People Counter 9 – Counting

web.archive.org/web/20180201143549/http://www.femb.com.mx/people-counter/people-counter-9-counting

Fede

January 26, 2017

Here's code:

```
##Contador de personas
##Federico Mejia
import numpy as np
import cv2
import Person
import time

#Contadores de entrada y salida
cnt_up   = 0
cnt_down = 0

#Fuente de video
#cap = cv2.VideoCapture(0)
cap = cv2.VideoCapture('peopleCounter.avi')

#Propiedades del video
##cap.set(3,160) #Width
##cap.set(4,120) #Height

#Imprime las propiedades de captura a consola
for i in range(19):
    print i, cap.get(i)

w = cap.get(3)
h = cap.get(4)
frameArea = h*w
areaTH = frameArea/250
print 'Area Threshold', areaTH

#Lineas de entrada/salida
line_up = int(2*(h/5))
line_down   = int(3*(h/5))

up_limit =   int(1*(h/5))
down_limit = int(4*(h/5))

print "Red line y:",str(line_down)
print "Blue line y:", str(line_up)
line_down_color = (255,0,0)
line_up_color = (0,0,255)
pt1 =  [0, line_down];
pt2 =  [w, line_down];
pts_L1 = np.array([pt1,pt2], np.int32)
pts_L1 = pts_L1.reshape((-1,1,2))
pt3 =  [0, line_up];
```

```
pt4 =  [w, line_up];
pts_L2 = np.array([pt3,pt4], np.int32)
pts_L2 = pts_L2.reshape((-1,1,2))

pt5 =  [0, up_limit];
pt6 =  [w, up_limit];
pts_L3 = np.array([pt5,pt6], np.int32)
pts_L3 = pts_L3.reshape((-1,1,2))
pt7 =  [0, down_limit];
pt8 =  [w, down_limit];
pts_L4 = np.array([pt7,pt8], np.int32)
pts_L4 = pts_L4.reshape((-1,1,2))

#Substractor de fondo
fgbg = cv2.createBackgroundSubtractorMOG2(detectShadows = True)

#Elementos estructurantes para filtros morfoogicos
kernelOp = np.ones((3,3),np.uint8)
kernelOp2 = np.ones((5,5),np.uint8)
kernelCl = np.ones((11,11),np.uint8)

#Variables
font = cv2.FONT_HERSHEY_SIMPLEX
persons = []
max_p_age = 5
pid = 1

while(cap.isOpened()):
##for image in camera.capture_continuous(rawCapture, format="bgr", use_video_port=True):
    #Lee una imagen de la fuente de video
    ret, frame = cap.read()
##    frame = image.array

    for i in persons:
        i.age_one() #age every person one frame
    #########################
    #   PRE-PROCESAMIENTO   #
    #########################

    #Aplica substraccion de fondo
    fgmask = fgbg.apply(frame)
    fgmask2 = fgbg.apply(frame)

    #Binariazcion para eliminar sombras (color gris)
    try:
        ret,imBin= cv2.threshold(fgmask,200,255,cv2.THRESH_BINARY)
        ret,imBin2 = cv2.threshold(fgmask2,200,255,cv2.THRESH_BINARY)
        #Opening (erode->dilate) para quitar ruido.
        mask = cv2.morphologyEx(imBin, cv2.MORPH_OPEN, kernelOp)
        mask2 = cv2.morphologyEx(imBin2, cv2.MORPH_OPEN, kernelOp)
        #Closing (dilate -> erode) para juntar regiones blancas.
        mask =  cv2.morphologyEx(mask , cv2.MORPH_CLOSE, kernelCl)
        mask2 = cv2.morphologyEx(mask2, cv2.MORPH_CLOSE, kernelCl)
    except:
        print('EOF')
        print 'UP:',cnt_up
        print 'DOWN:',cnt_down
        break
    #################
    #   CONTORNOS   #
    #################

    # RETR_EXTERNAL returns only extreme outer flags. All child contours are left behind.
    _, contours0, hierarchy =
cv2.findContours(mask2,cv2.RETR_EXTERNAL,cv2.CHAIN_APPROX_SIMPLE)
    for cnt in contours0:
        area = cv2.contourArea(cnt)
        if area > areaTH:
            #################
            #   TRACKING    #
            #################

            #Falta agregar condiciones para multipersonas, salidas y entradas de pantalla.

            M = cv2.moments(cnt)
            cx = int(M['m10']/M['m00'])
            cy = int(M['m01']/M['m00'])
            x,y,w,h = cv2.boundingRect(cnt)

            new = True
            if cy in range(up_limit,down_limit):
                for i in persons:
                    if abs(cx-i.getX()) <= w and abs(cy-i.getY()) <= h:
                        # el objeto esta cerca de uno que ya se detecto antes
                        new = False
                        i.updateCoords(cx,cy)   #actualiza coordenadas en el objeto and resets age
                        if i.going_UP(line_down,line_up) == True:
                            cnt_up += 1;
                            print "ID:",i.getId(),'crossed going up at',time.strftime("%c")
                        elif i.going_DOWN(line_down,line_up) == True:
                            cnt_down += 1;
                            print "ID:",i.getId(),'crossed going down at',time.strftime("%c")
                        break
                    if i.getState() == '1':
                        if i.getDir() == 'down' and i.getY() > down_limit:
                            i.setDone()
                        elif i.getDir() == 'up' and i.getY() < up_limit:
                            i.setDone()
                    if i.timedOut():
                        #sacar i de la lista persons
                        index = persons.index(i)
                        persons.pop(index)
                        del i     #liberar la memoria de i
                if new == True:
                    p = Person.MyPerson(pid,cx,cy, max_p_age)
                    persons.append(p)
                    pid += 1
            #################
            #   DIBUJOS     #
            #################
            cv2.circle(frame,(cx,cy), 5, (0,0,255), -1)
            img = cv2.rectangle(frame,(x,y),(x+w,y+h),(0,255,0),2)
            #cv2.drawContours(frame, cnt, -1, (0,255,0), 3)

    #END for cnt in contours0

    #########################
    # DIBUJAR TRAYECTORIAS  #
    #########################
    for i in persons:
##          if len(i.getTracks()) >= 2:
##              pts = np.array(i.getTracks(), np.int32)
##              pts = pts.reshape((-1,1,2))
##              frame = cv2.polylines(frame,[pts],False,i.getRGB())
##          if i.getId() == 9:
##              print str(i.getX()), ',', str(i.getY())
        cv2.putText(frame, str(i.getId()),(i.getX(),i.getY()),font,0.3,i.getRGB(),1,cv2.LINE_AA)

    #################
    #   IMAGANES    #
    #################
    str_up = 'UP: '+ str(cnt_up)
    str_down = 'DOWN: '+ str(cnt_down)
    frame = cv2.polylines(frame,[pts_L1],False,line_down_color,thickness=2)
    frame = cv2.polylines(frame,[pts_L2],False,line_up_color,thickness=2)
    frame = cv2.polylines(frame,[pts_L3],False,(255,255,255),thickness=1)
    frame = cv2.polylines(frame,[pts_L4],False,(255,255,255),thickness=1)
    cv2.putText(frame, str_up ,(10,40),font,0.5,(255,255,255),2,cv2.LINE_AA)
    cv2.putText(frame, str_up ,(10,40),font,0.5,(0,0,255),1,cv2.LINE_AA)
    cv2.putText(frame, str_down ,(10,90),font,0.5,(255,255,255),2,cv2.LINE_AA)
    cv2.putText(frame, str_down ,(10,90),font,0.5,(255,0,0),1,cv2.LINE_AA)

    cv2.imshow('Frame',frame)
    #cv2.imshow('Mask',mask)

    #preisonar ESC para salir
    k = cv2.waitKey(30) & 0xff
    if k == 27:
        break
#END while(cap.isOpened())

#################
#   LIMPIEZA    #
#################
cap.release()
cv2.destroyAllWindows()
```

—