider=[0,0]
            video=video+1
            out.release()
            timing=0
            out = cv2.VideoWriter(name+'_'+str(video)+'.avi',fourcc, fps,
(640,480))
            timing=timing+1
            out.write(frame_cam)
```



```
k = cv2.waitKey(1) & 0xFF
if k == 27:
    break

cv2.destroyAllWindows()
print bad
bad_file = open("bad_file.txt", "w")
bad_file.write(str(bad))
bad_file.write('\n\n'+[Snimek/video/soucast_cislo]')
bad_file.close
if zaznam:
    out.release()

root = Tk()
app=Application(root)
root.mainloop()
cap.release()
```