Border Surviellance Bot

Group V

Generated by Doxygen 1.7.4

Sun Apr 17 2011 00:28:56

Contents

1 Namespace Index **1**

1.1 Namespace List 1

2 Namespace Documentation **3**

2.1 erts Namespace Reference 3

2.1.1 Detailed Description 4

2.1.2 Function Documentation 4

2.1.2.1 detect 4

2.1.3 Variable Documentation 4

2.1.3.1 cascade 4

2.1.3.2 faceBeep 4

2.1.3.3 faceNotDetected 4

2.1.3.4 faces 5

2.1.3.5 falseDetection 5

2.1.3.6 firstFaceDetected 5

2.1.3.7 grayscale 5

2.1.3.8 image_size 5

2.1.3.9 multipleFaces 6

2.1.3.10 ser 6

2.1.3.11 storage 6

2.2 opencv Namespace Reference 6

2.2.1 Detailed Description 6

2.3 serial Namespace Reference 7

2.3.1 Detailed Description 7

2.4 sys Namespace Reference 7

2.4.1 Detailed Description 7

2.5 time Namespace Reference 7
2.5.1 Detailed Description 7

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all documented namespaces with brief descriptions:

erts 3
opencv (Importing opencv for face detection) 6
serial (Importing pyserial for serial communication) 7
sys (Importing system for handling signals for exit) 7
time (For sleep function) 7

Chapter 2

Namespace Documentation

2.1 erts Namespace Reference

Functions

- def detect

  The function to get the frame.

Variables

- tuple ser

  configure the serial connections (the parameters differs on the device you are connecting to)
- tuple image_size = cv.GetSize(image)
- tuple grayscale = cv.CreateImage(image_size, 8, 1)
- tuple storage = cv.CreateMemStorage(0)
- tuple cascade = cv.Load('haarcascade_frontalface_alt.xml')
- tuple faces = cv.HaarDetectObjects(grayscale, cascade, storage, 1.2, 2, cv.CV_-HAAR_DO_CANNY_PRUNING)
- int **device** = 1
- tuple **capture** = cv.CaptureFromCAM(device)
- int faceBeep = 0
- int firstFaceDetected = 0
- int falseDetection = 0
- int faceNotDetected = 0
- int multipleFaces = 0
- int **multipleFacesInit** = 0
- int **forward** = 0
- int **left** = 0
- int **right** = 0
- tuple **frame** = cv.QueryFrame(capture)

- **x** = left-right
- tuple **k** = cv.WaitKey(10)

2.1.1 Detailed Description

```
*********************************************************************************************
               Platform: Python 2.x and 2.x.x
               Title: Border Surveillance Bot
               Author:
                       1.Harish Tummalacherla
                       2.Sameer Mohammed
*********************************************************************************************
```

2.1.2 Function Documentation

2.1.2.1 def erts::detect (*image*)

The function to get the frame.

Parameters

image	the image input

Definition at line 102 of file erts.py.

2.1.3 Variable Documentation

2.1.3.1 erts::cascade = cv.Load('haarcascade_frontalface_alt.xml')

Parameters

haarcascade	for face detection detect objects loading the Haar Cascade

Definition at line 130 of file erts.py.

2.1.3.2 int erts::faceBeep = 0

Parameters

Used	for beeping control

Definition at line 173 of file erts.py.

2.1.3.3 int erts::faceNotDetected = 0

Parameters

used	for checking if no face is found

Definition at line 179 of file erts.py.

2.1.3.4 tuple erts::faces = cv.HaarDetectObjects(grayscale, cascade, storage, 1.2, 2, cv.CV_HAAR_DO_CANNY_PRUNING)

Parameters

cascade	haarcascade for face detection detecting the faces in the image These parameters are tweaked to RGB video captures, please refer to http://opencv.willowgarage.com/documentation/python/objdetect_-cascade_classification.html for tweaking your parameters.

Definition at line 136 of file erts.py.

2.1.3.5 int erts::falseDetection = 0

Parameters

For	false detection

Definition at line 177 of file erts.py.

2.1.3.6 int erts::firstFaceDetected = 0

Parameters

Used	for face detection

Definition at line 175 of file erts.py.

2.1.3.7 erts::grayscale = cv.CreateImage(image_size, 8, 1)

Parameters

image_size	create grayscale version

Definition at line 112 of file erts.py.

2.1.3.8 erts::image_size = cv.GetSize(image)

Parameters

image	image input Getting size of the image for handling generic image resolution, i.e. handling webcam with arbitrary resolution

Definition at line 107 of file erts.py.

2.1.3.9 int erts::multipleFaces = 0

Parameters

for	indicating that multiple faces

Definition at line 181 of file erts.py.

2.1.3.10 tuple erts::ser

Initial value:

```
1 serial.Serial(
2         port='/dev/ttyUSB0',
3         baudrate=9600,
4         parity=serial.PARITY_NONE,
5         stopbits=serial.STOPBITS_ONE,
6         bytesize=serial.EIGHTBITS
7 )
```

configure the serial connections (the parameters differs on the device you are connecting to)

Parameters

port	The port where the serial communication usb is present.
baudrate	Set the baud rate to 9600
parity	Parity Bit Disabled
stopbits	Use one stop bit
bytesize	Number of Data Bits = 8

Definition at line 89 of file erts.py.

2.1.3.11 erts::storage = cv.CreateMemStorage(0)

Parameters

0	create required storage for face detection

Definition at line 119 of file erts.py.

2.2 opencv Namespace Reference

importing opencv for face detection

2.2.1 Detailed Description

importing opencv for face detection

2.3 serial Namespace Reference

importing pyserial for serial communication

2.3.1 Detailed Description

importing pyserial for serial communication

2.4 sys Namespace Reference

importing system for handling signals for exit

2.4.1 Detailed Description

importing system for handling signals for exit

2.5 time Namespace Reference

for sleep function

2.5.1 Detailed Description

for sleep function

Index

cascade
    erts, 4

detect
    erts, 4

erts, 3
    cascade, 4
    detect, 4
    faceBeep, 4
    faceNotDetected, 4
    faces, 5
    falseDetection, 5
    firstFaceDetected, 5
    grayscale, 5
    image_size, 5
    multipleFaces, 5
    ser, 6
    storage, 6

faceBeep
    erts, 4
faceNotDetected
    erts, 4
faces
    erts, 5
falseDetection
    erts, 5
firstFaceDetected
    erts, 5

grayscale
    erts, 5

image_size
    erts, 5

multipleFaces
    erts, 5

opencv, 6

ser
    erts, 6
serial, 7
storage
    erts, 6
sys, 7

time, 7