

OpenCV Lib

Generated by Doxygen 1.8.11



# 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	CamCtrl Class Reference . . . . .	5
3.1.1	Constructor & Destructor Documentation . . . . .	5
3.1.1.1	CamCtrl() . . . . .	5
3.1.2	Member Function Documentation . . . . .	5
3.1.2.1	opnCam(int CamIndex=-1) . . . . .	5
3.1.2.2	saveVideo(CvCapture *pCapture_n, char *FilePath_n) . . . . .	5
3.2	KNNClass Class Reference . . . . .	6
3.2.1	Constructor & Destructor Documentation . . . . .	6
3.2.1.1	KNNClass(int k_n=5, int SmpClass_n=5, int ClassQty_n=200, int SmpSize_↔ n=400) . . . . .	6
3.2.1.2	~KNNClass() . . . . .	6
3.2.2	Member Function Documentation . . . . .	6
3.2.2.1	classify(char *FilePath_n) . . . . .	6
3.2.2.2	enhceData(char *SrcFilePath_n, char *DstFilePath_n, int SmpClass_n=5, int ClassQty_n=20) . . . . .	7
3.2.2.3	loadData(char *FilePath_n, int SmpClass_n, int ClassQty_n) . . . . .	7
3.2.2.4	selfTest(char *FilePath_n, int TestClass, int Max) . . . . .	7
3.2.2.5	setClassQty(int n) . . . . .	8

3.2.2.6	setFilePath(char *FilePath_n)	8
3.2.2.7	setK(int n)	8
3.2.2.8	setSmpClass(int n)	8
3.2.2.9	setSmpSize(int n)	8
3.2.2.10	setThresA(int n)	9
3.2.2.11	setThresB(int n)	9
3.3	ObjectCnt Class Reference	9
3.3.1	Constructor & Destructor Documentation	9
3.3.1.1	ObjectCnt()	9
3.3.1.2	~ObjectCnt()	9
3.3.2	Member Function Documentation	9
3.3.2.1	cntObjects(char *pFilePath)	9
3.3.2.2	cntObjects(IplImage *pSrcImg)	10
3.3.2.3	cntPrePrs(IplImage *pSrcImg)	10
3.3.2.4	objPrtPara()	10
4	File Documentation	11
4.1	/Users/Leakey/git/CVLib/CVLib/CVLib.cpp File Reference	11
4.1.1	Function Documentation	12
4.1.1.1	autoStcer(IplImage *SrcImgQueen[], int lenth)	12
4.1.1.2	cvGetSubImage(IplImage *SrcImg, CvRect Roi)	12
4.1.1.3	findMinRect(IplImage *SrcImg)	13
4.1.1.4	findRegion(IplImage *SrcImg, int *x, int *y, int Flag)	13
4.1.1.5	findX(IplImage *SrcImg, int *Min, int *Max)	13
4.1.1.6	findXRange(IplImage *SrcImg, int *Min, int *Max, int Flag)	14
4.1.1.7	findY(IplImage *SrcImg, int *Min, int *Max)	14
4.1.1.8	findYRange(IplImage *SrcImg, int *Min, int *Max, int Flag)	14
4.1.1.9	getThreshold(IplImage *pSrcImg, int *TA, int *TB)	14
4.1.1.10	getThreshold(char *FilePath_n, int *TA, int *TB)	15
4.1.1.11	InsertSort(unsigned char a[], int count)	15
4.1.1.12	max_Trackbar(int Threshold)	15

4.1.1.13	min_Trackbar(int Threshold)	15
4.1.1.14	picStcer(IplImage *SrcImgQue[], int lenth, int mode)	15
4.1.1.15	picStcer_sub(IplImage *ImgA, IplImage *ImgB, int mode)	16
4.1.1.16	prePrs(IplImage *SrcImg, int W_n, int H_n, int Mode)	16
4.1.1.17	resizeImg(char *FilePath_n, CvSize Size_n)	16
4.1.1.18	setDilate(int pDilate)	17
4.1.1.19	setErode(int pErode)	17
4.1.1.20	setKnlSize(int pKnlSize)	17
4.1.1.21	setTa(int Threshold)	17
4.1.1.22	setTb(int Threshold)	17
4.1.1.23	showImg(int TA, int TB)	18
4.1.1.24	showImg(int TA, int TB, int pKnlSize, int pErode, int pDilate)	18
4.1.2	Variable Documentation	18
4.1.2.1	CannyImg	18
4.1.2.2	Explmg	18
4.1.2.3	GrayImg	18
4.2	/Users/Leakey/git/CVLib/CVLib/CVLib.h File Reference	18
4.2.1	Macro Definition Documentation	20
4.2.1.1	HORIZONTAL	20
4.2.1.2	VERTICAL	20
4.2.2	Function Documentation	20
4.2.2.1	autoStcer(IplImage *SrcImgQueen[], int lenth)	20
4.2.2.2	cvGetSubImage(IplImage *SrcImg, CvRect Roi)	20
4.2.2.3	findMinRect(IplImage *SrcImg)	21
4.2.2.4	findRegion(IplImage *SrcImg, int *x, int *y, int Flag=1)	21
4.2.2.5	findX(IplImage *SrcImg, int *Min, int *Max)	21
4.2.2.6	findXRange(IplImage *SrcImg, int *Min, int *Max, int Flag=1)	21
4.2.2.7	findY(IplImage *SrcImg, int *Min, int *Max)	22
4.2.2.8	findYRange(IplImage *SrcImg, int *Min, int *Max, int Flag=1)	22
4.2.2.9	getThreshold(char *FilePath_n, int *TA, int *TB)	22

4.2.2.10	getThreshold(IplImage *pSrcImg, int *TA, int *TB)	23
4.2.2.11	InsertSort(unsigned char a[], int count)	23
4.2.2.12	max_Trackbar(int Threshold)	23
4.2.2.13	min_Trackbar(int Threshold)	23
4.2.2.14	picStcer(IplImage *SrcImgQueen[], int lenth, int mode)	23
4.2.2.15	picStcer_sub(IplImage *ImgA, IplImage *ImgB, int mode)	24
4.2.2.16	prePrs(IplImage *SrcImg, int W_n, int H_n, int Mode=1)	24
4.2.2.17	resizeImg(char *FilePath_n, CvSize Size_n)	24
4.2.2.18	setDilate(int pDilate)	25
4.2.2.19	setErode(int pErode)	25
4.2.2.20	setKnlSize(int pKnlSize)	25
4.2.2.21	setTa(int Threshold)	25
4.2.2.22	setTb(int Threshold)	25
4.2.2.23	showImg(int TA, int TB)	26
4.2.2.24	showImg(int TA, int TB, int pKnlSize, int pErode, int pDilate)	26

# Chapter 1

## Class Index

### 1.1 Class List

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

<a href="#">CamCtrl</a>	.....	5
<a href="#">KNNClass</a>	.....	6
<a href="#">ObjectCnt</a>	.....	9





## Chapter 2

# File Index

### 2.1 File List

Here is a list of all files with brief descriptions:

<a href="#">/Users/Leakey/git/CVLib/CVLib/CVLib.cpp</a> . . . . .	11
<a href="#">/Users/Leakey/git/CVLib/CVLib/CVLib.h</a> . . . . .	18



## Chapter 3

# Class Documentation

### 3.1 CamCtrl Class Reference

```
#include <CVLib.h>
```

#### Public Member Functions

- [CamCtrl](#) ()
- void [opnCam](#) (int CamIndex=-1)  
*Open camera and display frame.*
- void [saveVideo](#) (CvCapture \*pCapture\_n, char \*FilePath\_n)  
*Save frames in camera captureed frame.*

#### 3.1.1 Constructor & Destructor Documentation

##### 3.1.1.1 CamCtrl::CamCtrl ( )

#### 3.1.2 Member Function Documentation

##### 3.1.2.1 void CamCtrl::opnCam ( int *CamIndex* = -1 )

Open camera and display frame.

#### Parameters

<i>CamIndex</i>	Camera index
-----------------	--------------

##### 3.1.2.2 void CamCtrl::saveVideo ( CvCapture \* *pCapture\_n*, char \* *FilePath\_n* )

Save frames in camera captureed frame.

## Parameters

<i>pCapture</i> ↔ _n	Video ptr
<i>FilePath</i> _n	Save path

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

- /Users/Leakey/git/CVLib/CVLib/CVLib.h
- /Users/Leakey/git/CVLib/CVLib/CVLib.cpp

## 3.2 KNNClass Class Reference

```
#include <CVLib.h>
```

### Public Member Functions

- [KNNClass](#) (int *k*\_n=5, int *SmpClass*\_n=5, int *ClassQty*\_n=200, int *SmpSize*\_n=400)
- [~KNNClass](#) ()
- void [setK](#) (int n)  
*Set K number.*
- void [setThresA](#) (int n)  
*Set threshold.*
- void [setThresB](#) (int n)  
*Set threshold.*
- void [setSmpSize](#) (int n)  
*Set sample data's size.*
- void [setSmpClass](#) (int n)  
*Set sample data's classes.*
- void [setClassQty](#) (int n)  
*Set number of data in each class.*
- void [setFilePath](#) (char \**FilePath*\_n)  
*Set root folder path.*
- float [classify](#) (char \**FilePath*\_n)  
*Classify a pic.*
- void [selfTest](#) (char \**FilePath*\_n, int *TestClass*, int *Max*)  
*System test.*
- void [loadData](#) (char \**FilePath*\_n, int *SmpClass*\_n, int *ClassQty*\_n)  
*Load local data.*
- void [enhceData](#) (char \**SrcFilePath*\_n, char \**DstFilePath*\_n, int *SmpClass*\_n=5, int *ClassQty*\_n=20)  
*Copy data to increase sum number of data.*

### 3.2.1 Constructor & Destructor Documentation

3.2.1.1 [KNNClass::KNNClass](#) ( int *k*\_n = 5, int *SmpClass*\_n = 5, int *ClassQty*\_n = 200, int *SmpSize*\_n = 400 )

3.2.1.2 [KNNClass::~~KNNClass](#) ( )

### 3.2.2 Member Function Documentation

3.2.2.1 [float KNNClass::classify](#) ( char \* *FilePath*\_n )

Classify a pic.

## Parameters

<i>FilePath</i> ↔ _n	Pic path
-------------------------	----------

## Returns

Class of pic

**3.2.2.2** void KNNClass::enhceData ( char \* *SrcFilePath\_n*, char \* *DstFilePath\_n*, int *SmpClass\_n* = 5, int *ClassQty\_n* = 20 )

Copy data to increase sum number of data.

## Parameters

<i>SrcFilePath</i> ↔ _n	Data path
<i>DstFilePath</i> ↔ _n	Copied data path
<i>SmpClass_n</i>	Number of copied sample classes
<i>ClassQty_n</i>	Number of copied data in each class

**3.2.2.3** void KNNClass::loadData ( char \* *FilePath\_n*, int *SmpClass\_n*, int *ClassQty\_n* )

Load local data.

## Parameters

<i>FilePath_n</i>	Data path
<i>SmpClass</i> ↔ _n	Number of sample classes
<i>ClassQty</i> ↔ _n	Number of data in each calss

**3.2.2.4** void KNNClass::selfTest ( char \* *FilePath\_n*, int *TestClass*, int *Max* )

System test.

## Parameters

<i>FilePath</i> ↔ _n	Root folder path
<i>TestClass</i>	Number of sample classes
<i>Max</i>	Number of data in each class

### 3.2.2.5 void KNNClass::setClassQty ( int *n* )

Set number of data in each class.

Parameters

<i>n</i>	Number of data in each calss
----------	------------------------------

### 3.2.2.6 void KNNClass::setFilePath ( char \* *FilePath\_n* )

Set root folder path.

Parameters

<i>FilePath_n</i>	Root folder path
-------------------	------------------

### 3.2.2.7 void KNNClass::setK ( int *n* )

Set K number.

Parameters

<i>n</i>	Number
----------	--------

### 3.2.2.8 void KNNClass::setSmpClass ( int *n* )

Set sample data's classes.

Parameters

<i>n</i>	Number of sample classes
----------	--------------------------

### 3.2.2.9 void KNNClass::setSmpSize ( int *n* )

Set sample data's size.

Parameters

<i>n</i>	Data size(width=heigh)
----------	------------------------

3.2.2.10 void KNNClass::setThresA ( int *n* )

Set threshold.

Parameters

<i>n</i>	Threshold
----------	-----------

3.2.2.11 void KNNClass::setThresB ( int *n* )

Set threshold.

Parameters

<i>n</i>	Threshold
----------	-----------

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

- /Users/Leakey/git/CVLib/CVLib/[CVLib.h](#)
- /Users/Leakey/git/CVLib/CVLib/[CVLib.cpp](#)

## 3.3 ObjectCnt Class Reference

```
#include <CVLib.h>
```

### Public Member Functions

- [ObjectCnt](#) ()
- [~ObjectCnt](#) ()
- int [cntObjects](#) (char \*pFilePath)  
*Load a pic and count objects.*
- int [cntObjects](#) (IplImage \*pSrcImg)  
*Count objects in image.*
- IplImage \* [cntPrePrs](#) (IplImage \*pSrcImg)  
*Pre-Process image.*
- void [objPrtPara](#) ()  
*Print para.*

### 3.3.1 Constructor & Destructor Documentation

## 3.3.1.1 ObjectCnt::ObjectCnt ( )

## 3.3.1.2 ObjectCnt::~~ObjectCnt ( )

### 3.3.2 Member Function Documentation

3.3.2.1 int ObjectCnt::cntObjects ( char \* *pFilePath* )

Load a pic and count objects.

**Parameters**

<i>pFilePath</i>	Pic path
------------------	----------

**Returns**

Number of object

**3.3.2.2 int ObjectCnt::cntObjects ( IplImage \* pSrcImg )**

Count objects in image.

**Parameters**

<i>pSrcImg</i>	Image ptr
----------------	-----------

**Returns**

Number of object

**3.3.2.3 IplImage \* ObjectCnt::cntPrePrs ( IplImage \* pSrcImg )**

Pre-Process image.

**Parameters**

<i>pSrcImg</i>	Image ptr
----------------	-----------

**Returns**

Processed image

**3.3.2.4 void ObjectCnt::objPrtPara ( )**

Print para.

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

- /Users/Leakey/git/CVLib/CVLib/[CVLib.h](#)
- /Users/Leakey/git/CVLib/CVLib/[CVLib.cpp](#)



## Chapter 4

# File Documentation

### 4.1 /Users/Leakey/git/CVLib/CVLib/CVLib.cpp File Reference

```
#include "CVLib.h"
```

#### Functions

- void [findX](#) (IplImage \*SrcImg, int \*Min, int \*Max)  
*Find the first and last effective pixel in image.*
- void [findY](#) (IplImage \*SrcImg, int \*Min, int \*Max)  
*Find the first and last effective pixel in image.*
- CvRect [findMinRect](#) (IplImage \*SrcImg)  
*Get a pic's min non-empty pixel region.*
- IplImage \* [cvGetSubImage](#) (IplImage \*SrcImg, CvRect Roi)  
*Crop image from a ROI.*
- IplImage \* [prePrs](#) (IplImage \*SrcImg, int W\_n, int H\_n, int Mode)  
*Prepress a pic, get a min binary region. 1 for turn pixel into ~.*
- IplImage \* [resizeImg](#) (char \*FilePath\_n, CvSize Size\_n)  
*Resize pic into setted size.*
- void [showImg](#) (int TA, int TB)  
*Display image in window with para threshold setted.*
- void [min\\_Trackbar](#) (int Threshold)  
*Set threshold, also as a callback func.*
- void [max\\_Trackbar](#) (int Threshold)  
*Set threshold, also as a callback func.*
- void [getThreshold](#) (IplImage \*pSrcImg, int \*TA, int \*TB)  
*Create a window to display image and modify threshold.*
- void [getThreshold](#) (char \*FilePath\_n, int \*TA, int \*TB)  
*Create a window to display image and modify threshold.*
- void [findRegion](#) (IplImage \*SrcImg, int \*x, int \*y, int Flag)  
*Find the pixel region for a binary pic.*
- void [findXRange](#) (IplImage \*SrcImg, int \*Min, int \*Max, int Flag)  
*Find the first and last effective pixel in image.*
- void [findYRange](#) (IplImage \*SrcImg, int \*Min, int \*Max, int Flag)

- Find the first and last effective pixel in image.*

  - `IpImage * picStcer_sub (IpImage *ImgA, IpImage *ImgB, int mode)`

*Stitching pics simply.*
- `IpImage * picStcer (IpImage *SrcImgQue[], int lenth, int mode)`

*Stitching pics simply.*
- `IpImage * autoStcer (IpImage *SrcImgQueen[], int lenth)`

*Stitching pics with feature matching.*
- `void InsertSort (unsigned char a[], int count)`

*Sort algorithm.*
- `void showImg (int TA, int TB, int pKnlSize, int pErode, int pDilate)`

*Create a window to display image and modify paras.*
- `void setTa (int Threshold)`

*Set threshold.*
- `void setTb (int Threshold)`

*Set threshold.*
- `void setKnlSize (int pKnlSize)`

*Set kernel size.*
- `void setErode (int pErode)`

*Set erode times.*
- `void setDilate (int pDilate)`

*Set dilate times.*

## Variables

- `IpImage * GrayImg`
- `IpImage * CannyImg`
- `IpImage * Explmg`

## 4.1.1 Function Documentation

### 4.1.1.1 `IpImage* autoStcer ( IpImage * SrcImgQueen[], int lenth )`

Stitching pics with feature matching.

#### Parameters

<i>SrcImgQueen</i>	Image ptrs array
<i>lenth</i>	Sum width/heigh

#### Returns

Stitched image

### 4.1.1.2 `IpImage* cvGetSubImage ( IpImage * SrcImg, CvRect Roi )`

Crop image from a ROI.

## Parameters

<i>SrcImg</i>	Image ptr
<i>Roi</i>	ROI

## Returns

Cropped image

4.1.1.3 `CvRect findMinRect ( IplImage * SrcImg )`

Get a pic's min non-empty pixel region.

## Parameters

<i>SrcImg</i>	Image ptr
---------------	-----------

## Returns

Min non-empty pixel region

4.1.1.4 `void findRegion ( IplImage * SrcImg, int * x, int * y, int Flag = 1 )`

Find the pixel region for a binary pic.

## Parameters

<i>SrcImg</i>	SourcelImage
<i>x</i>	X pos
<i>y</i>	Y pos
<i>Flag</i>	White/Black background, 1 for white, 0 for black

4.1.1.5 `void findX ( IplImage * SrcImg, int * Min, int * Max )`

Find the first and last effective pixel in image.

## Parameters

<i>SrcImg</i>	Image ptr
<i>Min</i>	First X pos
<i>Max</i>	Last X pos

4.1.1.6 void findXRange ( *IpImage \* SrcImg*, int \* *Min*, int \* *Max*, int *Flag* = 1 )

Find the first and last effective pixel in image.

Parameters

<i>SrcImg</i>	Image ptr
<i>Min</i>	First X pos
<i>Max</i>	Last X pos
<i>Flag</i>	White/Black background,1 for white, 0 for black

4.1.1.7 void findY ( *IpImage \* SrcImg*, int \* *Min*, int \* *Max* )

Find the first and last effective pixel in image.

Parameters

<i>SrcImg</i>	Image ptr
<i>Min</i>	First Y pos
<i>Max</i>	Last Y pos

4.1.1.8 void findYRange ( *IpImage \* SrcImg*, int \* *Min*, int \* *Max*, int *Flag* = 1 )

Find the first and last effective pixel in image.

Parameters

<i>SrcImg</i>	Image ptr
<i>Min</i>	First Y pos
<i>Max</i>	Last Y pos
<i>Flag</i>	White/Black background,1 for white, 0 for black

4.1.1.9 void getThreshold ( *IpImage \* pSrcImg*, int \* *TA*, int \* *TB* )

Create a window to display image and modify threshold.

Parameters

<i>pSrcImg</i>	Image ptr
<i>TA</i>	Threshold A
<i>TB</i>	Threshold B

4.1.1.10 void getThreshold ( char \* *FilePath\_n*, int \* *TA*, int \* *TB* )

Create a window to display image and modify threshold.

Call a window and show a pic to ensure the threshold for binary a pic.

## Parameters

<i>pSrcImg</i>	Image path
<i>TA</i>	Threshold A
<i>TB</i>	Threshold B

4.1.1.11 void InsertSort ( unsigned char *a*[], int *count* )

Sort algorithm.

## Parameters

<i>a</i>	Char array
<i>count</i>	Lenth of char array

4.1.1.12 void max\_Trackbar ( int *Threshold* )

Set threshold, also as a callback func.

## Parameters

<i>Threshold</i>	Threshold needed
------------------	------------------

4.1.1.13 void min\_Trackbar ( int *Threshold* )

Set threshold, also as a callback func.

## Parameters

<i>Threshold</i>	Threshold needed
------------------	------------------

4.1.1.14 IpImage\* picStcer ( IpImage \* *SrcImgQueen*[], int *lenth*, int *mode* )

Stitching pics simply.

## Parameters

<i>SrcImgQueen</i>	Image ptrs array
<i>lenth</i>	Sum width/heigh
<i>mode</i>	Vertical/Horizontal, 1 for vertical, 0 for horizontal

**Returns**

Stitched image

4.1.1.15 `IpImage* picStcer_sub ( IpImage * ImgA, IpImage * ImgB, int mode )`

Stitching pics simply.

**Parameters**

<i>ImgA</i>	Image ptr A
<i>ImgB</i>	Image ptr B
<i>mode</i>	Vertical/Horizontal, 1 for vertical, 0 for horizontal

**Returns**

Stitched image

4.1.1.16 `IpImage* prePrs ( IpImage * SrcImg, int W_n, int H_n, int Mode = 1 )`

Prepress a pic, get a min binary region. 1 for turn pixel into ~.

**Parameters**

<i>SrcImg</i>	Image ptr
<i>W_n</i>	Expectant width
<i>H_n</i>	Expectant heigh
<i>Mode</i>	White/Black background

**Returns**

Processed image

4.1.1.17 `IpImage* resizelmg ( char * FilePath_n, CvSize Size_n )`

Resize pic into settet size.

**Parameters**

<i>FilePath↵ _n</i>	Pic path
<i>Size_n</i>	Expectant height*width

**Returns**

Resized pic

**4.1.1.18 void setDilate ( int *pDilate* )**

Set dilate times.

**Parameters**

<i>pDilate</i>	Dilate times
----------------	--------------

**4.1.1.19 void setErode ( int *pErode* )**

Set erode times.

**Parameters**

<i>pErode</i>	Erode times
---------------	-------------

**4.1.1.20 void setKnISize ( int *pKnISize* )**

Set kernel size.

**Parameters**

<i>pKnISize</i>	Kernel size(width=heigh)
-----------------	--------------------------

**4.1.1.21 void setTa ( int *Threshold* )**

Set threshold.

**Parameters**

<i>Threshold</i>	Threshold
------------------	-----------

**4.1.1.22 void setTb ( int *Threshold* )**

Set threshold.

**Parameters**

<i>Threshold</i>	Threshold
------------------	-----------

#### 4.1.1.23 void showImg ( int *TA*, int *TB* )

Display image in window with para threshold setted.

##### Parameters

<i>TA</i>	Threshold A
<i>TB</i>	Threshold B

#### 4.1.1.24 void showImg ( int *TA*, int *TB*, int *pKnlSize*, int *pErode*, int *pDilate* )

Create a window to display image and modify paras.

##### Parameters

<i>TA</i>	Threshold A
<i>TB</i>	Threshold B
<i>pKnlSize</i>	Kernel size
<i>pErode</i>	Erode times
<i>pDilate</i>	Dilate times

## 4.1.2 Variable Documentation

#### 4.1.2.1 IpImage\* CannyImg

#### 4.1.2.2 IpImage\* Explmg

#### 4.1.2.3 IpImage\* GrayImg

## 4.2 /Users/Leakey/git/CVLib/CVLib/CVLib.h File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <fstream>
#include <iostream>
#include <cv.h>
#include <ml.h>
#include <math.h>
#include <highgui.h>
#include "opencv2/highgui/highgui.hpp"
#include "opencv2/stitching/stitcher.hpp"
#include <opencv2/opencv.hpp>
#include <imgproc/imgproc.hpp>
```



## Classes

- class [CamCtrl](#)
- class [KNNClass](#)
- class [ObjectCnt](#)

## Macros

- #define [VERTICAL](#) 1
- #define [HORIZONTAL](#) 0

## Functions

- [IpImage \\* resizeImg](#) (char \*FilePath\_n, CvSize Size\_n)  
*Resize pic into setted size.*
- [IpImage \\* cvGetSubImage](#) (IpImage \*SrcImg, CvRect Roi)  
*Crop image from a ROI.*
- void [InsertSort](#) (unsigned char a[], int count)  
*Sort algorithm.*
- CvRect [findMinRect](#) (IpImage \*SrcImg)  
*Get a pic's min non-empty pixel region.*
- void [findRegion](#) (IpImage \*SrcImg, int \*x, int \*y, int Flag=1)  
*Find the pixel region for a binary pic.*
- void [findXRange](#) (IpImage \*SrcImg, int \*Min, int \*Max, int Flag=1)  
*Find the first and last effective pixel in image.*
- void [findYRange](#) (IpImage \*SrcImg, int \*Min, int \*Max, int Flag=1)  
*Find the first and last effective pixel in image.*
- void [findX](#) (IpImage \*SrcImg, int \*Min, int \*Max)  
*Find the first and last effective pixel in image.*
- void [findY](#) (IpImage \*SrcImg, int \*Min, int \*Max)  
*Find the first and last effective pixel in image.*
- [IpImage \\* prePrs](#) (IpImage \*SrcImg, int W\_n, int H\_n, int Mode=1)  
*Prepress a pic, get a min binary region. 1 for turn pixel into ~.*
- void [getThreshold](#) (char \*FilePath\_n, int \*TA, int \*TB)  
*Call a window and show a pic to ensure the threshold for binary a pic.*
- void [min\\_Trackbar](#) (int Threshold)  
*Set threshold, also as a callback func.*
- void [max\\_Trackbar](#) (int Threshold)  
*Set threshold, also as a callback func.*
- void [getThreshold](#) (IpImage \*pSrcImg, int \*TA, int \*TB)  
*Create a window to display image and modify threshold.*
- void [showImg](#) (int TA, int TB)  
*Display image in window with para threshold setted.*
- [IpImage \\* picStcer](#) (IpImage \*SrcImgQueen[], int lenth, int mode)  
*Stitching pics simply.*
- [IpImage \\* autoStcer](#) (IpImage \*SrcImgQueen[], int lenth)  
*Stitching pics with feature matching.*
- [IpImage \\* picStcer\\_sub](#) (IpImage \*ImgA, IpImage \*ImgB, int mode)  
*Stitching pics simply.*
- void [showImg](#) (int TA, int TB, int pKnlSize, int pErode, int pDilate)

Create a window to display image and modify paras.

- void [setTa](#) (int Threshold)  
Set threshold.
- void [setTb](#) (int Threshold)  
Set threshold.
- void [setKnlSize](#) (int pKnlSize)  
Set kernel size.
- void [setErode](#) (int pErode)  
Set erode times.
- void [setDilate](#) (int pDilate)  
Set dilate times.

## 4.2.1 Macro Definition Documentation

4.2.1.1 `#define HORIZONTAL 0`

4.2.1.2 `#define VERTICAL 1`

## 4.2.2 Function Documentation

4.2.2.1 `IplImage* autoStcer ( IplImage * SrcImgQueen[], int lenth )`

Stitching pics with feature matching.

Parameters

<i>SrcImgQueen</i>	Image ptrs array
<i>lenth</i>	Sum width/heigh

Returns

Stitched image

4.2.2.2 `IplImage* cvGetSubImage ( IplImage * SrcImg, CvRect Roi )`

Crop image from a ROI.

Parameters

<i>SrcImg</i>	Image ptr
<i>Roi</i>	ROI

Returns

Cropped image

#### 4.2.2.3 CvRect findMinRect ( IplImage \* *SrcImg* )

Get a pic's min non-empty pixel region.

##### Parameters

<i>SrcImg</i>	Image ptr
---------------	-----------

##### Returns

Min non-empty pixel region

#### 4.2.2.4 void findRegion ( IplImage \* *SrcImg*, int \* *x*, int \* *y*, int *Flag* = 1 )

Find the pixel region for a binary pic.

##### Parameters

<i>SrcImg</i>	SourcelImage
<i>x</i>	X pos
<i>y</i>	Y pos
<i>Flag</i>	White/Black background,1 for white, 0 for black

#### 4.2.2.5 void findX ( IplImage \* *SrcImg*, int \* *Min*, int \* *Max* )

Find the first and last effective pixel in image.

##### Parameters

<i>SrcImg</i>	Image ptr
<i>Min</i>	First X pos
<i>Max</i>	Last X pos

#### 4.2.2.6 void findXRange ( IplImage \* *SrcImg*, int \* *Min*, int \* *Max*, int *Flag* = 1 )

Find the first and last effective pixel in image.

##### Parameters

<i>SrcImg</i>	Image ptr
<i>Min</i>	First X pos
<i>Max</i>	Last X pos
<i>Flag</i>	White/Black background,1 for white, 0 for black

#### 4.2.2.7 void findY ( IplImage \* *SrcImg*, int \* *Min*, int \* *Max* )

Find the first and last effective pixel in image.

##### Parameters

<i>SrcImg</i>	Image ptr
<i>Min</i>	First Y pos
<i>Max</i>	Last Y pos

#### 4.2.2.8 void findYRange ( IplImage \* *SrcImg*, int \* *Min*, int \* *Max*, int *Flag* = 1 )

Find the first and last effective pixel in image.

##### Parameters

<i>SrcImg</i>	Image ptr
<i>Min</i>	First Y pos
<i>Max</i>	Last Y pos
<i>Flag</i>	White/Black background,1 for white, 0 for black

#### 4.2.2.9 void getThreshold ( char \* *FilePath\_n*, int \* *TA*, int \* *TB* )

Call a window and show a pic to ensure the threshold for binary a pic.

Create a window to display image and modify threshold.

##### Parameters

<i>FilePath↔ _n</i>	Image path
<i>TA</i>	Threshold A
<i>TB</i>	Threshold B
<i>pSrcImg</i>	Image path
<i>TA</i>	Threshold A
<i>TB</i>	Threshold B

Call a window and show a pic to ensure the threshold for binary a pic.

##### Parameters

<i>pSrcImg</i>	Image path
<i>TA</i>	Threshold A
<i>TB</i>	Threshold B

**4.2.2.10 void getThreshold ( *IplImage* \* *pSrcImg*, int \* *TA*, int \* *TB* )**

Create a window to display image and modify threshold.

**Parameters**

<i>pSrcImg</i>	Image ptr
<i>TA</i>	Threshold A
<i>TB</i>	Threshold B

**4.2.2.11 void InsertSort ( unsigned char *a*[], int *count* )**

Sort algorithm.

**Parameters**

<i>a</i>	Char array
<i>count</i>	Lenth of char array

**4.2.2.12 void max\_Trackbar ( int *Threshold* )**

Set threshold, also as a callback func.

**Parameters**

<i>Threshold</i>	Threshold needed
------------------	------------------

**4.2.2.13 void min\_Trackbar ( int *Threshold* )**

Set threshold, also as a callback func.

**Parameters**

<i>Threshold</i>	Threshold needed
------------------	------------------

**4.2.2.14 *IplImage*\* picStcer ( *IplImage* \* *SrcImgQueen*[], int *lenth*, int *mode* )**

Stitching pics simply.

**Parameters**

<i>SrcImgQueen</i>	Image ptrs array
<i>lenth</i>	Sum width/heigh
<i>mode</i>	Vertical/Horizontal, 1 for vertical, 0 for horizontal

**Returns**

Stitched image

#### 4.2.2.15 `IpImage* picStcer_sub ( IpImage * ImgA, IpImage * ImgB, int mode )`

Stitching pics simply.

**Parameters**

<i>ImgA</i>	Image ptr A
<i>ImgB</i>	Image ptr B
<i>mode</i>	Vertical/Horizontal, 1 for vertical, 0 for horizontal

**Returns**

Stitched image

#### 4.2.2.16 `IpImage* prePrs ( IpImage * SrcImg, int W_n, int H_n, int Mode = 1 )`

Prepress a pic, get a min binary region. 1 for turn pixel into ~.

**Parameters**

<i>SrcImg</i>	Image ptr
<i>W_n</i>	Expectant width
<i>H_n</i>	Expectant heigh
<i>Mode</i>	White/Black background

**Returns**

Processed image

#### 4.2.2.17 `IpImage* resizelmg ( char * FilePath_n, CvSize Size_n )`

Resize pic into settet size.

**Parameters**

<i>FilePath↵ _n</i>	Pic path
<i>Size_n</i>	Expectant height*width

**Returns**

Resized pic

**4.2.2.18 void setDilate ( int *pDilate* )**

Set dilate times.

**Parameters**

<i>pDilate</i>	Dilate times
----------------	--------------

**4.2.2.19 void setErode ( int *pErode* )**

Set erode times.

**Parameters**

<i>pErode</i>	Erode times
---------------	-------------

**4.2.2.20 void setKnISize ( int *pKnISize* )**

Set kernel size.

**Parameters**

<i>pKnISize</i>	Kernel size(width=heigh)
-----------------	--------------------------

**4.2.2.21 void setTa ( int *Threshold* )**

Set threshold.

**Parameters**

<i>Threshold</i>	Threshold
------------------	-----------

**4.2.2.22 void setTb ( int *Threshold* )**

Set threshold.

**Parameters**

<i>Threshold</i>	Threshold
------------------	-----------

#### 4.2.2.23 void showlmg ( int *TA*, int *TB* )

Display image in window with para threshold setted.

##### Parameters

<i>TA</i>	Threshold A
<i>TB</i>	Threshold B

#### 4.2.2.24 void showlmg ( int *TA*, int *TB*, int *pKnlSize*, int *pErode*, int *pDilate* )

Create a window to display image and modify paras.

##### Parameters

<i>TA</i>	Threshold A
<i>TB</i>	Threshold B
<i>pKnlSize</i>	Kernel size
<i>pErode</i>	Erode times
<i>pDilate</i>	Dilate times



# Index

/Users/Leakey/git/CVLib/CVLib/CVLib.cpp, [11](#)  
/Users/Leakey/git/CVLib/CVLib/CVLib.h, [18](#)  
~KNNClass  
    KNNClass, [6](#)  
~ObjectCnt  
    ObjectCnt, [9](#)  
  
autoStcer  
    CVLib.cpp, [12](#)  
    CVLib.h, [20](#)  
  
CVLib.cpp  
    autoStcer, [12](#)  
    CannyImg, [18](#)  
    cvGetSubImage, [12](#)  
    Explmg, [18](#)  
    findMinRect, [13](#)  
    findRegion, [13](#)  
    findXRange, [13](#)  
    findYRange, [14](#)  
    findX, [13](#)  
    findY, [14](#)  
    getThreshold, [14](#)  
    GrayImg, [18](#)  
    InsertSort, [15](#)  
    max\_Trackbar, [15](#)  
    min\_Trackbar, [15](#)  
    picStcer, [15](#)  
    picStcer\_sub, [16](#)  
    prePrs, [16](#)  
    resizeImg, [16](#)  
    setDilate, [17](#)  
    setErode, [17](#)  
    setKnlSize, [17](#)  
    setTa, [17](#)  
    setTb, [17](#)  
    showImg, [17](#), [18](#)  
  
CVLib.h  
    autoStcer, [20](#)  
    cvGetSubImage, [20](#)  
    findMinRect, [20](#)  
    findRegion, [21](#)  
    findXRange, [21](#)  
    findYRange, [22](#)  
    findX, [21](#)  
    findY, [21](#)  
    getThreshold, [22](#)  
    HORIZONTAL, [20](#)  
    InsertSort, [23](#)  
    max\_Trackbar, [23](#)  
  
    min\_Trackbar, [23](#)  
    picStcer, [23](#)  
    picStcer\_sub, [24](#)  
    prePrs, [24](#)  
    resizeImg, [24](#)  
    setDilate, [25](#)  
    setErode, [25](#)  
    setKnlSize, [25](#)  
    setTa, [25](#)  
    setTb, [25](#)  
    showImg, [25](#), [26](#)  
    VERTICAL, [20](#)  
  
CamCtrl, [5](#)  
    CamCtrl, [5](#)  
    opnCam, [5](#)  
    saveVideo, [5](#)  
  
CannyImg  
    CVLib.cpp, [18](#)  
  
classify  
    KNNClass, [6](#)  
  
cntObjects  
    ObjectCnt, [9](#), [10](#)  
  
cntPrePrs  
    ObjectCnt, [10](#)  
  
cvGetSubImage  
    CVLib.cpp, [12](#)  
    CVLib.h, [20](#)  
  
enhnceData  
    KNNClass, [7](#)  
  
Explmg  
    CVLib.cpp, [18](#)  
  
findMinRect  
    CVLib.cpp, [13](#)  
    CVLib.h, [20](#)  
  
findRegion  
    CVLib.cpp, [13](#)  
    CVLib.h, [21](#)  
  
findXRange  
    CVLib.cpp, [13](#)  
    CVLib.h, [21](#)  
  
findYRange  
    CVLib.cpp, [14](#)  
    CVLib.h, [22](#)  
  
findX  
    CVLib.cpp, [13](#)  
    CVLib.h, [21](#)  
  
findY  
    CVLib.cpp, [14](#)

- CVLib.h, 21
- getThreshold
  - CVLib.cpp, 14
  - CVLib.h, 22
- GrayImg
  - CVLib.cpp, 18
- HORIZONTAL
  - CVLib.h, 20
- InsertSort
  - CVLib.cpp, 15
  - CVLib.h, 23
- KNNClass, 6
  - ~KNNClass, 6
  - classify, 6
  - enhceData, 7
  - KNNClass, 6
  - loadData, 7
  - selfTest, 7
  - setClassQty, 7
  - setFilePath, 8
  - setSmpClass, 8
  - setSmpSize, 8
  - setThresA, 8
  - setThresB, 9
  - setK, 8
- loadData
  - KNNClass, 7
- max\_Trackbar
  - CVLib.cpp, 15
  - CVLib.h, 23
- min\_Trackbar
  - CVLib.cpp, 15
  - CVLib.h, 23
- objPrtPara
  - ObjectCnt, 10
- ObjectCnt, 9
  - ~ObjectCnt, 9
  - cntObjects, 9, 10
  - cntPrePrs, 10
  - objPrtPara, 10
  - ObjectCnt, 9
- opnCam
  - CamCtrl, 5
- picStcer
  - CVLib.cpp, 15
  - CVLib.h, 23
- picStcer\_sub
  - CVLib.cpp, 16
  - CVLib.h, 24
- prePrs
  - CVLib.cpp, 16
  - CVLib.h, 24
- resizeImg
  - CVLib.cpp, 16
  - CVLib.h, 24
- saveVideo
  - CamCtrl, 5
- selfTest
  - KNNClass, 7
- setClassQty
  - KNNClass, 7
- setDilate
  - CVLib.cpp, 17
  - CVLib.h, 25
- setErode
  - CVLib.cpp, 17
  - CVLib.h, 25
- setFilePath
  - KNNClass, 8
- setKnlSize
  - CVLib.cpp, 17
  - CVLib.h, 25
- setSmpClass
  - KNNClass, 8
- setSmpSize
  - KNNClass, 8
- setTa
  - CVLib.cpp, 17
  - CVLib.h, 25
- setTb
  - CVLib.cpp, 17
  - CVLib.h, 25
- setThresA
  - KNNClass, 8
- setThresB
  - KNNClass, 9
- setK
  - KNNClass, 8
- showImg
  - CVLib.cpp, 17, 18
  - CVLib.h, 25, 26
- VERTICAL
  - CVLib.h, 20