Eye-Bit Project Documentation

Generated by Doxygen 1.9.1

Глава 1

Интерфейс для создания бита

1.0.0.1 Описание:

Интерфейс включает в себя камеру для отработки трека взгляда в реальном времени и приложение. Пользователь запускает приложение и на экране появляется панель управления. С помощью взгляда осуществляется задание желаемых параметров, после чего сигнал преобразуется в аудиофайл. Пользователь может скачать полученный файл.

1.0.0.2 Технологии:

- ROS
- Python
- OpenCV

1.0.0.3 Цель проекта:

Реализация интерфейса с функцией Gaze tracking, исследование и применение методов и технологий

1.0.0.4 Задачи проекта:

- Настройка камеры для трека взгляда
- Дизайн приложения
- Разработка алгоритма генерации бита на основе полученных данных
- Разработка приложения

1.0.0.5 Запуск:

Для запуска приложения следует выполнить несколько шагов:

1) Клонировать проект в локальный репозиторий

2) Запустить файл requirements (для установки необходимых библиотек)

3) Переместить содержимое папки src, (папки gaze_tracker и spoproject) в папку, где располагаются ваши ROS пакеты (это может быть catkin_work_space/src)

4) в папке ~/catkin_work_space/ выполнить команду catkin_make

5) Перейти в папку ~/catkin_work_space/src/spoproject/scripts/

6) Запустить интерфейс из этой папки командой python Interface.py

Глава 2

Namespace Index

2.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Глава 3

Hierarchical Index

3.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Глава 4

Class Index

4.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Глава 5

File Index

5.1 File List

Here is a list of all files with brief descriptions:

Глава 6

Namespace Documentation

6.1 Eye-Bit Project Namespace Reference

Namespaces

- Blinking_count
- calibration
- colibrationWithCV
- eye
- gaze_tracking
- Interface
- listener
- pupil
- start_ros
- Trecker_with6acord

6.2 Eye-Bit Project.Blinking_count Namespace Reference

Functions

- def click (event, x, y, flags, param)
- def blink ()

  The function is intended for counting the number of blinking provides.

Variables

- bool start = False

6.2.1 Function Documentation

6.2.1.1 blink()

def Eye-Bit Project.Blinking_count.blink ( )

The function is intended for counting the number of blinking provides.

Returns

:approximate number of blinking provides.

6.2.1.2 click()

def Eye-Bit Project.Blinking_count.click (
event,
x,
y,
flags,
param)

6.2.2 Variable Documentation

6.2.2.1 start

bool Eye-Bit Project.Blinking_count.start = False

6.3 Eye-Bit Project.calibration Namespace Reference

Classes

- class Calibration

  This class calibrates the pupil detection algorithm by finding the best binarization threshold value for the person and the webcam.

6.4 Eye-Bit Project.colibrationWithCV Namespace Reference

Functions

- def click (event, x, y, flags, param)
- def getInhabitationOfPupilsWhenLookingOnScreen (s_w, s_h)

  This function is intended for the calibration.

Variables

- bool watchsOnPoint = False

6.4.1 Function Documentation

6.4.1.1 click()

```
def Eye-Bit Project.colibrationWithCV.click (
            event,
            x,
            y,
            flags,
            param )
```

6.4.1.2 getInhabitationOfPupilsWhenLookingOnScreen()

```
def Eye-Bit Project.colibrationWithCV.getInhabitationOfPupilsWhenLookingOnScreen (
            s_w,
            s_h )
```

This function is intended for the calibration.

It provides information about position of pupils on a frame when looking on the define point of the screen. This information is then used for avaluating the size of the domain in which pupils can be and the coordinats of the pupils when looking on the center of the screen.

Parameters

s_w	width of the screen;
s_h	height of the screen

Returns

:List of the coordinates of pupils corresponding to the following cases: [looking on the center, looking on the top, looking on the right, on the bottom, on the left]

6.4.2 Variable Documentation

6.4.2.1 watchsOnPoint

```
bool Eye-Bit Project.colibrationWithCV.watchsOnPoint = False
```

6.5 Eye-Bit Project.eye Namespace Reference

Classes

- class Eye

  This class creates a new frame to isolate the eye and initiates the pupil detection.

6.6 Eye-Bit Project.gaze_tracking Namespace Reference

Classes

- class GazeTracking

  This class tracks the user's gaze.

Functions

- def pupils_located (self)

  Check that the pupils have been located.
- def refresh (self, frame)

  Refreshes the frame and analyzes it.
- def pupil_left_coords (self)

  Returns the coordinates of the left pupil.
- def pupil_right_coords (self)

  Returns the coordinates of the right pupil.
- def horizontal_ratio (self)

  Returns a number between 0.0 and 1.0 that indicates the horizontal direction of the gaze.
- def vertical_ratio (self)

  Returns a number between 0.0 and 1.0 that indicates the vertical direction of the gaze.
- def is_right (self)

  Returns true if the user is looking to the right.
- def is_left (self)

  Returns true if the user is looking to the left.
- def is_center (self)

  Returns true if the user is looking to the center.
- def is_blinking (self)

  Returns true if the user closes his eyes.
- def annotated_frame (self)

  Returns the main frame with pupils highlighted.
- def landmarks_to_np (landmarks, dtype="int")

Variables

- eye_left
- eye_right
- frame

6.6.1 Function Documentation

6.6.1.1 annotated_frame()

def Eye-Bit Project.gaze_tracking.annotated_frame (
self)

Returns the main frame with pupils highlighted.

6.6.1.2 horizontal_ratio()

def Eye-Bit Project.gaze_tracking.horizontal_ratio (
self)

Returns a number between 0.0 and 1.0 that indicates the horizontal direction of the gaze.

The extreme right is 0.0, the center is 0.5 and the extreme left is 1.0

6.6.1.3 is_blinking()

def Eye-Bit Project.gaze_tracking.is_blinking (
self)

Returns true if the user closes his eyes.

6.6.1.4 is_center()

def Eye-Bit Project.gaze_tracking.is_center (
self)

Returns true if the user is looking to the center.

6.6.1.5 is_left()

def Eye-Bit Project.gaze_tracking.is_left (
self)

Returns true if the user is looking to the left.

6.6.1.6 is_right()

```
def Eye-Bit Project.gaze_tracking.is_right (
        self )
```

Returns true if the user is looking to the right.

6.6.1.7 landmarks_to_np()

```
def Eye-Bit Project.gaze_tracking.landmarks_to_np (
        landmarks,
        dtype = "int" )
```

6.6.1.8 pupil_left_coords()

```
def Eye-Bit Project.gaze_tracking.pupil_left_coords (
        self )
```

Returns the coordinates of the left pupil.

6.6.1.9 pupil_right_coords()

```
def Eye-Bit Project.gaze_tracking.pupil_right_coords (
        self )
```

Returns the coordinates of the right pupil.

6.6.1.10 pupils_located()

```
def Eye-Bit Project.gaze_tracking.pupils_located (
        self )
```

Check that the pupils have been located.

6.6.1.11 refresh()

```
def Eye-Bit Project.gaze_tracking.refresh (
        self,
        frame )
```

Refreshes the frame and analyzes it.

Parameters

:frame	(numpy.ndarray): The frame to analyze

6.6.1.12 vertical_ratio()

def Eye-Bit Project.gaze_tracking.vertical_ratio (
self)

Returns a number between 0.0 and 1.0 that indicates the vertical direction of the gaze.

The extreme top is 0.0, the center is 0.5 and the extreme bottom is 1.0

6.6.2 Variable Documentation

6.6.2.1 eye_left

Eye-Bit Project.gaze_tracking.eye_left

6.6.2.2 eye_right

Eye-Bit Project.gaze_tracking.eye_right

6.6.2.3 frame

Eye-Bit Project.gaze_tracking.frame

6.7 Eye-Bit Project.Interface Namespace Reference

Classes

- class Interface

  This class is creating GUI for an interaction with a gaze tracker.

Variables

- app = QApplication(sys.argv)
- w = Interface()

6.7.1 Variable Documentation

6.7.1.1 app

Eye-Bit Project.Interface.app = QApplication(sys.argv)

6.7.1.2 w

Eye-Bit Project.Interface.w = Interface()

6.8 Eye-Bit Project.listener Namespace Reference

Functions

- def play_chord (n)

  The function plays the chord corresponding to the message received.
- def callback (data)

  The callback function of the subscriber.
- def listener ()

  This function sets up the subsciber rosnode "listner" and subscibes on the topik "chatter".

Variables

- list list_audio = []

6.8.1 Function Documentation

6.8.1.1 callback()

def Eye-Bit Project.listener.callback (
data)

The callback function of the subscriber.

6.8.1.2 listener()

def Eye-Bit Project.listener.listener ( )

This function sets up the subsciber rosnode "listner" and subscibes on the topik "chatter".

6.8.1.3 play_chord()

def Eye-Bit Project.listener.play_chord (
n)

The function plays the chord corresponding to the message received.

6.8.2 Variable Documentation

6.8.2.1 list_audio

list Eye-Bit Project.listener.list_audio = []

6.9 Eye-Bit Project.pupil Namespace Reference

Classes

- class Pupil

  This class detects the iris of an eye and estimates the position of the pupil.

6.10 Eye-Bit Project.start_ros Namespace Reference

Functions

- def run (cmd, stdout, stderr)
- def start_process (cmd)
- def start ()

  This function starts roscore, the subscriber node (listener.py) and the publisher node (Trecker_with6acord.py) in three parallel subprocesses.

6.10.1 Function Documentation

6.10.1.1 run()

```
def Eye-Bit Project.start_ros.run (
                cmd,
                stdout,
                stderr )
```

6.10.1.2 start()

```
def Eye-Bit Project.start_ros.start ( )
```

This function starts roscore, the subscriber node (listener.py) and the publisher node (Trecker_with6acord.py) in three parallel subprocesses.

6.10.1.3 start_process()

```
def Eye-Bit Project.start_ros.start_process (
                cmd )
```

6.11 Eye-Bit Project.Trecker_with6acord Namespace Reference

Functions

- def get_screen_size (display)

  This function provides the width and the height of the screen.
- def domain (x, y, s_w, s_h)

  The function estimates the rectangle that includes the given point.
- def drowLine (cord, orient, size)

  The function provides the coordinates of the line.
- def talker ()

  This function tracks the user's gaze.

6.11.1 Function Documentation

6.11.1.1 domain()

```
def Eye-Bit Project.Trecker_with6acord.domain (
                x,
                y,
                s_w,
                s_h )
```

The function estimates the rectangle that includes the given point.

Parameters

	x and y coordinates of the point and width and height of the screen

Returns

: integer 0 <= n <= 5

6.11.1.2 drowLine()

def Eye-Bit Project.Trecker_with6acord.drowLine (
cord,
orient,
size)

The function provides the coordinates of the line.

Parameters

	starting x or y coordinate of the line, orientation (string. "vert" or "hor") and length of the line

Returns

: list of two points (start and end of the line)

6.11.1.3 get_screen_size()

def Eye-Bit Project.Trecker_with6acord.get_screen_size (
display)

This function provides the width and the height of the screen.

Parameters

	object Gdk.Display.get_default()

Returns

: touple (width, height)

6.11.1.4 talker()

def Eye-Bit Project.Trecker_with6acord.talker ( )

This function tracks the user's gaze.

First it runs the colibration. Using the information provided by colibration function it estimates the place on the screen where the user is looking. Whith frequence of 10 Hz the function publishes on the topic "chatter" the string representation of integer 0 <= n <= 5 which coresponds to the rectangle in which the user is looking.

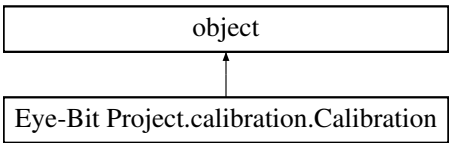
Глава 7

Class Documentation

7.1 Eye-Bit Project.calibration.Calibration Class Reference

This class calibrates the pupil detection algorithm by finding the best binarization threshold value for the person and the webcam.

Inheritance diagram for Eye-Bit Project.calibration.Calibration:



Public Member Functions

- def __init__ (self)
- def is_complete (self)

  Returns true if the calibration is completed.
- def threshold (self, side)

  Returns the threshold value for the given eye.
- def evaluate (self, eye_frame, side)

  Improves calibration by taking into consideration the given image.

Static Public Member Functions

- def iris_size (frame)

  Returns the percentage of space that the iris takes up on the surface of the eye.
- def find_best_threshold (eye_frame)

  Calculates the optimal threshold to binarize the frame for the given eye.

Public Attributes

- nb_frames
- thresholds_left
- thresholds_right

7.1.1 Detailed Description

This class calibrates the pupil detection algorithm by finding the best binarization threshold value for the person and the webcam.

7.1.2 Constructor & Destructor Documentation

7.1.2.1 __init__()

def Eye-Bit Project.calibration.Calibration.__init__ (
self)

7.1.3 Member Function Documentation

7.1.3.1 evaluate()

def Eye-Bit Project.calibration.Calibration.evaluate (
self,
eye_frame,
side)

Improves calibration by taking into consideration the given image.

Parameters

eye_frame	(numpy.ndarray): Frame of the eye
side	Indicates whether it's the left eye (0) or the right eye (1)

7.1.3.2 find_best_threshold()

def Eye-Bit Project.calibration.Calibration.find_best_threshold (
eye_frame) [static]

Calculates the optimal threshold to binarize the frame for the given eye.

Parameters

eye_frame	(numpy.ndarray): Frame of the eye to be analyzed

7.1.3.3 iris_size()

def Eye-Bit Project.calibration.Calibration.iris_size (
frame) [static]

Returns the percentage of space that the iris takes up on the surface of the eye.

Parameters

frame	(numpy.ndarray): Binarized iris frame

7.1.3.4 is_complete()

def Eye-Bit Project.calibration.Calibration.is_complete (
self)

Returns true if the calibration is completed.

7.1.3.5 threshold()

def Eye-Bit Project.calibration.Calibration.threshold (
self,
side)

Returns the threshold value for the given eye.

Parameters

side	Indicates whether it's the left eye (0) or the right eye (1)

7.1.4 Member Data Documentation

7.1.4.1 nb_frames

Eye-Bit Project.calibration.Calibration.nb_frames

7.1.4.2 thresholds_left

Eye-Bit Project.calibration.Calibration.thresholds_left

7.1.4.3 thresholds_right

Eye-Bit Project.calibration.Calibration.thresholds_right

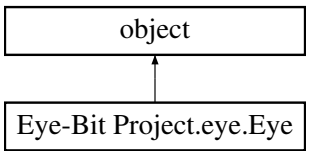
The documentation for this class was generated from the following file:

- calibration.py

7.2 Eye-Bit Project.eye.Eye Class Reference

This class creates a new frame to isolate the eye and initiates the pupil detection.

Inheritance diagram for Eye-Bit Project.eye.Eye:



Public Member Functions

- def __init__ (self, original_frame, landmarks, side, calibration, E_cords)

Public Attributes

- frame
- origin
- center
- pupil
- cods
- blinking

Static Public Attributes

- list LEFT_EYE_POINTS = [36, 37, 38, 39, 40, 41]
- list RIGHT_EYE_POINTS = [42, 43, 44, 45, 46, 47]

7.2.1 Detailed Description

This class creates a new frame to isolate the eye and initiates the pupil detection.

7.2.2 Constructor & Destructor Documentation

7.2.2.1 __init__()

def Eye-Bit Project.eye.Eye.__init__ (
self,
original_frame,
landmarks,
side,
calibration,
E_cords)

7.2.3 Member Data Documentation

7.2.3.1 blinking

Eye-Bit Project.eye.Eye.blinking

7.2.3.2 center

Eye-Bit Project.eye.Eye.center

7.2.3.3 cods

Eye-Bit Project.eye.Eye.cods

7.2.3.4 frame

Eye-Bit Project.eye.Eye.frame

7.2.3.5 LEFT_EYE_POINTS

list Eye-Bit Project.eye.Eye.LEFT_EYE_POINTS = [36, 37, 38, 39, 40, 41] [static]

7.2.3.6 origin

Eye-Bit Project.eye.Eye.origin

7.2.3.7 pupil

Eye-Bit Project.eye.Eye.pupil

7.2.3.8 RIGHT_EYE_POINTS

list Eye-Bit Project.eye.Eye.RIGHT_EYE_POINTS = [42, 43, 44, 45, 46, 47] [static]

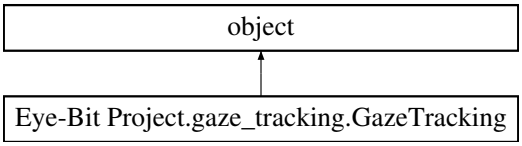
The documentation for this class was generated from the following file:

- eye.py

7.3 Eye-Bit Project.gaze_tracking.GazeTracking Class Reference

This class tracks the user's gaze.

Inheritance diagram for Eye-Bit Project.gaze_tracking.GazeTracking:



Public Member Functions

- def __init__ (self)
- def pupils_located (self)
  Check that the pupils have been located.
- def refresh (self, frame)
  Refreshes the frame and analyzes it.
- def pupil_left_coords (self)
  Returns the coordinates of the left pupil.
- def pupil_right_coords (self)
  Returns the coordinates of the right pupil.
- def horizontal_ratio (self)
  Returns a number between 0.0 and 1.0 that indicates the horizontal direction of the gaze.
- def vertical_ratio (self)
  Returns a number between 0.0 and 1.0 that indicates the vertical direction of the gaze.
- def is_right (self)
  Returns true if the user is looking to the right.
- def is_left (self)
  Returns true if the user is looking to the left.
- def is_center (self)
  Returns true if the user is looking to the center.
- def is_blinking (self)
  Returns true if the user closes his eyes.
- def annotated_frame (self)
  Returns the main frame with pupils highlighted.

Public Attributes

- frame
- eye_left
- eye_right
- calibration

7.3.1 Detailed Description

This class tracks the user's gaze.

It provides useful information like the position of the eyes and pupils and allows to know if the eyes are open or closed

7.3.2 Constructor & Destructor Documentation

7.3.2.1 __init__()

def Eye-Bit Project.gaze_tracking.GazeTracking.__init__ (
self)

7.3.3 Member Function Documentation

7.3.3.1 annotated_frame()

def Eye-Bit Project.gaze_tracking.GazeTracking.annotated_frame (
self)

Returns the main frame with pupils highlighted.

7.3.3.2 horizontal_ratio()

def Eye-Bit Project.gaze_tracking.GazeTracking.horizontal_ratio (
self)

Returns a number between 0.0 and 1.0 that indicates the horizontal direction of the gaze.

The extreme right is 0.0, the center is 0.5 and the extreme left is 1.0

7.3.3.3 is_blinking()

def Eye-Bit Project.gaze_tracking.GazeTracking.is_blinking (
self)

Returns true if the user closes his eyes.

7.3.3.4 is_center()

def Eye-Bit Project.gaze_tracking.GazeTracking.is_center (
self)

Returns true if the user is looking to the center.

7.3.3.5 is_left()

def Eye-Bit Project.gaze_tracking.GazeTracking.is_left (
self)

Returns true if the user is looking to the left.

7.3.3.6 is_right()

def Eye-Bit Project.gaze_tracking.GazeTracking.is_right (
self)

Returns true if the user is looking to the right.

7.3.3.7 pupil_left_coords()

def Eye-Bit Project.gaze_tracking.GazeTracking.pupil_left_coords (
self)

Returns the coordinates of the left pupil.

7.3.3.8 pupil_right_coords()

def Eye-Bit Project.gaze_tracking.GazeTracking.pupil_right_coords (
self)

Returns the coordinates of the right pupil.

7.3.3.9 pupils_located()

def Eye-Bit Project.gaze_tracking.GazeTracking.pupils_located (
self)

Check that the pupils have been located.

7.3.3.10 refresh()

def Eye-Bit Project.gaze_tracking.GazeTracking.refresh (
self,
frame)

Refreshes the frame and analyzes it.

Parameters

:frame	(numpy.ndarray): The frame to analyze

7.3.3.11 vertical_ratio()

def Eye-Bit Project.gaze_tracking.GazeTracking.vertical_ratio (
self)

Returns a number between 0.0 and 1.0 that indicates the vertical direction of the gaze.

The extreme top is 0.0, the center is 0.5 and the extreme bottom is 1.0

7.3.4 Member Data Documentation

7.3.4.1 calibration

Eye-Bit Project.gaze_tracking.GazeTracking.calibration

7.3.4.2 eye_left

Eye-Bit Project.gaze_tracking.GazeTracking.eye_left

7.3.4.3 eye_right

Eye-Bit Project.gaze_tracking.GazeTracking.eye_right

7.3.4.4 frame

Eye-Bit Project.gaze_tracking.GazeTracking.frame

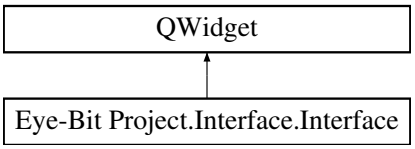
The documentation for this class was generated from the following file:

- gaze_tracking.py

7.4 Eye-Bit Project.Interface.Interface Class Reference

This class is creating GUI for an interaction with a gaze tracker.

Inheritance diagram for Eye-Bit Project.Interface.Interface:



Public Member Functions

- def __init__ (self)
- def center (self)

  Set the main window on the center of screen.
- def harm (self)

  Launch window for calibration and after that provide special panel with harmonic canvas for making music.
- def set_drums (self)

  Overlay a drum part on the music file.
- def count_blinks (self)

  Count user's blinks and set up the frame rate for music file.
- def play (self)

  Play the finished music file.

Public Attributes

- play_button
- freq
- const
- beat_button
- buttons_hbox
- ins_label
- drums_cb
- hihat_cb
- bass_cb
- cbs_hbox
- harm_button
- main_vbox

7.4.1 Detailed Description

This class is creating GUI for an interaction with a gaze tracker.

It provides main menu with several options for making custom music.

7.4.2 Constructor & Destructor Documentation

7.4.2.1 __init__()

def Eye-Bit Project.Interface.Interface.__init__ (
self)

7.4.3 Member Function Documentation

7.4.3.1 center()

def Eye-Bit Project.Interface.Interface.center (
self)

Set the main window on the center of screen.

7.4.3.2 count_blinks()

def Eye-Bit Project.Interface.Interface.count_blinks (
self)

Count user's blinks and set up the frame rate for music file.

7.4.3.3 harm()

def Eye-Bit Project.Interface.Interface.harm (
self)

Launch window for calibration and after that provide special panel with harmonic canvas for making music.

7.4.3.4 play()

def Eye-Bit Project.Interface.Interface.play (
self)

Play the finished music file.

It depends on selected options, which file will play.

7.4.3.5 set_drums()

def Eye-Bit Project.Interface.Interface.set_drums (
self)

Overlay a drum part on the music file.

Write result in the result_with_drums.mp3.

7.4.4 Member Data Documentation

7.4.4.1 bass_cb

Eye-Bit Project.Interface.Interface.bass_cb

7.4.4.2 beat_button

Eye-Bit Project.Interface.Interface.beat_button

7.4.4.3 buttons_hbox

Eye-Bit Project.Interface.Interface.buttons_hbox

7.4.4.4 cbs_hbox

Eye-Bit Project.Interface.Interface.cbs_hbox

7.4.4.5 const

Eye-Bit Project.Interface.Interface.const

7.4.4.6 drums_cb

Eye-Bit Project.Interface.Interface.drums_cb

7.4.4.7 freq

Eye-Bit Project.Interface.Interface.freq

7.4.4.8 harm_button

Eye-Bit Project.Interface.Interface.harm_button

7.4.4.9 hihat_cb

Eye-Bit Project.Interface.Interface.hihat_cb

7.4.4.10 ins_label

Eye-Bit Project.Interface.Interface.ins_label

7.4.4.11 main_vbox

Eye-Bit Project.Interface.Interface.main_vbox

7.4.4.12 play_button

Eye-Bit Project.Interface.Interface.play_button

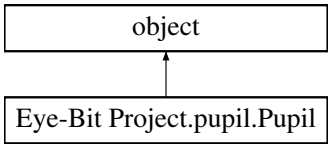
The documentation for this class was generated from the following file:

- Interface.py

7.5 Eye-Bit Project.pupil.Pupil Class Reference

This class detects the iris of an eye and estimates the position of the pupil.

Inheritance diagram for Eye-Bit Project.pupil.Pupil:



Public Member Functions

- def __init__ (self, eye_frame, threshold)
- def detect_iris (self, eye_frame)

  Detects the iris and estimates the position of the iris by calculating the centroid.

Static Public Member Functions

- def image_processing (eye_frame, threshold)

  Performs operations on the eye frame to isolate the iris.

Public Attributes

- iris_frame
- threshold
- x
- y

7.5.1 Detailed Description

This class detects the iris of an eye and estimates the position of the pupil.

7.5.2 Constructor & Destructor Documentation

7.5.2.1 __init__()

```
def Eye-Bit Project.pupil.Pupil.__init__ (
                self,
                eye_frame,
                threshold )
```

7.5.3 Member Function Documentation

7.5.3.1 detect_iris()

def Eye-Bit Project.pupil.Pupil.detect_iris (
self,
eye_frame)

Detects the iris and estimates the position of the iris by calculating the centroid.

Returns

:eye_frame (numpy.ndarray): Frame containing an eye and nothing else

7.5.3.2 image_processing()

def Eye-Bit Project.pupil.Pupil.image_processing (
eye_frame,
threshold) [static]

Performs operations on the eye frame to isolate the iris.

Parameters

eye_frame	(numpy.ndarray): Frame containing an eye and nothing else
threshold	(int): Threshold value used to binarize the eye frame

Returns

:A frame with a single element representing the iris

7.5.4 Member Data Documentation

7.5.4.1 iris_frame

Eye-Bit Project.pupil.Pupil.iris_frame

7.5.4.2 threshold

Eye-Bit Project.pupil.Pupil.threshold

7.5.4.3 x

Eye-Bit Project.pupil.Pupil.x

7.5.4.4 y

Eye-Bit Project.pupil.Pupil.y

The documentation for this class was generated from the following file:

- pupil.py

Глава 8

File Documentation

8.1 __init__.py File Reference

Namespaces

- Eye-Bit Project

8.2 Blinking_count.py File Reference

Namespaces

- Eye-Bit Project.Blinking_count

Functions

- def Eye-Bit Project.Blinking_count.click (event, x, y, flags, param)
- def Eye-Bit Project.Blinking_count.blink ()

  The function is intended for counting the number of blinking provides.

Variables

- bool Eye-Bit Project.Blinking_count.start = False

8.3 calibration.py File Reference

Classes

- class Eye-Bit Project.calibration.Calibration

  This class calibrates the pupil detection algorithm by finding the best binarization threshold value for the person and the webcam.

Namespaces

- Eye-Bit Project.calibration

8.4 colibrationWithCV.py File Reference

Namespaces

- Eye-Bit Project.colibrationWithCV

Functions

- def Eye-Bit Project.colibrationWithCV.click (event, x, y, flags, param)
- def Eye-Bit Project.colibrationWithCV.getInhabitationOfPupilsWhenLookingOnScreen (s_w, s↩ _h)

  This function is intended for the calibration.

Variables

- bool Eye-Bit Project.colibrationWithCV.watchsOnPoint = False

8.5 eye.py File Reference

Classes

- class Eye-Bit Project.eye.Eye

  This class creates a new frame to isolate the eye and initiates the pupil detection.

Namespaces

- Eye-Bit Project.eye

8.6 gaze_tracking.py File Reference

Classes

- class Eye-Bit Project.gaze_tracking.GazeTracking

  This class tracks the user's gaze.

Namespaces

- Eye-Bit Project.gaze_tracking

Functions

- def Eye-Bit Project.gaze_tracking.pupils_located (self)

  Check that the pupils have been located.
- def Eye-Bit Project.gaze_tracking.refresh (self, frame)

  Refreshes the frame and analyzes it.
- def Eye-Bit Project.gaze_tracking.pupil_left_coords (self)

  Returns the coordinates of the left pupil.
- def Eye-Bit Project.gaze_tracking.pupil_right_coords (self)

  Returns the coordinates of the right pupil.
- def Eye-Bit Project.gaze_tracking.horizontal_ratio (self)

  Returns a number between 0.0 and 1.0 that indicates the horizontal direction of the gaze.
- def Eye-Bit Project.gaze_tracking.vertical_ratio (self)

  Returns a number between 0.0 and 1.0 that indicates the vertical direction of the gaze.
- def Eye-Bit Project.gaze_tracking.is_right (self)

  Returns true if the user is looking to the right.
- def Eye-Bit Project.gaze_tracking.is_left (self)

  Returns true if the user is looking to the left.
- def Eye-Bit Project.gaze_tracking.is_center (self)

  Returns true if the user is looking to the center.
- def Eye-Bit Project.gaze_tracking.is_blinking (self)

  Returns true if the user closes his eyes.
- def Eye-Bit Project.gaze_tracking.annotated_frame (self)

  Returns the main frame with pupils highlighted.
- def Eye-Bit Project.gaze_tracking.landmarks_to_np (landmarks, dtype="int")

Variables

- Eye-Bit Project.gaze_tracking.eye_left
- Eye-Bit Project.gaze_tracking.eye_right
- Eye-Bit Project.gaze_tracking.frame

8.7 Interface.py File Reference

Classes

- class Eye-Bit Project.Interface.Interface

  This class is creating GUI for an interaction with a gaze tracker.

Namespaces

- Eye-Bit Project.Interface

Variables

- Eye-Bit Project.Interface.app = QApplication(sys.argv)
- Eye-Bit Project.Interface.w = Interface()

8.8 listener.py File Reference

Namespaces

- Eye-Bit Project.listener

Functions

- def Eye-Bit Project.listener.play_chord (n)

  The function plays the chord corresponding to the message received.
- def Eye-Bit Project.listener.callback (data)

  The callback function of the subscriber.
- def Eye-Bit Project.listener.listener ()

  This function sets up the subsciber rosnode "listner" and subscibes on the topik "chatter".

Variables

- list Eye-Bit Project.listener.list_audio = []

8.9 pupil.py File Reference

Classes

- class Eye-Bit Project.pupil.Pupil

  This class detects the iris of an eye and estimates the position of the pupil.

Namespaces

- Eye-Bit Project.pupil

8.10 README.md File Reference

8.11 start_ros.py File Reference

Namespaces

- Eye-Bit Project.start_ros

Functions

- def Eye-Bit Project.start_ros.run (cmd, stdout, stderr)
- def Eye-Bit Project.start_ros.start_process (cmd)
- def Eye-Bit Project.start_ros.start ()

  This function starts roscore, the subscriber node (listener.py) and the publisher node (Trecker_with6acord.py) in three parallel subprocesses.

8.12 Trecker_with6acord.py File Reference

Namespaces

- Eye-Bit Project.Trecker_with6acord

Functions

- def Eye-Bit Project.Trecker_with6acord.get_screen_size (display)

  This function provides the width and the height of the screen.
- def Eye-Bit Project.Trecker_with6acord.domain (x, y, s_w, s_h)

  The function estimates the rectangle that includes the given point.
- def Eye-Bit Project.Trecker_with6acord.drowLine (cord, orient, size)

  The function provides the coordinates of the line.
- def Eye-Bit Project.Trecker_with6acord.talker ()

  This function tracks the user's gaze.