

Finding Hans™

—

Door Flint, Honzik, Nick en Tim Z.

De opdracht

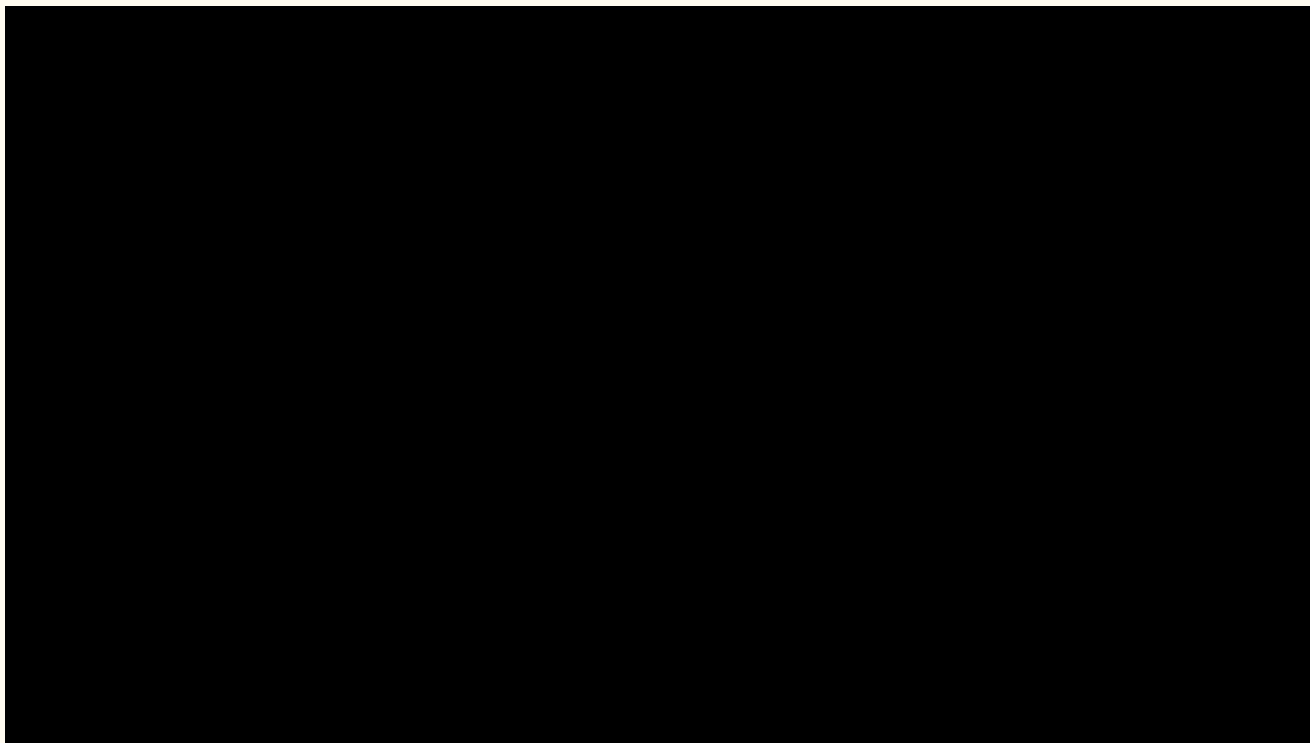
- Hans Verburg
- Swimm
- Zwembad
- Niet automatisch



Onze oplossing

- Het bad automatisch maken
- Positie bepalen
- Programma

Video



Materialen

Om ons idee te laten werken

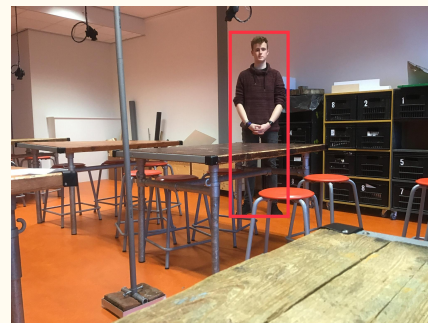
- Camera
- Computer (Jetson Nano)
- Programma



Programma

- Python
- OpenCV
- Verschil
- Grootste verschil zoeken
- Snelheid aanpassen

```
1 from imutils.video import VideoStream
2 from imutils.video import FPS
3 from datetime import datetime
4 import skimage
5 import argparse
6 import imutils
7 import time
8 import numpy
9 import cv2
10
11 print("Starting video stream.")
12 vs = VideoStream(0).start()
13 print("Done\n" + str(datetime.now()))
14 time.sleep(1.0)
15
16 print("Capturing background")
17 bg = vs.read()
18 print("Done\n" + str(datetime.now()))
19
20 while True:
21     if vs.is_not_ready():
22         frame = vs.read()
23         box = vs.read()
24
25         imageA = bg
26         imageB = frame
27
28         imageB = cv2.resize(imageB, (int(imageA.shape[1]), int(imageA.shape[0])), interpolation=cv2.INTER_AREA)
29         box = cv2.resize(box, (int(imageA.shape[1]), int(imageA.shape[0])), interpolation=cv2.INTER_AREA)
30
31         # convert the images to grayscale
32         grayA = cv2.cvtColor(imageA, cv2.COLOR_BGR2GRAY)
33         grayB = cv2.cvtColor(imageB, cv2.COLOR_BGR2GRAY)
34
35         # compute the Structural Similarity Index (SSIM) between the two
36         # images, ensuring that the difference image is returned
37         (score, diff) = skimage.metrics.structural_similarity(grayA, grayB, (7,7), cv2.blur(grayB, (7,7)), full=True)
38         diff = (diff * 255).astype("uint8")
39         print("SSIM: {}".format(score))
40
41         # threshold the difference image, followed by finding contours to
42         # obtain the regions of the two input images that differ
43         thresh = cv2.threshold(diff, 0, 255,
44                               cv2.THRESH_BINARY_INV | cv2.THRESH_OTSU)[1]
45         cnts = cv2.findContours(thresh.copy(), cv2.RETR_EXTERNAL,
46                               cv2.CHAIN_APPROX_SIMPLE)
47         cnts = imutils.grab_contours(cnts)
48
49         mw = 100
50         mh = 100
51         mx = 0
52         my = 0
53         hexists = False
54
55         # loop over the contours
56         for c in cnts:
57             # bounding box on both input images to represent where the two
58             # images differ
59             (x, y, w, h) = cv2.boundingRect(c)
60             cv2.rectangle(box, (x,y), ((x+w), (y+h)), (255,0,0), thickness=2)
61
62             if w > mw and h > mh:
63                 mw = int(w)
64                 mh = int(h)
65                 mx = int(x)
66                 my = int(y)
67                 hexists = True
68
69         if hexists == True and score < 0.92:
70             print("Target")
71             cv2.circle(frame, (int(mx + (mw/2)), int(my + (mh/2))), 2, (0, 0, 255), thickness=2)
72
73             print("PosX: " + str(int((mx + (mw/2))/frame.shape[1]*100)))
74
75             cv2.imshow("Frame", frame)
76             cv2.imshow("Diff", diff)
77             cv2.imshow("Box", box)
78
79             key = cv2.waitKey(1) & 0xFF
80             if key == ord('s'):
81                 initBB = cv2.selectROI("Frame", frame, fromContour=False,
82                                       showCrosshair=True)
83                 fps = FPS().start()
84             elif key == ord('q'):
85                 break
86
87 vs.stop()
88 cv2.destroyAllWindows()
```



Einde :)