

# augmented-reality-drawing

## 0.4

Generated by Doxygen 1.7.4

Tue Apr 17 2012 23:17:11



# Contents

<b>1</b>	<b>Class Index</b>	<b>1</b>
1.1	Class List . . . . .	1
<b>2</b>	<b>File Index</b>	<b>3</b>
2.1	File List . . . . .	3
<b>3</b>	<b>Class Documentation</b>	<b>5</b>
3.1	Cursor Struct Reference . . . . .	5
3.1.1	Detailed Description . . . . .	5
3.1.2	Member Data Documentation . . . . .	5
3.1.2.1	active . . . . .	5
3.1.2.2	area . . . . .	6
3.1.2.3	center . . . . .	6
3.1.2.4	color . . . . .	6
3.1.2.5	cornerA . . . . .	6
3.1.2.6	cornerB . . . . .	6
3.1.2.7	flag . . . . .	6
3.1.2.8	mask . . . . .	6
3.1.2.9	threshold . . . . .	6
<b>4</b>	<b>File Documentation</b>	<b>7</b>
4.1	librairie/libtrack.hpp File Reference . . . . .	7
4.1.1	Detailed Description . . . . .	8
4.1.2	Enumeration Type Documentation . . . . .	9
4.1.2.1	TYPE_TRACK . . . . .	9
4.1.3	Function Documentation . . . . .	9

4.1.3.1	binarisation . . . . .	9
4.1.3.2	calibration . . . . .	9
4.1.3.3	track . . . . .	9

# Chapter 1

## Class Index

### 1.1 Class List

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

[Cursor](#) (Structure used to receive and sent data about the track ) . . . . . 5



## Chapter 2

# File Index

### 2.1 File List

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

librairie/[libtrack.hpp](#) (Tracking Library header ) . . . . . 7





## Chapter 3

# Class Documentation

### 3.1 Cursor Struct Reference

Structure used to receive and sent data about the track.

```
#include <libtrack.hpp>
```

#### Public Attributes

- CvPoint [center](#)
- CvPoint [cornerA](#)
- CvPoint [cornerB](#)
- unsigned int [area](#)
- CvScalar [color](#)
- IpImage \* [mask](#)
- unsigned int [threshold](#)
- bool [active](#)
- [TYPE\\_TRACK](#) flag

#### 3.1.1 Detailed Description

Structure used to receive and sent data about the track.

#### 3.1.2 Member Data Documentation

##### 3.1.2.1 bool Cursor::active

determine whether the paint is active or not

**3.1.2.2 unsigned int `Cursor::area`**

area of the cursor

**3.1.2.3 `CvPoint` `Cursor::center`**

center pixel of the object area

**3.1.2.4 `CvScalar` `Cursor::color`**

HSV color of binarisation

**3.1.2.5 `CvPoint` `Cursor::cornerA`**

Up-Left corner of the object area

**3.1.2.6 `CvPoint` `Cursor::cornerB`**

Down-Right corner of the object area

**3.1.2.7 `TYPE_TRACK` `Cursor::flag`**

type of tracking

**3.1.2.8 `IplImage*` `Cursor::mask`**

mask or template used for tracking.

**3.1.2.9 unsigned int `Cursor::threshold`**

value of the threshold used for calibration

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

- [librairie/libtrack.hpp](#)

## Chapter 4

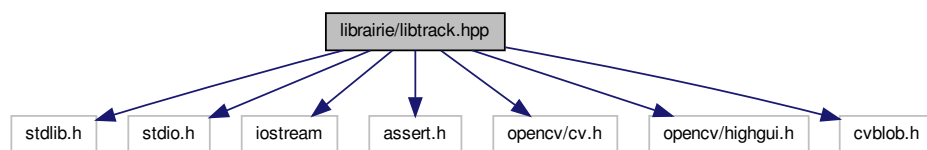
# File Documentation

### 4.1 librairie/libtrack.hpp File Reference

Tracking Library header.

```
#include <stdlib.h>
#include <stdio.h>
#include <iostream>
#include <assert.h>
#include <opencv/cv.h>
#include <opencv/highgui.h>
#include <cvblob.h>
```

Include dependency graph for libtrack.hpp:



### Classes

- struct [Cursor](#)

*Structure used to receive and sent data about the track.*

## Typedefs

- typedef struct [Cursor](#) **Cursor**

## Enumerations

- enum [TYPE\\_TRACK](#) { [TRACK\\_COLOR](#), [TRACK\\_SHAPE](#), [TRACK\\_BLOB](#) }  
*Used to chose the tracking method.*

## Functions

- [Cursor](#) \* [calibration](#) (IpImage \*source, CvPoint A, CvPoint B, [TYPE\\_TRACK](#) flag)  
*Initialize a structure for a TYPE\_TRACK Tracking.*
- int [track](#) (IpImage \*source, [Cursor](#) \*oldCursor)  
*Realize a TYPE\_TRACK Tracking, update the struct [Cursor](#).*
- [Cursor](#) \* [initColorTrack](#) (IpImage \*source, CvPoint A, CvPoint B)
- [Cursor](#) \* [initBlobTrack](#) (IpImage \*source, CvPoint A, CvPoint B)
- [Cursor](#) \* [initShapeTrack](#) (IpImage \*source, CvPoint A, CvPoint B)
- int [colorTrack](#) (IpImage \*source, [Cursor](#) \*oldCursor)
- int [blobTrack](#) (IpImage \*source, [Cursor](#) \*oldCursor)
- int [shapeTrack](#) (IpImage \*source, [Cursor](#) \*oldCursor)
- int [binarisation](#) (IpImage \*source, [Cursor](#) \*oldCursor)  
*Update the mask in oldCursor with the source IpImage.*
- CvScalar [colorAverage](#) (IpImage \*hsv, CvPoint A, CvPoint B)
- CvScalar [sampledColorAverage](#) (IpImage \*udrlmg, int nbPixels)
- CvScalar [mainColor](#) (IpImage \*hsv, CvPoint A, CvPoint B)
- CvPoint [center](#) (CvPoint A, CvPoint B)
- int [blobFounding](#) (IpImage \*source, [Cursor](#) \*oldCursor)
- IpImage \* [reshape](#) (IpImage \*source, CvRect roi)
- CvRect [underROI](#) (CvRect fullRect, int ratio)
- int [setNewCoord](#) ([Cursor](#) \*oldCursor)

### 4.1.1 Detailed Description

Tracking Library header.

#### Author

Pouer

#### Version

r150

#### Date

04/2012

Library used to realise an object tracking in a video stream

## 4.1.2 Enumeration Type Documentation

### 4.1.2.1 enum TYPE\_TRACK

Used to chose the tracking method.

#### Enumerator:

**TRACK\_COLOR** For a track based on the object color.

**TRACK\_SHAPE** For a track based on the object shape.

**TRACK\_BLOB** For a track based on the object.

## 4.1.3 Function Documentation

### 4.1.3.1 int binarisation ( IplImage \* source, Cursor \* oldCursor )

Update the mask in oldCursor with the source IplImage.

#### Parameters

<i>source</i>	: The colored source image you want to binarise
<i>oldCursor</i>	: structure to update, containing all information

#### Returns

0 if success, -1 if failure.

TODO : technical description...

### 4.1.3.2 Cursor\* calibration ( IplImage \* source, CvPoint A, CvPoint B, TYPE\_TRACK flag )

Initialize a structure for a TYPE\_TRACK Tracking.

#### Parameters

<i>source</i>	: Image from which apply the calibration
<i>A</i>	: one of the two pixels defining the object area to track
<i>B</i>	: one of the two pixels defining the object area to track
<i>flag</i>	: determine the tracking method to use.

#### Returns

A [Cursor](#) \* structure containing the track informations

### 4.1.3.3 int track ( IplImage \* source, Cursor \* oldCursor )

Realize a TYPE\_TRACK Tracking, update the struct [Cursor](#).

**Parameters**

<i>source</i>	: Image from which apply the track
<i>oldCursor</i>	: structure to update, containing all information

**Returns**

0 if success, -1 if failure.

# Index

- active
  - Cursor, [5](#)
- area
  - Cursor, [5](#)
- binarisation
  - libtrack.hpp, [9](#)
- calibration
  - libtrack.hpp, [9](#)
- center
  - Cursor, [6](#)
- color
  - Cursor, [6](#)
- cornerA
  - Cursor, [6](#)
- cornerB
  - Cursor, [6](#)
- Cursor, [5](#)
  - active, [5](#)
  - area, [5](#)
  - center, [6](#)
  - color, [6](#)
  - cornerA, [6](#)
  - cornerB, [6](#)
  - flag, [6](#)
  - mask, [6](#)
  - threshold, [6](#)
- flag
  - Cursor, [6](#)
- librairie/libtrack.hpp, [7](#)
- libtrack.hpp
  - binarisation, [9](#)
  - calibration, [9](#)
  - track, [9](#)
  - TRACK\_BLOB, [9](#)
  - TRACK\_COLOR, [9](#)
  - TRACK\_SHAPE, [9](#)
  - TYPE\_TRACK, [9](#)
- mask
  - Cursor, [6](#)
- threshold
  - Cursor, [6](#)
- track
  - libtrack.hpp, [9](#)
- TRACK\_BLOB
  - libtrack.hpp, [9](#)
- TRACK\_COLOR
  - libtrack.hpp, [9](#)
- TRACK\_SHAPE
  - libtrack.hpp, [9](#)
- TYPE\_TRACK
  - libtrack.hpp, [9](#)